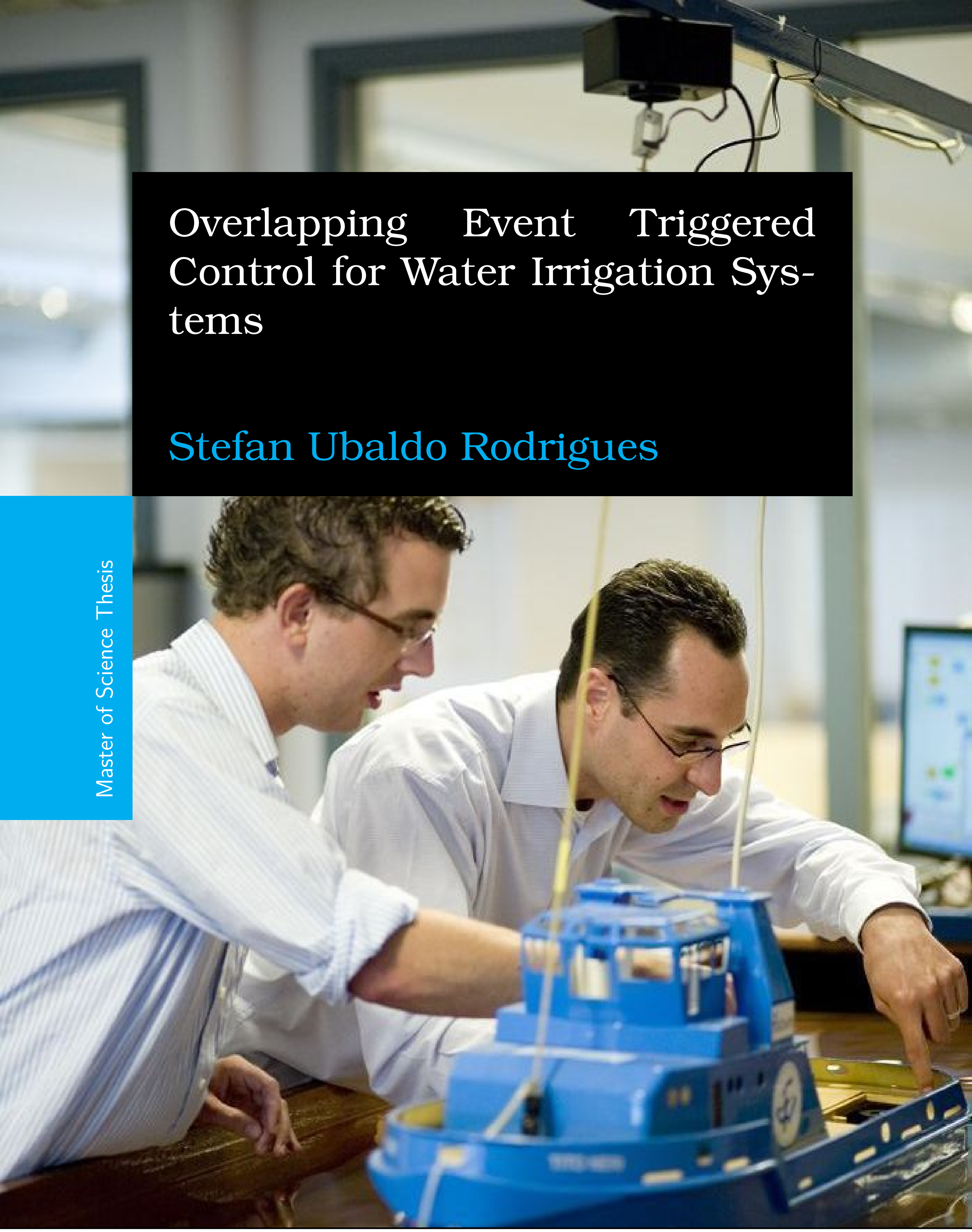


Overlapping Event Triggered Control for Water Irrigation Systems

Stefan Ubaldo Rodrigues

Master of Science Thesis

TU Delft Delft University of Technology

Delft Center for Systems and Control

Overlapping Event Triggered Control for Water Irrigation Systems

Master of Science Thesis

For the degree of Master of Science in Systems and Control at Delft University of Technology

Stefan Ubaldo Rodrigues

August 22, 2025

Faculty of Mechanical Engineering (ME) · Delft University of Technology

Abstract

Water scarcity is emerging as one of the most pressing global challenges, particularly in the context of agricultural irrigation, which consumes nearly 70 % of the world's freshwater, 40 % of which is wasted due to inefficient systems. Water Irrigation Systems (WISs), composed of interconnected open-channel networks, are vital for delivering water to farmland but often suffer from suboptimal performance due to decentralised control and lack of inter-pool communication. While centralised control can enhance efficiency, it is rarely scalable. This thesis will investigate overlapping control with neighbour-to-neighbour communication, supported by Networked Control Systems (NCS) and Event-Triggered Control (ETC) strategies, as an approach to optimise water and energy use. The research covers WIS modelling, controller architectures, system identification, and the integration of ETC, culminating in a proposed framework for smarter irrigation control.

Table of Contents

Preface and Acknowledgements ix

1 Introduction 1

1-1 Nomenclature . . . 2

2 Preliminaries 3

2-1 Water Irrigation Systems . . . 4

2-1-1 Simplified Model . . . 4

2-1-2 Local Loop Shaping . . . 5

2-2 Control Engineering . . . 7

2-2-1 Overlapping Control . . . 7

2-2-2 Event Triggered Control (ETC) . . . 8

2-2-3 LQR . . . 11

2-2-4 Observer . . . 11

2-2-5 Discretising . . . 12

2-3 Testbed Unit . . . 12

2-3-1 Layout . . . 12

2-3-2 Network Infrastructure . . . 13

2-4 Conclusion . . . 13

3 Control Design 15

3-1 Controller Formulation . . . 15

3-1-1 State Space Formulation . . . 16

3-1-2 Centralised Controller . . . 16

3-1-3 Decentralised Controller . . . 17

3-1-4 Overlapping Controller . . . 18

3-1-5 ETC overlapping controller . . . 22

3-2 Conclusion . . . 23

4 The Water Testbed Unit 25
4-1 Model . . . 25
4-1-1 Estimation of the Coefficients . . . 26
4-2 Gates . . . 28
4-2-1 Procedure . . . 30
4-2-2 Other Considerations . . . 30
4-2-3 Results . . . 31
4-3 Simulation . . . 33

5 Simulation Results 35
5-1 3-Pool Simulation . . . 35
5-1-1 Decentralised Controller . . . 35
5-1-2 Centralised Controller . . . 36
5-1-3 Overlapping Controller . . . 37
5-1-4 Comparisons . . . 37
5-1-5 Effects of Noise on the System . . . 40
5-2 6-Pool Simulation . . . 42
5-2-1 Effects of Noise . . . 44
5-3 Event triggered control . . . 46
5-3-1 3-Pool . . . 46
5-3-2 6-Pool . . . 49

6 Test Bed Results 53
6-1 Implementation . . . 53
6-2 Results . . . 54

7 Discussions and Conclusions 57
7-1 Conclusion . . . 57
7-2 Future Work . . . 58
7-2-1 Simulation . . . 58
7-2-2 Control . . . 58
7-2-3 Testbed . . . 59

A Simulink Model 61
A-1 3-Pool Simulator . . . 61
A-2 6-Pool Simulator . . . 61
A-3 Code . . . 61

B Proof 65

Bibliography 69

Glossary 73
List of Acronyms . . . 73
List of Symbols . . . 73

List of Figures

2-1 Water Irrigation System Topology 3

2-2 Cross Sectional profile of a WIS [18] 4

2-3 Diagram showing the interconnections between the pools [18] 5

2-4 Simplified closed-loop control of pool i with loop-shaping weight W_i and plant P_i, including reference r_i and disturbances v_i and d_i. 6

2-5 Schematic of Event-triggered Control [11] 9

2-6 Schematic of Decentralised Event-trigger control [11] 10

2-7 Testbed Setup in TU Delft [15] 12

2-8 Testbed Schematic [1] 13

2-9 Node configuration of the network for the Testbed [15] 13

3-1 Centralised controller closed-loop. 17

3-2 Decentralised controller closed-loop for subsystem i. 18

3-3 Overlapping controller closed-loop for subsystem i. 22

4-1 Ramp signal that was used to identify the Pool 26

4-2 Comparison of test and validation results for Pool 3. 27

4-3 Comparison of test and validation results for Pool 2. 28

4-4 Undershot gates [1] 29

4-5 Flow rate data before and after filtering 30

4-6 Prediction of the model comparing Predicted Flow Rate vs Measured Flow Rate using training data. 32

4-7 Prediction of the model comparing Predicted Flow Rate vs Measured Flow Rate using testing data. 33

4-8 Simulink model of the Testbed 34

5-1 Decentralised Controller 36

5-2 Centralised Controller 36
5-3 Overlapping Controller 37
5-4 Combined disturbance rejection for the three different controllers 38
5-5 Error plots for the 6-Pool simulation with three types of controllers 42
5-6 Disturbance rejection for different alpha values on 3-Pool simulation 46
5-7 Disturbance rejection for different alpha values with noise on 3-Pool simulation . 47
5-8 Disturbance rejection for different alpha values on 6-Pool simulation 50
5-9 Disturbance rejection for different alpha values with noise on 6-Pool simulation . 52

6-1 Disturbance rejection for 3 controllers on the testbed 54

A-1 Centralised Controller 61
A-2 Decentralised Controller 62
A-3 Overlapping Controller 62
A-4 6-Pool Simulator in Simulink 63

List of Tables

4-1 NRMSE of the estimated model vs the measured data for the Pools 28
4-2 Estimated Values for the models of the Pools 28
4-3 NRMSE of the estimated model vs the measured data for the Gate 31
4-4 Estimated Gate Constants 32

5-1 Average maximum absolute error (± variance) per pool after 2000s for each controller. 38
5-2 3-pool simulation results: performance metrics after disturbance ($t \geq 2000$ s) shown as mean ± standard deviation. 39
5-3 Performance metrics after disturbance ($t \geq 2000$ s) for the 3-pool system, shown as mean ± standard deviation. 40
5-4 3-pool simulation results: performance metrics after disturbance with process noise ($t \geq 2000$ s) shown as mean ± standard deviation 41
5-5 3-pool simulation results: performance metrics after disturbance with process noise ($t \geq 2000$ s) shown as mean ± standard deviation 41
5-6 Average maximum absolute error (± variance) per pool after 3000s for each controller 43
5-7 6-pool simulation results: performance metrics after disturbance ($t \geq 3000$ s) shown as mean ± standard deviation. 43
5-8 6-pool simulation results: performance metrics after disturbance with measurement noise ($t \geq 3000$ s) shown as mean ± standard deviation 44
5-9 6-pool simulation results: performance metrics after disturbance with process noise ($t \geq 3000$ s) shown as mean ± standard deviation 45
5-10 Effect on ETC α on 3-pool simulation: performance metrics after disturbance with process noise ($t \geq 2000$ s) 48
5-11 Effect on ETC α with noise on 3-pool simulation: performance metrics after disturbance with process noise ($t \geq 2000$ s) 49
5-12 Effect on ETC α on 6-pool simulation: performance metrics after disturbance with process noise ($t \geq 3000$ s) 51
5-13 Effect on ETC α with noise on 6-pool simulation: performance metrics after disturbance with process noise ($t \geq 3000$ s) 52

6-1 Performance Metrics for each Controller and Pool 54

Preface and Acknowledgements

Over the past nine months, I have been working on implementing a control application to regulate the water testbed at TU Delft. The project turned out to be more challenging than expected, mainly due to the many unforeseen issues with the testbed itself. Despite the setbacks, the process was very rewarding and allowed me to apply what I've learned throughout my studies.

I would first like to thank my supervisor, Dr. ir. Manuel Mazo Jr., for his guidance and continuous support throughout the project. I'm also grateful for the opportunity to attend the weekly group meetings, where I received valuable feedback and insights from other researchers.

I would also like to thank Wim Wein, Will Geest and Ralph Willekes for their help and technical support whenever something went wrong with the setup.

Finally, I want to thank my friends and family for their support and encouragement throughout this entire process. Their help kept me motivated, especially during the more difficult moments of this project.

Delft, University of Technology
August 22, 2025

Stefan Ubaldo Rodrigues

Chapter 1

Introduction

"The wars of next century will be on water... unless we change the way we manage it" [26]. This statement by Ismail Serageldin highlights the growing concerns surrounding water scarcity and mismanagement. As populations continue to rise and climate change intensifies existing challenges, the demand for freshwater is higher than ever [22]. Without suitable measures in place to manage fresh water, water disputes could become a defining issue of the future.

In particular, efficient water distribution is crucial for agriculture irrigation. Nearly 70% of the worlds fresh water is used for agriculture, while 40% of it is wasted by inadequate irrigation systems, evaporation and poor water management [6]. In order to mitigate these issues, advanced control strategies for agriculture must be explored to optimises their water usage.

Water Irrigation Systems (WISs) are open-channel networks which are designed to supply water for agriculture. Effective management and control of these channels are essential to ensure reliable water supply for these farming operations. Consequently, WISs must be both energy and water efficient to minimise the waste of resources. While some research has been done into improving the energy efficiency of WISs [10, 27], the challenge of enhancing the water efficiency still remains an open area of investigation.

WISs are comprised of a series of interconnected pools, each of which is traditionally controlled by a decentralised controller [31]. These controllers operate solely on local state information, to regulate the water levels within the individual pool. While these controllers are able to regulate the water-levels in the pools, the lack of communication between the pools leads to sub-optimal performance. Furthermore, off-take disturbances, such as a farmer opening a field gate to divert water, are not communicated across the system, leading to further performance degradation. A centralised controller for the system [4] has shown to achieve optimal performance, however, due to the large size of some WISs, implementing such a controller is not feasible due to scalability.

This thesis explored a promising alternative that uses a distributed controller that communicates with neighbouring pools. This approach could mitigate the lack of coordination between the pools, and improve system performance. Since WISs span several kilometres, effective communication is essential. As a result, a Network Control System (NCS) will be required. Additionally, Event-Triggered Control (ETC) strategies were also investigated with the goal of reducing communication.

This thesis begins with Chapter 2, which explores the modelling of a WIS. It also introduces the experimental testbed and network infrastructure that was used throughout the study. Chapter 3 examines how the overlapping structure of the WIS can be used to develop a distributed controller. Additionally, it presents the application of decentralised Event Triggered Control (ETC) to reduce the communication overhead. Chapter 5 presents simulation results for both 3 pool and 6 pool configurations, comparing the performance of overlapping, decentralised and centralised control strategies. The effectiveness of ETC approach in reducing the communication costs are also evaluated in the simulation. Chapter 6 details the experimental validation of the controllers on the testbed, similarly comparing the different control strategies. Finally, Chapter 7 concludes this thesis with a summary of the findings, discussions and recommendations for future work.

1-1 Nomenclature

We denote by $\mathbb{R}$ the set of real numbers, and by $\mathbb{R}^+ := \{x \in \mathbb{R} : x \geq 0\}$ the set of non-negative real numbers. For a vector $x \in \mathbb{R}^n$, the 2-norm is denoted by $\|x\| := \sqrt{x^\top x}$. For a matrix $A \in \mathbb{R}^{n \times m}$, $A^\top$ represents its transpose, $\operatorname{rank}(A)$ its rank, and $\lambda(A)$ its eigenvalues. A matrix A is Hurwitz if all $\lambda(A)$ have strictly negative real parts. The identity matrix is denoted by I. The rest of the symbols are defined near their use in equations, drawn from control literature unless redefined for clarity when multiple sources conflict.

Chapter 2

Preliminaries

Water Irrigation Systems consist of a network of interconnected channels or pools. These pools are equipped with gates that regulate the flow of water between them. In addition to the main channels, the system includes secondary channels, which are smaller branches that diverge from the primary network to deliver water to agricultural fields, farms and other users, as can be seen in Figure 2-1

This thesis focuses exclusively on the control of water levels within the main channels, which are arranged in a series topology. The regulation of flow through secondary channels is considered outside the scope of this work and is assumed to be managed independently.

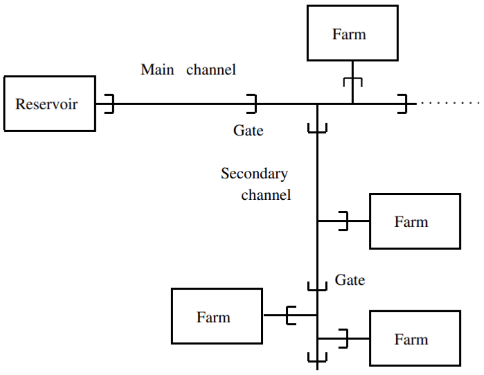


Figure 2-1: Water Irrigation System Topology

To evaluate the techniques intended for controlling the WISs, it is essential to develop a representative model. This model will be the base for the design, simulation and testing of the controllers. At TU Delft, there is a water testbed unit available for applying the developed controller. Therefore, the WIS model must accurately replicate the behaviour of the testbed unit, ensuring effective implementation and validation of the controllers.

2-1 Water Irrigation Systems

2-1-1 Simplified Model

WIS have traditionally been modelled using St. Venant equations [30, 5, 33], which are modelled using partial differential equations. Although these equations are good for prediction purposes, their use in real-time control is limited due to their high computational complexity. Hence, a simpler model will be proposed for prediction and control purposes.

According to [30], to design a controller for WIS, a model can be considered using a simplified mass balance [17]. This model is not able to capture the wave dynamics unlike the previous model, however it is sufficient in order to control the WIS, as long as the wave dynamics are not excited. This will further elaborated in Chapter 3.

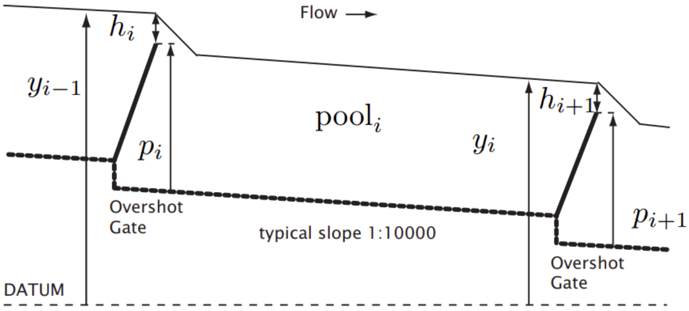


Figure 2-2: Cross Sectional profile of a WIS [18]

Since the primary objective of the controller is to regulate the water level, it will serve as the controlled variable, while the gate position will act as the input variable.

The dynamics of the pool can be expressed by the flow over the gate at each pool. The gate height is the only controllable parameter, hence, the relationship between the flow over rate and the gate height needs to be investigated. The relationship between them can be approximated by [2].

$$u_i \approx \gamma_i \cdot h_i^{3/2} \tag{2-1}$$

where γ_i is a flow constant for $pool_i$, and h_i is the head over gate i which is computed by subtracting the position of the top of the gate p_i from the water level in the pool y_i. A simplified model of the pool can now be derived [30] by using a simplified mass balance given by:

$$\alpha_i \dot{y}_i(t) = \gamma_i \cdot h_i^{3/2}(t) - \gamma_{i+1} \cdot h_{i+1}^{3/2}(t) - d_i(t) \tag{2-2}$$

where d_i models the off-take disturbance in $pool_i$, and α_i is a measure of the pool surface area. From the figure, it can be deduced that when the water flows out of $gate_i$, it does not instantly reach the next gate, however, it takes some time before it arrives at the point where the water level y_{i+1} is measured. From this, it can be concluded that some delay should be introduced in this system to account for this. By accounting for this delay, the equation now can be expressed by:

$$\alpha_i y_i(t+1) = \alpha_i y_i(t) + \gamma_i \cdot h_i^{3/2}(t-\tau_i) - \gamma_{i+1} \cdot h_{i+1}^{3/2}(t) - d_i(t) \tag{2-3}$$

where τ_i is used to represent the delay in the system. The equation can be simplified by directly substituting the variable for flow rate instead.

$$y_i(t+1) = y_i(t) + u_i(t-\tau_i) - v_i(t) - d_i(t) \tag{2-4}$$

where u_i is the flow over $gate_i$ and v_i is equivalent to u_{i+1}(outflow). The model can also be expressed as a frequency-domain model [18] which is given by:

$$y_i(s) = \frac{1}{\alpha_i s}(e^{-s\tau_i}u_i(s) - v_i(s) - d_i(s)) \tag{2-5}$$

The model derived, demonstrates that the outflow from one pool has a direct impact on the water level of the adjacent pool, as the outflow from one serves as the inflow for the next. This interconnected dynamic highlights how changes in one pool propagate through the system, influencing the behaviour of neighbouring pools. This interconnection is illustrated in Figure 2-3, which provides a visual representation of these interconnections. This model can be rewritten in the form of a transfer function, represented by:

$$P_i(s) = e^{-s\tau_i}\frac{1}{\alpha_i s}\pi_i \tag{2-6}$$

where, π_i is a polynomial that characterises the model dynamics. Depending on the complexity needed for the model, this can be a low order, which is ideal for control purposes, or a higher-order, which is ideal for simulation purposes.

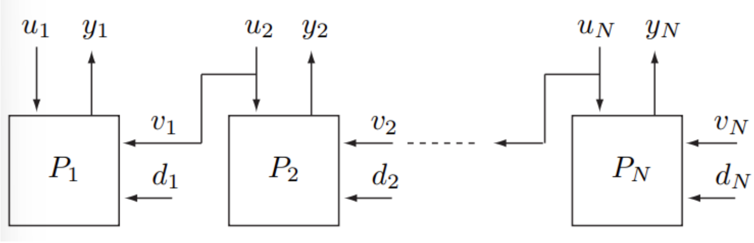


Figure 2-3: Diagram showing the interconnections between the pools [18]

2-1-2 Local Loop Shaping

As stated in Chapter 2, the frequency domain model of the water level in the pool is given by:

$$y_i(s) = \frac{1}{s \cdot \alpha_i}\left(e^{-s\tau_i}u_i(s) - v_i(s) - d_i(s)\right) \tag{2-7}$$

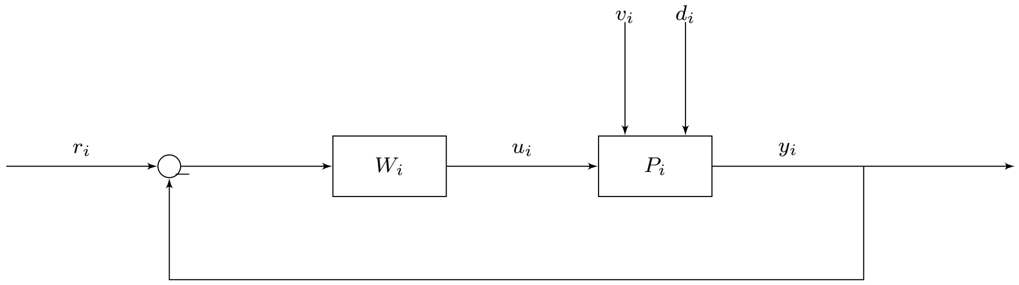


Figure 2-4: Simplified closed-loop control of pool i with loop-shaping weight W_i and plant P_i, including reference r_i and disturbances v_i and d_i.

Traditionally, WIS are controlled using decentralised controllers. Figure 2-4 illustrates how the closed-loop for a local pool with this ocntroller would look like. One way that is used to design controllers for these pools is via loop shaping. The local controllers associated with each P_i can be represented by the transfer function W_i. The goal when designing the loop-shaping weights is to shape the local loop-gain given by

$$L_i(s) = \frac{W_i(s)}{s\alpha_i} \tag{2-8}$$

so that the local closed-loop transfer functions

$$T_{r_i \to e_i}(s) = \frac{1}{1 + L_i(s)e^{-s\tau_i}} \tag{2-9}$$

$$T_{d_i,v_i \to e_i}(s) = \frac{1}{1 + L_i(s)e^{-s\tau_i}} \cdot \frac{1}{s\alpha_i} \tag{2-10}$$

$$T_{d_i \to u_i}(s) = \frac{L_i(s)}{1 + L_i(s)e^{-s\tau_i}} \tag{2-11}$$

are consistent with the local performance objectives. These objectives are water-level regulation, load disturbance rejection, and ensuring that the control action does not excite the dominant wave dynamics.

To achieve zero steady-state error for step disturbances, $W_i(s)$ must include at least one pole at the origin. In line with classical loop-shaping design, the loop-gain $|L_i(j\omega)|$ should be large at low frequencies, where reference signals and load disturbances are dominant, in order to achieve good tracking and rejection. However, the loop bandwidth must remain below both the effective delay limit $(1/\tau_i)$ and the dominant wave frequency φ_i; otherwise, instability or excitation of unmodelled wave dynamics may occur.

A suitable structure for $W_i(s)$ is given by an essentially PI-type compensator:

$$W_i(s) = \frac{\kappa_i(1 + s\phi_i)}{s(1 + s\rho_i)}, \tag{2-12}$$

where κ_i is used to set the loop-gain bandwidth, ϕ_i introduces phase lead around the crossover frequency to improve robustness, and ρ_i ensures additional roll-off at higher frequencies to attenuate wave excitation. This structure has been widely applied in both decentralised and centralised controller synthesis and provides a practical trade-off between performance and robustness for channel pool control.

2-2 Control Engineering

2-2-1 Overlapping Control

Overlapping control [14, 13] is a decentralised control scheme that is proposed for linear systems, composed of overlapping/interconnected subsystems.

Consider two linear time-invariant systems:

$$S : \dot{x} = Ax + Bu, \quad y = Cx \tag{2-13}$$

$$\tilde{S} : \dot{\tilde{x}} = \tilde{A}\tilde{x} + \tilde{B}\tilde{u}, \quad \tilde{y} = \tilde{C}\tilde{x} \tag{2-14}$$

where $x(t) \in \mathbb{R}^n$, $u(t) \in \mathbb{R}^m$, and $y(t) \in \mathbb{R}^p$ are the state, input, and output of system S, and $\tilde{x}(t) \in \mathbb{R}^{\tilde{n}}$, $u(t) \in \mathbb{R}^{\tilde{m}}$, and $\tilde{y}(t) \in \mathbb{R}^{\tilde{p}}$ are those of system $\tilde{S}$. The associated matrices have compatible dimensions: $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, $C \in \mathbb{R}^{p\times n}$ for S, and $\tilde{A} \in \mathbb{R}^{\tilde{n}\times\tilde{n}}$, $\tilde{B} \in \mathbb{R}^{\tilde{n}\times\tilde{m}}$, $\tilde{C} \in \mathbb{R}^{\tilde{p}\times\tilde{n}}$ for $\tilde{S}$. By construction, $\tilde{S}$ should satisfy:

$$\tilde{n} \geq n, \quad \tilde{m} \geq m, \quad \tilde{p} \geq p.$$

Now the pairs of linear transformations can be considered:

$$x = U\tilde{x}, \quad \tilde{x} = Vx \tag{2-15}$$

$$u = Q\tilde{u}, \quad \tilde{u} = Ru \tag{2-16}$$

$$y = S\tilde{y}, \quad \tilde{y} = Ty \tag{2-17}$$

where V, R and T are constant matrices with proper dimensions and full column ranks and U, Q and S are constant matrices with proper dimensions and full row ranks, where they satisfy

$$UV = I_n, \quad QR = I_m, \quad ST = I_p \tag{2-18}$$

According to Definition 2.7 from [14], *We say that the system S includes the system $\tilde{S}$, or equivalently, that S is included by $\tilde{S}$, if there exist a pair of matrices (U, V) and matrices R, S such that, for any initial state x_0 and any fixed input $u(t)$ of system S, the choice*

$$\tilde{x}_0 = Vx_0, \quad \tilde{u}(t) = Ru(t), \quad \textit{for all } t \geq 0$$

for the initial state and input of $\tilde{S}$ implies that

$$x(t; x_0, u) = U\tilde{x}(t; \tilde{x}_0, \tilde{u}), \quad y[x(t)] = S\tilde{y}[\tilde{x}(t)], \quad \textit{for all } t \geq 0.$$

This definition implies that $\tilde{S}$ contains all the necessary information about the system S. As a result, the stability of S can be concluded by assessing the stability of $\tilde{S}$. Often in practice, the system $\tilde{S}$ is constructed in order to exploit the interconnected properties of S. Lets consider that the system $\tilde{S}$ can be expressed as

$$\tilde{A} = VAU + M, \quad \tilde{B} = VBQ + N, \quad \tilde{C} = TCU + L \tag{2-19}$$

where M, N and L are complementary matrices. In practice, one has to construct an expanded system $\tilde{S}$ from a given S. To guarantee that $\tilde{S}$ is an expansion of S, Theorem 2.13 from [14] states *The system $\tilde{S}$ is an expansion of the system S if and only if*

$$\begin{aligned} UM^{i-1}V = 0, \quad UM^{i-1}NR = 0, \\ SLM^{i-1}V = 0, \quad SLM^{i-1}NR = 0, \end{aligned} \tag{2-20}$$

for all $i = 1, 2, \dot{;} \tilde{n}$

Now, lets consider the following linear feedback control law for system S

$$u = Kx + v \tag{2-21}$$

and for system $\tilde{S}\tilde{u} = \tilde{K}\tilde{x} + \tilde{v}$(2-22)

where K and $\tilde{K}$ are constant and of proper dimensions and $v \in R^m$, $\tilde{v} \in R^{\tilde{m}}$ are inputs for the resulting closed loop systems. If Theorem 2.23 from [14], the control law is also contractable, it implies that the closed loop system

$$S_c : \dot{x} = (A + BK)x + Bv, \quad y = Cx \tag{2-23}$$

is a contraction of the closed-loop system

$$\tilde{S}_c : \dot{\tilde{x}} = (\tilde{A} + \tilde{B}\tilde{K})\tilde{x} + \tilde{B}\tilde{v}, \quad \tilde{y} = \tilde{C}\tilde{x} \tag{2-24}$$

The idea behind overlapping control is to construct an extended system $\tilde{S}$ such that it contains all the essential dynamic behaviour of the original system S. By analysing $\tilde{S}$, particularly for stability, we can design a decentralized controller more easily. Once a suitable controller is developed for $\tilde{S}$, it can be contracted back to stabilize the original system S.

2-2-2 Event Triggered Control (ETC)

Nowadays, many control applications are now being applied across NCS. Since these are digital platforms, these control applications are discrete. Applying normal periodic control techniques to this systems is undesirable, due to the large waste in sensor and actuator communication [25]. Event-triggered Control is a control strategy in which sensor/actuator communication is only updated for stability or performance purposes.

Consider the following Linear Time-Invariant (LTI) system [11]:

$$\dot{x}^p = A^p x^p + B^p \hat{u} + B_w w, x^p \in \mathrm{R}^{n^p}, u \in \mathrm{R}^{n_u} \tag{2-25}$$

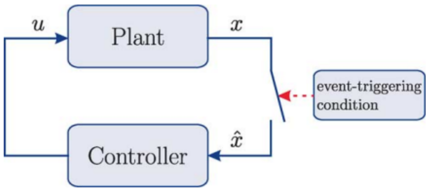


Figure 2-5: Schematic of Event-triggered Control [11]

where x^p and u^p are the state and input vectors of the plant. For traditional periodic control techniques, the linear feedback control law is :

$$\hat{u} = Kx_p, for \ \ t \in (t_k, t_{k+1}] \tag{2-26}$$

where t_k, $k \in N$ are the sampling times, which can be expressed as $t_k = kh, k \in N$, for a sampling interval $h > 0$. A Periodic Event-Triggered Control (PETC) [12] can now be proposed for this system, in which state values are transmitted at each sampling time $t_k = kh, k \in \mathrm{N}$, while the control action is updated when a event-triggering condition is satisfied. The control law for this system can now be given by:

$$\hat{u}(t) = K\hat{x}(t), for \ \ \mathrm{R}_+ \tag{2-27}$$

where $\hat{x}$ is a left-continuous signal, given for $t \in (t_k, t_{k+1}], k \in \mathrm{N}$, by

$$\hat{x}(t) \begin{cases} x(t_k), & when \ \ \mathcal{C}(x(t_k), \hat{x}(t_k) > 0) \\ \hat{x}(t_k), & when \ \ \mathcal{C}(x(t_k), \hat{x}(t_k) \leq 0) \end{cases} \tag{2-28}$$

From Figure 2-5, $\hat{x}(t)$ can be seen as the most recent update that has been transmitted from state x. The transmission of state x is governed by triggering condition seen in equation 2-28. In the case that the triggering condition is satisfied, the new state information is transmitted to the control, and $\hat{x}$ and $\hat{u}$ is updated. Conversely, if the triggering condition is not satisfied, the new state information is not transmitted, and the control action is not updated.

When discussing triggering conditions, a centralised approach is going to be considered when full state information $(x(t_k))$ is available. There are various kinds of triggering conditions. One such condition can be derived from State-Based error [20]. The condition can be expressed by:

$$||\hat{x}(t_k) - x(t_k)|| < \sigma||x(t_k)|| \tag{2-29}$$

where $\sigma > 0$. From this condition, it can be seen if the current state $x(t_k)$ deviates far from the last updated $\hat{x}(t_k)$, then the condition is satisfied, and the new state is transmitted to the controller and the control input is adjusted accordingly. There are also other triggering conditions that can be derived, i.e. input error based [7] and Lyapunov function based [23].

Decentralised Approach

The previous conditions were derived based on a centralised approach, where full state information is available. This approach would not be suitable for the test bed unit for two reasons. The first being that full state information is not available. As a result, output based triggering will need to be investigated. Furthermore, a centralised approach would mean that a centralised coordinator would need access to the all the information across all the subsystems. A decentralised approach [11] would only require local information for the condition.

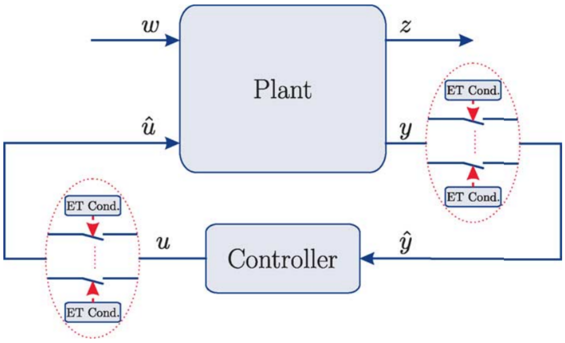


Figure 2-6: Schematic of Decentralised Event-trigger control [11]

The following LTI system can be considered:

$$\begin{cases} \dot{x}^p = A^p x^p + B^p \hat{u} + B_w w \\ y = C^p x^p \end{cases} \tag{2-30}$$

where $x^p \in \mathrm{R}^{n_p}$ and $\hat{u} \in \mathrm{R}^{\mathrm{n_u}}$ is the state and input of the plant. The subscript P is used to differentiate the properties associated with the plant. The properties associated with the controller are denoted with c. The plant is controlled by a discrete time controller whose LTI is given by:

$$\begin{cases} x^c_{k+1} = A^c x^c_k + B^c \hat{y}_k \\ u_k = C^c x^c_k + D^c \hat{y}_{k-1} \end{cases} \tag{2-31}$$

where $x^c \in \mathrm{R}^{n_c}$ is the state of the controller, $\hat{y}_k \in \mathrm{R}^{n-y}$ is the input to the controller, and $u_k \in \mathrm{R}^{n_u}$ is the control input in the system. Similarly, the sampling times is $t_k = kh$, $k\mathrm{N}$, where $h > 0$ is the sampling interval. At the sampling time, the triggering conditions will determine whether $\hat{u}$ and $\hat{y}$ values will be updated. The triggering condition is based on the values of $y(t_k)$ and $u(t_k)$.

In order to define the decentralised event triggering conditions, first v needs to be defined as:

$$v = \begin{bmatrix} y \\ u \end{bmatrix} \in \mathrm{R}^{n_v}, \hat{v} = \begin{bmatrix} \hat{y} \\ \hat{u} \end{bmatrix} \in \mathrm{R}^{n_v} \tag{2-32}$$

where $n_v := n_y + n_u$. the entries in v and $\hat{v}$ are grouped into N nodes, where the corresponding node $j \in \{1, .., N\}$ are given by v^j and $\hat{v}^j$. The event triggering condition can now be described by:

$$\hat{v}^j(t) \begin{cases} v^j(t_k), & if \;\; ||v^j(t_k) - \hat{v}^j(t_k)|| > \sigma_j ||v^j(t_k)|| \\ v^j(t_k), & if \;\; ||v^j(t_k) - \hat{v}^j(t_k)|| \leq \sigma_j ||v^j(t_k)|| \end{cases} \tag{2-33}$$

for $t \in (t_k, t_{k+1}]$, $k \in \mathrm{N}$, where $\sigma_j \geq 0$ are assigned constants. At each sampling time t_k, $K \in \mathrm{N}$, each node is able to check this difference $v^j(t_k) - v^j\hat{(t_k)}$. If this difference is large, the node j will transmit the signal of $v^j(t_k)$, and subsequently also update $\hat{v}^j$. For this case, it can be considered that each node has its own local event triggering condition, given by:

$$||v^j(t_k) - \hat{v}^j(t_k)|| > \sigma_j ||v^j(t_k)|| \tag{2-34}$$

2-2-3 LQR

2-2-4 Observer

If full state information is not available, an observer can be used to reconstruct the states. This is important in the case LQR controllers are used, as they require full state information. In order to reconstruct the state vector, the state space of he system needs to be observable. For a discrete-time system with dynamics

$$x(t+1) = Ax(t) + Bu(t), \quad y(t) = Cx(t),$$

the system is observable if the observability matrix

$$\mathcal{O} = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{n-1} \end{bmatrix} \tag{2-35}$$

One such observer that can be used is a Luenberger observer. The observer estimates the state $\hat{x}(t+1)$ using the following update equation:

$$\hat{x}(t+1) = A\hat{x}(t) + Bu(t) + L(y(t) - C\hat{x}(t)) \tag{2-36}$$

where $y(t)$ is the output measurement, $u(t)$ is the input and L is the observer gain. This observer gain is chosen so that the poles of $(A - LC)$ are within the unit circle. These poles were computed using the discrete-time Riccati equation.

2-2-5 Discretising

2-3 Testbed Unit

WISs can span for several kilometres, making direct testing impractical. To address this, a water testbed unit is available at TU Delft. This unit is a scaled down representation of a WIS, designed to replicate the dynamics of the real world system. This section will go into detail how the testbed unit operates.

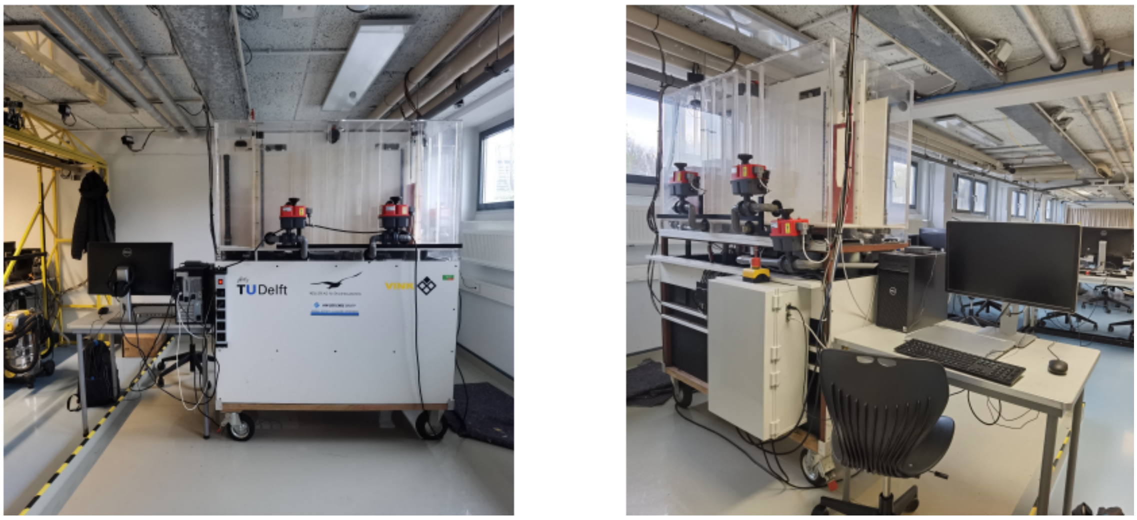


Figure 2-7: Testbed Setup in TU Delft [15]

2-3-1 Layout

A schematic of the testbed unit can be seen in Figure 2-8. This setup consists of four main pools which are connected in series with each other. The flow between the gates is governed by the gates (orange rectangles). The first three gates are undershot gates, while the last gate is an overshot gate, which water then flows back into the water tank. The water levels can be measured with the help of the pressure sensors (yellow circles). There are seven of these sensors, indicating that the water levels at these points can be measured. Disturbances are added into the system with the help of valves (black circles), which remove or add water into the pools.

Although there are four pools present in this system, the first pool will act as a water storage, hence, only the water level in the subsequent three pools will be actively controlled. Furthermore, each of the three controlled pools contain barriers with holes strategically placed inside them. These barriers are designed to slow down the flow of water within the system, thereby simplifying the control process. Pools 1 and 3 each have six barriers, while Pool 2 has two, which can be seen in Figure 2-8.

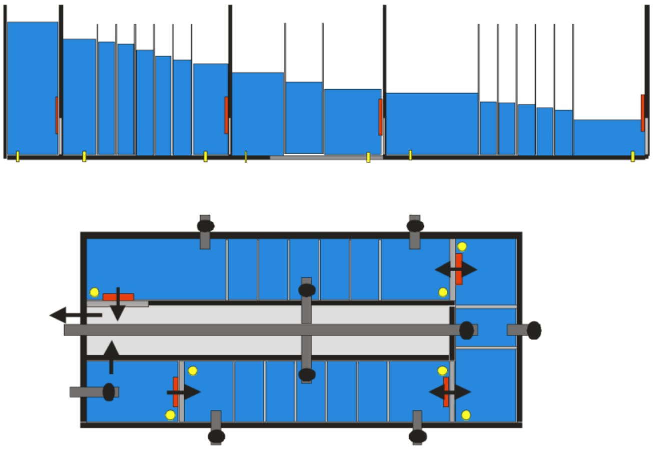
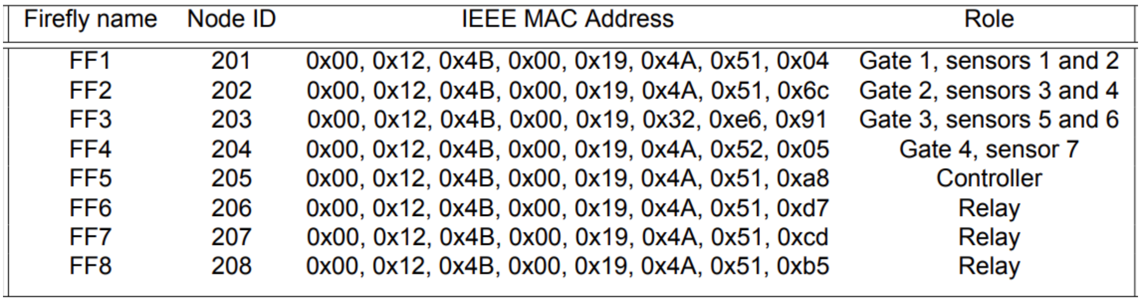


Figure 2-8: Testbed Schematic [1]

2-3-2 Network Infrastructure

The wireless communication network was developed by [1] and [16]. The network was on the WCB [28] communication protocol and Crystal, [29] with Glossy [9] as the underlying communication and time-synchronisation primitive. A more detailed description of these protocols can be found the respective papers. To summarise, the network consists of eight nodes, each with a specific function. The first four nodes, shown in Figure 2-9, are designated as "Sensor/Actuator" nodes. These nodes are responsible for reading data from their respective sensors and performing both global and local control to actuate the gates. The FF5 node serves as the Controller node, managing the network protocol and overseeing the global control of the testbed unit. Finally, the remaining nodes in the network are Relay nodes, which are tasked with maintaining connectivity to the WCB network.



Firefly name	Node ID	IEEE MAC Address	Role
FF1	201	0x00, 0x12, 0x4B, 0x00, 0x19, 0x4A, 0x51, 0x04	Gate 1, sensors 1 and 2
FF2	202	0x00, 0x12, 0x4B, 0x00, 0x19, 0x4A, 0x51, 0x6c	Gate 2, sensors 3 and 4
FF3	203	0x00, 0x12, 0x4B, 0x00, 0x19, 0x32, 0xe6, 0x91	Gate 3, sensors 5 and 6
FF4	204	0x00, 0x12, 0x4B, 0x00, 0x19, 0x4A, 0x52, 0x05	Gate 4, sensor 7
FF5	205	0x00, 0x12, 0x4B, 0x00, 0x19, 0x4A, 0x51, 0xa8	Controller
FF6	206	0x00, 0x12, 0x4B, 0x00, 0x19, 0x4A, 0x51, 0xd7	Relay
FF7	207	0x00, 0x12, 0x4B, 0x00, 0x19, 0x4A, 0x51, 0xcd	Relay
FF8	208	0x00, 0x12, 0x4B, 0x00, 0x19, 0x4A, 0x51, 0xb5	Relay

Figure 2-9: Node configuration of the network for the Testbed [15]

2-4 Conclusion

This Chapter introduced the simplified model for modelling the WIS, motivated by the need for a computationally efficient model that can also be used for real time control. A general modelling approach was proposed, whose complexity can be tuned depending on the requirements for the model. The testbed unit at TU Delft will serve as a platform for evaluating the developed controllers. This system is a scaled down representation of a WIS which is able to replicate the real world system

Chapter 3

Control Design

A Water Irrigation System (WIS) consists of interconnected pools, where the outflow of one pool serves as the inflow to another. Consequently, the dynamics of each pool have a direct influence on the others. One approach to controlling a WIS is through the design of a centralised controller [4, 19], which utilises complete state information from all pools to manage each gate. In theory, this method is optimal as it considers the dynamics of the entire system. However, it suffers from significant drawbacks, primarily scalability. The computational cost of this approach increases exponentially with the number of pools, rendering it impractical for large-scale, real-world systems.

In contrast, a decentralised control approach [31] assigns a dedicated controller to each pool, which operates using only its own state information. While this method is computationally less demanding, it introduces a critical challenge: each controller optimises its own objectives independently, inadvertently affecting other pools due to the interconnected nature of the system. This lack of coordination can lead to suboptimal performance across the entire WIS.

One possible solution to solve both of the issues , is to exploit the interconnected properties of the system and employ a decentralized controller that also takes in information from neighbouring subsystems.

3-1 Controller Formulation

The primary focus of this thesis is the development of a distributed controller for WISs using overlapping control. This section will provide a detailed explanation of how the overlapping controller is formulated. Additionally, it will describe the design of both centralised and decentralised controller for WISs. These controllers will serve as benchmarks for comparing the performance of the overlapping controller.

3-1-1 State Space Formulation

The first step in formulating the controllers involves obtaining a state-space representation of the system. Section 2-1-2 detailed how the loop-shaping weight W_i is designed to ensure that local performance objectives are satisfied. Regardless of whether the controller is centralised, decentralised or overlapping, each subsystem P_i is paired with its respective W_i to guarantee that these local objectives are met.

Alternatively, we can consider a generalised pool G_i, which represents the combination of P_i and W_i. An additional controller can then be designed for G_i, and its structure will depend on whether the control approach is centralised, decentralised or overlapping. To design this additional controller, it is necessary first to obtain a state-space representation of G_i. This state-space is constructed using Padé approximations to account for the delays in the plant. This approach, adapted from [18], has been modified so that it can use overlapping control techniques. The modified state space is given by:

$$\begin{bmatrix} \dot{y}_i \\ \dot{\Delta}_i \\ \dot{u}_i \\ \dot{\Omega}_i \\ \dot{y}_{i+1} \\ \dot{\Delta}_{i+1} \\ \dot{u}_{i+1} \\ \dot{\Omega}_{i+1} \end{bmatrix} = \begin{bmatrix} 0 & \frac{1}{\alpha_i} & \frac{-1}{\alpha_i} & 0 & 0 & 0 & \frac{-1}{\alpha_i} & 0 \\ 0 & \frac{-2}{\tau_i} & \frac{4}{\tau_i} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{-1}{\rho_i} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{\alpha_{i+1}} & \frac{-1}{\alpha_{i+1}} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{-2}{\tau_{i+1}} & \frac{4}{\tau_{i+1}} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{-1}{\rho_{i+1}} \end{bmatrix} \begin{bmatrix} y_i \\ \Delta_i \\ u_i \\ \Omega_i \\ y_{i+1} \\ \Delta_{i+1} \\ u_{i+1} \\ \Omega_{i+1} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \frac{\kappa_i \phi_i}{\rho_i} & 0 \\ \frac{\kappa_i(\rho_i - \phi_i)}{\rho_i^2} & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & \frac{\kappa_{i+1}\phi_{i+1}}{\rho_{i+1}} \\ 0 & \frac{\kappa_{i+1}(\rho_{i+1} - \phi_{i+1})}{\rho_{i+1}^2} \end{bmatrix} \begin{bmatrix} u_{k,i} \\ u_{k,i+1} \end{bmatrix} \tag{3-1}$$

where y_i is the water level in the $pool_i$, Ω_i corresponds to the loop shaping pole, Δ_i refers to the pole in the Padé approximation [3] and u_i is the flow over $gate_i$. From equation 3-1, it can be seen that state u_{i+1} affects y_i. From this, it can be seen why purely decentralised control will not give optimal results, as the water level prediction can not be accurate without using information from the neighbouring pools. This gives some justification into why overlapping control should be a promising alternative.

3-1-2 Centralised Controller

This section details the formulation of the centralised controller. The first step involves combining the system variables into block-diagonal matrices. Specifically,

$$G = \text{diag}(G_1, G_2, \ldots, G_i), \quad P = \text{diag}(P_1, P_2, \ldots, P_i), \quad W = \text{diag}(W_1, W_2, \ldots, W_i),$$

$$u = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_i \end{bmatrix}, \quad y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_i \end{bmatrix}, \quad R = \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_i \end{bmatrix}.$$

This representation combines all subsystems into a single large state-space model, which is required for centralised controller design since full system information is utilised.

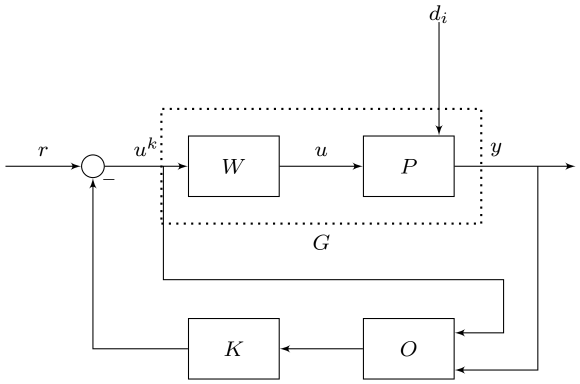


Figure 3-1: Centralised controller closed-loop.

The state-space representation in equation 3-1 will now be considered for a 3 pool system. To make it easy to follow, the state space will be simplified to take the following form:

$$A = \begin{bmatrix} a_{11} & a_{12} & 0 \\ 0 & a_{22} & a_{23} \\ 0 & 0 & a_{33} \end{bmatrix}, \quad B = \begin{bmatrix} b_{11} & 0 & 0 \\ 0 & b_{22} & 0 \\ 0 & 0 & b_{33} \end{bmatrix}$$
$$C = \begin{bmatrix} c_{11} & 0 & 0 \\ 0 & c_{22} & 0 \\ 0 & 0 & c_{33} \end{bmatrix}, \quad D = 0 \tag{3-2}$$

where,

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}, u = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} \tag{3-3}$$

Using the derived system, a centralised controller can be designed employing an LQR controller (K) in series with an observer(O). The LQR controller utilises the full state information to generate the control action u_k. The observer obtains all the pool water levels y and the control inputs u_k to estimate the complete state vector. It is also important to note, that the full system is observable, ensuring that the observer can reconstruct all the states with the given measurements y. Figure **??** illustrates the centralised control loop.

It is important to note that the system matrix A includes the cross-coupling terms (a_{12}, a_{23}) which represent the interconnection between the connected pools. These terms capture how the dynamics of one pool affect the others, ensuring that the controller accounts for interconnections and can coordinate control actions across the entire system.

3-1-3 Decentralised Controller

This section shows how the decentralised controller will be formulated. In this case, each subsystem is controlled independently, meaning that each controller only uses local measure-

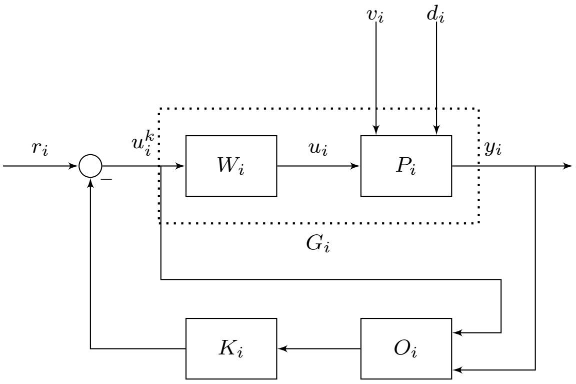


Figure 3-2: Decentralised controller closed-loop for subsystem i.

ments. To begin with, the system variables can be split up again, and be grouped according to their respective subsystem:

$$G_i, \quad P_i, \quad W_i, \quad u_i, \quad y_i, \quad r_i \quad fori = 1, 2, ...N$$

where i represents each subsystem. Considering the same simplified state-space representation in equation 3-2, new state spaces can be formulated for each of the three pools:

$$A_1 = a_{11}, \qquad B_1 = b_{11}, \qquad C_1 = c_{11}, \qquad D_1 = 0, \qquad x_1\ ,u_1 \tag{3-4a}$$

$$A_2 = a_{22}, \qquad B_2 = b_{22}, \qquad C_2 = c_{22}, \qquad D_2 = 0, \qquad x_2\ ,u_2 \tag{3-4b}$$

$$A_3 = a_{33}, \qquad B_3 = b_{33}, \qquad C_3 = c_{33}, \qquad D_3 = 0, \qquad x_3\ ,u_3 \tag{3-4c}$$

Using these subsystems, decentralised controllers can be designed similarly to the centralised case, using a LQR controller(K_i) in series with an observer(O_i). The main difference is that a separate controller and observer pair is implemented for each subsystem. each observer uses water level measurements from its receptive subsystem to reconstruct the local state vector. The LQR controller than generates the control action u_i^k to control the respective pool.

Unlike the centralised case, this method does not account for the cross coupling terms between subsystems, such as a_{12} and a_{23}. As a result, these interaction are not accounted for, which may effect the overall system performance.

3-1-4 Overlapping Controller

The section will do into detail about how overlapping properties of the WIS is going to be used to design a distributed controller. The goal when designing The controller (K_i), will be to design a controller that uses neighbouring state information alongside its own local state information to control P_i. To do this, an extended state space of the simplified state space from equation 3-2 should be found. By introducing repetition states, an expanded state space is found

$$A_e = \begin{bmatrix} a_{11} & a_{12} & 0 & 0 & 0 \\ 0 & a_{22} & 0 & 0 & 0 \\ 0 & 0 & a_{22} & a_{23} & 0 \\ 0 & 0 & 0 & a_{33} & 0 \\ 0 & 0 & 0 & 0 & a_{33} \end{bmatrix}, \quad B_e = \begin{bmatrix} b_{11} & 0 & 0 & 0 & 0 \\ 0 & b_{22} & 0 & 0 & 0 \\ 0 & 0 & b_{22} & 0 & 0 \\ 0 & 0 & 0 & b_{33} & 0 \\ 0 & 0 & 0 & 0 & b_{33} \end{bmatrix}$$
$$C_e = \begin{bmatrix} c_{11} & 0 & 0 & 0 & 0 \\ 0 & c_{22} & 0 & 0 & 0 \\ 0 & 0 & c_{22} & 0 & 0 \\ 0 & 0 & 0 & c_{33} & 0 \\ 0 & 0 & 0 & 0 & c_{33} \end{bmatrix}, \quad D_e = 0 \tag{3-5}$$

where

$$x_e = \begin{bmatrix} x_1 \\ x_2 \\ x_2 \\ x_3 \\ x_3 \end{bmatrix}, \quad u_e = \begin{bmatrix} u_1 \\ u_2 \\ u_2 \\ u_3 \\ u_3 \end{bmatrix} \tag{3-6}$$

By introducing the new repetition states, a new extended state space is formulated. In order to prove that it is an expansion of the original system, it must be shown that the condition from equation 2-20 are met.

This can be done first by defining all the necessary matrices.

$$V = \begin{bmatrix} I_{n_1} & 0 & 0 \\ 0 & I_{n_2} & 0 \\ 0 & I_{n_2} & 0 \\ 0 & 0 & I_{n_3} \\ 0 & 0 & I_{n_3} \end{bmatrix} \quad \in \mathbb{R}^{(n_1+2n_2+2n_3)\times(n_1+n_2+n_3)}$$

$$R = \begin{bmatrix} I_{m_1} & 0 & 0 \\ 0 & I_{m_2} & 0 \\ 0 & I_{m_2} & 0 \\ 0 & 0 & I_{m_3} \\ 0 & 0 & I_{m_3} \end{bmatrix} \quad \in \mathbb{R}^{(m_1+2m_2+2m_3)\times(m_1+m_2+m_3)}$$

$$T = \begin{bmatrix} I_{p_1} & 0 & 0 \\ 0 & I_{p_2} & 0 \\ 0 & I_{p_2} & 0 \\ 0 & 0 & I_{p_3} \\ 0 & 0 & I_{p_3} \end{bmatrix} \quad \in \mathbb{R}^{(p_1+2p_2+2p_3)\times(p_1+p_2+p_3)}$$

$$U = \begin{bmatrix} I_{n_1} & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2}I_{n_2} & \frac{1}{2}I_{n_2} & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2}I_{n_3} & \frac{1}{2}I_{n_3} \end{bmatrix} \quad \in \mathbb{R}^{(n_1+n_2+n_3)\times(n_1+2n_2+2n_3)}$$

$$
Q = \begin{bmatrix} I_{m_1} & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2}I_{m_2} & \frac{1}{2}I_{m_2} & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2}I_{m_3} & \frac{1}{2}I_{m_3} \end{bmatrix} \in \mathbb{R}^{(m_1+m_2+m_3)\times(m_1+2m_2+2m_3)}
$$

$$
S = \begin{bmatrix} I_{p_1} & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2}I_{p_2} & \frac{1}{2}I_{p_2} & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2}I_{p_3} & \frac{1}{2}I_{p_3} \end{bmatrix} \in \mathbb{R}^{(p_1+p_2+p_3)\times(p_1+2p_2+2p_3)}
$$

$$
M = \begin{bmatrix} 0 & A_{12} & -A_{12} & 0 & 0 \\ 0 & A_{22} & -A_{22} & \frac{1}{2}A_{23} & -\frac{1}{2}A_{23} \\ 0 & -A_{22} & A_{22} & -\frac{1}{2}A_{23} & \frac{1}{2}A_{23} \\ 0 & 0 & 0 & \frac{1}{2}A_{33} & -\frac{1}{2}A_{33} \\ 0 & 0 & 0 & -\frac{1}{2}A_{33} & \frac{1}{2}A_{33} \end{bmatrix} \in \mathbb{R}^{(n_1+2n_2+2n_3)\times(n_1+2n_2+2n_3)}
$$

$$
N = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2}B_{22} & -\frac{1}{2}B_{22} & 0 & 0 \\ 0 & -\frac{1}{2}B_{22} & \frac{1}{}B_{22} & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2}B_{33} & -\frac{1}{2}B_{33} \\ 0 & 0 & 0 & -\frac{1}{2}B_{33} & \frac{1}{2}B_{33} \end{bmatrix} \in \mathbb{R}^{(n_1+2n_2+2n_3)\times(n_1+2n_2+2n_3)}
$$

$$
L = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2}C_{22} & -\frac{1}{2}C_{22} & 0 & 0 \\ 0 & -\frac{1}{2}C_{22} & \frac{1}{2}C_{22} & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2}C_{33} & -\frac{1}{2}C_{33} \\ 0 & 0 & 0 & -\frac{1}{2}C_{3} & \frac{1}{2}C_{33} \end{bmatrix} \in \mathbb{R}^{(n_1+2n_2+2n_3)\times(n_1+2n_2+2n_3)}
$$

By showing that all the 4 conditions are met, it was proved that equation 3-5 is an expansion of equation 3-2. Looking at the expansion, it can be seen that due to the addition of the repetition states, the system can now decoupled into 3 different subsystems, with the following states and inputs:

$$
x_{e1} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \quad x_{e2} = \begin{bmatrix} x_2 \\ x_3 \end{bmatrix}, \quad x_{e3} = \begin{bmatrix} x_3 \end{bmatrix} \tag{3-7}
$$

$$
u_{e1} = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}, \quad u_{e2} = \begin{bmatrix} u_2 \\ u_3 \end{bmatrix}, \quad u_{e3} = \begin{bmatrix} u_3 \end{bmatrix} \tag{3-8}
$$

where the subsystems are defined by :

$$
A_{e1} = \begin{bmatrix} a_{11} & a_{12} \\ 0 & a_{22} \end{bmatrix}, \quad B_{e1} = \begin{bmatrix} b_{11} & 0 \\ 0 & b_{22} \end{bmatrix}, \quad C_{e1} = \begin{bmatrix} c_{11} & 0 \\ 0 & c_{22} \end{bmatrix}, \tag{3-9}
$$

$$
A_{e2} = \begin{bmatrix} a_{22} & a_{23} \\ 0 & a_{33} \end{bmatrix}, \quad B_{e2} = \begin{bmatrix} b_{22} & 0 \\ 0 & b_{33} \end{bmatrix}, \quad C_{e2} = \begin{bmatrix} c_{22} & 0 \\ 0 & c_{33} \end{bmatrix}, \tag{3-10}
$$

$$A_{e3} = \begin{bmatrix} a_{33} \end{bmatrix}, \quad B_{e3} = \begin{bmatrix} b_{33} \end{bmatrix}, \quad C_{e3} = \begin{bmatrix} c_{33} \end{bmatrix}. \tag{3-11}$$

Using decentralized control schemes (in this case LQR), a control law can be devised for each of the subsystems. The corresponding feedback gains are given by

$$K_{e1} = \begin{bmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{bmatrix}, \quad K_{e2} = \begin{bmatrix} k_{33} & k_{34} \\ k_{43} & k_{43} \end{bmatrix}, \quad K_{e3} = \begin{bmatrix} k_{55} \end{bmatrix}. \tag{3-12}$$

This control law can also be represented in terms of a block-diagonal feedback gain matrix for the expanded system, given by

$$K_e = \begin{bmatrix} k_{11} & k_{12} & 0 & 0 & 0 \\ k_{21} & k_{22} & 0 & 0 & 0 \\ 0 & 0 & k_{33} & k_{34} & 0 \\ 0 & 0 & k_{43} & k_{44} & 0 \\ 0 & 0 & 0 & 0 & k_{55} \end{bmatrix}. \tag{3-13}$$

This control law can now be contracted back to fit in the original system. The new contracted control law is given by

$$K = \begin{bmatrix} k_{11} & k_{12} & 0 \\ k_{21} & \frac{1}{2}(k_{22} + k_{33}) & k_{34} \\ 0 & k_{43} & \frac{1}{2}(k_{44} + k_{55}) \end{bmatrix}. \tag{3-14}$$

where the control laws for associated with each pool are defined by:

$$K_1 = \begin{bmatrix} k_{11} & k_{12} \end{bmatrix}, \tag{3-15}$$

$$K_2 = \begin{bmatrix} k_{21} & \frac{1}{2}(k_{22} + k_{33}) & k_{34} \end{bmatrix}, \tag{3-16}$$

$$K_3 = \begin{bmatrix} k_{43} & \frac{1}{2}(k_{44} + k_{55}) \end{bmatrix}. \tag{3-17}$$

.

From the structure of the contracted controller, it can be seen that the control input for Pool 2 depends not only on its local states, but on the states from the neighbouring pools. Therefore, in order to regulate the water levels in Pool 2, states information from the adjacent pools needs to be communicated. This is not the case for Pool 1 and 3, as they are the first and last pools in the 3-Pool system. They only need to communicate with Pool 2, as they only have one neighbouring pool. The control loop for the overlapping closed loop can be seen in Figure 3-3.

The main difference with this closed loop, as compared to the decentralised case, lies in the observer and the LQR controller, which are now computed by incorporating the neighbouring states as well. The observer O_i receives the measurements y_i, y_{i-1}, and y_{i+1}. Using these, it reconstructs the augmented state vector that captures the combined dynamics of P_i together with its neighbouring pools. An important advantage of this formulation is that the observer

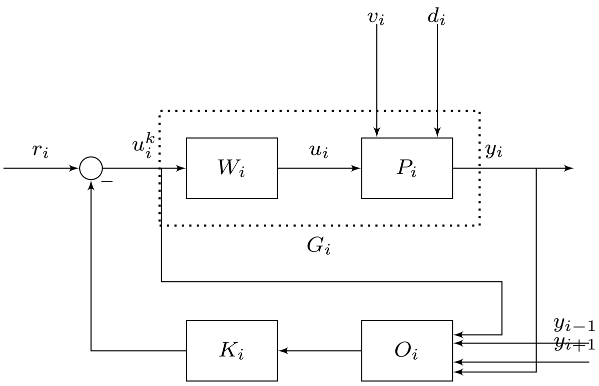


Figure 3-3: Overlapping controller closed-loop for subsystem i.

can estimate the disturbance v_i as part of the neighbouring pool dynamics, rather than treating it as an external unknown.

The reconstructed state vector is then passed to the local controller K_i, which implements the overlapping LQR control law derived earlier. In this way, the controller for pool P_i is able to account for both its own dynamics and the influence of adjacent pools. This allows K_i to exploit neighbouring information when computing the control input, leading to more coordinated and effective control actions across the system.

3-1-5 ETC overlapping controller

Another goal was to implement ETC control within the overlapping control framework. To achieve this, a decentralized output-based event-triggered control (ETC) scheme, as introduced in Section 2-2-2, is applied. A centralized scheme would defeat the main advantage of overlapping control, which is scalability and modularity of the controller. Moreover, since full state information is generally unavailable, an output-based ETC triggering condition is employed.

The triggering condition for each subsystem i can be described by:

$$\hat{v}^i(t) = \begin{cases} v^i(t_k), & \text{if } ||v^i(t_k) - \hat{v}^i(t_k)|| > \sigma_i ||v^i(t_k)||, \\ v^i(t_k), & \text{if } ||v^i(t_k) - \hat{v}^i(t_k)|| \leq \sigma_i ||v^i(t_k)||, \end{cases} \tag{3-18}$$

for $t \in (t_k, t_{k+1}]$, $k \in \mathbb{N}$, where $\sigma_i \geq 0$ are predetermined constants for each subsystem.

Intuitively, the first branch of (3-18) checks whether the output vector y_i of subsystem i has changed significantly since the last sampling instant t_k. If the difference between the current measurement $v^i(t_k)$ and the previous estimate $\hat{v}^i(t_k)$ exceeds the threshold defined by σ_i, the triggering condition is satisfied. In this case, the new measurement is accepted, $\hat{v}^i(t)$ is updated, and a new control input u_i is computed according to the overlapping control law.

If, on the other hand, the triggering condition is not met, the previous estimate $\hat{v}^i(t_k)$ is retained and the control input u_i remains unchanged. This ensures that control updates are only computed and communicated when necessary, reducing computational load and communication burden while maintaining stability and performance.

Once the control input u_i is updated, it is applied to the subsystem, and the ETC mechanism continues to monitor the output y_i. The process repeats for all subsystems in parallel, allowing the overlapping controller to operate efficiently in a distributed, scalable manner.

Overall, this approach combines the benefits of overlapping control with event-triggered updates, ensuring that control actions are only applied when significant deviations occur in the subsystem outputs, thereby reducing unnecessary control updates while maintaining closed-loop performance.

3-2 Conclusion

This chapter explored the challenges and methodologies associated with designing a controller for the WIS. Centralised control, while theoretically optimal, suffers with scalability issues, which it is likely to encounter when modelling large scale WIS. On the other hand, purely decentralised control, will lead to suboptimal solutions. To address these issues, overlapping control was introduced as a viable approach, which is able to use the interconnected properties of the WIS, to decouple the system.

The overlapping control framework involves designing decentralised controllers that account for interactions with neighbouring subsystems. In this context, LQR formulations were employed to design the controllers.

Furthermore, the principle of ETC was explored as an alternative to traditional periodic control strategies. By updating the control action only when necessary, ETC effectively reduced communication overhead while maintaining performance and stability.

From the research that was conducted, it was concluded that for testbed implementation, output-based triggering conditions will be adopted, ensuring that the controllers only update the control action solely based on available measurements. Additionally, a decentralised approach will be implemented to improve scalability in controller design.

Chapter 4

The Water Testbed Unit

This chapter outlines the approach used to identify the dynamics of the tested unit. To design effective controllers for the testbed unit, it is essential to first develop an accurate dynamical model of the system.

The modelling of the unit can be split into two distinct parts, one for the individual pools and the other for the model describing the dynamics of gate.

4-1 Model

As discussed in Chapter 2, varying the order of the π approximation results in models of different complexity. For the purposes of this project, both, a first and a third order model, will be derived. The first order model is given by the following equation:

$$P_i(s) = e^{-s\tau_i} \frac{1}{\alpha_i s} \tag{4-1}$$

where τ_i refers to the time delay in the system and α_i is the measure of the pool surface area. This model is good at identifying the general trend in the water levels. It will therefore be used for control design due to its simplicity and reduced computational demand.

Conversely, the third-order model is defined as

$$P_i(s) = e^{-s\tau_i} \frac{1}{\alpha_i s} \frac{\omega_{n,i}^2}{s^2 + 2\zeta_i \omega_{n,i} s + \omega_{n,i}^2} \tag{4-2}$$

where it is comprised of the same first order model, with another second order model combined with it. Here $\omega_{n,i}$ denotes the natural frequency and ζ_i represents the damping ratio of the system. This model offers a more detailed representation of the testbed dynamics, particularly by capturing the wave motion, which is not addressed in simpler first order approximations. As such, this model is going to be used for simulation purposes. To be able to find these parameters, these values need to be estimated using system identification techniques.

4-1-1 Estimation of the Coefficients

Procedure

1. **Gate Setup for Identification**

   - To identify $pool_i$, each pool $\geq i$, is emptied out, and their respective gates closed.
   - The pools before $\leq i$, are filled up to a level just to the limit of the $gate_i$

2. **Data Collection**

   - Apply an input signal to the respective gate
   - Continue recording the data until the water levels in all the pools reach steady-state, i.e. all the water levels stabilise and are the same in all the pools

System Identification

Once the data is collected, the parameters need to be identified. α_i can be directly derived from the physical properties of the test bed. Specifically, α_i represents the surface area of $pool_i$, hence it can be readily measured. τ_i is the time delay, this parameter can computed by looking at the difference in time between when the input signal is applied and the time when the pressure sensor on the other side of the pool records a change in pressure.

Grey-box estimation will be used for this system, which is appropriate given that the equations and some parameters are already known. By combining theoretical knowledge with experimental data, Grey-box estimation allows for more accurate predictions of the model [24]. This approach offers a balance between white-box and black-box models [32], improving both model accuracy and practical applicability. In order to do this, the System Identification Toolbox [21] from MATLAB is going to used.

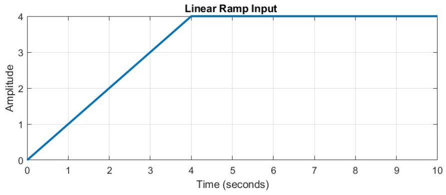


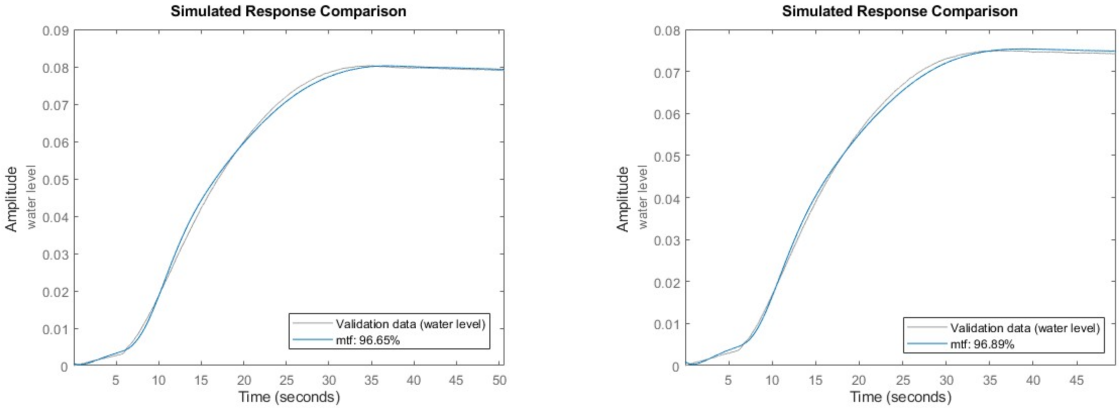
Figure 4-1: Ramp signal that was used to identify the Pool

A ramp signal (Figure 4-1) was selected as the input for system identification, in preference to a step input. While step input changes instantaneously, this is not feasible on the testbed due to the physical limitation of the servo motors, In contrast, a ramp signal varies linearly over time, allowing the servo motor to more accurately track the input. It is also important to note that the gate position is controlled for the identification, the actual input that it going to be used for the estimation is the flow rate. This is because the model that was derived previously using flow rate as the input to the model.

Results

Firstly, to analyse the fit of the models, the Normalised Root Mean Square Error (NRMSE) is going to be used, which measures how well the predicted response matches the measured data, normalised by the outputs variation. The results for the pool identification can be seen below. Figure 4-2 shows the prediction of the model with training and testing data. For the training data, multiple tests with various gate heights were used. One example of the training data for Pool 3 can be seen in Figure 4-2a. Comparing the prediction to the measured data, it can be seen that the estimated model achieved a 96.65% fit for one of the training data (Table 4-1). This could indicate the possibility of over fitting, as such a high fit could mean the model is capturing noise rather than underlying dynamics. However, looking at Figure 4-2b, which shows an equally high fit of 96.89% for the testing data, which could indicate that the model generalises well and is no overfitting. This consistent performance across both datasets suggests that the model captures the true system behaviour effectively.

Although the model performance equally well on both training and testing data, the test data comes from nearly identical setups with only the gate heights changing, which may introduce bias or data leakage. Therefore, the strong performance should be interpreted cautiously and confirmed with more varied testing scenarios, if done again.

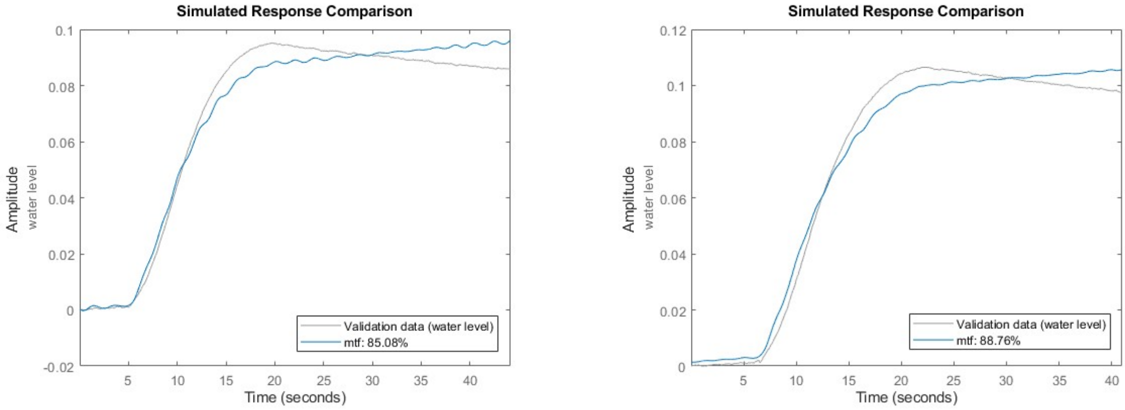


(a) Test Data

(b) Validation Data

Figure 4-2: Comparison of test and validation results for Pool 3.

Looking at the model estimation results in Figure 4-3, which were obtained using the same estimation techniques, it is evident that the model for Pool 2 demonstrates lower performance, with one of the training data fit being 85.08 % and a testing data fit of 88.76 % (Table 4-1). This discrepancy is not due to shortcomings in the estimation method but rather an issue with the physical test bed. Specifically, some of the gates in the system do not form a complete seal. As a result, even when a gate is fully closed, water can still leak through. This is especially the case with Pool 2, where water slowly drains out due to an imperfect seal to Pool 3's gate. This leakage becomes noticeable in the results from around 20 seconds onward, where a gradual reduction in water level is observed. This is not accounted for the transfer function derived, and which explains why the model struggles with predicting the water loss. In principle, such behaviour can be treated as a disturbance, which the controller should be able to expected to reject. Finally, the estimated coefficients from carrying out the identification can be seen in Table 4-2.



(a) Test Data

(b) Validation Data

Figure 4-3: Comparison of test and validation results for Pool 2.

	Expirement 1	Expirement 2	Expirement 3	Validation
Pool 1	94.81 %	94.15 %	90.93 %	94.42 %
Pool 2	87.74 %	85.08 %	89.21 %	88.76 %
Pool 3	97.33 %	96.65 %	97.26 %	96.89 %

Table 4-1: NRMSE of the estimated model vs the measured data for the Pools

	alpha	tau	omega_r	zeta
Pool 1	0.1853	2.34	2.4922	0.1156
Pool 2	0.1187	3.20	2.2629	0.1028
Pool 3	0.2279	4.21	1.5439	0.1794

Table 4-2: Estimated Values for the models of the Pools

4-2 Gates

The gates also need to be modelled accurately in order to be able to control the system. This is because the control law computes the flow rate, while the gate heights are what is controllable in the system. As a result, a model needs to be defined, which is going to relate these two values. In Chapter 2, equation 2-1 was derived to approximate the relationship between the flow rate and gate position. This approximate holds for overshot gates. This is gates where water flows over the gates.

However, the test bed at TU Delft uses undershot gates to regulate the flow in the channels. Undershot gates are gates that allow water to pass underneath them, as can be seen in Figure 4-4. Accordingly, the model needs to be modified in order to account for these gates.

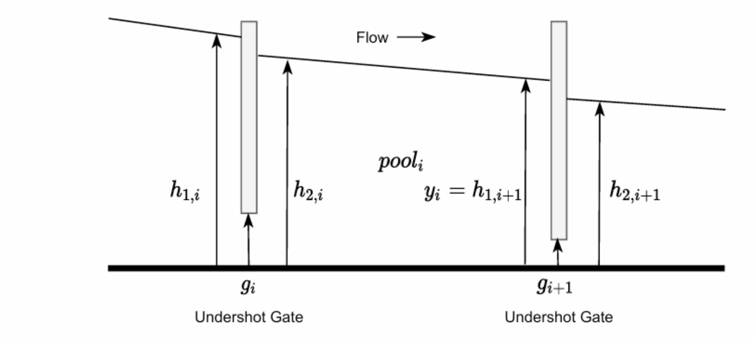


Figure 4-4: Undershot gates [1]

Assuming that the test bed unit exhibits downstream behaviour, i.e $h_{2,i}$ is smaller than $h_{1,i}$, the inflow into $pool_i$, can be expressed by [8]:

$$u_i \approx g_i \gamma_i \sqrt{h_{1,i} - h_{2,i}} \tag{4-3}$$

and the outflow of the $pool_i$, can be given by:

$$v_i \approx g_{i+1} \gamma_{i+1} \sqrt{h_{1,i+1} - h_{2,i+1}} \tag{4-4}$$

where $h_{1,i}$ is the water level right before $gate_i$, $h_{2,i}$ is the water level right after $gate_i$ and g_i is the gate opening. According to this approximation, the flow rate of the water is a function of the gate opening and also the difference in the water level before and after the gate.

In order to estimate the parameters, Non-Linear Least Squares (NLLS) was chosen. This method was justified as a result of the square root function in the model, making the relationship between the water level and the gate height non-linear.

To begin with, the function $f(\gamma_i)$ that predicts u_i, is given by:

$$\hat{u_i} = f(\gamma_i) = g_i \gamma_i \sqrt{h_{1,i} - h_{2,i}} \tag{4-5}$$

with this function defined, the cost function can now be defined. The goal of the NLLS is to find the value of γ_i that minimises the sum of squared residuals, given by:

$$min \sum_{i=1}^{n} (u_i - g_i \gamma_i \sqrt{h_{1,i} - h_{1,i}})^2 \tag{4-6}$$

with the cost function defined, the optimisation problem can be solved. To solve it, lsqnonlin function was used in MATLAB.

4-2-1 Procedure

To perform the system identification on the gates, data needs to be obtained between the flow rate and the gate height. Unfortunately, flow rate cannot be measured with the current sensors on the testbed. One way around this problem is that the flow rate can be calculated by basic mass balance between the pools. Since, the volume of water can be computed, the flowrate can also be computed subsequently.

This method is not without its disadvantages. One of the main challenges is its sensitivity to measurement noise. Since the volume is calculated from water level readings, which are already subject to sensor noise, any inaccuracies are amplified by the volume estimate. Furthermore, calculating flow rate requires differentiating the volume over time, a process that will amplify noise. As a result, the flow rate estimate can become quite noisy, as illustrated by the blue data in Figure 4-5.

In order to improve the quality of the data, a low-pass filter was applied to the flow rate signal using a cutoff frequency of 1 Hz. This filter effectively attenuates noise while preserving the underlying flow dynamics. Although the filter is not perfect, it is able to remove a substantial portion of the high-frequency fluctuations that could distort the computed volume and, consequently, the estimated flow rate. Figure 4-5 illustrates the extent to which the noise has been filtered out using the low-pass filter. Moreover, since a non-linear system identification approach is employed with a predefined analytical model, the remaining noise can be further attenuated during parameter estimation. The model structure itself helps regularize the effect of noise by fitting only the dominant trends in the data.

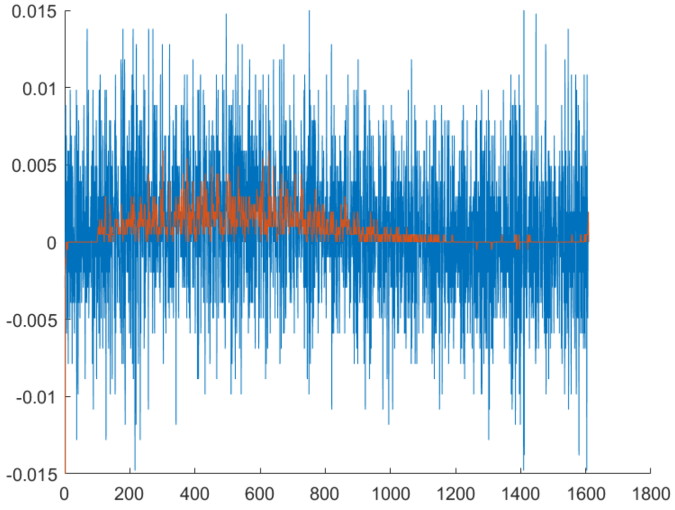


Figure 4-5: Flow rate data before and after filtering

4-2-2 Other Considerations

In order to accurately model the gate behaviour, additional physical characteristics of the system need to be incorporated into the simulation. Although the gate position should be set by the control inputs, in reality, the movement is driven by servos, which do not respond instantaneously. These gates have a finite actuation speed due to physical limitations. To capture this behaviour, a rate limiter was implemented in the simulations. This prevents unrealistic jumps in the gate positions and reflects how the system will respond in practice.

Moreover, the gate position is also physically limited. The gate height is saturated between 0 - 255 units, which corresponds to positions of the servo. Without enforcing these limits, the model could produce invalid or non-physical gate positions, especially during aggressive control inputs or large disturbances.

4-2-3 Results

The results from the non-linear least squares approximation can be seen in table 4-4. For the training data, the gates height that were chosen were 255, 100 and 50. To test the model, the gate position 150 was chosen. Figure 4-6 shows how the predicted model compares to the measured data. Table 4-3 presents the corresponding NRMSE values for both training and validation datasets, providing a quantitative measure of model accuracy.

From the NRMSE results, it can be seen that the the model achieves values between 36.91 % and 79.08 %. Looking at these results, gate positions of 255 and 50 seem to perform better than the 100 setting across the three gates. This could be due to the approximation not holding equally well across a large range of gate positions. As a result, the best-fit estimation across all three positions appears to favour the 255 and 50 settings. In the validation results, the 150 gate position shows an NRMSE similar to the 100 setting, which is reasonable since the two positions are relatively close in magnitude and likely share similar prediction characteristics. One possible reason for the relatively low NMRSE values through all the tests could be due to the noisy data. This could result in a large standard deviation, which would lead to a lower overall score.

Figure 4-6 also shows that the predicted model captures key behaviours, such as higher flow rates for more open gates and shorter drain times when gates are opened further. Overall, the obtained NRMSE values suggest the model performs reasonably well for both training and validation cases, though there is room for improvement, particularly at intermediate gate positions.

	Expirement 1255	**Expirement 2**100	**Expirement 3**50	**Validation** 150
Pool 1	79.08 %	65.61 %	72.54 %	70.43 %
Pool 2	55.43 %	36.91 %	56.05 %	37.04 %
Pool 3	67.35 %	55.06 %	65.45 %	53.00 %

Table 4-3: NRMSE of the estimated model vs the measured data for the Gate

Figure 4-7 shows the prediction of the model using testing data for a gate position of 150. The model is able to capture the main flow rate behaviour, similar to the training cases.

Towards the end of the predicted flow rate curve, the values do not approach zero as expected. In fact, the higher the gate position, the greater the deviation from zero. This is a limitation of the model. The model predicts zero flow when either the gate position or the water level

	Gate Constant
Gate 1	3.4134e-05
Gate 2	3.9928e-05
Gate 3	2.7687e-05

Table 4-4: Estimated Gate Constants

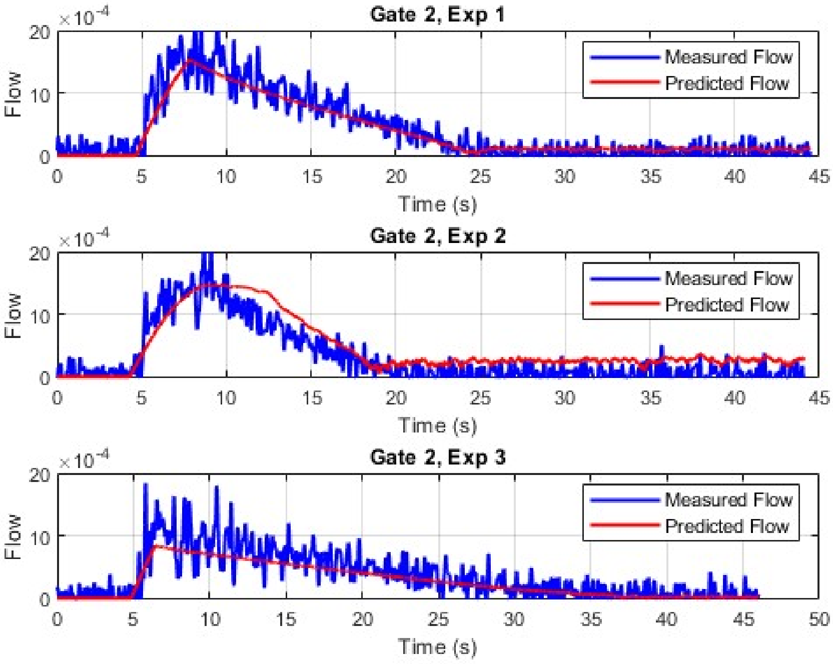


Figure 4-6: Prediction of the model comparing Predicted Flow Rate vs Measured Flow Rate using training data.

difference is zero. However, after draining, the water level difference often remains slightly non-zero due to high measurement precision and minor calibration offsets. When multiplied by a non-zero gate position, this leads to a small but non-zero predicted flow. This issue is unlikely to affect control applications, as the inverse model is used to predict gate position from a given flow rate and water level difference, making the model sufficiently reliable for prediction purposes.

The non-linear least squares model demonstrates reasonable accuracy across a range of gate positions, with particularly good performance at 255 and 50, and moderate accuracy at 100 and 150. While noise in the measured flow rate may influence the NRMSE values, the model successfully captures key physical trends and is considered sufficiently accurate for prediction and control purposes in this application.

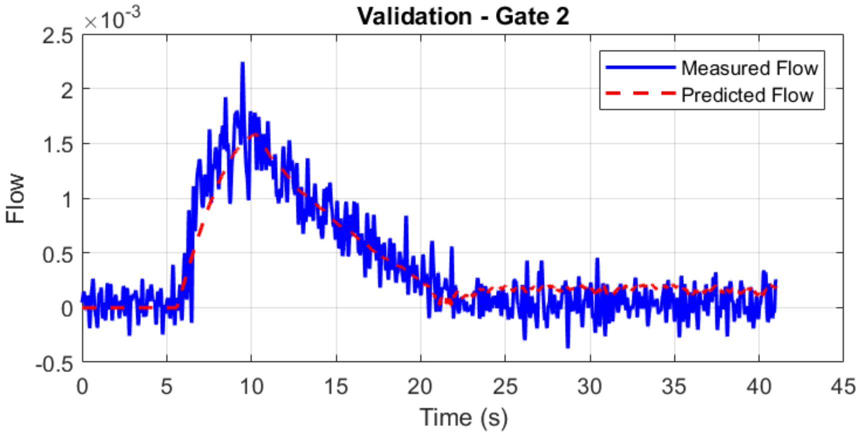


Figure 4-7: Prediction of the model comparing Predicted Flow Rate vs Measured Flow Rate using testing data.

4-3 Simulation

Since testing the water testbed system will also take a lot of time, a simulator was built. This was done in order to be able to test the controllers before actually applying them to the system. The model was made in Simulink. An overview of the model can be seen in Figure 4-8. More detailed Figure can be found in Appendix **??**.

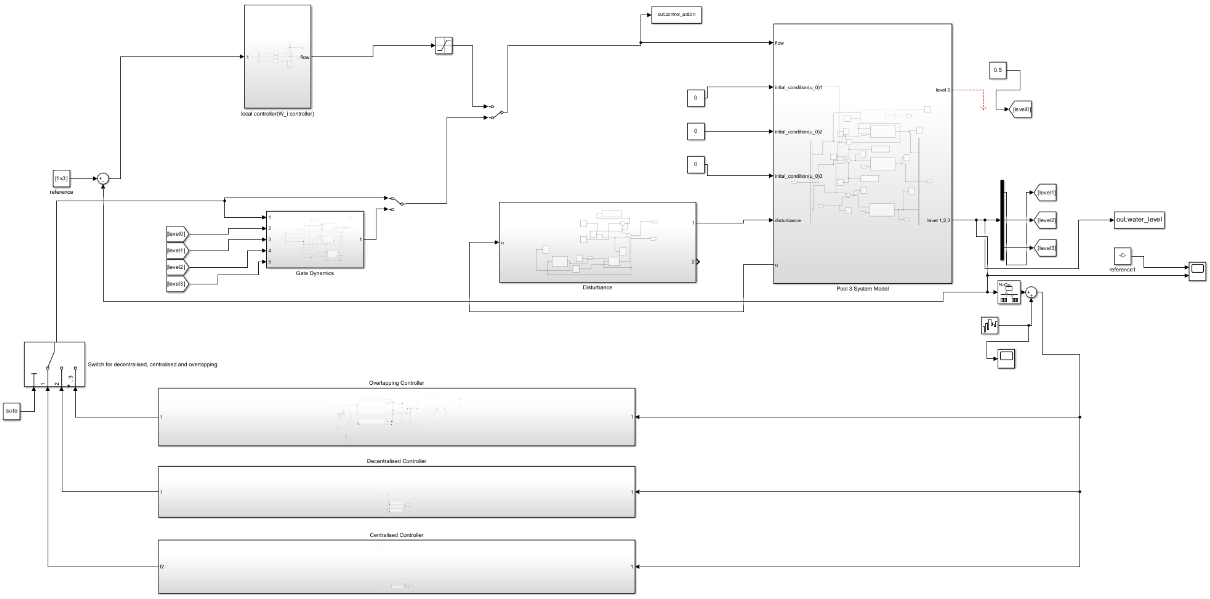


Figure 4-8: Simulink model of the Testbed

Chapter 5

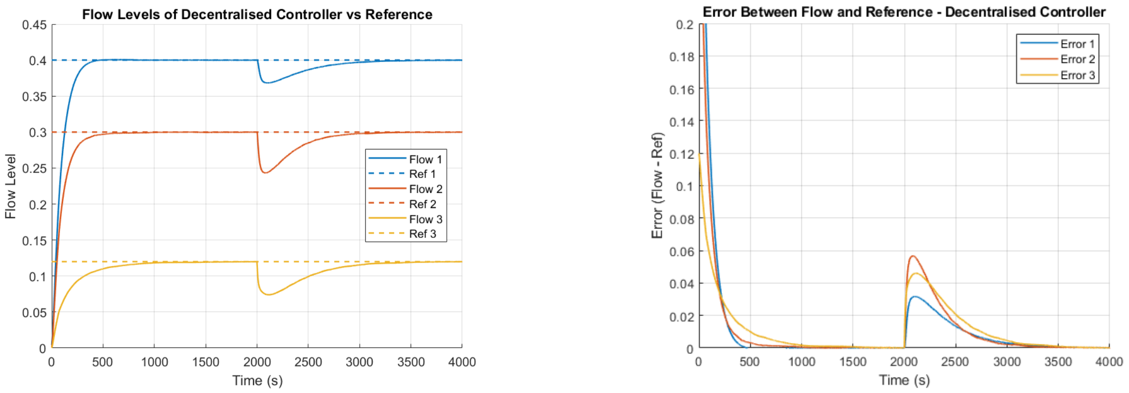
Simulation Results

To evaluate the performance of the controllers before the implementation on the real setup, simulations were carried out. Three different types of controllers were tested, centralised, decentralised and overlapping control. The advantage of the simulations were more tests could be carried out as in the real setup it took around 25 minutes per test. Multiple tests were run for all the controllers and the results were for all these tests were averaged.

5-1 3-Pool Simulation

5-1-1 Decentralised Controller

The first controller that was tested was decentralised controller. The controller that was developed was a combination of the local(W_i) and global controller(K_i) in series mentioned in Chapter 3-1. To ensure decentralisation, the global controller was designed using LQR techniques, focusing solely on its local states to maintain independence from other subsystems. The performance of this new controller is presented in Figure 5-1.



(a) Reference Tracking

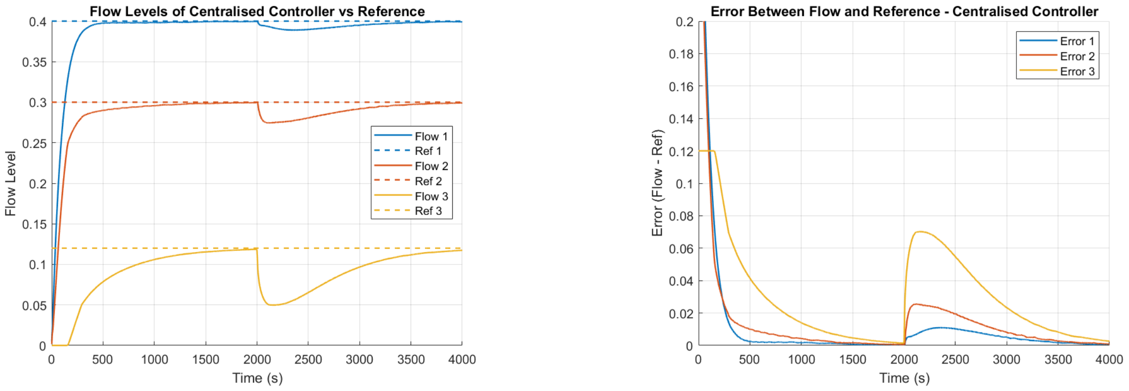
(b) Error Plot

Figure 5-1: Decentralised Controller

Figure 5-1a shows the tracking performance of the decentralised controllers when disturbance is added in the system. Looking at this figure, it can be seen that the disturbance/error gets propagated to the upstream pools, however, each pool is still able to reject the disturbance and track their own reference.

5-1-2 Centralised Controller

The second controller tested in simulation was a centralised controller, which utilises full state information from all the pools. Like the local controller, it consists of a local controller(W_i) and global controller(G_i) connected in series. The difference here is the LQR controller is a global one which communicates with all the pools. The centralised controller was implemented because, with access to complete state information, it is expected to provide close to optimal performance and serve as an upper bound for comparison. The results for this controller are presented in Figure 5-2.



(a) Reference Tracking

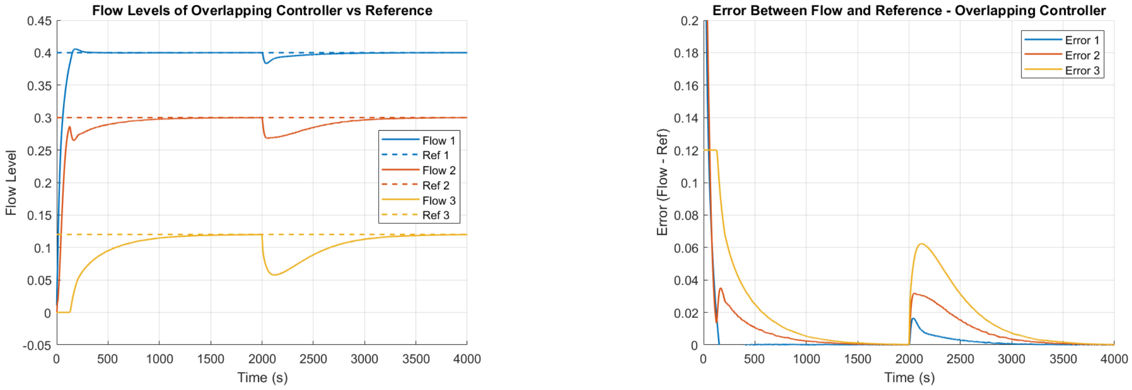
(b) Error Plot

Figure 5-2: Centralised Controller

Looking at Figure 5-2a, it can be seen that the controller is able to track the references when the disturbance is added in the system. Furthermore, looking at Figure 5-2b, it can be seen that the disturbance seems to propagate upstream, with less severe impact on the upstream pools.

5-1-3 Overlapping Controller

The last controller that was tested in simulation was the overlapping controller, which is able to only communicate with its neighbours. Similar to the other controllers, this controller uses a LQR controller in series with a PI compensator. The difference with this LQR controller as to the others is that it has state information from its neighbours only.



(a) Reference Tracking

(b) Error Plot

Figure 5-3: Overlapping Controller

Figure 5-3a shows the tracking performance of the overlapping controllers when a disturbance is added to the system. Looking at the figures, the controller can be seen to reject the noise and stabilise the system.

5-1-4 Comparisons

In evaluating the controller performance, the main focus is placed on disturbance rejection rather than reference tracking. This is because, in practical operation, the pools typically operate near their set points. Therefore, the ability of the controllers to reject disturbances and maintain stability is of greater importance than their tracking performance during setpoint transitions. To enable a statistically meaningful comparison of controller performance, 30 experiments were run under varying disturbance conditions and references for each controller to derive the statistics for the mean and standard deviation.

Figure 5-4 compares the water levels of the pools across the three controllers in response to a disturbance applied at Pool 3. Looking solely at Pool 3, where the disturbance was applied to, the centralised controller exhibits the largest deviation from the reference, followed by the overlapping controller, with the decentralised controller showing the smallest deviation. This trend is supported by the data in Table 5-1, where the average maximal errors for Pool 3 are 0.0397 ± 0.0213 m, 0.0324 ± 0.0182 m and 0.0232 ± 0.0312 m for the centralised, overlapping and decentralised controllers respectively. The standard deviations indicate that

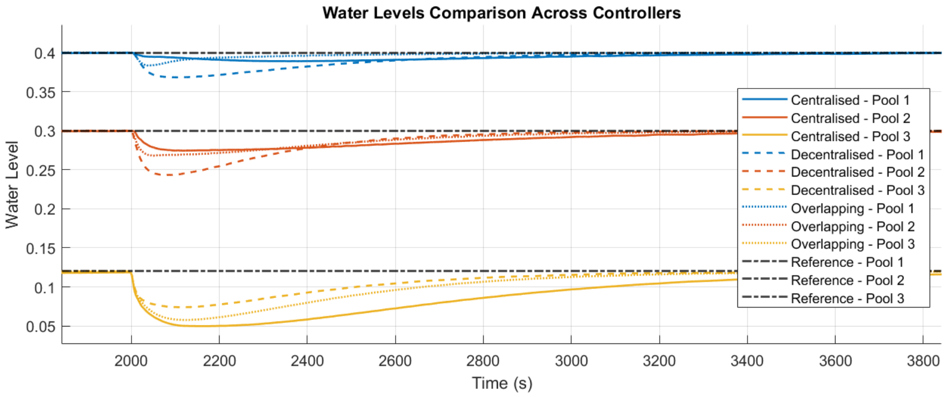


Figure 5-4: Combined disturbance rejection for the three different controllers

while the centralised controller has the highest average error, it also has the highest variability in performance.

In contrast, looking at Pool 2 and 1, which are upstream of the disturbance, the pattern switches. Here, the decentralised controller shows the highest maximal average, followed by overlapping and centralised. This is backed up by the table, indicating that the average is 0.0284 ± 0.0162 m , 0.0161 ± 0.092 m and 0.0142 ± 0.0077 m respectively for Pool 2 and 0.0159 ± 0.0091 m, 0.0082 ± 0.047 m, 0.0060 ± 0.0033 m for Pool 1. Here, the centralised controllers show low variance, suggesting better containment and consistency across different conditions.

Furthermore, another aspect that was looked into is how big the average maximal errors are in the pools compared to Pool 3. Looking at the decentralised controller, the average maximum error in Pool 2 is 124.5 % of that in Pool 3, indicating that the disturbance effect not only propagates upstream but intensifies. In Pool 1, the average maximum error reaches 69.4% of Pool 3's, further demonstrating the upstream spread of the disturbance.

In contrast, the centralised controller shows a different behaviour, where the average maximum error in Pool 2 drops to 36.5% of that in Pool 3, and in Pool 1, it is further reduced to just 15.7%. This shows the controller's ability to contain the disturbance locally and limit its propagation upstream.

The overlapping controller sits between the other two controllers, with the average maximum error in Pool 2 being 50.8% of that in Pool 3, and 25.6% in Pool 1. This suggests a moderate level of disturbance containment, offering a compromise between the decentralised and centralised approaches.

	Pool 3	**Pool 2**	**Pool 1**
Decentralised	0.0232 ± 0.0132	0.0284 ± 0.0162	0.0159 ± 0.0091
Centralised	0.0397 ± 0.0213	0.0142 ± 0.0077	0.0060 ± 0.0033
Overlapping	0.0324 ± 0.0182	0.0161 ± 0.0092	0.0082 ± 0.0047

Table 5-1: Average maximum absolute error (± variance) per pool after 2000s for each controller.

So far, comparisons have focused solely on maximal errors. However, it is also important to assess how effectively, each controller returns to its reference after a disturbance. To evaluate this, several performance metrics are introduced. The first is the Integral of Absolute Error (IAE), which treats all errors linearly, regardless of their magnitude. The equation is given by:

$$IAE = \int_0^T |e(t)|\, dt \tag{5-1}$$

While simple, this metric has two limitations, it does not penalise large errors more heavily and it does not account for how long the error persists. The second metric is the Integral of Squared Error (ISE), which addresses the first limitation by penalising larger errors more strongly due to the squaring. It equation is given by:

$$ISE = \int_0^T e^2(t)\, dt \tag{5-2}$$

Lastly, the Integral of Time-weighted Absolute Error (ITAE) penalises errors that persist over time, making it particularly useful for evaluating settling behaviour. It equation is given by:

$$ITAE = \int_0^T t \cdot |e(t)|\, dt \tag{5-3}$$

Cont	IAE	ISE	ITAE
Cent	55.80 ± 31.14	1.37 ± 1.25	152807.09 ± 88657.67
Dec	40.56 ± 23.34	0.75 ± 0.68	104696.19 ± 63970.79
Over	36.57 ± 21.02	0.75 ± 0.68	95450.67 ± 58066.71

Table 5-2: 3-pool simulation results: performance metrics after disturbance ($t \geq 2000$ s) shown as mean ± standard deviation.

The results from using these metrics can be seen in table 5-2. Looking at all the results from all the metrics, it can be seen that Overlapping performs the best showing the lowest values for IAE, ISE and ITAE, followed by Decentralised and Centralised. This indicates that the overlapping controller is able to have smaller errors and also faster error decay over time. Comparing this to the decentralised controller, it can be seen that the difference in performance is not too much, where their values for each of the metrics seem to be quite similar. Looking at the average settling time between them, it can be seen that the decentralised controller has a faster settling time of 3285.60 seconds as compared to the Overlapping controller of 3426.80 seconds. Looking at the results for the Centralised controller, it can be seen that the other two controllers seem to out perform it in all the metrics. This is contrary to the results shown in table 5-1, which indicated that other than Pool 3, the centralised controller had the best disturbance rejection. This is also backed up by Figure 5-4. The reason for the drastic difference in results could be due to the controller being better at reducing error propagation upstream. As a result, to better isolate and compare how each controller handles error propagation, the same metrics as table 5-2 are going to be used for Pool 1 and 2, disregarding Pool 3(where the disturbance originated from). This can be seen in Table 5-3.

Similar to the previous table, the overlapping controller seems to perform the best with the lowest IAE, ISE and ITAE scores. However now, the centralised controller is a close

Cont	**IAE**	**ISE**	**ITAE**
Cent	19.15 ± 10.67	0.18 ± 0.17	52788.51 ± 30537.04
Dec	24.77 ± 14.25	0.45 ± 0.41	63561.04 ± 38880.90
Over	14.05 ± 8.07	0.16 ± 0.15	36485.81 ± 22186.23

Table 5-3: Performance metrics after disturbance ($t \geq 2000$ s) for the 3-pool system, shown as mean ± standard deviation.

second, with the worst scores coming from the Decentralised controller, with the highest error accumulation and slowest convergence. This table reinforces the idea that the centralised controller has better error propagation. The centralised controller is design to optimise global performance over local set point tracking. As a result, its control strategy distributed the corrective effort across the system, which leads to a slower convergence and larger residual errors in Pool 3.

These results show that the overlapping controller gives the most balanced and efficient performance overall. It manages to reduce errors well, settle quickly, and limit how much the disturbance spreads. The centralised controller focuses more on the whole system, which means it does a good job handling upstream disturbances but ends up being less responsive where the disturbance actually happens. On the other hand, the decentralised controller reacts more strongly to the local disturbance, so it performs better in the affected pool, but this leads to more error spreading and slower settling in the other pools.

5-1-5 Effects of Noise on the System

In order to test the performance of the controllers, the performance of the controller under measurement and process noise must be tested. To enable a statistically meaningful comparison of controller performance under different types of noise, 30 experiments were run under varying references and disturbance conditions for each controller and level of noise to derive the statistics for the mean and standard deviation.

Measurement Noise

In this subsection, the effect of measurement noise is discussed. To test out various conditions, white noise was added using the Band-Limited White Noise block in Simulink to the water levels to simulate sensor noise. Different levels of noise were implemented to assess performance, corresponding to noise power of 0.00001(low), 0.00005(medium), and 0.0001(high). Table 5-4 summarises the controllers performance under low, medium, and high measurement noise levels respectively.

Looking at the performance of the controllers under low measurement noise, all three exhibit a slight deterioration compared to the no noise case. Despite this, the Overlapping controller still continues to show the best overall performance across the IAE, ISE and ITAE metrics.

Going on to the medium noise case, a further drop in performance is observed across all controller, which is expected. However, there is a notable shift in the results. At this noise level, it can be observed that the Decentralised controller begins to outperform the Overlapping

Noise	Cont	IAE	ISE	ITAE
Low	Cent	57.65 ± 30.31	1.38 ± 1.24	161016.05 ± 85226.35
	Dec	42.92 ± 22.61	0.75 ± 0.68	115054.93 ± 60998.10
	Over	40.69 ± 19.98	0.76 ± 0.68	112720.10 ± 54118.16
Medium	Cent	61.69 ± 29.07	1.40 ± 1.24	177601.27 ± 80800.36
	Dec	46.88 ± 21.73	0.76 ± 0.67	131626.63 ± 57841.98
	Over	47.68 ± 18.75	0.82 ± 0.67	140251.78 ± 50181.53
High	Cent	65.38 ± 28.16	1.44 ± 1.24	192192.76 ± 77774.50
	Dec	50.22 ± 21.06	0.79 ± 0.67	145178.18 ± 55619.32
	Over	53.54 ± 17.90	0.89 ± 0.67	162638.58 ± 47637.81

Table 5-4: 3-pool simulation results: performance metrics after disturbance with process noise ($t \geq 2000$ s) shown as mean ± standard deviation

controller across all the metrics. This gap between the performance of these two controllers increases, as the noise further increases.

A potential reason for this behaviour could stem from the performance of the observers. The Decentralised controller only requires local state information to reconstruct states, while the overlapping control also requires neighbouring state information to rebuild the states. As a result, the Overlapping observer could be more sensitive to noise, leading to overall worse performance as compared to the Decentralised case. This issue could potentially be mitigated through observer tuning, specifically tuning it to handle certain levels of measurement noise.

In conclusion, while the Overlapping controller performs best in low-noise settings, the Decentralized controller shows superior robustness to increasing noise, which is likely due to its observer being less sensitive to noise. This highlights how important it is to tune the observer to make sure the controller continues to work well.

Process Noise

Noise	Cont	IAE	ISE	ITAE
Low	Cent	59.76 ± 29.31	1.41 ± 1.23	171332.07 ± 81342.54
	Dec	51.23 ± 20.73	0.81 ± 0.67	149622.92 ± 54831.06
	Over	44.92 ± 18.80	0.79 ± 0.67	130917.18 ± 50209.20
Medium	Cent	68.60 ± 26.61	1.58 ± 1.21	208635.41 ± 72551.65
	Dec	69.08 ± 17.96	1.09 ± 0.65	219649.51 ± 46315.20
	Over	59.38 ± 16.26	1.00 ± 0.65	187799.61 ± 42509.73
High	Cent	76.63 ± 24.56	1.82 ± 1.19	240942.01 ± 66370.08
	Dec	84.11 ± 16.05	1.46 ± 0.64	276622.79 ± 40907.48
	Over	71.62 ± 14.56	1.27 ± 0.64	234313.10 ± 37690.23

Table 5-5: 3-pool simulation results: performance metrics after disturbance with process noise ($t \geq 2000$ s) shown as mean ± standard deviation

Table 5-5 shows the performance of the different controllers under various levels of process noise, corresponding to noise power of 0.000001(low), 0.000005(medium), and 0.00001(high).

Looking at low noise levels, the Overlapping controller achieves the best performance across all three metrics. The Decentralised controller is next, with slightly worse errors. The Centralised controller performs the worst according to these metrics, especially looking at the ISE score.

As the process noise increases to medium levels, the performances across all controllers decreases, with the decentralised controller having a greater decline. The IAE and the ITAE scores for the decentralised case is now greater than the Centralised case. The Overlapping controller at this noise level still outperforms the other controllers.

At the high noise levels, the trend becomes clearer. The Decentralised controller shows the highest errors across most the three metrics, which indicates that its performance is the most negatively impacted with process noise. The Centralised controller performs better than the Decentralised controller expect for the ISE metric. The Overlapping controller maintains a stronger performance showing lower scores for the IAE, ISE and ITAE compared to the other controllers.

5-2 6-Pool Simulation

The available test bed is limited to only three pools. One drawback of this setup is that due to the close communication between adjacent pools, Pool 2 effectively has access to full system information, while Pool 1 and 3 have nearly complete information as well. This close connectivity may add some bias in favour of overlapping control strategies. Additionally, expanding the system to include more pools would better reflect real-world WIS, which typically consist of several pools, which makes the simulation more realistic. Similarly to the 3-Pool simulations, 30 experiments were run under varying disturbance conditions for each controller to derive the statistics for the mean and standard deviation.

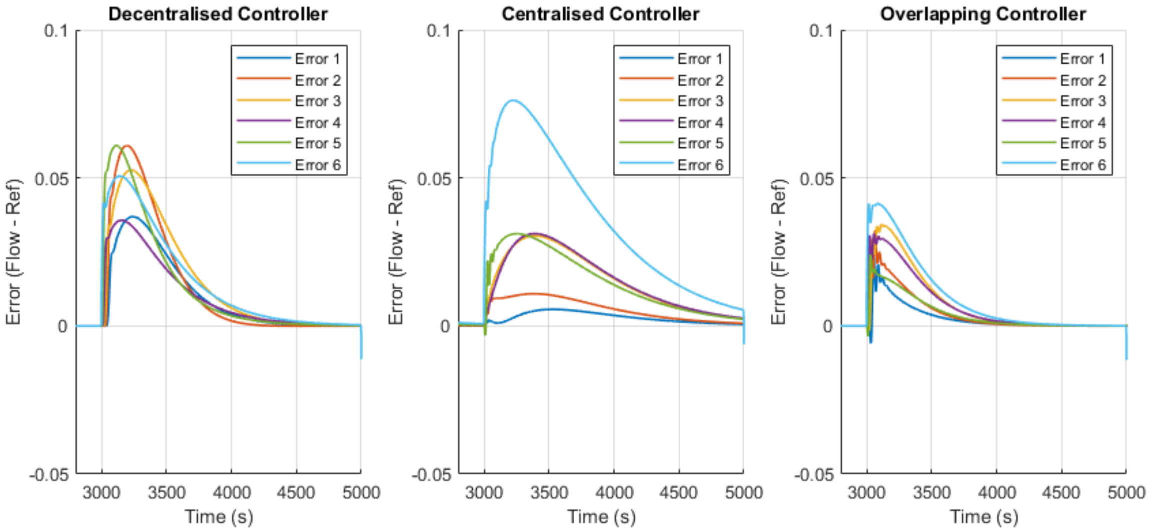


Figure 5-5: Error plots for the 6-Pool simulation with three types of controllers

Looking at the error plots in Figure 5-5, a clear performance difference can be seen between the different controllers. The centralised controller shows a clear focus on the global performance

Controller	Pool 1	Pool 2	Pool 3
Centralised	0.0035 ± 0.0019	0.0068 ± 0.0037	0.0188 ± 0.0104
Decentralised	0.0227 ± 0.0127	0.0375 ± 0.0209	0.0323 ± 0.0181
Overlapping	0.0150 ± 0.0086	0.0201 ± 0.0115	0.0212 ± 0.0120

Controller	Pool 4	Pool 5	Pool 6
Centralised	0.0193 ± 0.0106	0.0193 ± 0.0106	0.0472 ± 0.0259
Decentralised	0.0219 ± 0.0123	0.0374 ± 0.0210	0.0311 ± 0.0174
Overlapping	0.0193 ± 0.0110	0.0150 ± 0.0086	0.0257 ± 0.0147

Table 5-6: Average maximum absolute error (± variance) per pool after 3000s for each controller

of the system, with a clear focus on disturbance rejection upstream. This can be seen in the average maximal error decreasing significantly upstream, with the lowest error propagation at Pool 1 of 0.0035 m. This behaviours comes at the cost of lower local performance, which can be seen at the larger error at Pool 6 of 0.0472 m. This strategy also sleds to relatively slower response, as reflected in its higher IAE, ISE, ITAE and settling time seen in Table 5-7.

The decentralised controller shows some limitations in scalability. The average maximal error at Pool 6 is 0.0311 m, which is smaller than the centralised case. Furthermore, the IAE, ISE, ITAE and the settling time computed in Table 5-7, indicates that it has a better performance index captured to the centralised case, especially considering persistent errors. However, looking at the average maximal error across the pool, no clear trend can be drawn as the error fluctuates across the pools with no clear attenuation trend, for example Pool 2 has a higher max error than Pool 4, despite being further away from the disturbance. This lack of coordination could lead to stability issues, especially as the system scales and the disturbances propagate over large distances without begin properly damped.

The overlapping controller delivers the most balanced performance. It does not reach the lowest maximal error in Pool 1, however, it consistently maintains low error values across the series of pools, with no major spikes in the disturbance like the decentralised controller. Its IAE score is 61.112, ISE score is 0.921 and ITAE score is 234757.636. These scores are significantly lower than the centralised and decentralised controllers.

In conclusion, the 6 pool simulation results were able to reinforce and extend the findings from the 3 pool simulations. The centralised controller was shown to have great disturbance attenuation, but it came at the cost of lower local performance, especially at the pools where the disturbance is added. Although, the decentralised controller had worse disturbance attention than the centralised, it seemed to outperform the centralised approach with better local performance of the pools. However, the overlapping approach showed the most promising results over the two different simulations.

Contr	IAE	ISE	ITAE
Cent	157.415 ± 89.962	3.653 ± 3.301	664971.714 ± 409776.558
Dec	137.031 ± 78.908	3.509 ± 3.146	536557.067 ± 338332.945
Over	61.112 ± 35.157	0.921 ± 0.827	234757.636 ± 148221.939

Table 5-7: 6-pool simulation results: performance metrics after disturbance ($t \geq 3000$ s) shown as mean ± standard deviation.

5-2-1 Effects of Noise

Similar to the 3-Pool case, 30 experiments were run under varying disturbance conditions for each controller and level of noise to derive the statistics for the mean and standard deviation.

Measurement Noise

In this section, the effect on measurement noise is going to be analysed for the 6-Pool simulation.

Noise	Cont	IAE	ISE	ITAE
Low	Cent	211.512 ± 85.924	4.153 ± 3.434	$1056269.813 \pm 439950.826$
	Dec	167.174 ± 72.178	3.683 ± 3.204	$747785.350 \pm 330777.399$
	Over	112.328 ± 44.747	1.153 ± 0.888	$590043.007 \pm 261742.504$
Medium	Cent	251.957 ± 74.909	4.600 ± 3.401	$1345023.898 \pm 345900.958$
	Dec	187.486 ± 68.791	3.781 ± 3.203	$893476.604 \pm 298324.882$
	Over	149.076 ± 31.655	1.394 ± 0.854	$842379.189 \pm 153883.091$
High	Cent	284.890 ± 70.110	5.159 ± 3.383	$1573367.699 \pm 308253.823$
	Dec	203.359 ± 67.264	3.901 ± 3.202	$1005703.905 \pm 285210.117$
	Over	177.667 ± 26.664	1.695 ± 0.836	$1035688.155 \pm 108830.725$

Table 5-8: 6-pool simulation results: performance metrics after disturbance with measurement noise ($t \geq 3000$ s) shown as mean ± standard deviation

Looking at the the results from the low measurement noise, the overall performance of all the controllers deteriorate as compared to the no noise case. The Overlapping controller still demonstrates the best performance across all the three metrics. The Decentralised controller performs better than the Centralised controller, with noticeably lower IAE and ITAE scores. The Centralised controller exhibits the highest errors, suggesting that it is slower to return to the reference after noisy measurements.

Looking at the medium noise levels, the performance of all the controllers further deteriorates, as expected, with the Overlapping controller still performing the best across all the metrics. The Overlapping and Centralised controllers see quite a decline in performance as compared to the low noise case. This is not the case with the Decentralised controller, as it does not exhibit such a decline in performance. This could stem from the same issue mentioned in the 3-Pool simulation, suggesting that this could arise due to the observers not being tuned properly to reject the noise properly.

At high noise levels, the performance of all controllers deteriorates. The Overlapping controller still demonstrates a relatively strong performance, particularly in terms of IAE and ISE. However, as seen in previous cases, high measurement noise appears to affect the Overlapping and Centralised controllers more than the Decentralised one. Interestingly, due to the differing rates of performance degradation, the ITAE metric is actually lower for the Decentralised controller under high noise. This may suggest that if the noise level were increased further, the Decentralised approach could eventually outperform the others. A similar trend was observed in the 3-pool system, where the Decentralised controller became the best performer in ITAE already at medium noise levels.

This phenomenon could be explained by the relative impact of controller structure versus noise in larger systems. In smaller systems, like the 3-pool case, the structural differences between controllers are less pronounced, so noise degradation has a more immediate effect on performance rankings. In contrast, for larger systems such as the 6-pool network, the inherent differences in controller design and coordination requirements play a more dominant role in performance. As a result, the performance gap caused by control structure (e.g., centralized vs. decentralized) is initially more significant than the degradation introduced by noise. Only when the noise becomes large enough to overwhelm these structural differences does the benefit of the more noise-tolerant Decentralised controller begin to dominate.

Process Noise

In this section, the effect on process noise is going to be analysed for the 6-Pool simulation.

Noise	Cont	IAE	ISE	ITAE
Low	Cent	197.08 ± 83.20	3.86 ± 3.25	979115.59 ± 367513.56
	Dec	227.91 ± 71.23	4.08 ± 3.17	1182304.88 ± 295120.04
	Over	167.34 ± 29.21	1.54 ± 0.84	968736.12 ± 117010.92
Medium	Cent	259.37 ± 73.71	4.92 ± 3.19	1436351.10 ± 319767.16
	Dec	353.17 ± 64.26	6.31 ± 3.19	2039344.13 ± 260536.50
	Over	311.67 ± 24.09	3.97 ± 0.86	1931113.53 ± 96278.10
High	Cent	310.91 ± 67.14	6.30 ± 3.15	1800202.35 ± 288928.85
	Dec	450.86 ± 60.01	9.09 ± 3.22	2697373.59 ± 242237.10
	Over	423.10 ± 21.02	7.00 ± 0.89	2664198.58 ± 85879.06

Table 5-9: 6-pool simulation results: performance metrics after disturbance with process noise ($t \geq 3000$ s) shown as mean ± standard deviation

Analysing the effect of Process noise, several trends can be identified. Looking at the performance of the controllers under low process noise, the controllers performance seems to decline as compared to the no noise case. The Overlapping controller under these conditions is still able to perform the best across all the three metrics. The Centralised controller performs moderately well, while the Decentralised controller shows the highest error values.

Looking at the medium noise case, the performance of all controllers worsens but the rate at which they degrade is different. The Decentralised controller still performs the worst, particular according to the ISE metric. However, the Overlapping controller is not the best performing controller now, and The Centralised controller outperforms it across all the metrics listed.

At high levels of process noise, the degradation in performance becomes even more noticeable across all controllers. The Decentralised controller now shows a significant increase in all error metrics, particularly ITAE and ISE, indicating a poor ability to handle sustained disturbances in a larger interconnected system. Interestingly, while the Overlapping controller still performs better than Decentralised in IAE and ISE, it is now outperformed by the Centralised controller across all three metrics. This suggests that the full-state information available to the Centralised controller allows it to better compensate for persistent process disturbances, despite the system's increased size and complexity.

When compared to the 3-pool results, this behaviour seems to diverge. In the smaller system, the performance gap between controllers under process noise was more narrow, and the Overlapping controller maintained its advantage more consistently. However, in the 6-pool system, the increased complexity and interconnections may amplify the limitations of the Overlapping and Decentralised architectures under strong disturbances. The Centralised controller, with access to the global state, appears more robust in maintaining control performance as process noise increases, likely due to its ability to better coordinate corrective actions across the full system.

5-3 Event triggered control

In this section, the effects of implementing ETC for different triggering conditions are going to be investigated on the overlapping controller. Similar tests that were run in the previous sections, are going to be run again under varying triggering conditions to observe its impact. In addition to the metrics used in the previous sections, the amount of triggers and also the amount of triggers during settling, are going to be analysed. For both, the 3-Pool and 6-Pool simulations, 90 simulations were run for varying disturbances, references and noise levels for each alpha to derive the statistics for the mean and standard deviation.

5-3-1 3-Pool

To begin with, the ETC was tested on the 3-Pool model. Initial simulations were performance without noise to establish a clean baseline for assessing the effects of ETC. The results from the simulations are split into Table ?? and Figure 5-7.

No Noise

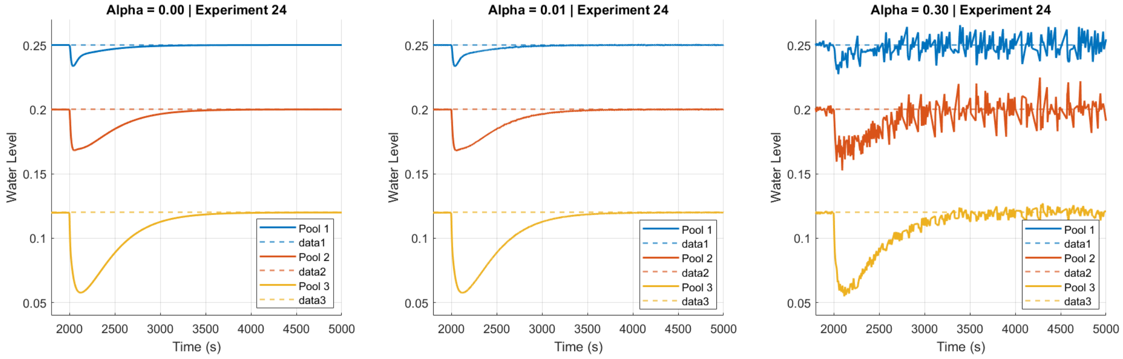


Figure 5-6: Disturbance rejection for different alpha values on 3-Pool simulation

Looking at the low alpha values (0.01 to 0.05), only a very minor degradation in all performance metrics is observed as alpha increases. Comparing the periodic control case (alpha = 0.00) to alpha = 0.05, the ISE increases slightly from 0.755 to 0.759. Similarly, the IAE rises modestly from 37.2 to 38.3, and the ITAE grows from 96,894 to 101,408. In contrast, the benefits in communication savings are substantial. At alpha = 0.01, the trigger count drops sharply to 669.9, already a 73% reduction compared to periodic control. Moreover, the settling time remains nearly unchanged, and the number of triggers during the settling phase decreases significantly, from 305.7 down to just 83.8. Furthermore, looking at Figure 5-7, specifically at the alpha 0.0 case and 0.01 case, it can be seen that they are almost identical. This highlights the efficiency of ETC at low alpha values, achieving similar control performance with far fewer communication events.

Looking at the more moderate alpha values (0.1 to 0.3), the error metrics seem to grow more noticeably (40.065 to 50.736). The ITAE values also seem to grow quite significantly, from 108285 to 149429. These values seem to indicate a significant degradation in performance. The amount of triggers also seem to reduce, however, this is at the cost the settling time increasing quite significantly. The ISE increases from 1.070 to 1.353, and the ITAE jumps from 191,147 to 234,278, indicating a further decline in performance. Meanwhile, the total trigger count reaches its lowest levels (208.5 and 208.2), and yet the settling time extends significantly to 1381.6 and 2068.2 seconds, respectively. Interestingly, the number of triggers during the settling period begins to rise again at these high alpha values, increasing to 52.0 and 92.2, possibly reflecting instability or oscillatory recovery dynamics introduced by overly sparse communication.

In summary, while increasing alpha provides substantial reduction in trigger count, especially at low values, this comes with a trade off. At moderate to high value of alpha, the control performance of the controller begins to suffer, which is seen as the settling times and error metrics increasing. Hence, it is important to select an alpha value that balances performance with communication efficiency. For the case of the 3-Pool system with no disturbance, these results seem to indicate that a low alpha value is suitable.

Noise

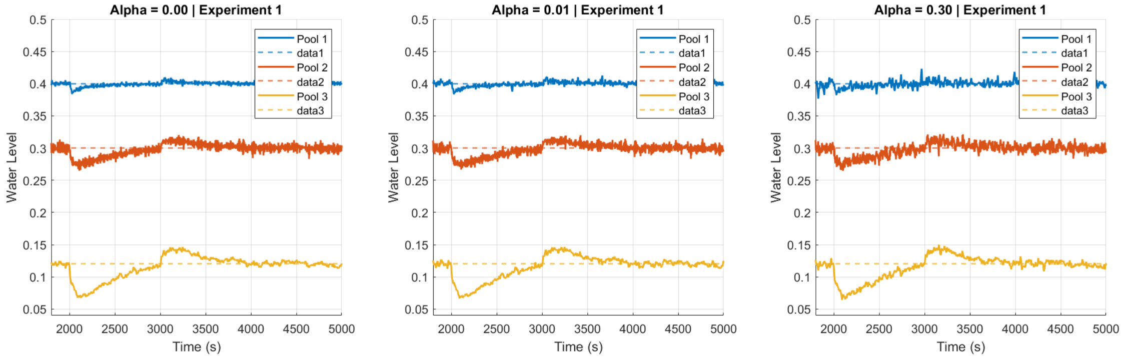


Figure 5-7: Disturbance rejection for different alpha values with noise on 3-Pool simulation

Now the effect of disturbance on ETC on the 3-Pool system can be analysed. Varying cases of both measurement and process noise were run together. Looking at the effect of alpha on controller performances under disturbance, a similar trend can be observed. Starting with low alpha values (0.01 to 0.05), there is a increase in the error metrics. For example, the ISE values increase from 0.788 for the ISE case to 0.823 for the 0.05 case. At the same time, the benefits in reduction of communication is notice, albeit not as much as the no noise case. For example, the trigger count drops from 2501 to 1141.4 at alpha 0.01, which is a 54 % reduction,

Alpha	ISE	IAE	ITAE
0.00	0.755 ± 0.677	37.224 ± 18.876	96894.719 ± 52847.007
0.01	0.757 ± 0.678	37.358 ± 18.865	97449.721 ± 52741.684
0.03	0.758 ± 0.679	37.793 ± 18.855	99262.971 ± 52518.124
0.05	0.759 ± 0.679	38.316 ± 18.871	101408.271 ± 52403.098
0.10	0.768 ± 0.684	40.065 ± 19.035	108285.544 ± 52528.581
0.30	0.860 ± 0.742	50.736 ± 21.188	149429.701 ± 59527.956
0.50	1.070 ± 0.862	62.087 ± 23.997	191146.854 ± 71005.888
0.70	1.353 ± 1.021	73.685 ± 26.948	234277.891 ± 84352.978

(a) ISE, IAE, and ITAE (mean ± std)

Alpha	Settling (s)	Trigger Count	Triggers @ Settling
0.00	611.33 ± 209.80	2501.0 ± 0.0	305.7 ± 104.9
0.01	614.75 ± 209.71	669.9 ± 58.4	83.8 ± 36.8
0.03	650.42 ± 221.62	523.2 ± 12.9	54.5 ± 19.8
0.05	666.17 ± 225.55	468.3 ± 11.1	43.9 ± 14.7
0.10	793.92 ± 381.38	381.7 ± 20.7	40.0 ± 18.4
0.30	1038.00 ± 550.58	254.4 ± 14.7	38.1 ± 26.9
0.50	1381.64 ± 514.16	208.5 ± 12.7	52.0 ± 19.4
0.70	2068.25 ± 623.03	208.2 ± 14.8	92.2 ± 27.4

(b) Settling Time and Triggering Statistics (mean ± std)

Table 5-10: Effect on ETC α on 3-pool simulation: performance metrics after disturbance with process noise ($t \geq 2000$ s)

which further reduces the bigger alpha gets. Comparing these results to the noise case, there are around a 70 % increase in triggers, however, comparing this to the periodic case, there is still significant improvements in communication costs.

This trend seems to continue as alpha increases. At moderate alpha values (0.1 to 0.3), the rate of degradation is more apparent. The IAE score increases from 51.7 to 56.6, and the ITAE grows from 157,225. Looking at the trigger count (997.42) and the triggers at settling time (109), it can also be seen that they drop, indicating that there is reduction in the trigger time. This comes with an expected increase of the settling time. At light alpha values (0.5 and 0.7), performance begins to deteriorate rapidly. The ISE value increase from 1.081 o 1.766. Furthermore, the ITAE scores increases from 215462 to 296934.

Everything else also continues to increase at these higher values, the IAE goes to 89.1 and the settling time reaches 1394.3 seconds at alpha 0.7. While these metrics clearly show a degradation in performance, the communication savings are significant. The total number of trigger events drops to just 326.1, with only 82.3 triggers occurring during the settling phase. This highlights that even under noisy conditions, ETC is still able to reduce communication costs. However, this comes at the cost of control performance, which begins to suffer as updates become more sparse.

It is also interesting to compare the settling time at high alpha values between the noise and no-noise cases. Comparing the disturbance settling times at high alpha values, it can be seen that it has a lower settling time. This might seem unusual at first, but it is likely due to how the settling time is defined. Since settling is determined based on a tolerance band, the addition of noise can make system to appear as if it has reached steady state sooner, simply because it fluctuates around the threshold. As a result, the settling time, and trigger at settling time metics computed in this section are only really comparable within the same class of noise.

To summarise, the ETC controller under disturbance continues to show strong reductions in communication, similar to the no noise case. Although performance does degrade with increasing alpha, the controller still performs significantly better than the periodic baseline in terms of communication cost. Through comparing both scenarios, it becomes clear that careful selection of the alpha value is essential, where setting it too high can result in slower convergence and increased error metrics.

Alpha	ISE	IAE	ITAE
0.00	0.788 ± 0.627	47.211 ± 18.445	140466.242 ± 50272.245
0.01	0.824 ± 0.628	51.424 ± 17.738	155968.834 ± 47915.349
0.03	0.822 ± 0.626	51.428 ± 17.762	156136.531 ± 48040.292
0.05	0.823 ± 0.622	51.678 ± 17.514	157224.963 ± 47240.605
0.10	0.827 ± 0.621	52.273 ± 17.414	159590.586 ± 46721.413
0.30	0.880 ± 0.630	56.560 ± 17.517	176183.800 ± 47273.583
0.50	1.081 ± 0.658	67.085 ± 17.762	215462.236 ± 48571.435
0.70	1.766 ± 0.846	89.116 ± 19.419	296934.117 ± 56951.164

(a) Error Metrics (mean ± std)

Alpha	Settling (s)	Trigger Count	Triggers @ Settling
0.00	802.50 ± 390.80	2501.0 ± 0.0	401.2 ± 195.4
0.01	857.25 ± 416.79	1141.4 ± 422.9	226.3 ± 167.4
0.03	870.50 ± 410.30	1087.6 ± 412.2	214.6 ± 148.1
0.05	876.83 ± 425.93	1038.2 ± 431.3	204.9 ± 147.5
0.10	900.75 ± 470.53	935.8 ± 438.7	189.1 ± 167.5
0.30	997.42 ± 385.86	613.8 ± 334.6	109.0 ± 67.7
0.50	1122.35 ± 681.74	407.5 ± 173.8	84.9 ± 61.8
0.70	1394.30 ± 824.04	326.1 ± 75.8	82.3 ± 51.9

(b) Settling Time and Trigger Metrics (mean ± std)

Table 5-11: Effect on ETC α with noise on 3-pool simulation: performance metrics after disturbance with process noise ($t \geq 2000$ s)

5-3-2 6-Pool

The 6 pool model was also tested with ETC in order to see if this can scaled up for larger systems.

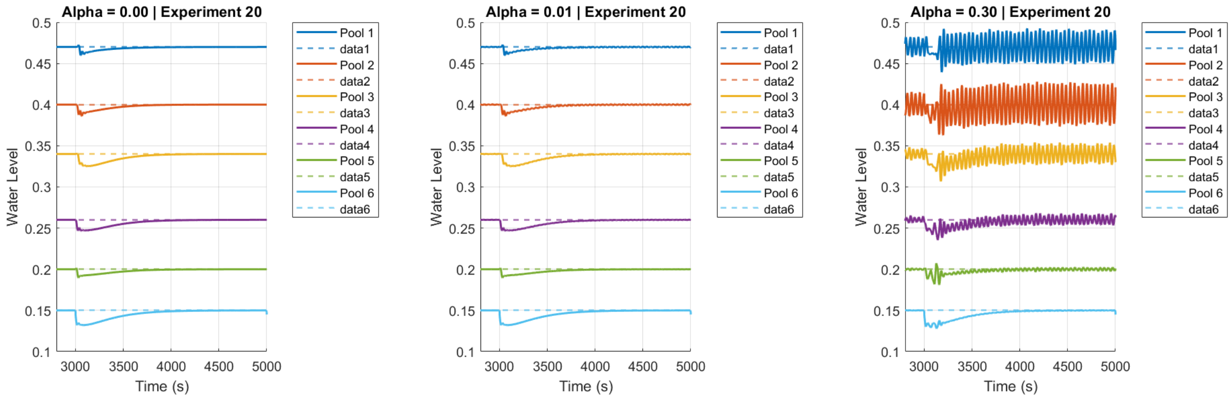


Figure 5-8: Disturbance rejection for different alpha values on 6-Pool simulation

No noise

The 6-Pool system without disturbance follows a similar trend to the 3-Pool case. As the alpha value increases, this leads to reduced communication at the cost of control performance. At low alphas (0.01 to 0.05), a gradual increase can be seen in the ISE, IAE and ITAE metrics. Looking at the communication reductions, over 60% reduction in the amount of triggers can be noticed as compared to the periodic case, while the settling time remains almost unchanged.

Unlike the 3-Pool simulations, the 6-Pool simulations seem to be more sensitive to alpha values. As alpha value increases past 0.10, the performance seems to degrade rapidly. This can be seen in the error metrics blowing up, like the ISE score going from 1.434 to 11.485. In fact, even though the alpha value becomes larger, the trigger count starts increasing again. This suggests that the ETC controller becomes unstable quicker in the 3-Pool case as opposed to the 6-Pool case. Figure 5-8 plots the water levels for different alphas. Looking at the 0.01 alpha case, similar to the 3 pool case, except for some minor oscillations, it looks almost identical to the periodic case. Indicating that while it reduces communication, not a lot of performance is lost. Looking at the 0.3 case, it can be seen that the oscillatory behaviour caused due to the ETC, is propagated upstream. This intuitively makes sense, as disturbances in a larger interconnected system can propagate more easily between pools, amplifying instability across the network.

Noise

The results from the 6-Pool system under disturbance again reveal similar trends as the previous simulations, but with a greater degradation as the alpha value increases. Looking at the low alpha values (0.01 to 0.05), it can be seen that the error metrics appear to increase relatively quickly, which was not seen in the other simulations. For example the ISE score increases from 2.385 to 4.352. During this, the communication rate drops from 5001 triggers for the periodic case to around 2463, however, unlike the 3-Pool case, the settling times already rapidly increase.

At higher alpha values (0.1 to 0.5), the controller breaks down. The error metrics all appear to blow up, especially the ISE score, which goes from 5.637 to 22.117. Looking at the triggers,

Alpha	ISE	IAE	ITAE
0.00	0.930 ± 0.819	61.954 ± 31.010	$236537.141 \pm 133621.287$
0.01	0.936 ± 0.822	68.735 ± 31.471	$285516.090 \pm 132451.984$
0.03	0.977 ± 0.844	84.098 ± 33.993	$392360.892 \pm 147007.848$
0.05	1.063 ± 0.892	100.525 ± 38.081	$504177.746 \pm 178177.118$
0.10	1.434 ± 1.120	140.336 ± 50.986	$770545.968 \pm 277686.029$
0.30	5.548 ± 4.300	315.464 ± 122.256	$1920658.338 \pm 786966.234$
0.50	11.485 ± 9.113	453.052 ± 181.079	$2818141.906 \pm 1197272.877$

(a) ISE, IAE, and ITAE metrics (mean ± std)

Alpha	Settling Time (s)	Trigger Count	Triggers @ Settling
0.00	859.75 ± 169.83	4999.0 ± 0.0	430.9 ± 84.9
0.01	859.33 ± 169.79	1885.8 ± 21.0	184.5 ± 32.8
0.03	857.83 ± 168.73	1839.3 ± 11.2	175.3 ± 31.3
0.05	858.83 ± 169.70	1819.8 ± 9.9	170.5 ± 30.8
0.10	1340.27 ± 1510.10	1804.6 ± 4.3	249.6 ± 271.3
0.30	2846.67 ± 1166.61	1822.6 ± 10.5	499.3 ± 222.1
0.50	4598.00 ± 1620.69	2003.4 ± 26.1	859.5 ± 324.6

(b) Settling time and triggering behaviour (mean ± std)

Table 5-12: Effect on ETC α on 6-pool simulation: performance metrics after disturbance with process noise ($t \geq 3000$ s)

it can be seen that the trigger count also stops reducing and starts increasing, indicating that the system is becoming more unstable. Looking at the 0.5 case, it can also be seen that the settling time is NaN, indicating that the system was not able to stabilise at all or under the simulation time. Figure 5-9 shows the water levels of the system under varying levels of alpha. Unlike the case with the no disturbance, it can be seen that oscillations plus the disturbances already start to propagate upstream in the 0.01 case.

To conclude, while ETC offers a large reduction in communications for the 6-Pool system, it comes with a much bigger trade off as compared to the 3-Pool case. The low alpha values, the controller is able to offer come reduction in communication with minimal performance drop, however, the system is noticeably more sensitive to noise. At high alpha values, the controller starts to become unstable, which is seen by the error metrics blowing up. This propagation of disturbance is also noticed in the disturbed case for 0.01. Overall, these results suggests that although ETC can be used to reduce communication costs, the stability margin of the systems reduce as these interconnected systems get larger. As a result, extra caution should be used while tuning the alpha values.

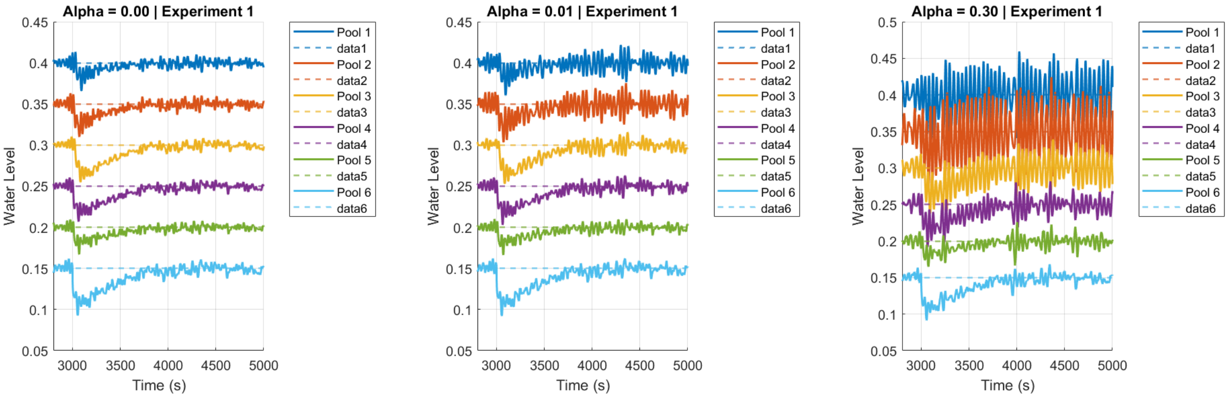


Figure 5-9: Disturbance rejection for different alpha values with noise on 6-Pool simulation

Alpha	ISE	IAE	ITAE
0.00	2.385 ± 1.415	217.991 ± 71.864	$1305667.616 \pm 460931.685$
0.01	3.505 ± 1.380	287.772 ± 57.136	$1778313.869 \pm 357229.586$
0.03	3.908 ± 1.509	305.706 ± 59.329	$1897206.041 \pm 368753.390$
0.05	4.352 ± 1.666	323.499 ± 62.112	$2013596.363 \pm 385068.442$
0.10	5.637 ± 2.239	367.850 ± 72.878	$2303330.517 \pm 454240.822$
0.30	12.281 ± 5.848	527.206 ± 120.052	$3335287.733 \pm 774783.743$
0.50	22.117 ± 10.949	688.693 ± 161.138	$4391489.650 \pm 1052562.471$

(a) ISE, IAE, and ITAE metrics (mean ± std)

Alpha	Settling Time (s)	Trigger Count	Triggers @ Settling
0.00	1268.28 ± 1151.64	5001.0 ± 0.0	635.5 ± 576.5
0.01	2019.51 ± 1518.16	2735.3 ± 505.9	610.1 ± 559.8
0.03	1778.92 ± 1009.25	2583.0 ± 459.5	471.2 ± 319.8
0.05	2633.23 ± 1832.32	2463.8 ± 413.3	645.5 ± 497.1
0.10	3332.33 ± 1631.74	2280.1 ± 332.5	702.8 ± 339.3
0.30	4686.67 ± 1231.44	2001.2 ± 197.3	896.0 ± 224.3
0.50	NaN	1850.0 ± 134.8	NaN

(b) Settling time and triggering behaviour (mean ± std)

Table 5-13: Effect on ETC α with noise on 6-pool simulation: performance metrics after disturbance with process noise ($t \geq 3000$ s)

Chapter 6

Test Bed Results

With the controllers shown to be working in simulation, the next step was to implement them on the testbed. The results from the testbed experiments are presented in this section.

6-1 Implementation

During the testbed setup, one issue became apparent. The current version of the testbed uses valves that are wirelessly controlled by a program. For some reason, this program would randomly close the valves, which posed a problem when trying to record data. Disturbance rejection for these controllers typically takes around 15 to 25 minutes, and the random valve closures often occurred midway through the experiment. As no quick fix could resolve this issue, the results obtained in this section are limited.

Due to the restricted number of tests that could be run on the testbed, a single experiment was devised to evaluate various properties of the controllers. It was decided to test reference tracking from an initial water level of 5 cm. However, since the focus of these controllers is on disturbance rejection, a valve in Pool 3 was intentionally left open at the beginning of the run. As Pool 3 fills, water simultaneously drains from it, creating a disturbance. These same conditions were tested across the three different controllers.

When moving to the testbed, some changes in the controllers needed to be modified in order to get it working on the setup. There were 2 main changes that were done. Firstly, the measurement data was noisy coming out of the setup, as a result the currently tuned observer that was used in simulation did not perform the best. The Luenberger observer for all the controllers was then tuned so that it was able to reject some of the noisy measurements. Specifically, the R matrix in the observers design was adjusted to increase the penalty on measurement noise, which helped reduce the observer gain and improve the noise rejection while maintain acceptable estimation performance.

The other change that was done compared to the simulation was that the controllers needed to be tuned. As expected they are some discrepancies with the simulation model and the actual testbed. As a result, the controllers needed to be fine tuned in order to achieve good performance. However, the controller parameters did not need to change significantly, indicating that the simulation model provides a decent approximation of the systems performance.

6-2 Results

Controller	Pool	Overshoot	Settling Time (s)	ISE	IAE	ITAE
Centralised	1	5.59	174	7.2571	41.4048	6064.36
	2	3.69	206	5.1031	35.1653	4907.44
	3	0.00	NaN	1.5346	38.1649	16769.59
Overlapping	1	9.79	192	7.4599	45.1090	7106.53
	2	11.74	178	4.6519	34.7233	5452.76
	3	0.00	958	1.5126	35.0204	13958.08
Decentralised	1	29.43	806	20.5760	115.5908	38619.62
	2	3.31	258	6.3853	46.4763	9985.35
	3	16.82	112	0.6094	19.3934	8631.64

Table 6-1: Performance Metrics for each Controller and Pool

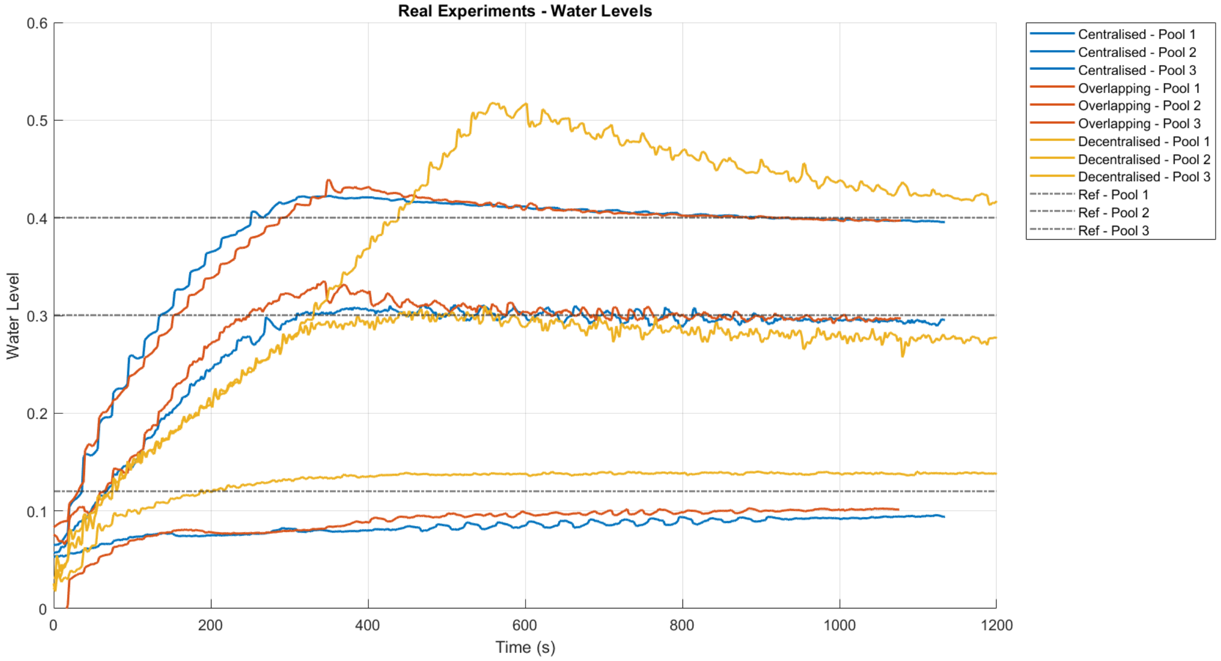


Figure 6-1: Disturbance rejection for 3 controllers on the testbed

Figure 6-1 compares the performance of the three controllers against each other, while Table 6-1 presents some of the metrics for these controllers for comparison. To begin with, it must be noted that in addition to the introduced off take disturbance in pool 3, the testbed is also subject to some inherit disturbances, in the form of gate leakage. Since the gates do not have a proper seal, water seeps through to the other pools, introducing some disturbances, that the controllers need to also reject.

Firstly, Pool 3's performance is going to be assessed across all the controllers. Looking at the decentralised case, the overshoot is 16%, with a settling time of 112 seconds. For the decentralised case, it is also important to note that there seems to be a steady state error of also around 16%. It also exhibits the lowest ISE at 0.6094, IAE at 19.39 and ITAE at 8631.64. This suggests that the controller exhibits an aggressive and responsive control strategy

Looking at the overlapping controller, there is no overshoot, with a settling time of 958 seconds, which is substantially slower than the decentralised case. The error metrics are 1.5126 for ISE, 35.0204 for IAE and 13958.08 for ITAE. These metrics indicate a more damped and coordinated response at the cost of responsiveness.

The centralised controller also shows no overshoot. The settling time was not able to be computed due to the experiment not going to completion, however, it is assumed that the settling time is the biggest out of the controllers. The error metrics are 1.5346 for ISE, 38.1649 IAE and 16769 ITAE, which is slightly higher than the overlapping controller. Similarly to the overlapping control, this controller provides a more coordinated response, however it comes at the cost of error accumulation over time which is reflected in the error metrics.

In conclusion, for Pool 3, the decentralised controller seems to perform the best in terms of the error metrics, even though it introduces an overshoot and steady state error. The next best controller is the overlapping controller, which offers a slightly better performance compared to the centralised controller. These results support the idea that the decentralised controller prioritises fast local performance, where as the communication increases, it will start favouring global stability at the cost of local performance.

Now, Pool 2's performance is going to be analysed. The centralised controller has an overshoot of 3.69%, with a settling time of 206 seconds. The error metrics for this controller are 5.1031 for ISE, 35.17 for IAE and 4,907.44 for ITAE. These values reflect a well balance approach with good disturbance rejection. The overlapping controller performs similarly, with an overshoot of 11.74%, and a settling time 178 seconds. The error metrics are 4.6519 for ISE, 34.7233 for IAE and 5452.75 for ITAE. These results are similar to the centralised case.

The decentralised controller has an overshoot of 3.31% with a settling time of 258 seconds. Looking at Figure 6-1, it can be seen that after the water level reaches the reference, it also ends up getting a steady state error. This is reflected in the error metics, which are 6.3853 for ISE, 46.4763 for IAE and 9985.35 for ITAE.

In summary, Pool 2 shows comparable results across the three controllers. The main difference is seen in the decentralised controller, which is Not able to get 0 steady state error, due to the lack of communication with the adjacent pools. As a result, there are some disturbances that the controller is not able to reject, which is reflected in the error metrics.

Pool 1 shows the largest variation in performance between the control strategies. Looking at the decentralised controller, it is seen that the highest overshoot is at 29.43%, along with a long settling time of 806 seconds. The error metrics are 20.5769 for ISE, 115.5908 for IAE and 16769.59 for ITAE. This suggests that the lack of communication between the downstream pools leads to poorer disturbance rejection. This is reflected in the results. Looking at the Figure 6-1, each pool can be seen to independently reaching their reference, this is why Pool 3 is able to fill up relatively fast. When Pool 2 reaches it reference, its gate closes which results in excess water staying in Pool 1. Due to the lack of communication, Pool 1 is not able to reject this in time, and that it why there is such a large overshoot observed.

The overlapping controller improves on this with a reduced overshoot of 9.79% and settling time of 192 seconds. The error metrics are also improved, with a score of 7.4599 for ISE, 45.1090 for IAE and 7106.53 for ITAE. These results suggest that communication with the neighbouring pools is able to improve the disturbance rejection. The centralised controller performs the best for this Pool, where its overshoot 5.59% with a settling time of 174 seconds. The error metrics are also better, with a score of 7.2571 for ISE, 41.4048 for IAE and 6064.36 for ITAE. This indicates that global communication allows to manage disturbances more efficiently.

Pool 1's performance highlights the limitation of using decentralised controllers for WISs. The system is unable to account for these large disturbances and performance suffers. The centralised controller shows to provide the best performance, as it is able to reject this disturbance the best due to global communication. The overlapping controller shows very similar behaviour to the centralised case with a slight performance drop.

In conclusion, the experimental results were able to validate the simulations, as they were able to come to the same conclusions. The decentralised controller prioritises local performance, which in some cases (Pool 3 in this case), leads to better performance. However, it also suffers from steady state errors and poor global performance. The centralised controller benefits from the full communication of the system, resulting in better global performance and better disturbance rejection as seen in Pool 1. One downside of this approach is that in some cases, it can led to slower responses such as in Pool 3. Overlapping control offers a middle ground between both approaches. It offers better coordination between the pools which leads to better disturbance rejection as seen in Pool 1, while also performing better locally as compared to the centralised case as in Pool 3. Furthermore, In the context of this three pool testbed, as mentioned in the simulation chapter, the overlapping and centralised controller have similar performance due to their communication structure being very similar. A better comparison would be to showcase these controllers on a system with more pools, however that is not possible with this testbed.

Chapter 7

Discussions and Conclusions

7-1 Conclusion

This thesis focused on investigating the use of overlapping control in the field of WIS and implementing it on the testbed unit at TU Delft. Another goal of this thesis was to also implement ETC to the system to be able to reduce to communications in the system.

The first step involved carrying out system identification on the testbed, where models were successfully obtained for both the pools and the gates. These models were then used as the basis for the simulator, that was built in Simulink to replicate the behaviour of the system.

Using the simulator, three different control approaches were tested, decentralised, overlapping and centralised. To ensure that the results were statistically meaningful, the performance of these controllers was then evaluated under various conditions. Furthermore, the system was scaled up to a 6 pools system to see how the controllers held up under a more complex interconnected system. Furthermore, decentralised ETC was also tested in both simulations.

The simulations were able to identify clear trends. The overlapping controller consistently delivered the best performance, as it had a good balance in local performance with good global performance. The decentralised controller suffered with low global performance seen in the form of error propagation up stream. Conversely, the centralised controller had the best global performance, but that came with the cost of bad local performance. These characteristics became more pronounced when switching to the 6 pool system.

With respect to communication efficiency, ETC proved to be effective in the 3-pool simulations, offering significant reductions in communication even in the presence of noise. However, when scaled up to the 6-pool system, the ETC strategy showed increased sensitivity to noise. Despite this, it was still able to achieve a substantial reduction in communication, demonstrating its potential as a viable method for communication reduction for WIS.

Finally, these controllers were implemented on the testbed. However, testing was limited due to various technical limitations. In spite of the limited results, the results obtained were promising. They served as validation to the results obtained in simulation, and similar conclusions were drawn from the experimental results. This reinforces the confidence in the simulation framework and also the implementation of overlapping control on WIS.

In conclusion, overlapping control shows strong potential in its application to WIS, offering a scalable controller, that is able to balance global and local performance needs of the system. The incorporation of ETC further enhances this by reducing communication overhead, making it suitable for future large scale WIS.

7-2 Future Work

7-2-1 Simulation

One of the goals of this thesis was to develop a simulator for the testbed. This was done for two main reasons. First, it is easier to test various controllers in simulation before implementing them on the testbed, ensuring that they will function as intended. The second reason, is that simulations are able to run at a fraction of the time as compared to the testbed, which make it a better alternative when needed to run multiple experiments.

At present, the simulator is able to assess whether controllers are capable of stabilising the system. However, it is not the most accurate representation of the testbed, as it does not account for water leakage of between the pools and gates. This difference in the behaviour of the system was especially noticeable when looking at the testbed experiments. Hence, if a more accurate simulator is wanted, these water losses need to be accounted for.

7-2-2 Control

Using the Proof in Appendix B, it can be proved that the system is stable using overlapping control. However, when referring to interconnected system, it is also important to asses the string stability of the system. This stability examines how disturbances propagate along the interconnected systems. Evaluating string stability ensures that small perturbations do not grow uncontrollably as they propagate upstream. Hence, it is recommended that this aspect be investigated further to establish a formal guarantee of string stability when using overlapping control.

Another objective of this thesis was to implement overlapping controller in order to regulate the system. In overlapping controller, each subsystem receives information from its neighbouring subsystem on both sides. The results of this controller from this thesis indicate that this approach is promising.

Since these control strategies operate over a NCS, another important goal is to reduce communication costs. One potential method for achieving this it to limit communication to only the downstream neighbouring pool. This is reasonable, as the downstream pool is pool that mainly influences the dynamics of a given pool as compared to the upstream pool. In theory, for a periodic controller, this would reduce communication costs approximately by half.

7-2-3 Testbed

One of the main limitations encountered when attempting to control the testbed was the behaviour of the valves, which would close unexpectedly and at random intervals. As a result, only a small portion of the experiments conducted were usable, as most were interrupted before the system could reach a stable state. This issue was particularly problematic given that each experiment lasted approximately 20 minutes, creating a significant likelihood that the valves would shut off at some point during the process.

It is therefore strongly recommended that this valve issue be resolved as a priority, as its current behaviour renders the testbed largely unusable for reliable experimentation

Once this issue is resolved, it is also recommended to run similar tests that were run in simulation, on the testbed. This would provide a better understanding wither overlapping control can effectively applied to to real WIS. Furthermore, ETC was not implemented on the testbed due to the issues mentioned above. It is therefore suggested that ETC also to be applied, in order to examine whether similar communication reduction to those observed in simulation can be achieved in practice.

Appendix A

Simulink Model

A-1 3-Pool Simulator

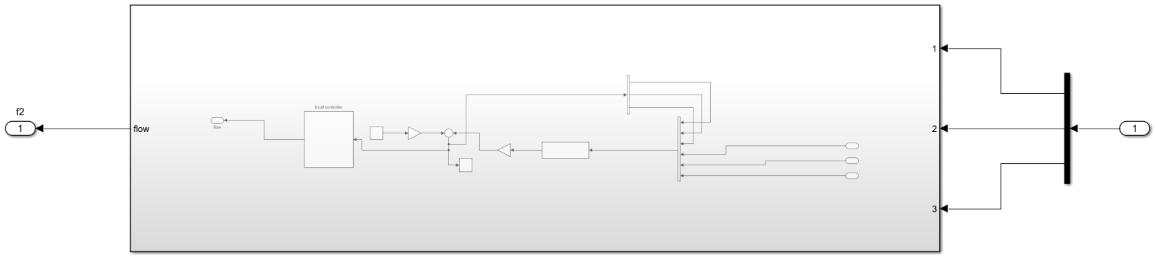


Figure A-1: Centralised Controller

A-2 6-Pool Simulator

A-3 Code

All the code developed for this thesis is organised into two main repositories for accessibility and reproducibility. These repositories contain the scripts, models and functions used throughout the research, including system identification, simulations and experimental control on the testbed.

The first repository focuses on system identification and simulation aspects. It includes all MATLAB and Simulink code used for model identification of the WIS pools and gates, as well as the simulation environments for testing controllers in 3-pool and 6-pool configurations. This encompasses scripts for data processing, model fitting and performance metric calculations such as NRMSE, IAE, ISE, and ITAE.

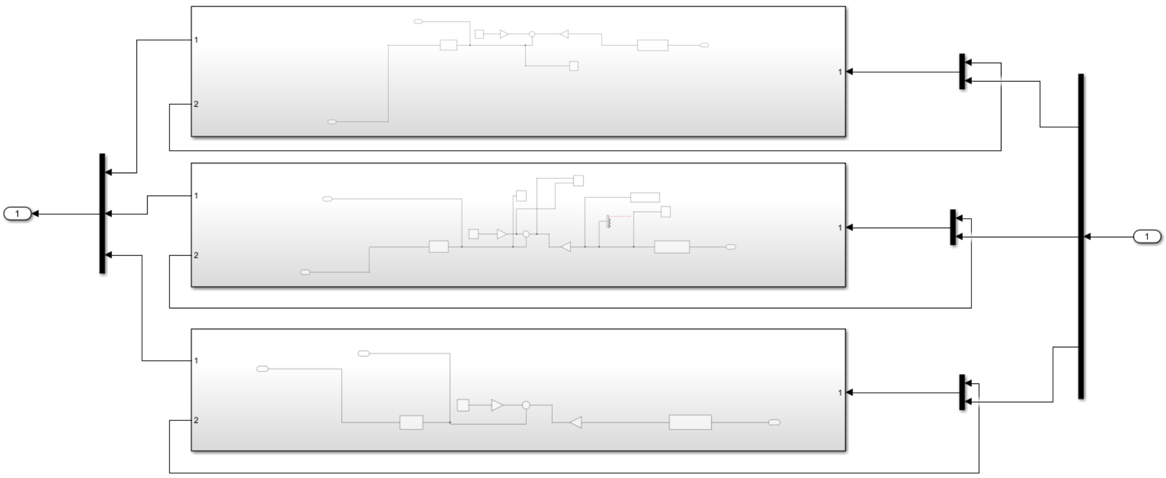


Figure A-2: Decentralised Controller

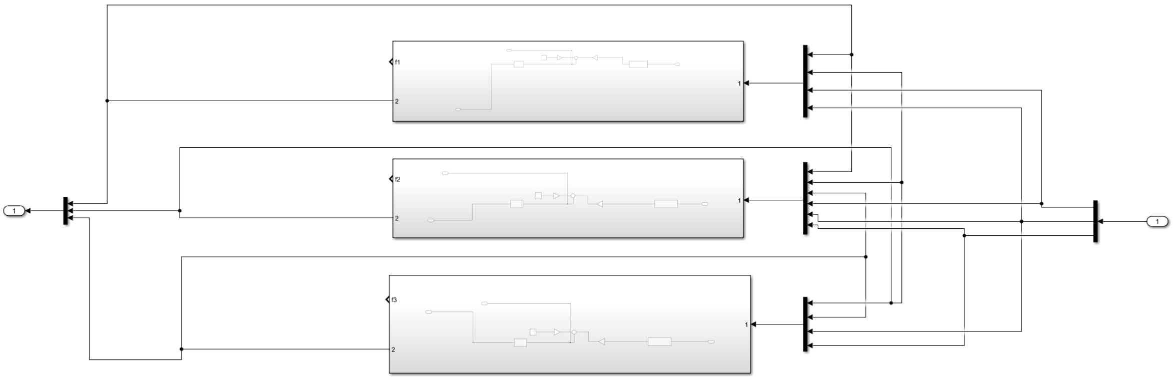


Figure A-3: Overlapping Controller

https://github.com/placeholder/identification-simulation

The second repository is dedicated to the testbed implementation. It contains code for logging sensor data from the TU Delft water testbed unit, converting raw logs into CSV files for analysis, and the Python-based control application. The control code implements the overlapping event-triggered controller, including LQR synthesis and Luenberger observers, enabling real-time regulation of water levels and flow rates on the physical setup.

https://github.com/placeholder/testbed-control

These repositories are structured with clear documentation, including README files detailing setup instructions, dependencies, and usage examples to facilitate replication of the results.

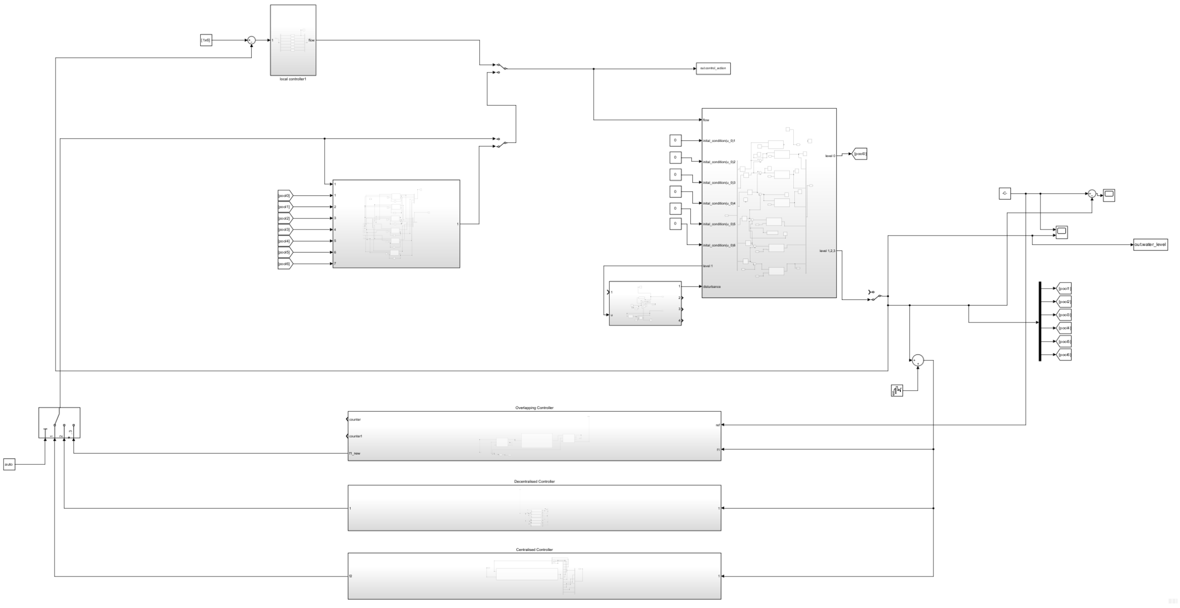


Figure A-4: 6-Pool Simulator in Simulink

Appendix B

Proof

In this chapter, we prove that the expanded system is an expansion of the original system. According to Theorem 2.13 from [14], $\tilde{S}$ is an expansion of S if the following conditions hold for all $i = 1, 2, \ldots, \tilde{n}$:

1. $UM^{i-1}V = 0$
2. $UM^{i-1}NR = 0$
3. $SLM^{i-1}V = 0$
4. $SLM^{i-1}NR = 0$

To begin with this proof, all the matrices can be defined first.

$$V = \begin{bmatrix} I_{n_1} & 0 & 0 \\ 0 & I_{n_2} & 0 \\ 0 & I_{n_2} & 0 \\ 0 & 0 & I_{n_3} \\ 0 & 0 & I_{n_3} \end{bmatrix} \qquad \in \mathbb{R}^{(n_1+2n_2+2n_3)\times(n_1+n_2+n_3)}$$

$$R = \begin{bmatrix} I_{m_1} & 0 & 0 \\ 0 & I_{m_2} & 0 \\ 0 & I_{m_2} & 0 \\ 0 & 0 & I_{m_3} \\ 0 & 0 & I_{m_3} \end{bmatrix} \qquad \in \mathbb{R}^{(m_1+2m_2+2m_3)\times(m_1+m_2+m_3)}$$

$$T = \begin{bmatrix} I_{p_1} & 0 & 0 \\ 0 & I_{p_2} & 0 \\ 0 & I_{p_2} & 0 \\ 0 & 0 & I_{p_3} \\ 0 & 0 & I_{p_3} \end{bmatrix} \qquad \in \mathbb{R}^{(p_1+2p_2+2p_3)\times(p_1+p_2+p_3)}$$

$$U = \begin{bmatrix} I_{n_1} & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2}I_{n_2} & \frac{1}{2}I_{n_2} & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2}I_{n_3} & \frac{1}{2}I_{n_3} \end{bmatrix} \in \mathbb{R}^{(n_1+n_2+n_3)\times(n_1+2n_2+2n_3)}$$

$$Q = \begin{bmatrix} I_{m_1} & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2}I_{m_2} & \frac{1}{2}I_{m_2} & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2}I_{m_3} & \frac{1}{2}I_{m_3} \end{bmatrix} \in \mathbb{R}^{(m_1+m_2+m_3)\times(m_1+2m_2+2m_3)}$$

$$S = \begin{bmatrix} I_{p_1} & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2}I_{p_2} & \frac{1}{2}I_{p_2} & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2}I_{p_3} & \frac{1}{2}I_{p_3} \end{bmatrix} \in \mathbb{R}^{(p_1+p_2+p_3)\times(p_1+2p_2+2p_3)}$$

$$M = \begin{bmatrix} 0 & A_{12} & -A_{12} & 0 & 0 \\ 0 & A_{22} & -A_{22} & \frac{1}{2}A_{23} & -\frac{1}{2}A_{23} \\ 0 & -A_{22} & A_{22} & -\frac{1}{2}A_{23} & \frac{1}{2}A_{23} \\ 0 & 0 & 0 & \frac{1}{2}A_{33} & -\frac{1}{2}A_{33} \\ 0 & 0 & 0 & -\frac{1}{2}A_{33} & \frac{1}{2}A_{33} \end{bmatrix} \in \mathbb{R}^{(n_1+2n_2+2n_3)\times(n_1+2n_2+2n_3)}$$

$$N = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2}B_{22} & -\frac{1}{2}B_{22} & 0 & 0 \\ 0 & -\frac{1}{2}B_{22} & \frac{1}{}B_{22} & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2}B_{33} & -\frac{1}{2}B_{33} \\ 0 & 0 & 0 & -\frac{1}{2}B_{33} & \frac{1}{2}B_{33} \end{bmatrix} \in \mathbb{R}^{(n_1+2n_2+2n_3)\times(n_1+2n_2+2n_3)}$$

$$L = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2}C_{22} & -\frac{1}{2}C_{22} & 0 & 0 \\ 0 & -\frac{1}{2}C_{22} & \frac{1}{2}C_{22} & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2}C_{33} & -\frac{1}{2}C_{33} \\ 0 & 0 & 0 & -\frac{1}{2}C_{3} & \frac{1}{2}C_{33} \end{bmatrix} \in \mathbb{R}^{(n_1+2n_2+2n_3)\times(n_1+2n_2+2n_3)}$$

Condition 1: $UM^{i-1}V = 0$ for all $i = 1, 2, \ldots, \tilde{n}$

We begin with the base case. Using

$$MV = \begin{bmatrix} 0 & A_{12} & -A_{12} & 0 & 0 \\ 0 & A_{22} & -A_{22} & \frac{1}{2}A_{23} & -\frac{1}{2}A_{23} \\ 0 & -A_{22} & A_{22} & -\frac{1}{2}A_{23} & \frac{1}{2}A_{23} \\ 0 & 0 & 0 & \frac{1}{2}A_{33} & -\frac{1}{2}A_{33} \\ 0 & 0 & 0 & -\frac{1}{2}A_{33} & \frac{1}{2}A_{33} \end{bmatrix} \begin{bmatrix} I_{n_1} & 0 & 0 \\ 0 & I_{n_2} & 0 \\ 0 & I_{n_2} & 0 \\ 0 & 0 & I_{n_3} \\ 0 & 0 & I_{n_3} \end{bmatrix}$$

$$= \begin{bmatrix} 0 & A_{12} - A_{12} & 0 \\ 0 & A_{22} - A_{22} & A_{23} - A_{23} \\ 0 & -A_{22} + A_{22} & -A_{23} + A_{23} \\ 0 & 0 & A_{33} - A_{33} \\ 0 & 0 & -A_{33} + A_{33} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Thus $MV = 0$. It follows that for all $i = 1, 2, \ldots, \tilde{n}$,

$$M^i V = M^{i-1}(MV) = M^{i-1} \cdot 0 = 0,$$

and therefore,

$$UM^{i-1}V = U \cdot 0 = 0.$$

This proves that the first condition $UM^{i-1}V = 0$ holds for all $i = 1, 2, \ldots, \tilde{n}$.

Condition 2: $UM^{i-1}NR = 0$ for all $i = 1, 2, \ldots, \tilde{n}$

We begin by looking at NR

$$\begin{aligned} NR &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2}B_{22} & -\frac{1}{2}B_{22} & 0 & 0 \\ 0 & -\frac{1}{2}B_{22} & \frac{1}{2}B_{22} & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2}B_{33} & -\frac{1}{2}B_{33} \\ 0 & 0 & 0 & -\frac{1}{2}B_{33} & \frac{1}{2}B_{33} \end{bmatrix} \begin{bmatrix} I_{m_1} & 0 & 0 \\ 0 & I_{m_2} & 0 \\ 0 & I_{m_2} & 0 \\ 0 & 0 & I_{m_3} \\ 0 & 0 & I_{m_3} \end{bmatrix} \\ &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & \frac{1}{2}B_{22} - \frac{1}{2}B_{22} & 0 \\ 0 & -\frac{1}{2}B_{22} + \frac{1}{2}B_{22} & 0 \\ 0 & 0 & \frac{1}{2}B_{33} - \frac{1}{2}B_{33} \\ 0 & 0 & -\frac{1}{2}B_{33} + \frac{1}{2}B_{33} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \end{aligned}$$

Thus $NR = 0$. It follows that for all $i = 1, 2, \ldots, \tilde{n}$,

$$UM^{i-1}NR = UM^{i-1} \cdot 0 = 0$$

Therefore, Condition 2 holds trivially.

Condition 3: $SLM^{i-1}V = 0$ for all $i = 1, 2, \ldots, \tilde{n}$

We start by computing the product SL.

$$\begin{aligned} SL &= \begin{bmatrix} I_{p_1} & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2}I_{p_2} & \frac{1}{2}I_{p_2} & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2}I_{p_3} & \frac{1}{2}I_{p_3} \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2}C_{22} & -\frac{1}{2}C_{22} & 0 & 0 \\ 0 & -\frac{1}{2}C_{22} & \frac{1}{2}C_{22} & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2}C_{33} & -\frac{1}{2}C_{33} \\ 0 & 0 & 0 & -\frac{1}{2}C_{33} & \frac{1}{2}C_{33} \end{bmatrix} \\ &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{4}C_{22} - \frac{1}{4}C_{22} & -\frac{1}{4}C_{22} + \frac{1}{4}C_{22} & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{4}C_{33} - \frac{1}{4}C_{33} & -\frac{1}{4}C_{33} + \frac{1}{4}C_{33} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{aligned}$$

This $SL = 0$. As a result, it follow that for all $i = 1, 2, \ldots, \tilde{n}$,

$$SLM^{i-1}V = (SL)M^{i-1}V = 0 \cdot M^{i-1}V = 0.$$

Hence, Condition 3 holds trivially.

Condition 4: $SLM^{i-1}NR = 0$ for all $i = 1, 2, \ldots, \tilde{n}$

We have already established that

$$SL = 0 \quad \text{and} \quad NR = 0.$$

Therefore, for all $i = 1, 2, \ldots, \tilde{n}$,

$$SLM^{i-1}NR = (SL)M^{i-1}(NR) = 0 \cdot M^{i-1} \cdot 0 = 0.$$

Hence, Condition 4 holds trivially.

Bibliography

[1] Boot Bas. Reducing wireless control communication for a water irrigation system. Master's thesis, TU Delft, 2021.

[2] M.G. Bos. *Discharge measurement structures*. Number 20 in Publication / International Institute for Land Reclamation and Improvement. ILRI, Netherlands, 1989.

[3] Claude Brezinski and Jeannette Van Iseghem. Padé approximations. volume 3 of *Handbook of Numerical Analysis*, pages 47–222. Elsevier, 1994.

[4] Michael Cantoni, Erik Weyer, Yuping Li, Su Ki Ooi, Iven Mareels, and Matthew Ryan. Control of large-scale irrigation networks. *Proceedings of the IEEE*, 95(1):75–91, 2007.

[5] Mieczyslaw Chalfen and Andrzej Niemiec. Analytical and numerical solution of saint-venant equations. *Journal of Hydrology*, 86(1):1–13, 1986.

[6] Cool Farm Alliance. How much water is used for farming?, 2024. Accessed: 2024-01-31.

[7] M.C.F. Donkers and W.P.M.H. Heemels. Output-based event-triggered control with guaranteed -gain and improved event-triggering. In *49th IEEE Conference on Decision and Control (CDC)*, pages 3246–3251, 2010.

[8] Karin Eurén. System identification of irrigation channels with overshot and undershot gates. 01 2004.

[9] Federico Ferrari, Marco Zimmerling, Lothar Thiele, and Olga Saukh. Efficient network flooding and time synchronization with glossy. In *Proceedings of the 10th ACM/IEEE International Conference on Information Processing in Sensor Networks*, pages 73–84, 2011.

[10] Aziz Haffaf, Fatiha Lakdja, Rachid Meziane, and Djaffar Ould Abdeslam. Study of economic and sustainable energy supply for water irrigation system (wis). *Sustainable Energy, Grids and Networks*, 25:100412, 2021.

[11] W. P. M. H. Heemels, M. C. F. Donkers, and Andrew R. Teel. Periodic event-triggered control for linear systems. *IEEE Transactions on Automatic Control*, 58(4):847–861, 2013.

[12] W.P.M.H. Heemels and M.C.F. Donkers. Model-based periodic event-triggered control for linear systems. *Automatica*, 49(3):698–711, 2013.

[13] M. Ikeda, D. D. Šiljak, and D. E. White. Decentralized control with overlapping information sets. *Journal of Optimization Theory and Applications*, 34(2):279–310, June 1981.

[14] M. Ikeda and D.D. Šiljak. Overlapping decentralized control with input and output inclusion. *IFAC Proceedings Volumes*, 17(2):1117–1122, 1984. 9th IFAC World Congress: A Bridge Between Control Science and Technology, Budapest, Hungary, 2-6 July 1984.

[15] Ganchev ivaylo. Traffic efficient control of a water irrigation system. Master's thesis, TU Delft, 2023.

[16] Jan Lont Jacob. Wireless event-triggered control for water irrigation systems. Master's thesis, TU Delft, 2020.

[17] Mohammad Aquib Wakeel Khan, Mohd Muzammil Zubair, Haleema Saleem, Alaa Al-Hawari, and Syed Javaid Zaidi. Modeling of osmotically-driven membrane processes: An overview. *Desalination*, 573:117183, 2024.

[18] Yuping Li and Michael Cantoni. Distributed controller design for open water channels. *IFAC Proceedings Volumes*, 41(2):10033–10038, 2008. 17th IFAC World Congress.

[19] Yuping Li, Michael Cantoni, and Erik Weyer. Design of a centralized controller for an irrigation channel using h loop-shaping. 2004.

[20] Jan Lunze and Daniel Lehmann. A state-feedback approach to event-based control. *Automatica*, 46(1):211–215, 2010.

[21] MathWorks. System identification toolbox, 2025. Accessed: 2025-01-28.

[22] Wondimu Musie and Girma Gonfa. Fresh water resource, scarcity, water salinity challenges and possible remedies: A review. *Heliyon*, 9(8):e18685, 2023.

[23] Romain Postoyan, Adolfo Anta, Dragan Nešić, and Paulo Tabuada. A unifying lyapunov-based framework for the event-triggered control of nonlinear systems. In *2011 50th IEEE Conference on Decision and Control and European Control Conference*, pages 2559–2564, 2011.

[24] Jiri Rehor. Grey-box model identification—control relevant approach. volume 43, pages 117–122, 08 2010.

[25] J.H. Sandee, P.M. Visser, and W.P.M.H. Heemels. Analysis and experimental validation of processor load for event-driven controllers. In *Proc. 2006 IEEE Conference on Control Applications*, number 1636734 (P, pages 1879–1884, United States, 2006. IEEE. IEEE Catalog number: 06TH8902; IEEE International Conference on Control Applications, CCA 2006 ; Conference date: 04-10-2006 Through 06-10-2006.

[26] Ismail Serageldin. Water: Conflicts set to arise within as well as between states. *Nature*, 459(7244):163, May 2009.

[27] José M. Tarjuelo, Juan A. Rodriguez-Diaz, Ricardo Abadía, Emilio Camacho, Carmen Rocamora, and Miguel A. Moreno. Efficient water and energy use in irrigation modernization: Lessons from spanish case studies. *Agricultural Water Management*, 162:67–77, 2015.

[28] Matteo Trobinger, Gabriel De Albuquerque Gleizer, Timofei Istomin, Manuel Mazo, Amy L. Murphy, and Gian Pietro Picco. The wireless control bus: Enabling efficient multi-hop event-triggered control with concurrent transmissions. *ACM Transactions on Cyber-Physical Systems*, 6(1), 2022.

[29] Matteo Trobinger, Timofei Istomin, Amy L. Murphy, and Gian Pietro Picco. Competition: Crystal clear: Making interference transparent. In *European Conference/Workshop on Wireless Sensor Networks*, 2018.

[30] Erik Weyer. System identification of an open water channel. *IFAC Proceedings Volumes*, 33(15):265–270, 2000. 12th IFAC Symposium on System Identification (SYSID 2000), Santa Barbara, CA, USA, 21-23 June 2000.

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

[32] Metodi P. Yankov, Mehran Soltani, Andrea Carena, Darko Zibar, and Francesco Da Ros. A comparison between black-, grey- and white-box modeling for the bidirectional raman amplifier optimization, 2024.

[33] Fatemeh Zarmehi, Ali Tavakoli, and Majid Rahimpour. On numerical stabilization in the solution of saint-venant equations using the finite element method. *Computers Mathematics with Applications*, 62(4):1957–1968, 2011.

Glossary

List of Acronyms