Identification and MPC control of a Water Delivery Canal

Decentralized and Reconfigurable Control for Water delivery Multipurpose Canal System - PTDC/EEA-CRO/102102/2008

Filipe Alexandre Diogo Maia Cadete
Supervisor: João M. Lemos
INESC-ID

February 15, 2011
**Acknowledgement** This report presents the results of a BI internship from 1/Sept./2010 till 31/Jan./2011 within project *Decentralized and Reconfigurable Control for Water delivery Multipurpose Canal System*, FCT contract PTDC/EEA-CRO/102102/2008. The work reported has been supported by this project and INESC-ID multiannual funding through the PIDAC program funds.
Chapter 1

Experimental Water Delivery Canal

1.1 Plant Description

The main target of this project is the control of the experimental canal located in Évora and belonging to Universidade de Évora, Núcleo de Hidraulica e Controlo de Canais (NUHCC).

This canal (Figure 1.1) has the following characteristics:

![Figure 1.1: Estructure of the experimental canal](image)

Length of 150m and 900mm height divided in four pools that interconnect with each other, with three orifice gates and one overflow gate situated in the last pool. This gates are structures that permit the control of water level in each section. This is done by letting more (or less) fluid flow beneath the orifice gates or over the overshoot gate. For each pool an offtake is used to simulate discharges made by non-controlled human interaction. The offtake is equipped with a flow meter and a controled electrical butterfly valve. The offtakes and the overshoot (last) gate discharges into the parallel return canal, which in turn will feed the main reservoir that delivers the needed water into the system. The input flow control is done using a MONOVAR valve. Then, the slope of the canal allows a continuous flow of the water (0.09m$^3$s$^{-1}$ of maximum nominal design) until it reaches the final pool.

The canal has three level sensors per pool, that allow to measure values between 0mm and 900mm corresponding to the canal border. This level sensors are placed at the beginning, middle and end of the pool. The gates are actuated using commands delivered by the user, based on the data observed on the sensors. All the information about the gates and water level are in millimeters, the MONOVAR and offtake flows can be expressed in m$^3$s$^{-1}$. 
1.2 Nonlinear model of the canal

The dynamics of the open-canal hydraulic system is described by the Saint-Venant equations. The nature of this nonlinear differential equations of Saint-Venant that provide the mathematical model is complex to manipulate. In order to obtain a model of the system useful for control design, it is necessary to linearize the equations around an equilibrium point. This is done here using process identification methods. This procedure will be shown in the next chapter. In the present section the nonlinear equations will be presented, related to the pool and gate structures that form the dynamics of the canal.

1.2.1 Pools structures

It is important to realize the mathematical fundamentals used to develop the model of the canal. The nonlinear model of the pools is provided by the Saint-Venant equations (Eq 1.1):

\[
\begin{align*}
\frac{\partial A}{\partial t} + \frac{\partial Q}{\partial x} &= q_l \\
\frac{\partial Q}{\partial t} + \frac{\partial Q^2}{A} \frac{\partial}{\partial x} + gA(S_f - S_0) &= Kq_l V
\end{align*}
\]

\[S_f = \frac{Q|Q|^n}{A^2R^{4/3}}\]  

Table 1.1: Variable description

<table>
<thead>
<tr>
<th>Variables</th>
</tr>
</thead>
<tbody>
<tr>
<td>t</td>
</tr>
<tr>
<td>x</td>
</tr>
<tr>
<td>A(x, t)</td>
</tr>
<tr>
<td>Q(x, t)</td>
</tr>
<tr>
<td>q_l(x, t)</td>
</tr>
<tr>
<td>k</td>
</tr>
<tr>
<td>g</td>
</tr>
<tr>
<td>h(x, t)</td>
</tr>
<tr>
<td>S_f(x, t)</td>
</tr>
<tr>
<td>V(x, t)</td>
</tr>
<tr>
<td>R(x, t)</td>
</tr>
<tr>
<td>n</td>
</tr>
</tbody>
</table>

This is a set of partial differential equations that reflects the flow below a pressure surface in a fluid. The first equation represents the continuity equation, that reflects conservation of mass and the second represents the equality between rate of change of momentum and net force applied to the system.
1.2.2 Gate structures

The flow of water through the gates \(Q\), will establish how the shallow water equations in the following pools will be changed along time. This flow \(Q\) is a general function of the opening of the gate and the level of water in the pool that precedes and follows the gate

\[ Q = f(h_{left}, h_{right}, h_{gate}) \]  

(1.3)

There are two types of gate, the first one called orifice gate where the water flows below the gate. Its characteristics are given by Equation

\[ Q = C_{ds}A \sqrt{2g(h_{left} - h_{right})} \]  

(1.4)

in which \(A(h_{gate})\) is the effective area of the orifice that depends on the opening gate level and \(C_{ds}\) is the gate’s discharged coefficient.

The second type, called overshoot gate, is modeled by

\[ Q = BC_d \sqrt{2g(h_{left} - h_{gate})^{3/2}} \]  

(1.5)

in which the water flows over the gate. Coefficient \(B\) is the width of the gate.

The model that was produced works with the high of water readings in each pool and not the discharge flow \(Q\) of water among pools.

These are the main mathematical principles embedded in the non-linear model used to design and test the control algorithms studied.
Chapter 2

System identification

Since the system is predominantly non-linear, it was decided to identify its dynamics using identification methods. To execute such task, the Matlab System Identification Toolbox is used. This toolbox assists in the realization of the mathematical model of the dynamic system based on measured data that were obtained from the nonlinear simulator of the canal. This data is used to adjust some parameters within given model structure until the model outputs coincide with the measured output of the real system.

System identification requires a model structure that represents the mathematical relationship between the inputs and the outputs of the system. The models considered are ARX, ARMAX, OE and BJ. Each of this models have the objective of solving the identification problem using the inputs and outputs known, figuring out:

1. The coefficients of the linear differential equations.
2. How many delayed outputs(poles)/inputs(zeros) to use in the description. (model order)
3. The time delay in the system (units of time that takes between a change in the input and a reaction in the output)

The steps of System Identification used to find the best model in the structure can be summarize in the following sequence:

1. Design a experiment collecting all the input-output data from the process to be identified.
2. Observe the data. Selecting only the sections of the data that is considered to have more impact on the model. Remove trends and filter of the data so that only the important frequencies are used.
3. Select and define a model structure within which a model is to be found.
4. Compute the best model in the model structure according to input-output data and a criterion of fit.

To identify the canal we start by identifying only the dynamics of one pool. To do so, in a first stage, we force all the inputs, that represent the opening of the following gates, to a constant regime obtaining, after the settling time, a static level of water in each pool, Figure 2.1. The resulting stationary water levels are in Table 2.1. The gate position of the chosen pool will then be changed periodically with a square and triangular wave shapes, simulating all the possible changes that can happen in the real system.

The pool chosen to start the identification was the second. The same procedure was then applied to all the others pools. After driving gate 2 position with a square wave, this pool reacts as shown in Figure 2.2.
The response of the system to this inputs is analysed in an effort to extract all information that can be useful to the identification process. It can be observed from Figure 2.1 that the settling time from the equilibrium points in the simulator is about 4000 seconds (\sim 1\text{hour}). The data that will be considered to perform the identification is obtained by making variations around this point, where the transient has vanished.

Figure 2.3b shows the level output of the second pool in more detail (the one that is being identified) in response to a sequence of gate opening in the form of square wave. Can be seen, there is a clear main frequency leaded by the input. All the remain frequencies are created by the non-linearities of the system and are to be eliminated to improve the identification process.

To remove this high frequencies, a second-order low-pass butterworth filter is used, with normalized cut-off frequency (\(W_n = 0.015\)) obtained by trail and error.

This particular filter was selected because it is characterized by a magnitude response that is maximally flat in the passband and monotonic overall, that means it have no undesire ripples in the passband that could miss guide the identification. The result of filtering the signal with the butterworth can be seen in Figure 2.4a.

It is also necessary to remove the mean, offsets or linear trends from the sampled time-domain input-output data signals. This data processing operation helps to estimate a more accurate linear model, because linear models cannot capture arbitrary differences between the input and output signal levels. Figure 2.4b shows both input/output signals after de-trending them.
After the data processing is completed, we proceed with the identification of the system. As referred before, the model structures that will be considered are:

**ARX** (autoregressive with exogenous terms) All the parameters of the structure are obtained using least squares. The estimation of the ARX model is the most efficient of linear model estimation methods because it is the result of solving linear regression equations in analytic form.

\[
y(t) = \frac{B(q)}{A(q)} u(t - n_k) + \frac{1}{A(q)} e(t) \tag{2.1}
\]

**ARMAX** (autoregressive-moving average with exogenous terms) Unlike the ARX model, the ARMAX model structure includes disturbance dynamics. ARMAX models are useful when having a dominating disturbances that enter early in the process, such as at the input.

\[
y(t) = \frac{B(q)}{A(q)} u(t - n_k) + \frac{C(q)}{A(q)} e(t) \tag{2.2}
\]
The Output-Error model structure describes the system dynamics separately. No parameters are used for modeling the disturbance characteristics.

\[ g(t) = \frac{B(q)}{F(q)} u(t - n_k) + e(t) \]  
(2.3)

The Box-Jenkins structure provides a complete model with disturbance properties modeled separately from system dynamics.

\[ g(t) = \frac{B(q)}{F(q)} u(t - n_k) + \frac{C(q)}{D(q)} e(t) \]  
(2.4)

For each one of this models the objective is to find an order and parameters that best represent the actual system dynamics. Because there is no simple way to find out which is the best structure so a script was construct for this purpose. This script receives the maximum desired number of poles(order of the system), the filter properties (order and cutoff frequency) and the data to analyze, returning the best model and configuration (zeros, poles and parameters calculated) that represent the best fitting between the data from the model and the real one acquired from the canal. The Figure 2.4 represents the fitting of the best models considering an input of different gains(different command of highs change for the gate).
From the results we can conclude that the identification is better using a lower variation in the gate position, and that the OE (Eq 2.3) is the manipulated model that bests fit the real system. It is now necessary to validate the model using another fragment of the signal created to see how the model follows the real dynamic of the system. It can be seen in the Figures 2.5, 2.6a, 2.6b that the performance of the system is suitable for use in the controller.

The best model obtained has 6 poles and 4 zeros, having the OE structure (Eq 2.5).

$$y(t) = \frac{B(q)}{F(q)} u(t - n_k) + e(t)$$  \hspace{1cm} (2.5)$$

and using a sampling time of 1 second, the following parameters are obtained

$$\begin{cases}
    B(q) = 0.0004657q^{-1} + 0.001519q^{-2} - 0.00164q^{-3} + 0.0005867q^{-4} \\
    F(q) = 1 - 4.213q^{-1} + 6.369q^{-2} - 3.327q^{-3} - 1.097q^{-4} + 1.767q^{-5} - 0.4983q^{-6}
\end{cases} \hspace{1cm} (2.6)$$

The realization of the state space 2.7 have 6 states represented as follow

<table>
<thead>
<tr>
<th>Opening of Gate by (mm)</th>
<th>0.01</th>
<th>0.04</th>
<th>0.05</th>
<th>0.07</th>
<th>0.1</th>
<th>0.2</th>
<th>0.3</th>
<th>0.4</th>
<th>0.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Best Model</td>
<td>OE</td>
<td>OE</td>
<td>OE</td>
<td>OE</td>
<td>OE</td>
<td>BJ</td>
<td>BJ</td>
<td>OE</td>
<td>OE</td>
</tr>
<tr>
<td>Fitting (%)</td>
<td>97.5</td>
<td>95.5</td>
<td>94.37</td>
<td>92.6</td>
<td>90.62</td>
<td>86</td>
<td>82.4</td>
<td>80.3</td>
<td>77.6</td>
</tr>
</tbody>
</table>
\[ x(k+1) = Ax(k) + Bu(k) \]
\[ y(k) = Cx(k) \]  \hspace{1cm} (2.7)

with \( x = [x_1, x_2, \ldots, x_6] \) corresponding to the 6 states of the system

\[
A = \begin{pmatrix}
4,213 & 1 & 0 & 0 & 0 & 0 \\
-6,368 & 0 & 1 & 0 & 0 & 0 \\
3,326 & 0 & 0 & 1 & 0 & 0 \\
1,096 & 0 & 0 & 0 & 1 & 0 \\
-1,766 & 0 & 0 & 0 & 0 & 1 \\
0,498 & 0 & 0 & 0 & 0 & 0 \\
\end{pmatrix}
\]
\[
B = \begin{pmatrix}
0,000465 \\
-0,001518 \\
0,001639 \\
-0,0005867 \\
0 \\
0 \\
\end{pmatrix}
\]  \hspace{1cm} (2.8)

\[
C = \begin{pmatrix}
1 & 0 & 0 & 0 & 0 & 0 \\
\end{pmatrix}
\]

this will be the final model that will represent the second pool, and which will be used in the predictive control algorithm. Each of the remaining pools are identified in the same way, obtaining the four section model. These steps allow to obtain the first controller to the canal using MPC theory (which will be explained in the next chapter).

The final configuration of the simulator with the controller can be seen in appendix A.1
Chapter 3

MPC

Model predictive control (MPC) is a well known family of control algorithms with high performance and wide practical use. The control input is obtained by solving a discrete-time optimal control problem over a given horizon, producing an optimal open-loop control input sequence. MPC is a technology that has the main goal of manipulating a system by combining prediction and control strategy. The prediction is formulated from the linear model that approximates the actual system, then the control strategy compares the model generated states to a set of objectives, adjusting the available actuators to achieve the objectives while respecting the plant constraints (physical limits, boundaries of safe operation..)

All MPC systems rely on the idea of generating values for process inputs as solutions of an on-line optimization problem. The data acquired from measurements provided the feedback and feedforward (if necessary) elements in the MPC structure.

3.1 MPC

The structure of MPC is designed to calculate the next set of action in two steps, estimation and optimization and these are repeated in a recursive way.

In the Estimation, the current state is calculated so that the controller knows in a reliable way the actual condition/states of the systems. Only in this way it is possible to make the correct moves in order to reach a goal. To accomplish this, the controller uses all past and current measurements, and a linear model of the system to control.

In the Optimization step values of setpoint, measured disturbance, and constraints are specified over a finite horizon of future samplings instants P (prediction horizon). The controller then computes M actions to perform in the actuators (where the number of actions is called control horizon). From this M actions only the first control action in the sequence is applied. At the next sampling instant, a new optimal control problem is formulated and solved based on the new information obtained by the measurements. This horizons are given such that the MPC algorithm can be able to see and avoid or minimize potential constrains in the future, acting in the present moment. Has well if the plant has a pure time delay the controller automatically take such effect in consideration, only needing that the time delay is smaller than de control horizon.

3.2 Analytical approach to MPC

Systems using this type of controller use a mathematical model of the plant. MPC algorithm is then able to predict de future movements and react concordantly the future estimations. The model is
presented in state space format:

\[
\begin{aligned}
x_m(k + 1) &= A_m x_m(k) + B_m u(k) \\
y(k) &= C_m x_m(k)
\end{aligned}
\] (3.1)

where \( x_m(k) \) is the state of the system that mathematically shows the actual status of the plant. \( A_m \) is the dynamic matrix that determines how the system will naturally evolve in time from a particular state to another. \( B_m \) reflects the state variation of this system for a particular input \( u(k) \). The output of the system \( y(k) \) is found using the matrix \( C_m \) and the actual state.

Because the identification process of the plant is incremental, it is necessary to convert the state space into the same form.

To obtaining an incremental form, \( x_m(k) \) is substracted from both sides of the equation:

\[
x_m(k + 1) - x_m(k) = A_m( x_m(k) - x_m(k - 1)) + B_m(u(k) - u(k - 1))
\] (3.2)

with

\[
\begin{aligned}
\Delta x_m(k) &= x_m(k + 1) - x_m(k) \\
\Delta u(k) &= u(k + 1) - u(k)
\end{aligned}
\] (3.3)

Now the output of the system is combined with the variation of the state. Its obtained the new state \( x(k) \) with the following structure (Equation 3.4)

\[
x(k) = [\Delta x_m(k) y(k)]^T
\] (3.4)

Using the difference in the system output leads to (Equation 3.5)

\[
y(k + 1) - y(k) = C_m ( x_m(k + 1) - x_m(k)) = C_m A_m (k + 1) \Delta x_m(k) + C_m B_m (k + 1) \Delta u_m(k)
\] (3.5)

Combining now equations 3.4 and 3.5, the new state space form is obtained:

\[
\begin{bmatrix}
\Delta x_m(k + 1) \\
y(k + 1)
\end{bmatrix} =
\begin{bmatrix}
A_m \\
C_m A_m
\end{bmatrix}
\begin{bmatrix}
\Delta x_m(k) \\
y(k)
\end{bmatrix} + 
\begin{bmatrix}
B_m(k) \\
C_m B_m
\end{bmatrix} \Delta u(k) = 
\begin{bmatrix}
0_m & 1
\end{bmatrix}
\begin{bmatrix}
\Delta x_m(k) \\
y(k)
\end{bmatrix}
\] (3.6)

where:

\[
A = 
\begin{bmatrix}
A_m \\
C_m A_m
\end{bmatrix}
\begin{bmatrix}
OT_x \\
1
\end{bmatrix}
B = 
\begin{bmatrix}
B_m(k) \\
C_m B_m
\end{bmatrix}
C = 
\begin{bmatrix}
0_m & 1
\end{bmatrix}
\] (3.7)
where $O_m = [00, 0]$ with dimension $n$. And the triplet $A, B, C$ is called the augmented model used in the structure of predictive control.

The construction of the future control trajectory can be presented as:

$$\Delta u(k_i), \Delta u(k_i + 1), \ldots, \Delta u(k_i + N_c - 1) \quad (3.8)$$

and the future state variables can be formulated as

$$x(k_i + 1|k_i), x(k_i + 2|k_i), \ldots, x(k_i + m|k_i), \ldots, x(k_i + 1 + N_p|k_i) \quad (3.9)$$

where $x(k_i + m|k_i)$ is the predicted state variable at $k_i + m$ with given current plant information $x(k_i)$.

Based on the state space model $(A, B, C)$, the future state variables are calculated sequentially using the set of future control parameters:

$$x(k_i + 1) = Ax(k_i) + B\Delta u(k_i)$$
$$x(k_i + 2) = Ax(k_i + 1|k_i) + B\Delta u(k_i + 1)$$
$$= A^2x(k_i) + AB\Delta u(k_i) + B\Delta u(k_i + 1)$$
$$\vdots$$
$$x(k_i + N_p|k_i) = A^{N_p}x(k_i) + A^{N_p-1}B\Delta u(k_i) +$$
$$+ A^{N_p-2}B\Delta u(k_i + 1) + \ldots + A^{N_p-N_c}B\Delta u(k_i + N_c - 1) \quad (3.10)$$

From the predicted states variables, the predicted output variable are by substitution:

$$y(k_i + 1) = CAx(k_i) + CB\Delta u(k_i)$$
$$y(k_i + 2) = CA^2x(k_i) + CAB\Delta u(k_i) + CB\Delta u(k_i + 1)$$
$$= CA^3x(k_i) + CA_2B\Delta u(k_i) + CAB\Delta u(k_i + 1) + CB\Delta u(k_i + 2)$$
$$\vdots$$
$$y(k_i + N_p|k_i) = CA^{N_p}x(k_i) + CA^{N_p-1}B\Delta u(k_i) +$$
$$+ CA^{N_p-2}B\Delta u(k_i + 1) + \ldots + CA^{N_p-N_c}B\Delta u(k_i + N_c - 1) \quad (3.11)$$

that in a compact format can be expressed as the vectors $Y$ and $\Delta U$

$$Y = [y(k_i + 1|k_i)y(k_i + 2|k_i)\ldots y(k_i + m|k_i)\ldots y(k_i + 1 + N_p|k_i)]$$
$$\Delta U = [\Delta u(k_i)\Delta u(k_i + 1)\ldots \Delta u(k_i + N_c - 1)] \quad (3.12)$$

The final format of the equation is now obtained using the equations $(3.10)$ and $(3.11)$:

$$Y = Fx(k_i) + \Phi \Delta U \quad (3.13)$$

where $F$ and $\Phi$ is given by:

$$F = \begin{bmatrix} CA \\ CA^2 \\ CA^3 \\ \vdots \\ CA^{N_p} \end{bmatrix} \Phi = \begin{bmatrix} CB & 0 & 0 & \ldots & 0 \\ CAB & CB & 0 & \ldots & 0 \\ CA^2B & CAB & CB & \ldots & 0 \\ \vdots \\ CA^{N_p-1}B & CA^{N_p-2}B & CA^{N_p-3}B & \ldots & CA^{N_p-N_c}B \end{bmatrix} \quad (3.14)$$

Having the basic structures defined, the next step is the optimization of the quadratic cost function

$$J = (R_s - Y)^T(R_s - Y) + \Delta U^T\bar{R}\Delta U \quad (3.15)$$
in the cost function presented can be distinguished two terms. The first term has the objective to minimize the difference between the desired reference at instant \( k_i \) \( (r(k_i)) \) and output of the system. \( R_s \) is a vector with the reference propagated through time, with the length of the prediction horizon.

\[
R_s^T = \begin{bmatrix} 1 & 1 & \ldots & 1 \end{bmatrix} r(k_i) \quad (3.16)
\]

The second term of Equation (3.15) gives us control about the command variations \( (\Delta U) \). This is an important feature because permits having a more robust and smooth MPC controller or a faster one but more unstable. Tuning this feature can be made with \( \bar{R} \) (Equation 3.17).

\[
\bar{R} = r_w I_{N_c \times N_c} \quad (3.17)
\]

where \( r_w \) is the parameter to tune for the desired close loop performance and \( I_{N_c \times N_c} \) is the identity matrix with dimension of the horizon control.

If \( r_w \) is choose to be very small (\( \approx 0 \)) the cost function is interpreted as the situation where we do not want to pay any attention to how large the \( \Delta U \) might be, having the only goal of reducing the error between the reference and output, as much as possible. If \( r_w \) is large then all attention is made on how big the variation of command will be made.

Putting all together and replacing \( Y \) from equation (3.13) into (3.15) the cost function will now be presented as:

\[
J = (R_s - Fx(k_i))^T (R_s - Fx(k_i)) - 2\Delta U^T \Phi^T (R_s - Fx(k_i)) + \Delta U^T (\Phi^T \Phi + \bar{R}) \Delta U \quad (3.18)
\]

To obtain the optimal variation on the Command \( \Delta U \) that will minimize the cost function, the cost function is derived in order to \( \Delta U \):

\[
\frac{\partial J}{\partial \Delta U} = -2\Phi^T (R_s - Fx(k_i)) + 2(\Phi^T \Phi + \bar{R}) \Delta U \quad (3.19)
\]

and then the minimum is obtained by:

\[
\frac{\partial J}{\partial \Delta U} = 0 \quad (3.20)
\]

the optimal solution for the controller have the final format:

\[
\Delta U = (\Phi^T \Phi + \bar{R})^{-1} \Phi^T (R_s - Fx(k_i)) \quad (3.21)
\]

or

\[
\Delta U = (\Phi^T \Phi + \bar{R})^{-1} \Phi^T (\bar{R}_s r(k_i) - Fx(k_i)) \quad (3.22)
\]
3.3 Results with model and MPC

To implement the algorithm, a complete simulator of the canal is used (Appendix A.1). Five experiments have been implemented to have the feeling on how the system behaves. The model used in the algorithm is the one presented in Eq. (2.8) with a prediction horizon interval of 20 seconds and a control horizon of 4 seconds. The weight control $\alpha$ have been set to 0.8, meaning a more robust than a faster control.

The first four experiments have been implemented to analyze how a control interaction in one pool influences the remaining pools. The last experience deals with more than one pool been actuated.

The first experience, Figure 3.2, a change in the reference of the first pool is performed after 1000 seconds. It can be observed that all the pools are affected by the change of reference but are also compensated by the controller implemented in each one of them. In this experiment the reference is achieved in about 120 seconds and have a overshoot of 2mm, performance that it is considered good for this type of slow systems. The settling time augments in the following pools because the controller in this stage can’t see what the neighbor pools are doing. A control of water level is made with the information acquired in the present moment. So the fourth pool tries to compensate the variations in the third pool and the third compensate the variation in the second and so on.

In the following experiences (Figures 3.3, 3.4 and 3.5) we can observe as expected that the following pools are more sensible to variation upstream than downstream. In Figure 3.3 it is visible that the first pool is almost insensible to the variations in the remaining pools. In Figure 3.5 the first and second pool barely apart from their previous state. Only the immediate neighbors have to compensate in a more aggressive way. This is way it is necessary a distributed algorithm. The distributed algorithm that will be presented in the next report are the GPC and SIORHC. In this algorithms the current state of the neighbor pools are considerate and the action upon the canal is distributed concordantly to that knowledge.

![Figure 3.2: B: Reference Level, R: Water Level, G: Gate Level](image)

The performance of the controllers from all the four pools working together can be seen in Appendix
A.2 Starting from a stable state the reference of the last pool is changed to 5mm below the actual value of water level and the second pool augments 1mm (at 2000 seconds). The controllers react by opening the overshoot gate from the last pool and slightly close the gate located between the second and third. This makes the excess of water fall into the parallel canal, lowering down the level of water in the fourth section and accumulates more water in the second pool because of the lowering gate between the second and third pool. Producing in turn, lesser water feeding the third pool. Doing so it will reduce the water level from the third pool deviating from his reference. The controller from the third pool will then react to this disturbance lowering down the gate connecting the two pools, increasing the water volume until it reach again the reference. A similar process can be observed in the rest of the experience. Another important physical feature is the several pool influence have upon each other. The first pool is almost independently from all the other will the last one is almost dependent of any reference deviation from upstream pools, so the weights on control action should be take in account separately.
Figure 3.4: B: Reference Level, R: Water Level, G: Gate Level

Figure 3.5: B: Reference Level, R: Water Level, G: Gate Level
Chapter 4

Conclusions

Identification of linear models of the canal for specific points of operation have been made. This models are used in the design of model predictive control. First, SISO models relating input/output data have been developed. Then, these models have been used to generate state-space models which are used in the predictive control algorithm.

The application of MPC algorithm presents a good approach for dealing with this type of system. Using a set of controller for each pool a stable system can be archived with a low online processing effort. All the main calculations are made offline and are used in the system in a matricial form instead of the classic online recursive procedure. The neighbor pools have a delayed reaction to reference changes in the central one, because it is not possible to the controller predict what is going to happen to their neighbors in his own pool. The compensation given by the controllers have a fast response but can be improved with the distributed algorithm.

The next step of this work is focused exclusively on these algorithms. The GPC and SIORHC algorithms will be approached in a decentralized way. These systems can be decomposed in subsystems and connected sequential, i.e. in which each subsystem is affected only by two neighbors. The results will be compared with the ones pre
Appendix A

Figures
Figure A.1: MPC controllers and simulator
Figure A.2: Control of all Sections (G: Gate Level, B: Reference, R: Water Level)