Treball de Fi de Màster

Master's degree in Automatic Control and Robotics

Reliable Model Predictive Control of Drinking Water Networks

MEMÒRIA

Autor:	Iván Martinez Hurtado
Director:	Vicenç Puig Cayuela
Director:	Carlos Ocampo-Martinez
Convocatòria:	Setembre 2016

Escola Tècnica Superior d'Enginyeria Industrial de Barcelona

To B. Fabiola

Abstract

This project proposes a reliable Model Predictive Control (MPC) applied to drinking water networks (DWNs). The system reliability is calculated by structural analysis of the whole system taking into in account the network topology. In this form it is possible to describe the system reliability as a function of actuators reliability. Therefore by using CSP the system reliability can be controlled by imposing some new control actions bounds. This approach maintain the convexity of optimization problem avoiding problems with local-minima.

The methodology, tested by simulation in the Barcelona DWN, shows that it is possible to control the evolution of the reliability level of the system by tuning only one parameter and also to describe a trade-off between reliability and operational costs.

Other approaches that include reliability inside the controller that are based on the actuators reliability does not consider the connections between the actuators and therefore new control bounds are established without taking into in account possible parallel paths. With this new approach presented in this project the results shown how through the redundancy of optimal solutions it is possible to increase reliability levels without economic cost.

Model predictive control, reliability control, structural analysis, drinking water networks, constraint satisfaction problem

ETSEIB

Acknowledgement

This project would have been impossible for me without the tireless help of my advisors Vicenç and Carlos. A few months ago I went to Vicenç to ask for a challenging research project. Since that request I always felt integrated and respected and the assistance received has been constant. I am also very grateful to all the colleagues of the SAC and the IRI that never hesitate to help when I need it.

I would also like to thank all the colleagues who have helped me improve in every aspect of my life. Those who have made my time in the university more enjoyable, whether teaching me or having coffee with me.

Por supuesto no puedo olvidarme de mi familia, a la cual escribiré en castellano. Ellos siempre han sido un ejemplo de esfuerzo y trabajo. Sin su apoyo incondicional y sin su confianza en mi, demostradas durante todo este largo camino, seria muy probable que nunca hubiera llegado hasta aquí. Para terminar tengo que darle las gracias a Fabiola por su paciencia, fuerza y amor infinitos y por ayudarme, sobretodo, en los peores momentos.

Iván Martínez Hurtado

Barcelona, June 2016

Notation

General Operators and relations

$\vert$	such that
$\in$	is element of
$\forall$	for all
$\leftarrow$	assign right-hand side to the left-hand side
$\rightarrow$	mapping

General Sets and Spaces

$\mathbb{R}$	set of real numbers
$\mathbb{R}_+$	set of non-negative real numbers, defined as $\mathbb{R}_+ \triangleq \mathbb{R} \setminus (-\infty, 0]$
$\mathbb{R}^n$	space of n-dimensional (column) vectors with real entries
$\mathbb{R}^{n \times m}$	space of n by m dimension matrix with real entries
$\mathbb{Z}$	set of integer numbers
$\mathbb{Z}_+$	set of non-negative integer numbers
$\subset$	subset

General Sets and Spaces

$\min$	minimum
$\max$	maximum

M^T	Transpose of a matrix or vector

Systems and Control Theory and Reliability analysis

n_x	number of states, $n_x \in \mathbb{Z}_+$
n_u	number of inputs, $n_x \in \mathbb{Z}_+$
n_d	number of disturbances, $n_x \in \mathbb{Z}_+$
x	state vector, $x \in \mathbb{R}^{n_x}$
u	input vector, $u \in \mathbb{R}^{n_u}$
d	disturbance vector, $d \in \mathbb{R}^{n_d}$
$\mathbb{X}$	set of admissible states, $\mathbb{X} \subset \mathbb{R}^{n_x}$
$\mathbb{U}$	set of admissible inputs, $\mathbb{U} \subset \mathbb{R}^{n_u}$
$\underline{u}$	minimum admissible input vector, $\underline{u} \in \mathbb{R}^{n_u}$
$\overline{u}$	maximum admissible input vector, $\overline{u} \in \mathbb{R}^{n_u}$
$\underline{u}_{CSP}$	minimum admissible input vector after the CSP, $\underline{u} \in \mathbb{R}^{n_u}$
$\overline{u}_{CSP}$	maximum admissible input vector after the CSP, $\overline{u} \in \mathbb{R}^{n_u}$
R_t	total system reliability
R_d	demand reliability, $R_d \in \mathbb{R}^{n_d}$
$R_{d_{path}}$	Path reliability, $R_{d_{path}} \in \mathbb{R}^{n_{path}} \;\; \forall \;\; d = [1, ..., nd]$
R_a	Actuator reliability, $R_a \in \mathbb{R}^{n_u}$

Contents

ETSEIB

List of Figures

List of Tables

Chapter 1

Introduction

Drinking water is provided by means of a drinking water network (DWN) to consumers and industry. DWNs are large-scale systems that can be structurally organised into several layers [5]

- Supply layer, composed of water sources, large reservoirs and also natural aquifers.
- Transportation layer, linking water treatment and desalination plants with reservoirs distributed all over the city.
- Distribution layer, used for meeting consumer demands and linking reservoirs with consumers.

This project is focused in the transportation layer also called Drinking Water Transport Network (DWTN). This networks are large-scale, expensive and strategical systems that need a sophisticate control strategies in order to achieve performance and robustness. This project presents a Reliable Model Predictive Controller based on topology reliability with the purpose to incorporate in a further step a fault tolerant control.

1.1 Motivation

Water is essential for life. For this reason is considered a human right and a strategical resource. Water is a renewable resource, however the overpopulation in the large urban areas make the water supply something critical. Therefore it is essential to work on improving the supply, treatment and transport of water in other to have always good quality water and saving operational cost.

For this reason it is necessary to improve the actual control systems and adapt them in order to work with a DWTN. At this moment exist lot of literature talking about DWTN. Specifically for Barcelona DWTN exist literature that implement MPC to Barcelona DWTN endowing inside the MPC reliability analysis and fault tolerant control. Since this project the system reliability has been controlled actuator by actuator without taking into in account the topology of the network.

Therefore the motivation of this project is to include inside the controller the reliability of the whole system taking into in account the topology of the network. In this form is possible establish a criteria that regulate not only the actuators reliability but the total system reliability. This reliability is understood as the probability to reach all of demands in a period of time that is interesting in order to guarantee the continuous water demand flow.

1.2 Project Objectives

This project is based on extended MPC for large-scale networked systems subject to constraints and to persistent stochastic disturbances. Particularly, the management of DWNs within a multiobjective optimisation framework will be studied, considering water demands as system disturbances.

The objectives of the project are:

> [1.]To establish a criteria and equations to determine the reliability of the whole system based on the reliability of the actuators and taking into in account the network topology. Defining reliability as the probability to reach all water demands. To design a method to control the loss of DWTN reliability in a certain time. To include the reliability inside the MPC controller based on the control actions contraction. To implement the MPC with Reliability constraints on the Barcelona DWTN as a real case of study and to extract relevant results.

1.3 Project structure

The dissertation is organized as follows:

Chapter: 2 Background

This chapter introduces some relevant information about MPC and the application of MPC to a drink water transport networks. Moreover has been presented a discussion about the state of the art of reliability, more specifically about how to describe a reliability of drinking water transport networks from different approaches.

Chapter: 3 Model Predictive Control with Reliability constraints

This chapter presents mathematical preliminaries related with the inclusion of the reliability constraint inside the MPC. During the chapter have been explored different solutions and finally has been explained the selected one. As example of the selected solution has been included a test example with a simple drinking water transport network describing how the solution works and analyzing the implementation results.

Chapter: 4 Implementation in the Barcelona DWTN

In this chapter a description of the case of study has been provided. Moreover have been analysed the results obtained from the Barcelona drink water transport network as a case of study after the application of the proposed solution.

Chapter: 5 Environmental Impact

This chapter analyses the environmental impact of the project taking into in account different possible consequences if the solution is finally implemented in a drinking water transport network.

Chapter: 6 Budget

In this chapter it is possible to see a budget breakdown of this project.

Chapter: 7 Concluding Remarks

This chapter summarises the contributions made in this project and discusses the main concluding ideas and ways for future research.

Chapter 2

Background

This chapter presents the basic ideas and a literature review of the different topics related with the project.

2.1 Fundamentals of Model Predictive Control

2.1.1 General Consideration

Model Predictive Control (MPC) is one of the most successful control technologies due to the capability to incorporate constraints and define a multiobjective cost function. The use of a cost function allows to describe the desired system behaviour and to specify preferences in a multi-objective optimal control problem. It is possible to specify whatever cost function but is highly recommended in the literature to use convex cost functions, like linear or quadratic ones.

The most important part inside an MPC is the dynamic model of the system. This model is similar to one constraint of the optimisation problem but the dynamical model can be linear or non-linear, deterministic or stochastic, discrete, continuous or hybrid. More details about theory and dissing of MPC can be found in [1] and [8].

Since a DWTN is a large-scale system it is highly recommended to guarantee the optimization problem convexity. In this form a fast solvers can be applied and problems with local minima can be avoided. For this reason the dynamics of the system are simplified as a control oriented linear ones and the cost functions are in a linear and quadratic form.

2.1.2 Problem Statement

A generic MPC uses information about the present and the past to predict the system evolution over a given period of time (prediction horizon) and solve an open-loop optimization problem to compute a sequence of control actions that achieve maximum performance satisfying the problem constraints. After the prediction only the first control action is applied. Therefore the procedure is repeated in each step.

The optimization problem can be described as:

Cost Function,

$$J^* = \min_{u_k,\dots u_{k+Hp-1}} \sum_{i=0}^{Hp-1} l_i(u_{k+i}, x_{k+i}) \tag{2.1}$$

Subject To:

Dynamics,

$$x_{k+i+1} = f(x_{k+i}, u_{k+i}) + d_k \tag{2.2}$$

And physical constraints,

$$x_{k+i} \in \mathbb{X} \tag{2.3}$$

$$u_{k+i} \in \mathbb{U} \tag{2.4}$$

Hp is the prediction horizon, x_k is the system state vector, u_k is the control input vector, l_i is the stage cost, f_i is the state system evolution, d_k is the disturbances vector and $\mathbb{X}, \mathbb{U}$ are the physical bounds. Since the $Hp \neq \infty$ the optimization problem is a finite horizon open-loop optimization problem.

2.2 Model Predictive Control of Drinking Water Transport Networks

A DWTN can be described as a large-scale flow system which is composed of different hydraulic elements [2] like: valves, pumps, tanks, pipes, etc. With the purpose of transport water from the sources to the demands, satisfying pressure and potability levels. From the side of control is possible to separate the elements into two classes, active elements like pumps or valves or passive elements like nodes or tanks. In this form is possible to construct a model by associating a control action for each active element and a state for each passive elements. On the other hand DWTN requires a instrumentation in order to obtain sufficient information about the system state and a telemetry system in order to do a tele-control of the system, which is indispensable due to the scale of a DWTN.

DWTN are redundant systems with a lot of states and actuators, moreover the demands of water can be predicted in a stochastic form. Therefore an optimal management of this systems becomes a hard task. Moreover DWTN are expensive and strategical systems and the operational cost are also important. For this reason in the last years important projects around DWTN have been strongly supported.

2.2.1 Control Model

DWTN can be synthesized as a interconnection of n_r sources, n_x tanks, n_u actuators (pumps and valves), n_d demands and n_q intersection nodes.

Then the system can be described with a control-oriented lineal discrete-time model for all time instant k:

$$x_{k+1} = Ax_k + Bu_k + D_d d_k \tag{2.5}$$

$$0 = E_u u_k + E_d d_k \tag{2.6}$$

$x_k \in \mathbb{R}^{n_x}$ are the tanks water volume, $u_k \in \mathbb{R}^{u_x}$ are the control inputs and $d_k \in \mathbb{R}^{n_d}$ are the demands. Eq. (2.5) describes the dynamics of the tanks volume and (2.6) is the statics equation that describes the mass balance into the nodes of the system. Therefore $A \in \mathbb{R}^{n_x \times n_x}, B \in \mathbb{R}^{n_x \times n_u}, D_d \in \mathbb{R}^{n_x \times n_d}, E_u \in \mathbb{R}^{n_q \times n_u}$ and $E_d \in \mathbb{R}^{n_q \times n_d}$ are time-invariant matrices that describe the relation between states, actuators and disturbances.

On the other hand system are subject to physical constraints. One is the minimum and maximum control inputs:

$$u \in \mathbb{R}^{n_u} | \, u_{min} \leq u \leq u_{max} \tag{2.7}$$

The other is the minimum and maximum tanks water volume:

$$x \in \mathbb{R}^{n_x} | \, x_{min} \leq x \leq x_{max} \tag{2.8}$$

$x_{min} \in \mathbb{R}^{n_x}$ and $x_{max} \in \mathbb{R}^{n_x}$ denote the minimum and maximum tank volumes in

The management criteria is based in a multi-objective cost function. This cost function has been provided by AGBAR and therefore take into in account their Barcelona DWTN knowledge [3]. It is possible to appreciate 3 different objectives that are the economic, the safety and the smoothness.

Economic: To minimize the economic cost associated to the transport of the water. The economic cost is due to two main factors, the cost of the electricity associated to pumping water and the cost of the water that can vary depending on the source.

$$J_{E,k} = \|(\alpha_1 + \alpha_{2,k})^T u_k\|_1 W_e \tag{2.9}$$

$J_{E,k} \in \mathbb{R}^1$ is the economic cost, $\alpha_1 \in \mathbb{R}^{n_u}$ are the water production cost and $\alpha_{2,k} \in \mathbb{R}^{n_u}$ are the pumping cost, that depends on k because the electricity tariff change depending the hour of the day.

Safety: To guarantee the water demands satisfaction taking into in account that the futures demands are estimations and therefore are not a deterministic values. For this reason exist a security level for each tank and a cost associated to not reach it.

$$J_{S,k} = \|\epsilon_k\|_2^2 W_s \tag{2.10}$$

$J_{S,k} \in \mathbb{R}^1$ is the security cost associated to the $\epsilon_k \in \mathbb{R}^{n_x}$ that are the infraction of the security level in terms of volume of water.

Smoothness: To operate with smoothness since the actuators are heavy.

$$J_{\Delta U,k} = \|\Delta u_k\|_2^2 W_u \tag{2.11}$$

$J_{\Delta U,k} \in \mathbb{R}^1$ is the penalization to control signal variations $\Delta u_k = u_k - u_{k-1} \in \mathbb{R}^{n_u}$

W_e, W_s and W_u are diagonal matrices that weight each decision variable.

The resulting objective function is a simply sum of the 3 costs (2.9), (2.10) and (2.11). Therefore the optimal has been obtained by minimising the total cost for each control action in each step during the prediction horizon.

$$J^* = \min_{u_k,\ldots,u_{k+N-1}} \sum_{i=0}^{N-1} [J_{E,k+i} + J_{S,k+i} + J_{\Delta U,k+i}] \tag{2.12}$$

2.3 Reliability

The reliability of one system (or component) can be defined as the probability of that system succeed during the execution of one task during a period of time. The reliability study can be useful in order to manage the operational risk of a system. On the other hand it is possible to study not only the reliability but also the reliability evolution. In this form it is possible to define how the possible actions affects the system reliability and therefore, penalizing actions that subtract more system reliability.

2.3.1 Reliability in Drinking Water Networks

In a large city like Barcelona all of people expect to have drinking water at home always. For this reason DWTN reliability is very important.

As stated in [6], reliability assessment in DWNs can be classified in two main categories: the hydraulic reliability, refers to transport of desired quantities and qualities of water at required pressures to the appropriate locations at the appropriate times. The topological reliability, refers to the probability that a given network is connected, given its components mechanical reliabilities, i.e. the components probabilities to remain operational at any time.

In the literature it is possible to see studies related with the reliability on DWTN from different point of view. Some are related with the hydraulic reliability of the DWTN, [3], focusing in satisfy demands when the demands prediction are not correct. On the other hand a reliability assessment can be computed by a Bayesian Network [12], [10], [9]. In this form it is possible to include inside the cost function some weighed parameters that involve the reliability of the system.

However this project is based on the Reliability described by [4]. In this form the total reliability of one system can be described by different parallel and serial connection of elemental subsystems. It means that if possible to describe the whole DWTN as a serial and parallel connection of actuators. It is used by [7] for the control of a DWTN describing the whole network as a serial and parallel connection of different elementary subsystems that are, in this case, the hydraulic actuators.

During the last years is prove that using MPC in DWTN have problems with respect to the network reliability. It is because the MPC is based on an optimization problem, so it is normal that the optimal solution in order to satisfy the demands is always similar or the same. On one side the optimal solution is based on pump water during the night when the electricity are cheaper, but more reliability is preserved when the total water is pumped with a constant flow. On the other side if one path is cheaper in order to satisfy one demand the optimal control always transport the water for that path. This fact reduce a lot the reliability of only few actuators instead of reduce a bit the all actuators reliability and in some cases the fact reduce a lot the total system reliability.

For this reason it is necessary to include the system reliability inside the controller in order to minimize the degradation of the system but preserving a low operational cost.

Chapter 3

Model Predictive Control with Reliability Constraints

This chapter presents the modelling principles and common management criteria of DWTNs and introduces the application of predictive control on these large-scale systems. Moreover has been introduced the idea of DWTN reliability in terms on probability that exits at least one path in order to supply all the demands of the system in a specific time by using a reachability analysis.

3.1 Formulation of MPC with Reliability Constraints

In this project reliability is understood as the probability that the DWTN are able to reach all water demands in a period of time. Therefore this reliability depends on the reliability of the actuators.

The first of all is to describe the reliability of the whole system R_g in terms on the reliability of the actuators. As has been said before in this project the global reliability can be described by decomposing the whole system into subsystems [4]. First of all the system has been decomposed into a serial subsystems, one for each demand. Then each demand has been decomposed into a parallel subsystems that correspond to all of possible paths that are able to satisfy the demand. Finally each path has been decomposed into a serial connection of actuators.

In order to merge different parallel subsystem n_p and different serial subsystems n_s have been used

(3.1) and (3.2).

$$R_p = 1 - \prod_{n_p}^{i=1}(1 - R_i) \tag{3.1}$$

$$R_s = \prod_{n_s}^{i=1}(R_i) \tag{3.2}$$

As has been said before it is possible to describe the reliability of a DWTN R_g as a union of serial subsystems that represents each one of the reliabilities demands of the network R_d (3.3). R_d are the probabilities to reach each demands separately.

$$R_g(k) = \prod_{n_d}^{i=1}(R_{d_i}(k)) \tag{3.3}$$

Each demands nodes have a several number of path that are able to to satisfy the water demand. Each one of this paths can be describe as a parallel subsystems (3.4).

$$R_d(k) = 1 - \prod_{n_{d_{path}}}^{i=1}(1 - R_{path_i}(k)) \tag{3.4}$$

Actually each path is a serial union of actuators, so the reliability of one path can be describe as the serial of actuators reliabilities (3.5).

$$R_{path}(k) = \prod_{n_{path_a}}^{i=1}(R_{a_i}(k)) \tag{3.5}$$

Finally the Reliability of each actuator follow a exponential law, eq. 3.6.

$$R_a(k) = R_{a_0} e^{-\lambda_a(k)t} \tag{3.6}$$

t is time and R_{a_0} is the initial reliability of the actuator.

The failure rate of actuators may vary in function of the control action. In this study the failure rate has been divide into two parts [7]. On one side the nominal failure rate λ_a^0, represents the degradation of one actuator without control actions, in other words represents a temporal degradation. On the other hand the control actions also have effect on the degradation. It means that higher control actions degrade the system more. Therefore a trade off between closed-loop performance and reliability appears.

The nominal failure rate change cause of control action following a exponential law (3.7).

$$\lambda_a(k) = \lambda_a^0 e^{(\beta_a u_a(k))} \tag{3.7}$$

Combining eq. 3.6 and eq, 3.7:

$$R_a(k) = R_{a_0} e^{-\lambda_a^0 e^{(\beta_a |u_a|(k))} t} \tag{3.8}$$

At this point it is possible to compute the reliability of the whole network in terms on the probability that exist at least one path in order to satisfy each demand.

Now the problem is how to include the reliability inside the MPC. One possibility is to include the reliability in the objective function of the optimisation problem and weigh the cost of the degradation in order to minimise the degradation but maintaining performance. With this solution due to the system reliability is non-lineal the optimisation problem becomes non-convex. A non-convex optimization problem leads to deal with local minima and long computational times solving the optimization problem and it is more important in large systems like the Barcelona DWTN.

Other possibility is not to include the reliability in the cost function but to add a total reliability constraint directly inside the optimization.

$$\alpha R_{t_0} \geq R_t \tag{3.9}$$

R_t is the final reliability after the prediction horizon, R_{t_0} is the initial reliability and α is the maximum system degradation in terms on initial system reliability R_{t_0}.
However to add this constraint directly to the problem also makes the optimisation non-convex.

3.2 Proposed solution

On the other hand it is possible to convert the reliability bound to a control action bound since the system reliability only depends on control inputs and some constants values.

In this project is proposed to use a CSP to find the new actuators for which is preserved the reliability constraint.

3.2.1 CSP concept

A Constraint Satisfaction Problem (CSP) is defined as the problem of finding a fixed set of variables satisfying a given set of constraint. In CSP the variables $v_1, v_2, ..., v_n$ are defined as a finite sets $D_1, D_2, ..., D_n$. This sets are also called the variables domains.

CSP can be formulate as a 3-tupla $H = V, D, C$ on:

- $V = \{v_1, ..., v_n\}$ is a finite set of variables
- $D = \{D_1, ..., D_n\}$ the domains set of the variables
- $C = \{c_1, ..., c_m\}$ the set of constraint related with the variables V

The solution of a CSP $S(H)$ is a set of solutions $\{Z_1, ..., Z_n\} \subset D$ that satisfy the set of constraints C. One variable is consistent if for $(z_1, ..., z_n) \in S(H)$, [11]:

$$\forall z_i \in Z_i \subset D_i \;\; \exists (z_1 \in D_1, ..., z_n \in D_n) \tag{3.10}$$

3.2.2 Including the CSP inside the MPC

Therefore, the idea is to synthesise the network as a connection of actuators from each source to each demand by using (3.3), (3.4) and (3.5). After that by using (3.6) is possible to describe the whole system reliability in terms on reliability of each actuator. Moreover the actuator reliability only depends on two factors: a temporal factor and a control input factor. So finally it is possible to associate the total system reliability to the control inputs of all actuators.

Now, transferring the CSP theory to the case of study the CSP has been described as follows:

- $V = \{u_1, ..., u_{n_u}, \alpha\}$, the control actions and the reliability degradation
- $D = \{\mathbb{U}_1, ..., \mathbb{U}_{n_u}, [\underline{\alpha}, \overline{\alpha}]\}$, the actuators and reliability bounds
- $C = \{\alpha = f(u)\}$, the function that connects reliability and control actions.

As a result is obtained a set of solutions Z that are the new control inputs bounds.

On the other hand the reliability is computed each step closing the CSP loop, Figure 3.1.

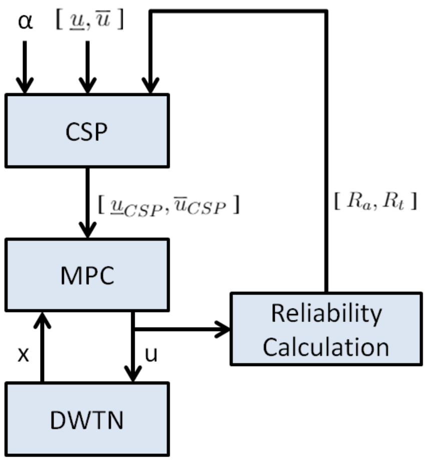

α
[$\underline{u}, \overline{u}$]
CSP
[$\underline{u}_{CSP}, \overline{u}_{CSP}$]
[R_a, R_t]
MPC
Reliability Calculation
x
u
DWTN

Figure 3.1: Methodology Flowchart

Therefore it is possible to contract the control inputs in order to preserve a desired level of reliability. Therefore this method preserve the convexity of the optimization problem.

$$u_{min} \rightarrow CSP \rightarrow u_{min_{csp}} \tag{3.11}$$

$$u_{max} \rightarrow CSP \rightarrow u_{max_{csp}} \tag{3.12}$$

3.3 Algorithmic implementation

In this subsection the general algorithmic implementation and the problematic around the CSP computation has been explained. Algorithm 1.

Algorithm 1 Computation of new actuator bounds

```
1: set($\alpha$)
2: set($R_{a_0}$)
3: $R_{t_0} \leftarrow$ Network Topology($R_{a_0}$)
4: for $k = 1$ to Horizon of Simulation do
5:     $\overline{u_k} \leftarrow$ solve($CSP$)
6:     $u_k \leftarrow$ solve($MPC$)
7:     $R_{a_k} \leftarrow$ Reliability Computation($u_k$)
8:     $R_{t_k} \leftarrow$ Network Topology($R_{a_k}$)
9: end for
```

3.3.1 Reachability analysis

First of all before the CSP it is necessary to compute the reliability equations that describe the system.

As has been said before in 3.1 the total system reliability has been calculated by decomposing the system into a small subsystems. These subsystems are the different paths that exist in order to satisfy each demand from each source. It is possible to extract the equation of the reliability by hand if the system are not too much large. In 3.4 has been explained a test model on it is possible to compute the reliability equation by hand. However it is not the case of the Barcelona DWTN that are a large-scale and complex system. For this reason has been implemented an algorithm in order to identify each path for each demand and then construct automatically the reliability equations.

The first step is to describe the network as a matrix by using (2.5) and (2.6). With the matrix description it is possible to construct a graph of the network. A graph is a description based on the connection of the system elements. In a graph each vertex is connected to others vertexes trough edges. For this system the vertexes are the tanks, the nodes and the sources and the edges are the actuators.

The connection of the states has been establish by using the matrix B and E (2.5). In these matrices the union between the states (tanks and nodes) and the actuators is represented using numerical non-zero values. So, the algorithm goes over all actuators (columns) an look for non-zero values, then by looking the sign of the values it is possible to determine also the direction of the connection. Therefore for each actuator has been established a link between the states taking into in account the direction of the connection. Moreover in these variables ,called in the code links, are added the actuator that connect the states as extra information. In the same way the connection between the sources and the states has been added.

The following step is to create a matrix that describe the relation between the states by using the links computed above. This relation takes into in account the direction of the actuator. Its means that maybe is possible to go from state "i" to state "j" but is not possible to go from state "j" to state "i."

Now it is necessary to determine the initial vertexes that in this case are the vertexes that have direct relation with the disturbances. The relation between states and disturbances has been described by the matrices Bp and Ed (2.6) in 2.2.1.

Starting from each disturbances (initial vertexes) the code analyze the connections that exist with other vertexes and write down a matrix with one column for each possible path. The algorithm takes the vertexes found in the last step and repeat the process growing the matrix. Finally when the vertex analyzed correspond to a source vertex the algorithm break the loop and continue looking for an other branch. Finally all paths end in a source vertex its means that all of possible paths in order to satisfy one demand have been found. Therefore the algorithm is repeated for each demand (disturbance) in order to find all paths of all demands.

Moreover the algorithm takes into in account the possibility that exist one or more loops in the network. A loop is an infinite close path that the connection between the states form. To avoid loops, when the paths are found, the code reject all paths that have more than one time the same vertex. The MatLab code of this algorithm can be found in Appendix B.

This process has been executed one time since the topology of the system is always the same. Therefore it is possible to use the paths matrices extracted from the code in other projects if is it necessary. On the other hand the code is useful in order to determine the paths of other systems with no effort since the code are made for the general case and not specifically for the Barcelona DWTN.

3.3.2 Writing down the input file for IBEX

The solver selected to compute CSP is the IBEX. IBEX is a C++ library and is execute outside MatLab. The C++ program is waiting one flat text file with the inputs of the procedure on the inputs are the constants values, the variables and the constraints. Full code is in Appendix C

Due to the reliability constraint is made for each step and the values of the actuator reliability varies in each step the input file also varies in each step. Therefore becomes essential one code that auto-generate the input file in each step.

The code write the actual reliability of each actuator as a constants values and the physical bounds of the actuators as a variables. Then it write two equations one is the eq. that constraint the maximum total reliability degradation. The second equation is the total reliability in terms on the actuators taking into in account the topology of the system. This equation are write by using the paths matrices obtained above. Therefore each path for each demand has been taken and using the 3.8, step by step, the total equation is written down. To get an idea, for the 17 tanks model of Barcelona DWTN, this equation occupies 100 pages in Arial 10 points.

3.3.3 C++ IBEX program

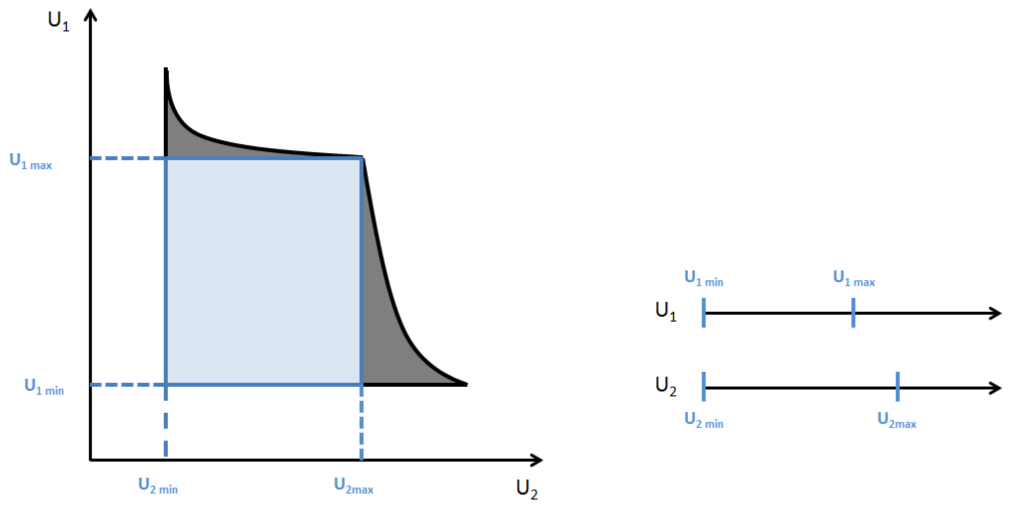
As has been said before the selected solver to compute the CSP is the IBEX library. Before to explore this solution others solutions has been explored like for example Interval Peleer but finally with the others solutions the results thus achieved are not good when the problem become too heavy.

IBEX is a C++ library for constraint processing over real numbers based on interval arithmetic. IBEX library is easy to install in a Linux or iOS and is possible to install in a Windows computer following the documentation in the IBEX web page *"www.ibex-lib.org"*.

IBEX is a ideal library to work with this kind of problems. Moreover include a set of defined functions that simplifies a lot the task of compute a CSP. In this case the C++ code consists in: to read of the input file, to set the solver parameters, to call the solve function, to read and simplify the solutions and to write the solutions in a text file. Code in Appendix D.

The main problem with IBEX is that uses *box* variables and compute a multidimensional solution. It means that the new actuators bounds depend on each other and the possible solution is an hypervolume inside of a polytope. This would be a problem because the purpose is to maintain the optimisation problem as a convex problem and it is not guaranteed in this case. Therefore after the computation of the CSP the solution is simplified taking only the higher minimum and the lower maximum introducing a conservative policy in the method.

In Figure 3.2 has been drawn a representation of this problem for a 2 actuator system. In the left side is represented the solution extract from the IBEX solver that is a two dimensional solution. In this solution the maximum value for the actuator 1 depends on the value of the actuator 2 and vice versa. It has sense since the total reliability depends on the combination of control actions so, if one actuator is degrading the system a little maybe the rest can degrade more the system and even satisfy the reliability constraints.



(a) Multidimensional solution (b) Multiple one-dimensional solution

Figure 3.2: CSP solution simplification

Then the solution is simplified as has been said before obtaining the blue area. It is possible to see that for the blue area the actuator 1 and 2 maximums are a constant values and no depends on the other actuator value. Therefore the solution can be represented like in the right side as a two unidimensional solution. However also is possible to see how the simplification despise the grey area that are a set of valid solution.

This example can be extrapolated to an n-dimensional one. In the Barcelona DWTN there are 61 actuators therefore with this simplification it is possible to convert a complex 61-dimensional space into a 61 one-dimensional space.

Moreover into the optimization problem is use the same control input bound for all the prediction horizon, that are also conservative. At this point seems difficult to control the reliability of the system analytically due the the fact that the method is very pessimistic. On the other hand it is possible to tune experimentally α in other to obtain the desired reliability.

It is important to mention that the evolution of the reliability depends on the actual reliability of all the actuators. For this reason it is necessary make the new actuators bounds computation on each iteration. It is specially important when the differences between actuators reliability are high.

3.4 Application to Test model

The purpose of this section is to implement the method that has been described above in a test system. The motivation to do this is twofold. For one side the test model is simple and do not take to much computational time in other to simulate them, therefore it is easier to extract results and to test some variations. The second motivation is to explain the method and the equations in a not too much complex example.

3.4.1 Case of study description

The test model consist in a small part of the Barcelona DWTN.

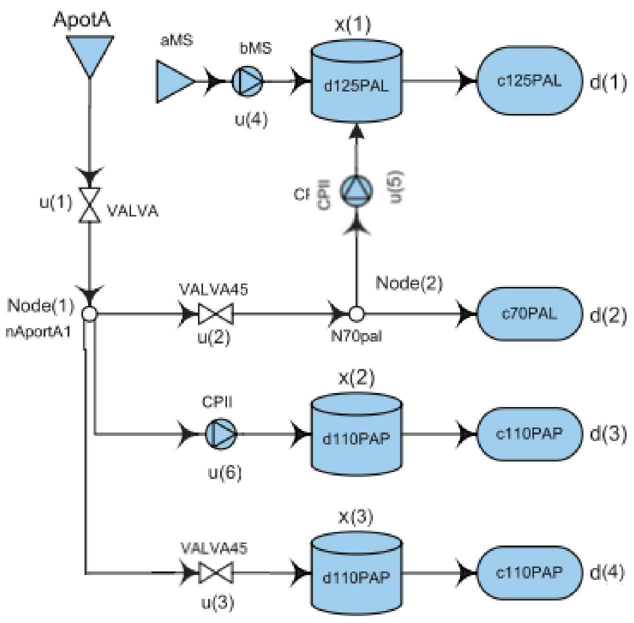


Figure 3.3: Test model diagram

The test model has 4 demands $n_d = 4$, 3 tanks $n_x = 3$, 2 nodes $n_q = 2$, 6 actuators $n_u = 6$ and 2 sinks $n_r = 2$. Figure 3.3. It is important to remark that not all of actuators have the same bounds and also tanks have different capacities.

The optimisation of the control action is achieved through the minimisation of the operational cost, that includes the electrical cost associated with the transport and the cost associated with the exploitation of the water resources, satisfying the demands in each consume sector. On the other hand the electricity tariffs change depending on the time of the day.

Due to the fact that the water demands describes a cycle of 24 hours the prediction horizon (Hp) is 24 hours and the sampling time (T_s) is 1 hour.

3.4.2 Implementation over the test model

Following the method the first step is to describe the reliability of the whole system decomposing the system in connections of smaller subsystems. It is possible following the methodology explained before in 3.1

The total reliability of the system R_g is represented by using (3.3). For this system with 4 demands:

$$R_g(k) = R_{d_1} R_{d_2} R_{d_3} R_{d_4} \tag{3.13}$$

Then using (3.4) is possible to decompose the demand reliability R_d.

For demand 1:

$$R_{d_1}(k) = 1 - (1 - R_{1_{path1}})(1 - R_{1_{path2}}) \tag{3.14}$$

Now is possible to compute the reliability of the both path ($R_{1_{path1}}$ and $R_{1_{path2}}$) that demand 1 has in terms on the reliability of actuators R_a by using the (3.5).

$$R_{1_{path1}}(k) = R_{a_4} \tag{3.15}$$

$$R_{1_{path2}}(k) = R_{a_1} R_{a_2} R_{a_5} \tag{3.16}$$

Combining (3.15) and (3.16) with (3.14),

$$R_{d_1}(k) = 1 - (1 - R_{a_4})(1 - R_{a_1} R_{a_2} R_{a_5}) \tag{3.17}$$

On the other hand demands 2, 3 and 4 do not have parallel paths, Figure 3.3. Therefore is possible to simplify the (3.4) :

$$R_{d_2}(k) = R_{a_1} R_{a_2} \tag{3.18}$$

$$R_{d_3}(k) = R_{a_1} R_{a_6} \tag{3.19}$$

$$R_{d_4}(k) = R_{a_1} R_{a_3} \tag{3.20}$$

So finally substituting (3.17), (3.18), (3.19) and (3.20) in (3.13) the system reliability equation in terms on actuators reliability is obtained.

$$R_g(k) = (1-(1-R_{a_4})(1-R_{a_1}R_{a_2}R_{a_5}))R_{a_1}R_{a_2}R_{a_1}R_{a_6}R_{a_1}R_{a_3} \tag{3.21}$$

Furthermore it is possible to describe the actuators reliability as:

$$R_{a_1}(k) = R_{a_{1_0}} e^{-\lambda_1^0 e^{(\beta_1 |u_1|(k))} t} \tag{3.22}$$

$$R_{a_2}(k) = R_{a_{2_0}} e^{-\lambda_2^0 e^{(\beta_2 |u_2|(k))} t} \tag{3.23}$$

$$R_{a_3}(k) = R_{a_{3_0}} e^{-\lambda_3^0 e^{(\beta_3 |u_3|(k))} t} \tag{3.24}$$

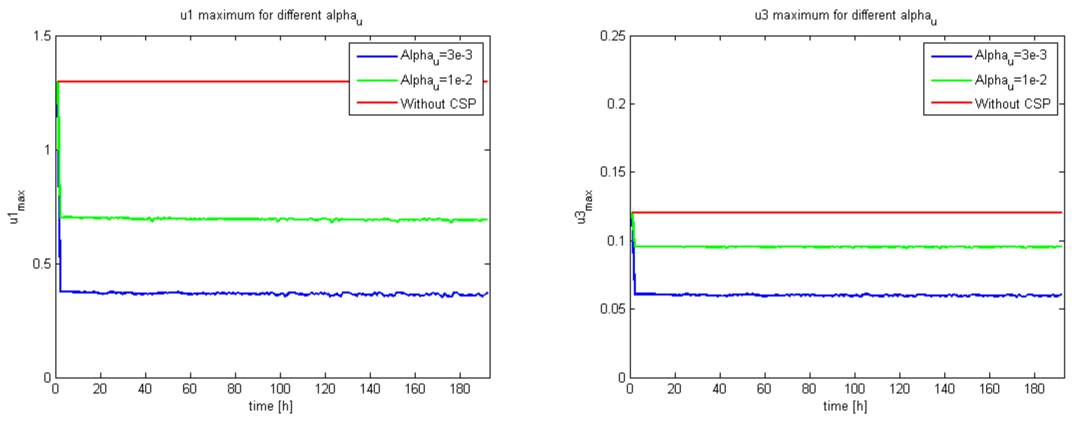
$$R_{a_4}(k) = R_{a_{4_0}} e^{-\lambda_4^0 e^{(\beta_4 |u_4|(k))} t} \tag{3.25}$$

Finally is possible to use (3.22), (3.23), (3.24) and (3.25) with (3.21) obtaining a expression of the reliability of whole system only in terms on initial reliability of actuators, constant parameters of actuators and the control action. Is suppose that the actual reliability of each actuator is know or can be estimated. Therefore it is possible to relate directly the system reliability with the control actions.

By using (3.3.2) is possible to determine the new bounds of $R_t(k+N)$ by fixing the tune parameter α and computing the actual system reliability $R_t(k)$. Therefore using CSP the $R_t(k+N)$ bounds are propagated by using (3.22), (3.23), (3.24), (3.25) and (3.21) obtaining finally a new actuators bounds. Of course as it have been said before this bound contraction has been computed in each iteration.

In this point is possible to see how the method works by present some results. As it has been said before the reliability constraint is actually convert into a control action constraint. For loose reliability

constraints it is possible that the control action bounds are no affected but if the reliability constraints are heavier then it is expected that the control action maximums become smaller. This effect is shown in Figure 3.4. As More restrictive reliability constraint as less control action maximum.
Is important to keep in mind that the constraint is the reliability of the whole system. It means that not all the control actions are affected in the same manner. It depends, for example, in: the number of path through the actuator, the number of parallel paths for one demand that involves the actuator, the actual actuator reliability, the degradation parameters β and λ, etc. For example in eq. 3.21 the term R_{a_1} appears several times, it is because all of demands can be achieved by using the actuator 1, therefore seems logical that the reliability of this actuator is very significant for the total reliability. This effect is shown in Figure 3.4 where the actuator 1 is more affected than the actuator 3.



(a) Control input 1 (b) Control input 3

Figure 3.4: Maximum control input along the simulation

However, it is possible that the contraction of actuators bounds does not have any effect on the system behaviour. It happens if the optimal solution is not for $u_{opt} = u_{max}$. On the other hand due to the redundancy of the system is possible that exist other solution with equal cost, (two global minima). In the first case the cost and the reliability of the system for this actuator is not affected due to the reliability constraints, in the second case is possible to improve the reliability of the system but maintain the same operational cost. With $\alpha_u = 3 \times 10^{-3}$ the reliability of the actuator has been improve because of the new actuator bound fixed by the reliability constraint of the system. Fig, 3.5. It is because the area of control action is the same (same total flow), but the maximum values of control action are less than the maximum values before the reliability constraint. Since the actuators degradation are exponential the reliability of this actuator has been improved. Moreover this actuators is a valve, therefore

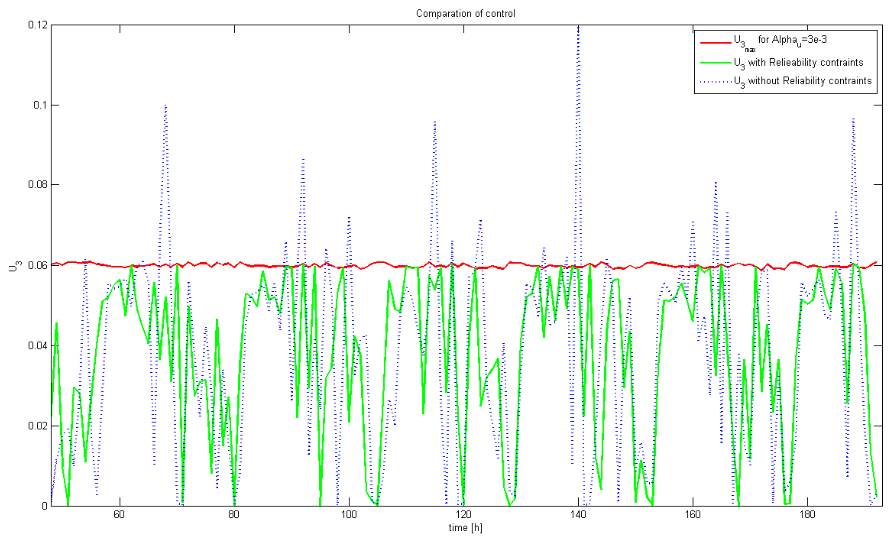


Figure 3.5: Control input 3 comparison (with vs. without) Reliability constraints. $\alpha = 3 \times 10^{-3}$

do not have associated operational cost. Its means that is possible to increase the reliability of this actuator without increasing the operational costs.

In order to endow to the system the possibility to work with more restrictive constraints avoiding possible infeasibility problems has been decide make the new actuator constraint as a soft constraints. With this method the new actuators bounds are not longer a hard constraints but are penalised in the optimisation problem cost function. In Figure 3.6 it is possible to see one example with $\alpha = 1 \times 10^{-3}$.

However the actuator 3 is a valve, fig 3.3. For the valves is not important work during the day or the during night because are no affected by the electricity cost. It is not the case of the actuator 6, that is a pump. Pumps try to work during the night, when the electricity is cheaper. But the new actuators constraints take out the possibility to only work during the night, Figure 3.7. Therefore the actuator is working during the day increasing the electric operational cost.

Different control action implies different evolution of the tank volumes. Fig, 3.8. In this case the evolution of the states follow the same pattern with small differences. It is because the objective function is the same so the optimal evolution is similar but with the adaptation to the new control action bounds.

The objective of the study is to increase the system reliability. In Figure 3.9 is shown the total system

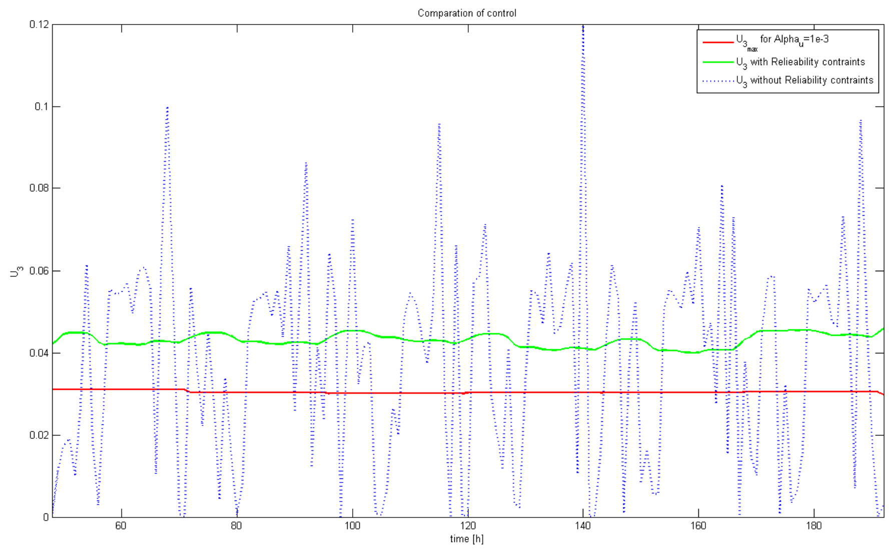


Figure 3.6: Control input 3 comparison (with vs. without) Reliability constraints. $\alpha = 1 \times 10^{-3}$

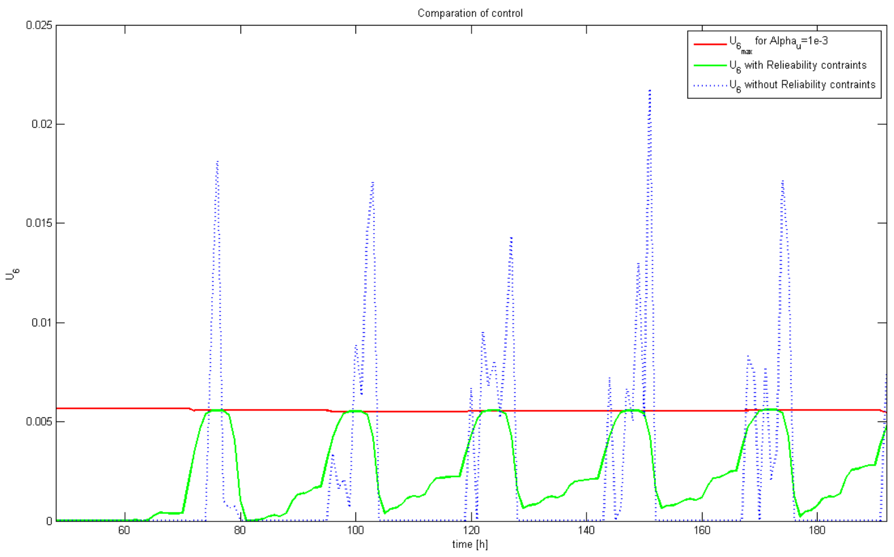


Figure 3.7: Control action 5 comparison (with vs. without) Reliability constraints. $\alpha = 1 \times 10^{-3}$

reliability evolution. Along the simulation time (8 days) the system is able to preserve more reliability with reliability constraints.

Putting some numbers, with $\alpha_u = 2 \times 10^{-3}$ it is possible to save the 11.9% of the loss of reliability in

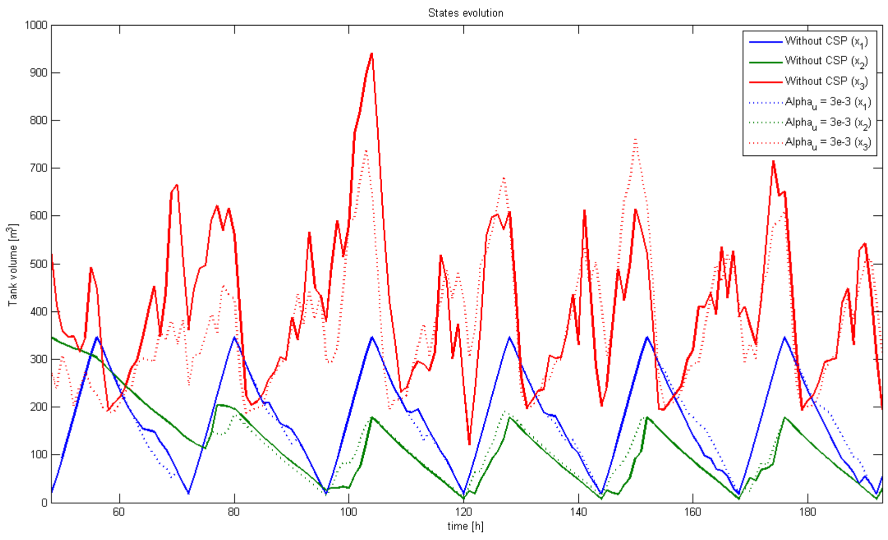


Figure 3.8: Tank volume comparison (with vs. without) Reliability constraints

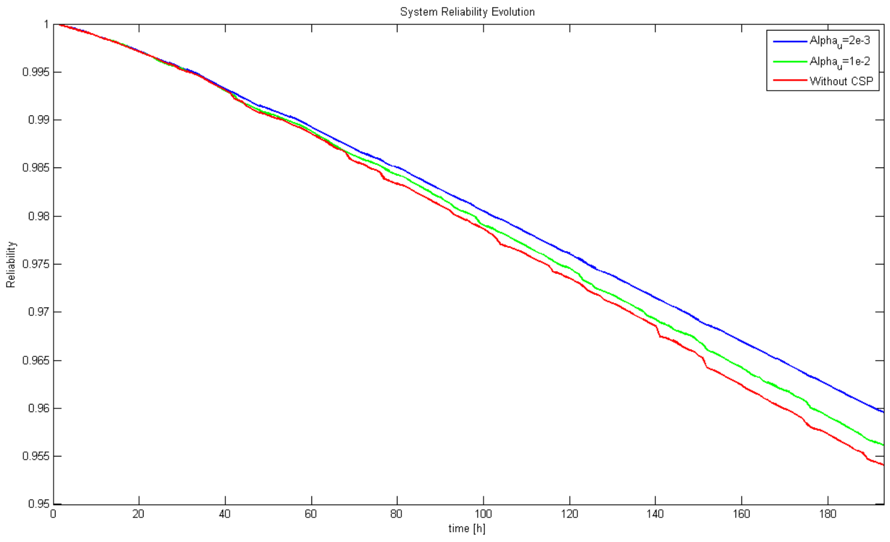
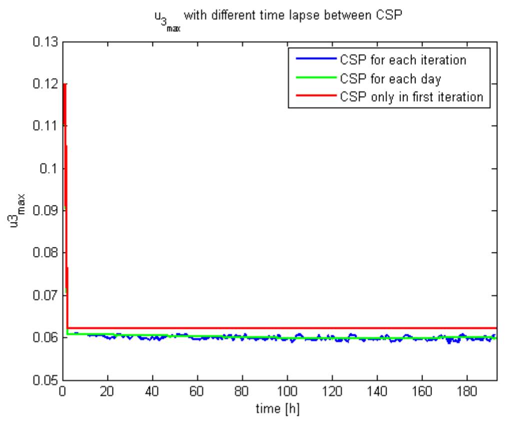
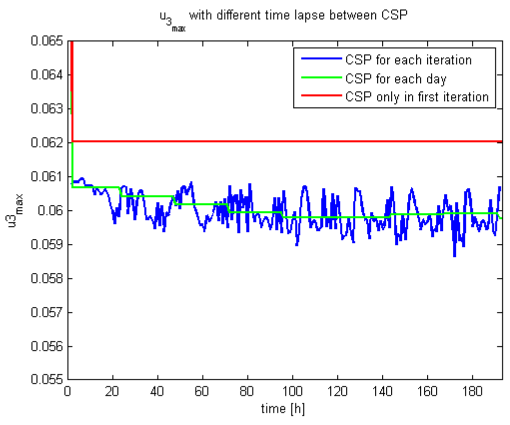


Figure 3.9: Total system reliability evolution

8 days. Moreover this saving becomes more important as longer the simulation is. It has sense since the degradation of the actuators are exponential, therefore the reliability of the system decrease faster if it is lower. It is possible to see also in fig, 3.9.



(a) Control action 3



(b) Control action 3 (Zoom in)

Figure 3.10: Maximum control input 3 compitung CSP per every hour (blue), once per day (green) and once per week (red)

Introduce the contraction of the control action bound in each iteration increase the computational cost. Although the actual system is small an the computation time is not a problem it can be a problem with the real system. For this reason has been test some variation of the algorithms. In order to save computation cost the contraction of the actuators bound is not longer made in each iteration but with a lapse time.

The results, fig 3.10, give useful information. It is possible to see that only contracting the bounds in the first simulation step the results are no sufficient good. Otherwise contracting the bound one time each day (24 simulation steps) it is possible to save a 96% of computation time expend in contraction without relevant results changes.

Chapter 4

Implementation in the Barcelona DWTN

4.1 Case of study description

The MPC approaches presented in this project will be assessed with a case study of a large-scale real system, specifically the Barcelona DWTN. The general role of this system is the spatial and temporal re-allocation of water resources from nature to human society, keeping in mind quantitative and qualitative aspects of water availability and human needs.

In a DWTN the water enter into the system from the sources. In Barcelona DWTN the water is taken from both: superficial sources(i.e., rivers) and underground sources (i.e., wells), providing together a ow of around 7 m^3/s. The water flow from any of the sources is limited and has an associated price depending on the required treatment and legal extraction canons.

Barcelona DWTN is a real system with real demands, actuators bounds, and tanks. For this study has been considered 9 water sources, that includes 5 underground and 4 superficial, 17 water tanks, 61 actuators (valves and pumps), 12 nodes and 25 demands. Figure 4.1 shown the topology of the network. Both the demand episode and the calibration set-up of the network are provided by AGBAR. The current AGBAR control centre has a tele-control system for the network management.

The Barcelona DWTN is also comprised of more than 98 remote stations, which manage in real time about 450 elements such as ow meters, pumping stations, valves, chloride dosing instruments, among others. The objective of the MPC as has been explained before is to minimise the multiobjective cost function (2.12).

As has been explained before the economical cost is associated to 2 factors (2.9). One of them is the electrical cost that is related with the electrical tariff. In Barcelona the cost of a kWh is not always the same and it is cheaper during the night. For this reason and in order to save money more electricity is expended during the night. The prediction horizon is 24 hours because the system and also the electrical tariff have periodicity of 1 day. The sampling time is 1 hour.

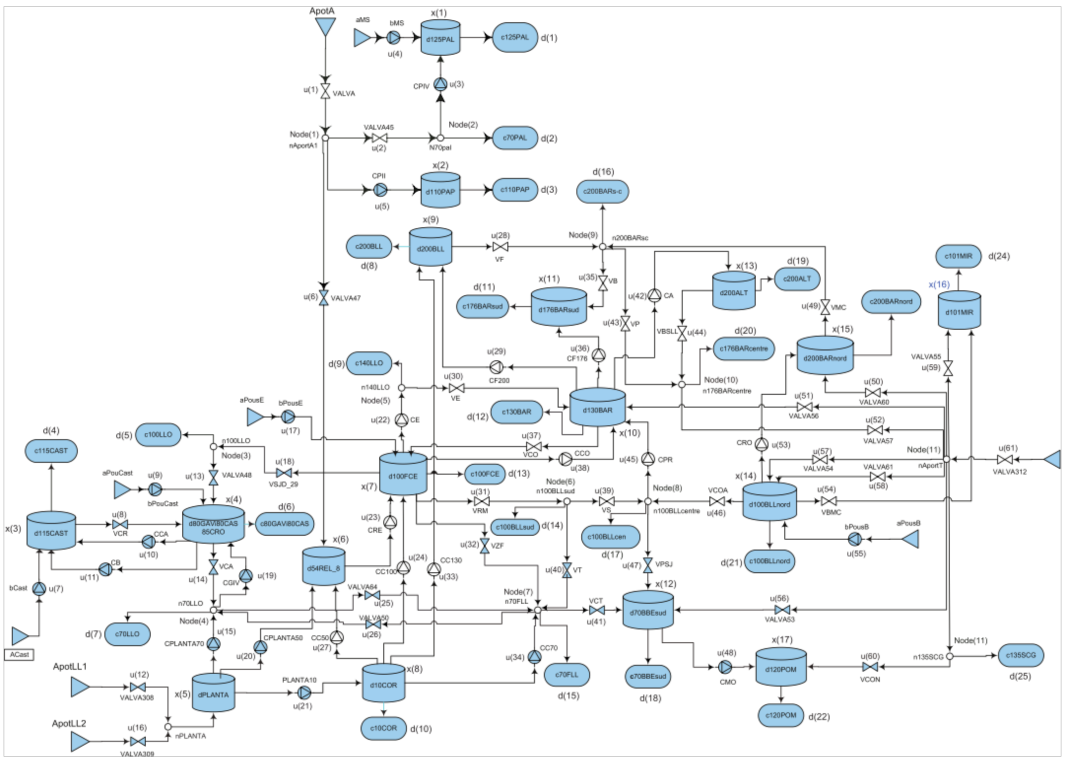


Figure 4.1: Barcelona DWTN 17 tank model

4.2 Results

In this section have been explained the results obtained from the application of the approach explained during the project to the Barcelona DWTN.

The first result is the analysis of the computation time expend during the simulations. All of simulation are computed in the same PC with an AMD phenom X6 1090T at 3.2 GHz CPU and 8 GB of

DDR3 RAM, without the support of video card. Also all of simulation have the same simulation horizon (192 hours), Table 4.1

Table 4.1: Computational Time

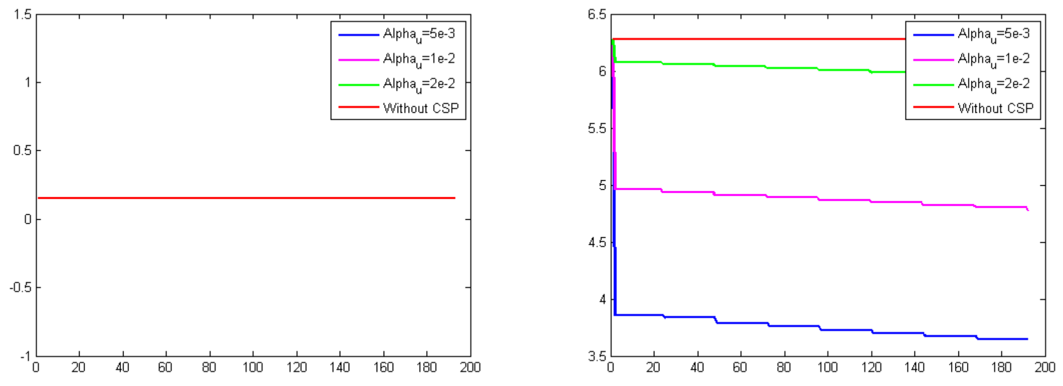
	Time MPC (min)	% MPC	Time CSP (min)	% CSP	Total Time (min)
With $\alpha = 2 \times 10^{-3}$	48	18.3	214	81.7	262
With $\alpha = 5 \times 10^{-3}$	40	16.6	201	83.4	241
With $\alpha = 2 \times 10^{-2}$	38	15.5	206	84.5	244
Without CSP	39	100	0	0	39

The percentage of the time used in order to compute the CSP will be discussed below in the conclusions chapter, but the main conclusion is that the solver used to solve the CSP is non-optimal for this application and for this reason the CSP computation is time-costly. On the other hand the time expend to solve the simulation is practically the same for all simulation with the exception of the simulations with a very restrictive reliability constraints. When the reliability constraint are more important, due to have lower values of α or because the system reliability is low, the computational time increase significantly.

The next point is the u_{max} analysis. As it happens the previous simplified model not all the actuators bounds are affected with the same severity. For example actuator 5 (Figure 4.1) is important because is the only way to satisfy the demand 3. Other actuators are also important because are involved in a lot of demand paths, for example actuators 12, 16 or 61. Others actuators like actuator 35 are not so important because exist a parallel path in order to satisfy the demand, in this case demand 11.

Therefore the degradation of each actuator affects to the degradation of the system in different ways and magnitudes. For this reason the new control action bounds resulting from the reliability constraints are different for each actuator.

As has been said before the actuator 35 reliability is not so relevant because exist a parallel actuator. In Figure 4.2 it is possible to see that the actuator bounds are not affected by the reliability con-



(a) Control input 35 (b) Control input 61

Figure 4.2: Maximum control input along the simulation

straints. On the other hand the actuator 61 is important because is involve in a lot of demands. For this reason the actuators bounds decrease as more restrictive reliability constraint are impose.

In Figure 4.2 also is possible to see that the bounds become more restrictive along the simulation time. As it has been said before the reliability of the actuators are exponential, its means that as less reliability more reliability the actuator lost for the same control action. This fact bring the result, as more work time, less reliability and more restrictive bounds in other to constraint the reliability of the whole system.

On the other hand a new more restrictive control action bounds maybe not suppose a change in the control action along the simulation. It means that the actuator have a surplus and the new restriction no affects their behaviour. In this case the actuator is not affected by the reliability constraint, therefore the reliability for the actuators is not preserved. It is the case of actuator 61, fig 4.3.

The previous case can be a problem if it is the case of all of actuators because, if it happens, the new reliability constraint not longer affects the system behaviour. If it not the case interesting, the result is quite interesting because only the more sensible actuator, those who affect in a more relevant manner to the reliability of the system, change their behaviour in order to maintain the system reliability. On the other hand, since it is a dynamic system, it is possible that the evolution of the actuator bounds change and in some point the behaviour of the actuator must change in order to preserve the reliability constraints.

On the other hand actuator 5 is affected by the new bounds, Figure 4.4. The new bound increase the

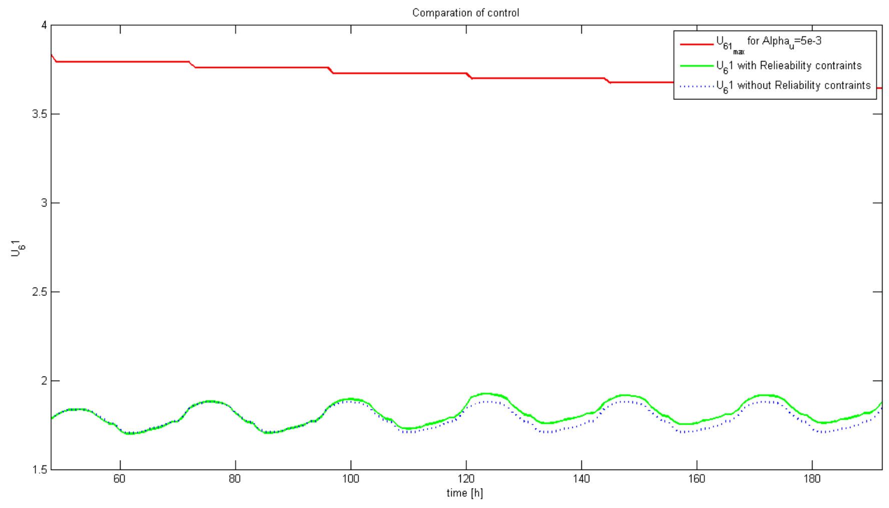


Figure 4.3: Control action 61 comparison (with vs. without) Reliability constraints

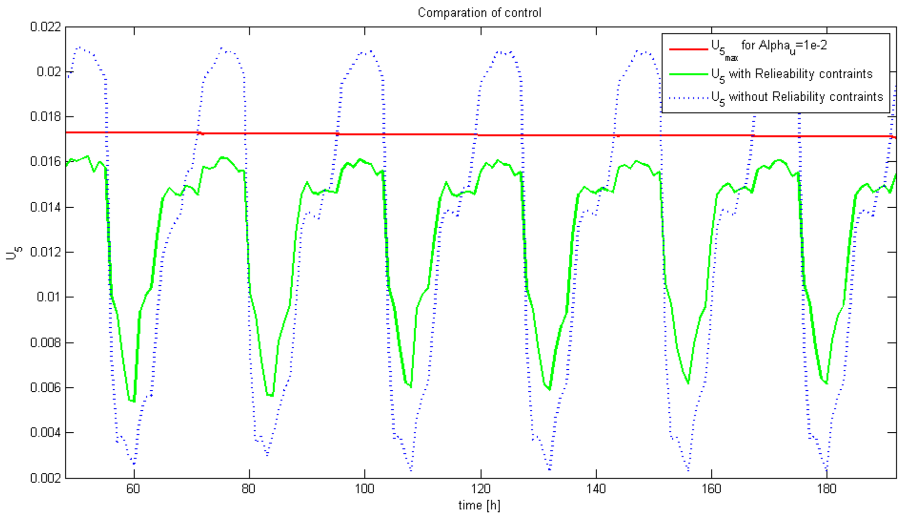


Figure 4.4: Control action 5 (with vs. without) Reliability constraints ($= 1 \times 10^{-2}$)

final reliability of the actuator and therefore the final reliability of the whole system.

In Figure 4.4 the actuator is able to pump almost all water during the night and scarcely during the day. Increasing the reliability constraint this is not longer possible, Figure 4.5. Therefore the operational cost increase in other to increase the reliability pumping water during the day.

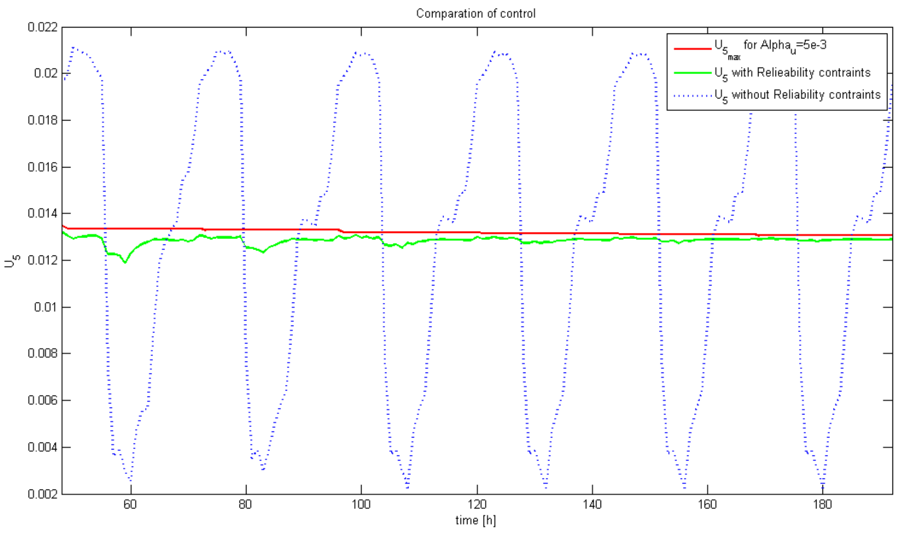


Figure 4.5: Control action 5 (with vs. without) Reliability constraints ($\alpha = 5 \times 10^{-3}$)

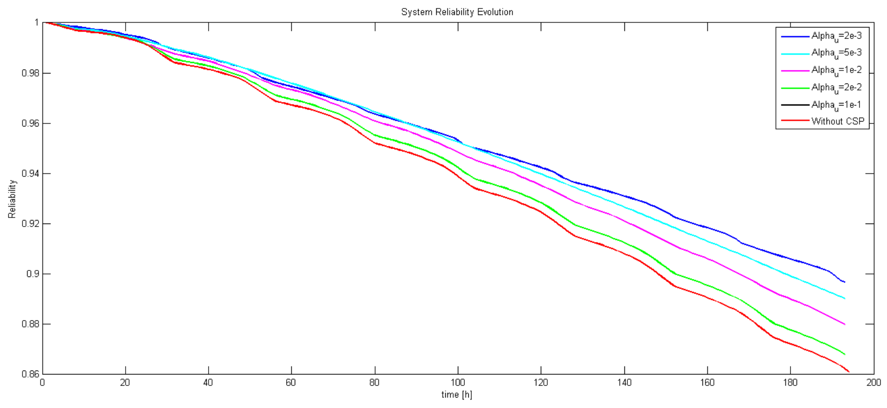


Figure 4.6: Total system reliability evolution

So finally we can compare the system reliability evolution for different constraints. As it is expected as more restrictive constraint as more final reliability the system have. Also it is possible to see that as more time lapse more increment on the reliability.

In Figure 4.4 and 4.5 it is possible to see that no all of constraint affects in the same manner to the system behaviour. Due to the fact that the system have more than one optimal solutions with the same economical cost is possible to change the behaviour of the system without an increment on the operational cost. For example pumping water from two pump instead of only one, therefore reducing the

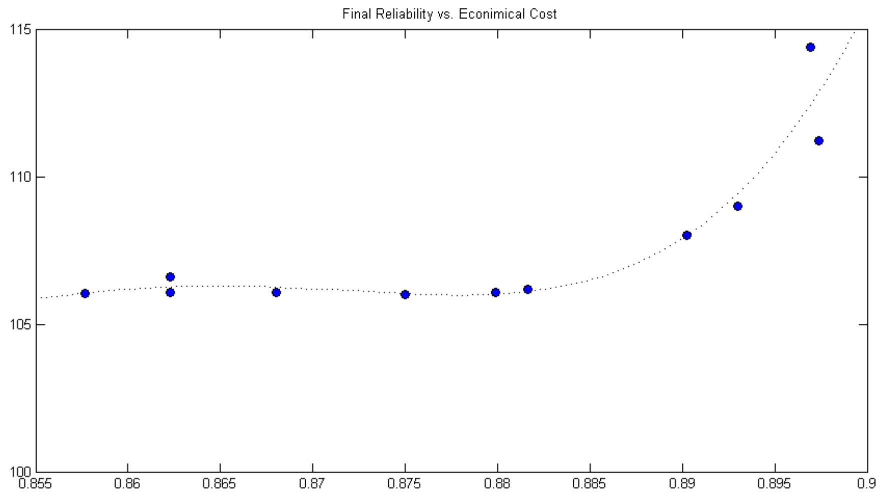


Figure 4.7: Final System Reliability vs. Economical Cost for different α

effort of actuators but with similar cost. Or pumping water during more hours of the night with the same cost. This is a relevant result because actually is possible to increase the reliability of the system at cost 0.

On the other hand if the reliability is crucial is possible to increase the restriction and preserve more reliability. It is possible by tuning α_u parameter. In this case the problem converts into a trade-off between reliability and operational cost. Figure 4.7 and Table 4.2.

Table 4.2: Final System Reliability vs. Economical Cost

	Final Relibility ($R_t(192)$)	Economic Cost for 192 h.
$\alpha = 2 \times 10^{-3}$	0.89691	114.37
$\alpha = 3 \times 10^{-3}$	0.89737	111.2
$\alpha = 5 \times 10^{-3}$	0.89027	108.01
$\alpha = 7 \times 10^{-3}$	0.88161	106.19
$\alpha = 1 \times 10^{-2}$	0.87989	106.09
$\alpha = 2 \times 10^{-2}$	0.86801	106.09
$\alpha = 3 \times 10^{-2}$	0.85767	106.03
$\alpha = 5 \times 10^{-2}$	0.85767	106.03
$\alpha = 1 \times 10^{-1}$	0.86229	106.06
Without CSP	0.86229	106

At this point can be interesting to see what happens if the reliability of the system is lower that in the previous cases.

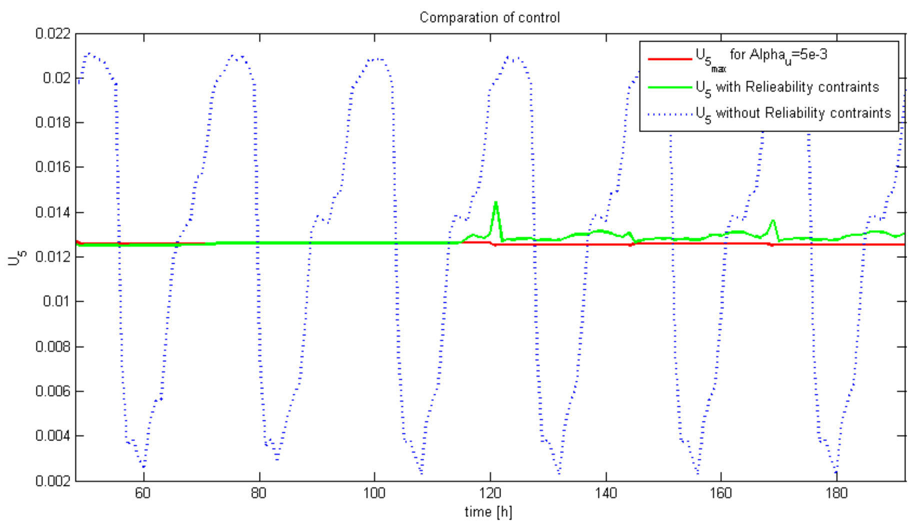


Figure 4.8: Control input 5 (with vs. without) Reliability constraints ($\alpha_u = 5 \times 10^{-3}$) and $R_{a_0} = 0.9$

Comparing fig 4.5 and 4.8 is possible to see that for the same α with less system reliability the reliability constraints affects more and the bounds are more restrictive. Also is possible to see that during the hour 116 it is not possible to respect the new bounds, due to the fact that the new actuator constraint is a soft constraint the system is able to continue working.

Fig 4.9 are the results when all of actuator of the system have a initial reliability equal to 0.7. Of course the bound are more restrictive and it is not possible to satisfy the new bound constraint. On the other hand and other relevant result in Figure 4.9 is how the bound evolves along the simulation time. As less reliability the system have more faster the bounds become more restrictive.

Finally in Figure 4.10 we have the evolution of the cost versus the reliability of the whole system. It has sense that for more system reliability less cost since the bounds are less restrictive when the reliability are higher.

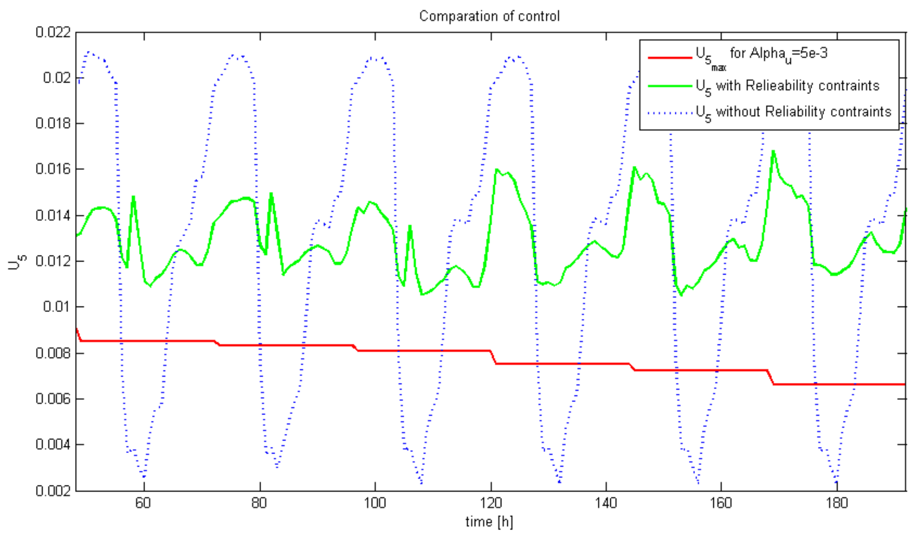


Figure 4.9: Control action 5 (with vs. without) Reliability constraints ($\alpha = 5e^{-3}$ and $R_{a_0} = 0.7$)

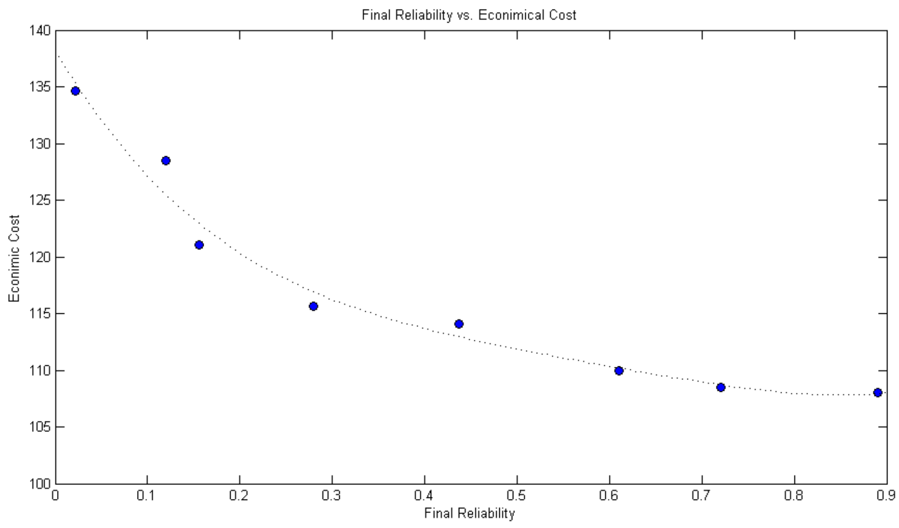


Figure 4.10: Final System Reliability vs. Economical Cost for different R_{a_0}

Chapter 5

Environmental Impact

Water is a fundamental and limited resource. During the last years the overpopulation in the cities has resulted in a over-exploitation of this resource and nowadays supply the demand becomes a challenge. For this reason is very important to invest founds in order to improve the water networks and to develop new modern control systems that allow to save water and energy.

Since the application of the new control techniques to the DWTN, like MPC, the performance of these has increase a lot, saving energy and water. With the new proposal explained in this project it is possible to achieve a new long term saving by using the reliability analysis in order to increase the life time and improving the state of the actuators.

Increase the life time of something is always good in terms on environmental impact. Moreover actuators in better conditions implies less energy consumption and less water loses. On the other hand increase the life time of the actuators implies less number of failures. A failure in this kind of systems implies the waste of energy in order to satisfy the demands since the optimal paths are not able and in some cases the waste of water.

Chapter 6

Budget

In this chapter a budget breakdown is shown. In the budget has been included the tools and licenses necessary to develop and to implement the project as well the references like papers and books. Also has been include the salary derived from the number of hours of an engineer used to research and to develop the implementation.

Table 6.1: Budget Breakdown

Concept	**Unitary Price**	**Units**	**Total Cost**
Salaries	30 €/h	280h	8,400 €
Laptop	700 €	1	700 €
MatLab License	2,000 €	1	2,000 €
YALMIP License	LGPL	1	0 €
Ibex library	LGPL	1	0 €
IBM ILOG CPLEX Optimization Studio	8,689.50 €	1	8,689.50 €
TOTAL			19,789.50 €

Chapter 7

Concluding Remarks

7.1 Contributions

This project has proposed an MPC including inside the optimisation problem the reliability constraints through the contraction of the actuators bounds by using CSP. It has been shown, through a real case study, the effectiveness of the proposed MPC designs, which enhanced a certainty equivalent MPC approach, by incorporating forecast demand and system health monitoring, to assure reliability in DWTNs and to minimise operational costs.

Some final remarks are drawn below.

- This method incorporates the reliability into the MPC taking into in account the whole system reliability. It is a important improvement compared to other approaches that describe the health of the system in terms on reliability of all of actuators without take into in account the network topology.

  With the new description not all actuators affect in the same way the system reliability. Therefore it is possible to increase the system reliability without operational cost by using for example parallel path in order to supply one demand.

- After simulation it is possible to compute the trade-off between cost and reliability. Therefore it is possible to select the system reliability level in terms of operational cost and to tune the reliability parameter in order to obtain this level.

- The proposed robust MPC controller that handle actuator health management, present higher long term economic performance since take into in account not only the economic cost of water and transport but also the economic cost associated with the maintenance.
- The model of the system treated in this thesis fits in the framework of flow networks, hence, the proposed controllers could be applied to other complex systems such as transport networks (e.g., gas, oil, energy), supply-chain problems, production planning, industrial processes with flows and storage control, among others.

7.2 Directions for Future Research

- In this project reliability is describes as the probability of exist at least one path to arrive to all of demands and does not take in consideration the possibility that this path are not able to satisfy the 100% of the demand. Therefore the next step is to use the same idea of CSP but in this case using the predicted demands to compute the security tanks level, of course taking into in account the network topology.
- The computational cost has been increased a lot after incorporate the contraction of the control inputs. However it is because the solver that has been used during the project in order to compute the CSP is not specifically designed for this purpose. It is shown that by using other solvers it is possible to save around the 98% of computational time expended in contraction, therefore before a real implementation of this method it is necessary the development of a dedicated solver for this problem.
- This project has the purpose to develop and to simulate a Reliable MPC for the Barcelona DTWN. The next step could be to merge the reliability control method proposed in this project with a Fault Tolerant Control (FTC) that is already developed for the Barcelona DWTN. Thus it is possible to improve the reliability analysis by using FTC techniques and to select optimal control strategies in order to maintain the reliability level even in a fault condition.

Acronyms

MPC	Model Predictive Control
NMPC	Nonlinear Model Predictive Control
FTC	Fault Tolerant Control
DWN	Drinking Water Network
DWTN	Drinking Water Transport Network
CSP	Constraint Satisfaction Problem

Appendix A

Main Code, Reliable Model Predictive Control

```

% 17 de Abril de 2016
% IRI-UPC

close all
clear all
yalmip('clear')
clc

h = waitbar(0,'Creating problem structure, please wait...');

%% Source data & System Description
% load('Data3tanks.mat');    % 3 tanks network
load('Data17tanks.mat');   % 17 tanks network
% load('Data63tanks.mat');  % 63 tanks network
% load('x_initial.mat');

% load('Sources3tanks.mat');
load('Sources17tanks.mat');
```

```
% load('Sources63tanks.mat');


%-----------------------------------------------%
%------------ Begin Model-MPC data ------------%
%-----------------------------------------------%

A = S.A; B = S.B.*3600; Bp = S.Bp.*3600;            % x_{k+1} = Ax_{k} + Bu_{k} + B_{p}d{k}
Eu = S.E.*3600; Ed = S.Ed.*3600; d = S.d;           % 0 = E_{u}u_{k} + E_{d}d_{k}
% A = S.A; B = S.B; Bp = S.Bp;                % x_{k+1} = Ax_{k} + Bu_{k} + B_{p}d{k}
% Eu = S.E; Ed = S.Ed; d = S.d;               % 0 = E_{u}u_{k} + E_{d}d_{k}
alpha1 = S.alpha1; alpha2 = repmat(S.alpha2,9,1);   % costs data
umax = S.umax'; umin = S.umin';               % limits for control actions
xmax = S.xmax'; xmin = S.xmin'*0;               % limits for control states
xs = S.xpenal';                              % safety volumes

% Reliability Parameters
Lamda=ones(1,size(S.umin,2))*0.00002;     %delta: temporal degradation
Beta=log(50)./S.umax;                        %beta in terms on (u/umax). log(x) x= max degradarion due to inputs respect to temporal degradation
% Alpha=0.998;                                % R(n)=Alpha*R(0)
uAlpha =5e-1;                            %Maximun allowed degradation due to the control actions
Ra(:,1)=ones(size(S.umin,2),1)*1;          %Initial actuator Reliability
```

```
Rt(1)=1;                                       %initial whole system Reliability
bReliability = 'true';                       %'True' for performing realibility bounding
CSPtime=24;                                    %Time between contraction: CSPtime=1 allways, =0 only one time, =24 one per day

dd= [d ; d];

nx = size(A,2); % number of states
nu = size(B,2); % number of control actions
nd = size(Bp,2); % number of demands
ne = size(Eu,1);     % Number of nodes (equality constraints)
ns = size(Sources,1); % Number of sources

p = [100;10;1;1000];                          % vector priority weights
p1 = p(1); p2 = p(2); p3 = p(3); p4 = p(4); % priority weights

days=8;        % days for the simulation
Hp = 24;       % horizon of prediction
Hs = 24*days; % horizon of simulation

coefs = 0.8*xmax; % coeficients for initial conditions
x0 = [xmin' + coefs']';                      % initial conditions


%% Run The pathfinder

FindPath();

%% ------------ Begin MPC controller ------------%

uopt=zeros(nu,1); % u{0} initial optimal control action
```

```

for t=1:Hs
    ti = clock;
    t % publish current iteration

    du=sdpvar(repmat(nu,1,Hp),repmat(1,1,Hp));      % \Delta u{k}=u{k}-u{k-1}
    u=sdpvar(repmat(nu,1,Hp),repmat(1,1,Hp));       % u{k}
    xi=sdpvar(repmat(nx,1,Hp+1),repmat(1,1,Hp+1)); % \xi{k}
    ui=sdpvar(repmat(nu,1,Hp),repmat(1,1,Hp));
    x=sdpvar(repmat(nx,1,Hp+1),repmat(1,1,Hp+1));

    const = [];
    obj = 0;
    x{1}=x0;

    % Run CSP
    if ((t==1) || (strcmp(bReliability,'false')))
      umin_csp(t,:)=umin;
      umax_csp(t,:)=umax;
    elseif ((strcmp(bReliability,'true')) && (t==2))
        fprintf('Contracting actuators bounds')
        % Compute new actuators bounds
        [alphamin(t,:),alphamax(t,:),umin_csp(t,:),umax_csp(t,:)]=ContractorCSP(Alpha,umin',umax',Lamda,Beta,Hp,Ra(:,t),'solver_in.bch',nd,nu,act_path,Rt(t));
    elseif ((strcmp(bReliability,'true')) && (mod(t,CSPtime)==1))
        fprintf('Contracting actuators bounds')
        % Compute new actuators bounds
        [alphamin(t,:),alphamax(t,:),umin_csp(t,:),umax_csp(t,:)]=ContractorCSP(Alpha,umin',umax',Lamda,Beta,Hp,Ra(:,t),'solver_in.bch',nd,nu,act_path,Rt(t));
    else
```

```
     umin_csp(t,:)=umin_csp(t-1,:);
     umax_csp(t,:)=umax_csp(t-1,:);
    end

    % Run MPC
    for k = 1:Hp

      dis=dd(t-1+k,:)';
      x{k+1} = A*x{k} + B*u{k} + Bp*dis;

      du{k}=u{k}-uopt;
      if k>1
        du{k}=u{k}-u{k-1};
      end

      f1=(alpha1+alpha2(t-1+k,:))*u{k};
      f2=du{k};
      f3=xi{k};
      f4=ui{k};
      obj=obj+norm(p1*f1,1)+norm(p2*f2,2)+norm(p3*f3,2)+norm(p4*f4,2);
      const=[const, Eu*u{k}+Ed*dis==0];
      const=[const, xmin <= x{k} <= xmax];
      const=[const, umin <= u{k} <= umax];
      const=[const, [umin_csp(t,:)'-ui{k}] <= u{k} <= [umax_csp(t,:)'+ui{k}]];
      const=[const, x{k} >= [xs-xi{k}]];
    end

    sdpsettings('solver','cplex')

    solvesdp(const,obj);      % solve for the optimization problem
    clc
    duopt=double(u{1})-uopt; % compute \Delta u{k}
    uopt=double(u{1});       % compute u*{k}
    xiopt=double(xi{1});     % compute xi*{k}

    %---- saving array data ----%
    ccmpc.x(t,:)=x0;
    ccmpc.u(t,:)=uopt;
    ccmpc.du(t,:)=duopt;
    %---------------------------%
    x0=A*x0 + B*uopt + Bp*dd(t,:)'; % update for states x0

    % Computing cost
    ccmpc.Water(t)= alpha1*abs(ccmpc.u(t,:)');
    ccmpc.Electric(t)=alpha2(t,:)*abs(ccmpc.u(t,:)');

    % Compute the new actuator Reliability
    [Ra(:,t+1),Rc(:,:,t+1),Rd(:,t+1),Rt(t+1),Ran(:,t+1),Rcn(:,:,t+1)
       ,Rdn(:,t+1),Rtn(t+1)]=Reliability_computation(act_path,nd,
       Beta,Lamda,ccmpc.u(t,:),Ra(:,t)',Hp);

    alphamax_2(t)=Rtn(t+1)/Rt(t+1)   %%TEST
    Alpha = alphamax_2(t)-uAlpha      %TEST Alpha in terms on
       temporal

    % Clock
    tf(t) = etime(clock,ti);
    waitbar(t/Hs,h,sprintf('Progress: %.2f%% completed \n Time
       remaining: %.2f minutes',...
            t/Hs*100,tf(t)/60*(Hs-t)));

end

close(h,'force')
```

```matlab


%% Results
figure(1)
subplot(2,2,1); plot(ccmpc.x,'linewidth',2); grid on;
title('Evolution of states x')
xlabel('time [hours]')
ylabel('volumes [m^3]')

subplot(2,2,2); plot(ccmpc.u,'linewidth',2); grid on;
title('Evolution of control actions u')
xlabel('time [hours]')
ylabel('flows [m^3/s]')

subplot(2,2,3); plot(ccmpc.du,'linewidth',2); grid on;
title('Evolution of variation of actions du')
xlabel('time [hours]')
ylabel('variations [m^3/s]')

uisave
```

Appendix B

Reachability analysis Code

```

% Programa para buscar todos los caminos desde todas las fuentes
    hasta
% todas las demandas de un sistema con nodos

%% Load system

MatrizA = S.A;
MatrizB = sign(S.B);
MatrizC = eye(size(MatrizB,2));

MatrizB2 = sign(S.E);
MatrizA2 = eye(size(S.Ed,1));

%% Make complete matrices
Bcompleta=[MatrizB;MatrizB2]';
Acompleta=[MatrizA, zeros(size(MatrizA,1),size(MatrizA2,2));zeros(
    size(MatrizA2,1),size(MatrizA,2)), MatrizA2];
Ccompleta=[MatrizC, zeros(size(MatrizC,1),size(MatrizA2,2))];

%% Load sources
```

```
Aux(:,1) = [size(Acompleta,1)+1:size(Acompleta,1)+size(Sources,1)];
Aux(:,2:3) = Sources;

% Call to compute connection between states
E = relacion_aristas(Bcompleta');
% Add source connection
E = [E;Aux];

%% Path finding

% Build matrix with the cnection between states
matnode=zeros(nx+ne+ns,nx+ne+ns);
for i=1:size(matnode,1)
    orig=find(E(:,2)==i);
    matnode(i,E(orig,1))=1;
end

% Determine the inital states. States connected with disturbances
D=[S.Bp;S.Ed];  %matrix with states and nodes, and the dist.
s=[];
for i=1:size(D,2)
    s=[s find(D(:,i)~=0)];
end
% the initial states. sorted dist 1 -> s(1)

% Find all path
jj=0;
for s=s     % For all states connected with one source
    jj=jj+1;
    %Inicializaci n

    orig=find(matnode(s,:)==1);     %Primeros nodos destino desde s
        %% busca en las columnas en el rengl n de la fuente de
    abastecimiento el valor 1
nodepath=repmat(s,1,length(orig)); %%hace una matriz en donde
    repite el valor de s, una sola vez para rengl n y del
    tama o de orig para columnas

%B squeda de caminos
fi=0;
k=2;
nodepathcycle=[];
while fi==0
    ndd=[];  %nodos destino
    nn=[];  %n mero de nodos destino
    inc=0;
    for i=1:length(s)  %%de 1 hasta el tama o del nodo de la
        fuente (que en esta versi n es 1)
        if s(i)~=0
            ndd=[ndd find(matnode(s(i),:)==1)];  %%se checa en
                el renglon de la fuente, la columna en donde se
                hace 1, es decir el nodo destino
            nn=[nn length(find(matnode(s(i),:)==1))];
        else
            nn=[nn 0];
        end
    end
    nodepath2=[];nd2=[];
    for ii=1:length(nn)     %Para cada nodo destino
        if nn(ii)~=0
            nodepath2=[nodepath2 repmat(nodepath(:,ii),1,nn(ii))
                ];%% si hay varios nodos esto tiene sentido, pero
                 como solo hay una fuente, no hace mucho sentido
            nd2=[nd2 ndd(1+inc:nn(ii)+inc)];
            inc=inc+length(1+inc:nn(ii)+inc);
        else
```

```
                nodepath2=[nodepath2 nodepath(:,ii)];
                nd2=[nd2 0];
            end
        end
        nodepath=nodepath2;
        ndd=nd2;

        nodepath(k,1:length(ndd))=ndd; %%en un segundo rengl n con
           columnas hasta el n mero de nodos destino que hay, se
           escribe el nodo destino
        k=k+1;
        s=ndd;   %Cambio de nodos origen (destinos de la anterior
           iteraci n)
        %Comprobaci n de ciclos
        elim=[];
        for iii=1:size(nodepath,2)
            nozero=find(nodepath(:,iii));
            if sum(nodepath(nozero(end),iii)==nodepath(1:length(
               nozero)-1,iii))>0  %%si la suma del valor que hay en
               el nodepath con la posici n celda(n mero de
               rengl n de posici n final de la matriz nozero, iii)
                %disp('Ciclo en este camino');                %% es
                   igual  al valor que hay en nodepath en la celda
                   (1 hasta el final de nozero -1, iii)
                elim=[elim;iii];
            end
        end
        if ~isempty(elim)  %%existe ciclo si elim tiene algo
           contenido
            for iiii=1:length(elim)
                nodepathcycle(1:size(nodepath(:,elim(iiii)),1),size(
                   nodepathcycle,2)+1)= ...
```

```
                    nodepath(:,elim(iiii));  %%caminos con ciclos de cualquier  ndole
            end
            s(elim)=[];
            nodepath(:,elim)=[];
        end
        %Comprobaci n de fin de b squeda de caminos
        if sum(nodepath(size(nodepath,1),:)==0)==size(nodepath,2)  % %hasta que los ultimos renglones son cero, que significa que ya no se tiene conexi n con ning n otro nodo
            fi=1;
        end
    end

    % Convert states path into actuators path.
    for j=1:size(nodepath,2)
        for i=1:(size(nodepath,1)-2)
            if (nodepath(i+1,j) ~= 0)
                q=find(E(:,1)==nodepath(i+1,j));
                w=E(q,2);
                e=find(w==nodepath(i,j));
                r=q(e);
                act_path(i,j,jj)=E(r(1),3);    %actuator of that connection
            else
                break
            end
        end
    end

end

```

```
clear i ii iii iiii j jj q w e elim orig nodepath nodepath2
   nodepathcycle
clear nozero r s MatrizA MatrizA2 MatrizB MatrizB2 MatrizC Acompleta
clear Bcompleta Ccompleta D E num_est ndd nd2 nn matnode inc fi Aux
```

Appendix C

CSP MatLab Code

```matlab

% Code to write the PEELER input data file, call C++ code to
    contract the
% problem and read the new actuator bounds.
% [min degradation, max degradation, input bounds]
% (maximun degradation, Real acutators bounds, Lamda, Beta,
    Prediction horiz, Actual Reliability, CSP name, number of sources
    , number of controls, caminos)

function [alphamin,alphamax,umin,umax]=ContractorCSP(Alpha,UMIN,UMAX
    ,Lamda,Beta,N,R_0,FILENAME,nd,nu,act_path,Rt_0)

%% Input Data
%   UMIN=[S.umin];  %Real actuator bound
%   UMAX=[S.umax]; %Real actuator bound
%  R_0=Ra(:,k); %Reliability time 0
% FILENAME='peeler.csp';   %PEELER file


%% Write .csp
file=fopen(FILENAME,'w');
```

```

% Constants
buffer = 'Constants';
fprintf(file,'%s \n',buffer);

%R0
for i=1:size(R_0,1)
    buffer = strcat('r',num2str(i),'_0 = ',num2str(R_0(i)),';');
    fprintf(file,'%s \n',buffer);
end

buffer = 'Variables';
fprintf(file,'%s \n',buffer);

%Alpha
buffer = strcat('alpha in [',num2str(Alpha),',',num2str(1),'];');
fprintf(file,'%s \n',buffer);

%U bound
for i=1:size(UMIN,2)
    buffer = strcat('u',num2str(i),' in [',num2str(UMIN(i)),',',
       num2str(UMAX(i)),'];');
    fprintf(file,'%s \n',buffer);
end

%rt
buffer=strcat('rt_',int2str(N-1),';');
fprintf(file,'%s \n',buffer);

%Equations
buffer = 'Constraints';
fprintf(file,'%s \n',buffer);

```

```
%Eq R(u)
for i=1:size(UMIN,2)
    tRa{i} = strcat('(exp(-', num2str(Lamda(i)),'*exp(', num2str(
       Beta(i)),'*u',int2str(i),')*',int2str(N-1),')*r',int2str(i),'
       _0)');
end

%eq camino
for kk=1:nd
    for i=1:size(act_path(:,:,kk),2)
        flag=find(act_path(:,i,kk)~=0);
        if ~isempty(flag)
            buffer=['('];
            for j=1:length(flag)-1
                buffer = strcat(buffer, tRa{(act_path(j,i,kk))},'*')
                    ;
            end
            buffer = strcat(buffer, tRa{(act_path(length(flag),i,kk)
               )},')');
            tRc{kk,i}= buffer;
        else
            break
        end
    end

end

%R para demanda
for kk=1:nd
    buffer=['(1-'];
    for i=1:size(act_path(:,:,kk),2)
        if i==size(act_path(:,:,kk),2)
            break
        else
            flag=find(act_path(:,i+1,kk)~=0);
            if ~isempty(flag)
                buffer = strcat(buffer, '(1-',tRc{kk,i},')*');
            else
                break
            end
        end
    end
    buffer = strcat(buffer, '(1-',tRc{kk,i},'))');
    tRd{kk}= buffer;
end


buffer=[];
for kk=1:nd-1
    buffer = strcat(buffer, tRd{kk},'*');
end
buffer = strcat(buffer, tRd{kk+1});
buffer = strcat(buffer, '-rt_',int2str(N-1),'=0;');
fprintf(file,'%s \n',buffer);

%Bound Reability
buffer = strcat(num2str(Rt_0),'*alpha-rt_',int2str(N-1),'=0;');
fprintf(file,'%s \n',buffer);


buffer = 'end';
fprintf(file,'%s \n',buffer);

fclose('all');

clear i j kk buffer flag
```

```

%% Call C++ function contractor PEELER

%system('C:\MinGW\msys\1.0\home\ivan\ibex\ibex-2.1.17\examples\
   modifsolver C:/MinGW/msys/1.0/home/ivan/ibex/ibex-2.1.17/benchs/
   solver_in.bch 1e-01 0.5 > solver_out.txt' );

system('modifsolver solver_in.bch 1e+03 0.5 > solver_out.txt' );

%% Read the resulting bounds
[t1,t2,t3]=textread('solver_out.txt','%s %s %s','headerlines',1);

% x3 and the bound data, extract the claudators
t3=strrep(t3,'[','');
t3=strrep(t3,']','');

aux=find(t3{1}==';');
alphamin=str2num(t3{1}(1:aux-1));
alphamax=str2num(t3{1}(aux+1:end));

for i=2:nu+1
    aux=find(t3{i}==';');
    umin(i-1)=str2num(t3{i}(1:aux-1));
    umax(i-1)=str2num(t3{i}(aux+1:end));
end


clear i t1 t2 t3 t4
```

Appendix D

CSP C++ IBEX program

```


// modified solver to obtain and provide the outer bounding box of
   the solution set
// STS
// 2014/10/27

#include "ibex.h"
#include <sstream>

using namespace std;
using namespace ibex;


double convert(const char* argname, const char* arg) {
  char* endptr;
  double val = strtod(arg,&endptr);
  if (endptr!=arg+strlen(arg)*sizeof(char)) {
    stringstream s;
    s << "\"" << argname << "\" must be a real number";
    ibex_error(s.str().c_str());
  }
  return val;
}

int main(int argc, char** argv){
  try{

    // check the number of arguments
    if (argc<4) {
      ibex_error("usage: modifsolver filename precision timelimit");
    }

    // load a system of equations
    System sys(argv[1]);
    cout << "Load file " << argv[1] << "." << endl;

        // configure the solver
    double prec        = convert("prec",argv[2]);
    double time_limit = convert("Timelimit",argv[3]);
    DefaultSolver s(sys,prec);
    s.time_limit=time_limit;
    s.trace=0;  // the solutions are not printed when they are found
    cout.precision(12);

    // get the solutions
    vector<IntervalVector> sols=s.solve(sys.box);

    // for each variable, obtain min and max values in the solution
       set
    if (sols.size() >0){
      Matrix my_sol(2,sys.nb_var);
      for (int j=0; j<sys.nb_var; j++)
      {
        my_sol[0][j] = sols[0][j].lb();
        my_sol[1][j] = sols[0][j].ub();
        for (int i=1; i<(sols.size()); i++)
        {
          if (sols[i][j].lb() < my_sol[0][j])
          {
            my_sol[0][j] = sols[i][j].lb();
          }
          if (sols[i][j].ub() > my_sol[1][j])
          {
            my_sol[1][j] = sols[i][j].ub();
          }
        }

        cout << "Variable_" << j << " in [" << my_sol[0][j] << ";"
           << my_sol[1][j] << "]" << endl;
      }
    }

    cout << "Number of solutions = " << sols.size() << endl;
    cout << "CPU time used = " << s.time << "s."<< endl;
    cout << "Number of cells = " << s.nb_cells << endl;
  }
  catch(ibex::SyntaxError& e) {
    cout << e << endl;
  }
}
```

Appendix E

Reliability Computation Code

```

%Funtion to compute the actual reliability of the whole system

function [Ra,Rc,Rd,Rt,Ran,Rcn,Rdn,Rtn]=Reliability_computation(
    act_path,nd,Beta,Lamda,u,Ra0,N)

Ra=exp(-Lamda.*exp(Beta.*abs(u))).*Ra0;     %Relibility for all
    actuators
Ran=exp(-Lamda.*(N-1)).*Ra0;

% eq R_0
for kk=1:nd

    for i=1:size(act_path(:,:,kk),2)
        flag=find(act_path(:,i,kk)~=0);
        if ~isempty(flag)
            Rc(i,kk)= prod(Ra(act_path(1:length(flag),i,kk)));
            Rcn(i,kk)= prod(Ran(act_path(1:length(flag),i,kk)));
        else
            break
        end
    end

end

Rc_aux = 1-Rc;
Rcn_aux = 1-Rcn;

%R para demanda
for kk=1:nd
    flag=find(Rc(:,kk)~=1);
    Rd(kk)=prod(Rc_aux(1:length(flag),kk));
    Rd(kk)=1-Rd(kk);
    Rdn(kk)=prod(Rcn_aux(1:length(flag),kk));
    Rdn(kk)=1-Rdn(kk);
end

%R total
Rt=prod(Rd);
Rtn=prod(Rdn);
```

Bibliography

[1] CAMACHO, E., AND BORDONS, C. *Model Predictive Control.* Springer-Verlag, 2004.

[2] CEMBRANO, G., WELLS, G., QUEVEDO, J., PÉREZ, R., AND ARGELAGUET, R. Optimal control of a water distribution network in a supervisory control system. *Control Engineering Practice 8* (2000), 1177–1188.

[3] GROSSO, J. M. *A Robust Adaptive Model Predictive Control to enhance the Management of Drinking Water Networks subject to Demand Uncertainty and Actuators Degradation.* PhD thesis, CSIC - UPC, 2012.

[4] GUENAB, F., WEBER, P., THEILLIOL, D., AND ZHANG, Y. M. Design of a fault tolerant control system incorporating reliability analysis and dynamic behaviour constraints. *International Journal of Systems Science 42* (2011), 219–233.

[5] OCAMPO-MARTÍNEZ, C., PUIG, V., CEMBRANO, G., AND QUEVEDO, J. Application of predictive control strategies to the management of complex networks in the urban water cycle. *IEEE, Control Systems 33* (2013), 15–41.

[6] OSTFELD, A. Reliability analysis of water distribution systems. *Journal of Hydroinformatics 6* (2004), 281–294.

[7] ROBLES, D., PUIG, V., OCAMPO-MARTINEZ, C., AND GARZA, L. E. Reliable fault-tolerant model predictive control of drinking water networks. *Submitted to: Control Engineering Practice* (2015).

[8] ROSSITER, J. *Model-Based Predictive Control: A Practical Approach.* CRC Press Control Series. CRC Press, 2003.

ETSEIB

[9] SALAZAR, J. C., NEJJARI, F., SARRATE, R., WEBER, P., AND THEILLIOL, D. Reliability importance measures for a health-aware control of drinking water networks. *Submitted to: 3rd Conference on Control and Fault-Tolerant Systems* (2016).

[10] SALAZAR, J. C., WEBER, P., THEILLIOL, D., SARRATE, R., AND NEJJARI, F. Mpc design based on a dbn reliability model: Application to drinking water networks. *9th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes* (2015).

[11] TORNIL-SIN, S., OCAMPO-MARTINEZ, C., PUIG, V., AND ESCOBET, T. Robust fault diagnosis of non-linear systems using interval constraint satisfaction and analytical redundancy relations. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans 44* (2014), 18–29.

[12] WEBER, P., SIMON, C., THEILLIOL, D., AND PUIG, V. Control design for over-actuated systems conditioned by reliability: a drinking water network application. *8th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes 42* (2012), CDROM.

ETSEIB