Delft University of Technology
Faculty of Civil Engineering and Geosciences
Water Resources Management

Master of Science Thesis

Performance assessment of tree-based model predictive control

by

P.M. Stive

Delft, 2011

Supervisory committee

Chairman	prof.dr.ir. N.C. van de Giesen[1]
Member	ir. L. Raso[1]
Member	dr. R.R. Negenborn[2]
Member	dr. D. Schwanenberg[3]

[1]TU Delft, Civil Engineering and Geosciences, Watermanagement
[2]TU Delft, Marine and Transport Technology, Transport Engineering and Logistics
[3]Deltares

Abstract

This research focuses on polder-belt canal systems. More is demanded from these systems every day. Man induced changes, like increasing population density and increasing land value on one hand and climate change in the form of longer dry spells and more extreme precipitation events on the other hand are the main sources. The operation of the structures in these systems plays a critical role in successfully dealing with these challenges. To get the most out of the current system and its structures, operation by humans alone is not enough, they need to be aided by computers. A promising technique is Model Predictive Control (MPC). A control algorithm that uses a model of the system and forecasts of the future disturbances to determine the control actions for the structures, whilst adhering to the constraints of the system. Forecasts are uncertain and are therefore provided in the form of ensemble forecasts that consist of multiple scenarios. MPC uses only one scenario and is thus vulnerable to these uncertainties. Tree-based Model Predictive Control (TBMPC) considers the complete ensemble to determine control actions.

TBMPC has, however, only been tested in theory. Only open loop simulations have been carried out, no continuous closed loop simulations have been done. TBMPC uses the complete ensemble, but to save calculation time reduces it to a tree-shaped representative ensemble with fewer nodes. This means aggregating nodes and scenarios on various points in the ensemble. There are multiple rules that determine which nodes and scenarios are aggregated, however, their optimal setting is not known. Also if TBMPC has more added benefit over MPC on certain system configurations (e.g. configurations with higher discharge or storage capacity) is not known either.

A model is developed to simulate the performance of MPC and TBMPC. It can deal with different precipitation series, forecasts, system configurations and control algorithm parameters.

All simulations have a duration of one year and a one hour time step. The rules that determine which nodes and scenarios are aggregated are investigated first. Transforming the inflow forecast scenarios to cumulative inflow scenarios before

determining which nodes and scenarios to aggregate yields better results. The threshold value is also important as it determines whether or not two scenarios are close enough to each other to be aggregated. Nothing was known about the right value for this parameter. One of the objectives was to be able to fine tune this value to the system configuration. Setting this value as a percentage of the maximum pump capacity of the system works well across different system configurations. The optimal value is 100% of the maximum pump capacity. The scenario reduction algorithm (i.e. the algorithm that creates the tree from the original ensemble) has two parts. First it reduces the number of scenarios in the ensemble to a predefined number and secondly it creates a tree out of the reduced ensemble. No information was available about the right amount of scenarios for the first part. The simulations show that using more than eight to 10 scenarios does not yield any better performance, but only increases calculation time.

To determine if TBMPC is more beneficial on certain system configurations 81 configurations are examined. The performance of MPC with a perfect forecast (i.e. equal to the inflow), MPC and TBMPC is simulated for these configurations. For all configurations TBMPC shows a considerable added benefit, however, there are different reasons for different configurations. For configurations with a high storage and pump capacity the increased performance can be attributed to a more stable water level and a slight improvement in the deviation from set point. For configurations with a low storage and pump capacity the added benefit of TBMPC is seen in a better peak event anticipation.

Overall significant improvements to TBMPC have been realized. It is shown that TBMPC not only works in theory, but provides benefits over MPC in practice for a multitude of configurations as well. TBMPC can now be tuned to the water system configuration it is used on and it can be set to reduce calculation time as much as possible without decreasing the performance.

Preface

This report describes the results of the research I did for my Master Thesis. It has been a challenge, but in the end I'm pleased with the result. One might ask why I've chosen this subject. Let me explain.

The Netherlands is very dependent on its water systems. Drinking water supply, agriculture and transportation are just a few of the beneficiaries of our extensive water systems. Due to the fact that our society and economy are so intertwined with our water systems they are prone to calamities as well. Throughout the last couple of hundred years we had many calamities, but we kept coming up a top every time. Mostly by building large infrastructural works, like canals, dikes and storm surge barriers. We are, however, running out of space, but the challenges will keep coming. One of the ways to deal with these is by making better use of the existing systems. In my opinion control theory and its algorithms are an important component of the solution to these challenges. Model Predictive Control (MPC) is a very promising technique to get the most out of our water systems. MPC, however, requires accurate forecasts, which are not (yet) available so it needs to deal with uncertain forecasts. So far, it does not cope well with these uncertainties. Tree-based model predictive control (TBMPC) is a promising step forward in dealing with these uncertainties. Analyzing the performance of TBMPC and making it ready for practical implementation is a necessary step in the direction of widespread use of these algorithms. I wanted to contribute to the development of these algorithms.

I would like to use this space to thank some people who have worked very closely with me during this research as well. First of all Luciano Raso for his continuous support, patience and ideas. Secondly Rudy Negenborn and Dirk Schwanenberg for their critical remarks every time I gave them an update on my progress. I also want to thank Nick van de Giesen for allowing me the time to get familiar with this new subject and not only applying my current knowledge but learning a lot of new things as well.

I cannot forget my parents who have always supported me throughout my studies, without ever doubting me. Thank you. Lastly I want to thank my girlfriend, Ivonne, for her unconditional support and love.

Philip
June 2011

Contents

Chapter 1

Problem analysis

1.1 Introduction

In most non-arid regions water systems are designed to drain water out of the system into an adjacent river or sea. Canals and structures are used to store and transport water; within and out of the system. Sluice gates and pumps that discharge water out of the system are the most important structures for the system, since these determine the maximum drainage capacity. These structures are designed to handle regular storm events. A heavy storm event that causes a disturbance (e.g. inflow) larger than the drainage capacity raises the water level in the system above the target water level (i.e. desired water level) and can lead to inundation and consequently damage to the inundated areas.

If the occurrence of a heavy storm event can be forecasted, anticipatory actions can be taken, like lowering the water levels in the water system in advance. This creates extra storage capacity that can be utilized during the heavy storm event. Water levels will not rise as high and inundation might be prevented. There are, however, boundaries to these anticipatory actions. Water levels cannot be lowered too much to avoid collapse of dikes. Also, changes to structures' settings need to be minimized to avoid wear and tear to the structures and strong transition waves in the canals.

At the time of writing the management of water levels within these boundaries is often performed by an operator, who needs to have extensive experience with the system. The operator needs to decide on the best control action based on the current state of the system, the precipitation forecast and the limited capacity of the structures. However this becomes an increasingly more difficult task as man induced changes to land use increase the value of land and storm events tend to become more intense [11]. One way to deal with this difficult task is to take structural measures, for example increasing the drainage capacity of the water

system. However structural measures often take a long time to become effective, are expensive and are spatially demanding. In today's world time, money and space are scarce. Another solution is to implement a (better) control algorithm. The control algorithm has the same objective as the operator, keeping the water level in the system as close to target level as possible, given the limited capacity of the structures. To satisfy this objective multiple control algorithms have been designed over the past few decades.

Feedback control is the simplest realization of a control algorithm. The state of the system (e.g. water level) is measured and compared with the desired state of the system. The difference is fed back into the control algorithm and a corrective control action will be taken. The disturbances on the system are only indirectly taken into account.

With feedforward control these disturbances are a direct input for the control algorithm. The amount of the disturbance is measured and counter balanced with a control action to minimize the effect on the water system state. The predicted disturbance for the next time step into, for example, a reservoir is equaled by a discharge of the same amount out of the reservoir. As the feedforward algorithm needs to predict disturbances, uncertainty within the prediction limits the capabilities of feedforward control. Feedforward control therefore is often combined with feedback control to compensate for this uncertainty.

The former control algorithm is not able to deal with a disturbance that is larger than the control capacity (e.g. drainage capacity). To deal with such a disturbance the control action needs to be started several time steps earlier, to create storage to prevent the water level to exceed the target water level. Such control algorithms are referred to as Model Predictive Control, which is defined as:

> Model Predictive Control is the range of control methods which make an explicit use of a model of the system to obtain the control signal by minimizing an objective function on a receding finite horizon whilst complying to the constraints of the system.

In the case of a water system the model is a combination of a hydraulic model that calculates the water levels based on the future in- and outflow and the current state of the system and a model defining the constraints within the system, like water levels (soft constraints) and pump capacities (hard constraints). The objective function defines penalties for, for example, water level, change of water level and structure setting changes. Each time step (e.g. every hour) the problem is optimized (i.e. the objective function is minimized) for a finite optimization horizon with the information available at that moment, and each time step a (or multiple) control action(s) is(are) determined. Since this is done every time step, with the information available at that moment, this is called real-time control

(RTC). Another characteristic of MPC is the receding strategy. At each time step the problem is optimized and control actions for the optimization horizon are calculated, but only the control action for the first time step is executed (sometimes more than one, but always less than calculated), the horizon is moved one time step towards the future and the whole process is repeated. MPC thus combines the benefits of feedforward and feedback control and adds the receding strategy and the ability to deal with constraints of the system.

MPC makes use of a model, measurements and forecasts, which all carry uncertainty. In the case of water systems the largest part of the total uncertainty is usually introduced by the forecast. Current implementations of MPC on water systems use a deterministic forecast of the future disturbances. However a deterministic format forces the forecaster to suppress information and judgment about uncertainty. A deterministic forecast may create the illusion of certainty in a user's mind, which can lead the user to suboptimal action [6]. In other words not explicitly taking the uncertainty within the forecast into account can lead to a suboptimal control action, given the information available in the forecast. It does not mean however that by explicitly taking uncertainty into account the control action that is taken will always be optimal in hindsight. This is only possible with a perfect deterministic forecast, where the forecasted disturbance is exactly equal to the actual disturbance.

The concept of Tree-based Model Predictive Control (TBMPC) [9] does take uncertainty into account by using an ensemble forecast. An ensemble forecast consists of a representative sample of the possible future states of a dynamic system, called scenarios or ensemble members. To decide on the optimal control action(s), given the uncertainty, a risk based approach needs to be taken. Risk is defined as the statistical expectation of costs. By evaluating control actions against all possible effects, given the scenarios in the ensemble, and multiplying these by their probability of occurrence the lowest expected costs can be determined. Again only the control action for the first time step is executed. It has been shown that such a risk based approach is more cost effective than decisions made in a deterministic framework [10]. The calculation time for TBMPC is longer than for MPC. To minimize the calculation time a scenario reduction algorithm is used. This algorithm reduces the original ensemble to a representative ensemble with fewer scenarios. The performance of TBMPC relies heavily on the scenario reduction algorithm, since with each elimination of a scenario, information is lost. The remaining scenarios are reduced to a tree structure to reduce the calculation time even more. On every time step the difference between the values of the remaining scenarios are compared and if they are below a predefined threshold value the scenarios will be aggregated until this point.

1.2 Problem definition

In [9] TBMPC is compared to MPC over the optimization horizon, so only the expected water levels are compared. It is not yet known if TBMPC also performs better in a closed loop simulation, in other words if the water levels that occur under the regime of TBMPC yield lower costs than those under the regime of MPC. It is expected that by not suppressing the information available in the forecast TBMPC does perform better in a closed loop simulation.

To minimize the calculation time of TBMPC the number of scenarios in the ensemble is reduced and the scenarios are put into a tree structure. The construction of the tree is very dependent on the threshold. No information is, however, available on the influence of the construction of the tree on the performance of TBMPC and if the threshold can be related to any physical characteristics of the system or the forecast.

No analysis is available on the performance of TBMPC in relation to the marginal information loss that occurs with the deletion of an extra scenario.

TBMPC has only been compared to MPC on one water system. The influence of the water system characteristics on the performance is thus still unknown. It is expected that, for example, the added benefit of TBMPC over MPC will be lower on systems with a relatively high pump and storage capacity, because such a system does not need that much information from the forecast to perform well, it can rely on its feedback capabilities.

To summarize the following problems are defined:

- The performance difference between TBMPC and MPC in a closed loop simulation has not yet been investigated.
- The influence of the tree construction on the performance of TBMPC and how the threshold needs to be defined are unknown.
- No analysis is available on the performance of TBMPC in relation to the marginal information loss that occurs with the deletion of an extra scenario.
- TBMPC has only been shown to be an improvement over MPC on one water system configuration. How both algorithms perform as a function of water system characteristics is unknown.

1.3 Objectives

MPC algorithms make use of a simplified model of the water system, in other words have a smaller state than the actual system and thus will always provide suboptimal operation. Computing time is limiting the use of more complex models, this limit will increase as technology advances, but will always continue to

exist. Optimizing the performance of a control algorithm will thus lead to the best possible control actions obtainable within the boundaries of computing time. It is in this light we want to assess and optimize the performance of TBMPC.

The problem definition states four aspects of TBMPC that need to be examined more closely. The objective of this research is to give insights in the relation between the performance of TBMPC and these aspects, more precisely:

- Compare the performance of TBMPC and MPC for a closed loop simulation.
- Evaluate the influence of the scenario tree construction algorithm and improve it if possible. The effect of preprocessing the scenarios and the influence of the aggregation threshold will be investigated to be able to fine-tune it to the water system it is applied on.
- Provide an analysis on the marginal information loss that occurs with the deletion of an extra scenario. This will give insight on the trade-off between performance and calculation time.
- Evaluate the influence of different water system characteristics. For which water systems TBMPC is more beneficial can be determined in this way.

These experiments will make the performance of a control algorithm more understandable, a problem currently facing water boards. There is no knowledge [within water boards] about the performance of a particular weather forecasting system, for the given water system, let alone information about the performance of the decision rules that depend on these forecasts [14]. With better insight and understanding, the confidence water managers have in control algorithms will rise as well.

The simulations will be done offline, so computing time will not be a limiting factor.

1.4 Boundary conditions

In general the following boundary conditions apply to the research:

- Only water quantity is controlled. Water quality is left out. Although taking water quality into account will most probably be very interesting it will take a lot more time to create and setup the models, make these a lot more complex and increase calculation time significantly. It is expected that only a water quantity model can provide interesting insights in the performance of control algorithms as well and provide the necessary results to give adequate answers to the questions posed for this research.

- The control algorithm is tree-based model predictive control. TBMPC is only compared to MPC, not to other control algorithms, because it will only be implemented if it is an improvement over the best algorithm used to date (MPC, see e.g. [12]). See for a comparison between other previously mentioned control algorithms [16].

- The research assumes the belt canals of a polder-belt canal system as the controllable system as this research is performed from a water management perspective. This gives a boundary for the system characteristics that need to be tested.

- The objective function solely consists of quantifiable objectives. The only objective is thus to keep the costs as low as possible. Other and conflicting interests, which are normally present in a water system due to the large amount of stakeholders, are not dealt with. The latter is politics, not control theory and is an a priori fact for the control algorithm.

- Performance is defined as the value of the objective function over the simulation horizon. The value of the objective function has no monetary unit, but is used to be able to relatively quantify the performance of TBMPC versus MPC.

Chapter 2

Model description

2.1 Introduction

To fulfill the objectives set forth in Chapter 1 a model of a water system with adjustable input and control algorithm is required, because running tests and simulations on a real system is not financially feasible. The model used in this research can be divided into two major components: the non-controllable and controllable part. This division has been schematized in Figure 2.2.

The non-controllable system represents the precipitation and the rainfall runoff processes. The controllable system represents the belt canal system of a polder that consists of storage and conveyance canals and discharges (i.e. control flow) to an adjacent sea or river. These pumps are controlled by the control algorithm, TBMPC in this case, that tries to keep the water level in the controllable system as close as possible to a certain target water level (h_{set}).

In the next paragraphs each component of this model is described in more detail.

Five definitions of certain periods or temporal parameters (simulation horizon, optimization horizon, information-prediction horizon, control time step and uncertainty function) are used throughout this chapter and the remainder of this research. These horizons can be confusing at first, so in Figure 2.1 a visual representation of these horizons is given.

2.2 Non-controllable system

The non-controllable system represents the precipitation process over the area that contributes to the inflow into the controllable system. In the case of a polder-belt canal system this is equal to the precipitation on the polder surface area that generates runoff to the drainage canals (which is pumped to the belt

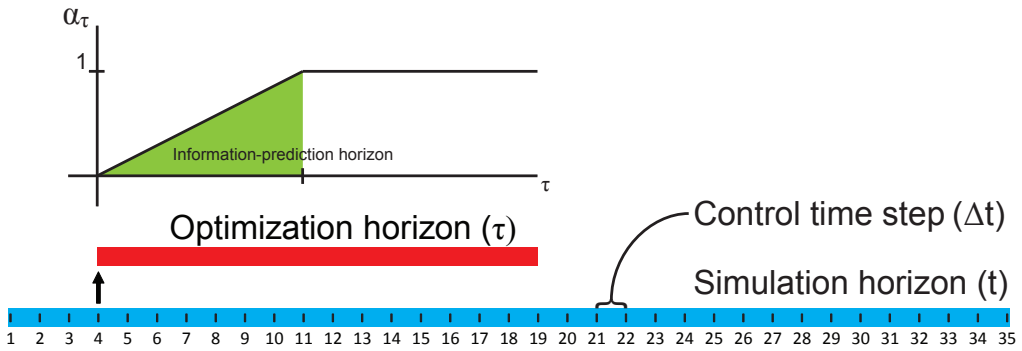


Figure 2.1: In this example the actual moment is $t = 4$. All values are for this example only. The simulation horizon is 34 time steps. The optimization horizon is 15 time steps. The optimization horizon begins at the actual moment (at $t = 4$ or $\tau = 1$). The uncertainty function (α_τ) starts at $\tau = 1$ and ends at $\tau = 16$. The information-prediction horizon is 7 time steps.

canals) and precipitation on the belt canals' surface area. These components are lumped together to create the inflow to the controllable system.

The precipitation time series used in this research are generated from a distribution derived from historical records. This has two reasons. The first is that by implementing a rain generator into the model, the simulations can easily be repeated with time series with varying characteristics (i.e. from different historical records). The second reason concerns uncertainty. The precipitation time series are generated from a distribution and based on this same distribution an ensemble forecast can be created (see for further details paragraph 2.2.2). This creates a comprehensive forecast (where the probability that the observed rainfall is present in the forecast approaches 1) and eliminates the uncertainty between the forecast and the actual precipitation. The influence of forecast characteristics can be investigated only in this way, because the performance is not influenced by the unknown (varying) uncertainty between the forecast model and the observed precipitation.

2.2.1 Precipitation generator

The precipitation generator gives an amount of precipitation for each time step within the simulation horizon (i.e. the period on which the control algorithm is active). Precipitation has two distinct characteristics which need to be present in the precipitation generator as well: (1) the sequence of occurrence and (2) the amount of precipitation. These characteristics are implemented under the following assumptions:

- The sequence of the occurrence of precipitation can be represented by a Markov chain.

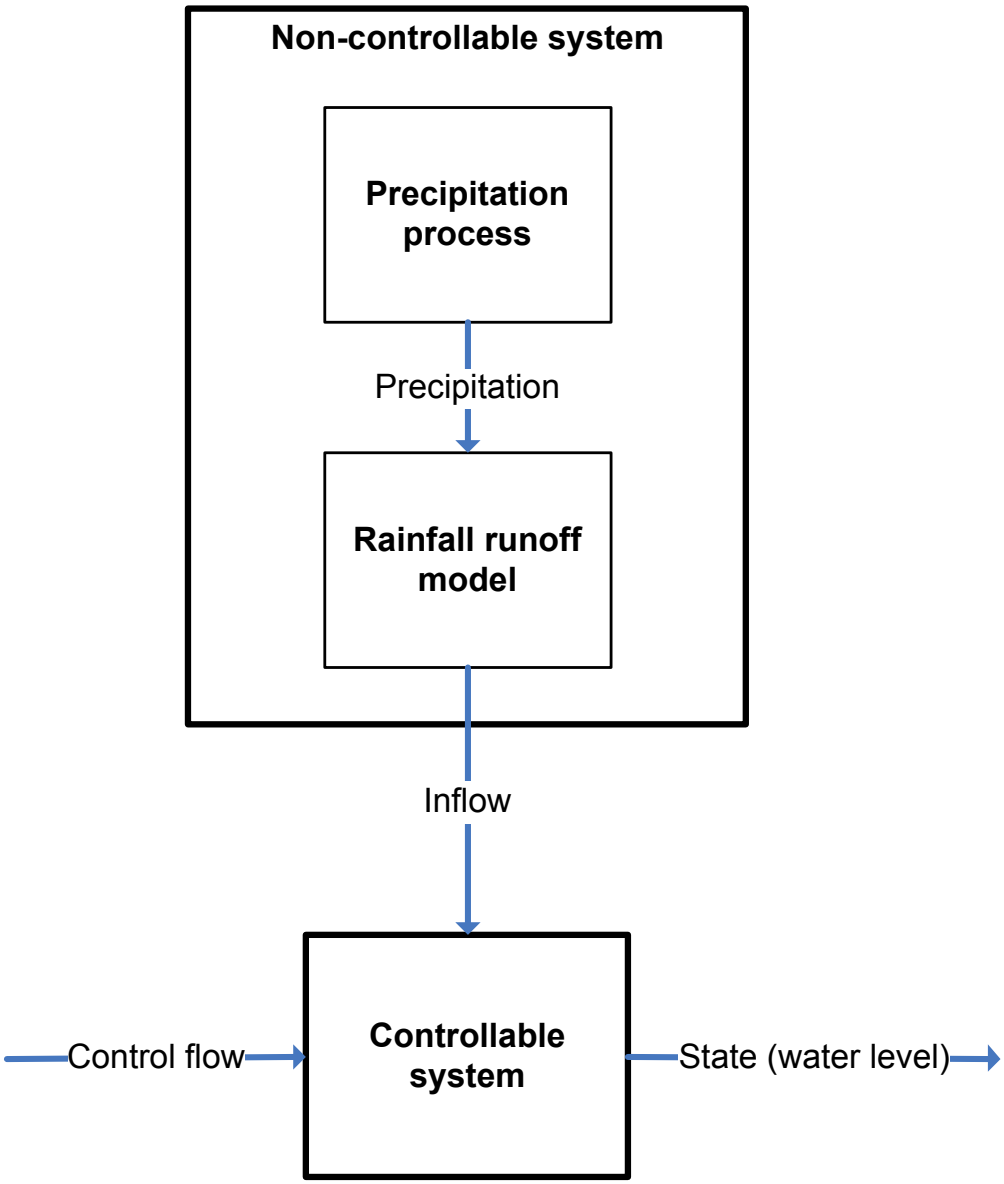


Figure 2.2: Simple schematic division of the model used in this research.

- The amount of precipitation can be fitted to an exponential distribution.

These assumptions are widespread and established for the simulation of precipitation (see e.g. [5] and [1]). The amount of precipitation (P_t) is calculated by multiplying the value for the occurrence of precipitation (B_t) with the value from the exponential distribution (E_t), or:

$$P_t = B_t \times E_t \tag{2.1}$$

Every time step, independently a value for B_t and E_t is generated. This is done because each E_t value is independent from the previous value and the probability of the value being zero is nearly zero (theoretically it is zero, because it is a continuous distribution, where the probability of a single value is always zero). The sequence of the occurrence of precipitation is represented by a first-order two state Markov chain. This means that for each time step only two states are possible, 0 or 1 in this case, where 0 denotes a dry time step and 1 a wet time step, or:

$$B_t = \begin{Bmatrix} 1 \\ 0 \end{Bmatrix} \tag{2.2}$$

Since this is a first-order Markov chain B_{t+1} is only dependent on B_t. This dependency is described by the transition matrix:

$$C = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix} \tag{2.3}$$

The values in the transition matrix represent the following probabilities:

- C_{11} is the probability that B_{t+1} equals 1 given that B_t equals 1.
- C_{12} is the probability that B_{t+1} equals 0 given that B_t equals 1.
- C_{21} is the probability that B_{t+1} equals 1 given that B_t equals 0.
- C_{22} is the probability that B_{t+1} equals 0 given that B_t equals 0.

For every time step an amount of precipitation is generated from an exponential distribution (E_t). The probability density function is given by:

$$f(x;\lambda) = \begin{cases} \lambda e^{-\lambda x}, x \geq 0 \\ 0, x \leq 0 \end{cases} \quad \text{where } \lambda = \frac{1}{\mu} \tag{2.4}$$

where x is the random variable, λ is the rate parameter (reciprocal of the scale parameter) and μ is the mean of the distribution.

2.2.2 Forecast generator

The forecast generator is similar to the precipitation generator as it derives the amount of precipitation from the same distribution (equation 2.4) and multiplies the occurrence value ($B_{t,n}$) with the exponential value ($E_{t,n}$), with the only difference that it does this in a probabilistic manner, in other words it generates scenarios (total number of generated scenarios is N_{ensemble} and n is the index of the scenario). For each time step and scenario an occurrence value is generated based on the dependency described by the transition matrix (C) and a value for the amount of precipitation is taken from the exponential distribution.

The forecast generator needs to generate a distribution with the same characteristics as the precipitation generator. At each step as many values for the precipitation as there are scenarios need to be generated. For each scenario the value needs to be generated from an exponential distribution. This can be done sequentially, but can be done at once as well by using the gamma distribution. The gamma distribution is a two-parameter frequency distribution and the probability density function is given by:

$$f(x) = x^{k-1} \frac{e^{\frac{-x}{\theta}}}{\theta^k \Gamma(k)} \text{ and } k, \theta > 0 \tag{2.5}$$

where x is the random variable, θ is the scale parameter, k is the shape parameter, Γ the usual gamma function and $f(x) = 0$ for $x < 0$ [13]. The lower bound is thus 0 and it is unlimited on the right. If k is an integer then the distribution is equal to the sum of k independent exponentially distributed random variables, with all of these having a mean equal to θ. The gamma distribution is conditioned on the actual amount of rainfall on each time step within the information-prediction horizon (T_{ip}). T_{ip} can be defined as the time span from the actual moment until the moment where the conditional distribution of future events, conditional to all actual information, becomes the same as the marginal (climatic) distribution of events [17]. To condition the gamma distribution the shape (k) and scale (θ) parameter need to be calculated for each time step. This is done by using the mean and variance definitions for the gamma distribution (2.6). The mean (μ_τ) is set to a weighted average of the generated actual precipitation (P_t) and the mean climatic precipitation (E_{clim}). The variance (σ_τ^2) is a fraction of the climatic variance (σ_{clim}^2). This weighted average and fraction are a function of the uncertainty function (2.8), which is close to zero at the actual moment (moment of running the optimization, t) and equal to 1 after T_{ip}.

$$\begin{aligned} \mu_\tau &= k_\tau \times \theta_\tau \\ \sigma_\tau^2 &= k_\tau \times \theta_\tau^2 \end{aligned} \tag{2.6}$$

$$\begin{aligned} \mu_\tau &= (1 - \alpha_\tau) P_{t+\tau} + \alpha_\tau \times E_{\text{clim}} \\ \sigma_\tau^2 &= \alpha_\tau \times \sigma_{\text{clim}}^2 \end{aligned} \tag{2.7}$$

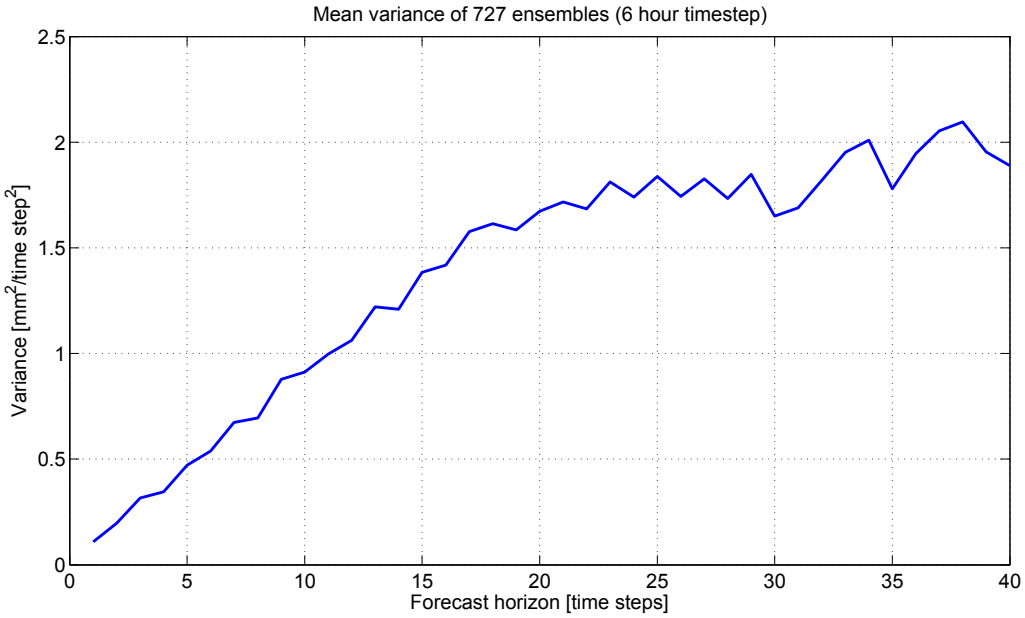


Figure 2.3: Evolution of the variance of 727 ensemble forecasts over the forecast horizon. Ensembles provided by the KNMI.

where

$$\alpha_\tau = \begin{cases} \frac{\tau}{T_{\text{ip}}}, \text{ for } \tau \leq T_{\text{ip}} \\ 1, \text{ for } \tau > T_{\text{ip}} \end{cases} \tag{2.8}$$

P_t represents the actual precipitation amount on time t, E_{clim} is the mean climatic rainfall amount, τ is the time step indicator in the optimization horizon and σ_{clim} is the standard deviation of the climatic rainfall amount. Since the rainfall amount is represented by an exponential distribution the mean is equal to the standard deviation. The shape of the uncertainty function is based on the shape of the variance of real ensemble forecasts. Figure 2.3 shows the variance of 727 ensemble forecasts in time. The ensemble forecasts are provided by the Royal Dutch Meteorological Institute (KNMI), are produced every 12 hours (i.e. update frequency), have a 6 hour time step and consist of 50 ensemble members (i.e. scenarios). The variance starts nearly at 0, increases linearly until a certain time (T_{ip}), flattens off and stays more or less stable. Only the pattern of the variance can be used for the uncertainty function and the variance used in equation 2.7, not the value of the variance itself, because the KNMI variance is based on six hour rainfall data, while here one hour rainfall data is used.

By equating 2.6 and 2.7 given 2.8 for each time step in the optimization horizon (τ) the shape and scale parameter for the gamma distribution can be calculated:

$$\theta_\tau = \frac{\alpha_\tau \times E_{\text{clim}}^2}{(1-\alpha_\tau)\, P_{t+\tau} + \alpha_\tau \times E_{\text{clim}}} \tag{2.9}$$

Measurement station	260 De Bilt
Period	1^{st} of January 2009 - 31^{st} of December 2009
Time step	1 hour
E_{clim}	0.7928 mm/hour
C	$\begin{bmatrix} 0.6755 & 0.3245 \\ 0.0409 & 0.9591 \end{bmatrix}$
q	$\begin{bmatrix} 0.1119 & 0.8881 \end{bmatrix}$

Table 2.1: Historic precipitation time series specifications.

$$k_\tau = \frac{(1-\alpha_\tau)\, P_{t+\tau} + \alpha_\tau \times E_{\text{clim}}}{\theta} \tag{2.10}$$

2.2.3 Calibration of the precipitation and forecast generator

The precipitation generator and the forecast generator can easily be calibrated to any desirable historical record by only defining two parameters: (1) the mean of the exponential distribution and (2) the transition matrix. Appendix C describes how these can be derived from a historical record and how the specifications in Table 2.1 were established.

2.2.4 Precipitation as a function of the area size

The historical records are values of a measurement station, in other words point measurements. These precipitation amounts can not be directly translated to the average precipitation amount for a large area (i.e. larger than 1 ha). The pattern and the amount will change according to the size of the area.

Precipitation can be categorized by two main types, small scale convective storms or large scale frontal storms. The first often has a higher intensity, but shorter duration. Extremes in the historical measurements could originate from such convective storms. The size of the area has an influence on how the historical pattern of the point measurements can be translated to the pattern of precipitation for an area with a certain size. A larger area means that a (convective) storm event will not always cover the complete area resulting in a reduced amount of precipitation for the complete area. A small area will sometimes even be completely missed by a (convective) storm event and sometimes be completely covered by it. This yields a more extreme precipitation pattern for smaller areas and a less extreme pattern for larger areas. The reduced amount of precipitation for a larger area will not be dealt with in this research. It is assumed that the amount is already calibrated on the area size. The accurate amount of precipitation for an area is dependent on so many factors, that trying to calibrate the amount would drive this research away from its generic framework.

The adjusted pattern, however, can be placed in a more general view, as the pattern will always be less extreme when increasing the area under the same climatic conditions. A moving average filter is applied to the precipitation series, where the window size is dependent on the area size:

$$P_t^{\mathrm{MA}} = \frac{\sum_{i=0}^{n-1} P_{t-i}}{n} \tag{2.11}$$

where P_t^{MA} is the precipitation amount after the moving average has been applied, P_t the precipitation amount from the precipitation distribution and n the window size. The latter is related to the area size through a natural logarithm function:

$$n = \ln\left(A_{\mathrm{system}}\right) + 1 \tag{2.12}$$

where A_{system} is the surface area of the water system in hectares.

2.2.5 Rainfall runoff model

The rainfall runoff process is very complex, however, rainfall runoff models don't necessarily need to be. Often simple models provide as good predictions as much more complex ones (see for example [7], [4] and [3]). A simple rainfall runoff model will suffice here, as long as it reasonably describes the rainfall runoff pattern in the type of system under investigation and is kept constant throughout the research.

Nearly all the precipitation drains through subsurface drains or overland flow to the drainage canals and is pumped into the controllable system. This pumping is considered to be instantaneous, but the drainage process can cause some smoothing of the runoff compared to the rainfall, due to storage in the soil. An autoregressive exogenous (ARX) rainfall runoff model will be used here:

$$R_t = \sum_{i=1}^{2} \alpha_i R_{t-i} + \beta P_{t-T_{\mathrm{L}}}^{\mathrm{MA}} \text{ where } \alpha_1 + \alpha_2 + \beta = 1 \tag{2.13}$$

where R_t equals the runoff, P_t the precipitation amount and α_i and β are the lumped parameters to represent the storage capability of the catchment. The parameters will be 0.35 for α_1, 0.15 for α_2 and 0.5 for β to represent a minor smoothing process, since soils are shallow, often have a high moisture content and greenhouses (often present in Dutch polders) cause immediate fast runoff, without any smoothening.

There can be a delay between the time of precipitation and the start of the runoff process, for example if the greenhouses in the system have some sort of storage capability. The pump process from the drainage canals to the belt canals is considered to be instantaneous, but if this is not the case the runoff is not immediately the inflow to the controllable system. Both situations can be represented by introducing a lead time (T_{L}) in the rainfall runoff model.

Parameter	Unit
E_{clim}	mm/hour
C	-
T_{ip}	hour
N_{ensemble}	-
α_τ	-
A_{system}	ha
α_1	-
α_2	-
β	-
T_{L}	hour

Table 2.2: Adjustable parameters of non-controllable system.

2.2.6 Adjustable parameters

Within the non-controllable system the parameters that can be changed are summarized in Table 2.2.

2.3 Controllable system

The controllable system consists of two major components: (1) the model of the water system it is representing and (2) the control algorithm that controls the structures present in the water system. The purpose of the control algorithm is to keep the water level in the water system as close to the target water level as possible by controlling the structures present in the system. The algorithm uses an objective function that translates water levels and pump amounts into costs. By minimizing this objective function the water level is kept as close to the target water level as possible. Two algorithms are tested during this research, MPC and TBMPC. Both use the forecasts generated with the forecast generator to determine the control action for the next time step(s). MPC uses one random scenario from the forecast and TBMPC uses multiple scenarios in a risk based framework.

The water system that is used throughout this research is first described in paragraph 2.3.1, followed (in paragraph 2.3.2) by the objective function that is closely related to the water system and is used by both MPC and TBMPC. These two algorithms will be addressed in paragraphs 2.3.3 and 2.3.4 respectively.

2.3.1 Water system model

A water system can be defined as a set of water bodies with their conveyance and regulating structures. These components are linked through natural and artificial processes. In this research the water system represents the belt canal system of

a polder-belt canal system. The belt canals have both a storage and conveyance function, but because these canals are wide and well interconnected they can be modelled as a reservoir system. The reservoir has in- and outflow components that originate from pumping.

The inflow (R_t) is the rainfall runoff into the canals in the polder that is immediately discharged by pumps on the belt canal summed with the direct precipitation on the belt canals' surface area. The outflow (u_t) equals the discharge of the pumps out of the belt canals (constrained by the maximum discharge of the pumps, $u_{\max}$) into an adjacent river or sea, assumed to have an infinite volume or discharge capacity. This assumption can be made because belt canal pumps in The Netherlands often discharge to the sea or large rivers (e.g. Delfland discharges to the 'Nieuwe Waterweg' and Rijnland to the North Sea). The model used is thus given by:

$$h_{t+1} = h_t + \frac{\Delta t}{A_{\text{storage}}} (R_t - u_t) \tag{2.14}$$

where h is the water level, Δt the control time step, A_{storage} the storage area (i.e. surface area of belt canals) and t the index of the time step.

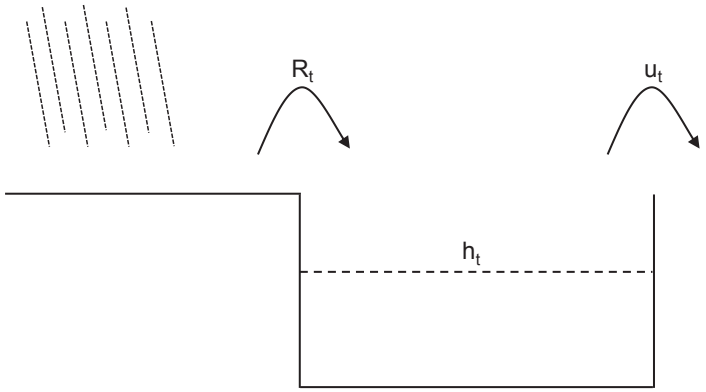


Figure 2.4: Water system overview.

Adjustable parameters

The water system has three variable parameters, the first one being the control time step and the other two are storage area and pump capacity. The latter are expressed in a ratio. The storage area reflects the capacity of the system to deal with extreme events through storage in the belt canals. When this ratio increases the total amount of precipitation on the system will have a smaller effect on the water level in the belt canals, or the preventive lowering of the water level in the belt canals creates a larger storage volume. A generic form of expressing this dynamic is the ratio between the storage area (m^2) and the total system area

Parameter	Unit
Δt	hour
$\frac{A_{\text{storage}}}{A_{\text{system}}}$	-
$\frac{u_{\max}}{A_{\text{system}}}$	m/hour

Table 2.3: Adjustable parameters of the water system.

(m^2). The pump capacity characterizes the capability of the system to deal with water level fluctuations in the belt canals through discharge. Upon a heavy rain event and when the water from the polder is immediately pumped to the belt canals, an inflow to the belt canals that causes an increase in water level can be equaled by a pump flow of the same magnitude to mitigate the water level rise. In other words it is the capacity of the system to deal with extreme events through discharge. The discharge capacity of the system is expressed in the ratio between pump capacity (m^3/hour) and system area (m^2). The three described parameters are summarized in Table 2.3.

2.3.2 Objective function

The objective function translates the state(s) of the system into costs. It penalizes states of the water system and structures with respect to certain interests. These interests can be weighted and summed to obtain an objective function that represents all these interests. Conflicting interests are normally present within a water system, these include ecological, transportation (shipping), flooding and recreational interests for example. These interests thus have a large influence on the shape of the objective function. As the objective function is used to calculate the performance of a control algorithm, the performance is dependent on it as well.

To be able to aggregate all interests into one objective function, all interests need to be quantified and expressed in the same units. This is done by using the costs associated with the state of the system. These costs can be split in two categories: (1) subjective and (2) objective. For example ecological and recreational costs depend on how much these interests are worth to the stakeholders, which is subjective. On the other hand transportation costs, for example, do not have this subjective value estimation. If the water drops below a certain threshold, a certain number of ships will not be able to sail, which results in a certain amount of lost income to the parties involved in the shipment. As this research aims to provide generic relations, subjective and often widely varying costs (per system) will not be taken into account.

Within the objective costs another distinction can be made: (1) high costs with a low probability and a short duration and (2) low costs with a high prob-

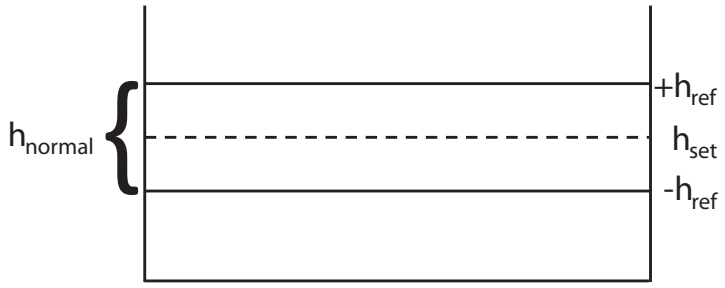


Figure 2.5: Schematic overview of set point and normal operating band.

ability and a longer duration. Flooding costs are a good example of the first category and shipping costs fall under the second.

The objective function is reduced to a cost function with four components. The first component in the cost function represents the wear and tear of the structures. A controllable system will always have structures and maintenance costs will always tried to be minimized. The second component represents other low costs with a high probability, these occur with water levels varying around set point (i.e. target level, h_{set}), such as damage to houseboats or shipping for example. These costs should be in the same order of magnitude as maintenance costs, since both are part of the normal operating expenses of the controllable system. The third and fourth component represent high costs with a low probability and short duration, flooding costs when water levels are above the normal operating levels ($+h_{\text{ref}}$) and collapse of embankments when water levels are below the normal operating levels ($-h_{\text{ref}}$) are examples of such costs.

In Figure 2.5 a schematic overview of the set point (h_{set}) and the range around it is presented.

The four components are equally weighted and can thus be combined into one step-cost function (2.15). Costs are proportional to the penalty and the exponent assigned to the different components in the cost function. The components are the change in pumping (Δu), the distance from the water level set point (h) and the upward (h_2) and downward (h_3) exceedance of normal operating levels. If w denotes the penalty and τ the time over which the costs are evaluated, the cost function per scenario is given by:

$$\sum_{\tau=1}^{T_o} g_\tau^n \text{ where}$$

$$g_\tau^n = w_{\text{u}} \left(u_{\tau-1}^n - u_{\tau-2}^n\right)^{E_{\text{u}}} + w_{\text{h}} \left(h_\tau^n - h_{\text{set}}\right)^{E_{\text{h}}} + ...$$

$$w_{\text{h}_2} \max\left(h_\tau^n - h_{\text{ref}}, 0\right)^{E_{\text{h}_2}} + w_{\text{h}_3} \max\left(-h_{\text{ref}}^n - h_\tau, 0\right)^{E_{\text{h}_3}} \tag{2.15}$$

For TBMPC the cost function J for all scenarios becomes:

$$J = \sum_{n=1}^{N_{\text{red}}} \sum_{\tau=1}^{T_o} \phi_n g_{\tau,n} \tag{2.16}$$

where ϕ_n denotes the probability of occurrence, n is the index of the scenario and N_{red} is total number of scenarios within the reduced ensemble.

2.3.3 Model Predictive Control (MPC)

During this research TBMPC is compared to MPC, which provides a solution to the problem of optimal control in a deterministic framework, in other words it uses only one scenario. The forecast generator gives 50 scenarios at every time step. To give an accurate representation of the performance of MPC one of these 50 scenarios is randomly chosen every time step and used for MPC. With a real ensemble (i.e. generated by the ECMWF) the scenario used for MPC will most likely be the 'operational run'. This is the scenario that originates from the model run at full resolution with the most likely initial conditions. It does not mean that this run has any specific statistical properties compared to the other ensemble members, like it being the median of the ensemble for example. After it is generated it is merely one of the 50 ensemble members and it can very well be the one that predicts the highest amount of rainfall or is the only member that says it will stay dry. The forecast generator creates 50 equally likely ensemble members, there is no information available to indicate on scenario as the 'operational run'. To pick, at random, one of the 50 scenarios most closely resembles the results achieved in real life, given the forecast generator used in this research. The performance assessment is done over a year long period (i.e. simulation horizon) with hourly values and an optimization time step of six hours (see paragraph 3.4) so a scenario will be chosen 1460 times over the simulation horizon.

2.3.4 Tree-based Model Predictive Control (TBMPC)

Model Predictive Control (MPC) provides a solution to the problem of optimal control in a deterministic framework. However certain components of the problem carry large uncertainties and therefore can be represented in a stochastic framework. Generally speaking there are four sources of uncertainty when controlling a water system: (1) measurements, (2) models, (3) actions and (4) forecasts. The dominant source of uncertainty, given the system presented in the introduction of this chapter, is the precipitation forecast. Meteorological institutions provide this forecast in the form of an ensemble forecast, which consists of a number of alternate predictions (i.e. scenarios) for the same forecast period. Through the rainfall runoff model this precipitation forecast translates into an inflow forecast. Tree-based Model Predictive Control (TBMPC) has the ability to give a robust

solution to the problem of optimal control given such a stochastic representation of the future inflow. This problem can mathematically be defined as:

$$\begin{aligned} \arg\min_{u_\tau^n} J\left(h_\tau^n; u_\tau^n\right) \ \tau = t, ..., t + T_\mathrm{o} \\ h_{\tau+1}^n = f\left(h_\tau^n; u_\tau^n; R_\tau^n\right) \\ c\left(h_\tau^n; u_\tau^n\right) \leq 0 \end{aligned} \tag{2.17}$$

where J is the cost function to be minimized, h the state vector per scenario, u the control vector per scenario, R the disturbance vector per scenario, f the model of the system, c the inequality constraints of the system, n the index of the scenario, t the actual moment, τ the index of the time step in the optimization horizon and T_o the optimization horizon. Every time step (at an interval equal to the optimization time step) J is minimized (one optimization) and control actions are determined. There is no dependency between two optimizations, the optimization only uses current and future information. The actions (u_τ^n) and water levels (h_τ^n) inherently have the same structure as the tree that is created from the ensemble forecast. How this tree is constructed is explained next.

The ensemble forecast (R_τ^n) consists of N_ensemble number of scenarios, which form a discrete approximation of the probability distribution of the future inflows to the system and is defined by the scenario n and the time step τ such that:

$$\begin{aligned} n\left(\tau\right) = \begin{cases} \epsilon \text{ when } \tau = t \\ n \text{ when } \tau > t \end{cases} \\ P\left[\epsilon = n\right] = \phi^n \quad n = 1, ..., N_\mathrm{ensemble} \\ \sum_{n=1}^{N_\mathrm{ensemble}} \phi^n = 1 \end{aligned} \tag{2.18}$$

where ϕ^n is the probability of each scenario.

With a large number of scenarios the calculation time needed for the optimization process will be large as well. To decrease the calculation time, the number of scenarios needs to be reduced. The optimal control action at the first time step based on this reduced ensemble should be close to the one obtained with the complete (non-reduced) ensemble. This problem is solved by heuristic algorithms [2]: the backward reduction algorithm and the scenario tree construction algorithm, which together form the scenario reduction algorithm. The backward reduction algorithm reduces the original ensemble to a reduced ensemble with a predefined number of scenarios, by aggregating scenarios. Which scenarios are aggregated is based upon the difference between scenarios and their probabilities. The next step is to create a tree of scenarios from this reduced ensemble. In this step no complete scenarios are aggregated, only parts of the scenarios are aggregated. The aggregation only occurs if the difference is smaller than a predefined threshold. In [9] the scenario reduction algorithm is characteristed by:

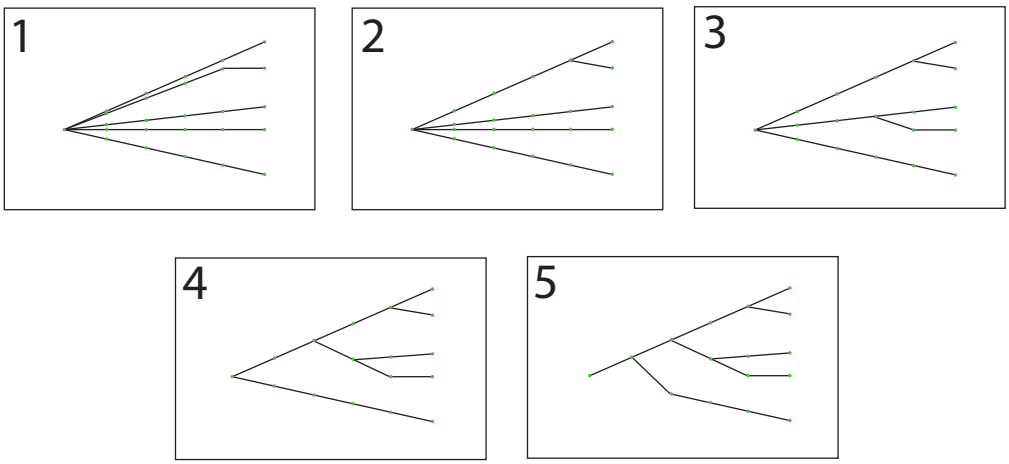


Figure 2.6: An example of a complete sequence of the scenario tree construction algorithm.

- The average difference between the values of two scenarios over the timespan under investigation is defined as 'distance'.
- The threshold is a fixed, pre-defined value, without a relation to the system.
- Aggregation and tree construction are done on inflow scenarios.

The backward reduction algorithm then works through the following steps:

1. Calculate a distance matrix, with all distances between the scenarios. This yields an N_{ensemble} by N_{ensemble} matrix.
2. Find the closest scenario for each scenario (i.e. smallest value per row of the distance matrix).
3. Multiply these with the probability of each scenario to get the probability weighted distances. This is done because there are at least two equal distances (e.g. distance between 2 and 3 and distance between 3 and 2) and the scenario with the lowest probability needs to be aggregated to the one with the highest, not the other way around.
4. Find the minimal probability weighted distance.
5. Aggregate this scenario to its closest scenario:
    a) Find the scenario to which it needs to be aggregated by searching for the smallest value in the row of the deleted scenario.
    b) Set the probability of the scenario that is deleted to 0 and add its original probability to the probability of the scenario it is aggregated to.

c) Remove the scenario that is aggregated from the distance matrix.

After step 5 the algorithm starts at step 2 again, until the dimension of the distance matrix is equal to specified number of reduced scenarios (N_{red}). An illustrative numerical example can be found in part 1 of Appendix A.

After the backward reduction algorithm the scenario tree construction starts. This is also a backward strategy. Starting at the second to last node, per node (time step) the total distance from node 1 until the actual node is calculated and the scenarios until the actual node are aggregated if the average distance is lower than the threshold. A schematic example is given in Figure 2.6 and in part 2 of Appendix A the result of the scenario tree construction on the reduced ensemble of part 1 is shown. Each branch of the tree (R_τ^n) is defined by the index of the scenario n and the time step τ:

$$\begin{aligned} n_{\tau+1} = \Xi\left(n_\tau;\tau\right) = \begin{cases} n_\tau \text{ when } \tau < \tau_n \\ \epsilon\left(n_\tau\right) \text{ when } \tau = \tau_n \end{cases} \\ P\left[\epsilon\left(n_\tau\right) = n_{\tau+1}\right] = \phi_{n_{\tau+1}} \\ \sum_n \phi_{n_{\tau+1}} = \phi_{n_\tau} \\ \phi_{n_\tau} = 1 \text{ when } \tau = t \end{aligned} \tag{2.19}$$

where τ_n is a bifurcation point. Every branch splits at its bifurcation point. The next branch after this point is a random branch out of all the possible next branches. This is of course dependent on the current branch (n_τ). The possibility of branches joining again is not taken into account. The predictability decreases with time, but this is not always the case on the entire ensemble length. Allowing branches to rejoin after a bifurcation point could be a useful improvement to the algorithm. This however considers an entire new approach to the scenario reduction algorithm and is thus classified as lying outside the scope of this research.

2.3.5 Adjustable parameters

Besides the parameters that TBMPC has in common with MPC, the core of TBMPC is the scenario reduction algorithm that can be defined by four components: (1) definition of distance, (2) threshold, (3) pre-processing of scenarios and (4) the number of reduced scenarios (N_{red}). The cost function can be altered by changing the value of the exponents and the penalties assigned to each component. Also the set point and normal operating band needs to be defined.

The adjustable parameters within the control algorithms and the objective function are presented in Table 2.4.

Parameter	Unit
Simulation horizon	hours
Distance definition	-
Scenario pre-processing	-
Threshold	-
N_{red}	-
E_{u}, E_{h}, E_{h_2} and E_{h_3}	-
w_{u}, w_{h}, w_{h_2} and w_{h_3}	-
h_{set}	m
h_{ref}	m

Table 2.4: Adjustable parameters of controllable system.

Coupled with the water system model described in paragraph 2.3.1 all the components for the optimization process are now defined and the control algorithm can be executed.

2.4 System overview

In the first paragraph of this chapter a simple schematized version of the system has been given. In Figure 2.7 this simple schematization is extended with the information given in all previous paragraphs to create a more complex, but complete schematization of the system.

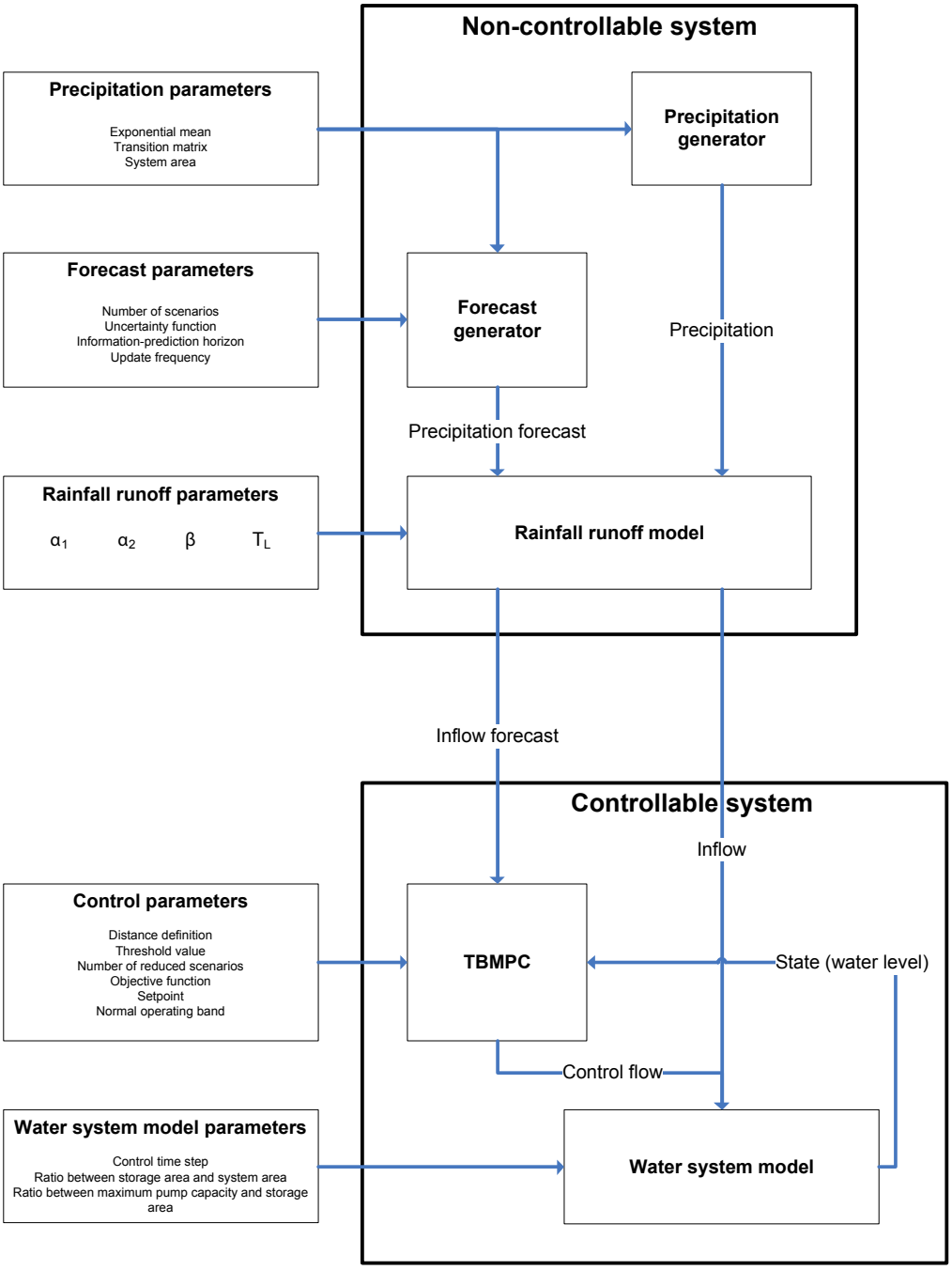


Figure 2.7: Complete model overview.

Chapter 3

Simulation framework

3.1 Introduction

The objectives set forth in Chapter 1 provide the basis for four experiments that will achieve those objectives. The experiments consist of a series of simulations that determine the influence of certain parameters on the performance of TBMPC. From Chapter 2 it is clear that there is a large number of variable parameters, not all of these are investigated. To efficiently perform the experiments a base case is set from where each experiment starts. Parameters under investigation are then varied; others are left at their base value. A certain set of parameters for which the performance is calculated is defined as a 'configuration'. A configuration defines the system that the experiments are performed upon. A common way to describe a controllable system is to say how close it is to 'certainty equivalence'. In [8] a system is said to be certainty equivalent when expected values can be used to identify optimal policies without loss of accuracy. In other words the average of the prediction can be used to determine actions with satisfactory results. To guarantee this, however, (1) the performance needs to be measured by quadratic functions, (2) system dynamics need to be linear, (3) no inequality constraints can exist and (4) uncertain inputs should be independent and normally distributed. The configurations (i.e. systems) used in this research are never certainty equivalent, because there are always inequality constraints (e.g. maximum pump capacity, finite storage capacity) and the objective function has thresholds in it, so they are never purely quadratic. Certain configurations, however, are more certainty equivalent than others. A larger storage capacity will make the system more certainty equivalent for example. Water levels are more likely to stay within the normal operating band, not exceeding any cost thresholds, yielding quadratic costs over the entire simulation horizon. When describing the configurations and analyzing the results this term will be

used.

Paragraph 3.2 describes how the performance of TBMPC and MPC is defined, the initial conditions and precipitation input are reviewed in paragraph 3.3, the base case values are presented in paragraph 3.4 and a detailed discussion of these values is presented in paragraph 3.5. Chapter 4 describes the individual experiments.

3.2 Performance indicator

The performance of a control algorithm can be defined in many ways, but it is always strongly correlated to the objectives set by the stakeholders. Ecologists that have a stake in a water system could measure the performance by the diversity of the ecosystem over a certain time span, but shipping authorities might use the number of days within a year that the water level is above the minimum required level for ships to be able to sail. The objective function in this research is a cost function with four components (see paragraph 2.3.2 for further details). The performance is thus the value of the objective function. To be able to compare performances based on the objective function a certain time span needs to be set where within the performance will be measured, this is the simulation horizon (T_s). Within this horizon the total costs are calculated and represent the performance of the configuration.

The cost function is not expressed in a certain monetary unit, nor does it reflect a realistic order of magnitude. This is only possible with detailed, system specific information. To be able to draw generic conclusions about the performance of TBMPC it is not necessary and also impossible to calculate the value of the cost function in this way. For performance comparison relative values suffice.

The absolute minimum of the objective function possible under certain climatic conditions, given the constraints of the controllable system will only be achieved when there are no uncertainties. There is however always uncertainty in the forecast, so the control actions are never optimal in hindsight. The forecast used here, although conditioned on the actual precipitation, is not a forecast without uncertainty. The performance of TBMPC will thus be expressed relative to the performance of the system with a perfect forecast (i.e. forecast is equal to the actual inflow). Since one of the objectives of this research is to compare TBMPC to MPC, in other words to determine the additional benefit of TBMPC over MPC, the performance will be expressed relative to MPC as well. To incorporate the aforementioned requirements the performance indicator will be the relative performance (V):

$$V = \frac{\mathrm{V_{mpc}} - \mathrm{V_{tbmpc}}}{\mathrm{V_{mpc}} - \mathrm{V_{perfect}}} \tag{3.1}$$

Processor	Intel Core2Duo E6550 @ 2.33Ghz
Memory	2GB DDR2
Windows version	Windows 7 Enterprise 64-bit
Matlab version	2009B SP1

Table 3.1: Specifications of computer system used for simulations.

A value of one indicates that TBMPC performs equal to MPC with a perfect forecast and 0 represents no improvement over MPC. This formula is unbounded below 0 and negative values represent situations where MPC performance better than TBMPC.

The MPC algorithm uses the same parameters as TBMPC, except for the ones that are not applicable to MPC (e.g. threshold value, number of reduced scenarios, etc.). The ensembles are generated by the forecast generator and thus are a realization of a random variable. To remove some of the randomness that these generated ensembles carry, five ensembles are generated for the precipitation series that is used. The performance of TBMPC is determined with each of these five ensembles. Since scenarios are picked randomly with MPC the performance of MPC is calculated 10 times for each of the five generated ensembles. The relative performance can thus be calculated 50 times. The presented relative performance is the average of these 50 values.

With some of the results the calculation time will be taken into account. Whenever this is is the case one needs to take into account that these values are obtained on one specific computer (see Table 3.1). Calculation time for other systems (i.e. different versions of Matlab, Windows or hardware components) could be materially different.

3.3 Experiments input and initial state

The performance of the control algorithm is measured over the simulation horizon, during which the inflow to the system needs to be defined. As emphasized in Chapter 1 the control algorithm needs not only to perform well under extreme conditions, but also under normal conditions. The precipitation series over the simulation horizon thus need to have average precipitation amounts, but also extreme precipitation events. By generating precipitation time series with a length of one year and at least one extreme precipitation event (defined as $P_t > 13$ mm/hour, because that is more than the storage capacity (12 mm) and the pump capacity (0.5 mm) during one time step of the system with base case parameters) both such normal conditions and extreme events are represented in the time series. The total performance value from such a long horizon illustrates more the effectiveness of a control algorithm, rather than gives theoretical insight

Parameter	Value [unit]
Inflow (R_0)	0 [m^3/hour]
Pump discharge (u_0)	0 [m^3/hour]
Water level (h_0)	0 [m]

Table 3.2: The initial conditions for the simulations.

in how the algorithm performs under varying conditions. However, by looking at the performance over time (i.e. the cumulative costs) and matching this with events during the simulation period it is expected to get such theoretical insight as well, if needed. Details on the precipitation distribution parameters are found in paragraph 3.5.

The initial state of the system encompasses the initial water level, inflow and outflow. The initial state is actually of little importance when evaluating the performance. The experiments should provide results that are comparable with real world performance. In the real world a control algorithm is never implemented at certain initial conditions, performs during a predefined time span and at the end a certain end condition needs to be met. To the contrary, it operates continuously, always trying to keep the water level at set point. All experiments will thus start from a steady-state system, with its water level at set point and in and outflow equal to zero (see Table 3.2). This avoids biased results, which might occur when the initial water level is equal to h_{ref}, for example. Effects of disturbances, especially at the beginning of the simulation period, are likely overestimated in that case.

3.4 Simulation parameters

Most of the experiments only focus on a small part of the model (e.g. only the system configuration) and its parameters. However with every experiment all parameters need to be defined in order to allow the simulation to run, which creates the need for a 'base case'. These are the values that the parameters not under investigation by the experiment are set to. The values of all model parameters that are used in the base case are presented in Table 3.3. The next paragraph discusses these parameters and how their base values were derived.

3.5 Simulation parameter discussion

Precipitation distribution (E_{clim} and C)

The mean rainfall amount (E_{clim}) and transition matrix (C) are based on a one year dataset from 'De Bilt' in The Netherlands. 'De Bilt' is a city located in the center of the Netherlands and is representative for Dutch climatic conditions.

Parameter	Value	Unit	Variable
E_{clim}	0.7928	mm/hour	No
C	$\begin{bmatrix} 0.6755 & 0.3245 \\ 0.0409 & 0.9591 \end{bmatrix}$	-	No
T_{s}	365	days	No
T_{o}	T_{ic}	hours	No
T_{ip}	variable	hours	Yes
Control time step (Δt)	1	hour	No
Update frequency	6	hours	No
Optimization time step	6	hours	No
Uncertainty function	see eq. 2.8	-	No
A_{system}	1	ha	No
α_1	0.35	-	No
α_2	0.15	-	No
β	0.50	-	No
T_{L}	0.0	time step	No
h_{set}	0.0	m	No
h_{ref}	0.15	m	No
$\frac{A_{\text{storage}}}{A_{\text{system}}}$	0.04	-	Yes
$\frac{u_{\text{max}}}{A_{\text{system}}}$	0.0005	m/hour	Yes
Distance	Average	m^3/h	No
Threshold	u_{max}	%	Yes
N_{ensemble}	50	Scenarios	No
N_{red}	8-10	Scenarios	Yes
E_{u}	2	-	No
E_{h}	2	-	No
E_{h_2}	2	-	No
E_{h_3}	2	-	No
w_{u}	$\left(\frac{h_{\text{ref}}^2}{u_{\text{max}}^2}\right) \times w_{\text{h}}$	time step^2/m^4	No
w_{h}	1	m/E_h	No
w_{h_2}	100	m/E_{h_2}	No
w_{h_3}	1000	m/E_{h_3}	No

Table 3.3: Simulation parameters. Those parameters that are indicated with 'Yes' are altered during the experiments and their influence on the performance is investigated.

These parameters are used throughout the research and will not be altered, because the research focuses on the performance of control algorithms on systems in The Netherlands. A more detailed description of these parameters and how they were derived can be found in Appendix C.

Simulation horizon (T_s)

The performance of the algorithm is measured over a certain period and that period consequently has a large influence on the performance, it is called the simulation horizon. It is set to one year so the algorithm will be tested under dry, wet and extreme conditions. This is especially important in a risk based framework, since the algorithm needs to perform well not only during extreme events, but under normal operating conditions as well (see for more details paragraph 3.3).

Control time step (Δt)

The control time step is the time between two subsequent actions. It is set to one hour, as many systems in the Netherlands have a control time step not larger than one hour. By keeping this value at one hour, an overestimation (due a too fast control time step) of the performance will be unlikely.

Update frequency and optimization time step

The control time step is one hour, so an action is taken every hour. Which action(s) to take is determined during the optimization process which is done at a certain interval (i.e. optimization time step). During this process the ensemble available at that moment is used, a new ensemble (i.e. forecast) becomes available at a certain interval (i.e. update frequency) as well. For ease of interpretation the update frequency and optimization time step are equal during this research. A larger optimization time step leads to a system further away from certainty equivalence, because actions are determined for moments further into the future where less accurate forecasts are available. A larger optimization time step also decreases computation time over a predefined simulation horizon. An optimization time step of six hours is used throughout this research based on the aforementioned observations. With an optimization time step of six hours, the first six actions need to be implemented. This requires that on the first six time steps there is only one possible set of actions, thus the tree can consist of only one scenario until $\tau = 6$. On the first six time steps scenarios are aggregated to one scenario independent of the threshold value. Currently the update frequency of the ensemble forecasts provided by the KNMI to the water boards is 12 hours, however, six hourly values are forecasted, instead of the hourly values used here.

Optimization horizon (T_o) and information-prediction horizon (T_{ip})

The optimization horizon is the horizon for which the control problem is solved. The risk (i.e. J) over this horizon is minimized every time step. An optimization horizon larger than the information-control horizon (T_{ic}) is not necessary, but only increases computation time. T_{ic} is defined as the time span from the actual moment until the moment from which information (e.g. forecasted precipitation amount) does not influence the control actions anymore [17]. The optimization horizon will be set equal to the T_{ic}, which is defined as:

$$T_{ic} = \frac{h_{normal} \times A_{system} \times \frac{A_{storage}}{A_{system}}}{\left(\frac{u_{max}}{A_{system}} \times A_{system}\right) - C} \tag{3.2}$$

where h_{normal} is the normal operating band in meters, T_{ic} the information-control horizon in hours, C the 'typical' inflow coefficient (1 m^3/hour in this case) and A_{system} the system area in m^2. An overview of the experiment that has been performed to determine equation 3.2 can be found in Appendix D.

The information-prediction horizon (T_{ip}) can be defined as the time span from the actual moment until the moment where the conditional distribution of future events, conditional to all actual information, becomes the same as the marginal (climatic) distribution of these events. The forecast generator used in this research is dependent on the uncertainty function (see eq. 3.3), which is dependent on T_{ip}. It determines not only where the forecast is not conditioned on the actual precipitation anymore, but also the slope of the uncertainty function. A longer T_{ip} also increases the quality of the forecast over the period before T_{ip}. It is thus also possible (in our case) to have a T_{ip} larger than T_{ic}. It is hard to derive a T_{ip} from real world ensembles. The current ensemble predictions provided by the KNMI show that after six days the variance of these ensemble forecasts does not increase anymore, but it remains stable at a certain value, which is assumed to be the climatic variance (see Figure 2.3). One could argue to use a T_{ip} of six days. However the precipitation is forecasted in six hour time brackets. This does not mean that the same accuracy can be achieved when these forecasts are downscaled to hourly estimates, uncertainty will probably be much higher. It is hard to find one value for T_{ip} that serves all experiments well. Therefor T_{ip} is determined separately for each experiment. Some experiments require a constant value for T_{ip} (e.g. to isolate the influence of the parameter under investigation), some a constant relative value (e.g. as a percentage of the optimization horizon) and with yet other experiments the same configuration with a varying T_{ip} is investigated.

Size of original ensemble ($N_{\mathbf{ensemble}}$)

In Europe ensemble predictions are provided by The European Center for Medium-Range Weather Forecasts (ECMWF). The KNMI downscales these predictions and water boards can use them in their decision support systems. The ensemble predictions of the ECMWF currently have 50 ensemble members. Future research is focused on increasing the spatial resolution, not so much on the number of ensemble members. The number of scenarios is thus kept constant at the current 50 in this research and the influence of more ensemble members is not investigated. Instead emphasis is put on the scenario reduction algorithm and the marginal information loss with every scenario deletion.

Uncertainty function (α_τ)

The uncertainty function defines the amount of uncertainty within the forecast in time. This can be seen as the variance of the ensemble members in time. When looking at this variance in Figure 2.3 a clear pattern is visible. The variance starts at nearly 0 and increases linearly until it flattens out and stays constant. The uncertainty function mimics this pattern:

$$\alpha_\tau = \begin{cases} \frac{\tau}{T_{\text{ip}}}, \text{ for } \tau \leq T_{\text{ip}} \\ 1, \text{ for } \tau > T_{\text{ip}} \end{cases} \tag{3.3}$$

How the uncertainty function is used in the forecast generator can be found in paragraph 2.2.2.

Rainfall runoff model parameters ($\alpha_1, \alpha_2, \beta$)

The rainfall runoff model is an autoregressive exogenous model with three parameters: two for the autoregressive part and one for the exogenous component. Values for these parameters are chosen based on the resulting general behavior of the model. As explained earlier the runoff process in a polder is fairly quick, but a smoothening of the precipitation pattern also occurs. By iteratively setting the parameters to 0.35, 0.15 and 0.5 respectively this behavior is realized. Much time and effort could have been put into a more complex model and refinement of the parameters; however it has often been shown that simple rainfall runoff models provide satisfactory results as well (see for example [7], [4] and [3]).

System area ($A_{\mathbf{system}}$)

The influence of the size of the system on the performance is modeled through the moving average filter on the precipitation, which is dependent on the system area. Since the purpose is to model a less extreme precipitation distribution, the actual size of the system is not relevant. Furthermore it is only used in ratios,

where the actual value is of no importance as well. The base value is 1 ha (no smoothening of the precipitation distribution).

Lead time ($T_{\mathbf{L}}$)

Normally there is no lead time between the runoff process and the inflow to the belt canals in a polder-belt canal system. However the use of the polder drainage canals as temporary storage introduces a lead time. Typically the time the water can be stored is short and thus the lead time will be short as well. Water can also for short periods be stored in greenhouses, which can be modeled with a lead time as well. The base value is zero.

Water system model parameters ($h_{\mathbf{set}}, h_{\mathbf{ref}}$)

The values for h_{set} and h_{ref} will be kept constant. Increasing h_{ref} effectively increases the storage volume within the system or increases the system's capacity to deal with increased water levels due to high inflows; however that can be accomplished as well by increasing the ratio between storage area and system area and the ratio between pump capacity and storage area respectively. Set point will be set to zero and h_{ref} will be 0.15 m, in the order of magnitude of the range water boards in the Netherlands currently use.

Ratio between storage area and system area $\left(\frac{A_{\mathbf{storage}}}{A_{\mathbf{system}}}\right)$

Table 3.4 shows this ratio for three large polder-belt canal systems in The Netherlands. This ratio can vary quiet dramatically between systems, because it is so dependent on the system's unique natural and artificial properties. A value of 0.04 is chosen here to be representative and is used for the base case.

Ratio between pump capacity and system area $\left(\frac{u_{\mathbf{max}}}{A_{\mathbf{system}}}\right)$

The capacity of belt canal pumps ("boezemgemalen") varies widely between systems (see Table 3.4), because of the reasons stated under the discussion of the ratio between storage area and system area. The base case value is set to 0.0005 (i.e. 5 m^3/h for a system area of 1 ha), the average (non rounded values where used to calculate the average) of the three systems presented in Table 3.4.

Scenario reduction algorithm parameters

The scenario reduction algorithm is defined by four components: (1) definition of distance, (2) threshold, (3) pre-processing of scenarios and (4) the number of reduced scenarios. There is no real world reference data available to derive suitable values for these definitions. It is unknown to what extent these parameters influence the performance, thus extensive simulations are required to determine the

Waterboard	$\frac{A_{\text{storage}}}{A_{\text{system}}}$ [-]	$\frac{u_{\max}}{A_{\text{system}}}$ [m/h]
Wetterskip Fryslân	0.0451	0.0002
Hoogheemraadschap van Rijnland	0.0409	0.0005
Hoogheemraadschap van Delfland	0.0178	0.0010

Table 3.4: Water system characteristics of three polder-belt canal systems in the Netherlands.

best values for these parameters. These simulations are performed in experiment 1, 2 and 3, described in paragraphs 4.2, 4.3 and 4.4 respectively.

Cost function parameters $(E_u, E_h, E_{h2}, E_{h3}, w_u, w_h, w_{h2}, w_{h3})$

As explained in paragraph 2.3.2 the structure of the cost function represents costs associated with water levels in polder areas well, but on the other hand it needs to be able to represent different kind of polder systems. There are eight parameters that could be varied to represent different systems; however this objective can be achieved by only changing two as well.

The four components need to be equally weighted, but also produce costs of different magnitudes. By setting all exponents equal, but use different penalties this can be achieved. Since a cost function in a water system typically has a quadratic shape [14], all exponents are two and are not altered throughout the research.

The quadratic penalization of structure changes represents how a gate's maintenance costs are influenced by how it is operated. However costs associated with the operation of pumps show no correlation with this penalization. Costs increase linear with the pump's discharge, but changing the discharge does not yield any additional costs. When primarily dealing with pumps in the controllable system, a linear penalization on the absolute discharge is preferred. This adjustment leans toward a more specific case. In the generic context of this research a quadratic penalization is used.

As the cost function cannot provide exact values, but only serves a relative comparison purpose, only the ratio between the four penalties is of concern. The first penalty must always be small in relation to the others, as maintenance costs are always small in relation to flooding costs, but in the same order of magnitude as the second penalty since both fall under normal operating expenses. The Delfland water board spent about €300.000 in 2009 on structure replacement. When comparing this with the costs of a flood, for example the flood on 19 and 20 September 2001, which resulted in costs of about €23.000.000 (adjusted for inflation) [15], maintenance costs are indeed relatively small (2 orders of magnitude difference). Note that in the cost function the increase in costs due to more frequent setting changes is expressed, not the regular maintenance costs. These

extra costs are again a fraction of the normal maintenance costs. By setting the first penalty to $\left(\frac{h_{\text{ref}}^2}{u_{\text{max}}^2}\right) \times w_{\text{h}}$ the first two components provide costs in the same order of magnitude. The ratio between the second and third/fourth penalty determines the slope of the quadratic function above and below h_{ref} and thus can represent a multitude of systems.

In the Netherlands the land surrounding belt canals is of high value and thus flooding there causes damage of completely different magnitudes than water level fluctuations within the belt canals. This flooding often occurs because pumping to the belt canals is discontinued, otherwise water levels in the belt canals would rise so high, water will over top the embankments. Consequently the water level in the polder will rise and cause damage. Since only the belt canal system is modelled here, this phenomenon needs to be represented in the belt canal model. The water level can rise infinitely in the belt canal and when a very high penalty is applied to water levels above the normal operating levels (at which point the pumping is often discontinued) this phenomenon can be mimicked. It is in this light the ratio between the second and third penalty is chosen to be 1:100 (see costs of flood on 19 and 20 September 2001). To represent different systems this ratio can be varied.

Water levels below normal operating levels will cause very high damages as well, since dikes might collapse, houseboats endure severe damage, ships can fall dry, etc. In addition water boards are very anxious to lower water levels below this normal operating range (e.g. to avoid water shortage during an unexpected dry spell); this is only done in very extreme situations. The ratio between the second and fourth penalty is thus 1:1000, to represent this last phenomenon. This type of damage will fluctuate less with varying systems and is thus kept constant.

Chapter 4

Results and discussion

4.1 Introduction

This chapter describes the experiments that are performed to provide answers to the objectives stated in Chapter 1. All experiments are performed with closed loop simulations, yielding a high number of closed loop simulations that together can answer the first objective, namely the difference in closed loop performance between TBMPC and MPC. The second and third objective focus on the scenario reduction algorithm. The first three experiments deal with these two objectives. The last experiment investigates the influence of different system configurations on the relative performance; the last objective.

The first three experiments are aimed at the objectives for the analysis and improvement of the scenario reduction algorithm. It has four components that are of interest for these objectives: (1) definition of distance, (2) threshold, (3) pre-processing of scenarios and (4) the number of reduced scenarios.

The first objective is to be able to fine-tune the scenario reduction algorithm to the water system it is applied on. The threshold value plays an important role in this process. The threshold determines whether or not two scenarios are close enough to each other to be aggregated over a certain time span. Two scenarios with an average inflow of 10% and 20% of the maximum pump capacity of a system respectively can easily be aggregated. Both do not require anticipatory actions, feedback control is sufficient for these scenarios. Scenarios that have an average inflow larger than the maximum pump capacity do need anticipation and should not be aggregated to the 10% or 20% scenario for example. In other words a difference equal to the maximum pump capacity of the system makes a material difference for the system and requires different anticipatory actions. This leads to the hypothesis that the threshold value should be a fraction of the maximum pump capacity of the system and that the optimal fraction is 100%.

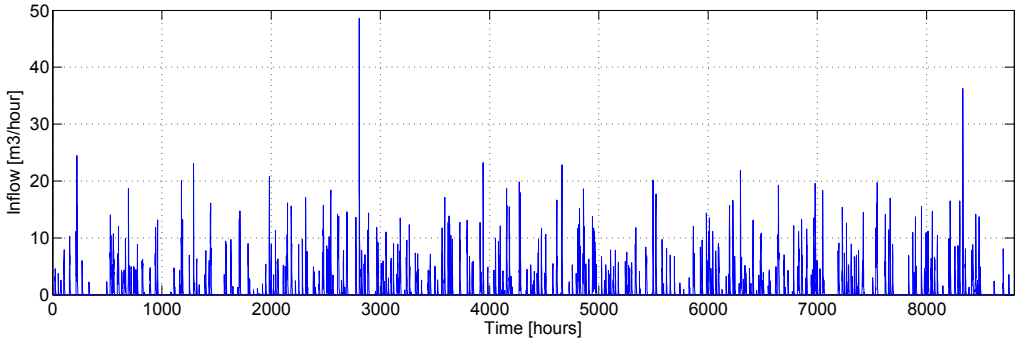


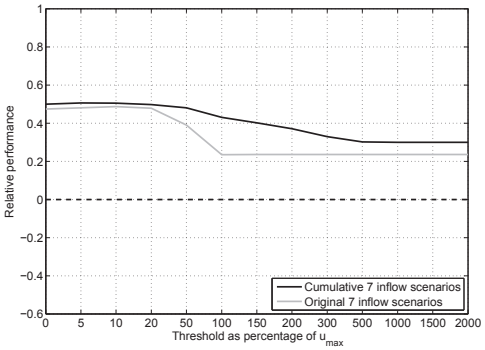
Figure 4.1: Inflow series used for pre-processing, threshold and marginal information loss experiments. Average inflow of 0.95 m^3/hour. Maximum inflow of 48.59 m^3/hour. Total inflow of 8390 m^3.

The above described threshold is based on average differences between scenarios. Thus the distance will be defined as the average difference between two scenarios over the evaluated time span. Since the scenario tree construction algorithm is a backward algorithm the average distance is first assessed over the time span from the first node until the second to last, subsequently from the first to the third to last and so on. Whenever the average distance between two scenarios over this time span is smaller than the threshold value the two scenarios are aggregated and a tree-shaped ensemble is thus created during this process.

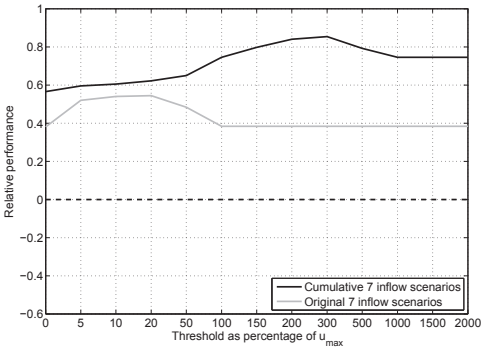
The first component of the scenarios reduction algorithm is determined; distance is defined as the average difference between two scenarios. Two experiments are designed to determine the second and third component. Both experiments are done under the following assumptions.

The scenario tree construction algorithm is the second part of the scenario reduction algorithm. In the first part a pre-defined number of scenarios is set to which the original ensemble is reduced. To see the effect of this parameter on the results of the first two experiments, they are done for three different number of scenarios, 7, 20 and 50. A more detailed analysis of the influence of the number of scenarios on the performance is carried out in experiment three.

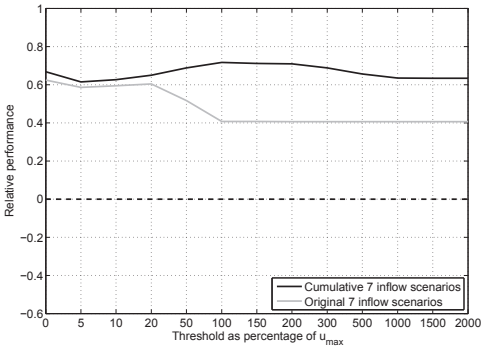
Threshold values from 0% to 2000% of the maximum pump capacity are evaluated. The experiments are first done for the base case configuration. To check whether the results are consistent with different values for the maximum pump capacity, storage area, information-prediction horizon and optimization horizon the storage area (A_{storage}) and the maximum pump capacity ($u_{\max}$) are varied. For all these configurations the information-prediction horizon (T_{ip}) is kept at 50% of the optimization horizon (i.e. information-control horizon T_{ic}, see for the calculation of T_{ic} Appendix D). Four configurations have been choosen to represent a variety of the aforementioned parameters:



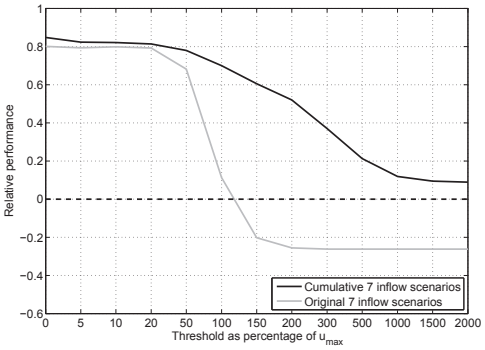
(a) $\frac{A_{\text{storage}}}{A_{\text{system}}} = 0.02$ and $\frac{u_{\max}}{A_{\text{system}}} = 0.0005$.



(c) $\frac{A_{\text{storage}}}{A_{\text{system}}} = 0.06$ and $\frac{u_{\max}}{A_{\text{system}}} = 0.0005$.



(b) $\frac{A_{\text{storage}}}{A_{\text{system}}} = 0.04$ and $\frac{u_{\max}}{A_{\text{system}}} = 0.0005$.



(d) $\frac{A_{\text{storage}}}{A_{\text{system}}} = 0.02$ and $\frac{u_{\max}}{A_{\text{system}}} = 0.00025$.

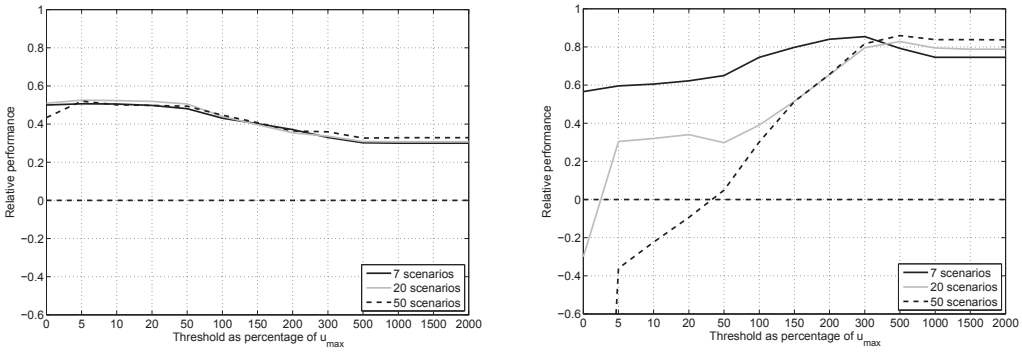
Figure 4.2: Relative performance comparison for TBMPC with the original scenarios and TBMPC with cumulative scenarios. The horizontal axis denotes the value of the threshold as a percentage of the maximum pump capacity.

- $\frac{A_{\text{storage}}}{A_{\text{system}}} = 0.04$, $\frac{u_{\max}}{A_{\text{system}}} = 0.0005$ and $T_{\text{o}} = 30$ (base case).
- $\frac{A_{\text{storage}}}{A_{\text{system}}} = 0.02$, $\frac{u_{\max}}{A_{\text{system}}} = 0.0005$ and $T_{\text{o}} = 15$.
- $\frac{A_{\text{storage}}}{A_{\text{system}}} = 0.06$, $\frac{u_{\max}}{A_{\text{system}}} = 0.0005$ and $T_{\text{o}} = 45$.
- $\frac{A_{\text{storage}}}{A_{\text{system}}} = 0.02$, $\frac{u_{\max}}{A_{\text{system}}} = 0.00025$ and $T_{\text{o}} = 40$.

The simulations are conducted on a randomly generated year long precipitation series that is shown in Figure 4.1.

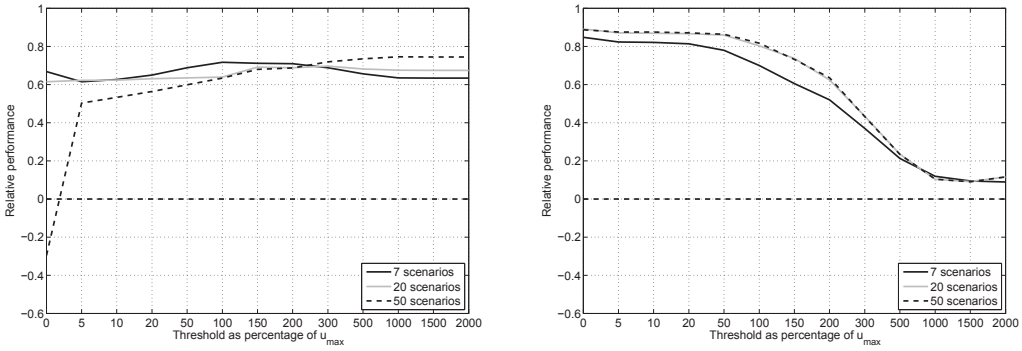
4.2 Experiment 1: Pre-processing of scenarios

The aggregation and the tree construction can be done on the original inflow scenarios, however these can be pre-processed as well. It is expected that translating these scenarios to cumulative scenarios yields better results. Consider two identical scenarios with one peak event and only a phase difference of one time



(a) $\frac{A_{\text{storage}}}{A_{\text{system}}} = 0.02$ and $\frac{u_{\max}}{A_{\text{system}}} = 0.0005$.

(c) $\frac{A_{\text{storage}}}{A_{\text{system}}} = 0.06$ and $\frac{u_{\max}}{A_{\text{system}}} = 0.0005$.



(b) $\frac{A_{\text{storage}}}{A_{\text{system}}} = 0.04$ and $\frac{u_{\max}}{A_{\text{system}}} = 0.0005$.

(d) $\frac{A_{\text{storage}}}{A_{\text{system}}} = 0.02$ and $\frac{u_{\max}}{A_{\text{system}}} = 0.00025$.

Figure 4.3: Relative performance versus threshold as percentage of the maximum pump capacity of the system for four different configurations.

step (assumed to be small relative to T_{o}). By using the original scenarios the difference between these two scenarios looks to be severe, however they actually are much alike. The same anticipatory actions need to be taken, only the timing of the event is slightly uncertain. When more information becomes available this uncertainty will disappear and since roughly the right anticipatory actions were already taken, there will be enough time to make the latest adjustments for the exact timing of the event. When using cumulative scenarios the difference between these two scenarios will be much smaller. Because absolute differences are used the peak event only causes one large difference with cumulative scenarios compared to two large differences with the original scenarios, yielding lower distance values for the cumulative case. A better distinction between scenarios that need really different anticipatory actions can thus be made. Because of the integrating behavior (which shows much resemblance with the cumulative inflow) of a reservoir, such a system should benefit particularly from using cumulative scenarios to make the distinction between scenarios.

This experiment is only done with seven scenarios, not with 20 and 50 scenarios, since the largest benefit is expected with a lower number of scenarios.

The more scenarios are preserved the less important the choice of which scenario to aggregate becomes. For all configurations the performance is assessed with threshold values in the range from 0% to 2000% of the maximum pump capacity. The results of this experiment with the aforementioned configurations are shown in Figure 4.2.

Using cumulative scenarios to base the tree-construction on yields better results for every configuration and every threshold value. The advantage of using cumulative scenarios is clear from these results. Indeed it looks like the reservoir system used here benefits from the better distinction between scenarios. Cumulative scenarios will be used in the subsequent experiments.

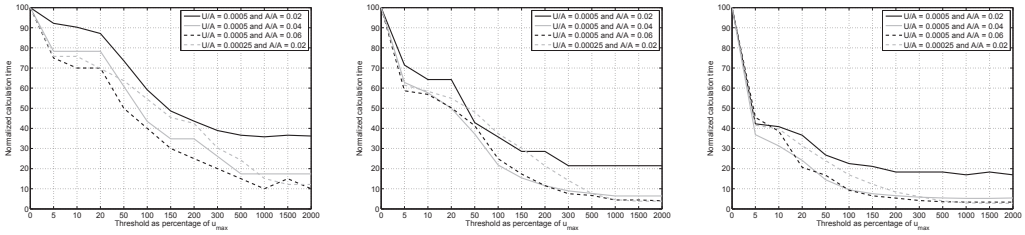
4.3 Experiment 2: Threshold value

The influence of the threshold value on the performance is investigated with this experiment. The simulations are done with 7, 20 and 50 scenarios for the aforementioned configurations. The results are shown in Figure 4.3.

The influence of the threshold value varies for all four systems. A value of 100% of the maximum pump capacity yields the highest relative performance only for the base case configuration with seven scenarios. Configuration 4.3a and 4.3d, however perform best without any tree construction at all (i.e. threshold value equals 0%), but the 100% value is not far off from the best results. Higher values than 100% provide worse results for these configurations. The opposite is shown for configuration 4.3c, where again the 100% value performs satisfactory, but higher values can even provide better performance. The hypothesis that a threshold value of 100% of the maximum pump capacity performs the best holds up as over this wide variety of configurations it provides a performance that is either the optimum or very close the optimum that can be achieved. Some threshold values show a higher relative performance than the 100% value in one case, but are unacceptably worse in other cases (e.g. compare 300% for configurations 4.3c and 4.3d).

It is clear that the threshold value and with it the tree that is created based on the distance and the threshold is an important component in the performance of TBMPC. From Figure 4.3 only the effect on the closed loop performance can be seen. How the tree is influenced and varies with different threshold values is not clear from these results. Also the effect on the calculation time is not yet examined. The latter is first briefly investigated, followed by a detailed analysis of the effect on the tree.

In Figure 4.4 the normalized calculation time of the four configurations under investigation is shown. Contrary to the effect on the closed loop performance, the effect on the calculation time is consistent for all configurations. A higher threshold value decreases the calculation time. A threshold value of 100% shows a



(a) Calculation times from the simulations with 7 scenarios.

(b) Calculation times from the simulations with 20 scenarios.

(c) Calculation times from the simulations with 50 scenarios.

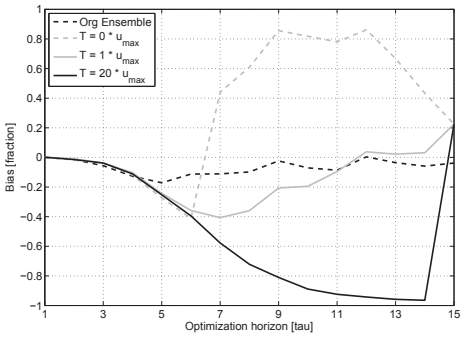
Figure 4.4: Normalized calculation time plotted against the threshold value as a percentage of the maximum pump capacity for the 4 tested configurations. Normalized in this case means that the largest value for each series is set to 100 and the other values are expressed as a fraction of that.

40% to 60% reduction in calculation time when seven scenarios are used to create a tree. When, however, 50 scenarios are used the calculation time reduction is 80% to 90%. The reduction in calculation time thus increases when the tree is created from more scenarios.

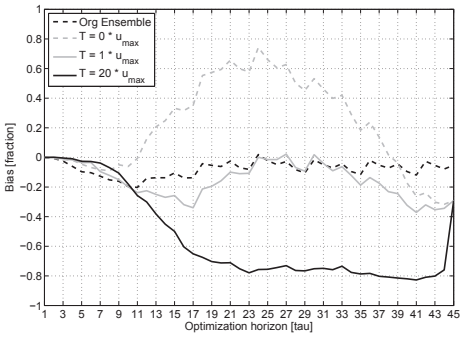
To give insight in whether or not the reduced ensemble (i.e. the created tree) is a good estimation of the actual inflow and is thus a representative ensemble, which is the objective of the scenario reduction algorithm, the ensembles themselves are investigated. If the probability weighted average of the ensemble is equal to the actual inflow it is a representative ensemble, however if the ensemble over- or underestimates the actual inflow it has a bias. Figure 4.5 shows the bias of the original ensemble, the reduced ensemble without tree construction (i.e. threshold = 0) and with a threshold value of 100% and 2000% of the maximum pump capacity. The bias is determined by comparing the difference between the actual inflow and the probability weighted average of the ensemble on each time step of the optimization horizon. This is evaluated over the complete simulation horizon and expressed as a percentage of the actual inflow. The averages of these results are shown in Figure 4.5. The calculations were done on the results of the simulations with a reduced ensemble with seven scenarios ($N_{\text{red}} = 7$).

As hypothesized the threshold has indeed a significant influence on the tree construction. Without the tree construction (i.e. threshold = 0) the reduced ensemble is overestimating the inflow, by setting an appropriate threshold this bias can be largely removed again. Choosing a too high threshold yields an underestimation of the inflow. Note that the bias for all reduced ensembles is equal at the last time step since no aggregation is done on this time step. For programming reasons the number of nodes at the last time step had to be equal to the number of reduced scenarios.

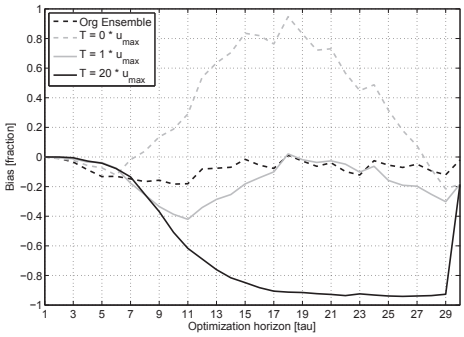
The effect of the tree construction and the threshold value on the bias of the reduced ensemble is the same for all four configurations, however the effect on the



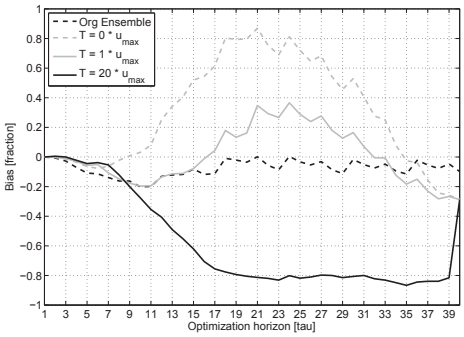
(a) $\frac{A_{\text{storage}}}{A_{\text{system}}} = 0.02$ and $\frac{u_{\max}}{A_{\text{system}}} = 0.0005$.



(c) $\frac{A_{\text{storage}}}{A_{\text{system}}} = 0.06$ and $\frac{u_{\max}}{A_{\text{system}}} = 0.0005$.



(b) $\frac{A_{\text{storage}}}{A_{\text{system}}} = 0.04$ and $\frac{u_{\max}}{A_{\text{system}}} = 0.0005$.



(d) $\frac{A_{\text{storage}}}{A_{\text{system}}} = 0.02$ and $\frac{u_{\max}}{A_{\text{system}}} = 0.00025$.

Figure 4.5: Bias of the original ensemble and the reduced ensemble with a threshold value of 0%, 100% and 2000% of the maximum pump capacity. The bias is the difference between the probability weighted average of the ensemble and the actual inflow as a fraction of the actual inflow. This is done for all time steps of the optimization horizon for all optimizations over the simulation horizon. The average of those values is shown here. Note that the bias for all reduced ensembles is equal at the last time step since no aggregation is done on this time step.

relative performance is not. Two effects can be observed from Figure 4.3. The combination of a high number of scenarios and not creating a tree result in a large drop in relative performance for configurations 4.3b and 4.3c. A high threshold value only leads to a significantly lower relative performance in the case of configuration 4.3d. The latter can be explained by certainty equivalence and the bias results shown in Figure 4.5. Configuration 4.3c is close to certainty equivalence, whereas configuration 4.3d is further away from certainty equivalence. In the latter case underestimation of peak events (due to a high threshold value) in the ensemble easily leads to an upward exceedance of the normal operating band and a sharp increase in costs. For configuration 4.3c, however, underestimation does not lead to an upward exceedance of the normal operating band as the system can easily buffer or pump out any inflow, by using its feedback capabilities.

The large decrease in relative performance for a high number of scenarios and

no tree creation for configurations 4.3b and 4.3c can be explained by the bias and certainty equivalence as well.

Reducing the ensemble to a lower number of scenarios in the first part of the scenario reduction algorithm leads to a larger positive bias in the ensemble. Also, the larger the threshold value, the smaller the difference between the ensembles with 7, 20 and 50 scenarios becomes. This has been examined and confirmed in a separate experiment, but results are not shown. Reduced ensembles with more scenarios thus have a smaller positive bias and thus predict lower inflow, which leads to on average higher water levels. A higher average water level leads to higher costs compared to the optimal situation of an average water level at set point.

A second effect is causing underestimation of the inflow (and thus higher water levels) as well. Consider an ensemble with a peak event halfway in the forecast. With 50 scenarios and a threshold value of zero the scenarios split after $\tau = 6$ (because the first six actions are implemented, they are not allowed to split earlier). Uncertainty is still low at this point in the ensemble, leading to multiple scenarios, close to each other. Not all scenarios will predict the peak event. At the time step of the peak event the scenarios start to really differ. This is the point where the scenarios should actually split, since this is where the uncertainty actually enters the system. A threshold value of 5% of the maximum pump capacity will aggregate a lot of scenarios already between $\tau = 6$ and the timestep of the peak event. A larger part of the ensemble then carries the possibility of the peak event occuring. This leads to higher pump actions calculated on the first six timesteps. Without the tree, lower pump actions are calculated and implemented creating the secondary underestimating effect. The average water levels for the four configurations in Figure 4.3 can be found in Appendix E, Table E.1.

Not only are there additional costs due to a higher average water level, but also due to additional pumping. When the system overestimates the inflow, it has lowered the water level too much by pumping too much. This situation is corrected by stopping the pumping and letting the water level increase. When the inflow is underestimated the water level was not lowered enough and this situation can only be corrected by 'emergency' additional pumping. Pumping changes are penalized quadratically leading to higher pump costs compared to when a correct forecast of the inflow was used. The total pump costs for the four configurations in Figure 4.3 can be found in Appendix E, Table E.2.

Both effects occur for all configurations. For a system close to certainty equivalence the total costs will be small compared to a system that is further away from certainty equivalence, where costs are dominated by upward exceedance of the normal operating band. The impact of the small increase in average water level is thus also small for a system further away from certainty equivalence (i.e. configuration 4.3a), but higher for a system closer to certainty equivalence

(i.e. configuration 4.3c). The same holds for the additional pumping costs, they are small compared to the total costs for systems further away from certainty equivalence and larger for systems closer to certainty equivalence. This is the reason for the different effects on the relative performance shown in Figure 4.3. The total pump costs and costs due to an upward exceedance of the normal operating band, both expressed as a percentage of the total costs for the four configurations in Figure 4.3 can be found in Appendix E, Table E.3.

This negative property of TBMPC only has a material effect on configurations where water levels do not exceed the upper boundary of the normal operating band anyway. The use of a large number of scenarios and no tree construction to keep as much information as possible is not needed here. A simple feedback controller would have performed equally well or better (i.e. see relative performance for a threshold value of 2000% of the maximum pump capacity for configuration 4.3c).

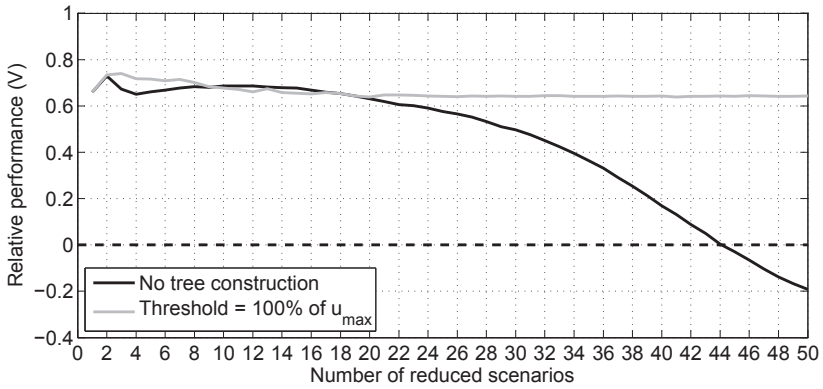
The two parts of the scenarios reduction algorithm need to go hand in hand. Not creating a tree yields non representative reduced ensembles, because they carry a bias. The presented results make a good case for the use of a threshold value of 100% of the maximum pump capacity in the second part of the scenario reduction algorithm. The bias that is present in the ensemble after the reduction of complete scenarios is largely removed, calculation time is reduced by 40% to 90% and the relative performance is either optimal or close to the optimum that can be achieved for varies threshold values.

4.4 Experiment 3: Influence of the number of scenarios

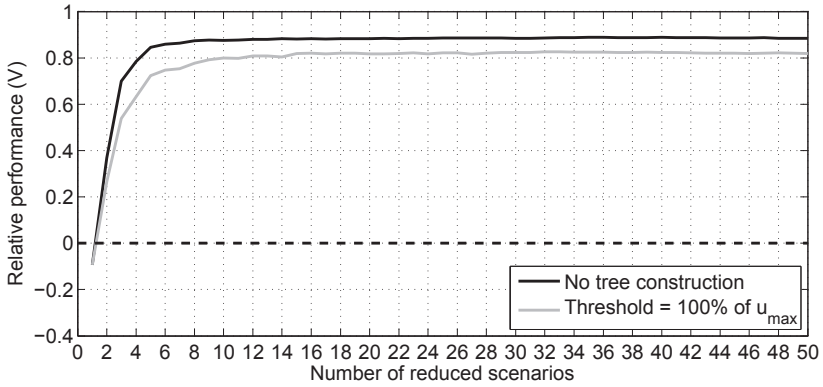
The second objective is to determine the influence of the number of scenarios. This is done by looking at the marginal information loss per deleted scenario, which is defined as the decrease in relative performance (V) per deleted scenario. The marginal information loss depends on the scenario aggregation and the tree construction. Simulations with a threshold value of 0% are done to show the influence of aggregation of complete scenarios on V. Evident from the previous experiments is the influence of the tree construction on V, especially for a high number of scenarios. Simulations with a threshold value of 100% are done to see if the marginal information loss pattern is the same as for the no tree construction case.

The number of scenarios in the reduced ensemble is incrementally reduced from 50 to 1. For each case V is calculated, yielding the marginal information loss curve.

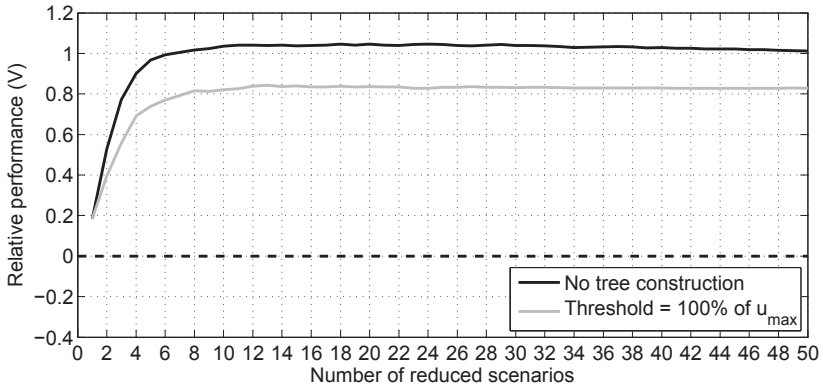
Three configurations are tested to see if the marginal information loss is consistent for different configurations. The base case is used, followed by a configuration with an equal T_o, but at the same time significantly further away from



(a) System characteristics are $\frac{A_{\text{storage}}}{A_{\text{system}}} = 0.04$ and $\frac{u_{\max}}{A_{\text{system}}} = 0.0005$.



(b) System characteristics are $\frac{A_{\text{storage}}}{A_{\text{system}}} = 0.01$ and $\frac{u_{\max}}{A_{\text{system}}} = 0.0002$.



(c) System characteristics are $\frac{A_{\text{storage}}}{A_{\text{system}}} = 0.01$ and $\frac{u_{\max}}{A_{\text{system}}} = 0.0003$.

Figure 4.6: Relative performance as a function of the number of reduced scenarios. The first configuration is closer to certainty equivalence than the other two. The influence of the tree construction can be seen by comparing the two lines in each figure.

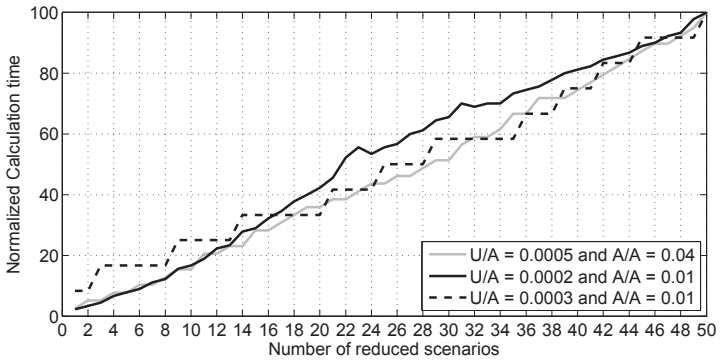


Figure 4.7: Normalized calculation time plotted against the number of reduced scenarios for the three tested configurations (threshold equals 100% of maximum pump capacity). Normalized in this case means that the largest value for each series is set to 100 and the other values are expressed as a fraction of that.

certainty equivalence. The last configuration is comparable in terms of certainty equivalence to the second configuration, but T_{o} is half of that of the second configuration. This last configuration is chosen to determine if the length of the ensemble has influence on the marginal information loss. An overview of the three configurations is shown below:

1. $\frac{A_{\mathrm{storage}}}{A_{\mathrm{system}}} = 0.04$, $\frac{u_{\mathrm{max}}}{A_{\mathrm{system}}} = 0.0005$ and $T_{\mathrm{o}} = 30$ (base case).
2. $\frac{A_{\mathrm{storage}}}{A_{\mathrm{system}}} = 0.01$, $\frac{u_{\mathrm{max}}}{A_{\mathrm{system}}} = 0.0002$ and $T_{\mathrm{o}} = 30$.
3. $\frac{A_{\mathrm{storage}}}{A_{\mathrm{system}}} = 0.01$, $\frac{u_{\mathrm{max}}}{A_{\mathrm{system}}} = 0.0003$ and $T_{\mathrm{o}} = 15$.

T_{ip} is kept constant at 15 hours for all three configurations, to generate comparable ensembles. The inflow series from Figure 4.1 is used again. The results are shown in Figure 4.6. Note that V with one scenario is the same for both cases, since one cannot create a tree from only one scenario.

Configurations two and three show an increasing relative performance with additional scenarios. For both systems the added value of more scenarios flattens off after about eight to 10 scenarios. For a configuration closer to certainty equivalence (i.e. base case) the pattern seen at the other two configurations is non existent. Additional scenarios here do not have any value, they merely increase calculation time. This is illustrated in Figure 4.7 where the linear relationship between the calculation time and the number of scenarios is shown.

For configurations two and three the marginal information loss pattern is the same for the 0% and the 100% case, however, the latter has a slightly lower relative performance. These two configurations are far away from certainty equivalence, thus additional information is aggregative to the relative performance. Configuration three has an T_{o} of half that of the second configuration, however,

	Average water level	Variance of water level
TBMPC	0.0024 m	3.4E-04 m^2
MPC	-0.0050 m	5.7E-04 m^2

Table 4.1: Characteristics of the complete simulation for TBMPC with one scenario and MPC, for the base case configuration.

the marginal information loss pattern is the same. It is thus concluded that the length of the ensemble itself has no influence on the marginal information loss.

The base case configuration displays the same phenomenon that was observed in Figure 4.3. A high number of scenarios combined with no tree construction has a negative influence on the relative performance for configurations that are close to certainty equivalence. A detailed analysis of this effect is given in paragraph 4.3 and thus not repeated here. The base case configuration results, however, clearly show this effect again. With an increasing number of scenarios a downward trend in relative performance for the case without tree construction can be seen. As argued in paragraph 4.3 this trend is eliminated when a threshold value of 100% of the maximum pump capacity is used.

From the situations with only one scenario the added benefit of the scenario reduction algorithm can be derived. In configuration one TBMPC performs significantly better than MPC, although both, in this case, are a deterministic optimization. Choosing the most representative scenario instead of a random one has its benefits. This can be explained be a more volatile and slightly less optimal average water level for MPC case. Table 4.1 shows the variance and the mean of the water level for TBMPC with one scenario and MPC. The water level is more volatile in the MPC case and the absolute distance from set point is larger as well. With a quadratic cost function, both lead to higher costs. The increase in costs is however small compared to costs that yield from an upward exceedance of the normal operating band. This is the reason why this difference only has an effect on the relative performance for configuration one. Total costs are small for the base case configuration and the phenomenon can have an effect on the relative performance, but in a situation where total costs are a few magnitudes larger and dominated by costs associated with an upward exceedance of the normal operating band the effects of this phenomenon do not show up in the relative performance value (see Table 4.2). Hence the added benefit of the scenario reduction algorithm does not show up for configurations two and three, it is however present there as well.

The results for the third configuration show a relative performance larger than one with more than six scenarios and no tree construction. This is the result of the selection of T_o. The formula derived in Appendix D was used to determine it, but this formula assumes it is not necessary to lower the water level

	Total costs	Upward exceedance [%]
Configuration 1	39	0.75
Configuration 2	183188	98
Configuration 3	40504	97

Table 4.2: Overview of the absolute amount of the total costs and the percentage of the total costs that are generated by upward exceedances of the normal operating band.

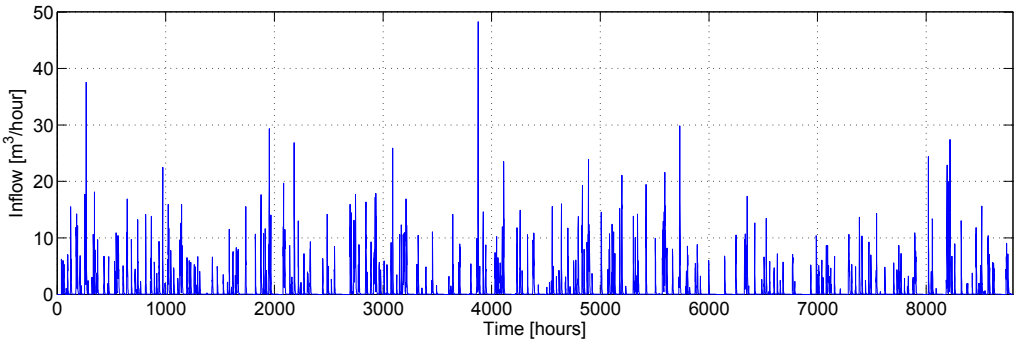


Figure 4.8: Inflow series used for water system characteristics experiments. Average inflow of 0.92 m^3/hour. Maximum inflow of 48.27 m^3/hour. Total inflow of 8141 m^3.

below the normal operating band. For a system this far away from certainty equivalence, this could however be necessary. More time is needed to lower the water level below the normal operating band and thus T_o should be increased to get the optimal control actions. MPC with a perfect forecast causes not enough lowering of the water level in this case. The ensemble without creating a tree, however, overestimates the inflow (see Figure 4.5) and this partly compensates the too short T_o. This yields a relative performance above 1. For TBMPC with 10 scenarios, MPC and MPC with a perfect forecast the simulations were re-done with higher T_o values. The performance of TBMPC and MPC did not improve much, however, that of MPC with a perfect forecast increased about 30% when T_o was increased from 15 to 20 hours. The relative performance for TBMPC with 10 scenarios and no tree construction became about 0.85. Even higher values for T_o did not lower V any further.

The above experience leads to the use of a safety factor in equation D.1. When using this equation in practice multiplying T_{ic} with a factor 1.3 to get T_o is advised. Due to calculation time constraints in this research equation D.1 is used as presented in Appendix D.

4.5 Experiment 4: Influence of water system characteristics

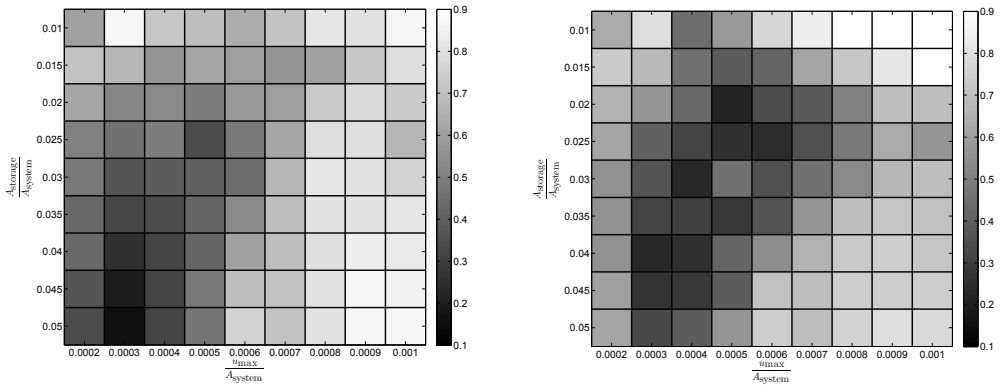
The last objective stated in Chapter 1 is to give insight on how the improvement of TBMPC over MPC varies with different system characteristics. This experiment tries to achieve that by investigating the two main water system characteristics on the following ranges:

- Storage capacity of the system $\left(\frac{A_{\text{storage}}}{A_{\text{system}}}\right)$: from 0.01 up to 0.05.
- Discharge capacity of the system $\left(\frac{u_{\max}}{A_{\text{system}}}\right)$: from 0.0002 up to 0.001.

The experiment is done for all combinations of these two parameters. The other parameters are set to the base case values, including the ones derived during the previous three experiments. A threshold value of 100% of the maximum pump capacity performs well for different configurations. From the marginal information loss experiments it became clear that between eight to 10 scenarios led to the most added benefit with the least amount of added calculation time. Given the wide variety of configurations and the desire to keep calculation time as limited as possible all simulations for this experiment will be done with eight scenarios and a threshold value of 100% of the maximum pump capacity. To isolate the influence of the storage area and the pump capacity the ensembles are the same for all configurations. T_{ip} is again kept constant at 15 hours. This is T_{ip} that is associated with the base case configuration ($\frac{A_{\text{storage}}}{A_{\text{system}}} = 0.04$ and $\frac{u_{\max}}{A_{\text{system}}} = 0.0005$ and $T_{\text{ip}} = 50\%$ of T_{ic}). All experiments are done under the forcing of the inflow series shown in Figure 4.8.

The hypothesis is that systems closer to certainty equivalence will show less improvement in performance by going from MPC to TBMPC. For a system close to certainty equivalence the average prediction suffices (for a more elaborate definition of the certainty equivalence principle, see paragraph 3.1). Randomly choosing one of the 50 scenarios of the prediction over a long period results in the average prediction. MPC should thus work better on systems close to certainty equivalence. The improvement of TBMPC will be smallest here and larger for the systems further away from certainty equivalence. The results are shown in Figure 4.9a as a grey shaded matrix to get a clear overview of the results.

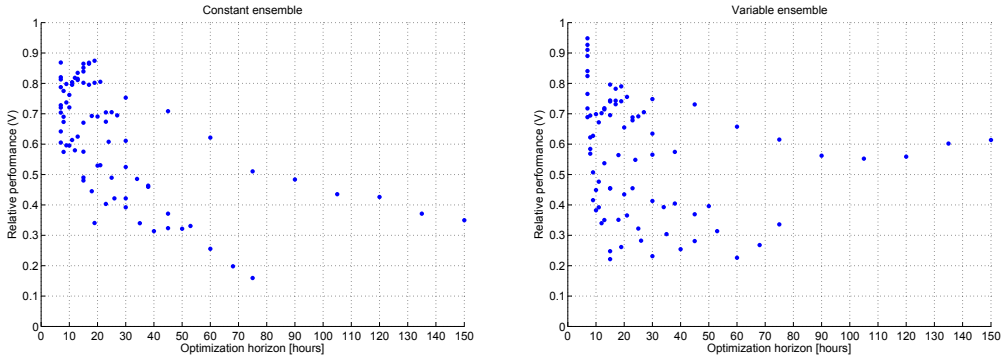
The hypothesis does not seem to hold. Configurations far from certainty equivalence (i.e. north west in the plot) show a positive relative performance, but configurations close to certainty equivalence (i.e. south east in the plot) show the highest relative performance. When plotting the relative performance against T_{o} (Figure 4.10a) it looks like if a longer T_{o} leads to a lower relative performance. Since T_{ip} is constant for all configurations it could be that this is caused by the fact that the quality of the forecast on the entire T_{o} is different for



(a) T_{ip} is 15 hours for all configurations.

(b) T_{ip} is 50% of T_o for each configuration configurations.

Figure 4.9: Relative performance for 81 different configurations. Figure 4.9a shows the results with a T_{ip} of 15 hours for all configurations. Figure 4.9b shows the results with a T_{ip} of 50% of the T_o of each configuration.



(a) T_{ip} is 15 hours for all configurations.

(b) T_{ip} is 50% of T_o for each configuration.

Figure 4.10: Relative performance plotted against the optimization horizon.

each system. Consider the configuration with $\frac{A_{storage}}{A_{system}} = 0.05$, $\frac{u_{max}}{A_{system}} = 0.0002$ and $T_o = 150$. After $\tau = 15$ the forecast is a realization of the climatic distribution. Choosing a random scenario here for a long period essentially yields the climatic mean of the forecast. Trying to distill extra information from this by deleting certain scenarios and creating a tree does not seem logical, since there is no extra information present in this part of the ensemble. To see if this is true the experiment is repeated with T_{ip} set to 50% of T_o for each configuration.

Figure 4.9b shows a different pattern than Figure 4.9a, however, systems close to certainty equivalence still show the highest relative performance as a class. Displaying the relative performance against T_o reveals that the trend of a lower relative performance with a higher T_o is somewhat diminished, but not completely gone.

It could be that the results are skewed due to the unseen component 'adaptivity'. This is the ability of the system to correct mistakes, because of its receding horizon. The adaptivity is high when only the first action is implemented and the receding horizon displaced one time step. When for example the first six actions are implemented and the receding horizon displaced six time steps the adaptivity is low (keeping all other parameters the same). A highly adaptive system can correct mistakes it made fairly easy, thus having a wrong forecast might not be such a big problem.

All configurations in the south east of Figures 4.9a and 4.9b have a short T_o (eight to 20 hours) combined with implementing the first six actions. All configurations in the south west have a long T_o (30 to 150 hours) combined with implementing the first six actions. Given the definition of adaptivity above, the latter configurations are thus highly adaptive and the former configurations are only slightly adaptive.

To see if the above hypothesis is an explanation for the results found so far the following experiment is carried out. The relative performance is investigated with varying number of actions that are implemented per optimization (i.e. the optimization time step is varied, see paragraph 3.5 for a more detailed description of this parameter). To see how this parameter influences the results of Figures 4.9a and 4.9b simulations have been done on three configurations. To be able to compare the results between these configurations, the relative performance is plotted against the ratio of the optimization time step and the optimization horizon. Results are shown in Figure 4.11.

A few conclusions can be drawn from the results. For a ratio below 0.25 indeed the relative performance increases with less adaptive configurations. Between a ratio of 0.25 and 0.6, however, the relative performance actually decreases. Yet for a ratio above 0.6 a strong upward trend in the relative performance for less adaptive configurations can be seen. The standard deviation of the relative performance increases with decreasing adaptivity. This is due to the fact that the random selection of a scenario for MPC is done less often and the actions that are determined are based on less certain data in the forecast, which results in greater difference between actions, yielding more difference between the 10 MPC simulations.

These results do not necessarily explain the earlier observed relative performance for various configurations. For example fixing the ratio at 0.2 for all configurations would have given the same results as in Figures 4.9a and 4.9b. On the other hand when the ratio would have been fixed at 0.8 or 0.9 another pattern would have emerged in the aforementioned figures. Configurations in the south west of Figure 4.9b would not have shown a much lower relative performance as compared to configurations in the south east of that figure, they would have been equal.

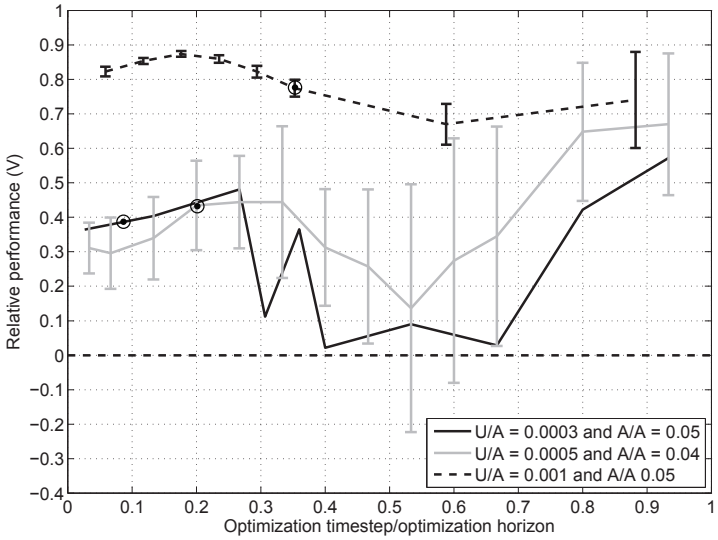


Figure 4.11: Relative performance against optimization time step for three configurations. For two configurations the standard deviation is shown. Standard deviation is not shown for the first configuration to keep the figure clear, however, it is similar in size and variation to that of the second configuration. The corresponding ratio that was used in Figures 4.9a and 4.9b is indicated by the black circles.

In paragraph 4.4 a high relative performance was shown for the base case configuration. A closer look was taken at the case with one scenario, concluding that a less volatile and closer to set point water level was the main reason for the high V. This phenomenon could also be an explanation for the high V values seen in Figure 4.9b for the configurations close to certainty equivalence. From the configurations with a $\frac{A_{\text{storage}}}{A_{\text{system}}}$ ratio larger than 0.025, a $\frac{u_{\max}}{A_{\text{system}}}$ ratio larger than 0.0005 and V above 0.65 not one has an upward exceedance of the normal operating band and total costs are small. These systems show much resemblance with configuration one of paragraph 4.4. In Table 4.3 the average and the variance of the water level for TBMPC and MPC of these configurations are shown. The results are similar to those in paragraph 4.4. The average water level is further away from set point in the case of MPC compared to TBMPC and the variance of the water level is larger in the case of MPC compared to TBMPC. These two factors are the main drivers of a positive relative performance for these configurations.

From all the results in this paragraph it is clear that the relative performance is positive for a wide variety of configurations and for some configurations even approaches the result of a deterministic optimization with a perfect forecast. A clear relationship between the relative performance and how close too or how far away from certainty equivalence a configuration is, is not found. The main

$\frac{u_{\max}}{A_{\text{system}}}$	$\frac{A_{\text{storage}}}{A_{\text{system}}}$	X_{mpc}	$X^{\text{var}}_{\text{mpc}}$	X_{tbmpc}	$X^{\text{var}}_{\text{tbmpc}}$
0.0006	0.045	-7.2E-03	6.0E-04	-1.4E-03	3.6E-04
	0.05	-6.8E-03	5.2E-04	-1.1E-03	3.1E-04
0.0007	0.04	-11.5E-03	7.4E-04	-4.2E-03	4.4E-04
	0.045	-9.6E-03	5.8E-04	-3.6E-03	3.5E-04
	0.05	-9.1E-03	5.1E-04	-3.1E-03	3.0E-04
0.0008	0.035	-14.6E-03	9.1E-04	-4.0E-03	4.9E-04
	0.04	-12.9E-03	7.4E-04	-2.8E-03	3.9E-04
	0.045	-11.6E-03	6.1E-04	-1.9E-03	3.2E-04
	0.05	-10.4E-03	5.3E-04	-1.4E-03	2.7E-04
0.0009	0.03	-18.0E-03	12.0E-04	-4.3E-03	6.3E-04
	0.035	-15.4E-03	8.9E-04	-2.6E-03	4.8E-04
	0.04	-13.5E-03	7.4E-04	-0.8E-03	3.7E-04
	0.045	-11.8E-03	6.1E-04	0.3E-03	3.0E-04
	0.05	-10.8E-03	5.3E-04	1.2E-03	2.4E-04
0.001	0.03	-19.1E-03	13.0E-04	-4.5E-03	6.7E-04
	0.035	-16.4E-03	9.7E-04	-1.8E-03	5.0E-04
	0.04	-12.3E-03	7.7E-04	0.2E-03	3.6E-04
	0.045	-11.3E-03	6.4E-04	1.7E-03	2.8E-04
	0.05	-9.7E-03	5.5E-04	2.7E-03	2.2E-04

Table 4.3: Comparison between the average water level (X) and the variance of the water level (X^{var}) for TBMPC and MPC.

reason for this is that there are multiple effects that can cause TBMPC to perform better than MPC. The hypothesis at the start of this experiment was that TBMPC would only be an improvement over MPC for configurations far away from certainty equivalence. When water levels easily exceed cost thresholds the average forecast will no longer provide good results. The results show positive relative performance values for these systems, however, it was not expected to see such values for configurations close to certainty equivalence as well. The effects of the lower water level variance and the closer to set point average water level on the relative performance were underestimated.

Chapter 5

Conclusions

In Chapter 1 control algorithms for water systems and TBMPC in particular were introduced. This overview defined four problems with TBMPC that needed further investigation, which led to four objectives. In the previous chapter it has been tried to achieve those objectives. The conclusions on these objectives are given in this chapter.

5.1 Closed loop simulation

Until this research TBMPC was only tested in open loop simulations. Throughout this research, however, the performance of TBMPC and MPC was measured in closed loop simulations. Every simulation consisted of hourly values for precipitation, inflow, water level and actions over a period of one year. This provided the ability to evaluate the performance not only for extreme situations (e.g. a design storm), but also under normal conditions. This is important, since if TBMPC reduces the damage of a certain extreme event (compared to MPC), but this reduction in costs is more than offset throughout the rest of the year, TBMPC provides little or no benefit. The performance was calculated as the total value of the objective function over the yearlong simulation. This was done for TBMPC, MPC and MPC with a perfect forecast (i.e. forecast equal to inflow), which together yield the relative performance value:

$$V = \frac{\mathrm{V_{mpc}} - \mathrm{V_{tbmpc}}}{\mathrm{V_{mpc}} - \mathrm{V_{perfect}}} \tag{5.1}$$

A value of one indicates that TBMPC performs equal to MPC with a perfect forecast and zero represents no improvement over MPC. This formula is unbounded below zero and negative values represent situations where MPC performance better than TBMPC.

The results in the previous chapter show a total of 698 relative performance values and 677 (97%) out of these are larger than zero. This shows that TBMPC performs better than MPC not only for extreme events, but also under normal operating conditions. These results consider all variations of TBMPC that have been analysed in this report, not only the optimal variations. The next two paragraphs will discuss the optimal parameters for the scenario tree construction algorithm and the number of scenarios.

5.2 Scenario tree construction algorithm

It was already acknowledged that the scenario tree construction algorithm has an influence on the performance of TBMPC. What the most optimal parameters are and if they could be set as a ratio to certain system characteristics was unknown. Therefore the pre-processing of scenarios and the influence of the aggregation threshold have been investigated.

The scenario tree construction algorithm has been applied on the original inflow scenarios and on the cumulative inflow scenarios. It was hypothesized that by using cumulative scenarios a better distinction between scenarios can be made by the algorithm. The results showed that for all water system configurations and all threshold values that were tested the cumulative scenarios indeed provided better results.

The threshold value is an important parameter in the scenario tree construction algorithm, as it determines if scenarios can be aggregated on a certain part of the ensemble. The magnitude of this value and if it needed to be a constant value, or one that varies with different system characteristics was not yet known. The hypothesis was that the threshold value needed to be a fraction of the maximum pump capacity of the system. For four different system configurations, 13 different fractions and three different number of scenarios the relative performance was determined.

A threshold value of 100% of the maximum pump capacity provided either optimal or close to the optimal relative performance and was most consistent between the four system configurations. The objective of the scenario reduction algorithm is to create a representative ensemble with the greatest reduction in calculation time. Good relative performance could indicate a representative ensemble was used, but a closer look at the ensemble itself was taken as well. It was concluded that the deletion of complete scenarios adds a positive bias to the ensemble, however, the scenario tree construction algorithm with a threshold value of 100% of the maximum pump capacity largely removed this bias. Higher values created a negative bias.

Not only gives a threshold value of 100% of the maximum pump capacity good relative performance, it reduces calculation time by 40% to 90% as compared to

no tree construction (i.e. threshold value of 0% of the maximum pump capacity). The reduction increases with increasing number of scenarios.

5.3 Number of scenarios

The scenario reduction algorithm consists of two parts, first reduce the number of scenarios in the ensemble to a predefined number and secondly create a tree from this reduced ensemble. No information was, however, available on which number of scenarios is optimal in terms of performance and calculation time.

For three different system configurations the relative performance and calculation time have been investigated. Simulations were done with one to 50 scenarios. Calculation time showed a clear linear relationship on the entire result space. The relative performance, however, did not show a linear relationship. For two configurations the relative performance increased from one to about eight to 10 scenarios, but flattened off for a higher number of scenarios. For the third configuration relative performance was flat on the range of tested number of scenarios. This configuration had no upward exceedances of the normal operating band, thus better performance did not came from better peak event prediction. TBMPC provided a more robust forecast, yielding more stable and closer to set point water levels, which resulted in lower costs compared to MPC. Contrary with better peak event anticipation, this effect already worked from one scenario on.

Combining the linear increase in calculation time and linear increase in relative performance until eight to 10 scenarios and the flat relative performance for more scenarios leads to the conclusion that eight to 10 scenarios is the optimal number of scenarios to be used with TBMPC. This does not necessarily hold for all control problems where TBMPC is used, but within the framework (e.g. water system, cost function, forecast) of this research eight to 10 scenarios is the ideal number.

5.4 Water system characteristics

The model build for this research has many parameters to define a water system, however, throughout this research two parameters have been defined as the most characteristic. The first one is the ability of the system to deal with disturbances through discharge, expressed in the ratio between maximum pump capacity and system area $\left(\frac{u_{\max}}{A_{\text{system}}}\right)$. The second one is the ability of the system to deal with disturbances through storage, expressed in the ratio between storage area and system area $\left(\frac{A_{\text{storage}}}{A_{\text{system}}}\right)$. To determine for which water system configurations TBMPC is more beneficial these two ratios have been varied and used to define the range of system configurations.

The relative performance was determined for 81 different system configurations. The aforementioned ratios ranged from 0.0002 to 0.001 and 0.01 to 0.05 respectively. It was expected that configurations with a low discharge capacity and low storage capacity would see the most benefit of TBMPC and the lowest relative performance values would be found for high values of both ratios.

Simulations were done with a constant quality of the ensemble and varying quality, by means of a varying information-prediction horizon (T_{ip}), however, the expected relationship was not found in both cases. It could be that the relationship that was expected is not found because of the hidden component of 'adaptability'. This is the ability of the system to correct mistakes it made or postpone the decision until more information comes available. A system with a long optimization horizon (T_{o}) is more adaptable than a system with a short optimization horizon (keeping all other parameters the same). MPC is expected to benefit more from a higher adaptability than TBMPC. The optimization horizon varies for the 81 different system configurations that have been investigated.

An analysis of the influence of the adaptability of the system on the relative performance shows that indeed it can have a significant influence, however, the results found so far do not provide an explanation for the failure of the hypothesis on the influence of water system characteristics on the relative performance.

A more plausible explanation is that there are two mechanisms generating positive relative performance values. The first is the better peak event anticipation of TBMPC. The second is less volatile and closer to set point average water levels. For configurations with a low discharge and storage capacity the peak event mechanism is most dominant. This was expected to generate positive relative performance values. For configurations with a high discharge and storage capacity, however, there are no upward exceedances of the normal operating band and thus better peak event anticipation is of no concern here. Hence, low positive relative performance values were expected. For these configurations, however, the second mechanism plays a dominant role and can provide high positive relative performance values as well.

It is thus concluded that TBMPC shows a significant increase in performance compared to MPC for various configurations of a linear reservoir system, however, no single relationship could be determined to express the added benefit of TBMPC for any given configuration.

Chapter 6

Recommendations

Although a large number of simulations were performed, the model that was developed for this research has the capability of investigating even more parameters that could have an influence on the relative performance. Due to time constraints a selection was made based on the most characteristic or expected to be most influential parameters. The relationship, however, between input parameters and the performance of a control algorithm can be a complex one, as was seen with the investigation of the influence of water system characteristics on the relative performance. It is therefore recommended to look further into the effect of the following parameters on the relative performance:

- Lead time between precipitation and inflow (e.g. when greenhouses can store precipitation and delay runoff to the canals) or a lead time between knowing the inflow and the inflow actually occurring (e.g. if the inflow is already known further upstream in a catchment and cannot change anymore before it reaches the controllable system).

- The effect of the area size on the pattern and the amount of the precipitation.

- The objective function. For example a more extreme cost function (e.g. higher or lower land values in a polder-belt canal system) or including other objectives (e.g. ecological impact or power generation constraints).

Besides these additional parameters it would be useful to investigate the complex relationship between adaptability, the scenario tree construction algorithm and the relative performance further. The optimization time step plays an important role in this relationship and can have a significant impact on the relative performance as was shown in paragraph 4.5.

The aggregation of the scenarios by the scenario reduction algorithm is not performed in a particular order, only based on the smallest probability weighted difference and the probability of scenarios themselves. Usually there is only one smallest distance and the two scenarios have a different probability, thus no 'order' needs to be defined. However, if two scenarios have the same probability and one of them needs to be aggregated (because their distance is the smallest of all other distances) the first one is aggregated to the second one. It could be useful to check whether or not the addition of an order in which to delete scenarios can improve the scenario reduction algorithm even further.

The possibility of branches of the tree joining again after a bifurcation point is not taken into account in the current version of the scenario reduction algorithm. The predictability decreases with time, but this is not always the case within the ensemble length. The exact amount of rain from a severe storm in four days could be uncertain, but a clear period after this storm could be very certain again. Allowing branches to rejoin after a bifurcation point could thus be a useful improvement to the algorithm.

All these simulations were done with a linear reservoir as a model of the belt canals of a polder-belt canal system. This is of course an approximation of the real world. A verification of the results in this research can be obtained by implementing both TBMPC and MPC into a real polder-belt canal system, or a very close approximation, like a detailed SOBEK model of such a system. This will also result in insight on whether the calculation time needed for each optimization for TBMPC is actually available within these systems.

The results presented in this report required a vast number of simulations and calculation time (approximately 3200 hours). The function *fmincon* was used in MATLAB, however, commercial packages (e.g. Tomlab) are available that can serve as a replacement for *fmincon*, but require significant less calculation time. Buying such a package and altering the MATLAB code to be compatible with it could prove to be a good investment.

Bibliography

[1] A. M. Feyerherm and L. Dean Bark. Statistical methods for persistent precipitation patterns. *Journal of Applied Meteorology*, 4(3):320–328, 1965. [cited at p. 10]

[2] N. Gröwe-Kuska, H. Heitsch, and W. Römisch. Scenario reduction and scenario tree construction for power management problems. In *IEEE Bologna Power Tech Proceedings*, pages 1980–81, 2003. [cited at p. 20]

[3] R.P. Hooper, A. Stone, N. Christophersen, E. de Grosbois, and H.M. Seip. Assessing the Birkenes model of stream acidification using a multisignal calibration methodology. *Water Resources Research*, 24(8):1308–1316, 1988. [cited at p. 14, 32]

[4] G.M. Hornberger, K.J. Beven, B.J. Cosby, and D.E. Sappington. Shenandoah watershed study: calibration of a topography-based, variable contributing area hydrological model to a small forested catchment. *Water Resources Research*, 21(12):1841–1850, 1985. [cited at p. 14, 32]

[5] R.W. Katz. Precipitation as a chain-dependent process. *Journal of Applied Meteorology*, 16(7):671–676, 1977. [cited at p. 10]

[6] R. Krzysztofowicz. The case for probabilistic forecasting in hydrology. *Journal of Hydrology*, 249(1-4):2–9, 2001. [cited at p. 3]

[7] K.M. Loague and R.A. Freeze. A comparison of rainfall-runoff modeling techniques on small upland catchments. *Water Resources Research*, 21(2):229–248, 1985. [cited at p. 14, 32]

[8] C.R. Philbrick and P.K. Kitanidis. Limitations of deterministic optimization applied to reservoir operations. *Journal of Water Resources Planning and Management*, 125(3):135–142, 1999. [cited at p. 25]

[9] L. Raso, P.J. Van Overloop, and D. Schwanenberg. Decisions under uncertainty: Use of flexible model predictive control on a drainage canal system. In *Proceedings of the 9th International Conference on Hydroinformatics*, Tianjin, China, 2009. HIC. [cited at p. 3, 4, 20]

[10] E. Roulin. Skill and relative economic value of medium-range hydrological ensemble predictions. *Hydrology and Earth System Sciences Discussions*, 3(4):1369–1406, 2006. [cited at p. 3]

[11] B. Schultz. Water management and flood protection of the polders in the Netherlands under the impact of climate change and man-induced changes in land use. *Journal of Water and Land Development*, 12:71–94, 2008. [cited at p. 1]

[12] W. Schuurmans, P.J. van Overloop, and P. Beukema. Delfland kiest voor volledige automatische bediening van kunstwerken. *H2O (in Dutch)*, 2003. [cited at p. 6]

[13] H.C.S. Thom. A note on the gamma distribution. *Monthly Weather Review*, 86(4):117–122, 1958. [cited at p. 11]

[14] SJ Van Andel. *Anticipatory Water Management: Using ensemble weather forecasts for critical events*. CRC Press/Balkema, 2009. [cited at p. 5, 34]

[15] Hoogheemraadschap van Delfland. *Jaarrekening 2009 van het Hoogheemraadschap van Delfland*. Hoogheemraadschap van Delfland, 2009. [cited at p. 34]

[16] P.J. van Overloop. *Model predictive control on open water systems*. IOS Press, 2006. [cited at p. 6]

[17] S.V. Weijs, P.E.R.M. van Leeuwen, P.J. van Overloop, and N.C. van de Giesen. Effect of uncertainties on the real-time operation of a lowland water system in The Netherlands. *IAHS PUBLICATION*, 313:463, 2007. [cited at p. 11, 31]

Appendices

Appendix A

Scenario reduction algorithm

A.1 Part 1: Numerical example of backward reduction component

The scenario reduction algorithm is explained in paragraph 2.3.4. A numerical example of the algorithm is given here for illustrative purposes.

The initial inflow ensemble for this example consists of 6 scenarios. Through 3 steps this is reduced to 3 scenarios. For each step the distance matrix, minimum distance vector (dVec), probability vector (pVec) and probability distance vector (distance multiplied with the probability, cVec) are shown. Red marks the scenario that is aggregated and blue marks to which scenario it is aggregated to. The distance in this example is defined as the average difference between two scenarios:

$$D = \frac{\sum_{i=1}^{n} |I_{k,i} - I_{j,i}|}{n} \tag{A.1}$$

Where i is the index of the node, n the length over which the distance is calculated and $I_{k,i}$ and $I_{j,i}$ the scenarios that are compared.

	1	2	3	4	5	6
1	Inf	2.45	4.81	6.62	12.61	10.19
2	2.45	Inf	3.72	5.62	11.52	8.77
3	4.81	3.72	Inf	3.17	7.80	7.17
4	6.62	5.62	3.17	Inf	6.17	4.51
5	12.61	11.52	7.80	6.17	Inf	6.94
6	10.19	8.77	7.17	4.51	6.94	Inf

Table A.1: Distance matrix before step 1.

1	2.45
2	2.45
3	3.17
4	3.17
5	6.17
6	4.51

1	0.17
2	0.17
3	0.17
4	0.17
5	0.17
6	0.17

1	0.41
2	0.41
3	0.53
4	0.53
5	1.03
6	0.75

Table A.2: dVec, pVec and cVec before step 1

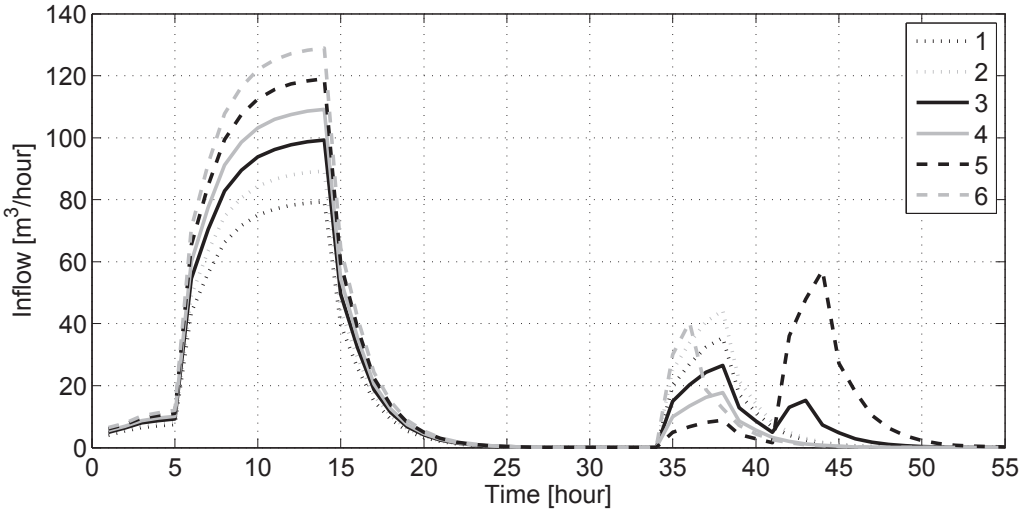


Figure A.1: The original ensemble with 6 scenarios.

	2	3	4	5	6
2	Inf	3.72	5.62	11.52	8.77
3	3.72	Inf	3.17	7.80	7.17
4	5.62	3.17	Inf	6.17	4.51
5	11.52	7.80	6.17	Inf	6.94
6	8.77	7.17	4.51	6.94	Inf

Table A.3: Distance matrix after step 1.

2	3.72
3	3.17
4	3.17
5	6.17
6	4.51

2	0.33
3	0.17
4	0.17
5	0.17
6	0.17

2	1.24
3	0.53
4	0.53
5	1.03
6	0.75

Table A.4: dVec, pVec and cVec after step 1.

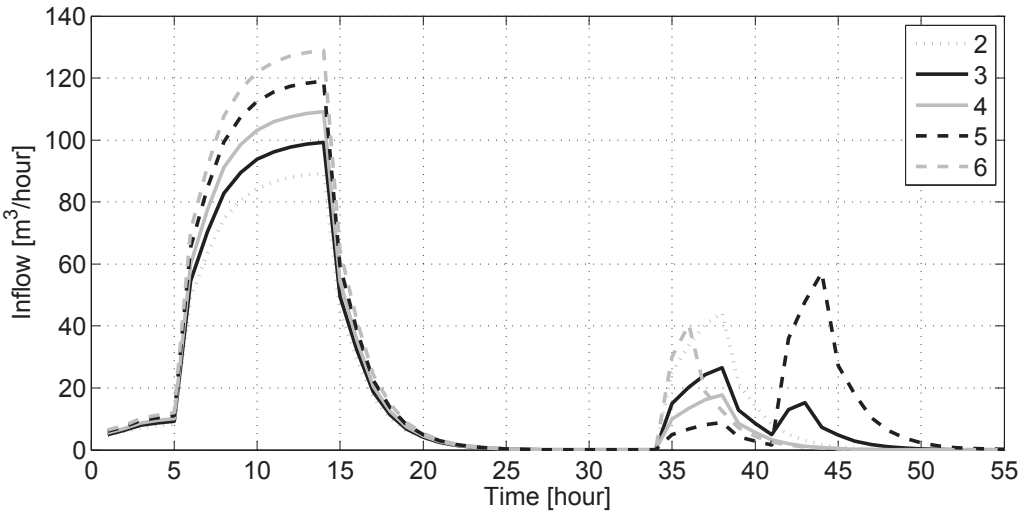


Figure A.2: Ensemble after step 1.

	2	4	5	6
2	Inf	5.62	11.52	8.77
4	5.62	Inf	6.17	4.51
5	11.52	6.17	Inf	6.94
6	8.77	4.51	6.94	Inf

Table A.5: Distance matrix after step 2.

2	5.62
4	4.51
5	6.17
6	4.51

2	0.33
4	0.33
5	0.17
6	0.17

2	1.87
4	1.50
5	1.03
6	0.75

Table A.6: dVec, pVec and cVec after step 2

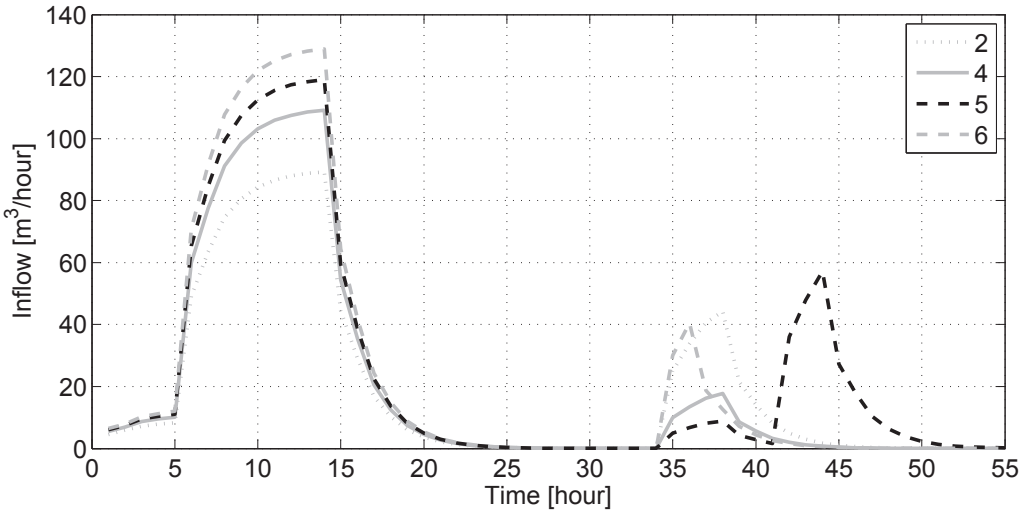


Figure A.3: Ensemble after step 2.

	2	4	5
2	Inf	5.62	11.52
4	5.62	Inf	6.17
5	11.52	6.17	Inf

Table A.7: Distance matrix after step 3.

2	5.62
4	5.62
5	6.17

2	0.33
4	0.5
5	0.17

2	1.87
4	2.81
5	1.03

Table A.8: dVec, pVec and cVec after step 3.

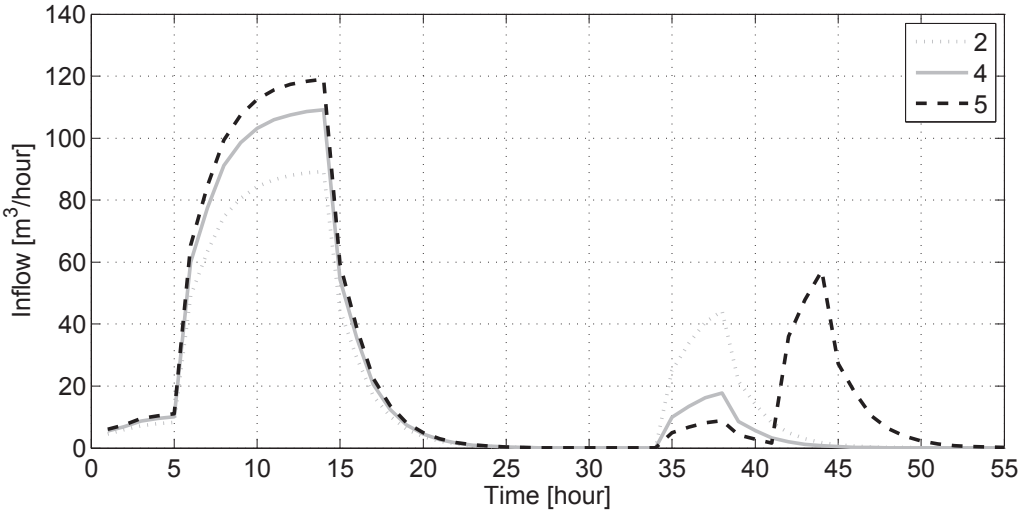


Figure A.4: Ensemble after step 3.

A.2 Part 2: Numerical example of tree construction

The scenario reduction algorithm reduced the original ensemble to an ensemble with 3 scenarios.

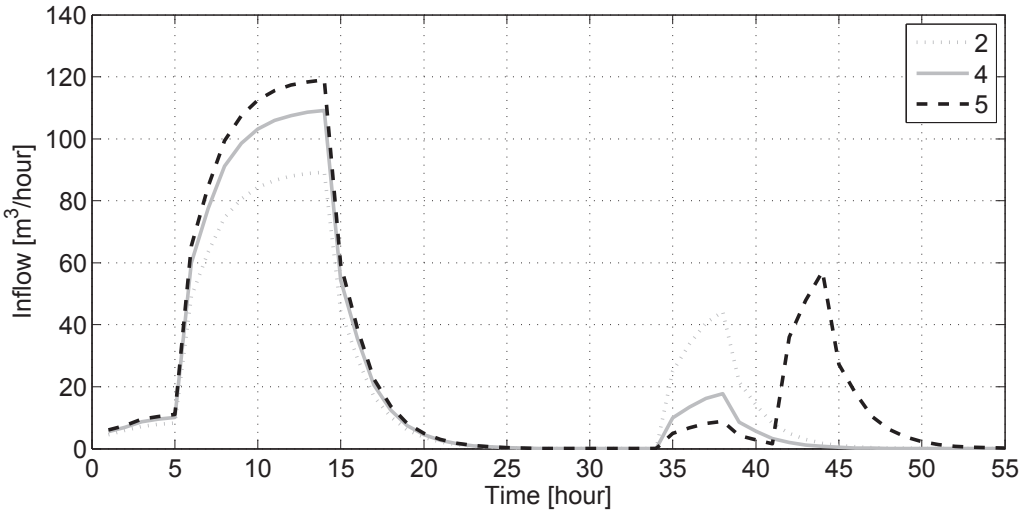


Figure A.5: Ensemble after the first part of the scenario reduction algorithm.

The scenario tree construction algorithm is applied to this reduced ensemble. If the distance between scenarios is smaller than the threshold value, it will aggregate these scenarios. As explained in paragraph 2.3.4 this is done with a step wise backward strategy. The threshold value is set to 4 m^3/hour. Only the final result is shown.

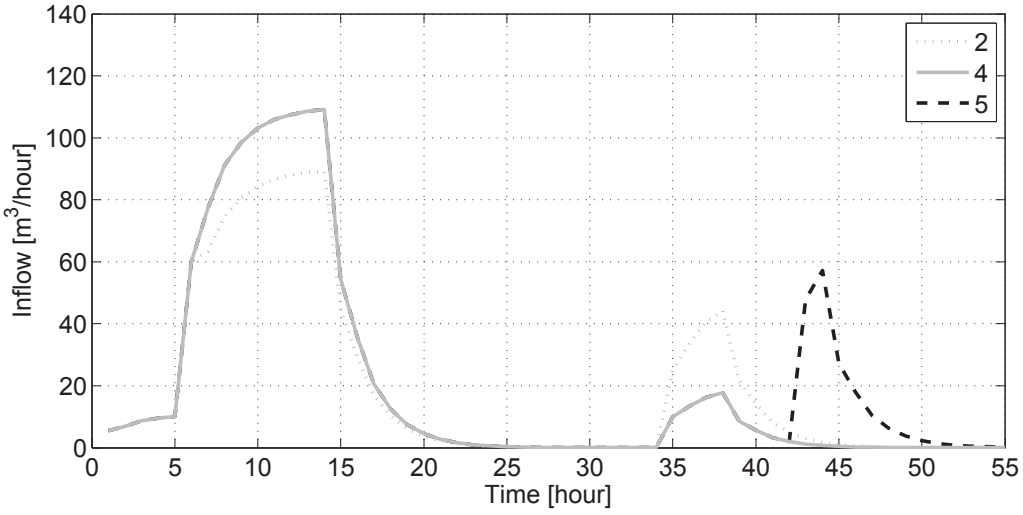


Figure A.6: Tree shaped ensemble after scenario tree construction algorithm is applied to the ensemble from figure A.5.

Appendix B

Analysis of the daily precipitation in De Bilt from 1906 - 2010

Daily average precipitation measurements from the De Bilt were analyzed. The Royal Netherlands Meteorological Institute (KNMI) has these measurements available from 1906. A time series from 01-01-1906 till 30-09-2010 was used for this analysis.

The average daily precipitation over this period was 2.45 mm with a standard deviation of 4.75 mm. A gamma distribution was fitted to the dataset, to find daily precipitation amounts for certain return periods. This was done by making use of the GAMMAINV function in MS Excel. This function requires a value for input parameters α and β, which can be derived from the mean and variance of the dataset, through the following equations:

$$\text{Mean} = \alpha \times \beta \tag{B.1}$$

$$\text{Variance} = \alpha * \beta^2 \tag{B.2}$$

In Table B.1 various amounts and corresponding return periods can be found.

Precipitation amount [mm/day]	Return period (/year)
33	1
52	10
71	100

Table B.1: Daily precipitation amounts with corresponding return periods based on daily precipitation values from 1906 - 2010.

Appendix C

Calibration parameters for the precipitation and forecast generator

The precipitation generator and the forecast generator can easily be calibrated to any desirable historical record by only defining two parameters: (1) the mean of the exponential distribution (E_{clim})and (2) the transition matrix (C). The first parameter can be derived by calculating the total amount of precipitation within a certain historical record and dividing it with the number of non-zero values.

To create the transition matrix the occurrence of four different sequences within the historical record needs to be determined:

- P_{t+1} is larger than 0 and P_t is larger than 0 (1).
- P_{t+1} is equal to 0 and P_t is larger than 0 (2).
- P_{t+1} is larger than 0 and P_t is equal to 0 (3).
- P_{t+1} is equal to 0 and P_t is equal to 0 (4).

The transition matrix than becomes:

$$C = \begin{bmatrix} \frac{\#1}{\#1+\#2} & \frac{\#2}{\#1+\#2} \\ \frac{\#3}{\#3+\#4} & \frac{\#4}{\#3+\#4} \end{bmatrix} \tag{C.1}$$

Where # represents the number of occurrences within the historical record.
To check if the values of the transition matrix are accurate, the steady state vector belonging to this transition matrix can be calculated. This vector represents the probabilities of wet and dry on all time steps, and is independent of the value of

the first time step. The steady state vector $\begin{bmatrix} q_1 & q_2 \end{bmatrix}$ is derived in the following way:

$$q = \lim_{\tau \to \infty} P_t \tag{C.2}$$

Since q is independent from the initial conditions it must be unchanged when transformed by C. Since this makes it an eigenvector it can be derived in the following way from C:

$$C = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix} \tag{C.3}$$

$$qC = q \tag{C.4}$$

$$qC = qI \tag{C.5}$$

$$q(C - I) = 0 \tag{C.6}$$

$$\begin{bmatrix} q_1 & q_2 \end{bmatrix} \left(\begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix} - \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right) = 0 \tag{C.7}$$

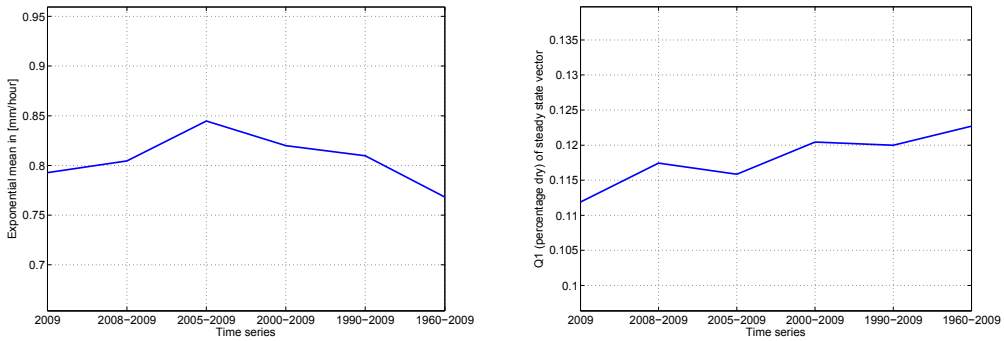
$$\begin{bmatrix} q_1 & q_2 \end{bmatrix} \begin{bmatrix} C_{11} - 1 & C_{12} \\ C_{21} & C_{22} - 1 \end{bmatrix} = 0 \tag{C.8}$$

Since q_1+q_2 equals 1 this can be solved and yields:

$$\begin{bmatrix} 1 - q_2 & \frac{1-C_{11}}{1-C_{11}+C_{21}} \end{bmatrix} \tag{C.9}$$

These values can be checked with the number of non-zero and zero values within the historical record. q_1 equals the percentage of wet values and q_2 the percentage of dry values.

For this research values representative for the Dutch climate are used. These have been derived from 6 historical time series at the De Bilt, provided by the KNMI (Royal Dutch Meteorological Institute). The 6 time series all consist of hourly values, however over different measurement periods: 2009, 2008 - 2009, 2005 - 2009, 2000 - 2009, 1990 - 2009 and 1960 - 2009. The above procedure has been executed for all these time series. The exponential mean and q_1 (percentage of wet values) of the steady state vector are plotted in Figure C.1a and Figure C.1b respectively.



(a) Mean of historic precipitation time series. (b) q_1 of historic precipitation time series.

Figure C.1: Historic precipitation series analysis for 'De Bilt'.

Both values do not converge to a particular value, implying that the values of the time series of the year 2009 are not biased because of the shorter measurement period. Therefor the values of the most recent time series will be taken, as these should be closer to current climatic conditions than those of longer measurement periods. All specifications of the 2009 time series are found in Table C.1.

Measurement station	260 De Bilt
Period	1^{st} of January 2009 - 31^{st} of December 2009
Time step	1 hour
E_{clim}	0.7928 mm/hour
C	$\begin{bmatrix} 0.6755 & 0.3245 \\ 0.0409 & 0.9591 \end{bmatrix}$
q	$\begin{bmatrix} 0.1119 & 0.8881 \end{bmatrix}$

Table C.1: Historic precipitation time series specifications.

Appendix D

Experiment: Information-control horizon

The information-control horizon (T_{ic}) is the horizon after which information does not influences the decision at the first time step anymore. All available information on the time span before T_{ic} does have an influence on the decision. Setting a correct optimization horizon is thus important, because:

1. An optimization horizon smaller than T_{ic} negatively influences the performance of both MPC and FMPC, because information further in the future would have influenced the control action at the first time step.

2. An optimization horizon that is longer than T_{ic} only increases calculation time, but does not increase performance.

One can theoretically argue that the T_{ic} of any system will be in the order of magnitude of the dynamic of the system, which can be defined as the time needed to lower the water level from the upper boundary of the normal operating band to the lower boundary. In other words this is the time needed to make all the storage in the system available for a future precipitation event. The objective of this experiment is to find a relation between the dynamic of the system and T_{ic}. By using this relation during the remainder of this research a correct T_{ic} will be used in any case.

Consider a polder belt-canal system of 1 ha with a pump capacity of 7.5 m^3/hour ($\frac{u_{\max}}{A_{\text{system}}} = 0.00075$), a normal operating band (h_{normal}) of 30 centimeters and a ratio between storage area and system area of 0.04 ($\frac{A_{\text{storage}}}{A_{\text{system}}} = 0.04$). Total storage volume is thus 120 m^3. It than takes 16 hours to make all the storage of this system available. This assumes however that no precipitation occurs in the period before the precipitation event that needs anticipatory actions. By adding

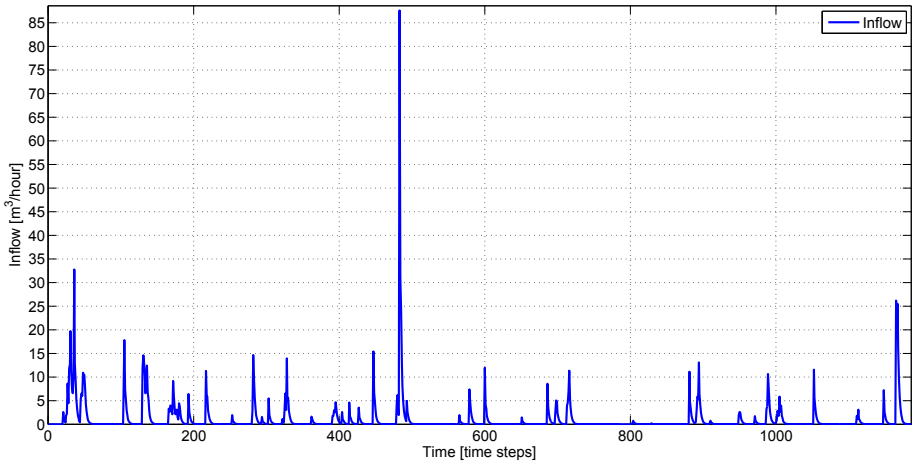
a 'typical' precipitation intensity over these 16 hours a more accurate value for the available pump capacity can be found. In Appendix B daily precipitation amounts from 1906 till 2010 measured at De Bilt have been analyzed. The average daily precipitation amount is 2.45 mm. This yields an intensity of around 0.102 mm/hour, which corresponds to an inflow to the belt canals of around 1 m^3/hour. Assuming this intensity is present during the anticipatory period, the pump capacity during this period is thus reduced to around 6.5 m^3/hour. The anticipatory period becomes around 18 hours in this case. The described method can mathematically be defined as:

$$T_{\text{ic}} = \frac{h_{\text{normal}} \times A_{\text{system}} \times \frac{A_{\text{storage}}}{A_{\text{system}}}}{\left(\frac{u_{\max}}{A_{\text{system}}} \times A_{\text{system}}\right) - C} \tag{D.1}$$

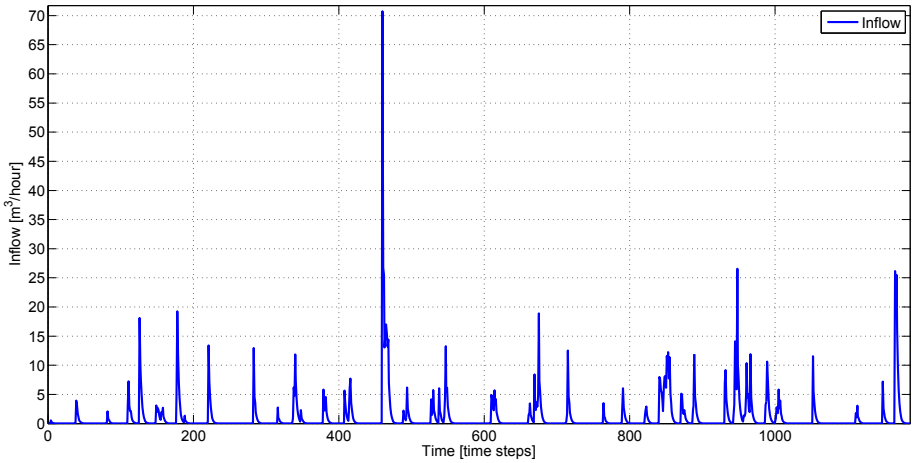
where h_{normal} is the normal operating band in meters, T_{ic} the information-control horizon in hours, C the 'typical' inflow coefficient, 1 m^3/hour in this case and A_{system} the system area in m^2.

To validate this formula the performance with varying optimization horizons has been tested. The optimization horizon after which the performance does not improve anymore is regarded as the T_{ic}. The simulations are done for the three different pump capacities (2, 3 and 7 m^3/hour). To avoid biased results three different precipitation time series were generated which yield three different inflow time series (see figures D.1a, D.1b and D.1c). The time series are 1186 hours long and the control algorithm performed on the first 720 hours (simulation horizon of 720 hours). The information-prediction horizon (T_{ip}) is kept constant at 144 hours for all simulations. In this way there is always to same amount (and quality) of information available and how much of that is used is only dependent on the T_{ic}. The number of scenarios in the reduced ensemble was 5. All other parameters are set to the base case values.

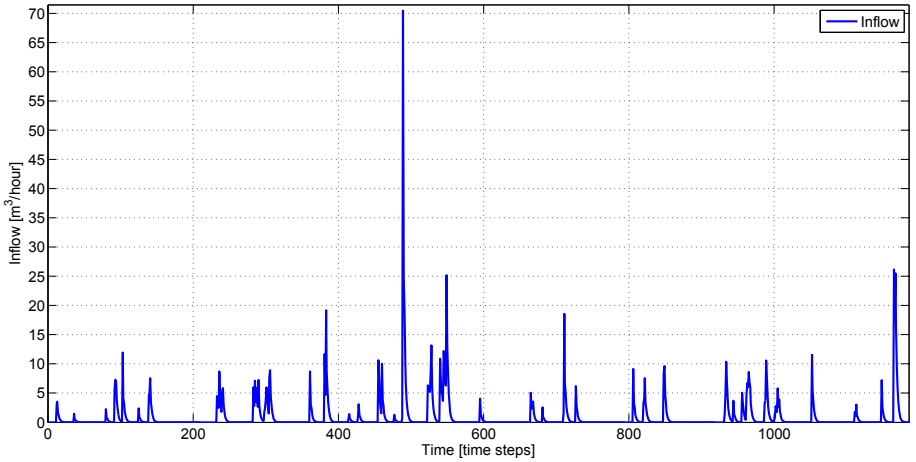
Figure D.2 shows the result of the simulations. Corresponding values are displayed in Table D.1. From these figures it is clear that equation D.1 works in practice as well. It will thus be used for the remainder of this research.



(a) Inflow time series 1. Average inflow of 1.10 m^3/hour. Maximum inflow of 87.64 m^3/hour. Total inflow of 1309.40 m^3.

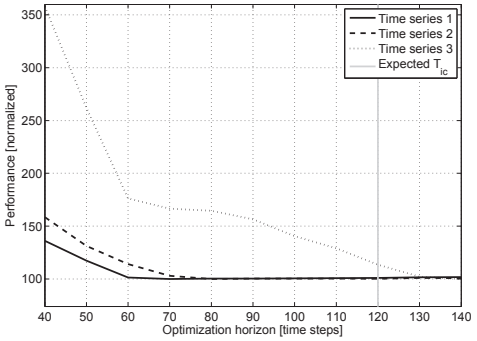


(b) Inflow time series 2. Average inflow of 1.17 m^3/hour. Maximum inflow of 70.73 m^3/hour. Total inflow of 1393.20 m^3.

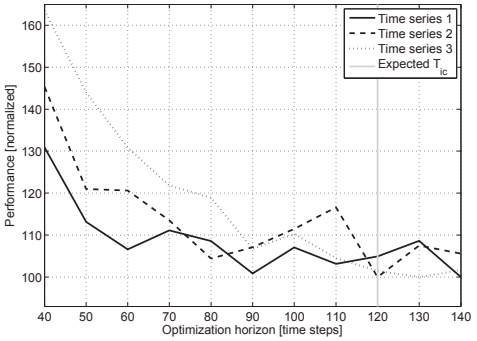


(c) Inflow time series 3. Average inflow of 1.01 m^3/hour. Maximum inflow of 70.48 m^3/hour. Total inflow of 1192.90 m^3.

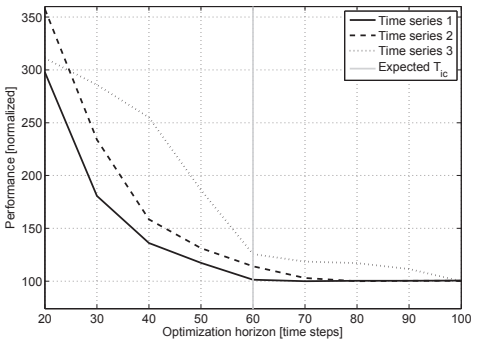
Figure D.1: Inflow time series.



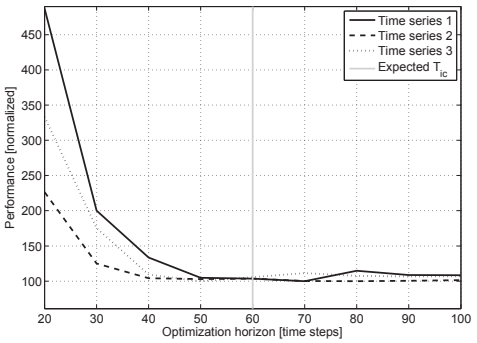
(a) MPC, $\frac{u_{\max}}{A_{\text{system}}}$ equals 0.0002 and the expected T_{ic} equals 120.



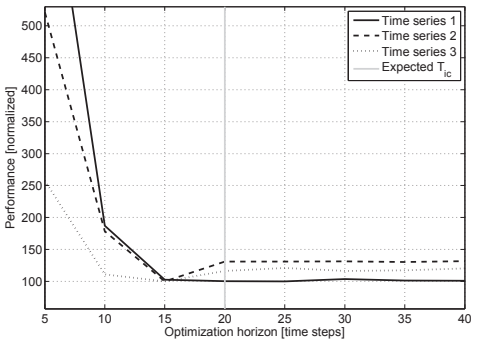
(d) TBMPC, $\frac{u_{\max}}{A_{\text{system}}}$ equals 0.0002 and the expected T_{ic} equals 120.



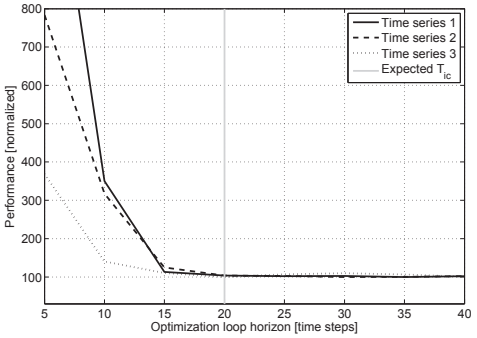
(b) MPC, $\frac{u_{\max}}{A_{\text{system}}}$ equals 0.0003 and the expected T_{ic} equals 60.



(e) TBMPC, $\frac{u_{\max}}{A_{\text{system}}}$ equals 0.0003 and the expected T_{ic} equals 60.



(c) MPC, $\frac{u_{\max}}{A_{\text{system}}}$ equals 0.0007 and the expected T_{ic} equals 20.



(f) TBMPC, $\frac{u_{\max}}{A_{\text{system}}}$ equals 0.0007 and the expected T_{ic} equals 20.

Figure D.2: Performance of MPC and TBMPC as a function of the optimization horizon.

	T1	T2	T3
40	136.15	158.48	357.96
50	117.28	131.21	261.71
60	101.35	114.05	176.25
70	100.00	103.06	166.44
80	100.29	100.00	164.53
90	100.28	100.30	156.42
100	100.48	100.43	140.42
110	100.77	100.25	129.03
120	100.97	100.20	113.46
130	101.46	100.88	102.19
140	101.76	100.69	100.00

(a) MPC, $\frac{u_{\max}}{A_{\text{system}}}$ equals 0.0002.

	T1	T2	T3
20	297.68	357.38	311.05
30	180.75	233.94	285.88
40	136.15	158.48	254.92
50	117.28	131.21	186.37
60	101.35	114.05	125.52
70	100.00	103.06	118.53
80	100.29	100.00	117.17
90	100.28	100.30	111.39
100	100.48	100.43	100.00

(b) MPC, $\frac{u_{\max}}{A_{\text{system}}}$ equals 0.0003.

	T1	T2	T3
5	819.81	521.07	257.52
10	187.04	178.35	110.96
15	102.67	100.00	100.00
20	100.33	130.91	116.30
25	100.00	130.93	120.85
30	103.45	131.43	116.23
35	101.30	130.37	117.22
40	100.96	131.53	120.19

(c) MPC, $\frac{u_{\max}}{A_{\text{system}}}$ equals 0.0007.

	T1	T2	T3
40	130.86	145.32	163.57
50	113.17	120.97	143.98
60	106.62	120.60	130.83
70	111.11	113.49	121.83
80	108.56	104.38	118.89
90	100.83	107.12	106.70
100	107.06	111.46	110.25
110	103.16	116.59	104.51
120	104.93	100.00	101.47
130	108.64	107.51	100.00
140	100.00	105.58	101.66

(d) TBMPC, $\frac{u_{\max}}{A_{\text{system}}}$ equals 0.0002.

	T1	T2	T3
20	487.00	226.40	333.11
30	200.09	125.15	175.17
40	133.54	104.27	109.15
50	104.68	102.84	100.00
60	103.45	103.52	105.67
70	100.00	100.22	111.37
80	114.70	100.00	107.56
90	108.55	100.62	106.51
100	108.45	101.36	106.90

(e) TBMPC, $\frac{u_{\max}}{A_{\text{system}}}$ equals 0.0003.

	T1	T2	T3
5	1398.77	783.77	368.49
10	350.32	317.40	140.58
15	113.03	125.09	110.58
20	103.83	103.81	100.00
25	101.96	102.51	106.88
30	102.53	100.00	109.85
35	100.00	100.45	106.68
40	101.81	102.16	100.88

(f) TBMPC, $\frac{u_{\max}}{A_{\text{system}}}$ equals 0.0007.

Table D.1: Normalized performance values of MPC and TBMPC as a function of the optimization horizon.

Appendix E

Data on water levels and cost components for Experiment 2

	0%	100%
7	-0.0126	-0.0027
20	-0.0116	-0.0026
50	-0.0029	-0.0023

(a) $\frac{A_{\text{storage}}}{A_{\text{system}}} = 0.02$ and $\frac{u_{\max}}{A_{\text{system}}} = 0.0005$.

	0%	100%
7	0.0018	0.0011
20	0.0066	0.0029
50	0.0217	0.0031

(b) $\frac{A_{\text{storage}}}{A_{\text{system}}} = 0.04$ and $\frac{u_{\max}}{A_{\text{system}}} = 0.0005$.

	0%	100%
7	0.0059	0.0017
20	0.0143	0.0052
50	0.0295	0.0059

(c) $\frac{A_{\text{storage}}}{A_{\text{system}}} = 0.06$ and $\frac{u_{\max}}{A_{\text{system}}} = 0.0005$.

	0%	100%
7	-0.0460	-0.0215
20	-0.0533	-0.0315
50	-0.0420	-0.0315

(d) $\frac{A_{\text{storage}}}{A_{\text{system}}} = 0.02$ and $\frac{u_{\max}}{A_{\text{system}}} = 0.00025$.

Table E.1: Average water level for the four configurations of Experiment 2. The columns specify the value of the threshold as a percentage of the maximum pump capacity of the system. All values are in meters.

	0%	100%
7	15.94	15.46
20	17.16	16.17
50	19.80	16.19

(a) $\frac{A_{\text{storage}}}{A_{\text{system}}} = 0.02$ and $\frac{u_{\max}}{A_{\text{system}}} = 0.0005$.

	0%	100%
7	7.06	6.38
20	7.56	6.73
50	10.03	6.77

(b) $\frac{A_{\text{storage}}}{A_{\text{system}}} = 0.04$ and $\frac{u_{\max}}{A_{\text{system}}} = 0.0005$.

	0%	100%
7	5.31	4.47
20	5.47	4.82
50	7.40	4.93

(c) $\frac{A_{\text{storage}}}{A_{\text{system}}} = 0.06$ and $\frac{u_{\max}}{A_{\text{system}}} = 0.0005$.

	0%	100%
7	44.86	31.27
20	44.59	32.43
50	45.62	32.18

(d) $\frac{A_{\text{storage}}}{A_{\text{system}}} = 0.02$ and $\frac{u_{\max}}{A_{\text{system}}} = 0.00025$.

Table E.2: Total pump costs for the four configurations of Experiment 2. The columns specify the value of the threshold as a percentage of the maximum pump capacity of the system.

	Q costs [%]	Upw. costs [%]
7	2.62	72.51
20	2.87	73.63
50	2.93	77.88

(a) $\frac{A_{\text{storage}}}{A_{\text{system}}} = 0.02$ and $\frac{u_{\max}}{A_{\text{system}}} = 0.0005$.

	Q costs [%]	Upw. costs [%]
7	18.54	0.15
20	18.77	0.12
50	12.79	0.29

(b) $\frac{A_{\text{storage}}}{A_{\text{system}}} = 0.04$ and $\frac{u_{\max}}{A_{\text{system}}} = 0.0005$.

	Q costs [%]	Upw. costs [%]
7	22.06	0
20	14.37	0
50	7.65	0

(c) $\frac{A_{\text{storage}}}{A_{\text{system}}} = 0.06$ and $\frac{u_{\max}}{A_{\text{system}}} = 0.0005$.

	Q costs [%]	Upw. costs [%]
7	2.35	65.64
20	2.68	58.67
50	2.70	66.24

(d) $\frac{A_{\text{storage}}}{A_{\text{system}}} = 0.02$ and $\frac{u_{\max}}{A_{\text{system}}} = 0.00025$.

Table E.3: Pump costs and costs due to upward exceedance of the normal operating band as a percentage of the total costs for the four configurations of Experiment 2. Only the values for the cases with a threshold value of 0% are shown. Values for the cases with a threshold value of 100% are lead to the same conclusion.

List of Symbols and Abbreviations

A_{storage}	Area of belt canals	m^2
A_{system}	Non controllable system area	m^2
B	Value for the occurrence of precipitation	-
C	Transition matrix	-
E_{clim}	Mean of exponential distribution	mm/hour
E_{h}	Normal operating band exponent	-
E_{h_2}	Upward exceedance of h_{normal} exponent	-
E_{h_3}	Downward exceedance of h_{normal} exponent	-
E_{u}	Pump exponent	-
N_{ensemble}	Number of scenarios in original ensemble	Scenarios
N_{red}	Number of reduced scenarios	Scenarios
P	Amount of precipitation	mm/hour
R	Inflow	m^3/hour
T_{ic}	Information-control horizon	hour
T_{ip}	Information-prediction horizon	hour
T_{L}	Lead time	hour
T_{o}	Optimization horizon	hour
T_{s}	Simulation horizon	hour
V	Relative performance of TBMPC	m
h	Water level	m
h_{normal}	Normal operating band around set point	m
h_{ref}	Half of h_{normal}	m
h_{set}	Set point	m
k	Shape parameter	-
t	Index of time step in simulation horizon	hour
$u_{\max}$	Maximum pump capacity of belt canal pumps	m^3/hour
w_{h}	Normal operating band penalty	m/E_h
w_{h_2}	Upward exceedance of h_{normal} penalty	m/E_{h_2}
w_{h_3}	Downward exceedance of h_{normal} penalty	m/E_{h_3}
w_{u}	Pump penalty	time step^2/m^4
x	Random variable	-
α_1	Rainfall runoff model parameter 1	-
α_2	Rainfall runoff model parameter 2	-
α_τ	Uncertainty parameter	-
β	Rainfall runoff model parameter 3	-
Δt	Control time step	hour
θ	Scale parameter	-
λ	Rate parameter	-
μ	Mean of the distribution	-
σ^2	Variance	-
τ	Index of time step in optimization horizon	hour

List of Figures

List of Tables