OPTIMIZING CONTROL BY ROBUSTLY FEASIBLE MODEL PREDICTIVE CONTROL AND APPLICATION TO DRINKING WATER DISTRIBUTION SYSTEMS

Vu Nam Tran[1], Mietek A. Brdys[1],[2]

[1] *School of Electronic, Electrical and Computer Engineering*
College of Engineering and Physical Sciences.
University of Birmingham, Birmingham, B15 2TT, U.K.
e-mail: tvngachduoith@gmail.com

[2]*Department of Control Systems Engineering,*
Faculty of Electrical and Control Engineering,
Gdansk University of Technology, ul. Narutowicza 11/12, 80 – 233 Gdansk, Poland
e-mail:m.brdys@bham.ac.uk

Abstract

The paper considers optimizing Model Predictive Control (MPC) for nonlinear plants with output constraints under uncertainties. Although the MPC technology can handle the constraints in the model by solving constraint model based optimization task, satisfying the plant output constraints under the model uncertainty still remains a challenge. The paper proposes Robustly Feasible MPC (RFMPC), which achieves feasibility of the outputs in the controlled plant. The RFMPC which is applied to control quantity in Drinking Water Distribution Systems (DWDS) is illustrated by application to the DWDS example. In the simulation exercise, Genetic Algorithm is selected as the optimization solver and the reduced search space methodology is applied in the implementation under MATLAB-EPANET environment.

1 Introduction

Model Predictive Control has been an advanced technology and widely used in process control industry due to its ability to control multivariable systems with the presence of constraints. MPC actually belongs to a class of model based controller design concepts. The basic idea of the MPC algorithm remains unchanged regardless whatever kind of plant models are considered. It determines the optimal control actions by minimizing the user-defined objective function, or performance index. The current control actions are determined on-line, at each control step, by solving a finite-horizon open loop optimization problem, using the current state of the plant process as the initial state. However, only the first part of the optimized control input sequence is

applied to the plant in the next time step. At the next control step, the prediction horizon moves forward and the same procedure repeats [2, 10].

Due to its operation on a receding horizon, MPC is also referred as receding control horizon or moving horizon optimal control. There are two significant factors that determine how effective an MPC is. The first factor is the accuracy of the plant model since it is explicitly used to predict the plant outputs. The second factor is how effective optimization solvers are. Although with the best plant models, MPC technology is still challenged by the uncertainty existing in the system such as a model structure error, a state estimation error, and disturbances. Fulfilling constraints is essential in many process plants for reason of safety, productivity, and environment protection. The controller out-

puts, which are based on the plant model, may not meet the plant output constraint due to the model-reality mismatch. The mismatch is often caused by the difference between predicted disturbance and actual disturbance. Feasible control input may become infeasible when they are applied to the plant if there is no robustly feasible controller. The robustness meeting of the output constraints or state constraints under system uncertainties is the main objective of the robustly feasible MPC. In this paper, the optimizing RFMPC is considered. The robust feasibility will be assessed by the robust output prediction over the reduced horizon. Safety zones are employed to tighten the output constraints in order to achieve robustly feasible control input. The control method is applied to control quantity in DWDS.

2   Representation of RFMPC

The structure of the RFMPC [3] consists of several units as illustrated in Fig.1.The MPC optimizer solves the MPC optimization task to produce control inputs. In this task, the plant outputs are predicted basing on the nominal model of the plant. In the nominal model, the disturbance inputs are represented by their predictions, while the internal model uncertainties are represented by a selected scenario. Before the control input is applied to the plant, its robust feasibility is assessed by the "Constraint Violation Checking" unit. The feasibility assessment is based on the robust output prediction that is generated by the "Robust Output Prediction" unit. Given the control input, the corresponding robust output predictions over the prediction horizon are a region in the output space in which all the plant outputs generated by the control input and all possible scenarios of the disturbance inputs are contained. The input robust feasibility is checked by confronting the output constraints with the robust output prediction. If the control feasibility passed its assessment, then the proposed control input is applied to the plant. Otherwise, robust output prediction is fed into the "Safety Zone Generator" unit. The safety zones as such are used to tighten the output constraints. The control actions produced by the MPC optimizer under modified (tighten) output constraints are expected to produce the real plant outputs that satisfy the plant constraints although they still may violate the modified constraints. Such

control actions and the corresponding safety zones are called robustly feasible.

3   Robust Output Prediction (ROP)

Consider a continuous time plant, with the piecewise constant inputs, where input-output dynamics is modelled by the following discrete time systems:

$$
\begin{cases}
x(t+1) = f(x(t), u(t), z(t)) \\
y(t) = F(x(t), u(t))
\end{cases} \tag{1}
$$
$$\textit{with the initial conditions}: \ x(t_o) = x_o$$

where t is discrete time variable, $u(t)$ and $z(t)$ are the control and distuarbance input respectively. The disturbance input $z(t)$ is not known, and only bounds $z^{\min}$ and $z^{\max}$ on its instantaneous values are available, that is $z(t) \in [z^{\min}, z^{\max}]$

Given inputs $u(t+k|t)$ for $k \in [1, H_p]$, where H_p is the prediction horizon, the plant output over H_p can be predicted by using the plant model as:

$$
\begin{cases}
x(t+k+1|t) = \\
\quad f(x(t+k|t), u(t+k|t), z(t+k|t)) \\
y(t+k|t) = F(x(t+k|t), u(t+k|t))
\end{cases} \tag{2}
$$
$$\textit{with the initial conditions}: \ x(t_o) = x_o$$

where $z(t+k|t)$ denotes prediction of the disturbance input at $t+k$ produced at the time instant t.

The vector of control inputs and outputs over the prediction horizon are respectively defined as:

$$
\hat{U} = [u(t|t) \ ... \ u(t+H_m-1|t)
$$
$$
\underbrace{u(t+H_m|t) \quad \cdots \quad u(t+H_p-1|t)}_{from \ (t+H_m) \ to \ (t+H_p-1)} \Big]^T \tag{3}
$$

$$
\hat{Y} = [\ y(t+1|t) \quad \cdots \quad y(t+H_p|t) \ ]^T \tag{4}
$$

where $u(t+i|t), y(t+i|t)$ are the control inputs and model outputs at time $t+k$ produced at time instant t, and H_m, H_p are the *input horizon* and *prediction horizon* respectively.

Given the $x(t|t)$ and control inputs over the horizon $u(t+k|t), \ k = 0, ..., H_p - 1$. Let $y(t+k|t)$, $k = 1, ..., H_p$ be the corresponding plant outputs.

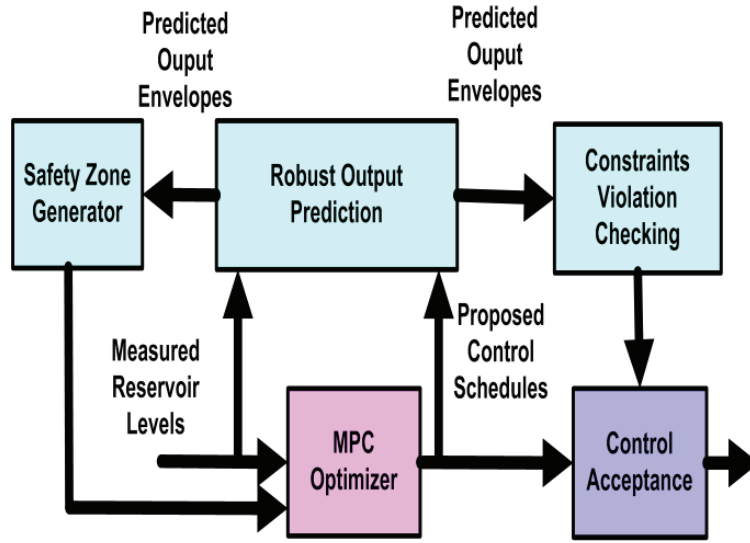**Figure 1**. Structure of Robustly Feasible MPC

The robust prediction of $y(t+k|t)$ is composed of two envelopes:

$$Y_p^l = [y_p^l(t+1|t)...y_p^l(t+H_p|t)]^T \qquad (5)$$

$$Y_p^u = [y_p^u(t+1|t)...y_p^u(t+H_p|t)]^T \qquad (6)$$

where $y_p^l(t+k|t)$ and $y_p^u(t+k|t)$ are the upper and lower limits that robustly bound the plant output at prediction time step k:

$$y_p^l(t+k|t) \leq y(t)|_{t=t+k} \leq y_p^u(t+k|t) \atop k = 1,...,H_p \qquad (7)$$

The least conservative bounding envelopes $y_p^l(t+k|t)$ and $y_p^u(t+k|t)$ can be determined as:

$$
\begin{aligned}
y_p^l(t+k|t) &= \min_{z(t|t),z(t+1|t),..,z(t+k-1|t)} y(t+k|t) \\
&= \min_{z(t|t),z(t+1|t),..,z(t+k-1|t)} F(x(t+k|t),u(t+k|t))
\end{aligned}
$$
$$(8)$$

$$
\begin{aligned}
y_p^u(t+k|t) &= \max_{z(t|t),z(t+1|t),..,z(t+k-1|t)} y(t+k|t) \\
&= \max_{z(t|t),z(t+1|t),..,z(t+k-1|t)} F(x(t+k|t),u(t+k|t))
\end{aligned}
$$
$$(9)$$

where uncertainty at time $t+k$: $z(t+k|t) \in [z^{min}, z^{max}]$, $\forall k \in \overline{1:H_p}$; the states $x(t+k|t)$ are obtained from the state space equations (2) with known initial condition $x(t|t)$.

Generating $y_p^l(t+k|t)$ and $y_p^u(t+k|t)$ also produce plant state bounding envelopes $x_p^l(t+k|t)$ and $x_p^u(t+k|t)$, for $k = 1,...,H_p$.

Since the robust output prediction is calculated over the horizon H_p, there are H_p optimization problems to be solved to find H_p values of $y_p^l(t+k|t)$ and $y_p^u(t+k|t)$. As k increases from 1 to H_p, the optimization also increases the number of variables from 1 to H_p.

Indeed, when k $=H_p$, (8) and (9) have H_p variables $z(t), z(2), ..., z(t+H_p-1)$. The more variables the optimization has, the more computing time the solvers require. As these computations are carried out online, it is desired to reduce the time computing as much as possible.

3.1 Stepwise Robust Output Prediction (SWROP)

In the previous section, solving optimization problems (8) and (9) give at least conservative solution of robust output prediction (ROP). This approach is so called the exact optimization method. In contrast to the exact optimization method, we propose in this section an approximated optimization method where its advantage is to reduce the optimization process computing time.

Instead of solving the optimization task with respect to k variables $z(t|t), z(t+1|t), ..., z(t+$

$k-1|t)$, one could approximate the least conservative robust output prediction (LCROP) by solving the optimization tasks ((8) and (9) with respect to only one variable $z(t+k-1|t)$ while $z(t|t)$, $z(t+1|t)$, ..., $z(t+k-2|t)$ are obtained from the optimization in the previous time steps. In other words,

instead of simultaneous optimization with respect to all disturbance inputs, a step by step optimization is applied with respect to one disturbance input at the time starting with $x_p^l(t+k|t)$ and $x_p^u(t+k|t)$.

Hence,

$$y_p^l(t+k|t) = \min_{z(t+k-1|t)} y(t+k|t)\Big|_{z(t|t)=z^{\min}(t|t),....,z(t+k-2|t)=z^{\min}(t+k-2|t)} \tag{10}$$

$$y_p^u(t+k|t) = \max_{z(t+k-1|t)} y(t+k|t)\Big|_{z(t|t)=z^{\min}(t|t),....,z(t+k-2|t)=z^{\min}(t+k-2|t)} \tag{11}$$

where $z^{\min}(t+i|t)$ and $z^{\max}(t+i|t)$ can be obtained by solving:

$$z^{\min}(t+i-1|t) = \arg\min_{z(t+i-1|t)} y(t+i|t)\Big|_{z(t|t)=z^{\min}(t|t),....,z(t+k-2|t)=z^{\min}(t+i-2|t)} \qquad \forall i \in \overline{1:k} \tag{12}$$

and

$$z^{\max}(t+i-1|t) = \arg\max_{z(t+i-1|t)} y(t+i|t)\Big|_{z(t|t)=z^{\min}(t|t),....,z(t+k-2|t)=z^{\min}(t+i-2|t)} \qquad \forall i \in \overline{1:k} \tag{13}$$
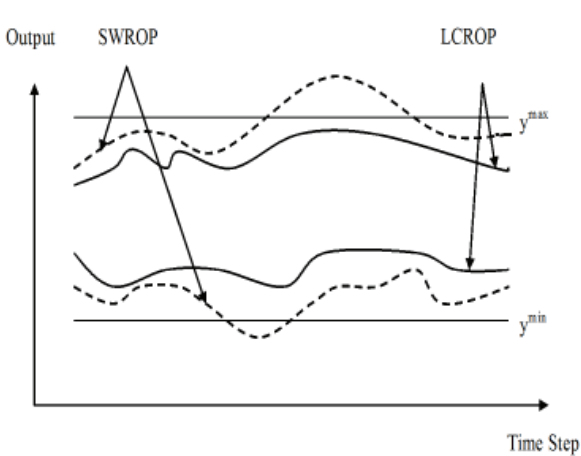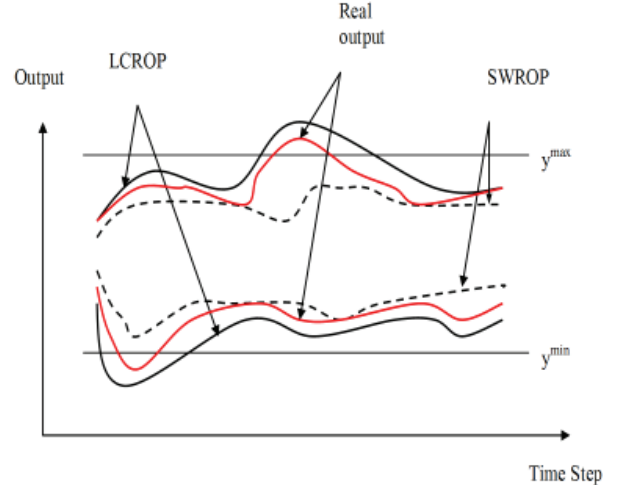

Figure 2. SWROP stays outside LCROP

Figure 3. SWROP lies entirely inside LCROP

The resulting bounding envelopes are more conservative but the computing time is vastly reduced. Unfortunately, the expressions (10) and (11) generate the ROP only for some class of systems. The paper objective is to apply RFMPC to DWDS and there such a class has a clear interpretation, hence can clearly be identified. In order to assess the robust feasibility by SWROP, one should ensure that the LCROP entirely remains inside the SWROP as described in Fig.2. Otherwise the real output may

possibly violate the upper or lower constraint even though the SWROP does not, as described in Fig 3.

In practice, there are some classes of the system that have the characteristic as depicted in Fig.2 while some will have the characteristic of Fig.3. Hence, in order to avoid the situation of having robustly infeasible control input, designers in practice should take that into consideration of choosing the appropriate method to calculate the robust output prediction.

3.2 Reduced Robust Feasibility Horizon

So far the ROP has been considered over the whole output prediction horizon H_p set up for the RFMPC. This has been done in order to secure the existence of the robustly feasible safety zones at any control time step. However, as computing of ROP over H_p is computationally very demanding and this may not meet the time constraints set up by on-line computing requirements. We should consider reducing this demand by shortening the ROP horizon. Clearly the cost to be paid is an increased risk of non existence of robustly feasible safety zones at certain control time steps. As only the first control action out of a whole sequence determined by the RFMPC is applied to the plant, we must secure the robust feasibility over the first time step. This is how far we can go with the reduction of the ROP horizon from H_p to H_r. An attractive outcome of the ROP horizon reduction is that the very attractive computing SWROP method may become applicable over the reduced horizon while may not be applicable over the entire horizon. (see Fig.4)
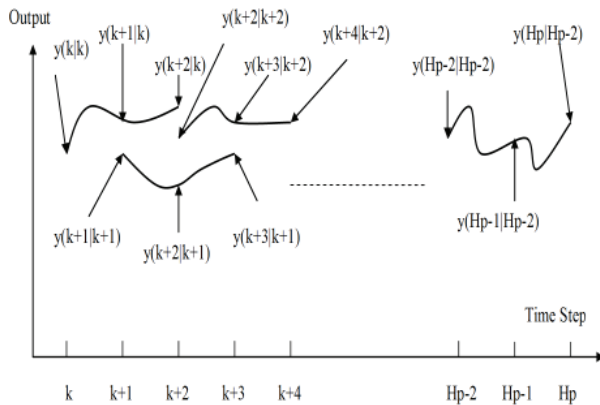

Figure 4. Example of reduced robust feasibility horizon to two time steps - $H_r = 2$

4 Safety Zone Generator

Using safety zones is not a new idea to meet the system constraint under unknown factors. It is widely used in engineering area, such as a conservative design in many electrical devices. When the input from the nominal model base MPC controller is applied to the plant, due to the uncertainties of the system, the output constraints may not be fulfilled and their violations may be unacceptable at certain time instants. If the violation occurs, it is important to correct or modify the constraints that apply to the nominal MPC. The safety zones generator is the unit that modifies the output constraints via the iterative scheme.
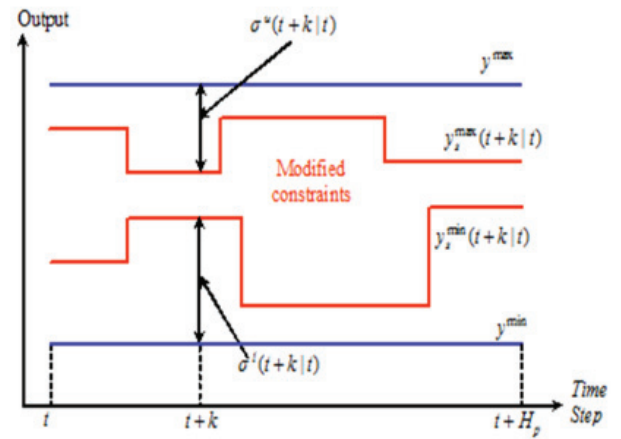

Figure 5. The output constraints modified by safety zones

Consider over the prediction horizon, the vectors of the lower and upper limits on the plant output $Y^{min} = [\ y^{min}\ \cdots\ y^{min}\ ]^T$; $Y^{max} = [\ y^{max}\ \cdots\ y^{max}\ ]^T$ and the vectors of the safety zones $\sigma^l = [\ \sigma^l_1\ \cdots\ \sigma^l_{H_p}\ ]^T$, $\sigma^u = [\ \sigma^u_1\ \cdots\ \sigma^u_{H_p}\ ]^T$ for the lower and upper output constraints, respectively where σ^l_i and σ^u_i are non negative real numbers. The vectors $Y^{min}_s = Y^{min} + \sigma^l$ and $Y^{max}_s = Y^{max} - \sigma^u$ are composed of the lower and upper bounds of the modified output constraints over H_p, respectively.

The "Safety Zones Generator" produces iteratively robustly feasible safety zones by using the following relaxation algorithm [3]:

(i) Set $x = [\ \sigma^l\ \ \sigma^u\ ] = 0$;

(ii) Solve MPC optimization task with modified output constraints H_p

(iii) A vector V composed of the output constraint violation over the prediction horizon is calculated as:

$$V = [V_1 \cdots V_{2H_p}]^T$$
$$\triangleq \left[\begin{array}{cc} (Y^{\min} - Y_p^l)^T & (Y_p^u - Y^{\max})^T \end{array} \right]^T$$

Define $f(V_i) \triangleq \max\{0, V_i\}$ and $C(\sigma^l, \sigma^u) \triangleq [f(V_1) \cdots f(V_{2H_p})]^T$
If

$$C(\sigma^l, \sigma^u) = 0 \qquad (14)$$

is satisfied then go to step (vi), Else go to step (iv);

(iv) Calculate the safety zone corrections by using $\delta^{(k)} = -\nu C(x^{(k)})$ where $\nu = \max([diag[\nabla C(0)]]^{-1})$ is called the relaxation gain

(v) $x^{(k+1)} = x^{(k)} + \delta^{(k)}$, go to step (ii)

(vi) The robustly feasible safety zones have now been found and the control inputs $u(t|t)$ are applied to the plant.

5 Optimizing Control of DWDS by RFMPC

In the daily operation of water distribution systems, a period of water demand prediction ahead of current time is usually needed to be the basis for generating optimal pump actions so as to achieve certain control objective, e.g. the least pumping cost. Since too long or infinitive time horizon demand prediction is not accurate or unavailable, a relatively short prediction horizon is more realistic, and this is applied in a receding horizon manner, which forms the key idea of MPC technique. The corresponding optimization that reflects the control strategy is solved under the RFMPC structure and only the first part of the control input is applied. In this section, the formulation of the optimizing control problem and the simulation environment implementation by MATLAB – EPANET are presented. The Genetic Algorithm is selected as the optimization solver and the *reduced search space* methodology that has been used in the implementation will also be explained.

5.1 Formulation of the Optimizing Control Problem

The main goal of DWDS is to supply water to customers and satisfy their quantity and quality demand. There are two major aspects in the control of DWDS: quantity and quality. The quality control deals with water quality parameters. Having disallowed the concentration of the chemical parameter, for instance chlorine, cause serious heath dangers. Maintaining concentrations of the water quality parameters within the prescribed limits throughout the network is a major objective. When the quantity control is considered, the objective is to minimize the electrical energy cost of pumping, while satisfying consumer water demand and physical constraints such as pressure at nodes or reservoir levels, by producing optimized control input such as optimized pump speeds and valve control schedules [5]. The uncertainty is in the demand and structure and parameters of DWDS model. In this paper, only the quantity control aspect is considered by applying RFMPC technique. The quality issues are addressed in [4, 15] for example.

Objective function- pumping cost control: It is a very common control objective to achieve the least pumping cost while satisfying constraints. Moreover, in order to achieve a sustainable operation day after day, it is expected that tank levels can come back to their original states after a certain period. For the DWDS example, the network is operated daily and the prediction horizon is $H_p = 24h$. It is desired that after 24 hours, the tank level could have a similar level.

Hence, the overall objective function at $t = \bar{t}$ reads:

$$J = \sum_{t=\bar{t}}^{\bar{t}+H_p-1} \gamma(t)\Delta t \sum_{j=1}^{G_p} \sum_{i=1}^{U_j} \frac{\xi q_{j,i}(t)\Delta h_j(t)}{\eta_{j,i}(t)}$$
$$+\rho \sum_{s=1}^{S} |r_s(\bar{t} + H_p) - r_s(\bar{t})|$$

where H_p is the prediction time horizon, ρ is a weighting factors, $\gamma(t)$ is a power unit charge in /kWh for the $(t+1)$ time stage, r_s is the sth reservoir/tank level, $s = 1, ..., S$, ξ is a unit conversion factor for electrical power relating water quantities to electrical energies, and η_i is the pump efficiency of the ith pump in the jth pump group, $i = 1, ..., U_j$ and $j = 1, ..., G_p$.

The decision variables are the pump speed over

the prediction horizon. They are automatically computed by the optimization solver to minimize the objective junction while satisfying the optimization constraints. The optimization constraints compose of equality constraints and inequality constraints. However, from the implementation point of view, the optimization constraints could be classified into implicit and explicit ones.

Explicit constraints: are often the control input constraints. It can be the sequence of the pump speed schedule or the ON/OFF state of pumps and need to satisfy the physical constraints. They are explicitly embedded into the optimization solver i.e. $u^{\min} \leq u(t+k|t) \leq u^{\max}$.

Implicit constraints: are the ones that have to be implicitly embedded into the optimization solvers. Implicit constraints are nonlinear and consist of all equality constraints defining the hydraulic equilibrium state of the system. They correspond to the set of flow continuity equations, volume/mass balance equations, and energy conservation equations. The output constraint such as tank heads and flows are also considered as implicit constraints. Implicit constraints are composed of:

– Nodal flow continuity equations: $\sum_{j \in J_i^-} q_j - \sum_{j \in J_i^+} q_j - d_i = \begin{cases} 0 & \text{for } i \in M \\ lq_i & \text{for } i \in M_l \end{cases}$

– Water elements head-flow equations: $h_{N_j^+} - h_{N_j^-} = \Delta h_j(q_j, u_j)$ where q_j is the flow at arc j (liter/sec); h_i is the head at node i (m); d_i is the demand flow at node i (liter/sec); lq_i is the leakage flow at node i (liter/sec); u_j is the control variable representing the state of valve or pump at arc j; Δh is the head-flow characteristic function at arc j; $M(M_l)$ is the set of non-leaky (leaky) nodes; $J_i^+(J_i^-)$ is the set of arcs whose start (end) node are i; and $N_j^+(N_j^-)$ is the start (end) node j

– Volume mass balance equations of tanks/reservoirs

– Output constraints. They are in the form of lower and upper bounds on certain flows, junction heads, and on all tank heads in order to meet the tank capacity constraints.
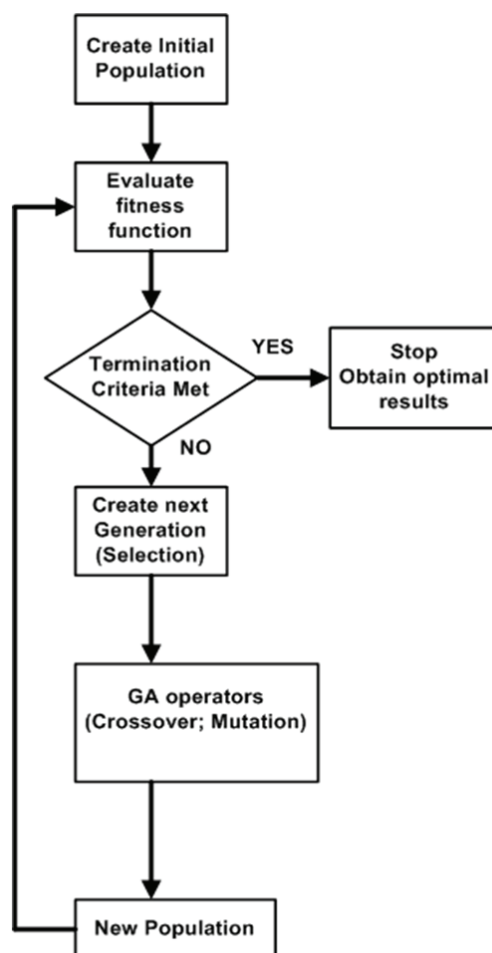

Figure 6. Flowchart of the standard GA

5.2 Genetic Algorithm – the Optimization Solver

GA represents a discrete variable optimization technique based on the principles of genetics and natural selection. The method was originally proposed by Holland in 1975 [9], and further developed by Goldberg in 1989 [7]. GA has been extensively used in many industrial engineering applications and so far represents one of the most commonly employed natural optimization techniques for design of water distribution networks as evidenced by use of GA for sizing of pipes [14, 17], evaluation of system reliability [16], and placement of early warning detection sensors [12]. GA has successfully solved the discontinuous, the non-differentiable, the non-convex, the multiple peaks, or highly nonlinear function optimization problems which traditional analytical and numerical methods are not well suited [8]. Advantages of this method include:

– The GA searches from a population of decision variable sets, not a single decision variable set.

– The GA can optimize discrete or continuous variables.

– The GA uses the objective function itself, not the derivative information. [8]

– The GA algorithm uses probabilistic (not deterministic) search rules.

– GA does not require discretization of state variables.

– GA does not require transition probabilities.

– GA models results in optimal or near optimal solutions.

However, GA has also some disadvantages:

– Cannot efficiently handle large number of constraints.

– Computationally difficult to provide very long string length with binary coding.

– Every iteration needs objective function evaluation.

– Global solutions are not guaranteed.

Briefly, the GA algorithm is initiated with a random population of individuals in which each individual is represented by a binary string (i.e., chromosome) for one possible solution. The standard structure of the GA is illustrated in Fig.6. The major components of the algorithm are explained as below:

1. At the beginning of the computation process, users specify the initial population in advance or GA randomly creates an initial population. For each population generation, a measure of the fitness with respect to an objective function is calculated.

2. If one of the pre-established criteria of the algorithm such as a number of generations, time limit, fitness limit, stall generations, stall time limit, and fitness tolerance are met, then the optimization terminate and the optimal results are obtained. Otherwise, go to step 3.

3. Based on this fitness value, the individuals are selected to create the next generation (Selection). They are called parents.

4. GA use operators to produce children (*offspring*) for the next generation of population. There are different operators e.g. crossover, mutation. Children are produced either by making random changes to a single parent (*mutation)* or by combining the vector entries of a pair of parents (*crossover)*. In the mutation process, all children are mutated with a certain probability, which ensures the probability of searching a particular subspace of the problem, the space is never zero. It is intended to prevent premature convergence and loss of genetic diversity. Whereas in the *crossover* process, the individuals are recombined to produce offspring. The purpose of crossover is to let individuals exchange useful information with each other, and get higher fitness in order to have better individual in the next generation and have preferable performance.

5. Use new generated population for a further run of the algorithm. Go to step 2.

In order to obtain the best result from GA, users usually need to experiment with different options. Selecting the best options for a problem involves trial and error. There are quite a number of different ways/options to improve results. Few of the most important factors that need to be tuned in order to get the good result are described as below:

– Population Size: represents the diversity of the population. It is one of the most important factors that determines the performance of the genetic algorithm performs is the *diversity* of the population. If the average distance between individuals is large, the diversity is high; if the average distance is small, the diversity is low. Getting the right amount of diversity is a matter of trial and error. If the diversity is too high or too low, the genetic algorithm might not perform well. However, the *Population size* should be at least the value of *Number of variables*, so that the individuals in each population span the space being searched.

– Initial range: specifies the range of the vectors in the initial population that is generated by a creation function. If users know or have knowledge approximately where the solution to a problem lies, the initial range should be specified so that the guess for the solution is contained and it can enable GA search more effectively. However, the genetic algorithm can find the solution even if it does not lie in the initial range, provided that the populations have enough diversity.

– Crossover fraction and Mutation function: Crossover fraction effect the ability of the algorithm to extract the best genes from different individuals and recombine them into potentially superior children. Different crossover fractions lead to different performances of GA. Similarly, different mutation function adds to the diversity of a population and thereby increases the likelihood that the algorithm will generate individuals with better fitness values. For certain problems, adjusting the amount of mutation fraction can possibly make significant improvement of the solutions.

– Initial population: has significant impact on GA's performance in the complex nonlinear constrained problem. The optimizing control problem of DWDS consists of many variables and nonlinear constraints that GA cannot directly handle. For GA in order for GA to perform effectively, initial population needs to be provided. Different initial population might lead to different solutions. Without knowledgeable initial population, GA are still able to randomly create initial population itself, but it could lead to the situations of no feasible solution being found, or take tremendous computing time to converge.

Regarding to the application of complex DWDS, the standard GA needs to be enhanced in order to exploit specific features of the optimization task and achieve required computing efficiency [11].

5.3 Simulation Environment Implementation

The computer implementation is based on MATLAB-EPANET environment. The optimization problems are solved by standard GA [6] which

can be called through MATLAB Genetic Algorithm Toolbox [1]. The simulation implementation is illustrated in Fig.7. In order to solve the DWDS problem efficiently, the *reduced search space* methodology was employed by means of which the water distribution network simulator (EPANET) [13] is embedded directly to the optimization solver (GA). Specifically, in the Predictive Controller block, starting with the feasible set of control inputs, the optimization solver i.e. GA passes the control input into the EPANET for use in explicitly satisfying the implicit system constraints (equality constraints) and in evaluating the implicit bound constraints (inequality constraints). The water network information (i.e. nodal pressure, tank level, heads, flows) that are obtained by the EPANET are passed back to GA for determining the objective function. This process iteratively runs until the optimal control inputs at the present time step are found. The EPANET here plays as the *model* of the DWDS. The flow path of the optimization procedure within the Model Predictive Controller block is shown in Fig.8.
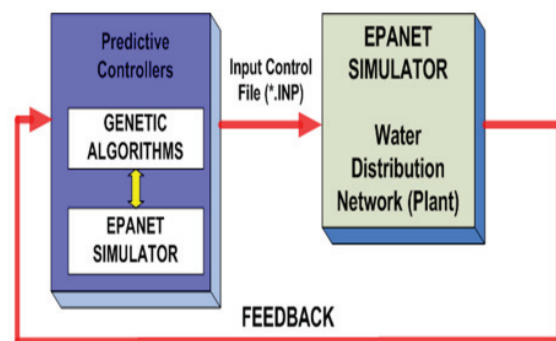

Figure 7. Simulation environment implementation

At each time step only the first part of the control inputs is applied to the Water Distribution Network (plant) block. The EPANET is used again as the *plant* of the DWDS. Note that there is always a model-plant mismatch i.e. disturbances (predicted demand and actual demand). The plant generates the massive output data in which only the states (tank levels) are fed back to the Model Predictive Controllers block for the next time step use.

The EPANET that is embedded into GA takes care of all hydraulic equations of DWDS. In other words, it reduces the number of equality constraints of the optimization problem; hence GA will only have to find the solution in the *reduced search*

space. Consequently, the overall computing time is reduced. It is an important feature of *reduced search space* methodology since the online operation of the RFMPC requires a tremendous computing effort.
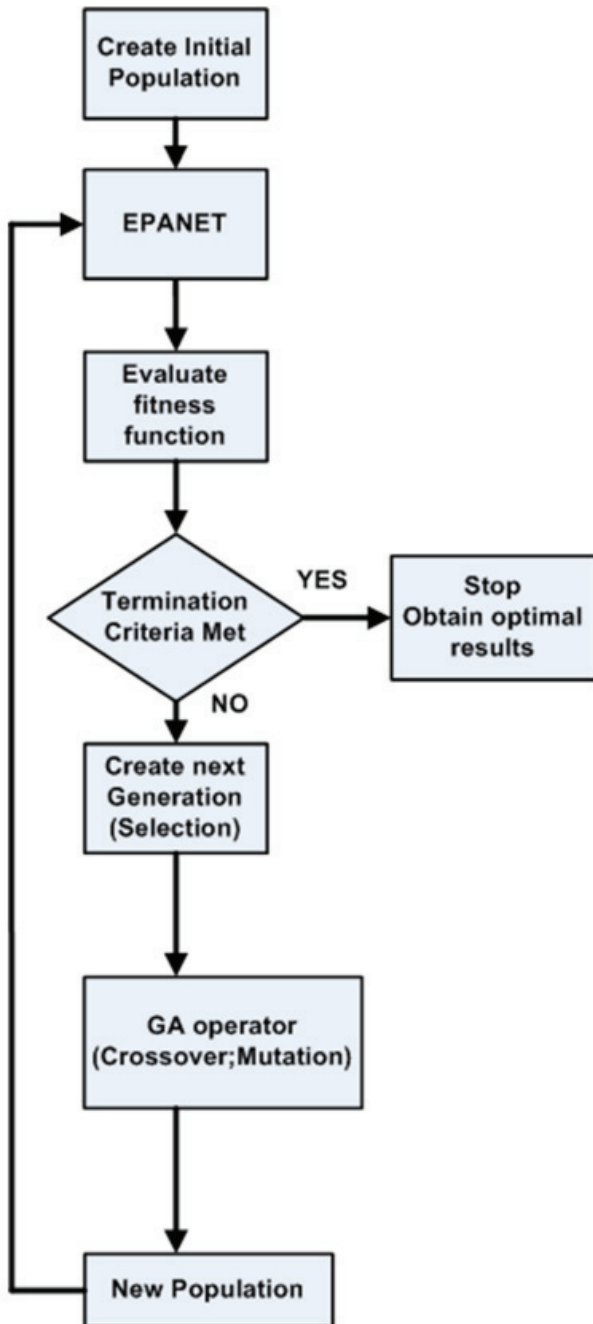

Figure 8. Optimization procedure of MPC

The main specifications of the software and hardware in the simulation:

– EPANET : Version 2.0

– MATLAB: Version 7.6.0.324 (R2008a)

– GA-Toolbox: Version 2.3 (2008a)

– CPU/Memory: 2.00 GHz CPU, 2.00GB RAM.

– Operating System: Windows XP Professional

5.4 Application of RFMPC to Example Case Study DWDS

The DWDS, which is depicted in Figure 9 includes 1 source reservoir and 1 storage tank. Water is pumped from the reservoir source by the pump station and can also be supplied by the storage tank (node 7).

The assumed positive flow is expressed in the way the link ID shows, e.g. link 203 denotes the positive flow direction from Node 2 to Node 3. The negative value of the flow indicates the flow direction is opposite to the assumed positive flow direction, e.g. $q_{402} < 0$ means the flow direction is from Node 3 to Node 2.

The prediction horizon is $H_p = 24$. The interested control input is pump speed sequence over 24 hours period. RFMPC is applied to produce control the input sequence. The tank level limits are: $r_s^{\min} \leq r_s \leq r_s^{\max}$, and these are the output constraints. The demands are predicted with the error of 10% at each consumption node.
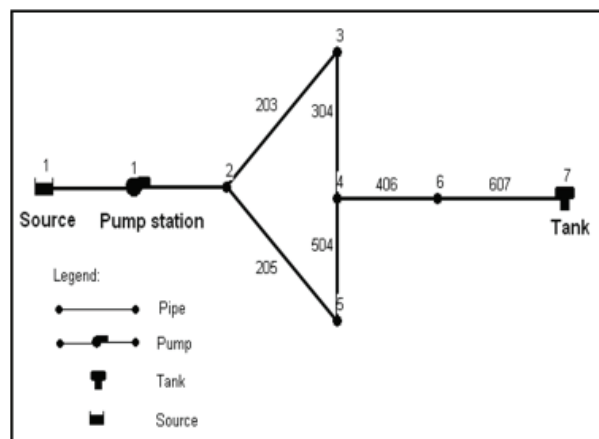

Figure 9. Diagram of an DWDS example

For the operational control purposes, the configuration of the DWDS example is given in the table 1, 2, 3, and 4 which includes the nodal elevation, nodal base demand, the operating constraints for nodal pressure, daily nodal demand profile, tank initial level and level range pipe, pipe and pump installation data, and time dependant electricity tariff.

Table 1. Nodal data for the pipe network

Node ID	Elevation(m)	Minimum head(m)	Maximum head(m)	Base demand(l/s)
2	15.0	18.0	32.0	5.0
3	14.0	16.0	30.0	5.0
4	12.0	12.0	28.0	5.0
5	14.0	14.0	30.0	5.0
6	8.0	10.0	28.0	30.0
1	5.0			
7	—— Tank/Reservoir nodes ——			

Table 2. Tank (Reservoir) data of the example DWDS

Node ID	Elevation(m)	Initial level above bottom(m)	Min level above bottom(m)	Max level above bottom (m)	Tank diameter(m)
7	10.0	5.0	3.0	10	15.0
1	15.0	——————			

Table 3. Pipe data of the example DWDS

Pipe ID	Start node	End node	Length(m)	Diameter(mm)	C Value
203	2	3	1000	400.0	100
205	2	5	1000	400.0	100
304	3	4	1000	400.0	100
504	5	4	1000	300.0	100
406	4	6	1000	500.0	100
607	5	7	1000	500.0	100

Table 4. Pump data of the example DWDS

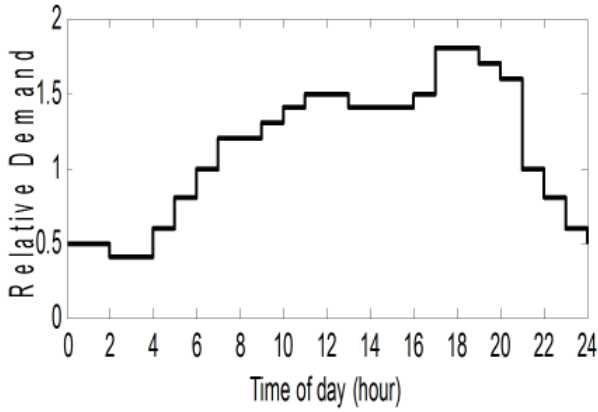| Pump ID	Head node	Tail node	Coefficient $\bar{A}$ (10^{-2})	Coefficient $\bar{B}$	Cut-off head $\bar{C}$ (m)
1	1	2	-0.5419	0	200

Figure 10. Daily demand profile

Optimization problems are formulated with a moving horizon $H_p = 24$hours ahead of present time and the sampling period is fixed to 1 hour. Electricity tariff difference is taken into account: low tariff is charged at 4.51p/kWh for the night time between 10:00pm–6:00am and high tariff is 9.72p/kWh for the daytime between 10:00pm–6:00am

5.4.1 Designing RFMPC

The MPC task is solved by GA solver with the optimization search in the reduced space. In this search, the GA is coupled to the EPANET simulator solving the DWDS equality constraints.
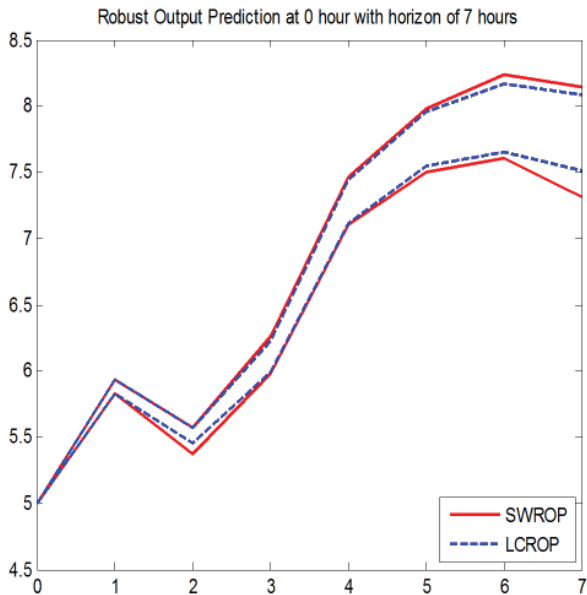
Figure 11. Robust output prediction at t = 0, Hr = 7

A method for generating ROP is chosen by observing the simulation results shown in Fig.11 The

SWROP and the LCROP are applied at t = 0 over 7 time steps. It can be seen in Fig.11 that the SWROP method generates envelopes that are outside the region determined by the LCROP method. Hence, the SWROP is applicable to our example DWDS. Moreover, the envelopes calculated by the two methods are very close over the first 6 steps. The ROP horizon therefore is further reduced to 2 steps and the SWROP method is to be applied.

Also the relaxation gained in the algorithm for determining the robustly safety zones (RFSZ) is selected by simulation where several gain values are tried and the results are illustrated in Fig.12 The equality (14) in the step (iii) of the RFSZ relaxation algorithm has more than one solution. Clearly, the smaller safety zones are, the less conservative control actions are, and consequently a better controller performance is achieved. In Fig.12, this is obtained for small gain values. On the other hand, the computing time is essential; hence the number of iterations needed to reach the RFSZ should be minimized. This is obtained for high gain values as described in Fig.8. Therefore, gain $v = 0.6$ is chosen in order to trade between the two aspects.
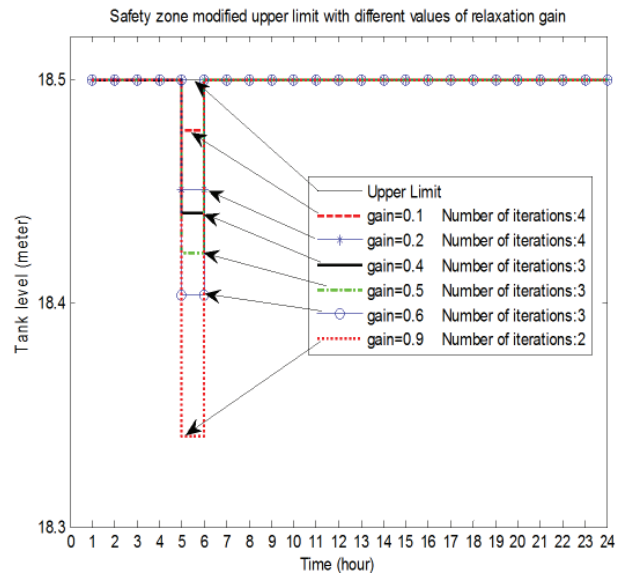

Figure 12. Robustly feasible safety zones and the corresponding modified tank upper limit for different relaxation gain values.

In the considered objective function of the corresponding MPC that has been applied to the DWDS, a long term sustainable operation of the distribution system is guaranteed by a periodical control of the tank level back to its original status. This

issue is embedded into the optimization problem as a suitable penalty term.

There are some parameters such as termination criteria, initial range, and population size in the GA options that have been chosen by experiments in order to obtain good simulation results within a reasonable period of computing time. The initial population is also provided to GA rather than let it create itself. The property settings of GA are listed as below:

```
StallTimeLimit=500;

PopulationSize=48;

StallGenLimit=10;

PopInitRange=[0.2;1.6];
```

5.4.2 Simulation Results

Firstly, the RFMPC is applied to the example DWDS at t = 0. Robust feasibility at the obtained control sequence is checked over the horizon $H_r = 2$ and the first two control inputs are assessed as robustly feasible. Hence, there is no need to activate the "Safety Zone Generator". In Fig.13 two tank trajectories are illustrated: one in a dash line is obtained by applying the control sequence to the model with the demand prediction while the second one in solid is the tank trajectory seen in the real system where the demand may differ from the predicted one up to 10%.

It can be seen in Fig.13 that the upper limit tank constraint is violated during 5 hour to 7 hour time period. Clearly, we are not aware of this violation at t = 0. However, a lesson to be learnt is that applying a whole control sequence obtained at t = 0 to the network is not recommended not only in this case but in general. Therefore, the RFMPC is kept applying to produce the control actions on-line by employing feedback and all its mechanism described in this paper.
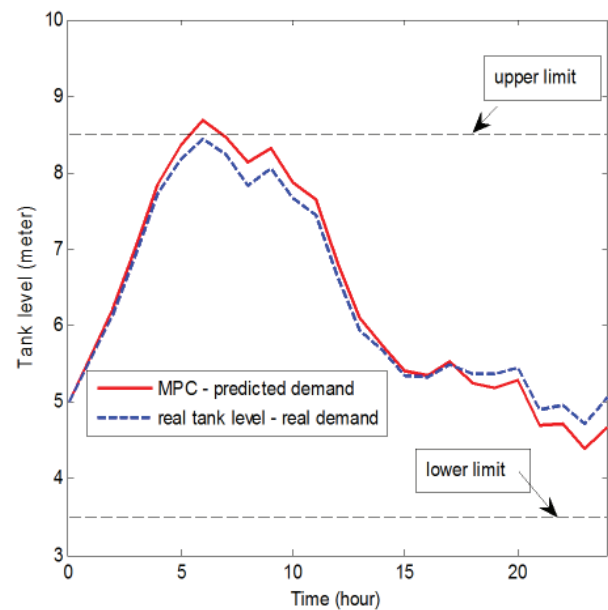

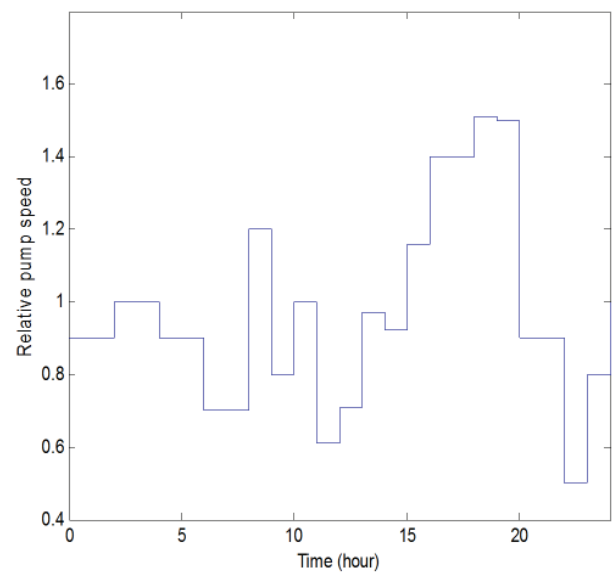**Figure 13**. Predicted tank level trajectory by RFMPC over the horizon at time instant t = 0 and Hr = 2

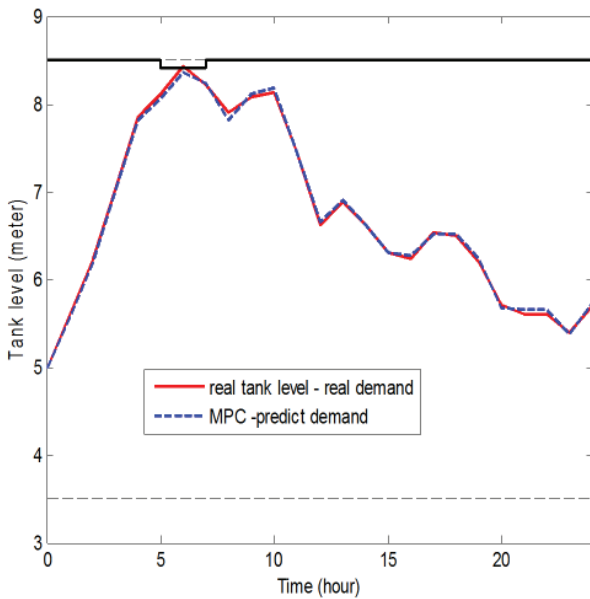**Figure 14**. Control actions - relative pump speed

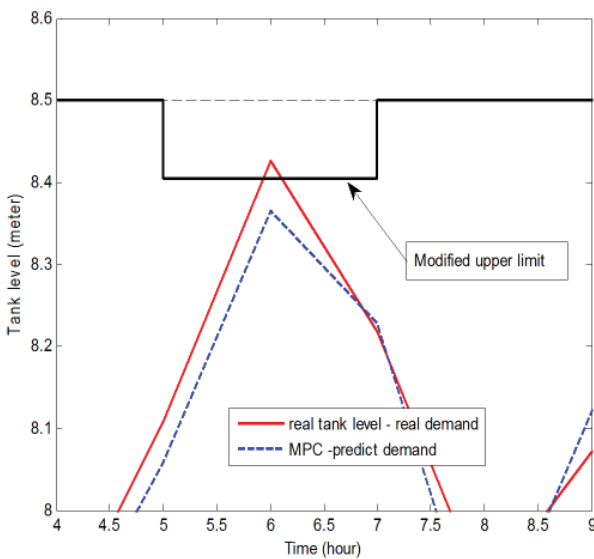**Figure 15**. Tank trajectory over the 24 hours

Figure 16. Zoom-in of Fig.10 during 4–9 hours

The results are illustrated in Fig.14–16. It can be seen in Fig.15 that the upper tank level constraint had to be modified by robustly feedback safety zones over 5, 6, and 7 time steps in order to achieve robust feedback of the control action over these time steps. Although the modification does not tighten the constraints excessively, its conservatism would be improved by extending the robust prediction horizon. The details of the situation over 5, 6, and 7 time steps are illustrated in Fig.16.

In order to assess the RFMPC feedback strength, the control actions generated on-line are

also applied to the DWDS model. The resulting tank trajectory and the control input are shown in Fig.15 and Fig.14, respectively. The two trajectories are much closer in Fig.15 than in Fig.13. Hence, the possible impact of the feedback in compensating the demand error impact is noticeable.

Lastly, as shown in Fig.16, the modified constraints are satisfied in the model but not in reality. However, the actual constraint is met in reality, showing the effectiveness of the RFSZ mechanism.

6   Conclusions

This paper has further developed Robustly Feasible Model Predictive Control Method for the on-line optimizing control of nonlinear plants with output constraints under uncertainty. The RFMPC has been applied to quantity control in Drinking Water Distribution Systems. It has been illustrated by simulation based on an example DWDS. The effects of the robust output prediction, shortening the robust output prediction horizon, robustly feasible safety zones, and the feedback strength of RFMPC have been shown.

The combination of GA and EPANET via the reduced space methodology has been applied in the implementation of the DWDS. Nevertheless, meeting the requirement of computing time during on-line operation is still a big challenge for RFMPC designers. In addition to the reduced space methodology, the gradient-type optimization solvers can be utilized due to its advantage of taking the derivative information. This interesting and important topic is currently under future research.

References

[1] *MATLAB Genetic Algorithm Toolbox Manual*.

[2] A. Bemporad and M. Morari. Robust model predictive control: A survey,. In *Lecture Notes in Control and Information Sciences*, volume 245, pages 207–226. Springer-Verlag, 1999.

[3] M.A. Brdys and T. Chang. Robust model predictive control under output constraints. In *Proc. of the 15th IFAC World Congress*, Barcelona, 2002.

[4] M.A. Brdys, K. Duxinkiewicz, T. Chang, M.Polycarpou, A. Wang, J. Uber, and M. Propato. Hierachical control of integrated quality and quantity in drinking water distribustion systems.

In *Proc. of the IFAC International Conference on Technology, Automation and Control of Wastewater Systems- TiASWiK'02*, Gdansk - Sobieszewo, Poland, June 19–21, 2002.

[5] M.A. Brdys and B. Ulanicki. *Operational Control of Water Systems: Structures, Algoritms and Applications*. 1994.

[6] K. Deb. An efficient constraint handling method for genetic algorithms. *Computer Methods in Applied Mechanics and Engineering*, 186:311–338, 2000.

[7] D.E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison Wesley, 1989.

[8] M. Haestad, T.M. Walski, D.V. Chase, D.A. Savic, W. Grayman, S. Beckwith, and E. Koelle. *Advanced Water Distribution Modeling and Management*. Haestead Press, Waterbury, CT, 2003.

[9] J.H. Holland. *Adaptation in Natural and Artificial Systems*. MI: University of Michigan Press, Ann Arbor, 1975.

[10] J.M.Maciejowski. *Predictive Control with Constraints*. 2000.

[11] W. Kurek and M.A. Brdys. Genetic solver of optimisation task of mpc for optimising control of integrated quantity and quality in drinking water distribution systems. volume 11. IFAC– PapersOnLine, Elsevier, 2008.

[12] A. Ostfeld and E. Salomons. Optimal layout of early warning detection stations for water distribution systems security. *Journal of Water Resources Planning and Management-Asce*, 130:377–385, 2004.

[13] L.A. Rossman. *EANET 2.0 Users manual*. US Environmental Protection Agency, Cincinnati, 2000.

[14] D.A. Savic and G.A. Walters. Genetic algorithms for least-cost design of water distribution networks. *Journal of Water Resources Planning and Management-Asce*, 123:66–67, 1997.

[15] T.Chang and M. A. Brdys. Performance comparison of chlorine controllers based on input-output and state-space models. In *Proc. of the 1st Annual Environmental & Water Resources Systems Analysis (EWRSA) Symposium, A.S.C.E. Environmental & Water Resources Institute (EWRI) Annual Conference*, Roanoke, Virginia, May 19-22, 2002.

[16] B.A. Tolson, H.R. Maier, A.R. Simpson, and B.J. Lence. Genetic algorithms for reliability-based optimization of water distribution systems. *Journal of Water Resources Planning and Management-Asce*, 130:63–72, 2004.

[17] Z.Y. Wu and A.R. Simpson. Competent genetic-evolutionary optimization of water distribution systems. *Journal of Computing in Civil Engineering*, 15:89–101, 2001.