Multi-objective Nonlinear Model Predictive Control: Lexicographic Method

Tao ZHENG, Gang WU, Guang-Hong LIU and Qing LING
University of Science and Technology of China
China

1. Introduction

The design of most process control is essentially a dynamic multi-objective optimization problem (Meadowcroft *et al.*, 1992), sometimes with nonlinear characters, and in which both economic benefit and social benefit should be considered. Commonly speaking, there are contradictory objectives such as quantity of products, quality of products, safety of manufacturing, cost of manufacturing, environment protection and so on. Since the different relative importance of these objectives cannot be ignored in the process of the controller design, we should manage the different priority of each objective correctly and exactly. Therefore, multivariable process control could be formulated as a complicated dynamic multi-objective optimization problem.
Traditionally, a multi-objective control problem could be transformed into a single-objective dynamic optimization with the quadratic objective function, where the weights denote the different relative importance of different objectives. This method is easy to understand, but the value of the weight coefficients usually could be only decided by try-and-error method, based on engineering experiences, repeating simulations and other information, while there is no accurate theoretical analysis of these weight coefficients yet. So it can be seen that, the design process of the traditional method is complicated and time-consuming indeed. Especially, when the situation of manufacturing changes (such as sudden load increasing of a power supplier and so on), it is very hard for operators to renew the weights rapidly. Therefore, a new framework of multi-objective controller is desired, it should be driven by the relative importance of different objectives, which reflect the practical requirement of control problems, and it also should be convenient to redesign for engineers and operators, when the values or priorities of the objectives are changed.
Using lexicographic method, which also called completely stratified method, Meadowcroft *et al.* proposed a priority-driven framework of controller: Modular Multivariable Controller (MMC), and analyzed its steady-state properties (Meadowcroft *et al.*, 1992). It sorts objectives sequentially according to their relative importance, and then satisfies them as many as possible in the corresponding control modules by the order as Fig. 1., where one module handles with only one objective. Later, because of its advantages, researchers have extended MMC to the dynamic optimization of linear systems with model predictive control (MPC) and other controllers in past years (Ocampo-Martinez *et al.*, 2008, Wu *et al.*, 2000).

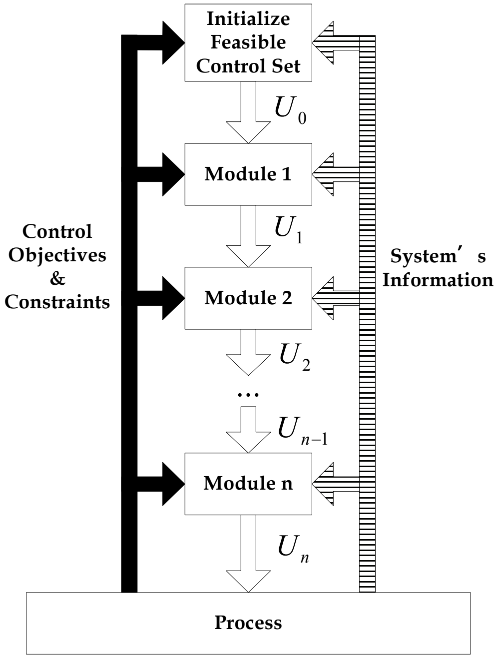

Fig. 1. Lexicographic structure of Modular Multivariable Controller

While has the mentioned advantages, the lexicographic structure still has some serious problems. First, in this structure, the priorities of objectives are absolute and rigid, if an objective cannot be completely satisfied (usually a objective with a setpoint form or an extremum form), the objectives with lower priorities than it will not be considered any more, even if they can be satisfied without any bad influence on other objectives. Second, in some practical cases, it is hard to distinguish the difference on priorities between some "parallel objectives", and it is also not necessary indeed. In practical need, the number of priorities is no need to equals to the number of objectives, it can be smaller, that means a certain priority may have several objectives. So sometimes, the partially stratified structure is more flexible than completely stratified structure (lexicographic structure), the number of priorities could be determined by the essential control problem, and the objectives with relatively lower importance can be handled in a same priority together for simplicity.

Besides the structure of the controller, the control algorithm is also important in multi-objective control nowadays. Since the control demand of modern process industry is heightening continuously, nonlinearity of systems cannot be ignored in controller design, to utilize the advantages of MPC in process control, nonlinear model predictive control (NMPC) now are developing rapidly (Alessio & Bemporad, 2009, Cannon, 2004). Naturally, for multi-objective NMPC in many industrial cases, the priority-driven method is also necessary. We have tried to combine lexicographic structure (or partially stratified structure) and NMPC directly, as dynamic MMC of linear systems (Ocampo-Martinez *et al.*, 2008, Wu *et al.*, 2000). But the nonlinear character makes it difficult to obtain analytic solution of control problem, and the modular form for stratified structure seems to be too complex for nonlinear systems in some extent. Both these facts lead us to find a new way for the nonlinear multi-objective control problem. Genetic algorithm (GA) now is recognized as an efficient computing means for single-objective NMPC already (Yuzgec *et al.*, 2006), and it also can be used to solve lexicographic optimization (Coello, 2000). So, in this chapter, a series of dynamic coefficients are used to make up a combined fitness function of GA, which makes GA be able to handle lexicographic optimization or partially stratified optimization in multi-objective NMPC. It can solve the nonlinear multi-objective control problem in the same way as MMC, but with a simple structure and much little computational load.

Since the partially stratified structure could be modified from lexicographic structure easily (or lexicographic structure can be seen as a special case of partially stratified structure), in this chapter, we will introduce lexicographic method as the main content, then the corresponding content of partially stratified method can be obtained directly. The rest of this chapter is organized as follow, Section 2 will introduce the basic theory of lexicographic optimization and partially stratified optimization, then the modified GA for them will be proposed in Section 3, lexicographic NMPC and partially stratified NMPC based on the proposed GA will be studied in Section 4, using the control problem of a two-tank system as a case study. At last, conclusions and acknowledgements will be done in Section 5.

2. Lexicographic optimization and partially stratified optimization

2.1 Lexicographic optimization

Lexicographic optimization is a strategy of multivariable optimization derived from priority-driven thought, without loss of generality, we just considers the minimization of multi-objective problem in this chapter.

Suppose a complex goal $g=\{g_1,g_2,\cdots,g_n\}$ contains n objectives, and the subscript also describe the relative importance of each objective, where g_1 is the most important one and g_{i-1} is always more important than g_i. The solution $g^{(1)}=\{g_1^{(1)},g_2^{(1)},\cdots,g_n^{(1)}\}$ is better than the solution $g^{(2)}=\{g_1^{(2)},g_2^{(2)},\cdots,g_n^{(2)}\}$, if and only if $g_k^{(1)}<g_k^{(2)}$ and $g_i^{(1)}=g_i^{(2)}=\min g_i$ hold for certain $k\le n$ and all $i<k$. It means that, before priority k, all objectives are satisfied in both $g^{(1)}$ and $g^{(2)}$, but on priority k, $g^{(1)}$ is preferred to $g^{(2)}$, so it is a better solution for the whole multi-objective optimization, no matter what will be on the objectives of lower priorities than g_k. Thus the formulation of the lexicographic minimization problem can be written as follow (Meadowcroft *et al.*, 1992):

$$\begin{aligned}&\min g_k, k=1,2,\cdots n\\ &\text{s. t. } \; g_i=\min g_i, i<k\end{aligned} \tag{1}$$

Therefore, lexicographic optimization would be defined as the computing process of a lexicographic minimum solution of a multi-objective problem (or sometimes maybe a maximum solution). This solution usually is not the optimal solution of any quadratic objective function and vice versa. As mentioned in Section 1, in lexicographic optimization, one priority can have only one objective, so it also called completely stratified optimization. If needed, the readers can find more about the definition of lexicographic optimization from other references.

2.2 Partially stratified optimization

Still suppose the complex goal $g=\{g_1,g_2,\cdots,g_n\}$ contains n objectives, and all objectives need to be minimized. If these n objectives can be divided into m priorities ($m \le n$), the complex goal can be rewritten as $G=\{G_1,G_2,\cdots,G_m\}$, where $G_i=\sum_j \lambda_{ij} g_{ij}$ is a combined goal of a certain priority i that contains j goals, and the goals in the same priority still could be combined with weight coefficients.

Because the relation between priorities is still lexicographic, the subscript of G_i also describes the relative importance, where G_1 is the most important and G_{i-1} is always more important than G_i. The solution $G^{(1)}=\{G_1^{(1)},G_2^{(1)},\cdots,G_m^{(1)}\}$ is better than the solution $G^{(2)}=\{G_1^{(2)},G_2^{(2)},\cdots,G_m^{(2)}\}$, if and only if $G_k^{(1)}<G_k^{(2)}$ and $G_i^{(1)}=G_i^{(2)}=\min G_i$ hold for certain $k \le m$ and all $i<k$. Similar to de definition of lexicographic minimization in (1), the partially stratified optimization now can be defined as the computing process of a partially stratified minimum solution:

$$\begin{aligned} &\min G_k, k=1,2,\cdots m \\ &\text{s. t. } \; G_i=\min G_i, i<k \end{aligned} \tag{2}$$

Simply speaking, partially stratified multi-objective optimization here means lexicographic method between priorities and traditional weight coefficients method on goals in the same priority. Specially, if the number of priorities equals to the number of goals ($m=n$), partially stratified multi-objective optimization will equal to lexicographic multi-objective optimization.

3. GA for lexicographic optimization and partially stratified optimization

3.1 GA for lexicographic optimization

In GA, the survival opportunity and competitiveness of individuals are only determined by fitness function. So the key to a lexicographic genetic algorithm is a special fitness function, which is suitable for lexicographic optimization for multi-objective control.

Still suppose a complex goal $g=\{g_1,g_2,\cdots,g_n\}$ contains n objectives, and the fitness function of each objective is $F_i \in [0,1], 1 \le i \le n$, while $F_i=1$ means objective i has been completely satisfied. Since lexicographic optimization can only deal with a certain objective when all the objectives with higher priority have been achieved already, a series of dynamic coefficients is introduced to describe this decision procedure:

$$\delta_i=\begin{cases} 1 & i=1 \\ 1 & F_1=F_2=\cdots F_{i-1}=1, 2 \le i \le n \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

Here, since F_1 is the fitness function of the most important objective, which has no objective with higher priority than itself, so $\delta_1=1$ should be held all the time. Then the combined fitness function could be:

$$F=\delta_1F_1+\delta_2F_2+\cdots+\delta_nF_n=\sum_{i=1}^{n}\delta_iF_i \tag{4}$$

Using this lexicographic combined fitness function in multi-objective GA, the lexicographic optimum solution can be obtained directly, and there are no special rules on coding method, crossover operator, mutation operator or any other parameters of GA. Constraints on the value of individuals can be matched by lethal penalty or other kinds of penalties in GA, and to ensure the solution's convergence to the optimal solution, the best individual should be remained in every evolution. For the convenience to readers, we will describe the steps of this modified GA for lexicographic optimization briefly as follow:

Step 1: create M initial parent individuals randomly.

Step 2: create M offspring individuals by crossover operator, mutation operator with proper operation on constraints.

Step 3: compute the fitness of all the 2M individuals (parents and offspring) respectively by (4).

Step 4: choose M individuals with higher fitness among the 2M individuals as new parent individuals.

Step 5: if the ending condition for evolution computation is matched, output the individual of the highest fitness, or return to Step 2.

3.2 GA for partially stratified optimization and some discussion

The only difference between GA for lexicographic optimization (LMGA) and partially stratified GA (PSMGA) is the definition of F_i, in LMGA it is the fitness function for a single objective, but in PSMGA it is the fitness function for all the objectives in a same priority. Since the form of fitness fuction is depend on the problem will be solved, the fitness fuction of NMPC based on proposed GA will be introduced in detail later in the Section 4.

Both LMGA and PSMGA are quite different from many other multi-objective GA. Traditional multi-objective GA usually need to find out Pareto Surface (Coello, 2000), which contains a set of Pareto optimal solutions, then choose a best solution by the given criterion. But LMGA and PSMGA don't need this additional selection after evolutionary computing, since the optimal solution of the multi-objective optimization can be obtained directly. For controller design, what we need is just an optimal solution, no matter what the Pareto Surface is, so PSMGA and LMGA's disposal is quite suitable and time-saving.

In LMGA and PSMGA, if the priority order of objectives changes, we only need to modify the logical descriptions of the priorities in the combined fitness function, and if the value of objectives changes, we only need to modify the numerical description of the combined fitness function, while there is no parameters need to be tuned.

4. Multi-objective NMPC based on GA: a case study

4.1 The model of the two-tank system

To be used in this chapter to carry out simulations, the nonlinear model of a two-tank system in Fig. 2. would be introduced here as (5), which is obtained by mechanism modelling, and the sample time of this discrete system is 1 second. Here outputs $y_1(k), y_2(k)$ denote the height of water in two tanks T_1 and T_2 respectively, and control input $u(k)$ is the water fed into tank T_1 from the valve V_3. The manual valve V_1 and V_2 are kept open at the maximal position all the time, and magnetic valve V_3 is controlled by PC to be the actuator of the system, to control the fluid speed of water from pulp P_1. (5-1) and (5-2) is the fluid mechanical character of T_1 and T_2 and (5-3) is the constraints on outputs, input, and the increment of input respectively. For convenience, all the variables in the model are normalized to the scale 0%-100%.

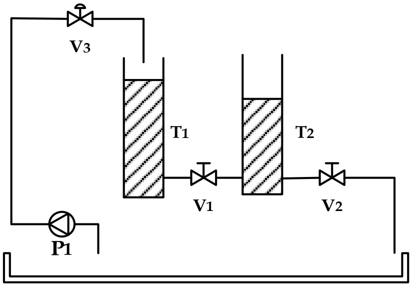

Fig. 2. Structure of the two-tank system

$$y_1(k+1) = y_1(k) + 0.01573 \cdot u(k) - 0.2232 \cdot \mathrm{sign}(y_1(k) - y_2(k)) \cdot \sqrt{|y_1(k) - y_2(k)|} \quad (5\text{-}1)$$

$$y_2(k+1) = y_2(k) - 0.1191\sqrt{y_2(k)} + 0.2232 \cdot \mathrm{sign}(y_1(k) - y_2(k))\sqrt{|y_1(k) - y_2(k)|} \quad (5\text{-}2)$$

$$\begin{aligned} \text{s. t.} \quad & y_1(k), y_2(k) \in [0\%, 100\%] \\ & u(k) \in [20\%, 80\%] \\ & u(k) - u(k-1) = \Delta u(k) \in [-5\%, 5\%] \end{aligned} \quad (5\text{-}3)$$

where the sign function is

$$\mathrm{sign}(x) = \begin{cases} 1 & x \geq 0 \\ -1 & x < 0 \end{cases}.$$

4.2 The basic control problem of the two-tank system

The NMPC of the two-tank system would have two forms of objective functions, according to two forms of practical goals in control problem: setpoint and restricted range.

For goals in the form of restricted range $g: y_i(k) \in [y_{i-low}, y_{i-high}], i = 1,2$, suppose the predictive horizon contains p sample time, k is the current time and the predictive value at time k of future output is denoted by $\hat{y}_i(\cdot | k)$, the objective function can be chosen as:

$$J(k) = \sum_{j=1}^{p} [\mathrm{pos}(\hat{y}_i(k+j|k) - y_{i-high}) + \mathrm{neg}(\hat{y}_i(k+j|k) - y_{i-low})]^2, i = 1,2 \quad (6)$$

where the positive function and negative function are

$$\mathrm{pos}(x) = \begin{cases} x & x \geq 0 \\ 0 & x < 0 \end{cases}, \quad \mathrm{neg}(x) = \begin{cases} 0 & x \geq 0 \\ -x & x < 0 \end{cases}.$$

In (6), if the output is in the given restricted range, the value of objective function J(k) is zero, which means this objective is completely satisfied.

For goals in the form of setpoint $g: y_i(k) = y_{i-set}, i = 1,2$, since the output cannot reach the setpoint from recent value immediately, we can use the concept of reference trajectories, and the output will reach the set point along it. Suppose the future reference trajectories of output $y_i(k)$ are $w_i(k), i = 1,2$, in most MPC (NMPC), these trajectories often can be set as exponential curves as (7) and Fig. 3. (Zheng *et al.*, 2008)

$$w_i(k+j) = \alpha_i \cdot w_i(k+j-1) + (1-\alpha_i) \cdot y_{i-set}, 1 \le j \le p, i = 1,2 \tag{7}$$

where

$$w_i(k) = y_i(k) \text{ and } 0 \le \alpha_i < 1 .$$

Then the objective function of a setpoint goal would be:

$$J(k) = \sum_{j=1}^{p} (\hat{y}_i(k+j\,|\,k) - w_i(k+j))^2, i = 1,2 \tag{8}$$

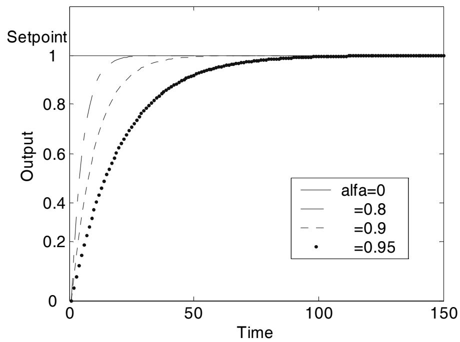

Fig. 3. Description of exponential reference trajectory

4.3 The stair-like control strategy

To enhance the control quality and lighten the computational load of dynamic optimization of NMPC, especially the computational load of GA in this chapter, stair-like control strategy (Wu *et al.*, 2000) could be used here. Suppose the first unknown increment of instant control input is $\Delta u(k) = u(k) - u(k-1)$, and the stair constant β is a positive real number, in stair-like control strategy, the future control inputs could be decided as follow (Wu *et al.*, 2000, Zheng *et al.*, 2008):

$$\Delta u(k+j) = \beta \cdot \Delta u(k+j-1) = \beta^j \cdot \Delta u(k), 1 \le j \le p-1 \tag{9}$$

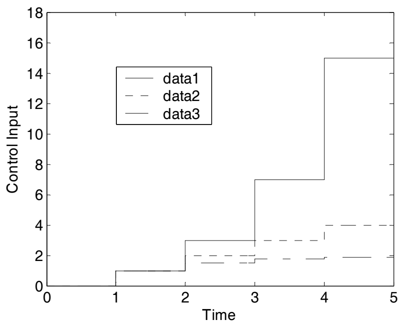

Fig. 4. Description of stair-like control strategy

With this disposal, the elements in the future sequence of control input $u(k)\quad u(k+1)\quad \cdots \quad u(k+p-1)$ are not independent as before, and the only unknown variable here in NMPC is the increment of instant control input $\Delta u(k)$, which can determine all the later control input. The dimension of unknown variable in NMPC now decreases from $i \cdot p$ to i remarkably, where i is only the dimension of control input, thus the computational load is no longer depend on the length of the predictive horizon like many other MPC (NMPC). So, it is very convenient to use long predictive horizon to obtain better control quality without additional computational load under this strategy. Because MPC (NMPC) will repeat the dynamic optimization at every sample time, and only $u(k) = \Delta u(k) + u(k-1)$ will be carried out actually in MPC (NMPC), this strategy is surely efficient here. At last, in stair-like control strategy, it also supposes the future increment of control input will change in the same direction, which can prevent the frequent oscillation of control input's increment, while this kind of oscillation is very harmful to the actuators of practical control plants. A visible description of this control strategy is shown in Fig. 4.

4.4 Multi-objective NMPC based on GA

Based on the proposed LMGA and PSMGA, the NMPC now can be established directly. Because NMPC is an online dynamic optimal algorithm, the following steps of NMPC will be executed repeatedly at every sample time to calculate the instant control input.

Step 1: the LMGA (PSMGA) initialize individuals as different $\Delta u(k)$ (with population M) under the constraints in (5-3) with historic data $u(k-1)$.

Step 2: create M offspring individuals by evolutionary operations as mentioned in the end of Section 3.1. In control problem, we usually can use real number coding, linear crossover, stochastic mutation and the lethal penalty in GA for NMPC. Suppose P_1, P_2 are parents and O_1, O_2 are offspring, linear crossover operator $0 < \gamma < 1$ and stochastic mutation operator ε is Gaussian white noise with zero mean, the operations can be described briefly as bellow:

$$\begin{aligned} O_1 &= \gamma \cdot P_1 + (1-\gamma) \cdot P_2 + \varepsilon_1 \\ O_2 &= \gamma \cdot P_2 + (1-\gamma) \cdot P_1 + \varepsilon_2 \end{aligned}, \quad 0 < \gamma < 1 \tag{10}$$

Step 3: predictions of future outputs ($\hat{y}_i(k+1|k)\quad \hat{y}_i(k+2|k)\quad \cdots \quad \hat{y}_i(k+p|k)$, i=1,2) are carried out by (5-1) and (5-2) on all the 2M individuals (M parents and M offspring), and their fitness will be calculated. In this control problem, the fitness function F of each objective is transformed from its objective function J easily as follow, to meet the value demand of $F \in [0,1]$, in which J is described by (6) or (8):

$$F = 1/(J+1) \tag{11}$$

To obtain the robustness to model mismatch, feedback compensation can be used in prediction, thus the latest predictive errors $e_i(k) = y_i(k) - \hat{y}_i(k|k-1), i = 1,2$ should be added into every predictive output $\hat{y}_i(k+j|k), 1 = 1,2, 1 \le j \le p$.

Step 4: the M individuals with higher fitness in the 2M individuals will be remained as new parents.

Step 5: if the condition of ending evaluation is met, the best individual will be the increment of instant control input $\Delta u(k)$ of NMPC, which is taken into practice by the actuator. Else, the process should go back to Step 2, to resume dynamic optimization of NMPC based on LMGA (PSMGA).

4.5 Simulations and analysis of lexicographic multi-objective NMPC

First, the simulation about lexicographic Multi-objective NMPC will be carried out. Choose control objectives as: $g_1 : y_1(k) \in [40\%,60\%]$, $g_2 : y_2(k) \in [20\%,40\%]$, $g_3 : y_2 = 30\%$. Consider the physical character of the system, two different order of priorities can be chosen as: [A]: $g_1 > g_2 > g_3$, [B]: $g_2 > g_1 > g_3$, and they will have the same initial state as $y_1(0) = 80\%$, $y_2(0) = 0\%$ and $u(0) = 20\%$. Parameters of NMPC are $\alpha = 0.95, \beta = 0.85$ for both y_1 and y_2, and parameters of GA are $\gamma = 0.9$, while ε is a zero mean Gaussian white noise, whose variance is 5. Since the feasible control input set is relatively small in our problem according to constraints (5-3), it is enough to have only 10 individuals in our simulation, and they will evolve for 20 generations. While in process control practice, because the sample time is often has a time scale of minutes, even hours, we can have much more individuals and they can evolve much more generations to get a satisfactory solution. (In following figures, dash-dot lines denote g_1, g_2, dot line denote g_3 and solid lines denote y_1, y_2, u)

Compare Fig. 5. and Fig. 6. with Fig. 7. and Fig. 8., although the steady states are the same in these figures, the dynamic responses of them are with much difference, and the objectives are satisfied as the order appointed before respectively under all the constraints. The reason of these results is the special initial state: $y_1(0)$ is higher than g_1 (the most important objective in order [A]: $g_1 > g_2 > g_3$), while $y_2(0)$ is lower than g_2 (the most important objective in order [B]: $g_2 > g_1 > g_3$). So the most important objective of the two orders must be satisfied with different control input at first respectively. Thus the difference can be seen from the different decision-making of the choice in control input more obviously: in Fig. 5. and Fig. 6. the input stays at the lower limit of the constraints at first to meet g_1, while in Fig. 7. and Fig. 8. the input increase as fast as it can to satisfy g_2 at first. The lexicographic character of LMGA is verified by these comparisons.

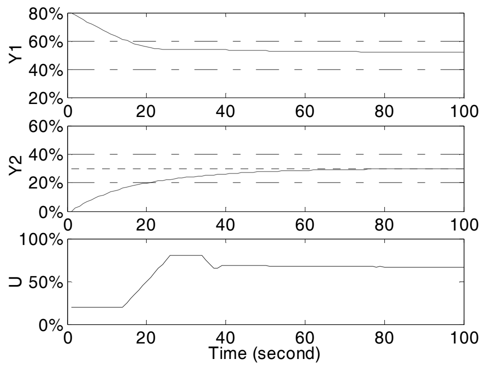

Fig. 5. Control simulation: priority order [A] and p=1

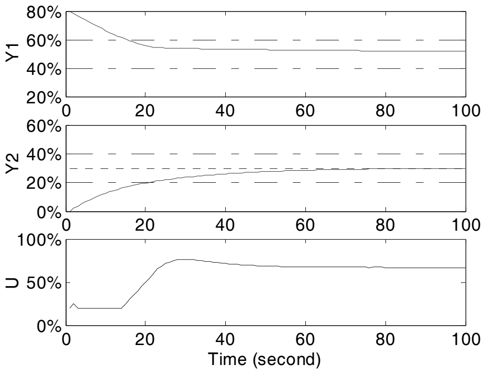

Fig. 6. Control simulation: priority order [A] and p=20

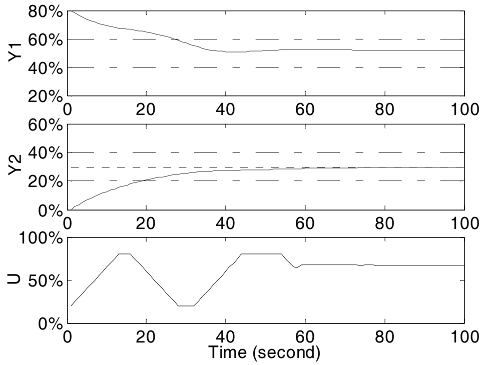

Fig. 7. Control simulation: priority order [B] and p=1

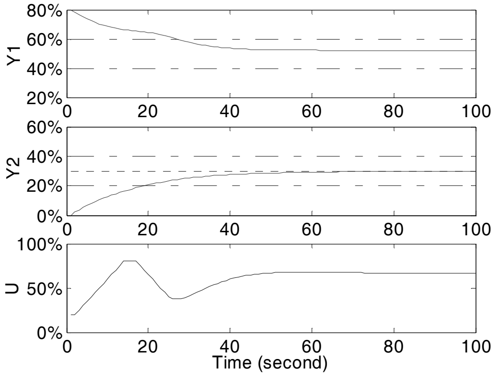

Fig. 8. Control simulation: priority order [B] and p=20

And the difference in control input with different predictive horizon can also be observed from above figures: the control input is much smoother when the predictive horizon becomes longer, while the output is similar with the control result of shorter predictive horizon. It is the common character of NMPC.

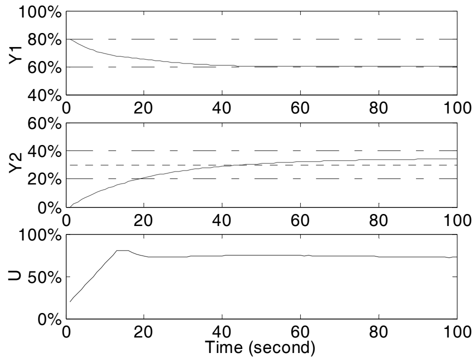

Fig. 9. Control simulation: when an objective cannot be satisfied

In Fig. 9., g_1 is changed as $y_1 \in [60\%,80\%]$, while other objectives and parameters are kept the same as those of Fig. 6., so that g_3 can't be satisfied at steady state. The result shows that y_1 will stay at lower limit of g_1 to reach set-point of g_3 as close as possible, when g_1 must be satisfied first in order [A]. This result also shows the lexicographic character of LMGA obviously.

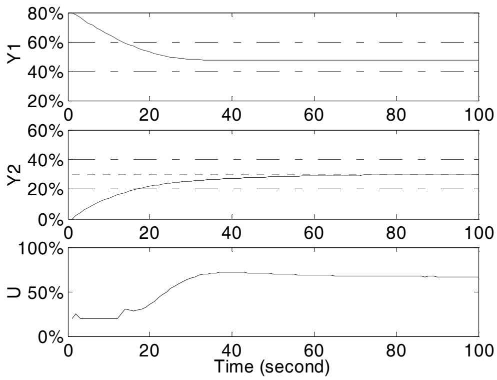

Fig. 10. Control simulation: when model mismatch

Finally, we would consider about of the model mismatch, here the simulative plant is changed, by increasing the flux coefficient 0.2232 to 0.25 in (5-1) and (5-2), while all the objectives, parameters and predictive model are kept the same as those of Fig. 6. The result in Fig. 10. shows the robustness to model mismatch of the controller with error compensation in prediction, as mentioned in Section 4.4.

4.6 Simulations and analysis of partially stratified multi-objective NMPC

To obtain evident comparison to Section 4.5, simulations are carried out with the same parameters ($\alpha = 0.95, \beta = 0.85$ for both y_1 and y_2, predictive horizon p=20 and the same GA parameters), and the only difference is an additional objective on y_1 in the form of a setpoint.

The four control objectives now are $g_1 : y_1(k) \in [40\%, 60\%]$, $g_2 : y_2(k) \in [20\%, 40\%]$, $g_3 : y_2 = 30\%$, $g_4 : y_1 = 50\%$, and then choose the new different order of priorities as: [A]: $g_1 > g_2 > g_3 > g_4$, [B]: $g_2 > g_1 > g_3 > g_4$, if we still use lexicographic multi-objective NMPC as Section 4.5, the control result in Fig. 11. and Fig. 12. is completely the same as Fig. 6. and Fig. 8., when there are only three objectives g_1, g_2, g_3. That means, the additional objective g_4 (setpoint of y_1) could not be considered by the controller in both situations above, because the solution of g_3 (setpoint of y_2) is already a single-point set of u. (In following figures, dash-dot lines denote g_1, g_2, dot line denote g_3, g_4 and solid lines denote y_1, y_2, u)

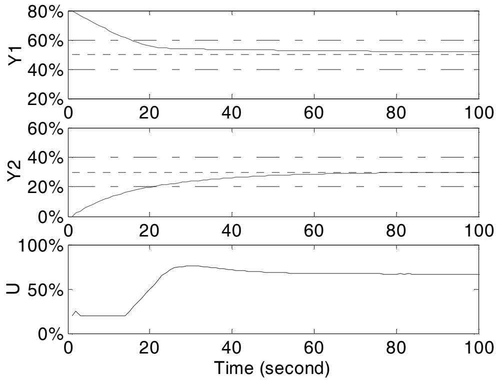

Fig. 11. Control simulation: priority order [A] of four objectives, NMPC based on LMGA

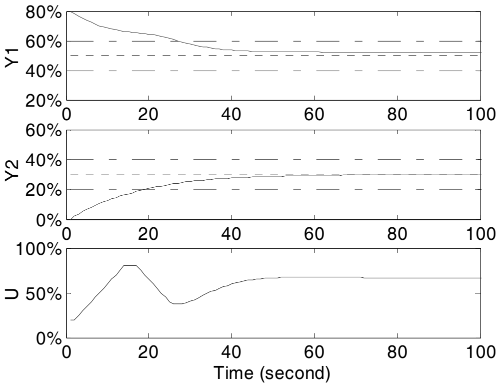

Fig. 12. Control simulation: priority order [B] of four objectives, NMPC based on LMGA

In another word, in lexicographic multi-objective NMPC based on LMGA, if optimization of an objective uses out all the degree of freedom on control inputs (often an objective in the form of setpoint), or an objective cannot be completely satisfied (often an objective in the form of extremum, such as minimization of cost that can not be zero), the objectives with lower priorities will not be take into account at all. But this is not rational in most practice cases, for complex process industrial manufacturing, there are often many objectives in the form of setpoint in a multi-objective control problem, if we handle them with the lexicographic method, usually, we can only satisfy only one of them. Take the proposed two-tank system as example, g_3 and g_4 are both in the form of setpoint, seeing about the steady-state control result in Fig. 13. and Fig. 14., if we want to satisfy $g_3 : y_2 = 30\%$, then y_1 will stay at 51.99%, else if we want to satisfy $g_4 : y_1 = 50\%$, then y_2 will stay at 28.92%, the errors of the dissatisfied output are both more than 1%.

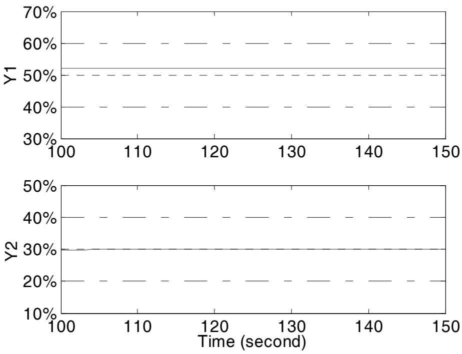

Fig. 13. Steady-state control result when g_3 is completely satisfied

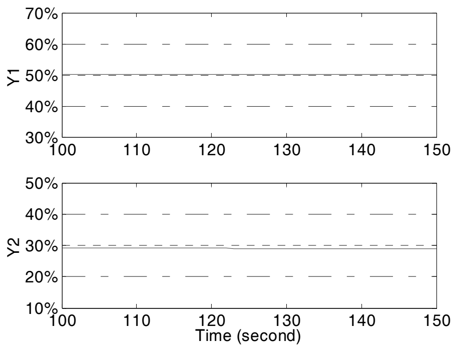

Fig. 14. Steady-state control result when g_4 is completely satisfied

In the above analysis, the mentioned disadvantage comes from the absolute, rigid management of lexicographic method, if we don't develop it, NMPC based on LMGA can only be used in very few control practical problem. Actually, in industrial practice, objectives in the form of setpoint or extremum are often with lower importance, they are usually objectives for higher demand on product quality, manufacturing cost and so on, which is much less important than the objectives about safety and other basic manufacturing demand. Especially, for objectives in the form of setpoint, under many kinds of disturbances, it always can not be accurately satisfied while it is also not necessary to satisfy them accurately.

A traditional way to improve it is to add slack variables into objectives in the form of setpoint or extremum. Setpoint may be changed into a narrow range around it, and instead of an extremum, the satisfaction of a certain threshold value will be required. For example, in the two-tank system's control problem, setpoint $g_3 : y_2 = 30\%$ could be redefined as $g_3 : y_2 \in 30\% \pm 1\%$.

Another way is modified LMGA into PSMGA as mentioned in Section 3, because sometimes there is no need to divide these objectives with into different priorities respectively, and they are indeed parallel. Take order [A] for example, we now can reform the multi-objective control problem of the two-tank system as: $G_1 > G_2 > G_3 = g_1 > g_2 > \lambda_3 g_3 + \lambda_4 g_4$. Choose weight coefficients as $\lambda_3 = 30, \lambda_4 = 1$ and other parameters the same as those of Fig. 6., while NMPC base on PSMGA has the similar dynamic state control result to that of NMPC based on LMGA, the steady state control result is evidently developed as in Fig. 15. and Fig. 16., y_1 stays at 50.70% and y_2 stays at 29.27%, both of them are in the 0.8% neighborhood of setpoint in g_3, g_4 .

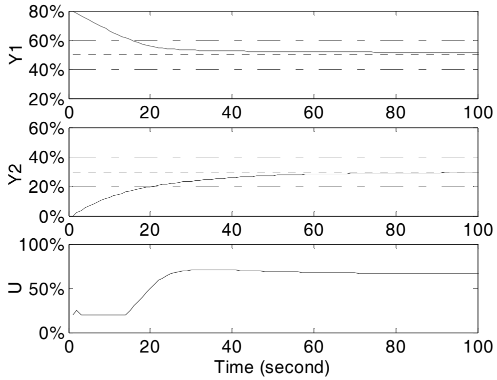

Fig. 15. NMPC based on PSMGA: priority order [A]

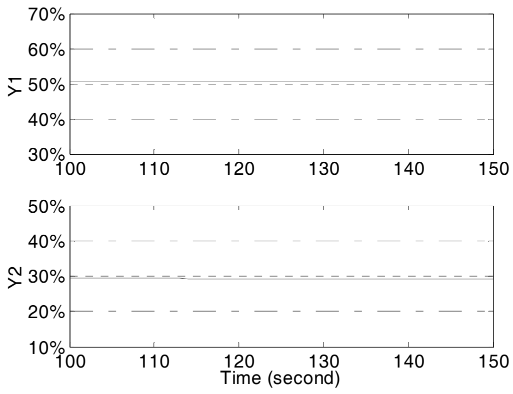

Fig. 16. Steady-state control result of NMPC base on PSMGA

4.7 Some discussions

In the above simulative examples, there is only one control input, but for many practical systems, coordinated control of multi-input is also a serious problem. The brief discussions on multi-input proposed NMPC can be achieved if we still use priorities for inputs. If all the inputs have the same priority, in another word, it is no obvious difference among them in economic cost or other factors, we can just increase the dimension of GA's individual. But, in many cases, the inputs actually also have different priorities: for certain output, different input often has different gain on it with different economic cost. The cheap ones with large gain are always preferred by manufacturers. In this case, we can form anther priority list, and then inputs will be used to solve the control problem one by one, using single input NMPC as the example in Section 4, that can divide an MIMO control problem into some SIMO control problems.

We should point out that, the two kinds of stratified structures proposed in this paper are basic structures for multi-objective controllers, though we use NMPC to realize them in this chapter, they are independent with control algorithms indeed. For certain multi-objective control problem, other proper controllers and computational method can be used.

Another point must be mentioned is that, NMPC proposed in this paper is based on LMGA and PSMGA, because it is hard for most NMPC to get an online analytic solution. But the LMGA and PSMGA are also suitable for other control algorithms, the only task is to modify the fitness function, by introducing the information from the control algorithm which will be used.

At last, all the above simulations could been done in 40-200ms by PC (with 2.7 GHz CPU, 2.0G Memory and programmed by Matlab 6.5), which is much less than the sample time of the system (1 second), that means controllers proposed in this chapter are actually applicable online.

5. Conclusion

In this chapter, to avoid the disadvantages of weight coefficients in multi-objective dynamic optimization, lexicographic (completely stratified) and partially stratified frameworks of multi-objective controller are proposed. The lexicographic framework is absolutely priority-driven and the partially stratified framework is a modification of it, they both can solve the multi-objective control problem with the concept of priority for objective's relative importance, while the latter one is more flexible, without the rigidity of lexicographic method.

Then, nonlinear model predictive controllers based on these frameworks are realized based on the modified genetic algorithm, in which a series of dynamic coefficients is introduced to construct the combined fitness function. With stair–like control strategy, the online computational load is reduced and the performance is developed. The simulative study of a two-tank system indicates the efficiency of the proposed controllers and some deeper discussions are given briefly at last.

The work of this chapter is supported by Fund for Excellent Post Doctoral Fellows (K. C. Wong Education Foundation, Hong Kong, China and Chinese Academy of Sciences), Science and Technological Fund of Anhui Province for Outstanding Youth (08040106910), and the authors also thank for the constructive advices from Dr. De-Feng HE, College of Information Engineering, Zhejiang University of Technology, China.

6. References

Alessio A. & Bemporad A. (2009). A survey on explicit model predictive control. Lecture Notes in Control and Information Sciences (Nonlinear Model Predictive Control: Towards New Challenging Applications), Vol. 384, pp 345-369, ISSN 0170-8643.

Cannon M. (2004). Efficient nonlinear model predictive control algorithms. Annual Reviews in Control. Vol.28, No.2, pp229-237, ISSN 1367-5788.

Coello C. A. C. (2000). An Updated Survey of GA-Based Multiobjective Optimization Techniques, ACM Computing Surveys, Vol.32, No.2, pp109-143, ISSN 0360-0300.

Meadowcroft T. A.; Stephanopoulos G. & Brosilow C. (1992). The modular multivariable controller: I: steady-state properties. AIChE Journal, Vol.38, No.8, pp1254-1278, ISSN 0001-1541.

Ocampo-Martinez C.; Ingimundarson A.; Vicenç P. & J. Quevedo. (2008). Objective prioritization using lexicographic minimizers for MPC of sewer networks. IEEE Transactions on Control Systems Technology, Vol. 16, No.1, pp113-121, ISSN 1063-6536.

Wu G.; Lu X. D.; Ying A. G.; Xue M. S.; Zhang Z. G. & Sun D. M. (2000). Modular Multivariable Self-tuning Regulator. Acta Automatica Sinica, Vol.26, No.6, pp811-815, ISSN 0254-4156.

Yuzgec U.; Becerikli Y. & Turker M. (2006). Nonlinear Predictive Control of a Drying Process Using Genetic Algorithms, ISA Transactions, Vol.45, No.4, pp589-602, ISSN 0019-0578.

Zheng T.; Wu G.; He D. F.; Yue D. Z. (2008). An Efficient Model Nonlinear Predictive Control Algorithm Based on Stair-like Control Strategy, Proceedings of the 27th Chinese Control Conference, Vol.3, pp557-561, ISBN 9787811243901, Kunming, China, July, 2008, Beihang University Press, Beijing, China.

Model Predictive Control

Edited by Tao Zheng

ISBN 978-953-307-102-2
Hard cover, 304 pages
Publisher Sciyo
Published online 18, August, 2010
Published in print edition August, 2010

Frontiers of Model Predictive Control Robust Model Predictive Control Nonlinear Model Predictive Control Excellent Applications Guide for Researchers and Engineers Recent Achievements of Authors over the World Theory with Practical Examples Kinds of Algorithms for Choice