Robust MPC for nonlinear multivariable systems

Bouzouita Badreddine*, Bouani Faouzi§, Ksouri Mekki*

*École Nationale d’Ingénieurs de Tunis, Tunisia.
§Institut National des Sciences Appliquées et de Technologie, Tunis, Tunisia.
E-mail : Badreddine.Bouzouita@enit.rnu.tn, Faouzi.Bouani@insat.rnu.tn, Mekki.Ksouri@insat.rnu.tn

Abstract—In this work, a robust predictive controller of uncertain nonlinear multivariable systems is developed. The control design is based on Multi-Input Multi-Output (MIMO) Nonlinear Auto Regressive Moving Average (NARMA) model. To cope with uncertain dynamic behavior of the system, the structured uncertainty is adopted. In fact, the main limitation of the robust predictive controllers is the computational burden leading to a lack of on line implementation. In this work, an efficient method is proposed. This method is based on transformation variables which reduce the initial non-convex problem to a convex programming. The efficiency of the proposed method is tested and compared with LMI and genetic algorithms optimizers on benchmark functions. The robustness of the proposed control law is experimented on three tanks system.

I. PROBLEM FORMULATION

Linear MPC approach resorts to linear model to predict the future behavior of the system to be controlled. Although this controller have found successful applications [1], its success is restricted to given operating point. In fact, linear models are not able to describe the global behavior of the system over the whole operating range. This motivates the employ of nonlinear system description that leads to nonlinear model predictive control (NMPC) [2]. In this paper, the nonlinear MIMO NARMA model is adopted. The proposed representation has known considerable interest in applications due the fact that it deals with nonlinearity on input and output signals [3]–[5].

In addition to nonlinear behavior, many industrial process models are characterized by the presence of time-varying uncertainties such as unknown process parameters and external disturbances which, if not accounted for in the controller design, may cause performance deterioration and even closedloop instability.

A. MIMO NARMA representation

The presence of a numeric model is a necessary condition for the development of the predictive control. Since, it permits to predict the future behavior of the process. To cope with nonlinear multi-variable systems with \( n \) outputs and \( p \) inputs, the MIMO-NARMA model is adopted in this work. Indeed, this last provides a unified representation for a wide class of non-linear systems [6], [7]. The outputs of the system are given by:

\[
y_j(k) = f_j(y(k-1), ..., y(k-ny), U(k-1), ..., U(k-nu))
\]

where \( j = 1, ..., n \); \( Y(k) = [y_1(k), ..., y_n(k)] \) \( \in \mathbb{R}^n \) and \( U(k) = [u_1(k), ..., u_p(k)] \) \( \in \mathbb{R}^p \) are system outputs and inputs, respectively; \( ny \) and \( nu \) are their associated maximum lags; \( f_j(.) \) are unknown nonlinear functions.

Leontaritis and Billings have been demonstrated that polynomial representation of NARMA model work well in practical application [7]. Thus, \( f_j(.) \) is expressed as a polynomial of degree \( L \) (where \( L \) is the degree of the nonlinearity):

\[
f_j(x_1, x_2, ..., x_n) = \sum_{i}^{N} \theta_{ji} \prod_{p=1}^{q_1} x_{p_1} \prod_{p=2}^{q_2} x_{p_2} \cdots \prod_{p=r}^{q_r} x_{p_r}
\]

with \( q_1, q_2, ..., q_r \geq 0 \) and \( q_1 + q_2 + ... + q_r \leq L \)

Then, from equations (1) and (2), the outputs of the system can be rearranged in a compact form:

\[
Y(k) = \theta \phi(k)
\]

where \( \theta \) is a matrix of scalar parameters and \( \phi(k) \) represents the data vector which includes the past values of inputs and outputs:

\[
\theta = \begin{bmatrix}
\theta_{10} & \cdots & \theta_{1N} \\
\vdots & \ddots & \vdots \\
\theta_{n0} & \cdots & \theta_{nN}
\end{bmatrix}
\]

\[
\phi(k) = [1, u_1(k-1), ..., u_p(k-nu), y_1(k-1), ..., y_n(k-ny), u_1^r(k-p) \times \cdots \times u_p^r(k-q), ..., y_1^r(k-h) \times \cdots \times y_n^r(k-s)]^T
\]

Note that the polynomial model is nonlinear in the output and input variables but linear in the parameters. Therefore, the set of coefficients can be estimated by a Least-Squares (LS) algorithm.

The robustification of the predictive controller consists to take into account, in an explicit manner, the uncertainties at the time of calculation of the control law. Most predictive controllers consider additive uncertainties [8]–[11]. However, this type of uncertainties is limited to measure errors on output signals and it is not suitable to describe the uncertain behavior of the physical system. In the present paper, structured uncertainty is adopted. This last is adequate to model the uncertain dynamic behavior of the system [12], [13].

To deal with uncertain behavior of physical system, structured uncertainty is considered. Using a polytope
description of uncertainties, the future outputs of the process are given by:

\[ Y(k+j) = \sum_{i=1}^r \alpha_i \theta_i \phi(k+j) \]  

(6)

where \( \alpha_i \geq 0, \sum_{i=1}^r \alpha_i = 1 \).

B. Control law

To indicate how well the process follows the desired trajectory, predictive control employs a cost function. This last depends on the future error between output signals and setpoints, and future increment controls to be optimized. In this work, the cost function to be optimized is as follows:

\[ J = \sum_{i=1}^{N_y} [Y(k+i) - W(k+i)]^T Y(k+i) - W(k+i) ] + \sum_{j=0}^{N_u-1} [\Delta U(k+j)] R [\Delta U(k+j)]^T \]

(7)

where \( Y(k+i) = [y_1(k+i), ..., y_n(k+i)] \), \( W(k+i) = [w_1(k+i), ..., w_n(k+i)] \) and \( \Delta U(k+j) = [\Delta u_1(k+j), ..., \Delta u_p(k+j)] \) are, respectively, the vectors of output predictions, setpoints and future increment of inputs; \( N_y \) is the output prediction horizon, \( N_u \) is the control horizon and \( R \) is the control weighting diagonal matrix.

Robust predictive control is based on worst case strategy. The control represents the best solution for the worst case defined by the set of uncertain models [10], [14]. Hence, the input control is obtained by the resolution of the following min-max optimization problem:

\[ \min_{\Delta U \in \Omega} - \max_{\alpha_{j=1,...,r}} J(\Delta U, \theta) \]

(8)

where \( \Omega \) is the set of constraints on the input/output signals.

Robust predictive control algorithm suffers from a great computational burden leading to a lack of on line implementation [8], [15]. In fact, many authors have proposed the linear matrix inequality (LMI) optimization technique to solve the min-max problem [16]–[19]. This method requires a substantial computing time which increases exponentially with control horizon and the number of variables to be optimized [19]. To overcome this problem, Generalized Geometric Programming (GGP) can be used [20]. The adopted optimization method is addressed to problems of minimizing or maximizing a multivariate polynomial under polynomial constraints. Indeed, this kind of problem is encountered in a wide variety of applications in production planning, engineering design, risk management, etc [21], [22].

This paper is organized as follows: Section II presents the GGP optimization algorithm. The effectiveness of this algorithm opposite LMI and genetic algorithms is also exhibited in this section. In section III, the implementation and robustness of the proposed control law are demonstrated through simulations using three tanks system example. Finally in section IV, the conclusions are presented.

II. GLOBAL SOLUTION

In the present work, the Generalized Geometric Programming optimization method is adopted. The proposed method is addressed to solve non-convex problems of which the objective function and constraints are polynomials. The mathematical formulation of GGP is defined as follows [23]:

\[
\min \sum_{j=1}^{T_f} c_j z_j \\
\text{subject to} \\
\sum_{q=1}^{K} h_{kq} z_{kq} \leq l_k, \, k = 1, \ldots, K \\
\sum_{i}^{n} \alpha_i z_i = x_{1}^{\alpha_1} x_{2}^{\alpha_2} \cdots x_{n}^{\alpha_n} \\
z_{kq} = \beta_{k1} \beta_{k2} \cdots \beta_{kn} \\
x_i \leq x_i \leq \bar{x}_i 
\]

(9)

where \( c_j, h_{kq}, l_k, \alpha_{pi} \) and \( \beta_{kqj} \) are called signomial term.

Usually the domain is \( x_i \in \mathbb{R}^r \). This is, however, no essential restriction since simple translations of the variables can often be used to fulfill the requirement for variables originally taking negative values.

A. Convexification strategies

Several methods have been proposed for solving this kind of problem. These methods are based on variable transformations and some other techniques [23]–[27].

**Lemma 1:** For a twice-differentiable function \( f(x) = c \prod_{i=1}^{n} x_i^\alpha_i, \quad X = (x_1, ..., x_n) \), \( x_i \geq 0, \ s, \ c, \ \alpha_i \in \mathbb{R}, \forall i \), let \( H(X) \) be the Hessian matrix of \( f(X) \). The determinant of \( H(X) \) can be expressed as [27]:

\[ \det H(X) = (-c)^n \left( \prod_{i=1}^{n} \alpha_i x_i^{\alpha_i-2} \right) \left( 1 - \sum_{i=1}^{n} \alpha_i \right) \]

(10)

**Remark 1:** if \( c \geq 0, x_i \geq 0, \) and \( \alpha_i \leq 0 \) (for all \( i \)), then \( \det H(X) \geq 0 \) [27].

**Remark 2:** if \( c \leq 0, x_i \geq 0, \alpha_i \geq 0 \) (for all \( i \)) and \( 1 - \sum_{i=1}^{n} \alpha_i \geq 0 \), then \( \det H(X) \geq 0 \) [27].

Using remarks 1 and 2, we give the following propositions:

**Proposition 1:** A twice-differentiable function \( f(X) = c \prod_{i=1}^{n} x_i^\alpha_i \) is convex for \( c \geq 0, x_i \geq 0, \) and \( \alpha_i \leq 0 \) (for \( i = 1, \ldots, n \)) [27].

**Proposition 2:** A twice-differentiable function \( f(X) = c \prod_{i=1}^{n} x_i^\alpha_i \) is convex for \( c \leq 0, x_i \geq 0, \alpha_i \geq 0 \) (for \( i = 1, \ldots, n \)) and \( 1 - \sum_{i=1}^{n} \alpha_i \geq 0 \) [27].

The convexification strategy consists to transform each non-convex monomial term of the problem (9) to convex one. For instance, considering the following function:

\[ f(x_1, ..., x_n) = cx_1^{\alpha_1} x_2^{\alpha_2} \cdots x_n^{\alpha_n}, \]

(11)

then the convexification of this function depends of the sign of \( c \) and the values of \( r_i \):

- if \( c > 0 \) and \( r_i > 0 \) then new variables \( X_i \) are introduced according to \( x_i = \exp(X_i) \). Thus, the function given by (11) can be rewritten as:

\[ cx_1^{r_1} x_2^{r_2} \cdots x_n^{r_n} = c \exp(r_1 X_1 + r_2 X_2 + \ldots + r_n X_n) \]

(12)
which is the exponentiel of affine function. Therefore, it is convex.

- if $c < 0$, and "$r_i < 0$ or $\sum_{i=1}^{n} r_i \geq 1$" then let $x_i = x_i^\frac{r_i}{c}$ where $\sum_{i=1}^{n} r_i \leq R$. So, the function (11) is given by:

$$cx_1^{r_1}x_2^{r_2}...x_n^{r_n} = cX_1^{r_1}X_2^{r_2}...X_n^{r_n} \quad (13)$$

which is convex by proposition 2.

B. Evaluation of GGP

In order to evaluate the effectiveness of generalized geometric programming, the performance of this last is compared with Linear Matrix Inequality optimization algorithm and Genetic Algorithm (GA) through solving a set of benchmark problems listed in the appendix. To avoid misinterpretation of the optimization results related to the choice of any particular initial points, each of the algorithms was run 100 times from random initial points.

The following criteria summarize the results from 100 times minimization per function:

- Errors: it is the sum of errors between the reached solution and the global minima given in the appendix.
- The CPU time: is the total time (in second) put for 100 times minimization per function.

In fact, GA optimization algorithm requires some parametrization. For these benchmark functions, it is configured as follows:

- The real codification, arithmetic crossover and arithmetic mutation are used.
- Number of individuals in initial population is equal to 50.
- The algorithm is stopped when the maximal number of generation is reached, which equal to 100.

Table I presents the computational results obtained by GGP presented in this work, ‘GloptiPoly’ which based on LMI optimization algorithm [28], [29] and genetic algorithm.

From the errors obtained by GGP and LMI, we can conclude that both algorithms converge to global minimum whatever the starting points. Whereas, it is not the case for GA. Indeed, the error in the case of ‘Colville’ and ‘Rosenbrok’ functions is great. This due to the fact that GA is based on stochastic rules and decisions.

In Table I, the CPU times of the proposed GGP is more faster than LMI and GA. In fact, the computational burden of LMI increases exponentially with number of variables. However, the CPU time of GA depends on the number of generation and the number of individuals.

Therefore, the proposed optimization method (GGP) is an alternative for control fast systems and solve non-convex optimization problem.

C. Implementation of control law

The min-max optimization problem presented by equation (8) is bilevel [30]. It gives the solution of the best design in terms of future increments of control $\Delta U$ for the worst case defined by the uncertain model. Therefore, the equation (8) can also be expressed by the equations (14) and (15) as follows:

$$\min_{\Delta U} - \max_{\alpha_j=1...r} \ J(\Delta U, \theta) = \min_{\Delta U} J^*(\Delta U) \quad (14)$$

with

$$J^*(\Delta U) = \max_{\alpha_j=1...r} \ J(\Delta U, \theta) = \min_{\alpha_j=1...r} - J(\Delta U, \theta) \quad (15)$$

Equation (15) maximizes the objective function with respect to the uncertain parameter $\theta$, and after, minimizes it with respect to $\Delta U$ (equation (14)).

From relation (2), we can prove that the output prediction $y_i(k+j)$ is non-convex and under polynomial shape. Therefore, the criterion $J$ given by (7) is non-convex and it can be transformed to a convex function by using on each sigimonial term the correspond convexification rule as given in section II-A. After the transformation of the criterion, we can use a standard optimization technique to solve it. Consequently, the computation time is reduced and the global solution is reached.

III. SIMULATION STUDY

A. System description

In this section the performance of the developed controller is tested on interconnected tank system depicted in figure (1). The process is composed of three cylindric tanks numbered from 1 to 3 which are connected through valves $\mu_{13}$ and $\mu_{23}$. The valves $\mu_{10}$, $\mu_{20}$ and $\mu_{30}$ are the emptying valves to the main tank. Tanks 1 and 2 of section equal to 0.049 $m^2$ are fed into water respectively by respectively pump 1 and pump 2. The section of tank 3 is 0.0638 $m^2$.

The level for each tank depends on the sum of the water flowing into and flowing off the tank that can be adjusted by the flow rate of the pump 1 and pump 2. Then, the system can be conveniently represented by:

$$S_i \frac{dh_i}{dt} = q_1 - q_{i0} - q_{i3} \quad (16)$$

$$S_2 \frac{dh_2}{dt} = q_2 - q_{20} - q_{23} \quad (17)$$

$$S_3 \frac{dh_3}{dt} = q_{13} + q_{23} - q_{30} \quad (18)$$

where $h_i$ is the tank level, $q_1$ and $q_{i0}$ are the input flows, $S_j$ is the section of tank $j$ and $q_{ij}$ represents the water flow rate from tank $i$ to $j$ ($i$, $j$ = 1, 2, 3), which, according to Torricelli rule, is given by:

$$q_{ij} = s_{ij} \mu_{ij} \text{sign}(h_i - h_j) \sqrt{2g |h_i - h_j|} \quad (19)$$

with $s_{ij} = 6.36 \times 10^{-5} m^2$ is the section of valve, $g$ is the gravity coefficient and $\mu_{ij} \in [0, 1]$ (where $\mu_{ij} = 0$ denotes that the valve is close and $\mu_{ij} = 1$ indicates that the valve is open).

Notice that $q_{i0}$ ($i$ = 1, 2, 3) represents the outflow rate with:

$$q_{i0} = s_{i0} \mu_{i0} \sqrt{2gh_i} \quad (20)$$
where $s_{i0} = 6.36 \times 10^{-5} \text{ m}^2$ is the section of corresponding valve and $\mu_{i0} \in \{0, 1\}$ (0 for close and 1 for open).

We aim to control the water levels of tanks 1 and 2 by adjusting the flow rate of pump 1 and 2.

B. Modeling and identification

The process, although non-differentiable, may be regarded as a hybrid system. Indeed, it has many possible state locations ($h_1 > h_3$ or $h_1 < h_3$ or $h_2 > h_3$ or $h_2 < h_3$). Furthermore, the dynamic of the system depends on the state of the valves ($\mu_{i0}$). In this simulation, we assume that the state of valves $\mu_{i0}$ ($i = 1, 2, 3$) can be modified at any time but the other valves are always open.

A general inspection reveals that a linear second order system is a good representation for small variations of the inputs and of the outputs. This means that the global nonlinear model, after linearization, should provide a representation allows to model this multivariable nonlinear system with a small modeling error.

C. Results

The simulation experiments have been performed in order to emphasize the robustness property of the proposed control scheme opposite the uncertain dynamic behavior of the process. Hence, during the simulations the valves $\mu_{i0}$ have modified as mentioned in table (II).

Figure 3 plots the evolution of water levels ($h_1$ and $h_2$) and the flow rates ($q_1$ and $q_2$). As shown in this simulation, the robust nonlinear multivariable predictive control exhibit good performances. Indeed, although the change of the states of valves which accompany at iterations 151, 301 and 701, the output signals arrive to reach the desired setpoints.

In the second simulation, we aim to test the robustness of the developed controller opposite the tracking capability. From figure 4, we note that the water levels of tank 1

![Figure 1. Three tanks system.](image-url)
and tank 2 flow the desired reference despite the change of the dynamic behavior of the process.

In these simulations, the average time required to compute the control inputs \( q_1 \) and \( q_2 \) for each sample time is 0.11s. Therefore, we can conclude that the proposed optimization method, generalized geometric programming, presents an alternative to control fast systems.

**IV. CONCLUSIONS**

This paper has proposed the robust nonlinear multivariable predictive control based on multivariable NARMA model with structured uncertainties. This kind of uncertainty, we allow to deal with uncertain dynamic behavior of the system. The control law is formulated as non-convex min-max problem. An efficient optimization technique is presented to overcome this problem. The efficiency of the proposed algorithm is tested on benchmark functions and compared with LMI and genetic algorithms optimizers. The obtained results show the superiority of the proposed algorithm.

The robustness of the proposed controller has been tested on three tanks system. The obtained simulation results have showed that the developed controller can deal with the uncertain physical behavior of the process.

**APPENDIX**

**LIST OF BENCHMARK FUNCTIONS [31]**

**Price (2 variables):**

\[
\begin{align*}
    \text{Price}(x) &= (2x_1^3x_2 - x_2^3)^2 + (6x_1 - x_2^3 + x_2)^2 \\
    \text{s.t.} \quad &-10 \leq x_i \leq 10 \\
    \text{3 global minimums:} \quad &x^* = (0, 0); x^* = (2, 4); x^* = (1.464352, -2.506012); \text{Price}(x^*) = 0
\end{align*}
\]

**Colville4 (4 variables):**

\[
\begin{align*}
    \text{Colville}(x) &= 100(x_2 - x_3^2)^2 + (1 - x_1)^2 + 90(x_4 - x_3^2)^2 + (1 - x_3)^2 + 10.1((x_2 - 1)^2 + (x_4 - 1)^2) + 19.8(x_2 - 1)(x_4 - 1) \\
    \text{s.t.} \quad &-10 \leq x_i \leq 10 \\
    \text{1 global minimum:} \quad &(x)^* = (1, 1, 1, 1); \text{Colville}(x^*) = 0
\end{align*}
\]

**Booth (2 variables):**

\[
\begin{align*}
    \text{Booth}(x) &= (x_1 + 2x_2 - 7)^2 + (2x_1 + x_2 - 5)^2 \\
    \text{s.t.} \quad &-10 \leq x_i \leq 10 \\
    \text{1 global minimum:} \quad &(x_1, x_2)^* = (1, 3); \text{Booth}((x_1, x_2)^*) = 0
\end{align*}
\]

**Schwefel 3.2 (3 variables):**

\[
\begin{align*}
    \text{Schwefel}(x) &= (x_1 - x_2^2)^2 + (1 - x_2)^2 + (x_1 - x_2^2)^2 + (1 - x_3)^2 \\
    \text{s.t.} \quad &-10 \leq x_i \leq 10 \\
    \text{1 global minimum:} \quad &(x_1, x_2, x_3)^* = (1, 1, 1); \text{Schwefel}((x_1, x_2, x_3)^*) = 0
\end{align*}
\]

**Extended Rosenbrok (7 variables):**

\[
\begin{align*}
    \text{Rosenbrok}(x) &= \sum_{i=2}^{7} 100(x_i - x_{i-1})^2 + (1 - x_{i-1})^2 \\
    \text{s.t.} \quad &-10 \leq x_i \leq 10 \\
    \text{1 global minimum:} \quad &(x_1, ..., x_7)^* = (1, ..., 1); \text{Rosenbrok}((x_1, ..., x_7)^*) = 0
\end{align*}
\]

**REFERENCES**


