Multivariable Generalized Predictive Control Using an Improved Particle Swarm Optimization Algorithm

Moussa Sedraoui and Samir Abdelmalek
Laboratoire (PI : MIS) Problèmes Inverses: Modélisation, Information et Systèmes,
Université 08Mai 1945, Guelma, Algérie
E-mail: msedraoui@gmail.com

Sofiane Gherbi
Laboratoire d'Automatique de Skikda (LAS),
Université 20 Aout1955, Skikda, Algérie
E-mail: sgherbi@gmail.com

Keywords: improved particle swarm optimization, multivariable generalized predictive control, feasible region

Received: November 16, 2010

In this paper, an improvement of the particle swarm optimization (PSO) algorithm is proposed. The aim of this algorithm is to iteratively resolve the cost problem of the Multivariable Generalized Predictive Control (MGPC) method under multiple constraints previously reduced. An ill-conditioned chemical process modelled by an uncertain Multi-Input & Multi-Output (MIMO) plant is controlled in order to verify the validity and the effectiveness of the proposed algorithm. The performances obtained are compared with those given by the MGPC method using the standard PSO algorithm. The simulation results shows that the proposed algorithm outperforms standard PSO algorithm in terms of performance and robustness.

Povzetek: Predstavljena je metoda optimiranja z roji za nadzor s splošnim napovedovanjem in več spremenljivkami.

1 Introduction

Multivariable generalized predictive control (Morari & Lee, 1999) is a very powerful method. It has been the subject of many researches during the last few years and it was applied successfully in industry, particularly in chemical processes. It is based on MIMO predictive model [1], [2] where the expected behaviour of the system can be predicted in the extended time horizon. The MGPC law is obtained by minimizing linear or non-linear criterion (Magni, 1999, Duwaish, 2000). This criterion is composed by the sum of the square prediction errors between the predicted and desired outputs, the weighted sum of the square change-controls (control-increments) and others [3]. The constraints inclusion (as mathematical inequalities type) distinguishes most clearly MGPC from other process control paradigms as suggested in (Richalet, 1993, Qin 1997, Rawlings, 1999). These constraints are imposed in order to ensure a better stability and performance robustness (Al Hamouz and Duwaish, 2000, Imsland, 2005). The MGPC method formulates the constraint optimization problem at every step time for solving the optimal control move vector [4]. At the next sampling time, a new process measurement is received, the process is updated, and a new constraint optimization problem is solved for the next control move vector. An efficient randomized constraint optimization algorithm is suggested to the MGPC method named by PSO algorithm (Rizvi & al, 2010, Yousuf & al, 2009, Al Duwaish, 2010). This algorithm explores the search space using a population of particles, each one with a particle or an agent, starting from a random velocity vector and a random position vector. Each particle in the swarm represents a candidate solution (treated as a point) in an *n*-dimensional space for the constraint optimization problem, which adjusts its own "flying" according to the other particles [5]. The PSO algorithm can resolve successively various constraint optimization problems, such as linear or non-linear, convex or non-convex problems. Unfortunately, it cannot provide satisfactory results when the MGPC method is applied to poorly modelled processes [6] operating in ill-defined environments. This is, as often, the case when the plant has different gains for the operational range designed by user's trial-and- error. In addition, the PSO algorithm's convergence cannot satisfy multiple time domain specifications if the process (to be controlled) is constrained by a high number of hard constraints (Leandro dos Santos Coelho & al, 2009). Several heuristic algorithms have been developed in recent years to improve the performance and set up the parameters of the PSO algorithm [7].This paper investigates the analysis of the above mentioned problems. Two main contributions are proposed in this paper in order to improve the performances of the MGPC method. The first one consists to reduce (if possible) the imposed inequality constraints which are reformulated as boundary constraints. The second one is to resolve the bounds constraints optimization problem by the improved PSO algorithm.

2 Unconstrained MGPC Method

All the considered matrices are in discrete time domain.

A *CARIMA* (Controller Auto Regression Integrated Moving Average) model for an m inputs and m outputs multivariable process can be expressed by [8]:

$$A(q^{-1})\Delta y(t) := B(q^{-1})\Delta u(t-1) + C(q^{-1})\zeta(t) \qquad (1)$$

Where

$y(t) \in \Re^{m\times 1} := [y_1(t) \quad y_2(t) \quad \dots \quad y_m(t)]^T$

$u(t) \in \Re^{m\times 1} := [u_1(t) \quad u_2(t) \quad \dots \quad u_m(t)]^T$

$A(q^{-1})$, $B(q^{-1})$ and $C(q^{-1})$ are $m \times m$ monic polynomial matrices. Set $C(q^{-1})$ equal to the unity diagonal matrix. $\zeta(t)$ is an uncorrelated random process and $\Delta(q^{-1}) = 1 - q^{-1}$, this form enables to introduce an integrator in the control law. Without lost of generality one can suppose A as diagonal polynomial matrix.

$y_i(t) \in \Re$, $u_i(t) \in \Re$ denotes respectively, the process output and the control input of the channel number 'i'. q^{-1} denotes the backward shift operator. The role of $\Delta(q^{-1})$ is to ensure an integral action of controller in order to cancel the effect of the step varying output in the channel 'i'.

As in all receding horizon predictive control strategies, the control law provides that, for each channel 'i', the control-increment $\Delta u_i(t)$ which minimizes the following unconstraint cost problem of the MGPC method [8]:

$$J := \sum_{i=1}^{m} \left(\sum_{j=1}^{N_2^i} [\hat{y}_i(t+j/t) - w_i(t+j)]^2 + \lambda_i \sum_{j=1}^{N_u^i} [\Delta u_i(t+j-1)]^2 \right) \qquad (2)$$

Where

$\hat{y}_i(t+j) \in \Re$ is an optimum j-step-ahead prediction of the system output vector on data up to time t, therefore, the expected value of the output vector at time t if the past input vector, the output vector, and the future control sequence are known. Noting that $\hat{y}_i(t+j)$ is depending to the control-increment Δu_i from resolving two Diophantine equations (more details are available in the reference [9]).

$w_i(t) \in \Re$ is the future set-point or the reference sequence for the output $y_i(t)$.

N_2^i, N_u^i (with respect: $N_u^i \le N_2^i$) denotes respectively, the maximum output prediction horizon (assumed equal to $N_2 \in \Re^+$) and the maximum control prediction (assumed equal to $N_u \in \Re^+$) for each channel 'i'. $\lambda_i \in \Re^+$ denotes the positive parameter weighting the control input for each channel 'i'.

3 Classification of Constraints and Problem Formulation

In constrained control, a set of inequality constraints may be set as addition of the control objective and the variation limits of certain variables to the given ranges:

$\underline{v}_i \le v_i(t+j) \le \overline{v}_i$, with $i := 1,2,..m$ and $j := N_{s1},..., N_{s2}$.

Where

$v_i(t+j) \in \Re$ is a variable under restriction,

$\underline{v}_i \in \Re$ and $\overline{v}_i \in \Re$ are the lower and the upper boundaries,

N_{s1} and N_{s2} are the lower and the upper constraint horizons respectively.

The two main objectives of constrained predictive control are set-point tracking and prevention / reduction of constraint transgressions. These constraints can be imposed (with respect to the time index) on the control-increment vector, or/and on the control vector as follows:

- Constrained on the control-increment:

$$\underline{\Delta u}_i \le \Delta u_i(t+j) \le \overline{\Delta u}_i \qquad (3)$$

Where $i = 1,2,..m$ and $j = 0,..., N_u - 1$.

- Constrained on the control:

$$\underline{u}_i \le u_i(t+j) \le \overline{u}_i \qquad (4)$$

Where $i = 1,2,..m$ and $j = 0,..., N_u - 1$.

By using:

$$u_i(t+j) := u_i(t-1) + \sum_{k=0}^{j} \Delta u_i(t+k) \qquad (5)$$

The control constraints (4) becomes as follow:

$$\underline{u}_i - u_i(t-1) \le \sum_{k=0}^{j} \Delta u_i(t+k) \le \overline{u}_i - u_i(t-1) \qquad (6)$$

The constraints on the control vector and the rate of control changes, with respect to the batch index, can be easily combined together:

$$A_{inq} \cdot \Delta U \le B_{inq} \qquad (7)$$

Where

$\Delta U_{(m\cdot N_u)\times 1} := \begin{pmatrix} \Delta U(t) \\ \Delta U(t+1) \\ \vdots \\ \Delta U(t+N_u-1) \end{pmatrix}$ denotes the design parameter vector which will be determined later by the PSO algorithm, it contains the future control- increment vector $(\Delta U(t+j))_{m\times 1}$ of each channel as:

$$\Delta U(t+j) := \begin{pmatrix} \Delta u_1(t+j) \\ \Delta u_2(t+j) \\ \vdots \\ \Delta u_m(t+j) \end{pmatrix}_{j=0,1,\cdots,N_u-1}$$

$(A_{inq})_{(4.m.N_u)\times(m.N_u)}$, $(B_{inq})_{(4.m.N_u)\times 1}$ are defined by:

$$A_{inq} := \begin{pmatrix} diag(I_{m\times m}) \\ -diag(I_{m\times m}) \\ tril(I_{m\times m}) \\ -tril(I_{m\times m}) \end{pmatrix}$$

Where

$diag(I_{m\times m}) \in \Re^{(m.N_u)\times(m.N_u)}$ denotes the unity diagonal matrix, and $tril(I_{m\times m}) \in \Re^{(m.N_u)\times(m.N_u)}$ denotes the lower triangular matrix of the unity diagonal matrix $(I_{m\times m})$.

$$B_{inq} := \begin{pmatrix} [\overline{\Delta u_i}]_{(m\cdot N_u)\times 1} \\ -[\underline{\Delta u_i}]_{(m\cdot N_u)\times 1} \\ [\overline{u}_i - u_i(t-1)]_{(m\cdot N_u)\times 1} \\ -[\underline{u}_i - u_i(t-1)]_{(m\cdot N_u)\times 1} \end{pmatrix}$$

The cost index (2) can be expressed in matrix form as:

$$J(\Delta U, t) := \Delta U^T \cdot Q_2 \cdot \Delta U + Q_1^T \cdot \Delta U + Q_0 \quad (8)$$

Where $Q_2 := G^T \cdot G + \Lambda$, $Q_1^T := 2(\Gamma - W)^T G$ and $Q_0 := (\Gamma - W)^T \cdot (\Gamma - W)$

$\Lambda := \lambda \cdot I_{(m\cdot N_u)\times(m\cdot N_u)}$ is diagonal matrix weighting the control-increment vector, and $W_{(m\cdot N_2)\times 1}$ is the projected set-point vector.

$G_{(m\cdot N_2)\times(m\cdot N_u)}$, $\Gamma_{(m\cdot N_2)\times 1}$ are the polynomial matrices which are determined by the recursively resolution of the two Diophantine equations [9].

The cost index (8) and the inequality constraints (7) formulate the following constraint optimization problem as:

$$\begin{cases} \min\limits_{\Delta U} J(\Delta U, t) := \Delta U^T \cdot Q_2 \cdot \Delta U + Q_1^T \cdot \Delta U + Q_0 \\ \text{s.t}: \quad A_{inq} \cdot \Delta U \le B_{inq} \end{cases} \quad (9)$$

Now, an optimal control vector is given by the PSO algorithm. This algorithm should minimize the objective function (8) under $4 \times m \times N_u$ inequality constraints. The computational requirements of the PSO algorithm depend heavily on the number and the type of the constraints to be satisfied. An efficient off-line constraint PSO algorithm, suggested by Ichirio & al, 2009, can resolve this problem [10]. Unfortunately, this algorithm is difficult to extend to the MGPC method for two reasons: the first one is due to a large dimension of the inequality constraints which needs excessive computation time. The second one is due to a real-time output feedback implementation of the MGPC method which requires a minimum consuming time. To resolve these above problems, the inequality constraints should be reduced, for each step time, and reformulated as bounds constraints type. Only those constraints (which limit the feasible region) must be taken into account. The efficiency of the PSO algorithm can be increased if the superfluous constraints (which do not limit the feasible region) should be eliminated [11]. In this paper we propose a systematic method that determines the minimum set vector of limiting constraints. The lower and the upper bounds of the feasible region are given as below:

For each channel $i := 1, \cdots, m$, the control-increment $\Delta u_i(t)$ is simultaneously constrained by:

1- For the control prediction horizon $j = 0$:

$$\begin{cases} \underline{\Delta u}_i \le \Delta u_i(t) \le \overline{\Delta u_i} \\ \underline{u}_i - u(t-1) \le \Delta u_i(t) \le \overline{u_i} - u(t-1) \end{cases} \quad (10)$$

It is easy to see that the new lower and upper bounds are determined by:

$$\underline{v}_i(t) \le \Delta u_i(t) \le \overline{v}_i(t) \quad (11)$$

Where

$$\underline{v}_i(t) := \max\{\underline{\Delta u}_i \quad \underline{u}_i - u(t-1)\} \quad (12)$$

$$\overline{v_i}(t) := \min\{\overline{\Delta u}_i \quad \overline{u_i} - u(t-1)\} \quad (13)$$

2- For the control prediction horizon $j = 1$:

$$\begin{cases} \underline{\Delta u}_i \le \Delta u_i(t+1) \le \overline{\Delta u}_i \\ \underline{u}_i - u(t-1) \le \Delta u_i(t+1) + \Delta u_i(t) \le \overline{u}_i - u(t-1) \end{cases} \quad (14)$$

The new lower and upper bounds are determined for $j = 1$ by:

$$\underline{v}_i(t+1) \le \Delta u_i(t+1) \le \overline{v}_i(t+1) \quad (15)$$

Where

$$\underline{v}_i(t+1) := \max\{\underline{\Delta u}_i \quad \{\underline{u}_i - u(t-1)\} - \underline{v}_i(t)\} \quad (16)$$

$$\overline{v_i}(t+1) := \min\{\overline{\Delta u}_i \quad \{\overline{u}_i - u(t-1)\} - \overline{v}_i(t)\} \quad (17)$$

This procedure is repeated until the control prediction horizon $j = N_u - 1$. Therefore the control-increment $\Delta u_i(t + N_u - 1)$ is constrained by the new bounds:

$$\underline{v}_i(t + N_u - 1) := \max\{\underline{\Delta u}_i \quad \{\underline{u}_i - u(t-1)\} - \sum_{k=0}^{N_u-2} \underline{v}_i(t+k)\} \quad (18)$$

$$\overline{v}_i(t + N_u - 1) := \min\{\overline{\Delta u}_i \quad \{\overline{u_i} - u(t-1)\} - \sum_{k=0}^{N_u-2} \overline{v}_i(t+k)\} \quad (19)$$

Then, for each step time value $t_0, t_1, \cdots, t_{\max}$, the feasible region $D(i,j,t) := (\underline{v}_i(t+j) \quad \overline{v}_i(t+j))_{\substack{j=0,\cdots,N_u-1 \\ i=1,\cdots,m}}$ can be determined by the following proposed algorithm:

3.1 Reduced constraints algorithm

For each point time $t_0, t_1, \cdots, t_{\max}$, the feasible region is determined by the followings steps:

[Step 1]: *Set the first counter* $i \leftarrow 1$ *which denotes the number of channels, and go to the next step.*

[Step 2]: *Set the second counter* $j \leftarrow 0$ *which denotes the control horizon prediction, and go to the next step.*

[Step 3]: *Set the parameters* $h_{\max} \leftarrow 0$, $h_{\min} \leftarrow 0$ *and go to the next step.*

[Step 4]: *Build the followings ranges:*

$bound\_\max_i := \{\overline{\Delta u}_i \quad \{\overline{u}_i - u_i(t-1)\} - h_{\max}\}$.

$bound\_\min_i := \{\Delta\underline{u}_i \quad \{\underline{u}_i - u_i(t-1)\} - h_{\min}\}$.

[Step 5]: *Calculate the new upper and the new lower bounds which limit the control- increment $\Delta u_i(t+j)$ by:*

$\overline{v_i}(t+j) := \min\{bound\_\max_i\}$

$\underline{v_i}(t+j) := \max\{bound\_\min_i\}$

From these above bounds, the feasible region is determined as follow:

$D(i,j,t) := (\underline{v_i}(t+j) \quad \overline{v_i}(t+j))$

[Step 6]: *Update the parameters: $h_{\min}$, $h_{\max}$ as follows*

$h_{\max} \leftarrow h_{\max} + \overline{v_i}(t+j)$,

$h_{\min} \leftarrow h_{\min} + \underline{v_i}(t+j)$, and g*o to the next step*

[Step 7]: *Update the second counter $j \leftarrow j+1$ and go back to the step 4 if $j \prec N_u - 1$. Otherwise, go to the next step.*

[Step 8]: *Update the first counter $i \leftarrow i+1$ and stop algorithm if $i := m$. Otherwise go back to the step 2.*

From this above algorithm, the constraint optimization problem (9) under inequality constraints is reformulated as the bounds optimization problem:

$$\begin{cases} \min_{\Delta U} J(\Delta U, t) := \Delta U^T \cdot Q_2 \cdot \Delta U + Q_1^T \cdot \Delta U + Q_0 \\ \text{s.t}: \quad \underline{\Delta U} \le \Delta U \le \overline{\Delta U} \end{cases} \tag{20}$$

Where $\underline{\Delta U}$, $\overline{\Delta U}$ denotes respectively, the new lower and the new upper bounds vector which limit the feasible region $D_{(m.N_u)\times 2} := (\underline{\Delta U}, \overline{\Delta U})$, with:

$$\left(\underline{\Delta U}\right)_{(m.N_u)\times 1} := \left(\underline{v}_i(t) \quad \underline{v}_i(t+1) \quad \cdots \quad \underline{v}_i(t+N_u-1)\right)^T_{i=1,2\cdots m}$$

$$\left(\overline{\Delta U}\right)_{(m.N_u)\times 1} := \left(\overline{v_i}(t) \quad \overline{v_i}(t+1) \quad \cdots \quad \overline{v_i}(t+N_u-1)\right)^T_{i=1,2\cdots m}$$

From (20), it is easy to see that the inequality constraints number is reduced to $m \times N_u$ constraints at each step time. This dramatic reduction has a capital importance for the success of the PSO algorithm.

Now, we are able to find the optimal control of the MGPC law. The new constraint optimization problem (20) should be resolved for each step time $t := t_0, t_1, \cdots, t_{\max}$, its solution vector ΔU^* denotes the optimal design parameter vector. Only the first m rows of ΔU^* is used to obtain the optimal desired control-increment vector of each channel (' i '). The optimal control vector is obtained by adding the previous control vector to the optimal control-increment vector as follow:

$$u_i(t) := u_i(t-1) + \Delta u_i^*(t) \tag{21}$$

4 Improved PSO Algorithm [6]

Particle swarm optimization algorithm, introduced first by Kennedy and Eberhart in (1995), is one of the modern heuristic algorithms which belong to the category of *Swarm Intelligence* method (Kennedy, 2001). The PSO algorithm uses a swarm consisting of $N_p \in \aleph$ particles for each control-increment vector $\left(\Delta u_i(t+j)\right)_{\substack{j=0,1,\cdots N_u-1 \\ i=1,2,\cdots m}}$ to get an optimal solution $\Delta u_i^*(t+j)$ which minimizes the optimization problem (20). The position of (i_{th}) particle and its velocity are respectively denoted as [12]:

$$\Delta u_i(t+j) := \left(\Delta u_{i,1}(t+j) \quad \Delta u_{i,2}(t+j) \quad \ldots \quad \Delta u_{i,N_p}(t+j)\right)^T$$

$$\psi_i(t+j) := \left(\psi_{i,1}(t+j) \quad \psi_{i,2}(t+j) \quad \ldots \quad \psi_{i,N_p}(t+j)\right)^T$$

.

Then, the position of the (i_{th}) particle, $\Delta u_i(t+j)$, is based on the following update law:

for $\ell = 1,2,\ldots \ell_{\max}$, which indicates the iteration number [12],

$$\psi_i^{\ell+1} := c_0\psi_i^\ell + c_1 r_{1,i}^\ell \left(H_i^{best,\ell} - \Delta u_i^\ell\right) + c_2 r_{2,i}^\ell \left(h_{swarm}^{best,\ell} - \Delta u_i^\ell\right) \tag{22}$$

$$\Delta u_i^{\ell+1} := \Delta u_i^\ell + \psi_i^{\ell+1} \tag{23}$$

Where c_1 and c_2 are respectively, the cognitive (individual) and the social (group) learning rates and are both positive constants. The value of cognitive parameter c_1 signifies a particle's attraction to a local best position based on its past experience. The value of social parameter c_2 determines the swarm's attraction towards a global best position.

$c_0 \in \Re^+$ is the inertia weight factor whose value decreases linearly with the iteration number (Shi & Eberhart, 1999) as [13]:

$$c_0 := \theta_{\max} - \left(\frac{\theta_{\max} - \theta_{\min}}{\ell_{\max}}\right) \cdot \ell \tag{24}$$

Where $\theta_{\max}$ and $\theta_{\min}$ are the initial and the final values of the inertia weight, respectively. The values of $\theta_{\max} = 0.9$ and $\theta_{\min} = 0.4$ are commonly used [13].

The random numbers $r_{1,i}^\ell$ and $r_{2,i}^\ell$ are uniformly distributed in $[0,1]$.

$H_i^{best,\ell}$ and $h_{swarm}^{best,\ell}$ denotes respectively, the best previously obtained position of the (i_{th}) particle (the position giving the lower value of the objective criterion) and the best position in the entire swarm at the current iteration ℓ [10]:

$$H_i^{best,\ell} := \arg\min_{\Delta u_i^r}\{J(\Delta u_i^r), 0 \le r \le \ell\} \tag{25}$$

$$h_{swarm}^{best,\ell} := \arg\min_{\Delta u_i^\ell}\{J(\Delta u_i^\ell), \forall i\} \tag{26}$$

From equation (23), some current position of (i_{th}) particle (in each dimension) can exceed the corresponding lower bound or upper bound of the feasible region. Consequently, the given optimal control vector of the MGPC method cannot satisfy some specifications and also some constraints are non-satisfactoriness' in some range time. To avoid, we should improve the convergence of PSO algorithm by adjusting only the corrupted position of (i_{th}) particle with the region around the current established solution, if it is too smaller than the corresponding lower bound, its value $\underline{v}_i$ should be replaced. If it is too higher than the corresponding upper bound, then its value is replaced by $\overline{v_i}$. The proposed modification can be formulated as follows:

Let consider:

$\Delta u_i^q(t+j)$: The corrupted position of (i_{th}) particle given at current iteration $\ell := q$.

$\underline{v}_i(t+j)$, $\overline{v_i}(t+j)$: The lower bound and upper bound which are determined by the reduced constraints algorithm. So that, the above corrupted position can be adjusted by using the following inequalities:

$$\Delta u_i^q(t+j) := \begin{cases} \underline{v}_i(t+j) : if \quad \Delta u_i^q(t+j) \prec \underline{v}_i(t+j) \\ \overline{v_i}(t+j) : if \quad \Delta u_i^q(t+j) \succ \overline{v_i}(t+j) \end{cases} \quad (27)$$

Consequently, from the equation (23), the current velocity should be limited by the following bounds:

$$\underline{v}_i - \Delta u_i^{q-1} \prec \psi_i^q \prec \overline{v_i} - \Delta u_i^{q-1} \quad (28)$$

Now, the modified current positions with their modified velocity are used to improve the next best position and their velocity vector for the next iteration as follow:

$$\psi_i^{q+1} := c_0\psi_i^q + c_1 r_{1,i}^q \left(H_i^{best,q} - \Delta u_i^q\right) + c_2 r_{2,i}^q \left(h_{swarm}^{best,q} - \Delta u_i^q\right) \quad (29)$$

$$\Delta u_i^{q+1} := \Delta u_i^q + \psi_i^{q+1} \quad (30)$$

The improved PSO algorithm consists of the following steps:

4.1 Proposed algorithm

For each step time $t := t_0, t_1, \cdots, t_{\max}$ the optimal control-increment is determined by the following steps:

[Step 1]: *Determine the lower bound and the upper bound* $\underline{v}_i(t+j)$, $\overline{v_i}(t+j)$ *which are corresponding the design parameter* $\left[\Delta u_i(t+j)\right]_{\substack{i:=1,\cdots,m \\ j:=0,\cdots,N_u-1}}$.

[Step 2]: *Initialize random swarm positions and velocities:*

initialize a population (array) of particles with random positions and velocities (array) from the search domain $D := (\underline{\Delta U}, \overline{\Delta U})$.

Set the counter $\ell \leftarrow 1$ *and go to the next step.*

[Step 3]: *Evaluate the objective criterion (20) and obtain* $H_i^{best,\ell}$, $h_{swarm}^{best,\ell}$ *according to (25) and (26).*

[Step 4]: *Update of a particle's velocity and its position according to (22) and (23).*

[Step 5]: *Check each parameter of the particle's position by the following corresponding lower bound and upper bound* $\underline{v}_i(t+j)$, $\overline{v_i}(t+j)$. *Replace only those exceeding these above bounds.*

[Step 6]: *Update the counter* $\ell \leftarrow \ell + 1$ *and go back to the step 3 if* $\ell \prec \ell_{\max}$. *Otherwise, stop algorithm and take the best position vector as an optimal solution which minimize the constrained optimization problem (20).*

5 Simulation Results and Discussion

In this section, a multivariable generalized predictive control method using a modified particle swarm optimization algorithm is applied to a distillation column which is MIMO plant with two input and output vectors (benchmark problem, see [14]). The two inputs are the reflux and the vapour boil up rate and the outputs are the distillate and the bottom product. The results are compared with those given by the MGPC method using the standard PSO algorithm. The mathematical model is given by [14]:

$$G(s) := \frac{1}{75s+1}\begin{bmatrix} 0.878 & -0.864 \\ 1.082 & -1.096 \end{bmatrix}\begin{bmatrix} K_1 e^{-\tau_1 s} & 0 \\ 0 & K_2 e^{-\tau_2 s} \end{bmatrix} \quad (31)$$

$K_{i(i=1,2)} \in \left[0.8 \quad 1.2\right]$, $\tau_{i(i=1,2)} \in \left[0.0 \quad 1.0\right]$.

Where

τ_i, K_i denotes respectively, the uncertainty temperatures and uncertainty gains of the process.

The time domain specifications are formulated, for the time range $t \in [0,400]$ minutes, as below:

a- For the first set-point reference vector: $w = \begin{pmatrix}1 & 0\end{pmatrix}^T$, the first and the second output channels y_1 and y_2 must satisfy[14]:

(S$_1$): $y_1(t) \geq 0.9$ in more than 30 minutes.

(S$_2$): $y_1(t) \leq 1.1$: the maximum over-shoot corresponding the first output channel cannot exceed 11% for all range time $t \in [0,400]$.

(S$_3$): $0.99 \leq y_1(\infty) \leq 1.01$: the static error value cannot exceed 1% ($\left|y_1(\infty) - w_1(\infty)\right| \leq 1\%$).

(S$_4$): $y_2(t) \leq 0.5$: the maximum over-shoot of the second output channel cannot exceed to 50% for all range time $t \in [0,400]$.

(S$_5$): For $t \to \infty : -0.01 \leq y_2(t) \leq 0.01$: the static error value cannot exceed 1%. From another word:

$\left|y_2(\infty) - w_2(\infty)\right| \leq 1\%$.

(S$_6$): Closed loop stability.

(S$_7$): Control signals should be limited by $\left[-200 \quad +200\right]$.

(S$_8$): Control-increment signals should be limited by $\left[-12 \quad +12\right]$.

For the set-point reference vector: $w = \begin{pmatrix}1 & 0\end{pmatrix}^T$, the sampling time $T_e = 1$ minute is used to determine a *CARIMA* predictive model of the chemical process for two followings parameters cases [14]:

$K_{1,2} = \tau_{1,2} = 1$ and $K_1 = 1.2, K_2 = 0.8, \tau_{1,2} = 1$.

b-The same previous time domain specifications should be satisfied for the second set-point reference vector $w = \begin{pmatrix}0 & 1\end{pmatrix}^T$ corresponding to the low gains direction $K_1 = K_2 = 0.8$ and the same time delay constants $\tau_1 = \tau_2 = 1$.

The MGPC method is tuned by choosing: $(N_2^i, N_u^i, \lambda_i)_{i=1,2} = (8,6,0.01)$ at time range $t := [0,400]$ minutes.

For each step time: $t := t_0, t_1, \cdots, 400$, the feasible region is determined from the following constraints:

$$-200 \le u_i(t+j)_{\substack{j=0,\cdots,5\\ i=1,2}} \le +200 .$$

$$-12 \le \Delta u_i(t+j)_{\substack{j=0,\cdots,5\\ i=1,2}} \le +12 .$$

From the reduced constraints algorithm (see section 3.1), these above inequality constraints are reduced in order to determine the search space D at each step time. The constrained optimization problem is resolved by standard and improved PSO algorithms according to the following parameters:

- Swarm size: $N_p := 24$.
- Maximum iteration: $\ell_{\max} := 100$.
- Cognitive and social learning rates: $c_1 = c_2 := 1$.

For the set-point reference vector: $w = (1 \quad 0)^T$ and the parameter system's: $K_{1,2} = \tau_{1,2} = 1$, the figures 1.1 to 1.3 shows the results given by the MGPC method using the standard PSO algorithm (dashed curves), and the MGPC method using the improved PSO algorithm (line curves). The table1 summaries the results obtained by the two algorithms.

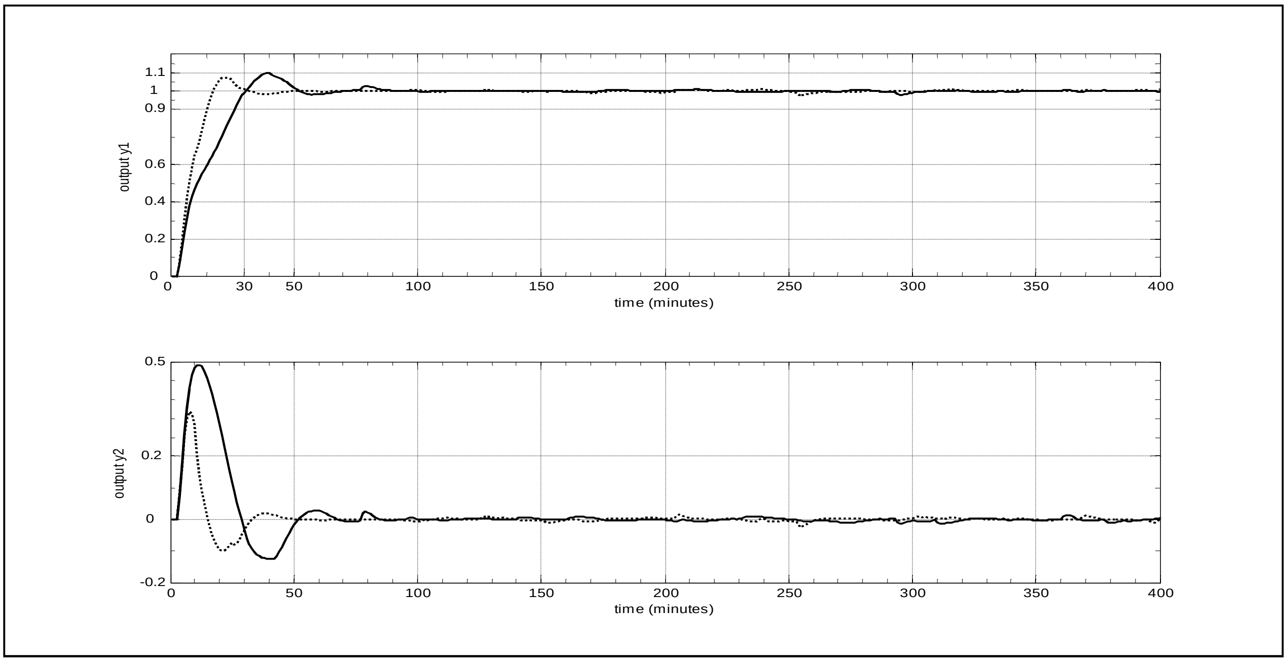

Figure 1.1: Set-point tracking results with standard and improved PSO algorithms for $w = (1 \quad 0)^T$ and $K_{1,2} = \tau_{1,2} = 1$.

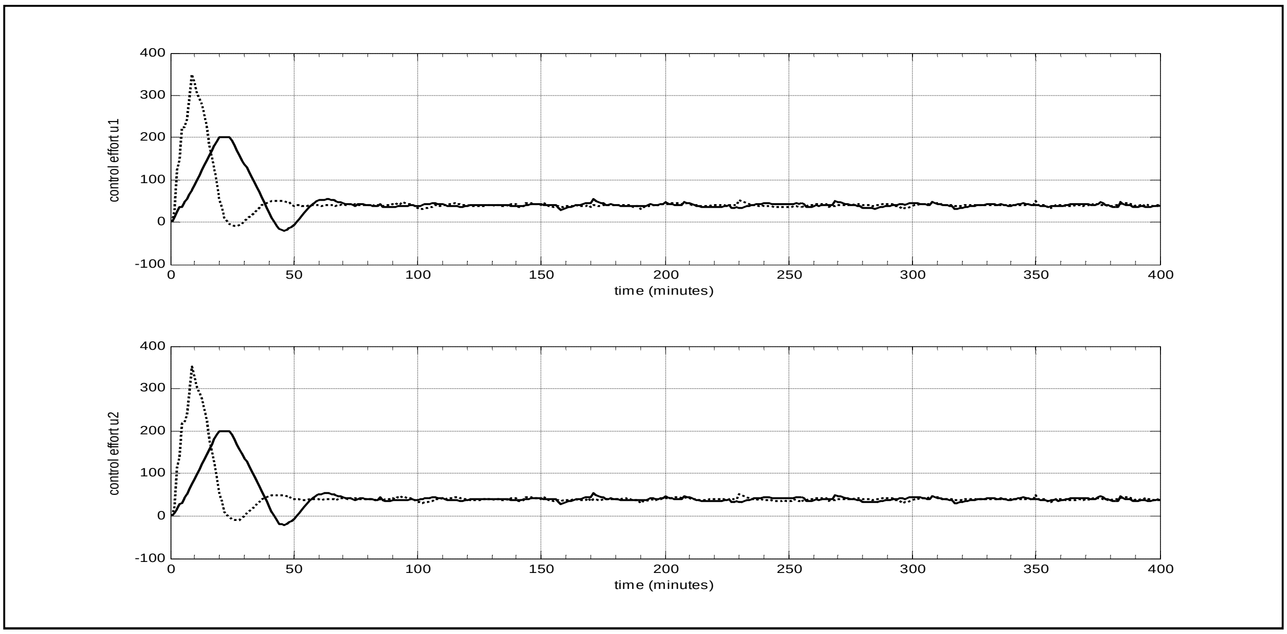

Figure 1.2: Control effort results with standard and improved PSO algorithms for $w = (1 \quad 0)^T$ and $K_{1,2} = \tau_{1,2} = 1$.

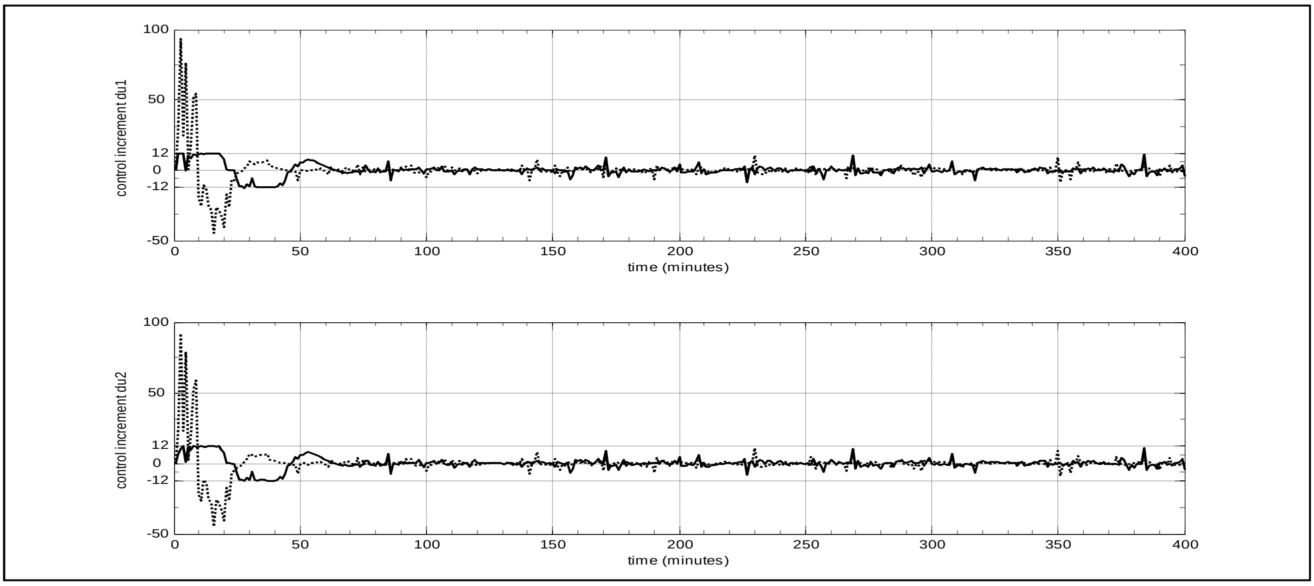

Figure 1.3: Control-increment results with standard and improved PSO algorithms for $w=(1 \quad 0)^T$ and $K_{1,2}=\tau_{1,2}=1$.

Specifications (S_k): / *Algorithms:*	*(S1) $y_1(30)$*	*(S2)*		*(S3) $y_1(400)$*	*(S4)*		*(S5) $y_2(400)$*	*(S6) stable / unstable*	*(S7) Unsatisfactory/ satisfactory constraints*	*(S8) Unsatisfactory/ satisfactory constraints*	*Decision & reasons*
		max(y_1)	time		max(y_2)	time					
Standard PSO	*1.010*	*1.074*	*23*	*1.010*	*0.343*	*8*	*0.006864*	*stable*	***unsatisfactory*** ***$-9.1 \le u_1 \le 349.3$*** ***$-9.9 \le u_2 \le 351.4$***	***unsatisfactory*** ***$-44 \le \Delta u_1 \le 93$*** ***$-44 \le \Delta u_2 \le 91$***	***Rejected*** *algorithm (S7),(S8)*
Improved PSO	*0.990*	*1.096*	*40*	*0.9975*	*0.4846*	*13*	*0.002512*	*stable*	*Satisfactory* *$-20.1 \le u_1 \le 200$* *$-20.3 \le u_2 \le 199$*	*Satisfactory* *$-12 \le \Delta u_1 \le 12$* *$-12 \le \Delta u_2 \le 12$*	*Accepted algorithm*

Table 1: Summary of the results (unsatisfactory performances are in bold) for the nominal model and the set-point reference $w=(1 \quad 0)^T$.

According to the figure 1.1, we can see that, the tracking dynamic of set-point reference vector found by MGPC method based on a standard PSO algorithm is better than the other algorithm but unfortunately, the time domain specifications: (S_7) and (S_8) are not satisfied.

In the figure 1.2, the obtained control signals of the MGPC method based on standard PSO algorithm exceed the constraint ranges at $t:=\{5,6,\cdots 15\}$ minutes such as: $u_{1\max}(9)=349.3$ and $u_{2\max}(9)=351.4$. In addition, the control-increment signals presented in the figure 1.3 also violate the constraint ranges at times: $t:=\{(2\cdots 5),(7\cdots 11),(13\cdots 22)\}$ minutes. Consequently, the performance robustness of this method is very poor in comparison with the MGPC method using the improved PSO algorithm which is capable to satisfy all time domain specifications. These results confirm the usefulness and the robustness of the proposed algorithm.

Figures 2.1, 2.2, 2.3 and table 2 give the results of the MGPC method with the following parametric changes in the process: $(K_1=1.2, K_2=0.8, \tau_{1,2}=1)$ for the set-point reference vector $w=(1 \quad 0)^T$.

According to the figures 2.1 to 2.3, the better results are obtained by the improved PSO algorithm which satisfies all time specifications (S_1 to S_8). These results can be explained by the best stability robustness against the process parametric disturbances. Furthermore, the control and the control-increment signals from the standard PSO algorithm show a dramatic oscillation at transient time region and exceed the constraint ranges. In fact, this algorithm cannot fulfill the three followings time domain specifications: (S_2) with $\max(y_1)=11.124\%$, (S_7) with $u_{1\max}=251$, $u_{2\max}=374$ and(S_8)with $-26.4 \le \Delta u_1 = 70.76$, $-39.7 \le \Delta u_2 = 99.05$, which can be explained by a high sensitivity to the parametric process variations. Thus, from these figures and table 2, we confirm the superiority of the proposed algorithm.

Figures 3.1, 3.2, 3.3 and table 3 give the results of the MGPC method using both algorithms when low gains directions of the process and set-point reference vector change simultaneously as follows: $(K_1=0.8, K_2=0.8, \tau_{1,2}=1)$, $w=(0 \quad 1)^T$.

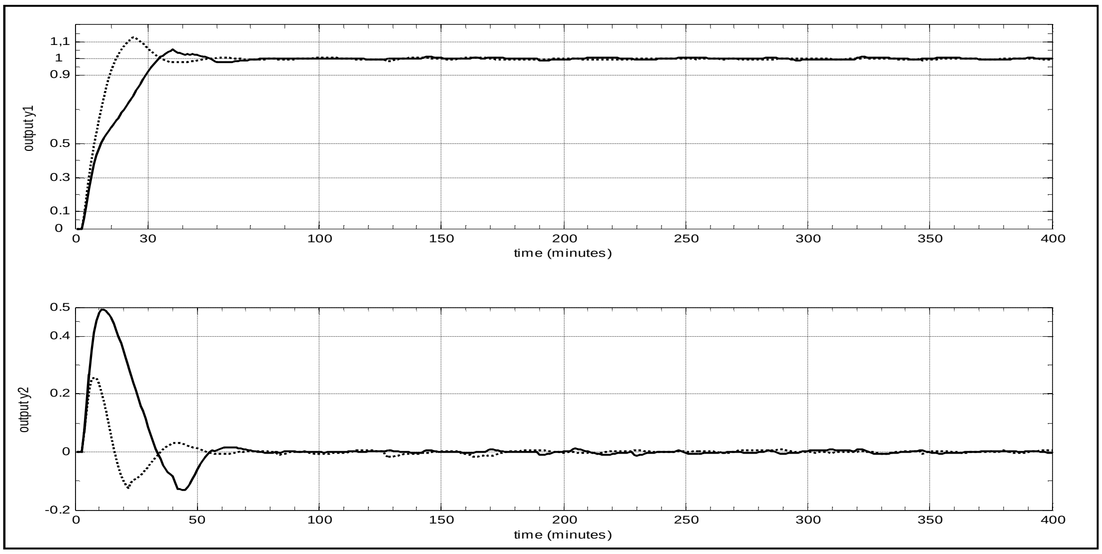

Figure 2.1: Set-point tracking results with standard and improved PSO algorithms for $w=(1 \quad 0)^T$ and $(K_1=1.2, K_2=0.8, \tau_{1,2}=1)$.

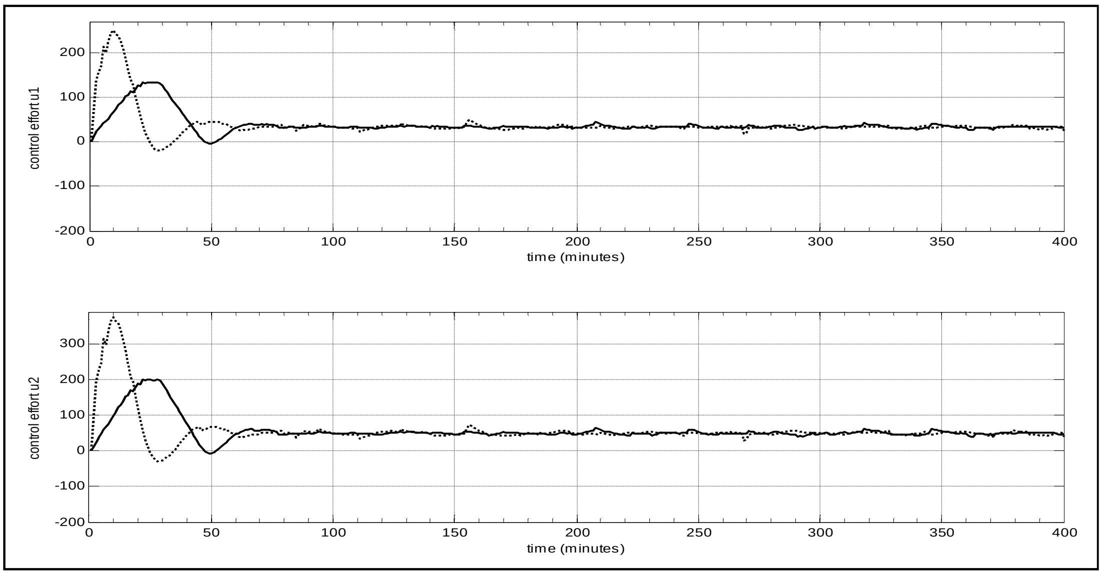

Figure 2.2: Control effort results with standard and improved PSO algorithms for $w=(1 \quad 0)^T$ and $(K_1=1.2, K_2=0.8, \tau_{1,2}=1)$.

These above figures clearly show the performance superiority of the proposed PSO algorithm over standard PSO. For this case, the time domain specifications: S_2, S_7 and S_8 are satisfied with the proposed PSO algorithm, while the same specifications are not satisfactoriness with standard PSO. In addition, the obtained outputs by the standard PSO algorithm converge to the set-point references but unfortunately, two other specifications cannot be satisfied at time $t = 208$ minutes which are:

(S_3): $|y_2(208)-w_2(208)|=5\%$.

(S_5): $|y_1(208)-w_1(208)|=3\%$.

6 Conclusion

In this study, we proposed an improvement of the PSO algorithm, it has been introduced and applied to solve the constrained MGPC problem. In order to find a feasible region, the constraints on the controls and their increments have been previously reduced at each step time, the obtained convergences by improved PSO algorithm are well improved in comparison with the standard PSO algorithm. The efficient of the proposed algorithm is clearly shown and the performances robustness and the stability robustness are guaranteed with little still sensitivity to a set-point references changes and parametric model uncertainties. The results of the proposed algorithm justifies its efficiency and presents quite promising results and can be a subject of an interesting investigations.

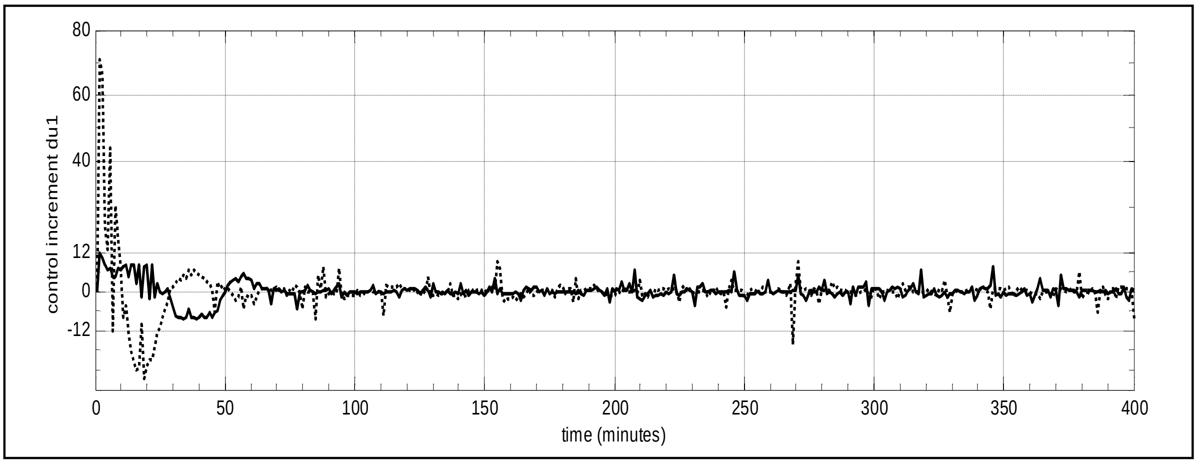

Figure 2.3: Control-increment results with standard and improved PSO algorithms for $w=(1 \quad 0)^T$ and $(K_1=1.2, K_2=0.8, \tau_{1,2}=1)$.

Specifications (S_k):	*(S1) $y_1(30)$*	*(S2)*		*(S3) $y_1(400)$*	*(S4)*		*(S5) $y_2(400)$*	*(S6) stable / unstable*	*(S7) Satisfactory/ unsatisfactory constraints*	*(S8) Satisfactory/ unsatisfactory constraints*	*Decision for reasons (S_k)*
Algorithms:		max(y_1)	time		max(y_2)	time					
Standard PSO	*1.061*	***1.1124***	*24*	*0.9953*	*0.2562*	*8*	*0.006142*	stable	***unsatisfactory*** *$-19.2 \le u_1 \le 251$* *$-29.4 \le u_2 \le 374$*	***unsatisfactory*** ***$-26.4 \le \Delta u_1 \le 70.76$*** ***$-39.7 \le \Delta u_2 \le 99.05$***	***Rejected algorithm*** *(S_2),(S_7),(S_8)*
Improved PSO	*0.920*	*1.054*	*40*	*0.9953*	*0.493*	*12*	*0.006142*	*stable*	*Satisfactory* *$-3.5 \le u_1 \le 133.5$* *$-6.4 \le u_2 \le 199$*	*Satisfactory* *$-8.38 \le \Delta u_1 \le 12$* *$-12 \le \Delta u_2 \le 12$*	*Accepted algorithm*

Table 2: Summary of the results (unsatisfactory performances are in bold) for the uncertainty model $(K_1=1.2, K_2=0.8, \tau_{1,2}=1)$ and the set-point reference $w=(1 \quad 0)^T$.

References

[1] D.W Clarke, C. Mohtadi & P.S Tuffs, Generalized Predictive Control –Part I : The basic algorithm – Part II: extensions and interpretation. *Automatica*, 23(2), 1987, 137-160.

[2] D.W Clarke & C. Mohtadi, Properties of generalized predictive control, *Automatica*, 25(6), 1989, 859-875.

[3] P. Codron & P. Boucher, Improvement for multivariable generalized predictive control with multiple reference models. *Proc. 12th IASTED International Conf. Modeling. Identification. And Control, Innsbrûck,* Austria, Feb.1993.

[4] T.S. Chang & D.E. Seborg, A linear approach programming for multivariable feedback control with inequality constraints, *International journal of control* 37,1983, 583-597.

[5] M. Zribi, M. Al-Rashed & M. Alrifai. Adaptive decentralized load frequency control of multi-area power systems. *Electrical Power and Energy Systems,* 27:575583, 2005.

[6] J.P. Coelho, P.B.M. Oliveira & J.B. Cunha. Greenhouse air temperature control using the particle swarm optimization algorithm. *15th Triennial world congress.* Barcelona, Spain. IFAC, 2002.

[7] L.D.S. Coelho & al. Model-free adaptive control optimization using a chaotic particle swarm approach. *Chaos, solutions and fractals.* Elsevier. 41, 2009, pp2001-2009.

[8] P. Codron & P. Boucher. Multivariable generalized predictive control with multiple reference models: A new choice for the reference model, *ECC'93. European Control Conf.*, June 28- July1993.

[9] P. Zelinka, B.R. Ilkiv & A.G. Kuznetsov. Experimental verification of stabilizing predictive control. *Control engineering practice7.* 1999, pp 601-610.

[10] I. Maruta, T. Hyoung & T. Sugie: Fixed-Structure H_∞ Controller: A Meta-Heuristic Approach Using Simple Constrained Particle Swarm Optimization. pp:553-559.Automatica 45(2009).

[11] E.F. Camacho. Constrained Generalized Predictive Control, *IEEE Transaction on Automatic Control*, AC-38 (2), 1993, 327-332.

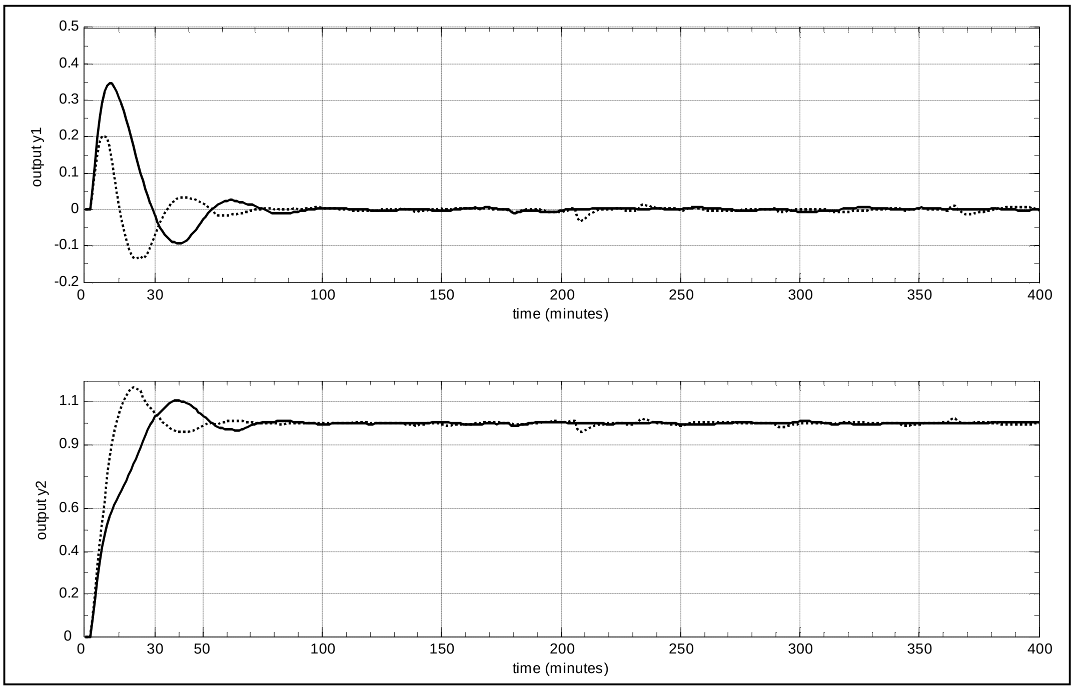

Figure 3.1: Set-point tracking results with standard and improved PSO algorithms for $w = \begin{pmatrix} 0 & 1 \end{pmatrix}^T$ and $\left(K_1 = 0.8, K_2 = 0.8, \tau_{1,2} = 1\right)$.

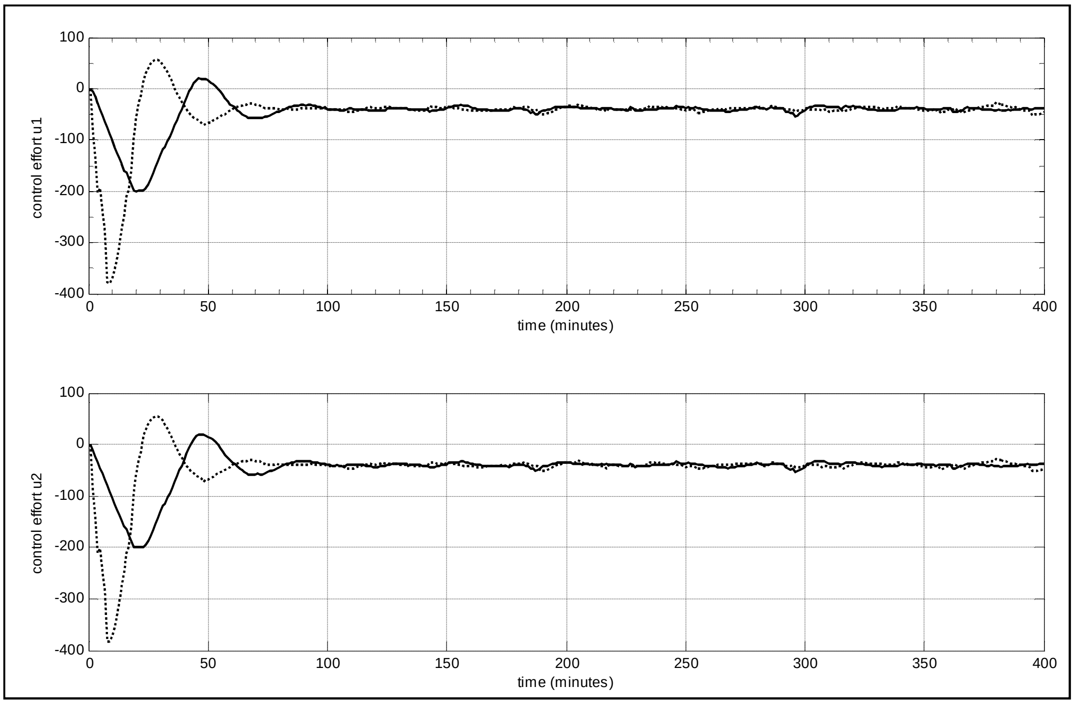

Figure 3.2: Control effort results with standard and improved PSO algorithms for $w = \begin{pmatrix} 0 & 1 \end{pmatrix}^T$ and $\left(K_1 = 0.8, K_2 = 0.8, \tau_{1,2} = 1\right)$.

[12] H.N. Al-Duwaish, S.Z. Rizvi & al. PSO based Hammerstein modeling and predictive control of non linear multivariable boiler. *Control Engineering practice,* pp.1-12, 2010.

[13] S. R. Singiresu: Engineering Optimization: Theory and Practice, Fourth Edition, by John Wiley & Sons, Inc. Hoboken, New Jersey. pp:708-714, & pp:761-768, 2009.

[14] D.J.N Limebeer. The specification and purpose of a controller design case study, *IEEE Conf. Decision Control, Brighton*, U.K, 1991, 1579-1580.

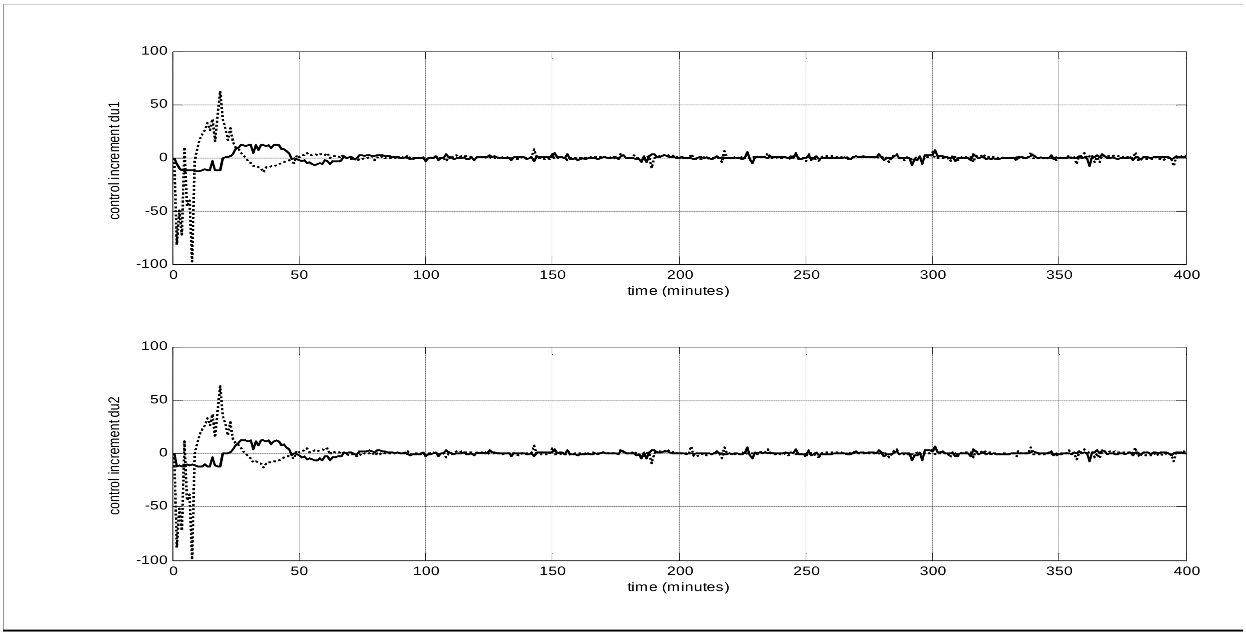

Figure 3.3: Control-increment results with standard and improved PSO algorithms for $w = (0 \quad 1)^T$ and $(K_1 = 0.8, K_2 = 0.8, \tau_{1,2} = 1)$.

Specifications (S_k):	***(S1) $y_2(30)$***	***(S2)***		***(S3) $y_1(400)$***	***(S4)***		***(S5) $y_2(400)$***	***(S6) stable / unstable***	***(S7) Satisfactory/ unsatisfactory constraints***	***(S8) Satisfactory/ unsatisfactory constraints***	***Decision for reasons (S_k)***
Algorithms:		max(y_2)	time		max(y_1)	time					
Standard PSO	*1.046*	***1.167***	*21*	*0.9985*	*0.202*	*8*	*-0.003*	stable	***unsatisfactory*** ***$-379.7 \le u_1 \le 56$*** ***$-383 \le u_2 \le 55.3$***	***unsatisfactory*** ***$-97.7 \le \Delta u_1 \le 62.4$*** ***$-99.2 \le \Delta u_2 \le 62.5$***	***Rejected algorithm*** *(S_2),(S_7),(S_8)*
Improved PSO	*1.032*	*1.10*	*40*	*1.005*	*0.345*	*12*	*0.0009*	*stable*	*Satisfactory* *$-199 \le u_1 \le 20.3$* *$-200 \le u_2 \le 19.9$*	*Satisfactory* *$-12 \le \Delta u_1 \le 12$* *$-12 \le \Delta u_2 \le 12$*	*Accepted algorithm*

Table 3: Summary of the results (unsatisfactory performances are in bold) for the uncertainty model $(K_1 = 0.8, K_2 = 0.8, \tau_{1,2} = 1)$and the change set-point reference $w = (0 \quad 1)^T$.