Automatic Tuning of Model Predictive Control Using Particle Swarm Optimization

Ryohei Suzuki[1], Fukiko Kawai[2], Hideyuki Ito[2], Chikashi Nakazawa[2], Yoshikazu Fukuyama[2], Eitaro Aiyoshi[1]
1 Keio University, 3-14-1, Hiyoshi, Kohoku-ku, Yokohama, 223-8522, Japan
2 Fuji Electric Advanced Technology Co., Ltd., 1, Fujimachi, Hino, Tokyo 191-8502, Japan
(kawai-fukiko, ito-hideyuki, nakazawa-chikashi, fukuyama-yoshikazu)@fujielectric.co.jp

Abstract – This paper presents an automatic tuning method of model predictive control (MPC) using particle swarm optimization (PSO). Although conventional PID is difficult to treat constraints and future plant dynamics, MPC can treat this issues and practical control can be realized in various industrial problems. One of the challenges in MPC is how control parameters can be tuned for various target plants and usage of PSO for automatic tuning is one of the solutions. The numerical results show the effectiveness of the proposed PSO-based automatic tuning method.

I. INTRODUCTION

Various control methods have been utilized in industries. Among these control methods, MPC becomes one of the major control strategies because of its intuitive control concept. Recently, MPC has many successful applications including chemicals, food processing, automotive, and aerospace applications [1].

Tuning of the MPC parameter is considered to be easy for skillful MPC controller designers. However, the relation between the tuning parameters and physical meanings is not always clear. Therefore, the parameter tuning is difficult for unskilled controller designers.

Tuning techniques for the MPC have been proposed in many literatures. Iino et al. proposed a parameter tuning method considering robust stability based on frequency response analysis [2]. Rowe et al. proposed the H-infinity loopshaping method [3]. Drogies et al. proposed a heuristic tuning method based on expert rules [4]. Iino's and Rowe's methods require complicated computing procedure and it is difficult to use it by unskilled ordinary controller designers. Practically, Drogies' expert rules have to be modified for the target problem. Namely, the previous tuning methods are still difficult for unskilled controller designers and simple MPC tuning method is eagerly required in industries.

Tuning of MPC can be formulated as an inverse problem and optimization techniques can be applied. Since MPC should be treated as a blackbox in the inverse problem, conventional mathematical programming type optimization techniques cannot be applied and evolutionary computation type optimization techniques should be applied. In the formulation, state variables are continuous and PSO [5] is one of the best candidates for the problem. Recently, PSO has been applied in industries practically [6-8] and a PSO-based tuning method must be accepted.

This paper presents an automatic tuning of the model predictive control using particle swarm optimization. The proposed PSO-based Automatic Tuning for MPC (PSO-MPC) computes the optimal tuning using time-domain performance criterion. In the proposed PSO-MPC, as described above, MPC is treated as a blackbox and unskilled ordinary controller designers do not have to care MPC itself. Moreover, it is verified through simulations that the proposed PSO-MPC can tune MPC parameters appropriately using the same PSO parameters. Namely, it is suggested that unskilled ordinary controller designers do not have to tune PSO and consequently they can tune MPC easily. The numerical results show the effectiveness of the proposed PSO-MPC.

II. MODEL PREDICTIVE CONTROL

In this section, brief summary of model predictive control and target parameters is described. Model predictive control has been first proposed by J.Richalet and C.Cutler independently at the end of 70's [9][10]. MPC estimates future behavior of the control target within a certain period using a model of the control target inside the controller. Then, it determines manipulated signals so that an objective function is minimized [1][11-13].

Illustrated comparison between MPC and conventional PID is shown in fig. 1. While a PID controller determines a manipulated signal using instantaneous observed and target values directly, MPC estimates behavior of the control target within a certain period in the future. Therefore, it consider not also instantaneous control property but also control property within a certain period in the future. While PID can only treat limiters as constraints, MPC can treat

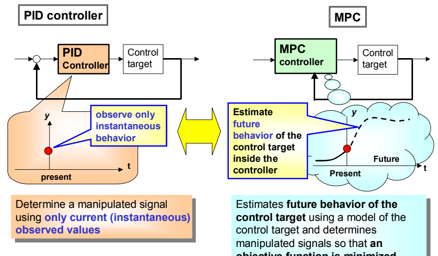

Fig. 1. Comparison between conventional PID controllers and MPC.

1-4244-0708-7/07/$20.00 ©2007 IEEE

various constraints of the control target when it minimize an objective function. This is another advantage compared with conventional PID controllers.

MPC can be formulated as follows. Concept of MPC is shown in Fig. 2. MPC utilizes predictive outputs $\hat{\boldsymbol{y}}$, which is estimated within $\boldsymbol{H_p}$ steps (prediction horizon) in the future using a internal model. It minimize the following objective function, which consists of weighted sum of tracking error of predictive outputs $\hat{\boldsymbol{y}}$ to set points $\boldsymbol{r}$, manipulated signals u, and variances of manipulated signals $\Delta\boldsymbol{u}$. A model predictive control problem can be formulated as an optimization problem, which determine input signals $\boldsymbol{u}(k)$, ..., $\boldsymbol{u}(k+H_c-1)$ within $\boldsymbol{H_c}$ steps (control horizon) in the future so that the objective function is minimized considering the following constraints [14].

(1) Objective function of MPC

$$J = \sum_{i=0}^{H_p-1}\{\left\|W_y(i)[r(k+i+1)-\hat{y}(k+i+1)]\right\|^2 + \left\|W_u(i)\hat{u}(k+i)\right\|^2 + \left\|W_{\Delta u}(i)\Delta\hat{u}(k+i)\right\|^2\} \quad (1)$$

$$\Delta\hat{u}(k+i) = \hat{u}(k+i) - \hat{u}(k+i-1),$$

where $W(i)$ is weight for each term, $r(i)$ is set point, $\hat{y}(i)$ is predictive output, $\hat{u}(i)$ is predictive control input.

(2) Constraints of MPC

$$y_{\min} \le \hat{y}(k+i) \le y_{\max} \quad ,i=0,\cdots,H_P-1 \quad (2)$$

where $y_{\min}$ is lower bound of output, $y_{\max}$ is upper bound of output.

$$u_{\min} \le \hat{u}(k+i) \le u_{\max} \quad ,i=0,\cdots,H_P-1 \quad (3)$$

where $u_{\min}$ is lower bound of control input, $u_{\max}$ is upper bound of control input.

$$\Delta u_{\min} \le \Delta\hat{u}(k+i) \le \Delta u_{\max} \quad ,i=0,\cdots,H_P-1 \quad (4)$$

where $\Delta u_{\min}$ is lower bound of control input difference, $\Delta u_{\max}$ is upper bound of control input difference.

It is assumed that input signals after control horizon are equal ($u(k+H_c-1)=u(k+H_c)=...=u(k+H_p-1)$). The only input signals at the first step are utilized as manipulated signals. After one sampling period, actual outputs are input and the same procedure (an optimization procedure of MPC) is applied using the prediction horizon, which is shifted one step to the future direction.

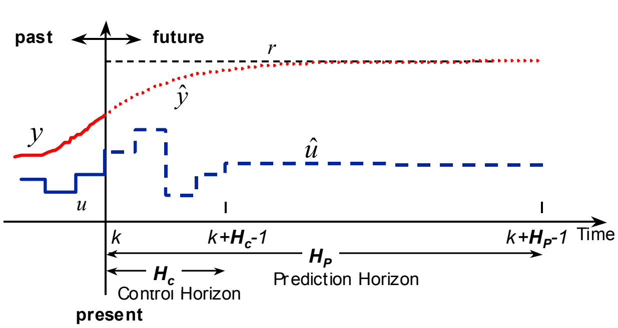

Fig. 2. Concept of MPC.

III. FORMULATION OF AN AUTOMATIC TUNING PROBLEM OF PARAMETERS IN MPC

This section shows a formulation of an automatic tuning problem of parameters in MPC. In the tuning problem, plant outputs from t=1, ..., tmax calculated by numbers of MPC are evaluated.

3.1 Target tuning Parameters of MPC

Generally, since prediction horizon $\boldsymbol{H_p}$, control horizon $\boldsymbol{H_c}$, and constraints are fixed, weighted coefficients $\boldsymbol{W_y}$, $\boldsymbol{W_u}$, and $W_{\Delta u}$ are tuned in this paper. In this paper, for the sake of simplicity, $\boldsymbol{W(i)}$ is fixed to constant values. Namely, in the tuning problem, it is assumed that objective function can be a function of weights of MPC, $\boldsymbol{W}$.

3.2 Problem Formulation

The target tuning problem can be formulated as follows:

(1) Objective function

The basic (single) objective function can be expressed as follows:

$$f_i = 1/G_i$$
$$G_i = (1-e^{-\beta})(E_{ss_i} + M_{p_i}) + e^{-\beta}(t_{s_i} - t_{r_i}) \quad (5)$$
$$M_{pi} = 100 \times \frac{M_{pi\max}}{y_{t\max}},$$

where β is weighted coefficient for response and variance property, E_{ssi} is steady state error, t_{si} is settling time, t_{ri} is rising time, $M_{pi\max}$ is maximum value of y, $y_{t\max}$ is y at the end time. If we have number of outputs, the objective function can be expressed as a min-max type function.

$$\max_x \min\{f_1, f_2, \cdots, f_m\}, \quad (6)$$

where m is number of output.

A general weighted sum type objective function can be utilized as well. However, according to our experience, min-max type function has better performance and we decided to utilize the min-max type objective function.

(2) Constraints

Upper and lower bounds of weights should be considered.

$$W_{yi\min} \le W_{yi} \le W_{yi\max}, \quad i = 1,\dots,m \tag{7}$$

$$W_{ui\min} \le W_{ui} \le W_{ui\max}, \quad i = 1,\dots,m \tag{8}$$

$$W_{\Delta ui\min} \le W_{\Delta ui} \le W_{\Delta ui\max}, \quad i = 1,\dots,m, \tag{9}$$

where W_{yimin} is lower bound of output y_i, W_{yimax} is upper bound of output y_i, W_{uimin} is lower bound of input u_i, W_{uimax} is upper bound of input u_i, $W_{\Delta\text{uimin}}$ is lower bound of input variance Δu_i, $W_{\Delta\text{uimax}}$ is upper bound of input variance Δu_i.

The concept of the tuning problem is shown in Fig. 3. Parameters (weights of the objective functions of MPC control problem) are searched in the upper tuning problem by PSO and they are sent to the lower level MPC problem. Next, the outputs of the process are calculated by MPC and they are sent to the upper level PSO. Then, objective function values in the upper level tuning problem are calculated. This procedure is repeated until the predetermined iteration number. As shown in the figure, using this tuning concept, MPC parameters are automatically tuned by PSO and unskilled control designers can utilize it easily. When state variables are updated at the one iteration step of PSO, lower optimization problem (MPC) is calculated as shown in Fig.3. Therefore, total optimization procedure (integration of upper and lower level) is expected to be time-consuming. However, because of recent progress of computer performance, this kind of brute force procedure can be realized.

3.3 Characteristics of the Problem

In this section, characteristics of the tuning problem, especially the shape of the objective function is examined. Fig. 4 and 5 show shapes of the objective functions of PID controllers and MPC. In PID, proportional, derivative, and Integral parameters (k_p, k_d, k_i) are tuned. The two controllers are applied to the same process and have the same manipulator. Conventional Controller tuning literatures by PSO have already published [15][16]. As shown in these figures, a MPC tuning problem has to treat a more complicated and multimodal objective function than a PID controller tuning problem.

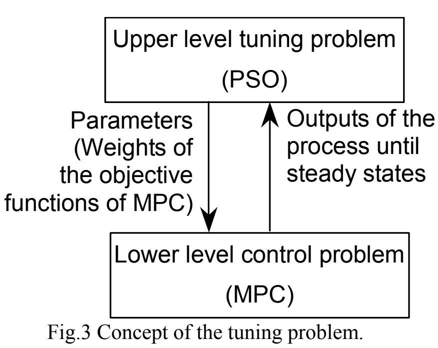

Fig.3 Concept of the tuning problem.

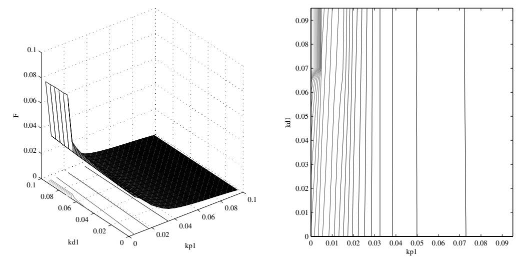

Fig. 4. An example of the shape of an objective function of a PID tuning problem (two PIDs).

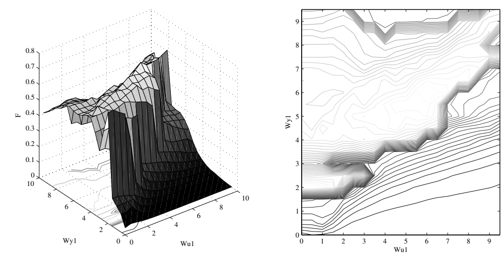

Fig. 5. An example of the shape of an objective function of a MPC tuning problem.

IV. AUTOMATIC TUNING OF MPC USING PSO

This section shows the way how automatic tuning of MPC is realized using PSO. The target tuning problem described in chap.III can be rewritten as the following optimization problem with upper and lower bound constraints:

$$\max_{W} F(W_u, W_{\Delta u}, W_y) \tag{10}$$

$$F = \min\{f_1, f_2, \cdots, f_n\} \tag{11}$$

$$W_{i\max} \le W_i \le W_{i\max}, \quad i = 1,\cdots,3m.$$

Equation (10)(11) can be expressed as an optimization problem (13) without constraints using the following transformation function (12):

$$h(W_i^{'}) = \begin{cases} W_{i\min} & if\ W_i^{'}(k) < W_{i\min}, \\ W_i^{'}(k) & if\ W_{i\min} < W_i^{'}(k) < W_{i\max}, \\ W_{i\max} & if\ W_{i\max} < W_i^{'}(k). \end{cases} \tag{12}$$

$$\max_{W'} F(h(W_y^{'}, W_u^{'}, W_{\Delta u}^{'})) \tag{13}$$

$$\begin{pmatrix} W^p_{u,1} \\ \vdots \\ W^p_{u,m} \\ W^p_{\Delta u,1} \\ \vdots \\ W^p_{\Delta u,m} \\ W^p_{y,1} \\ \vdots \\ W^p_{y,m} \end{pmatrix} = \begin{pmatrix} W^p_1 \\ \vdots \\ W^p_{3m} \end{pmatrix} \quad (14)$$

The basic "Gbest" model is utilized in this paper. An update equation of agents can be expressed as follows:

$$v_i^p(k+1) = \lambda v_i^p(k) + c_1(W_i^{pbest^p} - W_i'^p(k)) + c_2(W_i^{gbest} - W_i'^p(k)) \quad (15)$$

$$W_i'^p(k+1) = W_i'^p(k) + v_i^p(k+1) \quad (16)$$

The update equations of *gbest* and *pbest*s can be expressed as follows:

$$pbest^p = \arg\max_l \{F(W^p(l) \mid l = 0,1,\cdots,k)\}, (p = 1,\cdots,P) \quad (17)$$

$$gbest = \arg\max_p \{F(W^{pbest^p} \mid p = 0,1,\cdots,P)\}, \quad (18)$$

where P is total number of agents.

A calculation procedure of PSO-MPC can be expressed as follows:

Step1 [Initialization procedure]

Initialize $W_i^p(0)$ of agent p(i=1,...,3m) using uniform random numbers within upper and lower bounds of W_i. Weights $W_i^p(0)$ calculated at step 1 are sent to MPC and a control model simulates the target plant behavior until *t*max. The evaluation values are calculated by (5). *Pbest*s and gbest are calculated using (17)(18). The procedure is repeated for p=1,...,P.

Step2 [Update of *pbest*s and *gbest*]

Weights W_i^p(k) calculated at step 1 are sent to MPC and a control model simulates the target plant behavior until *t*max. The evaluation values are calculated by (5). *Pbest*s and gbest are calculated using (17)(18). The procedure is repeated for p=1,...,P. Considering upper bounds of v_i^p, v_i^p can be expressed as follows:

$$v_i^p(k) = \text{sgn}(v_i^p(k)) \min\{\| v_i^p(k) \|, 0.5(W_{i\max} - W_{i\min})\} \quad (19)$$

*pbest*s and *gbest* are updated using(17)(18).

Step3 [Stop criterion and update of the weighting function]

The procedure is stopped when k reaches $k$max (maximum iteration number). Otherwise, k=k+1 and λ of (14) is updated using the following equation (inertia weights approach) and go to Step2:

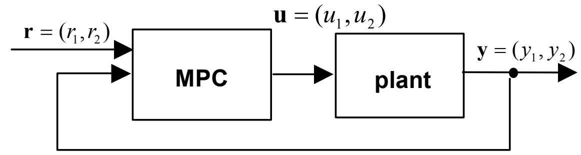

Fig. 6. Block diagram of the plant with a MPC controller.

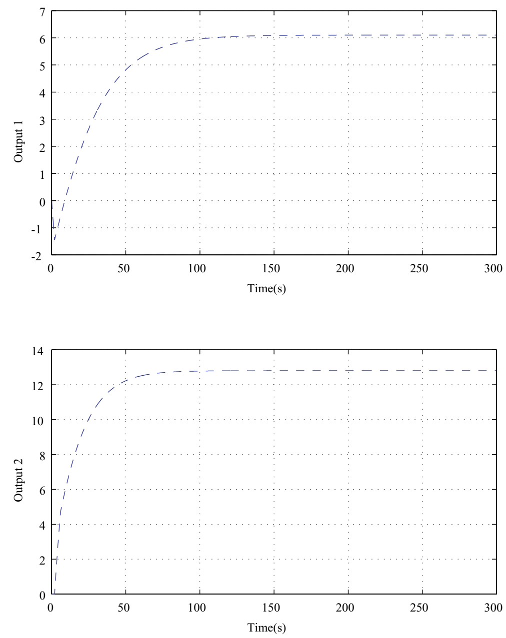

Fig. 7. Step response of the plant without the PSO-MPC controller.

$$\lambda = \lambda_{\max} - \frac{\lambda_{\max} - \lambda_{\min}}{k_{\max}} \times k \quad (19)$$

V. NUMERICAL EXAMPLES

For the purpose of illustrating the effectiveness of the proposed method, two examples are studied numerically.

5.1 Numerical Example No.1

(1) Simulation Conditions

The following plant is tested in example No.1.

$$\begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix} = \begin{bmatrix} \dfrac{-12.8e^{-s}}{16.7s+1} & \dfrac{18.9e^{-3s}}{21s+1} \\ \dfrac{-6.6e^{-7s}}{10.9s+1} & \dfrac{19.4e^{-3s}}{14.4s+1} \end{bmatrix} \begin{bmatrix} U_1 \\ U_2 \end{bmatrix} \quad (20)$$

The block diagram of PSO-MPC controller and the plant is shown in Fig. 6. Fig. 7 shows the original step responses of the plant without the PSO-MPC controller. As shown in these figures, steady state errors are appeared in the original responses. In this simulation, the following parameter settings are utilized through pre-simulations:

$\beta = 1, P = 6, k_{\max} = 50, c_1 = c_2 = 2.0, \lambda_{\max} = 0.9, \lambda_{\min} = 0.4;$

These parameters are fixed for all simulations. The following MPC controller parameters are also utilized in the simulation:

$$0 \le w_{\Delta u,i} \le 10,\ 0 \le w_{u,i} \le 10,\ 0 \le w_{y,i} \le 10$$
$$-220 \le u_i \le 220,\ -\infty \le \Delta u_i \le \infty,\ 0 \le y_i \le 1 \quad i = 1,2$$
$$H_p = 10,\ H_c = 2.$$

As shown in the Fig.3, MPC parameters are automatically tuned by PSO and unskilled control designers can utilize it easily. Therefore, practical applicability of the proposed method depends on whether *appropriate* parameters can be obtained within *appropriate* calculation time. In addition,

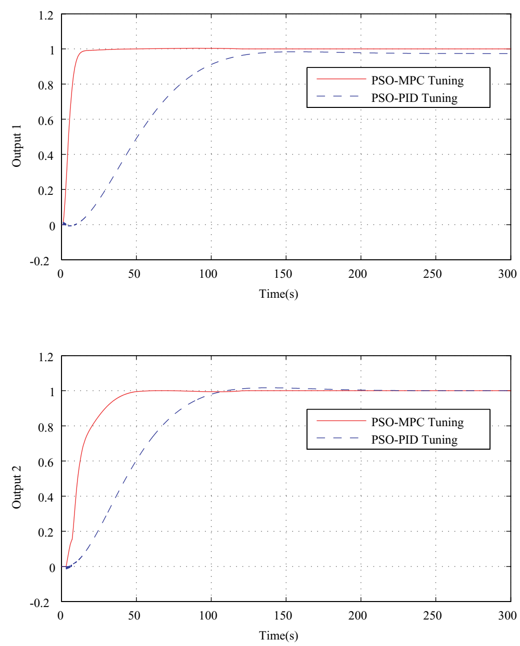

Fig. 8. Step response of the plant with different controllers.

Table 1 The optimal parameters.

PSO- PID tuning		PSO-MPC tuning	
$k_{p,1}$	0.000000	$W_{u,1}$	0.4997
$k_{d,1}$	0.000175	$W_{\Delta u,1}$	9.4196
$k_{d,1}$	0.085051	$W_{y,1}$	1.8154
$k_{p,2}$	0.036728	$W_{u,2}$	0.8394
$k_{i,2}$	0.003548	$W_{\Delta u,2}$	9.4899
$k_{d,2}$	0.008873	$W_{y,2}$	3.9619

Table 2 PIDC & MPC tuning results.

Objective function value	PID tuning	MPC tuning
Mean value	0.067875	0.5488
Best value	0.079716	0.9039
Worst value	0.055561	0.3235
Standard deviation	4.42321e-05	0.031269

in order to verify the advantage of the proposed PSO-MPC, we also apply a PSO based PID tuning method (PSO-PID) to the same plant with two PID controllers. In this simulation, the same fixed parameters β, P, $k_{\max}$, c_1, c_2, λ_{max}, λ_{min} are utilized for PSO-PID. Upper and lower bounds of PID parameters are set to 0 and 0.01. PSO-MPC and PSO-PID are compared using the same objective function (5).

(2) Simulation Results

The best values of the optimized parameters are given in Table 1. In order to obtain close parameters in Table 1, skilled control designers spent approximately 8 hours. On the contrary, it only takes 3 minutes by PSO-MPC. Fig. 8 shows step responses of the plant with the PSO-MPC and the PSO-PID controllers. In order to compare their convergence characteristics, best, average, worst, and standard deviation of the objective function values are evaluated through 50 trials with different initial conditions as shown in Table 2. According to the results, the PSO-MPC controller has better evaluation values than the PSO-PID controller. Since, PSO-PID can generate the best control responses of the conventional method, it is said that *appropriate* parameters can be obtained within *appropriate* calculation time by PSO-MPC.

5.2 Numerical Example No.2

The proposed method is applied to a practical plant model with 4 inputs and 4 outputs. Output signals are mutually coupled (interfered or have strong connections) in the plant. Simulation for 500 cases (500 combinations of parameters) takes approximately 100 hours. On the contrary, it only takes approximately 18 minutes by PSO-MPC. Best, average, worst, and standard deviation of the objective function values are evaluated through 50 trials with different initial conditions as shown in Table 3. Fig. 9 shows step responses of the plant with and without the PSO-MPC (best results). According to the results, it is observed that appropriate parameters are obtained by PSO-MPC.

Table 3. Objective function values through 50 trials.

Mean value	45.7207
Best value	66.7299
Worst value	32.3397
Standard deviation	75.69687

VI. CONCLUSIONS

This paper presents an automatic tuning method of model predictive control (MPC) using particle swarm optimization (PSO-MPC). This is a first application of PSO for automatic tuning of MPC. Using PSO-MPC, unskilled control designer can tune MPC easily within appropriate tuning time.

Future works are as follows. In this paper, only step responses are evaluated. We will develop PSO-MPC evaluating step responses, disturbance responses, and other items. In this paper, weights ***W*** are assumed to be fixed all time. Calculation of ***W(i)*** (different W for each time) is one of the future works for improvement of control property. Moreover, development of appropriate man-machine interface (MMI) is one of the main issues for unskilled control designers.

REFERENCES

[1] S.J.Qin and T.A.Badgwell, "A survey of industrial model predictive control technology", Control Engineering Practice, No.11, pp733–764, 2003.

[2] Y. Iino, et al., “A New Input/Output Constrained Model Predictive Control with Frequency Domain Tuning Technique and Its Application to an Ethylene Plant,” Proceedings of the IECON '93, 1993.

[3] C. Rowe et al., “Tuning MPC using H-infinity Loop Shaping,” Proceedings of the 2000 American Control Conference, Vol. 2, 2000.

[4] S. Drogies et al., “Predictive Control: Propositions for the Design Methodology,” Proceedings of the 1999 American Control Conference, Vol. 1, 1999.

[5] J. Kennedy and R. Eberhart, *Swarm Intelligence*, Morgan Kaufmann Publishers, 2001.

[6] Y. Fukuyama, "Foundation of Particle Swarm Optimization", in chapter 5 of *tutorial text on Modern Heuristic Optimization Techniques with Application to Power Systems*, organized by IEEE Power Engineering Society, IEEE Winter Power Meeting, January 2002.

[7] Y. Fukuyama, et al, "A Particle Swarm Optimization for Reactive Power and Voltage Control Considering Voltage Security Assessment", *IEEE Trans. on Power Systems*, Vol.15, No.4, 2000.

[8] S. Kitagawa and Y. Fukuyama, "Comparison of Particle Swarm Optimizations for Optimal Operational Planning of Energy Plants", *Proc. of Swarm Intelligence Symposium (SIS2005)*, 2005.

[9] J.Richalet, A.Rault, J.L.Testud and J.Papon, "Model predictive heuristic control: applications to industrial processes", Automatica, Vol.14, No.5, pp.413-428, 1978.

[10] C.R.Cutler and B.L. Ramaker, "Dynamic matrix control – A computer control algorithm", In Proceedings, Joint American Control Conferernce, San Francisco, U.S.A., 1980.

[11] J.M.Maciejowski, Predictive Control with Constraints, Prentice Hall, 2001.

[12] R.Soeterboek, Predictive Control. A unified approach, Prentice Hall, 1992.

[13] C.A.Garcia, D.M.Prett and M.Morari, "Model Predictive Control: Theory and Practice –a Survey", Automatica, Vol.25, No.3, pp.335-348, 1989.

[14] A.Bemporad, N.L.Ricker and J.G.Owen, "Model Predictive Control - New tools for design and evaluation", In Proceedings, American Control Conference, Boston, U.S.A., 2004.

[15] M. Abido, "Optimal Design of Power System Stabilizers using Particle Swarm Optimization", IEEE Trans. on Energy Conversion, Vol.17, No.3, pp.406-413, Sep. 2002.

[16] Z. L. Gaing, "A Particle Swarm Optimization Approach for Optimum Design of PID Controller in AVR system", IEEE Trans. on Energy Conversion, Vol.19, No.2, pp.384-391, June 2004.

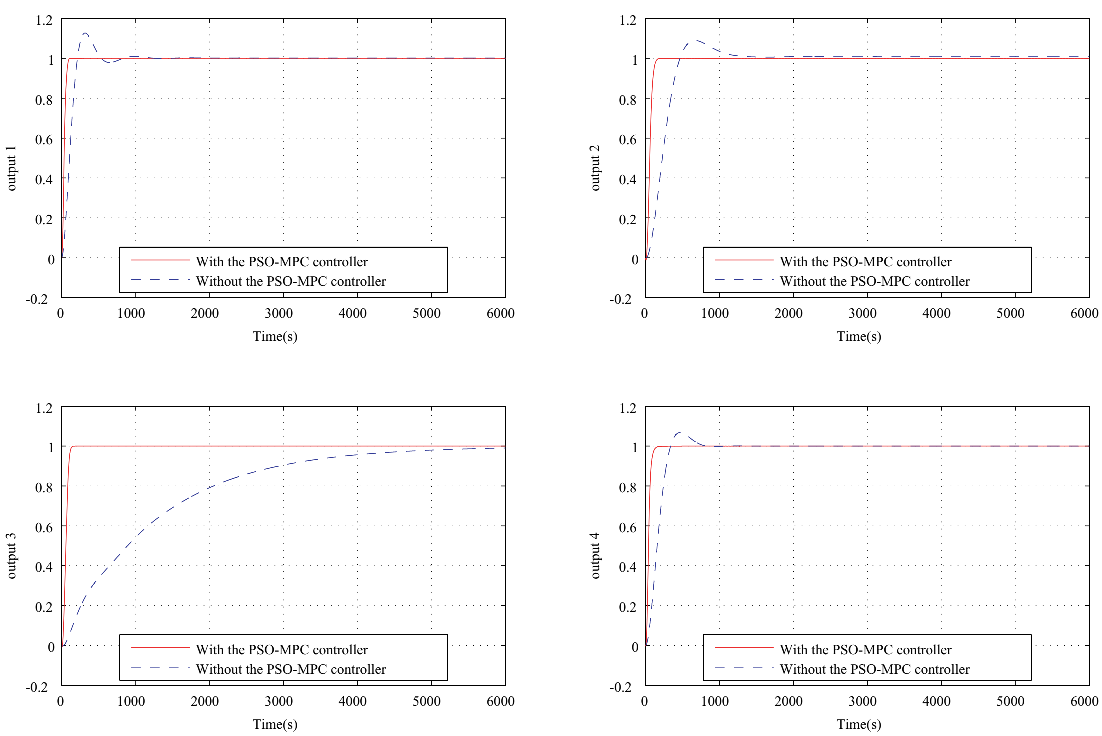

Fig. 9. Step response of the 4 inputs and 4 outputs plant with the PSO-MPC controller.