

Mean arterial pressure control system using model predictive control and particle swarm optimization

Te-Jen Su^{1,3} · Shih-Ming Wang¹ · Hong-Quan Vu¹ · Jau-Ji Jou¹ · Cheuk-Kwan Sun²

Received: 26 October 2016 / Accepted: 16 November 2016
© Springer-Verlag Berlin Heidelberg 2016

Abstract Linear controllers have been designed to regulate mean arterial pressure (MAP) in treating various cardiovascular diseases. For patients with hemodynamic fluctuations, the MAP control system must be able to provide more sensitive control. Therefore, in this paper, a model predictive control (MPC) approach is presented to improve the sensitivity of MAP control system. The MPC principle can effectively handle the dead times in nonlinear systems, and can also optimize the system responses when subjected to constraints of process states and control signals. Besides, particle swarm optimization (PSO) is employed to solve the optimization problem of MPC at each control interval. According to our simulations, the MAP control system with combined MPC–PSO approach is superior in control qualities to the MAP control system with conventional linear control method. The MPC–PSO MAP control system is possible to be realized through a field-programmable gate array.

1 Introduction

Mean arterial pressure (MAP) is the average blood pressure (BP) over a cardiac cycle and is determined by the cardiac output (CO), systemic vascular resistance (SVR) and

central venous pressure (CVP). The stable control of MAP is important in the prevention of acute life-threatening condition such as hemorrhagic stroke and the deterioration of chronic hypertension-associated morbidities. Previous studies have shown that MAP is more accurate than systolic BP, diastolic BP and pulse pressure in predicting future metabolic syndrome among the normotensive elderly population (Hsu et al. 2015). According to that research, an MAP higher than the cutoff value indicates an elevated risk of developing metabolic syndrome. For the patients following cardiac arrest, hypoxic-ischemic brain injury is the major cause of death (Padmanabhan et al. 2015). While an MAP below the auto-regulatory threshold leads to additional ischemia and further brain injury, an elevated MAP above the auto-regulatory threshold causes excessive brain perfusion that may result in increased cerebral edema and worsening the pre-existing brain injury (Sekhon et al. 2016). It is, therefore, suggested that keeping the MAP within an optimal range using the relationship between brain tissue regional saturation of oxygen and MAP is critical for survival for this patient population (Sekhon et al. 2016). Hypotensive anesthesia, which is a widely used technique aimed at reducing intraoperative bleeding and the need for post-operative blood transfusions in general surgical procedures, requires administration of multiple drugs to regulate key physiological variables, such as the level of unconsciousness, heart rate, mean arterial pressure (MAP), respiratory rates, and other vital parameters within desired limits (Jeong et al. 2016). The fast-acting vasodilating agent, sodium nitroprusside, is often used for the induction of hypotensive anesthesia to lower blood pressure using the recommended close-loop feedback control that involves fine tuning of the drug infusion rate according to the rate of pressure measurement. Under this circumstance, manual control by clinical personnel is not preferred because it is

✉ Jau-Ji Jou
jjjou@kuas.edu.tw

¹ Department of Electronic Engineering, National Kaohsiung University of Applied Sciences, Sanmin District, Kaohsiung 80778, Taiwan
² Department of Emergency Medicine, E-Da Hospital, I-Shou University, Yan-Chao District, Kaohsiung 82445, Taiwan
³ Graduate Institute of Clinical Medicine, Kaohsiung Medical University, Sanmin District, Kaohsiung 80708, Taiwan

both time-consuming and risky for patients because of the lack of accuracy in dosage control.

Previous studies (Luginbühl et al. 2006; Zhu et al. 2008; Hoeksel et al. 2001; Liu et al. 2009) have reported a number of classical approaches to realize the MAP control systems, and the MAP is an important clinical parameter to be kept within a narrow physiological range in the settings of anesthesia, surgery, and critical care. To refine the conventional MAP control systems, soft computing approaches such as fuzzy logic, neural networks, reinforcement learning have been proposed (Furutani et al. 2004; Kumar et al. 2009; Gao and Er 2003; Lin et al. 2008; Ferreira et al. 2009; Padmanabhan et al. 2015). Some techniques of these MAP control systems rely on the use of proportional-integral-derivative (PID) controllers, for which rather complex design methodologies are used. Using field-programmable gate array (FPGA) technology, the controllers can be realized as integrated circuits (ICs) (Sánchez-Durán et al. 2012). This study tested the validity of an MAP control system based on the combination of model predictive control (MPC) and particle swarm optimization (PSO). The control qualities of this approach were compared with those of the classical control approaches.

2 Model predictive control

Model predictive control, which is also known as receding horizon control or moving horizon control, bases on the past, present, and future information of the process and environment to generate the control actions (Holkar and Waghmare 2010). The principle of MPC is shown in Fig. 1 (Holkar and Waghmare 2010). Suppose the output's reference value, $\mathbf{w}(t)$, is already known in the time range $(t, t + N)$ in which t is the current time and N is the predictive horizon, a set of N_u (control horizon, $N_u \leq N$) values of control variable $\mathbf{u}(t + j|t)$, $0 \leq j \leq N_u$ will be calculated to

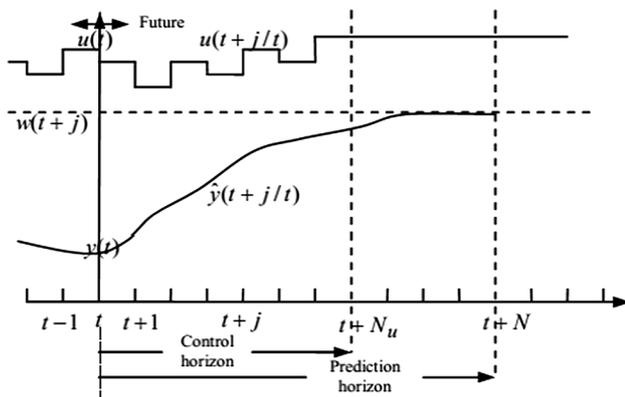


Fig. 1 The principle of model predictive control (MPC)

minimize the error between the predicted outputs $\hat{\mathbf{y}}(t + j|t)$ and the reference $\mathbf{w}(t + j)$ in the range $1 \leq j \leq N$ while subject to constraints of control variables. Consider the following cost function and constraints:

$$J(t) = \sum_{i=0}^{N-1} \|W_y(i + 1)[\mathbf{w}(t + i + 1) - \hat{\mathbf{y}}(t + i + 1|t)]\| + \sum_{i=0}^{N_u} W_{\Delta u}(i) \|\mathbf{u}(t + i) - \mathbf{u}(t + i - 1)\|, \quad (1)$$

$$\mathbf{x}(t + 1) = f(\mathbf{x}(t), \mathbf{u}(t)), \quad (2)$$

$$\mathbf{y}(t) = g(\mathbf{x}(t), \mathbf{u}(t)), \quad (3)$$

$$\mathbf{u}_{\min} \leq \|\mathbf{u}(t)\| \leq \mathbf{u}_{\max} \quad (4)$$

$$\mathbf{x}_{\min} \leq \|\mathbf{x}(t)\| \leq \mathbf{x}_{\max}, \quad (5)$$

This problem is to minimize the cost function (1) with constraints (2–5), where $W_y(i)$, $W_{\Delta u}(i)$ are the weighted coefficient matrices of outputs and control variables; \mathbf{x}_{\min} , \mathbf{x}_{\max} are the lower bound and upper bound of state variables; \mathbf{u}_{\min} , \mathbf{u}_{\max} are the lower bound and upper bound of control variables, respectively. This optimization problem is either linear or nonlinear; the constraints also are linear or nonlinear, convex or non-convex. Nature-inspired metaheuristic optimization methods are alternative when the optimization problems are high-dimensional, nonlinear with the complex constraints.

The MPC algorithm is described in four steps (Holkar and Waghmare 2010).

Step 1: At the current time t , measure the current outputs $\mathbf{y}(t)$ and the current states $\mathbf{x}(t)$.

Step 2: Estimate the future states $\mathbf{x}(t + j)$ and outputs $\mathbf{y}(t + j)(j = 1, 2, \dots, N)$. After that, construct the cost function and the constraints.

Step 3: Minimize the cost function and get the optimal control sequence:

$$U^*(t) = (\mathbf{u}_0^*(t), \mathbf{u}_1^*(t), \dots, \mathbf{u}_{N_u-1}^*(t)).$$

Step 4: Apply $\mathbf{u}_0^*(t)$ to the real process. In the next sampling time $t = t + 1$, the algorithm is repeated from Step 1.

3 Particle swarm optimization

Particle swarm optimization (PSO) is a meta-heuristic optimization algorithm inspired by food-seeking behaviors of bird flockings or fish schoolings. This method, which needs no gradient information of object function as well as its mathematical representation, is suitable for solving high

dimension, non-linear and non-convex optimization problems. The standard PSO algorithm works as follows: Imagine that, a flock of birds is searching for food in an area. All of the birds do not know where exactly the food is but the sense of the distance from each one to the food. Their behavior is following the birds that are closest to the food. PSO mimics that strategy to solve the optimization problems. In PSO, each ‘bird’ is modelled as a particle, which is a potential solution for the problem. The ‘distance from each bird to the food’ becomes the value of cost function (or fitness value) at the corresponding ‘particle’. The best position of each particle found until now is called the local best particle $\mathbf{p}_{i,best}$. The particle, which has optimum fitness value overall group of particles, is called the global best particle \mathbf{g}_{best} . After each iteration, the position and velocity of each particle is updated following these ‘best’ particles $\mathbf{p}_{i,best}$ and \mathbf{g}_{best} .

The algorithm of minimizing $f(x)$ in search space $\mathbf{x} \in D$ using PSO is presented through the following steps (Yang 2010; Kennedy and Eberhart 1995):

Step 1: Randomly initialize P particles with its position \mathbf{p}_i^0 and velocity \mathbf{v}_i^0 , the best positions $\mathbf{p}_{i,best}$ and \mathbf{g}_{best} as follows:

$$\begin{aligned} \mathbf{x}_{\min} &\leq \mathbf{p}_i^0 \leq \mathbf{x}_{\max} \\ \mathbf{v}_{\min} &\leq \mathbf{v}_i^0 \leq \mathbf{v}_{\max} \\ \mathbf{p}_{i,best}^0 &= \mathbf{p}_i^0 \quad (1 \leq i, k \leq P) \\ \mathbf{g}_{best}^0 &= \{\mathbf{p}_k^0 | f(\mathbf{p}_k^0) = \min_{1 \leq i \leq P} f(\mathbf{p}_i^0)\}. \end{aligned}$$

Step 2: At the iteration j ($1 \leq j \leq iteration_{\max}$), update the velocity of all particles as Eq. (6).

$$\mathbf{v}_i^j = \omega^j \mathbf{v}_i^{j-1} + c_1 r_1 (\mathbf{p}_{i,best}^{j-1} - \mathbf{p}_i^{j-1}) + c_2 r_2 (\mathbf{g}_{best}^{j-1} - \mathbf{p}_i^{j-1}) \quad (6)$$

in which r_1, r_2 are the random numbers within $[0, 1]$, c_1, c_2 are the factors used to control the related weighting of the corresponding terms, ω^j is the parameter which control the dynamic of velocity, determined as:

$$\omega^j = \omega_{\max} - \frac{\omega_{\max} - \omega_{\min}}{iteration_{\max}} j, \text{ with } \omega_{\max}, \omega_{\min} \text{ are pre - defined positive values.}$$

The use of random variables endows the PSO with the ability of stochastic searching. After updating, the velocities of all the particles should be checked and clamped to the legal range to avoid violent random walking as follows:

$$\mathbf{v}_i^j = \begin{cases} \mathbf{v}_i^j & \text{if } \mathbf{v}_{\min} \leq \mathbf{v}_i^j \leq \mathbf{v}_{\max} \\ \mathbf{v}_{\min} & \text{if } \mathbf{v}_i^j < \mathbf{v}_{\min} \\ \mathbf{v}_{\max} & \text{if } \mathbf{v}_i^j > \mathbf{v}_{\max}. \end{cases}$$

Step 3: Update the position of all the particles by formula (7):

$$\mathbf{p}_i^j = \mathbf{p}_i^{j-1} + \mathbf{v}_i^j. \quad (7)$$

After updating, new positions should be checked and modified so that they still belong to the legal range, as follows:

$$\mathbf{p}_i^j = \begin{cases} \mathbf{p}_i^j & \text{if } \mathbf{p}_{\min} \leq \mathbf{p}_i^j \leq \mathbf{p}_{\max} \\ \mathbf{p}_{\min} & \text{if } \mathbf{p}_i^j < \mathbf{p}_{\min} \\ \mathbf{p}_{\max} & \text{if } \mathbf{p}_i^j > \mathbf{p}_{\max}. \end{cases}$$

Step 4: Update $\mathbf{p}_{i,best}$ and \mathbf{g}_{best} as (8) and (9):

$$\mathbf{p}_{i,best}^j = \begin{cases} \mathbf{p}_i^j & \text{if } f(\mathbf{p}_i^j) < f(\mathbf{p}_{i,best}^{j-1}) \\ \mathbf{p}_{i,best}^{j-1} & \text{otherwise,} \end{cases} \quad (8)$$

$$\mathbf{g}_{best}^j = \begin{cases} \mathbf{p}_i^j & \text{if } f(\mathbf{p}_i^j) < f(\mathbf{g}_{best}^j) \text{ for all } 1 \leq i \leq P \\ \mathbf{g}_{best}^{j-1} & \text{otherwise.} \end{cases} \quad (9)$$

Step 5: Repeat Step 2 to Step 4 until terminal conditions are met (usually a sufficient good fitness or a maximum number of iterations). Output the best particle and corresponding best fitness value.

4 Mean arterial pressure model

The relationship between drug infusion rates and the rates of reduction in MAP is shown in the following transfer function (de Moura Oliveira et al. 2014):

$$G_p(s) = \frac{Y(s)}{U(s)} = \frac{K(1 + T_3s)e^{-\theta s}}{((1 + T_3s)(1 + T_2s) - \alpha)(1 + T_1s)},$$

where Y is the drop in pressure, due to the drug effect, U represents the rate of drug infusion. K is the system gain; T_1, T_2, T_3 represent time constants associated with the drug action; θ is the system time delay and α is the fraction of recirculated drug. The parameters of the model depend directly on each specific patient. In most common cases, we can take $K = 2.5$ mmHg/ml/h, $T_1 = 50$ s, $T_2 = 10$ s, $T_3 = 30$ s, $\theta = 60$ s, $\alpha = 0.5$.

The clear model for controller design is written in (10):

$$G_p(s) = \frac{Y(s)}{U(s)} = \frac{5(1 + 30s)e^{-60s}}{1 + 130s + 4600s^2 + 30000s^3} \quad (10)$$

Figure 2 plots the response of the given system with a step change at $t = 0$ s. The rising time $T_r \cong 180$ s, $T_s \cong 250$ s. Fig. 3 shows the Bode plot of the model. It is observed that the cutting frequency is $f_c \cong 0.011$ rad/s.

The design objectives are to control the system represented by model (10) in order to achieve following time-domain criteria: fast response, small overshoot, and zero

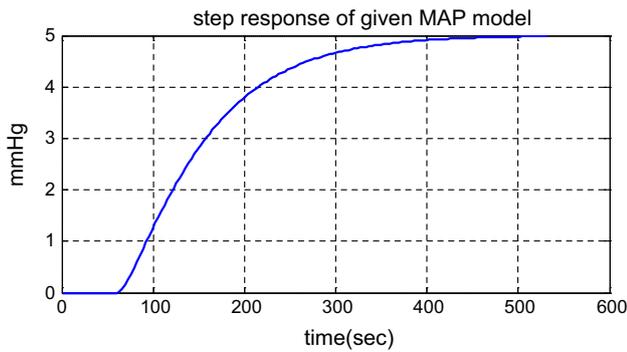


Fig. 2 The step response of the given mean arterial pressure (MAP) system

steady-state error. The designed controllers should also satisfy actuator clinical constraints such as saturation limits and drug infusion rate of change.

The PID control designs for MAP are proposed by many studies, and it worked well in most patients. For example, Fig. 4 shows the performances of an optimal PID controller design (de Moura Oliveira et al. 2014). The rise time of this PID control system is about 110 s. However, the settling time is about 200 s. For certain groups of patients, particularly those sensitive to drugs or those with fluctuating blood pressure, the system performance needs a significant improvement. In this design, we aim at improving system performances by using MPC–PSO approach.

5 Design of MPC–PSO controller

The first design step is changing the continuous model to discrete model. The sampling frequency should be at least two times bigger than the cutting frequency of the system or $T_s \leq 45$ s. The higher the sampling frequency, the better

Fig. 3 The Bode diagram of the given mean arterial pressure (MAP) system

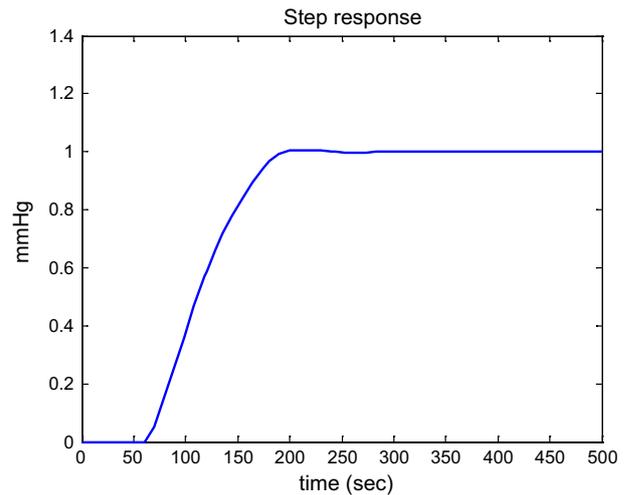
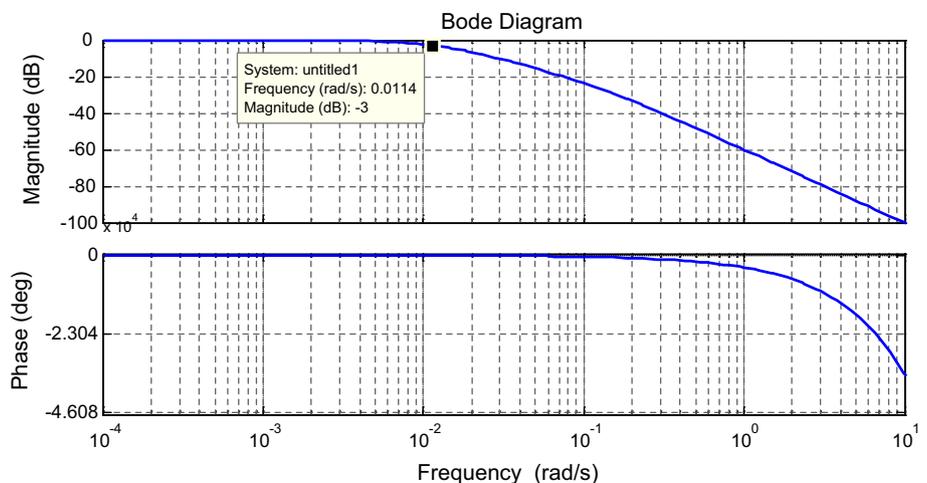


Fig. 4 The control performance of an optimized PID controller

the control quality with a higher calculation cost. In this design, we choose $T_s = 10$ s. The discrete model of the system is:

$$G_p(z) = z^{-6} \frac{0.1735z^2 - 0.00075z - 0.08}{z^3 - 1.992z^2 + 1.224z - 0.125}, \quad (11)$$

The output of the system can be calculated as:

$$y(k) = 1.992y(k - 1) - 1.224y(k - 2) - 0.125y(k - 3) + \dots + 0.1735u(k - 7) - 0.00075u(k - 8) - 0.08u(k - 9). \quad (12)$$

The second step is design the parameters of the MPC. The predicted horizon N should be bigger than the input delay. For a small value of N , the system response is very quick, but tends to have large overshoots. In contrast, if N is big, the system responds slower but without overshooting. For the good choice of N , the simulation software is

Table 1 Parameters of PSO

Parameters	Values
Swarm size (P)	20
Iteration limit	40
Search space D	$[-2, 2]$
c_1	2
c_2	2
ω_{\min}	0.4
ω_{\max}	0.9

used. The control horizon is fixed at $N_c = 2$. The cost function becomes:

$$J(t) = \sum_{i=1}^N ((\hat{y}(t+i|t) - y^{ref})^2) + \sum_{i=1}^2 0.1(u(t+i) - u(t+i-1))^2. \quad (13)$$

The constraint for proposed controller is the limitation of the drug infusion rate:

$$u \leq 2 \text{ (ml/h)}. \quad (14)$$

The implemented procedures of the proposed controller are described through following steps:

Step 1: At the current time t , measure the current outputs $y(t)$ and (or observe) the current states $x(t)$.

Step 2: Estimate the future outputs $\hat{y}(t+i|t)$ with $i = 1, 2, \dots, N$ by (12) assuming zero initial conditions. After that, form the cost function (13) and the constraint (14).

Step 3: Minimize the cost function (13) using PSO. The implementation procedure of PSO includes five steps described in Sect. 3. All the parameters of PSO are listed in Table 1. After finishing, the set of global best particles, which is also the best solution of optimization problem, is obtained:

$$U^*(t) = (u_0, u_1)$$

Step 4: Only apply u_0 to the real process.

Fig. 5 The simulation structure of the model predictive control–particle swarm optimization

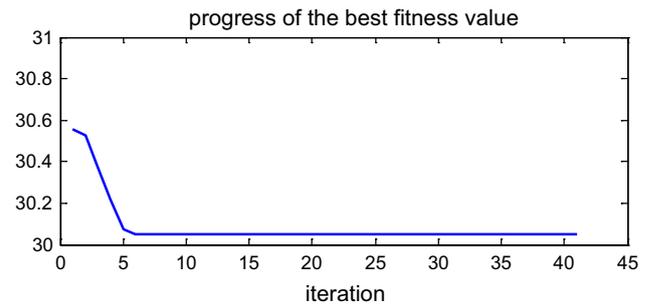
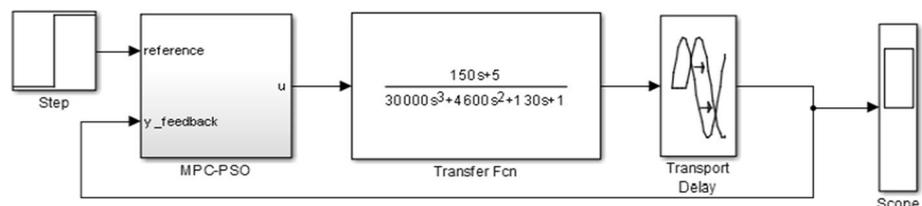


Fig. 6 The typical convergence progress of the PSO algorithm

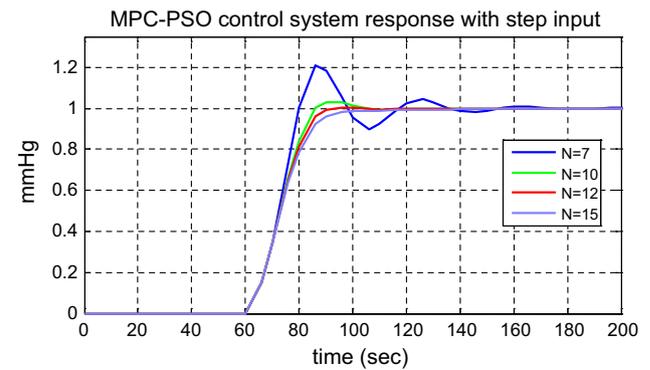


Fig. 7 The responses of the MPC–PSO control system with a step change at input

Step 5: In the next sampling time $t + 1$, repeat the Step 1 to Step 4.

The simulation diagram using MATLAB/SIMULINK is shown in Fig. 5.

6 Simulation results and discussions

Figure 6 describes the typical convergence progress each time running PSO. Just after several iterations, the fitness value reaches the smallest one. As the simulation time long enough, these smallest values gradually converge to zero. It proves the strong ability of PSO in finding the optimal solutions without requiring complex mathematical transformations.

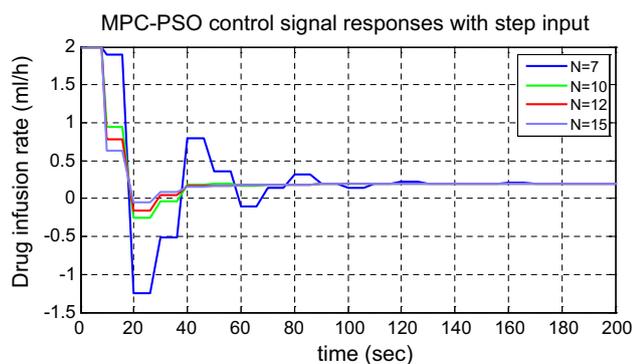


Fig. 8 The control signals of the MPC–PSO control system with the step change at input

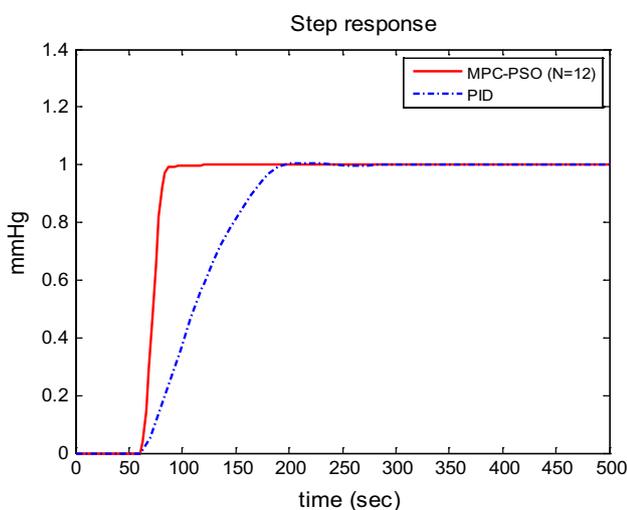


Fig. 9 The step response comparison between the MPC–PSO controller and the PID controller

Figures 7 and 8 show the outputs and control signals of the designed MPC–PSO control system with four different values of predicted horizon, $N = 7, 10, 12, 15$. For all the designed cases, the magnitudes of control signal are smaller than 2, which satisfy the constraints. With $N = 7$, both the output and control signal present some oscillations. Because of using largest control signal, the response of this system is faster than the others. However, the settling time is still significant. When N increases, the response gets slower but presenting less overshoots as observed in case of $N = 10, 12$ and 15. When the predicted horizon is $N = 12$, the overshoot is approximately zero while the settling time is smallest (about 100 s). At this point, the control performance reaches climax.

The comparison in Fig. 9 clearly proves that, the control performances of MPC–PSO controller (in case of $N = 12$) is much better than that of the optimized PID controllers in

de Moura Oliveira et al. (2014). It also should be noticed that, for all the simulated values of predicted horizon, the system always response faster than PID controllers. That is the main advantages of MPC principle. With the significant improvement in response, this control approach is promised for a wider range of patients.

7 Conclusions

The MPC–PSO approach has been used to design our MAP control system. The results showed that the substantial delay between drug infusion and change in blood pressure posed a real challenge for PID controller design in the MAP control system. In contrast, the MPC–PSO controller effectively handled the delay and the limitation of control signal. The simulation results clearly depicted the better performance of the MPC–PSO MAP control system compared to that of the conventional PID MAP control system. Further studies are needed to deal with the variation of the model parameters and the disturbance from environments. The MPC–PSO MAP controller will be also able to design as a chip through a FPGA.

References

- de Moura Oliveira PB, Durães J, Pires EJS (2014) Mean arterial pressure PID control using a PSO-BOIDS algorithm. In: *Advances in Intelligent Systems and Computing*, pp 91–99
- Ferreira A, Boston JR, Antaki JF (2009) A control system for rotary blood pumps based on suction detection. *IEEE Trans Biomed Eng* 56:656–665
- Furutani E, Araki M, Kan S, Aung T, Onodera H, Imamura M, Shirakami G, Maetani S (2004) An automatic control system of the blood pressure of patients under surgical operation. *Int J Control Autom Syst* 2:39–54
- Gao Y, Er MJ (2003) Adaptive fuzzy neural control of mean arterial pressure through Sodium Nitroprusside infusion. In: *IEEE 42nd Conf. on Decision and Control*, vol. 3, pp 2198–2203
- Hoeksel SAAP, Blom JA, Jansen JRC, Maessen JG, Schreuder JJ (2001) Computer control versus manual control of systemic hypertension during cardiac surgery. *Acta Anaesthesiol Scand* 45:553–557
- Holkar KS, Waghmare LM (2010) An overview of model predictive control. *Int J Control Autom* 3:47–63
- Hsu CH, Chang JB, Liu IC, Lau SC, Yu SM, Hsieh CH, Lin JD, Liang YJ, Pei D, Chen YL (2015) Mean arterial pressure is better at predicting future metabolic syndrome in the normotensive elderly: a prospective cohort study in Taiwan. *Prev Med* 72:76–82
- Jeong J, Portnof JE, Kalayeh M, Hardigan P (2016) Hypotensive anesthesia: comparing the effects of different drug combinations on mean arterial pressure, estimated blood loss, and surgery time in orthognathic surgery. *J Cranio-Maxillofac Surg* 44:854–858
- Kennedy J, Eberhart RC (1995) Particle swarm optimization. In: *Proc IEEE Int Conf on Neural Networks*, vol. 4, pp 1942–1948
- Kumar ML, Harikumar R, Vasank AK, Sudhama VK (2009) Fuzzy controller for automatic drug infusion in cardiac patients. In:

- Proc. of the International MultiConference of Engineers and Computer Scientists (IMECS 2009), vol. 1, pp 18–20
- Lin AY, Huang HN, Su YC, Shiu CY, Hwang JL (2008) Implementation of fuzzy controller for measuring instantaneous arterial blood pressures via tissue control method. *IET Control Theory Appl* 2:40–50
- Liu GZ, Zhang YT, Wang L (2009) A robust closed-loop control algorithm for mean arterial blood pressure regulation. In: 2009 Sixth International Workshop on Wearable and Implantable Body Sensor Networks, pp 77–89
- Luginbühl M, Bieniok C, Leibundgut D, Wymann R, Gentilini A, Schnider TW (2006) Closed-loop control of mean arterial blood pressure during surgery with alfentanil clinical evaluation of a novel model-based predictive controller. *J Am Soc Anesthesiol* 105:462–470
- Padmanabhan R, Meskin N, Haddad WM (2015) Closed-loop control of anesthesia and mean arterial pressure using reinforcement learning. *Biomed Signal Process Control* 22:54–64
- Sánchez-Durán JA, Oballe-Peinado Ó, Castellanos-Ramos J, Vidal-Verdú F (2012) Hysteresis correction of tactile sensor response with a generalized Prandtl–Ishlinskii model. *Microsyst Technol* 18:1127–1138
- Sekhon MS, Smielewski P, Bhate TD, Brasher PM, Foster D, Menon DK, Gupta AK, Czosnyka M, Henderson WR, Gin K, Wong G, Griesdale DE (2016) Using the relationship between brain tissue regional saturation of oxygen and mean arterial pressure to determine the optimal mean arterial pressure in patients following cardiac arrest: a pilot proof-of-concept study. *Resuscitation* 106:120–125
- Yang XS (2010) Nature-inspired metaheuristic algorithms. Luniver, Frome, United Kingdom
- Zhu KY, Zheng H, Zhang DG (2008) A computerized drug delivery control system for regulation of blood pressure. *Int J Intell Comput Med Sci Image Process* 2:1–13