A Particle Swarm Optimization Based Predictive Controller for Delay Compensation in Networked Control Systems

Abdin Yousif Elamin, Nurul Adilla Mohd Subha(✉), Norikhwan Hamzah, and Anita Ahmad

Faculty of Electrical Engineering, Universiti Teknologi Malaysia, 81310 Skudai, Johor, Malaysia
nuruladilla@utm.my

Abstract. This paper addresses transmission delays problem in network control systems. Network-induced delay is an inherent constraint in NCS implementation that could lead to system degradation and destabilization. A particle swarm optimization (PSO) tuning algorithm was adopted to optimally tune the parameters of Generalized Predictive Controller (GPC) to solve networked-induced delay problem. Furthermore, a modified PSO-GPC was designed by replacing the standard GPC objective function with an Integral Time Squared Error (ITSE) performance index in the GPC controller design. A particle swarm optimization based PI controller in the Smith predictor structure is designed to compare the performances of the original PSO-GPC and the modified PSO-GPC. The results show that the modified PSO-GPC performed better than the PSO-GPC in terms of transient response and enhanced NCS performance in the occurrence of network delays.

Keywords: Network control systems · Delay compensation · Particle swarm optimization · Generalized predictive control · Smith predictor

1 Introduction

NCS are started to be utilized in various industrial applications such as automotive [1], human surveillance [2], and process control [3] due to several clear advantages. When control system is designed using communication network, it is known as a networked control system (NCS). In NCS, the control signals are shared among the system's elements in form of information packets through wireless or hard wired connections. Utilization of networks communication allow for efficient centralized control systems with minimal wiring across the system, hence reducing initial cost of a system [4]. Furthermore, NCS is a flexible structure that allows the addition or reduction of system components without any significant change to the hardware of the system [5].

In general, there are two common problems with NCS application. NCS prone to problem of network delay and data dropout. These problems may lead to system performance degradation or even system destabilization. Current progress in data dropout management can be review in [5–7]. Network-induced delay occurred due to

M.S. Mohamed Ali et al. (Eds.): AsiaSim 2017, Part II, CCIS 752, pp. 417–431, 2017.
DOI: 10.1007/978-981-10-6502-6_37

number of factor, such as limited bandwidth, network congestion, and network transmission protocols. Short time delays delay effect on the NCS is studied using Markov model shows that network-induced delays can lead to uncontrollable or unobservable system [8, 9]. In case of network-induced delay, system stability depends on the upper and lower bounds of the time delay [10]. Several control methodologies have been formulated to compensate the negative effects of network delays in NCS based on different network configurations, constraints, and behaviors.

The event-based control methodology is introduced to control robotic manipulators over the Internet. For example, in [11, 12] the optimal stochastic control methodology is used, which treat the network delays as a Linear–Quadratic–Gaussian (LQG) problem. There is time-based methodology such as Model Predictive Control (MPC) which predict future plant output to compensate network delay problem. The main difference between event-based and time-based control methodology is that the event-based control treats a system motion as a system reference [8]. Other than that, robust control theory which doesn't need any prior knowledge about the network delays have been studied in [13]. Combination of these methodology has also been explored by other researchers [14–16].

There is a growing interest in MPC-based approach due to its ability to predict future plant outputs, hence effectively compensate the network induced constraints such has network delay and packets dropouts [17, 18] even within system with state and input constraints [19, 20]. In model predictive control, there is no unique method to determine the control algorithm, but rather a wide variety of methods to predict future plant outputs from current plant outputs along a specific prediction horizon at each sampling interval [21]. In a study conducted in [17], a novel generalized predictive control (GPC) algorithm is proposed to design the control signals which include the employment of buffer in order to compensate both the control-to-actuator (C-A) and sensor-to-controller (S-C) delays. In [18], to solve for random time delays and packet dropouts, the delays are modeled using Markov chains, a modified GPC algorithm is proposed and the stability analysis is established.

The study of MPC for nonlinear NCSs is more practical than linear NCSs. However nonlinear NCS is more technically challenging due to increased computational complexities. In literature, several promising results of networked nonlinear MPC have been proposed such as Lyapunov based MPC (LMPC) strategy which able to control a nonlinear system subjected to constraints [22] and LMPC strategy for nonlinear NCSs with time-varying network-induced delays [23].

Many studies have been conducted in MPC tuning, both heuristic and deterministic [24–26]. However, most previous research focused on improving performance for systems without consideration of communication networks-induced problems. A summary of tuning methods for GPC and Dynamical matrix control (DMC) based on the Integral Square Error (ISE) as a performance criterion is illustrated in [27]. Some methods suggest heuristics while others are based on stability criteria, closed loop analysis, analysis of variance [28], optimization-based algorithms [29]. Most studies agree on the influence of these parameters in improving the system performance but which parameter has the highest influence is still debatable. While some suggest that the weighting factor is the most significant parameter, others suggest the prediction horizon having the most effect on system performance [30]. In this paper, a particle swarm optimization (PSO) is adopted to optimally tune the parameters of the generalized predictive control (GPC) algorithm. PSO effects on NCS performance is investigated with focus on constant network-induced delay. The main aim of the algorithm is to compensate the network-induced delays through the generated output and control input prediction sequences.

This paper is structured as follows: Sect. 2 introduces the generalized predictive control algorithm and its parameters; Sect. 3 introduces the particle swarm optimization algorithm and its formulation to solve the GPC cost function; Sect. 4 presents results from MATLAB simulation; the conclusion is presented in Sect. 5.

2 Generalized Predictive Control in Delay Compensation

Characteristic of network delay depends on the network and transmission protocol which can be constant or time variant [31]. In this paper, Fig. 1 represents the basic structure of the considered NCS consisting of a sensor that sends information through a network to a controller, which then produced control signals to an actuator.

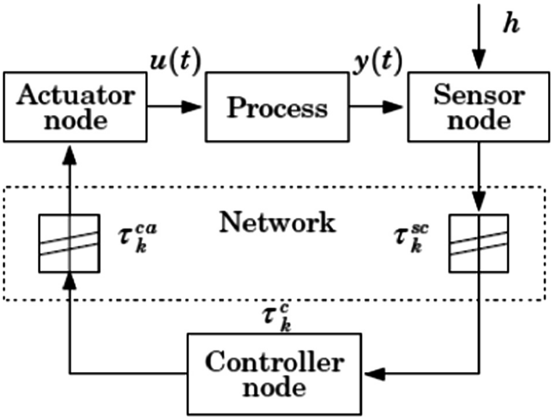

Fig. 1. A NCS with time delays

2.1 NCS with Network-Induced Delays

Network-induced delay exist in the controller to actuator channel (feed forward delay) and the sensor to controller channel (feedback delay), denoted by τ^{ca} and τ^{sc} respectively.

From Fig. 1, τ^{sc} is the first delay which represents the time taken to generate a control signal from the sensor. The controller to actuator delay τ^{ca} indicates the time taken for a control signal to reach the actuator. Another delay called the computation delay τ^{c} also exists in the system and is defined as the time taken to generate a control signal from the sensor feedback signal. However, for simplification the computation delay τ^{c} is considered to be embedded within the overall network-induced delays τ [7].

Assumption 1. The clock-driven sensor samples the plant outputs periodically at specific sampling instant T_S.

Assumption 2. The event-driven controller and actuator act as soon as the sensor data and control data become available.

Assumption 3. The network-induced delays in the system are time varying but bounded. Moreover, let $\overline{a_k^{max}}$, $\overline{\mathrm{b}_\mathrm{k}^{\mathrm{max}}}$, $\overline{a_k^{min}}$ and $\overline{b_k^{min}}$ symbolize the maximum and minimum delays in the feedforward channel and in the feedback channel. Number of sampling periods is a positive integer.

Remark 1. The delay considered in the feed forward and feedback channels have maximum and minimum value, therefore we can treat the network delays as dead time denoted by *d*.

Since the delays present in the system are time varying but bounded by maximum and minimum values,

$$d_m \leq d \leq d_M. \tag{1}$$

Where d_m is the minimum delay and d_M is the maximum delay which determined by the following equations,

$$d_m = \overline{a_k^{min}} + \overline{b_k^{min}}. \tag{2}$$

$$d_M = \overline{a_k^{max}} + \overline{b_k^{max}}. \tag{3}$$

Based on (2) and (3), the delay d is calculated using the mean value of the delays present in the communication network which is described by the following relation [23],

$$d = \frac{d_m + d_M}{2}. \tag{4}$$

2.2 Generalized Predictive Control Algorithm

Generalized predictive control uses the following CARIMA model to describe the controlled object

$$A(z^{-1})y(t) = z^{-d}B(z^{-1})u\,(t-1) + \frac{e\,(t)C(z^{-1})}{D\,(z^{-1})}. \tag{5}$$

Where (t), $u\,(t-1)$ and $e\,(t)$ are the plant output, control signal, and white noise with zero mean value. It is assumed that there is no disturbance acting on the system, therefore $e(t)$ will be zero. $A(z^{-1})$ and $B(z^{-1})$ are the plant polynomials while $C(z^{-1})$ and $D(z^{-1})$ are the disturbance polynomials having the following expressions,

$$A(z^{-1}) = 1 + a_1 z^{-1} + \cdots + a_{n_A} z^{-n_A}. \tag{6}$$

$$\mathrm{B(z^{-1}) = b_0 + b_1 z^{-1} + \cdots + b_{n_A} z^{-n_B}}. \tag{7}$$

$$C\,(z^{-1}) = 1. \tag{8}$$

$$\mathrm{D\,(z^{-1}) = 1 - z^{-1}}. \tag{9}$$

Where n_A and n_A represent the polynomial degrees.

The control algorithm for a generalized predictive controller (GPC) consists of two steps, first is the prediction model which predicts the future plant outputs, based on past and current input values. The prediction model has the form below

$$\begin{aligned} \hat{y}(t+N|T_S) = {} & G(z^{-1})D(z^{-1})z^{-d-1}u(t+N|T_S) + \\ & \frac{H(z^{-1})D(z^{-1})}{C(z^{-1})}u(t-1|T_S) + \frac{F(z^{-1})}{C(z^{-1})}y(t|T_S). \end{aligned} \tag{10}$$

Where N is the prediction horizon, $\hat{y}(t+N|T_S)$ are the predicted plant outputs a computed at time k and $u(t+N|T_S)$ are the future control signals computed at every sampling time T_S. To determine the polynomial $G(z^{-1})$, $H(z^{-1})$ and $F(z^{-1})$ two Diophantine equations are used,

$$\frac{C(z^{-1})}{A(z^{-1})D(z^{-1})} = E(z^{-1}) + z^{-(N-d)}\frac{F(z^{-1})}{A(z^{-1})D(z^{-1})}. \tag{11}$$

$$E(z^{-1})B(z^{-1}) = C(z^{-1})G(z^{-1}) + z^{-(N-d)}H(z^{-1}). \tag{12}$$

Where

$$E(z^{-1}) = 1 + e_1 z^{-1} + \cdots + a_{n_E} z^{-n_E}. \tag{13}$$

$$F(z^{-1}) = f_0 + f_1 z^{-1} + \cdots + f_{n_F} z^{-n_F}. \tag{14}$$

$$G(z^{-1}) = g_0 + g_1 z^{-1} + \cdots + g_{n_G} z^{-n_G}. \tag{15}$$

$$H(z^{-1}) = h_0 + h_1 z^{-1} + \cdots + h_{n_H} z^{-n_H}. \tag{16}$$

With

$$n_E = N - d - 1. \tag{17}$$

$$n_F = \max(n_A + n_D - 1, n_c - (N - d)). \tag{18}$$

$$n_G = N - d - 1. \tag{19}$$

$$n_H = \max(n_C, n_B + d) - 1. \tag{20}$$

After determining the values of the previous polynomials and collecting the N step predictions, the prediction model can be written in a matrix notation as,

$$\hat{y} = Gu_d + \hat{y}_0. \tag{21}$$

Where $\hat{y}$ is predicted plant output vector, u_d is the vector of future control sequences, G is the system dynamic matrix and $\hat{y}_0$ represents the predicted free response vector.

$$\hat{y} = [\hat{y}(t+d+1|T_S) \quad \hat{y}(t+d+2|T_S) \ldots \hat{y}(t+d+N|T_S)]^T. \tag{22}$$

$$u_d = [\Delta u(t|T_S) \quad \Delta u(t+1|T_S) \ldots \Delta u(k+N-1|T_S)]^T. \tag{23}$$

$$\hat{y}_0 = [\hat{y}_0(t+d+1|T_S) \quad \hat{y}_0(t+d+1|T_S) \ldots \hat{y}_0(t+d+1|T_S)]^T. \tag{24}$$

$$G = \begin{bmatrix} g_0 & 0 & \ldots & 0 \\ g_1 & g_0 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ g_{N-1} & g_{N-2} & \cdots & g_0 \end{bmatrix}. \tag{25}$$

The second step in the GPC controller design consists on determining the optimal control sequence. The optimizer calculates these signals by taking into consideration the objective function *J*. The objective function is based on the minimization of both the controller output and tracking error, the control weighting factor λ is introduced to make a trade-off between these objectives.

$$J = (Gu_d + \hat{y}_0 - w)^T (Gu_d + \hat{y}_0 - w) + \lambda u_d^T u_d. \tag{26}$$

Where w is the reference trajectory vector, by minimizing the above objective function $\left(\frac{\partial J}{\partial u_d} = 0\right)$ the optimal control signal is expressed as,

$$u_d^* = \left(G^T G + \lambda I\right)^{-1} G^T [w - \hat{y}_0]. \tag{27}$$

The control algorithm is calculated in a recursive (off-line) manner, which has the advantage of very fast computation.

2.3 GPC Parameter Tuning

The objective function from (26) can be rewritten as

$$j = \sum\nolimits_{k=1}^{N} (\hat{y}(t+k) - w(t+k)) + \lambda \sum\nolimits_{k=1}^{N_u} \Delta \mathrm{u}(\mathrm{t}+\mathrm{k}-1). \tag{28}$$

Equation (28) shows there are three parameters that affect the control signal generated in (27); N, N_U, and λ. Prediction horizon N and control horizon N_U are related to each other. Prediction horizon is selected early in the controller design and then holds it constant while tuning other controller settings. The control horizon is used to reduce computational processes by minimizing computational variables at each control interval. The value of the control horizon is in between 1 and the value of the prediction horizon. To deal with network-induced delays, its recommended that the difference between the prediction horizon and the control horizon must be larger than delay introduced by the communication network at each sampling instant, which is denoted by d [32]. The control weighting factor λ is present in (26) to restrict the control signal. The aggressiveness of the control signal is inversely proportional to λ. If λ has a small value, the controller will minimize the error between the reference and the output, forgetting about the control effort. Thus, the response of the system will tend to be faster but this might also result in an increase in overshoot and response oscillation.

3 Particle Swarm Optimization Based Approach for GPC Tuning

In this section, an overview on particle swarm optimization will be presented. The particle swarm optimization algorithm will then be implemented to tune the parameters of the GPC controller based on two objective functions. Finally, the PSO-GPC controller performance will be compared with the performance of a Smith predictor-PI controller in which the PSO will be applied to tune K_p and K_i.

3.1 Particle Swarm Optimization

The term optimization refers to the process of selecting the best element from a group of alternative elements based on a defined goal called the objective function. Mathematically, this is achieved by finding the values of the parameters that will maximize or minimize the objective function. Solving optimization problems analytically is quite tedious because the objective function might be non-linear, multidimensional, constrained or have many local optimums. Heuristic optimization method is an efficient alternative to solve the problem [29–31].

Particle swarm optimization is meta-heuristic optimization method developed by Kennedy and Eberhart to imitate the seeking behavior in bird flocks or fish schools [33]. In PSO, the solution for the optimization problem is represented by a vector called a particle which contains a set of parameters and every particle has the same number of parameters. Initially, a population containing a number of particles is initialized with random parameters and then enters an iterative process to search for the optimum solution. After each iteration, particles are compared and evaluated by substituting the values of their parameters in the objective function. In one iteration it might occur that one of the particles comes out with the best solution, called the globally best solution. In the succeeding iteration, another particle might be the globally best solution. Hence, at the end of each iteration, a velocity estimate for each particle is calculated based on its best solution called the personal best and the globally best. Furthermore, the velocity is used to update the particle following these equations,

$$v_i^{K+1} = Wv_i^k + C_1R_1\left(P_i - x_i^k\right) + C_2R_2\left(P_g - x_i^k\right). \tag{29}$$

$$x_i^{k+1} = x_i^k + v_i^{K+1}. \tag{30}$$

Where v_i^k represents the current velocity for particle i at iteration k while v_i^{K+1} is the velocity at iteration $K+1$. x_i^k represents the current position for particle i at iteration k while x_i^{k+1} is the updated position at iteration $K+1$. C_1 and C_2 are the cognitive coefficient and social coefficient which help modulate the steps taken by a particle in the direction of its personal best and global best. R_1 and R_2 are random values between 0 and 1. P_i represents the personal best of the particle i, P_g is the global best and W is the inertia weighting coefficient. As iterations continue, the particles are updated and they all move from different directions towards the global best which results in the best solution.

3.2 PSO Based Generalized Predictive Control Design

As mentioned in the previous section, the main parts of the GPC are (10) and (26). Other equations are used to get the optimal control input in (27) to minimize the objective function found in (26). However, the objective function relies on three GPC parameters which are the prediction horizon N, control horizon $\mathrm{N_U}$ and the weighting factor λ. Hence, a particle swarm optimization method can be implemented to tune these parameters and minimize the objective function as shown in Fig. 2. The particles are represented by $P = [N, N_u, \lambda]^T$ and formulated as follows:

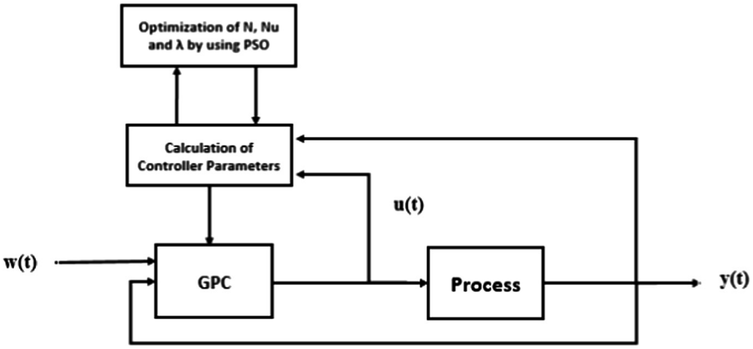
Optimization of N, Nu and λ by using PSO
Calculation of Controller Parameters
u(t)
w(t)
GPC
Process
y(t)

Fig. 2. PSO-GPC block diagram

(1) Identify the number of particles, the upper and the lower boundary of each tunable parameter, number of iterations and search parameters: cognitive coefficient (C_1), social coefficient (C_2) and inertia weighting coefficient (*W*).
(2) The particle position and velocity are initialized randomly.
(3) Simulate the GPC with the tuning parameters (N, N_U and λ) for each particle.
(4) Evaluate the objective function for each particle.
(5) Update, if any, the personal best Pi and global best Pg.
(6) Update the particle positions and velocities with values of step 5.
(7) Repeat step 3 to 6 until the last iteration count or the desired precision is achieved. The particle that produces the latest global best is the optimal value.

Equation 26 reveals the objective function as a summation of two terms. The first term is $\sum_{k=1}^{N}(\hat{y}(t+k)-w(t+k))$ which ensures fast transient response, settling time, and minimizes overshoot. The other term is $\lambda\sum_{k=1}^{N_u}\Delta \mathrm{u}(\mathrm{t}+\mathrm{k}-1)$ that prevents the control signal from increasing indefinitely as it can lead to actuator saturation. Furthermore, to achieve better performance and tracking accuracy, the objective function in (28) will be replaced by one of the time domain integral performance indices called the Integral Time Square Error (ITSE) which will be solved by the particle swarm optimization.

$$e(t)=w(t)-y(t). \tag{31}$$

$$ISE=\int_0^{t_{ss}} e(t)^2 dt. \tag{32}$$

$$IAE=\int_0^{t_{ss}} |e(t)| dt. \tag{33}$$

$$ITSE=\int_0^{t_{ss}} te(t)^2 dt. \tag{34}$$

Where $e(t)$ is the error signal and t_{ss} is the time it takes to reach steady state. The goal behind this replacement is to formulate the tuning selection to account for the time domain performance goals such as settling time, rise time, and overshoot. The control systems based on these indices has fast response speed, large oscillation, relatively weak stability. In addition, control system based on ITSE force the error to be small at future instants with minimal oscillation compared to other performance indices such as Integral Absolute Error (IAE) and Integral Squared Error (ISE) [34].

3.3 PSO Implemented on a Smith Predictor Controller

The Smith predictor is one of the most used control strategies for time delay compensation. Smith predictor introduces an internal feedback into a controller to counterbalance the effects of delays on the main controller. Figure 3 illustrates the basic structure of a smith predictor controller in NCS [35].

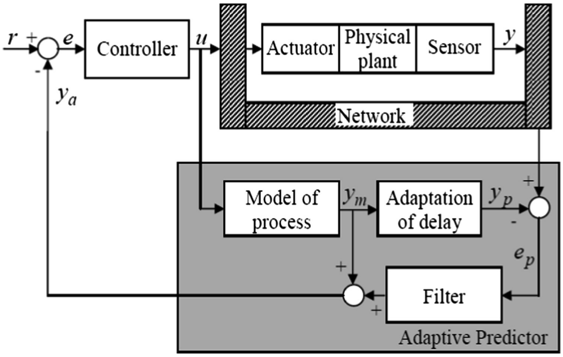

Fig. 3. Smith predictor structure in NCS

In Fig. 2, *u, y,* y_m, y_p and e_p are the control signal, actual output, predicted output, estimated output and output error, respectively. The controller shown above is selected to be a PI controller, which is described by the discrete form,

$$G_{PI}(z) = \frac{K_p T_s z}{K_i z - K_i}. \tag{35}$$

K_p and K_i represent the proportional gain and the integral gain respectively and are the controller parameters that will be tuned offline with the particle swarm optimization algorithm. The Integral Square Error (ISE) is used as the objective function to ensure the error signal approach zero while achieving faster transient response and minimum overshoot.

4 Results and Discussion

Simulation is carried out using MATLAB/Simulink 2016a, and it is assumed that both feedback and feedforward channel exhibit constant network-induced delays of 2 s $(\tau^{ca} = \tau^{sc} = 2)$. The controlled plant is a liquid level tank with the following discrete transfer function,

$$G(z^{-1}) = \frac{0.001703z^{-1} + 0.005419z^{-2}}{1 - 0.9718z^{-1} - 0.025z^{-2}} = \frac{A(z^{-1})}{B(z^{-1})}$$

The particle swarm optimization method is then used to tune the parameters of the GPC over a predefined search region such that the objective functions in (28) is minimized. The upper and lower value of the search regions are specified as listed in Table 1. The lower values of the prediction horizon and control horizon are selected based on the works in [24] to address the network-induced delays. The upper bound of the prediction horizon and control horizon are set at 40 and 10 respectively to minimize computational time. The lower search region for the Smith Predictor parameters are

Table 1. Search region of the tunable parameters

Parameters	Lower values	Upper values
Prediction horizon (N)	30	40
Control horizon (N_U)	5	10
Weighting factor (λ)	0.001	1
Proportional gain (K_P)	0.0001	200
Integral gain (K_I)	0.0001	200

chosen to be 0.0001 for both, while the upper values have a value of 200 to prevent destabilizing the plant by introducing high value gains. In PSO, the number of particles crucial in ensuring accurate results. The number of particles is set to 20 based on the work in [33] which illustrate that a suitable number of particles is in between 20 to 50. The number of iterations and the PSO search parameters used in the simulation is presented in Table 2.

Table 2. Parameters of the PSO algorithm

Parameters	Values
Particle dimensions	3
Number of particles	20
Number of iterations	30
Cognitive coefficient (C_1)	1.5
Social coefficient (C_2)	1.5
Inertia weighting coefficient (w)	.4–.9

To compare with GPC, the PSO procedure is repeated on a Smith predictor to obtain the optimal parameter values as tabulated in Table 3. In Table 4, the control performance in terms of transient response of each controller is presented. It is clear that the PSO-Smith predictor outperformed the PSO-GPC in terms of transient response.

Table 3. Tuning parameters obtained through PSO

Controllers	Prediction horizon (N)	Control horizon (N_U)	Weighting factor (λ)	Proportional gain (K_P)	Integral gain (K_I)
PSO-GPC	40	6	0.1329	–	–
Modified PSO-GPC	40	9	0.0094	–	–
PSO-Smith predictor	–	–	–	93.75	3.906

Table 4. Transient response analysis for step input signal

Controllers	Rise time(s)	Settling time (s)	Overshoot (%)
PSO-GPC	2.038	11.171	2.6%
Modified PSO-GPC	0.458	4.9	–
PSO-Smith predictor	0.251	4.56	–

In Fig. 4, it is obvious that the PSO-Smith predictor produces larger undershoot compared to the PSO-GPC when the water level is reduced. This indicates that the PSO-GPC is more efficient in dealing with non-minimum phase systems compared to PSO-Smith Predictor. However, the PSO-GPC resulted with a small overshoot and slower transient response toward the set point. Thus, a modified PSO-GPC is proposed through the replacement of the cost function in (28) by the ITSE objective function shown in (34). With a modified PSO-GPC, faster settling time and rise time can be achieved. Plus, GPC algorithm also allows for incorporation of output constraint in the optimization to tackle the overshoot problem. This is a clear advantage of GPC compared to the other control algorithm.

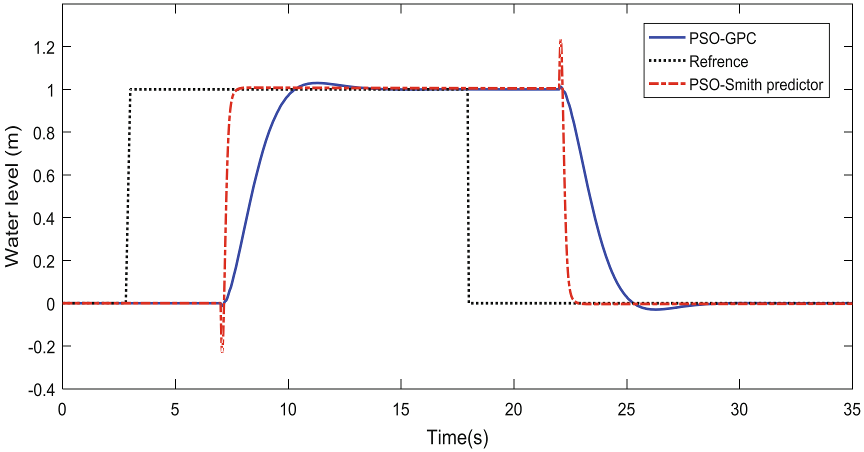

Fig. 4. Comparison of closed-loop step response

From Fig. 5, it can be seen that the modified PSO-GPC produced a better transient response than the original PSO-GPC. When compared with PSO-Smith predictor, the modified controller was slightly outperformed in terms of settling time and rise time as illustrated in Table 4, but produced a smaller undershoot. Thus, it can be concluded that GPC is a preferable controller in NCS applications because it is capable to deal with non-minimum systems, ability to handle constraints and its clear potential for other types of network-induced delays such as random time delay.

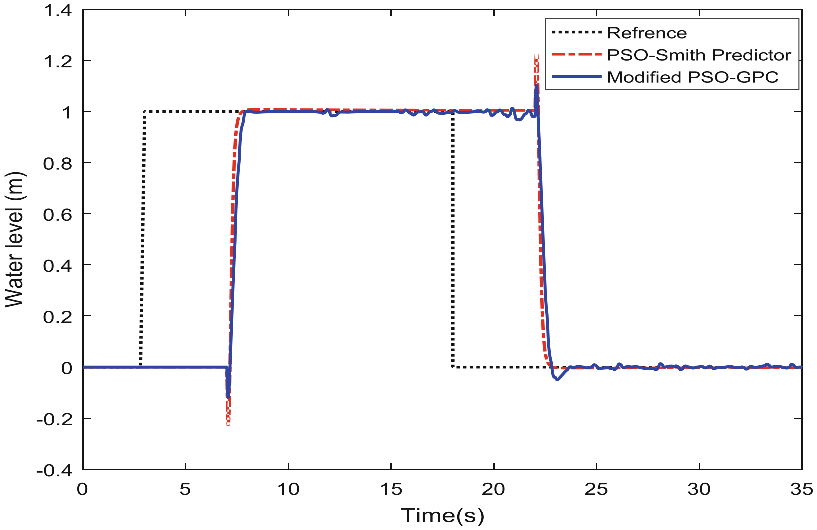

Fig. 5. Modified PSO-GPC closed-loop step response

5 Conclusion

In this paper, a standard GPC objective function and an ITSE performance index were applied in the generalized predictive controller design by using a particle swarm optimization to compensate the network-induced delays occurring between the components of a networked control system. The aim of the particle swarm optimization is to optimally tune the GPC parameters based on the above objective functions. The simulations result in MATLAB/SIMULINK show significant improvement of the controllers using the proposed techniques compared to PSO-Smith predictor. Although both predictive controllers can compensate the effects of network delays, the modified PSO-GPC, which is based on the ITSE performance index, achieved better transient response when compared with the original PSO-GPC.

References

1. Zhang, L., Gao, H., Kaynak, O.: Network-induced constraints in networked control systems-a survey. IEEE Trans. Ind. Informat. **9**(1), 403–416 (2013)
2. Kaneko, S.-I., Capi, G.: Human-robot communication for surveillance of elderly people in remote distance. IERI Procedia **10**, 92–97 (2014)
3. Dasgupta, S., Halder, K., Banerjee, S., Gupta, A.: Stability of Networked Control System (NCS) with discrete time-driven PID controllers. Control Eng. Practice **42**, 41–49 (2015)
4. Yang, S.-H.: Internet-Based Control Systems. Springer, Heidelberg (2011)
5. Saif, A.W.A., AL-Shihri, A.: Robust design of dynamic controller for nonlinear networked control systems with packet dropout. In: 2015 IEEE International Conference on Control System, Computing and Engineering (ICCSCE), Penang, pp. 557–562 (2015)

6. Elia, N., Eisenbeis, J.: Limitations of linear control over packet drop networks. IEEE Trans. Autom. Control **56**(4), 826–841 (2011)
7. Dong, X., Zhang, D.: H» Fault detection observer design for networked control systems with packet dropout using delta operator. In: 2016 12th World Congress on Intelligent Control and Automation (WCICA), Guilin, pp. 2547–2551 (2016)
8. Zhu, Q.X.: Controllability of multi-rate networked control systems with short time delay. In: Advanced Materials Research, pp. 219–220 (2011)
9. Zhu, Q.X.: Observability of multi-rate networked control systems with short time delay. Commun. Comput. Inf. Sci. **144**, 396–401 (2011)
10. Zhang, H., Zhang, Z., Wang, Z., Shan, Q.: New results on stability and stabilization of networked control systems with short time-varying delay. IEEE Trans. Cybern. **46**(12), 2772–2781 (2016)
11. Mahmoud, M.S., Sabih, M., Elshafei, M.: Event-triggered output feedback control for distributed networked systems. ISA Trans. **60**, 294–302 (2016)
12. Wu, J., Zhou, Z.-J., Zhan, X.-S., Yan, H.-C., Ge, M.-F.: Optimal modified tracking performance for MIMO networked control systems with communication constraints. ISA Trans. **68**, 14–21 (2017)
13. Feng, Y., Luo, J.: Robust H8 control for networked switched singular control systems with time-delays and uncertainties. In: Seventh International Symposium on Computational Intelligence and Design, Hangzhou, pp. 457–460 (2014)
14. Niu, Y., Liang, Y., Yang, H.: Event-triggered robust control and dynamic scheduling co-design for networked control system. In: 2017 14th International Bhurban Conference on Applied Sciences and Technology (IBCAST), Islamabad, pp. 237–243 (2017)
15. Zhang, Z., Zhang, H., Wang, Z., Feng, J.: Optimal robust non-fragile H∞ control for networked control systems with uncertain time-delays. In: Proceeding of the 11th World Congress on Intelligent Control and Automation, Shenyang, pp. 4076–4081 (2014)
16. Zhang, L., Wang, J., Ge, Y., Wang, B.: Robust distributed model predictive control for uncertain networked control systems. IET Control Theory Appl. **8**(17), 1843–1851 (2014)
17. Tang, P.L., de Silva, C.W.: Compensation for transmission delays in an ethernet- based control network using variable-horizon predictive control. IEEE Trans. Control Syst. Technol. **14**(4), 707–718 (2006)
18. Yu, B., Shi, Y., Huang, J.: Modified generalized predictive control of networked systems with application to a hydraulic position control system. J. Dyn. Syst. Meas. Control **133**(3) (2011)
19. Song, H., Yu, L., Zhang, W.A.: Stabilisation of networked control systems with communication constraints and multiple distributed transmission delays. IET Control Theory Appl. **3**(10), 1307–1316 (2009)
20. Li, H., Shi, Y.: Network-based predictive control for constrained nonlinear systems with two-channel packet dropouts. IEEE Trans. Ind. Electron. **61**(3), 1574–1582 (2014)
21. Yang, L., Yang, S.H.: Multi-rate control in internet-based control systems. IEEE Trans. Syst. Man Cybern. **37**(2), 185–192 (2007)
22. Das, B., Mhaskar, P.: Lyapunov-based offset-free model predictive control of nonlinear systems. In: 2014 American Control Conference, Portland, pp. 2839–2844 (2014)
23. Liu, J., Muñoz de la Peña, D., Christofides, P.D., Davis, J.F.: Lyapunov-based model predictive control of nonlinear systems subject to time-varying measurement delays. Int. J. Adaptive Control Sig. Process. **23**(8), 788–807 (2009)
24. Bunin, G.A., Fraire, F., François, G., Bonvin, D.: Run-to-Run MPC tuning via gradient descent. Comput. Aided Chem. Eng. **30**, 927–931 (2012)

25. Zermani, M.A., Feki, E., Mami, A.: Application of genetic algorithms in identification and control of a new system humidification inside a newborn incubator. In: 2011 International Conference on Communications, Computing and Control Applications (CCCA), Hammamet, pp. 1–6 (2011)
26. van der Lee, J.H., Svrcek, W.Y., Young, B.R.: A tuning algorithm for model predictive controllers based on genetic algorithms and fuzzy decision making. ISA Trans. **47**(1), 53–59 (2008)
27. Yamuna Rani, K., Unbehauen, H.: Study of predictive control tuning methods. Automatica **33**, 2243–2248 (1997)
28. Neshasteriz, R., Khaki-Sedigh, A., Sadjadian, H.: An analysis of variance approach to tuning of generalized predictive controllers for second order plus dead time models. In: IEEE ICCA 2010, Xiamen, pp. 1059–1064 (2010)
29. Aicha, F.B., Bouani, F., Ksouri, M.: Automatic tuning of GPC synthesis parameters based on multi-objective optimization. In: 2010 XIth International Workshop on Symbolic and Numerical Methods, Modeling and Applications to Circuit Design (SM2ACD), Gammath, pp. 1–5 (2010)
30. Rossiter, J.A.: Model-Based Predictive Control: A Practical Approach. CRC Press, Boca Raton (2003)
31. Zhang, X.M., Han, Q.L., Wang, Y.L.: A brief survey of recent results on control and filtering for networked systems. In: 2016 12th World Congress on Intelligent Control and Automation (WCICA), Guilin, pp. 64–69 (2016)
32. Camacho, E., Bordons, C.: Model Predictive Control Book. Springer, Heidelberg (2007)
33. Poli, R., Kennedy, J., Blackwell, T.: Particle swarm optimization an overview. Swarm Intell. **1**, 33–57 (2007). Springer
34. Pan, F., Liu, L.: Research on different integral performance indices applied on fractional-order systems. In: 2016 Chinese Control and Decision Conference (CCDC), Yinchuan, pp. 324–328 (2016)
35. Witrant, E., Georges, D., Canudas-de-Wit, C., Alamir, M.: On the use of state predictors in networked control systems. In: Chiasson, J., Loiseau, J.J. (eds.) Applications of Time Delay Systems. LNCIS, vol. 352, pp. 17–35. Springer, Heidelberg (2007). doi:10.1007/978-3-540-49556-7_2