International Journal of Automation and Computing 12(6), December 2015, 648-656
DOI: 10.1007/s11633-015-0897-7

Time-delay Compensation Method for Networked Control System Based on Time-delay Prediction and Implicit PIGPC

Zhong-Da Tian[1] Xian-Wen Gao[2] Bi-Lian Gong[2] Tong Shi[3]

[1]School of Information Science and Engineering, Shenyang University of Technology, Shenyang 110870, China

[2]School of Information Science and Engineering, Northeastern University, Shenyang 110819, China

[3]Department of Humanities, Liaoning Forestry Vocation-Technical College, Shenyang 110101, China

Abstract: A network time-delay compensation method based on time-delay prediction and implicit proportional-integral-based generalized predictive controller (PIGPC) is proposed. The least squares support vector machine (LSSVM) is used to predict the current time-delay, the parameters of the least squares support vector machine are optimized by particle swarm optimization (PSO) algorithm, and the predicted time-delay is used instead of the actual time-delay as the parameters of the network time-delay compensation controller. In order to improve the compensation effect of implicit generalized predictive controller (GPC), this paper puts forward an implicit generalized predictive control algorithm with proportional-integral-based (PI) structure and designs the controller based on implicit PIGPC. Through the simulation results, the effectiveness of this design in the paper is verified.

Keywords: Networked control system, time-delay compensation, particle swarm optimization (PSO), least squares support vector machine (LSSVM), implicit proportional-integral-based generalized predictive controller (PIGPC).

1 Introduction

Networked control system (NCS) is a fully distributed and networked real-time feedback control system between sensors, actuators and controller signals transmitted over the network[1]. The communication network in the feedback control loop introduces a significant additional delay, either constant or time varying, that makes the analysis and control design more complex[2, 3]. Many scholars have conducted a lot of research work on network induced time-delay compensation problem. Robust control[4, 5], intelligent control[6,7], predictive control[8−10] and other time-delay compensation methods have made a lot of developments and applications.

Generalized predictive control (GPC) was proposed by Clarke in 1987. GPC has many advantages: The first is based on the parameters of the model. Secondly, it retains the advantage of adaptive control, and has better robustness than adaptive control. Finally, due to the adoption of multi-step prediction, rolling optimization and feedback correction strategy, the control effect is more suitable for industrial process control requirements. Autoregression model is utilized to predict the time-delay and improved GPC algorithm for time-delay compensation was proposed[11], but details of the simulation parameters were not given. Based on a random time-varying network delay, GPC network based on a state space model is used to access an effective tracking performance[12]. The work in [13] gives a predictive control method for forward channel and feedback channel with random time-delay, and gives the closed-loop predictive control system stabilizing conditions. However, GPC algorithm requires solving Diophantine equations and matrix inversion operation, which lead to excessive online calculation. On the other hand, since the time-delay is random, the parameter selection for GPC algorithm requires a larger prediction step size, thereby increasing the calculation time.

With the development of network technology, wireless communication networks become more sophisticated. And in the industrial field data acquisition, industrial control applications and other areas gradually deepened. In industrial control, communication network based on the wireless network is also called "wireless NCS". Because wireless NCS reduces the network connection, system design has finer flexibility, and maintenance is more convenient. However, opposed to wired networks, wireless networks are highly unreliable, with lesser bandwidth, and greater transmission delay and packet loss characteristics. Although there is much research on wired NCS time-delay compensation method, which has made great achievements, there is little research for wireless NCS. So the study of the wireless NCS time-delay compensation method has significant theoretical and practical significance.

In summary, a network time-delay compensation method

Regular paper
Manuscript received January 8, 2014; accepted December 4, 2014
This work was supported by National Natural Science Foundation of China (No. 61034005) and Liaoning Province Doctor Startup Fund (No. 20141070)
Recommended by Associate Editor Hong-Nian Yu

based on time-delay prediction and implicit proportional-integral-based generalized predictive controller (PIGPC) is proposed. Firstly, particle swarm optimization (PSO) is used to optimize least squares support vector machine (LSSVM) model to predict time-delay, an implicit GPC algorithm with proportional-integral-based (PI) structure based on the predicted time-delay is proposed to compensate network time-delay. Because recursive solving Diophantine equations is not needed, implicit PIGPC can save online calculation time, and achieve better control effect by the combination of PI feedback structure and prediction function of GPC. Meanwhile, in order to reflect the effectiveness of method in this paper, IEEE 802.11b wireless network is selected for the simulation. Compared to wired NCS, time-delay of wireless NCS has more randomness and uncertainty, which will demonstrate the effectiveness of compensation method in the paper.

2 Time-delay prediction based on PSO-LSSVM

The time-delay of sensor to the controller, and time-delay of controller to the actuator present different variation rules, and different impact to the system performance. Under normal circumstances at the time k, the sensor to the controller time-delay $\tau_{sc}(k)$ can be measured, and the controller to actuator time-delay $\tau_{ca}(k)$ and controller computing time $\tau_c(k)$ are unknown, but $\tau_{ca}(k)$ and $\tau_c(k)$ before moment k is known, so the historical time-delay can be used to predict the current network time-delay. In the NCS discussed in this paper, controller and actuator are event-driven, whereas sensor is clock-driven. Therefore, the forward and feedback channel time-delay of NCS can be combined[11,14], the system total delay can be expressed as $\tau(k) = \tau_{ca}(k) + \tau_{sc}(k) + \tau_c(k)$. Because the time-delay can be combined, the total system time-delay can be predicted through the prediction model.

2.1 LSSVM algorithm

Assuming the clock is synchronous in networked control system, which can be implemented by regular high-priority synchronization signal. The time-delay of an NCS can be measured and defined as the following sample time-delay vector:

$$D_i = [d_i, d_{i-1}, \cdots, d_{i-m}], \quad i = 1, 2, \cdots, N \tag{1}$$

where $\{d_i, i = 1, 2, \cdots, N\}$ is network time-delay sequence, N is the length of the time-delay sequence.

Based on LSSVM algorithm, the sample space is mapped into a high-dimensional or even infinite dimensional feature space through a nonlinear mapping $\varphi(\cdot)$[15]. In this feature space, there is $y(x) = w\varphi(x) + b$, where w is the weight coefficient vector, b is the constant bias. Optimal w and b can be obtained by minimizing the objective function.

$$\min_{w,b,e} J(w, e) = \frac{1}{2}w^{\mathrm{T}}w + \frac{1}{2}\gamma\sum_{k=1}^{N} {e_k}^2. \tag{2}$$

Lagrange function is established for solving constrained optimization problems mentioned above:

$$L(w, b, e; \alpha) = J(w, e) - \sum_{k=1}^{N} \alpha_k\{w^{\mathrm{T}}\varphi(D_k) + b + e_k - y_k\} \tag{3}$$

where a_k is a Lagrange multiplier. The Lagrange function is used to obtain extreme value, and the above optimization problem is transformed into solving the linear equations.

According to Mercer conditions, the presence of mapping function $\varphi(\cdot)$ and kernel function $K(\cdot)$ satisfy the following equation:

$$K(D_i, D_j) = \varphi(D_i)^{\mathrm{T}}\varphi(D_j). \tag{4}$$

Nonlinear mapping ability of LSSVM prediction model is determined by the kernel function. Kernel function is used for LSSVM prediction model samples mapping from input space to feature space. Therefore, different kernel function has different learning ability and generalization ability. The most widely used is radial basis function (RBF) kernel, which is applicable for low-dimensional, high-dimensional, small sample, large sample, etc. Because of wide domain of convergence, RBF kernel function is an ideal kernel[16,17]. This paper chooses RBF as the kernel function.

$$K(D_i, D_j) = \mathrm{e}^{\frac{-\|D_i - D_j\|^2}{2\sigma^2}} \tag{5}$$

where σ^2 is the width of RBF kernel function. Predicted value of time-delay at time $k+1$ can be obtained from the equation

$$\widehat{d}_{k+1} = \sum_{i=1}^{N} \alpha_i K(D_i, D_k) + b. \tag{6}$$

Assuming the input time-delay sequence is D, D is transformed into the matrix form as

$$X = \begin{bmatrix} d_1 & d_2 & d_3 & \cdots & d_m \\ d_2 & d_3 & d_4 & \cdots & d_{m+1} \\ \vdots & \cdots & \vdots & \cdots & \vdots \\ d_{N-m} & d_{N-m+1} & \cdots & \cdots & d_{N-1} \end{bmatrix} \tag{7}$$

where d_t represents a time-delay value at time t, N is the total length of the input sequence of time-delay, m is the embedding dimension, and p is the number of prediction steps. Therefore, the output matrix Y can be expressed as

$$Y = \begin{bmatrix} d_{m+1} & \cdots & d_{m+p} \\ d_{m+2} & \cdots & d_{m+p+1} \\ \vdots & \cdots & \vdots \\ d_N & \cdots & d_{m+p+N-1} \end{bmatrix}. \tag{8}$$

NCS time-delay prediction method based on LSSVM can be expressed as the following steps:

Step 1. Build an NCS, measure N groups of time-delay sample data, select the appropriate prediction parameters γ, σ^2 and m.

Step 2. Transform the time-delay to the input and output vectors in accordance with (7) and (8).

Step 3. The RBF is selected as the kernel function, regression coefficients a_k and the offset value b are calculated.

Step 4. Calculate column vector $K(D_k, D_i)$, calculate the next p time time-delay prediction value through (6).

The predictive effect of LSSVM has a great relationship with regularization parameter γ and the width of radial basis kernel function σ^2. Therefore, how to select these parameters is very important. This paper selects the particle swarm optimization algorithm to optimize parameters of least squares support vector machine.

2.2 PSO algorithm

PSO algorithm is a parallel search algorithm based on population[18]. It has simple concept and implementation, and faster convergence for the treatment of high-dimensional problems.

In a D-dimension target searched space, consisting of N particles to form a population, the i-th particle is represented as vectors with D-dimension, which can be written as

$$X_i = (x_{i1}, x_{i2}, \cdots, x_{iD}), \quad i = 1, 2, \cdots, N. \tag{9}$$

The i-th particle velocity is also a D-dimensional vector, which can be written as

$$V_{\rm i} = (v_{i1}, v_{i2}, \cdots, v_{iD}), \quad i = 1, 2, \cdots, N. \tag{10}$$

The best search position of the i-th particle at present is called $p_{\rm best}$, which can be written as

$$p_{\rm best} = (p_{i1}, p_{i2}, \cdots, p_{iD}), \quad i = 1, 2, \cdots, N. \tag{11}$$

The optimal position of the whole particle swarm so far to search called $g_{\rm best}$, can be written as

$$g_{\rm best} = (g_{i1}, g_{i2}, \cdots, g_{iD}), \quad i = 1, 2, \cdots, N. \tag{12}$$

When these two optimal values are found, particles can update the velocity and position according to (13) and (14):

$$\begin{aligned} v_{id}^{k+1} = \omega \times v_{id}^k + c_1 \times rand_1^k \times (p_{{\rm best}_{id}}^k - x_{id}^k) + \\ c_2 \times rand_2^k \times (g_{{\rm best}_{id}}^k - x_{id}^k) \end{aligned} \tag{13}$$

$$x_{id}^{k+1} = x_{id}^k + v_{id}^{k+1}. \tag{14}$$

2.3 LSSVM optimization by PSO algorithm

Regularization parameter γ, radial basis kernel function width σ^2 and delay sequence embedding dimension m impact LSSVM prediction accuracy seriously. These 3 parameters are considered as particles of PSO algorithm. The optimization steps can be described as follows:

Step 1. Initialize parameters of PSO algorithm, including inertia weight, learning factor, population size, the maximum number of iterations, etc.

Step 2. Use the corresponding training samples of each particle and predict by LSSVM, the mean square error (MSE) of predicted value and the actual value of time-delay is chosen as the fitness of each particle. Each particle′s fitness and the current optimal particle′s fitness value is compared. If the current particle is better, then choose the particle current position as the optimal location of the particles.

Step 3. Each particle′s best position of fitness value of optimal particle location and swarm fitness value are compared. If the particle is better, the best position of the particles is selected as the best position of the swarm.

Step 4. Use (13) and (14) to update the particle velocity and position.

Step 5. Check whether the optimal end conditions are met. If satisfied, then end the optimizing process and find optimal solutions. Otherwise, go to Step 2, continue to search for a new round.

3 Implicit PIGPC time-delay compensation controller

3.1 Implicit GPC control algorithm

Controlled autoregressive integrated moving average (CARIMA) model is used to represent a real process with non stationary noise.

$$A(q^{-1})y(t) = B(q^{-1})u(t-1) + \frac{C(q^{-1})\varepsilon(t)}{\Delta} \tag{15}$$

with

$$\begin{aligned} A(q^{-1}) &= 1 + a_1 q^{-1} + \cdots + a_{na} q^{-n} \\ B(q^{-1}) &= b_0 + b_1 q^{-1} + \cdots + b_{nb} q^{-n} \\ C(q^{-1}) &= c_0 + c_1 q^{-1} + \cdots + c_{nc} q^{-n} \end{aligned}$$

where $y(t)$ is output, $u(t)$ is input, q^{-1} is backward shift operator, $\Delta = 1 - q^{-1}$ is the differential operator, $\varepsilon(t)$ is uncorrelated random sequence. A, B and C are polynomials of q^{-1}.

In order to enhance the robustness of the system, the objective function considers the impact of control value $u(k)$ on the system′s future state. Therefore, the target function is selected as

$$\begin{aligned} &\min J(t) = \\ &{\rm E}\left\{\sum_{j=1}^{P} [y(t+j) - w(t+j)]^2 + \sum_{j=1}^{M} \lambda(j)[\Delta u(t+j-1)]^2\right\} \end{aligned} \tag{16}$$

where P is maximum predicted length of target function, M is control length of target function, w is output expected value. In order to soften the control, the controlled objective output is not directly tracking the reference value, but tracking reference trajectory. The reference trajectory is determined by reference value $y_{\rm ref}$, output value y and diffusion coefficient a $(0 < a < 1)$, and can be expressed as

$$w(k+j) = a_j y(k) + (1-a_j)y_{\rm ref}. \tag{17}$$

GPC problem can be attributed to the solution of $\Delta u(k), \Delta u(k+1), \cdots, \Delta u(k+M-1)$. In order to minimize the objective function value, Diophantine equation is introduced to solve the optimal predictive value.

$$y(k+j) = G_j(q^{-1})\Delta u(k+j-1) + F_j(q^{-1})y(k), \quad j = 1, 2, \cdots, n \tag{18}$$

The objective function (16) is minimized, then the optimal control is expressed as

$$\Delta U = (G^{\rm T}G + \lambda I)^{-1}G^{\rm T}(w-f). \tag{19}$$

GPC directly identifies parameters of the original system model. It requires solving Diophantine equations and online recursive calculation of controller parameters. Control value calculation time is longer. Implicit GPC directly identifies parameters G and F, because implicit algorithm does not need to recursive solution of Diophantine equation, it can save computation time.

By (19), the solution of matrix ΔU requires the knowledge of matrix G and the open-loop predictive vector f. Implicit GPC makes use of input and output data according to the identification of G and f. By (16), P predictive controllers are obtained as

$$\left\{\begin{aligned} &y_M(k+1) = g_1\Delta u(k) + f(k+1)\\ &y_M(k+2) = g_2\Delta u(k) + g_1\Delta u(k+1) + f(k+2)\\ &\quad\cdots\\ &y_M(k+P) = g_P\Delta u(k) + \cdots + g_1\Delta u(k+P-1) +\\ &\qquad\qquad f(k+p). \end{aligned}\right. \tag{20}$$

The above equation shows that all the elements of the matrix G have appeared in the last equation, so the matrix G can be obtained by the last equation identification.

The last equation in (20) is written in the form of

$$y_M(k+P) = \varphi(k)\theta(k). \tag{21}$$

In the above equation,

$$\begin{aligned} &\varphi(k) = [\Delta u(k), \Delta u(k+1), \cdots, \Delta u(k+P-1), 1]\\ &\theta(k) = [g_P, g_{P-1}, \cdots, g_1, f(k+p)]^{\rm T}. \end{aligned}$$

According to the recursive least squares method, after $\theta(k)$ is solved, the matrix G and the elements $g_1, g_2, \cdots, g_P$ and $f(k+p)$ in the vector f can be identified.

3.2 Implicit PIGPC algorithm

To enhance network time-delay compensation effect, an improved implicit GPC algorithm with PI structure is proposed. Implicit PIGPC combines advantages of implicit GPC and PI control, i.e., PI feedback structure and GPC prediction function. It is an adaptive controller with more robustness, and it is more appropriate for industrial applications. Implicit GPC control algorithm is adopted as the objective function:

$$J(k) = {\rm E}\{\sum_{\rm j=1}^{\rm P}[{\rm e(k+j)}]^2 + \sum_{\rm j=0}^{\rm M-1}\lambda[\Delta{\rm u(k+j)}]^2\} \tag{22}$$

where the error sequence $e(k+j)$ satisfies

$$e(k+j) = w(k+j) - y(k+j). \tag{23}$$

As derived based on the above model and objective function, the implicit GPC control algorithm only has integral structure. This paper modifies the above objective function, making GPC control algorithm also have the proportional structure.

$$\begin{aligned} J(k) = {\rm E}\{&\sum_{\rm j=1}^{\rm P}[{\rm K_p}(\Delta{\rm e(k+j)})^2 + {\rm K_i e(k+j)}^2] +\\ &\sum_{j=0}^{M-1}\lambda[\Delta u(k+j)]^2\} \end{aligned} \tag{24}$$

where $K_p \geq 0$, $K_i \geq 0$ are given constants, called scale factor and integrating factor. To obtain the system output predicted value, the following two Diophantine equations are solved.

$$\begin{aligned} &1 = A(q^{-1})\Delta E_j(q^{-1}) + q^{-j}F_j(q^{-1})\\ &E_j(q^{-1})B(q^{-1}) = G_j(q^{-1}) + q^{-j}H_j(q^{-1}). \end{aligned}$$

The output predicted value at time $k+j$ is

$$y(k+j) = G_j\Delta u(k+j-1) + F_j y(k) + H_j\Delta u(k-1). \tag{25}$$

Denoting by $f(k+j) = F_j y(k) + H_j\Delta u(k-1)$, the above equation can be written as

$$y(k+j) = G_j\Delta u(k+j-1) + f(k+j). \tag{26}$$

Using the predicted value instead of the actual value, deviation and increment value between output and set value can be defined as

$$e(k+j) = w(k+j) - f(k+j) - G_j\Delta u(k+j-1) \tag{27}$$

$$\begin{aligned} \Delta e(k+j) = &[\Delta w(k+j) - \Delta f(k+j)] -\\ &[G_j\Delta u(k+j-1) - G_{j-1}\Delta u(k+j-2)]. \end{aligned} \tag{28}$$

By substituting (27) and (28) into the objective function (22), and let $\partial J/\partial\Delta u = 0$, then the following equation is obtained as

$$\begin{aligned} \Delta u = &(\lambda I + K_p G_p^{\rm T}G_p + K_i G_i^{\rm T}G_i)^{-1}\times\\ &[K_p G_p^{\rm T}(\Delta w - \Delta f) + K_i G_i^{\rm T}(w-f)]. \end{aligned} \tag{29}$$

In the above equation:

$$G_i=\begin{bmatrix} g_0 & 0 & 0 & \cdots & 0\\ g_1 & g_0 & 0 & \cdots & 0\\ & & & \cdots & \\ g_{M-1} & g_{M-2} & g_{M-3} & \cdots & g_0\\ & & & \cdots & \\ g_{P-1} & g_{P-2} & g_{P-3} & \cdots & g_{P-M}\end{bmatrix}$$

$$G_p=\begin{bmatrix} g_0 & 0 & \cdots & 0\\ g_1-g_0 & g_0 & \cdots & 0\\ g_2-g_1 & g_1-g_0 & \cdots & 0\\ & & \cdots & \\ g_{P-2}-g_{P-3} & \cdots & \cdots & g_0\\ g_{P-1}-g_{P-2} & g_{P-2}-g_{P-3} & \cdots & g_{M-1}-g_{M-2}\end{bmatrix}.$$

Let $R_p=(\lambda I+K_pG_p^{\rm T}G_p+K_iG_i^{\rm T}G_i)^{-1}K_pG_p^{\rm T}$, $R_i=(\lambda I+K_pG_p^{\rm T}G_p+K_iG_i^{\rm T}G_i)^{-1}K_iG_i^{\rm T}$, (29) can be described as

$$\Delta u=R_p\Delta e+R_ie. \tag{30}$$

Then control value of implicit PIGPC is

$$u(k)=u(k+1)+R_p\Delta e+R_ie. \tag{31}$$

To further simplify the objective function, the following expression is defined as

$$S=\begin{bmatrix} 1 & 0 & 0 & \cdots & 0\\ -1 & 1 & 0 & \cdots & 0\\ 0 & -1 & 1 & \cdots & 0\\ 0 & 0 & \cdots & -1 & 1\end{bmatrix}.$$

Substituting $G_p=SG_i$, $\Delta w=Sw$, $\Delta f=Sf$ into (30),

$$\begin{aligned}\Delta u=&(\lambda I+K_pG_i^{\rm T}S^{\rm T}SG_i+K_iG_i^{\rm T}G_i)^{-1}\times\\ &[K_pG_i^{\rm T}S^{\rm T}(Sw-Sf)+K_iG_i^{\rm T}(w-f)].\end{aligned} \tag{32}$$

Let $\Omega=K_iI+K_pS^{\rm T}S$, then

$$\Delta u=(\lambda I+G_i^{\rm T}\Omega G_i)^{-1}G_i^{\rm T}\Omega(w-f). \tag{33}$$

Equation (33) is the form of the controller, compared with the standard implicit GPC algorithm, which does not increase the computational complexity, but can achieve better control effect.

4 Time-delay compensation method

Network time-delay compensation method in the paper is described as follows: Since implicit PIGPC algorithm is used in the controller, so the control value sequence of the future N cycles at time k (N is length of control sequence) can be calculated, but the specific use of the control sequence which acts on the actuator can be determined according to the relationship between time-delay and the sampling period.

Assuming the system sampling period is T, the time-delay of the current control cycle is τ. Let

$$n=\begin{cases}\left\lfloor\dfrac{\tau}{T}\right\rfloor, \text{ if } \dfrac{\tau}{T} \text{ is an integer}\\ \left\lfloor\dfrac{\tau}{T}\right\rfloor+1, \text{if } \dfrac{\tau}{T} \text{ is not an integer.}\end{cases} \tag{34}$$

Select $u(k+n|k)$ as the current control value. So after a delay of τ, control value is acting on the controlled object at $k+n$ moment, hence the control system with time-delay is transformed into a control system without time-delay.

In summary, the steps of time-delay compensation method in the paper can be described as follows:

Step 1. True time toolbox is used to build simulation environment, time-delay sample data is measured, LSSVM model parameters γ, σ^2 and m are optimized by PSO algorithm.

Step 2. At time k, LSSVM algorithm is used to predict the current network time-delay, predicted time-delay is used as the controller parameters instead of the actual value.

Step 3. Implicit PIGPC is used to get control increment value Δu.

Step 4. According to network time-delay compensation method in this paper, the control value $u(k+n|k)$ is calculated and sent to the actuator.

Step 5. Step 2 is returned to continue at time k+1.

5 Simulation

True time toolbox is used to build an IEEE 802.11b wireless NCS. Sensors are time-driven, controllers and actuators are event-driven, and the controlled object is

$$G_p(s)=\frac{100s+1}{s^2+100s+1}.$$

The sampling period is 10 ms, the network speed is 80 000 bits/s, the minimum frame size is 80 bits, the packet size is 80 bits, the interfering node is added to the system, the interference ratio is 20%, and interference can make network time-delay randomly varying.

Four hundreds groups of network time-delay in the built simulation system are obtained, the initial 300 groups for model training, the following 100 groups are used to verify the prediction accuracy of the model. When prediction steps 1 to 4 were taken, the algorithm computing time was noted as shown in Table 1.

Table 1 Computing time with different prediction steps

Prediction steps	Computing time (ms)
1	3.41
2	6.01
3	7.06
4	8.41

Table 1 shows that computing time is less than the sampling period, and when the prediction step is smaller, computing time is shorter. So single step prediction is selected in this paper for predicting current network time-delay online. To optimize LSSVM parameters by PSO algorithm, the optimization time was 39.001 ms, this time is much larger than the sampling period. Therefore, the parameter optimization process should be off-line.

PSO simulation parameters are chosen as follows: Population size N is 30, the maximum number of iterations is 300, learning factor c_1 is 1.5, c_2 is 1.7, the inertia weight is 1, fitness function is MSE between predicted and actual time-delay value, the value range of parameters is $\gamma \in [0.01, 1\,000]$, $\sigma^2 \in [0.01, 1\,000]$, $m \in [1, 30]$.

The particle current optimal position is updated constantly until the optimization condition is satisfied, then we can get the optimal regularization parameter γ as 57.676 6, RBF kernel width σ^2 as 3.083 9, embedded dimension m as 13, PSO fitness curve as shown in Fig. 1. Using LSSVM model with optimal parameters to predict network time-delay, simulation results are showed in Fig. 2.

For researching the predictive effect, we compare the predictive effect of the method based on PSO-LSSVM in this paper with other prediction methods, including the grid search optimized LSSVM (optimized parameters γ is 5.217 4, σ^2 is 5.492 4), autoregressive integrated moving average (ARIMA) prediction method (autoregressive order is 3, the number of differential is 1, the moving average order is 2), error back propagation (BP) neural network (the learning rate is 0.1, expected error minimum value is 0.000 04, maximum number of training samples is 5 000). Figs. 3–5 are the comparing graphs between 100 actual and predicted values based on several algorithms. They show that time-delay predicted values of PSO-LSSVM method are better than the other methods.

For further comparison, Table 2 presents a comparison between mean square error and mean absolute percentage error (MAPE) of several algorithms. It can be seen that PSO-LSSVM in this paper has higher prediction accuracy.

Table 2 Comparison of predictive effect

The prediction method	MSE (ms)	MAPE (ms)
PSO-LSSVM	2.23	2.36
Grid search LSSVM	3.94	3.71
BP neural network	11.63	8.63
ARIMA	17.52	11.37

In order to compare the effect of time-delay compensation, the implicit PIGPC parameters are as follows: Prediction step P is 6, the control step M is 5, the weighting coefficient λ is 1, forgetting factor μ is 1, softening coefficient α is 0.1, scale factor K_p is 0.5, integrating factor K_i is 0.5. The implicit GPC parameter are: Prediction step P is 6, the control step M is 5, the weighting coefficient λ is 1, forgetting factor μ is 1, softening coefficient α is 0.1. Input reference signal is a square wave output, the tracking curve is shown in Fig. 6, and the control value curve is shown in Fig. 7.

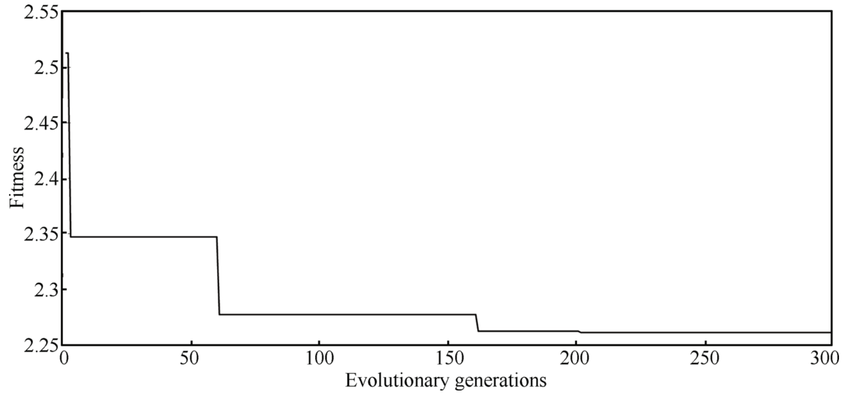


Fig. 1 PSO algorithm optimization fitness curve

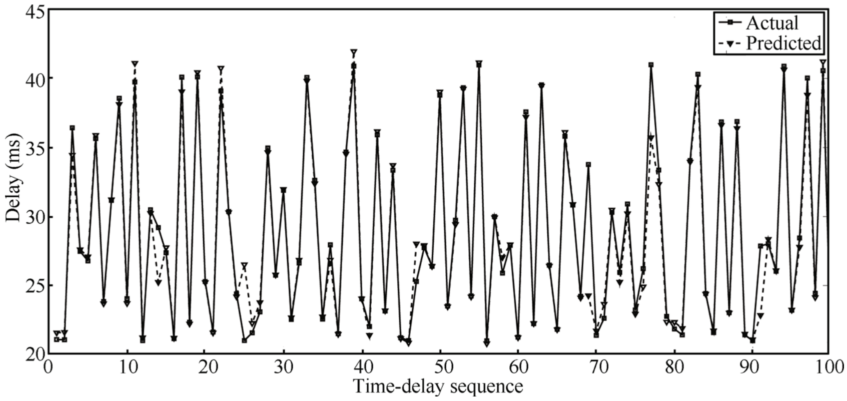


Fig. 2 PSO-LSSVM prediction curve

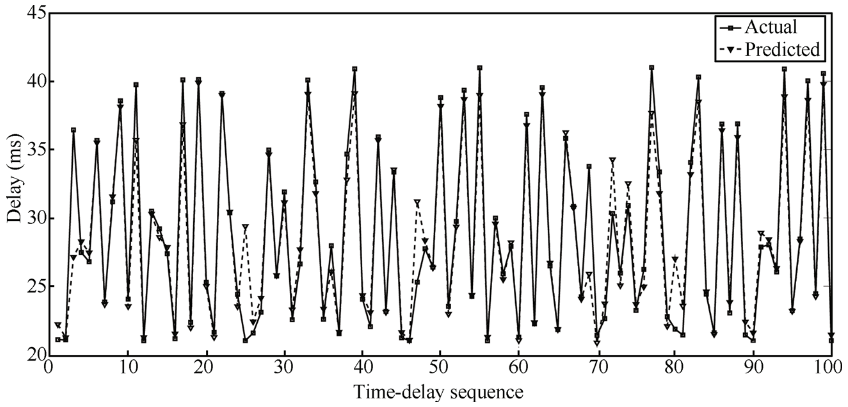


Fig. 3 Grid search LSSVM prediction curve

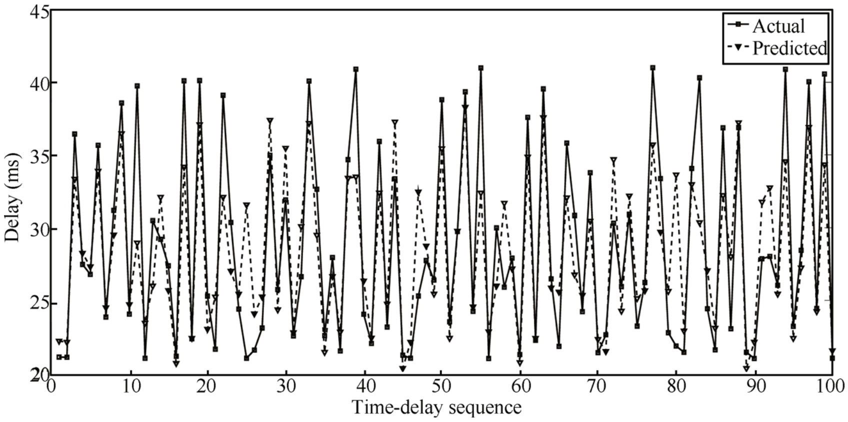


Fig. 4 BP neural network prediction curve

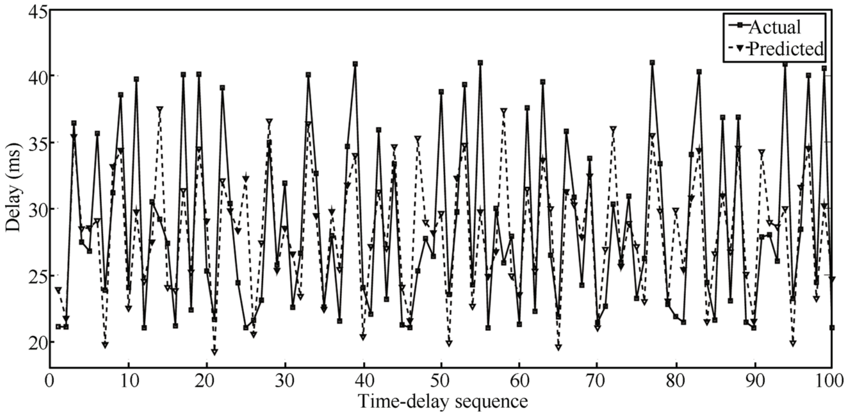


Fig. 5 ARIMA prediction curve

From Figs. 6 and 7, it can be found that the control effect of implicit PIGPC is better than implicit GPC algorithm. A measured control cycle needs 6.5 ms in standard GPC and 1.9 ms in implicit GPC and implicit PIGPC. The results showed that the implicit GPC and implicit PIGPC are about 3.42 times faster than the standard GPC. So the implicit PIGPC enhances the time-delay compensation capability and does not increase the executing time compared with the implicit GPC. Hence, it is more suitable for the time-delay compensation of NCS.

6 Conclusions

The paper selects PSO algorithm optimized LSSVM to predict network time-delay, uses the predictive time-delay instead of the actual time-delay as the controller parameters in the design of a time-delay compensation controller. An implicit GPC algorithm with PI structure is given for network time-delay compensation. The wireless network simulation environment was built by true time toolbox, and the effectiveness of the proposed method is verified through simulation results. Our future research work directions are

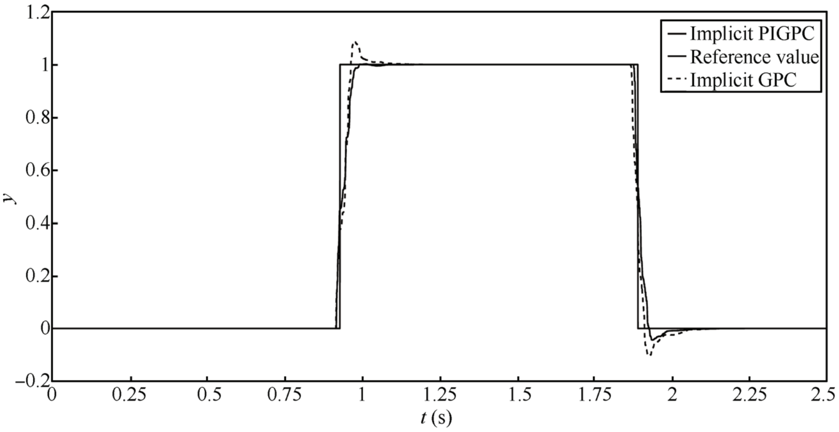


Fig. 6 The tracking curve for reference signal

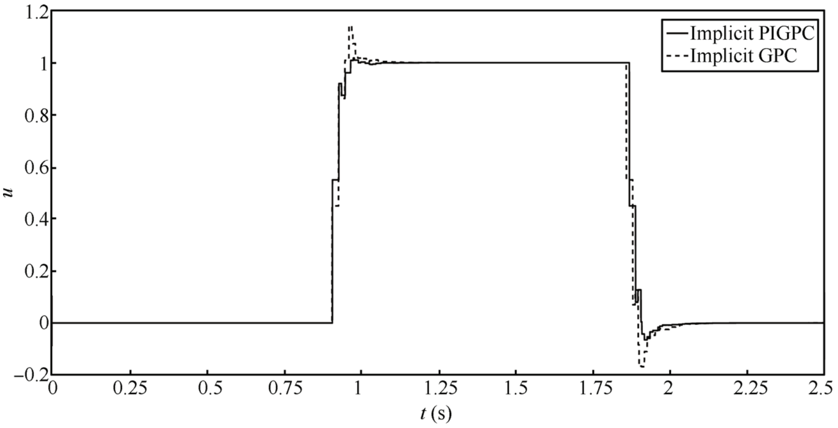


Fig. 7 The control value curve

as follows:

1) Network time-delay is affected by factors such as traffic load change, instrument connection, etc. The simulation section in this paper, network speed, packet size, interference and other parameters are given. LSSVM model trained by the time-delay sample data has good prediction effect. However, the changing of simulation parameters may result in reduced prediction accuracy of the model. The paper will proceed with a detailed study in future work.

2) Due to the limited by the hardware development, in this paper, the true time toolbox is utilized for simulation. The simulation results show that the proposed method worked well in the wireless network. The following work is to make field experimental verification.

References

[1] Z. D. Tian, X. W. Gao, K. Li. A hybrid time-delay prediction method for networked control system. *International Journal of Automation and Computing*, vol. 11, no. 1, pp. 19–24, 2014.

[2] N. Vatanski, J. P. Georges, C. Aubrun, E. Rondeau, S. L. Jämsä-Jounela. Networked control with delay measurement and estimation. *Control Engineering Practice*, vol. 17, no. 2, pp. 231–244, 2009.

[3] W. L. Li, X. B. Zhang, H. M. Li. Co-simulation platforms for co-design of networked control systems: An overview. *Control Engineering Practice*, vol. 23, pp. 44–56, 2014.

[4] C. J. Zhang, J. Gu. Robust fault detection filter for nonlinear state-delay networked control system. *International Journal of Automation and Control*, vol. 6, no. 3–4, pp. 215–230, 2012.

[5] Y. Shi, J. Huang, B. Yu. Robust tracking control of networked control systems: Application to a networked DC motor. *IEEE Transactions on Industrial Electronics*, vol. 60, no. 12, pp. 5864–5874, 2013.

[6] Z. S. Chen, Y. He, M. Wu. Robust fuzzy tracking control for nonlinear networked control systems with integral quadratic constraints. *International Journal of Automation and Computing*, vol. 7, no. 4, pp. 492–499, 2010.

[7] T. House, M. J. Keeling. Epidemic prediction and control in clustered populations. *Journal of Theoretical Biology*, vol. 272, no. 1, pp. 1–7, 2011.

[8] R. N. Yang, G. P. Liu, P. Shi, C. Thomas, M. V. Basin. Predictive output feedback control for networked control systems. *IEEE Transactions on Industrial Electronics*, vol. 61, no. 1, pp. 512–520, 2014.

[9] W. Fu, X. Y. Yang, W. Feng, G. Q. Liu. Predictive control compensation for time delay in networked control systems. *Systems Engineering and Electronics*, vol. 33, no. 9, pp. 2066–2071, 2011. (in Chinese)

[10] H. B. Song, G. P. Liu, L. Yu. Networked predictive control of uncertain systems with multiple feedback channels. *IEEE Transactions on Industrial Electronics*, vol. 60, no. 11, pp. 5228–5238, 2013.

[11] W. G. Shi, C. Shao, Z. Y. Sun. Improved GPC network-control algorithm based on AR model time-delay prediction. *Control and Decision*, vol. 27, no. 3, pp. 477–480, 2012. (in Chinese)

[12] B. Tang, G. P. Liu, W. H. Gui, Y. L. Wang. State-space model based generalized predictive control for networked control systems. In *Proceedings of the 17th IFAC World Congress*, IFAC, Coex, South Korea, pp. 13006–13011, 2008.

[13] G. P. Liu, Y. Q. Xia, J. Chen, D. Rees, W. S. Hu. Networked predictive control of systems with random network delays in both forward and feedback channels. *IEEE Transactions on Industrial Electronics*, vol. 54, no. 3, pp. 1282–1297, 2007.

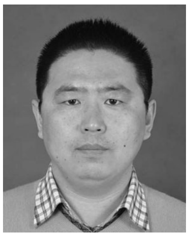
[14] G. C. Walsh, O. Beldiman, L. G. Bushnell. Error encoding algorithms for networked control systems. *Automatica*, vol. 38, no. 2, pp. 261–267, 2002.

[15] J. A. K. Suykens, L. Lukas, J. Vandewalle. Sparse approximation using least squares support vector machines. In *Proceedings of IEEE International Symposium on Circuits and Systems*, IEEE, Geneva, Switzerland, pp. 757–760, 2000.

[16] Z. D. Tian, X. W. Gao, T. Shi. Combination kernel function least squares support vector machine for chaotic time series prediction. *Acta Physica Sinica*, vol. 63, no. 16, Article number 160508, 2014. (in Chinese)

[17] Z. D. Tian, X. W. Gao, K. Li. Networked control system time-delay compensation algorithm based on predictive control. *Systems Engineering and Electronics*, vol. 35, no. 10, pp. 2165–2169, 2013. (in Chinese)

[18] D. M. Wonohadidjojo, G. Kothapalli, M. Y. Hassan. Position control of electro-hydraulic actuator system using fuzzy logic controller optimized by particle swarm optimization. *International Journal of Automation and Computing*, vol. 10, no. 3, pp. 181–193, 2013.



Zhong-Da Tian received the B. Sc. degree in communication engineering from Liaoning University, China in 2001, received the M. Sc. degree in communication and information system from Northeastern University, China in 2004, and received the Ph. D degree in control theory and control engineering from Northeastern University, China in 2013. He is currently a lecturer in College of Information Science and Engineering, Shenyang University of Technology, China.

His research interests include delay compensation and scheduling for networked control system, time series prediction.

E-mail: tianzhongda@126.com (Corresponding author)

ORCID iD: 0000-0003-0379-4048

Xian-Wen Gao received the B. Sc. degree in control science from Shenyang University of Chemical Technology, China in 1978, and received the M. Sc. and Ph. D degrees in control science from Northeastern University, China in 1993 and 1998, respectively. He is currently a professor in college of information science and engineering, Northeastern University, China.

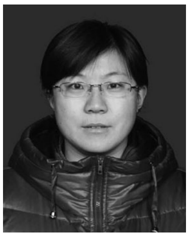
His research interests include complex industrial process modeling and intelligent optimal control theory and application.

E-mail: gaoxianwen@ise.neu.edu.cn

Bi-Lian Gong received the B. Sc. and M. Sc. degrees in control theory and control engineering from Northeastern University, China in 2010 and 2012, respectively.

Her research interest is time-delay compensation method for networked control system.

E-mail: gongbilian@yahoo.cn



Tong Shi received the B. Sc. degree from Bohai University, China in 2004. She is currently a lecturer in department of humanities, Liaoning Forestry Vocation-Technical College, China.

Her research interest is time-delay compensation for networked control system.

E-mail: shitongjz@163.com