*Algorithms* **2015**, *8*, 3-18; doi:10.3390/a8010003

OPEN ACCESS

*algorithms*

**ISSN 1999-4893**

www.mdpi.com/journal/algorithms

*Article*

# Networked Control System Time-Delay Compensation Based on Time-Delay Prediction and Improved Implicit GPC

**Zhong-Da Tian *, Shu-Jiang Li, Yan-Hong Wang and Hong-Xia Yu**

College of Information Science and Engineering, Shenyang University of Technology,
Shenyang 110870, China; E-Mails: lisj2005@126.com (S.-J.L.); sshuang123456@163.com (Y.-H.W.);
hongxia7512@163.com (H.-X.Y.)

* Author to whom correspondence should be addressed; E-Mail: tianzhongda@126.com;
Tel.: +8624-2549-6349; Fax: +8624-2549-6380.

Academic Editor: Xiaowen Chu

*Received: 18 November 2014 / Accepted: 31 December 2014 / Published: 19 January 2015*

**Abstract:** The random time delay in a networked control system can usually deteriorate the control performance and stability of the networked control system. In order to solve this problem, this paper puts forward a networked control system random time-delay compensation method based on time-delay prediction and improved implicit generalized predictive control (GPC). The least squares support vector machine is used to predict the future time delay of network. The parameters of the least squares support vector machine time-delay prediction model are difficult to determine, and the genetic algorithm is used for least squares support vector machine optimal prediction parameter optimization. Then, an improved implicit generalized predictive control method is adopted to compensate for the time delay. The simulation results show that the method in this paper has high prediction accuracy and a good compensation effect for the random time delay of the networked control system, has a small amount of on-line calculation and that the output response and control stability of the system are improved.

**Keywords:** networked control system; time delay; prediction; least squares support vector machine; improved implicit GPC

## 1. Introduction

A 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, especially in complex networks [2]. The time delay can be constant or time varying, which makes the analysis and control design more complex [3]. For the network-induced time-delay compensation problem, many scholars have conducted a great deal of research work. Robust control [4,5], intelligent control [6,7], predictive control [8,9], pinning control [10–12] and other time-delay compensation methods have led to a lot of developments and applications.

Generalized predictive control (GPC) was proposed by Clarke in 1987. GPC has many advantages: the first one 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 the feedback correction strategy, the control effect is more suitable for industrial process control requirements.

The auto regression model is utilized to predict the time delay, and an improved GPC algorithm for time-delay compensation was proposed [13]; however, the author did not give the details of the simulation parameters. Based on a random time-varying network delay, the GPC network based on a state space model is used to access an effective tracking performance [14]. Liu *et al.* gives a predictive control method for a forward channel and feedback channel with random time delay and gives the closed-loop predictive control system stable conditions [15]. However, the GPC algorithm requires solved Diophantine equations and a matrix inversion operation, which led to excessive online calculation; on the other hand, since the time delay is random, the parameter selection for the GPC algorithm requires a larger prediction step size, thereby increasing the calculation time. Furthermore because the network delay is uncertain, the GPC algorithm requires a larger prediction step, thus increasing the computing time, reducing the real-time performance of the system [16].

Based on the above analysis, the research motivation of this paper includes two aspects. The first is to find a suitable predictive method to predict time delay, to reduce the prediction step size of the GPC algorithm. The second is to improve the GPC algorithm, reducing the calculation amount of the GPC algorithm. This paper will carry out research work on these two aspects. In the literature [17], the author used the particle swarm optimization (PSO) algorithm optimized least squares support vector machine (LS-SVM) to predict the future time delay, compensating time delay through implicit GPC, but the embedding dimension is not optimized in the paper. At the same time, the control value of implicit GPC needs to wait for the next integer time sampling period to be applied to the controlled object. It is easy to cause the system response speed to slow and to produce a big overshoot, so it is necessary to improve it. On the basis of the literature [17], this paper discussed LS-SVM prediction accuracy with different parameters, firstly using the genetic algorithm-optimized parameters of the LS-SVM time-delay prediction model, and then, an improved implicit generalized predictive control method is used to compensate for the random time delay; the output performance of the system is improved. Through the simulation, the effectiveness of the proposed method in this paper is verified.

## 2. LS-SVM Time-Delay Prediction Model

The time delay in the networked control system includes: the sensor for the controller time delay $\tau_{sc}$, the controller to the actuator time delay $\tau_{ca}$ and the controller calculated time delay $\tau_c$, shown as in Figure 1.

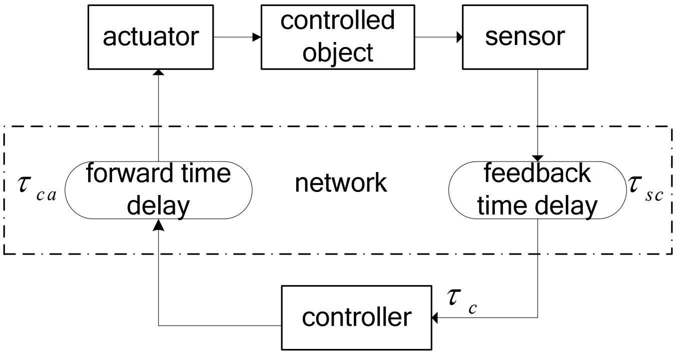

**Figure 1.** Networked control system with time delay.

The sensor for the controller time delay and the controller to actuator time delay do not have the same variation rules, and the effect on the system performances is not the same either. Under normal circumstances at 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 just moments before *k* is known, so the historical time delay can be used to predict the current network time delay. In the networked control system discussed in this paper, the controller and actuator are event driven, and the sensor is clock-driven; therefore, the forward and feedback channel time delay of networked control system can be combined for analysis [13]. 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 for analysis, the total system time delay $\tau(k)$ can be predicted through an appropriate prediction model.

Figure 2 gives 500 groups' time delay test data collected from an IEEE 802.11b-based networked control system. From the figure, it can be seen that time delay has a strong, nonlinear characteristic, and the mean value of the time delay sequence is within a range of variation over a longer time, but, in a certain interval during a shorter time.

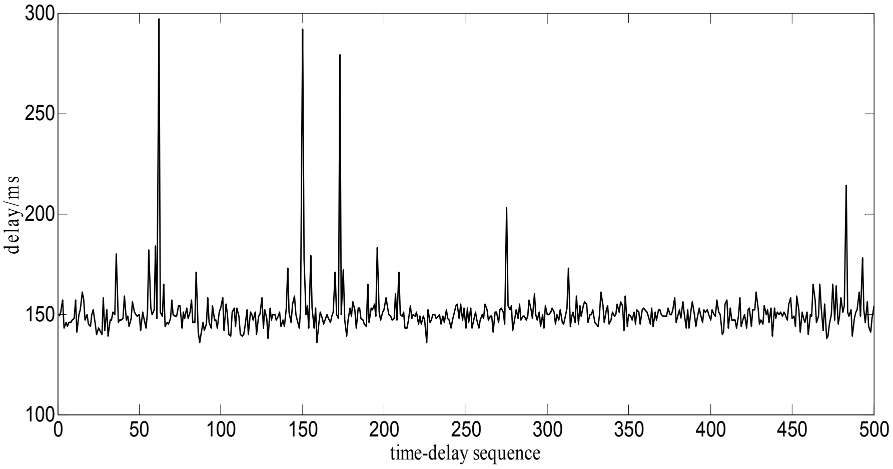

**Figure 2.** Typical random time delay of a network.

The predictability and similarity of the time-delay sequence can be measured by the Hurst parameter; the time-delay sequence has self-similarity when $H \in (0.5,1)$; the greater $H$ is, the greater is the sequence similarity. In this paper, the *R*/*S* (rescaled range analysis) method is used to calculate the Hurst parameter of the time-delay sequence, and it can be described as:

$$(R/S)_n = An^H \tag{1}$$

where in $n$ is the number of samples, $R$ is the rescaling range, that is $R = \max(X_{t,n} - \min X_{t,n}), t = 1,2,\cdots,n$, $S$ is the standard deviation, $H$ is the Hurst parameter and $A$ is a constant. Given the chart of the relationship between $\lg(R/S)_n$ about lg *n*, the least squares method is used for calculating the slope, and then, the slope is Hurst parameter.

The Hurst parameter of the time-delay sequence is commutated by the *R*/*S* method, as shown in Figure 3. Using a straight line linear fitting of these data points, the slope of the line is the Hurst parameter, *H* is 0.936, apparently satisfying $0.5 < H < 1$, which also shows that the time-delay sequence has self-similarity. Therefore, the time-delay sequence can be predicted through the LS-SVM model.

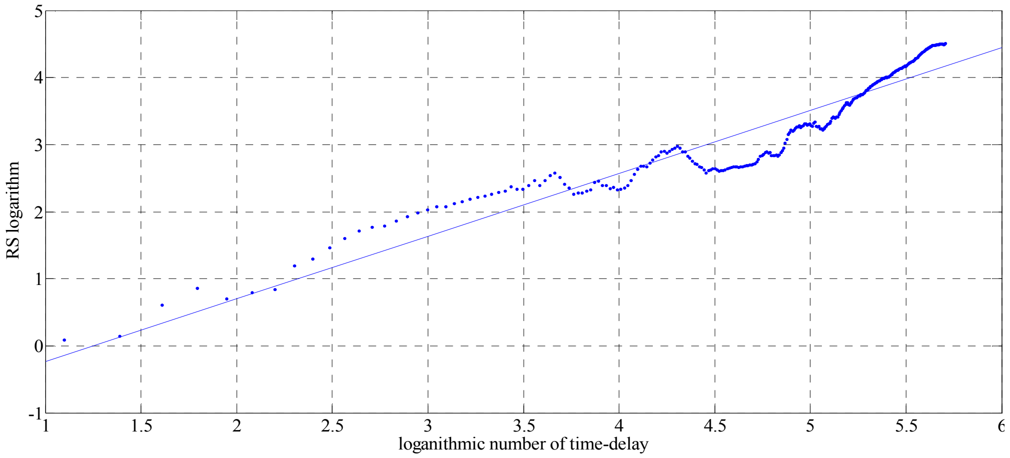

**Figure 3.** Hurst parameter of the time delay sequence.

The standard SVM algorithm is complex; based on the standard SVM, LS-SVM sets the loss function of SVM as square errors, converts the inequality constraints into equality constraints and reduces the undetermined parameters. At the same time, it transforms the quadratic programming problem into linear equations, which can reduce the complexity of the solution [17].

LS-SVM uses the following function to estimate the random time delay of the networked control system [18]:

$$y = w^T \Phi(x) + b \tag{2}$$

Nonlinear function $\Phi(\cdot)$ can transform the input space to a high dimensional feature space, so that the nonlinear fitting of input space can be seen as a linear fitting of the high dimensional feature space. Let $\{x_i, y_i\}, i = 1,2,\cdots,N$ be a training set, based on the risk minimization principle; the regression problem can be expressed as a constrained optimization problem:

$$\min J(w,e) = \frac{1}{2} w^T w + \frac{1}{2} \gamma \sum_{i=1}^{N} e_i^2, \gamma > 0 \tag{3}$$

$$y_i = w^T \Phi(x_i) + b + e_i \tag{4}$$

wherein $x_i$ is a time-delay input set, $y_i$ is a time-delay output prediction set, $x_i$ and $y_i$ can be expressed as $x_i = [d_i \; d_{i+1} \; \cdots \; d_{i+m-1}]$, $y_i = [d_{i+m}]$ and $m$ is the embedding dimension; and the current time and past $m - 1$ time-delay value can be used to predict the next time-delay value of the system. $\lambda$ is the regularization parameter. $b$ is the constant value deviation. The Lagrange function can be established to solve the above optimization problem:

$$L(w,b,e,a) = J(w,e) - \sum_{i=l}^{N} a_i (w^T \Phi(x_i) + b + e_i - y_i) \tag{5}$$

$a_i$ is the Lagrange multiplier. Calculate the partial differential of *w*, *b*, *e*, *a*, and simplify it; then, the next equation can be obtained:

$$\begin{bmatrix} 0 & l^T \\ l & \Omega + \gamma^{-1} l \end{bmatrix} \begin{bmatrix} b \\ a \end{bmatrix} = \begin{bmatrix} 0 \\ y \end{bmatrix} \tag{6}$$

For the given training set $\{x_i, y_i\}$, Equation (6) is used to calculate parameters *a* and *b*, at the same time. When the given set *x* combines with the actual training set $x_i$, the following equation can be used to calculate *y*, the prediction output of the system.

$$y = \sum_{i=l}^{N} a_i K(x, x_i) + b \tag{7}$$

In this paper, the radial basis function (RBF) is chosen as the kernel function of LS-SVM, it is:

$$K(x_i, x_j) = \exp\left( -\frac{\left\| x_i - x_j \right\|^2}{2\sigma^2} \right) \tag{8}$$

From the above modeling process, based on the kernel function of RBF, LS-SVM performance depends mainly on $\gamma$ and $\sigma^2$, but the selection of two parameters lacks uniform standards and theoretical guidance; therefore, the selection of LS-SVM parameters is an unsolved problem [19]. At the same time, for the time-delay prediction problem, the embedding dimension *m* is an important parameter also; *m* that is too small will reduce the prediction accuracy; *m* that is too large will increase the computational complexity. The following Tables 1–3 give the mean square error (MSE) comparison of three different values of the parameters.

**Table 1.** The influence of *m* on prediction precision.

| *m* | $\gamma$ | $\sigma^2$ | MSE |
|---|---|---|---|
| 2 | 10 | 0.001 | 9.0126 |
| 4 | 10 | 0.001 | 4.6340 |
| 6 | 10 | 0.001 | 1.8533 |
| 8 | 10 | 0.001 | 1.9399 |
| 10 | 10 | 0.001 | 1.9049 |

**Table 2.** The influence of γ on prediction precision.

| ***m*** | **γ** | **$\sigma^2$** | **MSE** |
|---|---|---|---|
| 4 | 1 | 0.001 | 58.7254 |
| 4 | 10 | 0.001 | 4.6340 |
| 4 | 100 | 0.001 | 2.9191 |
| 4 | 1,000 | 0.001 | 2.8989 |
| 4 | 10,000 | 0.001 | 2.8987 |

**Table 3.** The influence of $\sigma^2$ on prediction precision.

| ***m*** | **γ** | **$\sigma^2$** | **MSE** |
|---|---|---|---|
| 4 | 10 | 0.001 | 4.6340 |
| 4 | 10 | 0.01 | 4.4829 |
| 4 | 10 | 0.1 | 4.9062 |
| 4 | 10 | 1 | 12.6718 |
| 4 | 10 | 10 | 28.6581 |

From the above data in the Tables, three parameters can greatly affect the prediction accuracy, and only optimizing a parameter separately cannot guarantee that the other parameters are optimal, leading to the prediction precision being not ideal. In order to obtain the best prediction results, it is necessary to make *m*, γ , $\sigma^2$ achieve the optimal values at the same time.

## 3. Genetic Algorithm

The genetic algorithm (GA) uses the biological genetics view, through natural selection, exchange, mutation operation, *etc.* Through the population, evolution is carried out, and it has a strong global search ability and can obtain the global optimal solution in a relatively short time [20].

### *3.1. Individual Coding*

The LS-SVM time-delay prediction model needs the parameters *m*, γ and $\sigma^2$. The individual design is shown in Figure 4: the former $n_1$ bits represent *m*; the middle $n_2$ bits represent γ; the later $n_3$ bits represent $\sigma^2$.

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| $p_m^1$ | ··· | $p_m^{n_1}$ | $p_\gamma^1$ | ··· | $p_\gamma^{n_2}$ | $p_{\sigma^2}^1$ | ··· | $p_{\sigma^2}^{n_3}$ |

**Figure 4.** Design of individuals.

Each of the individual can convert a binary classification string into decimal representation as the actual parameters:

$$p = \min(p) + \frac{\max(p) - \min(p)}{2^k - 1} \times d \tag{9}$$

In the above equation, $p$ is the parameters decimal value, min($p$) is the parameters minimum, max($p$) is the parameters maximum value, $k$ is the length of the parameters binary string, $d$ is the decimal value of the parameters binary string.

### *3.2. Population Initialization*

The standard GA algorithm initial population is random, has a lot of uncertainty and no uniformity, and it cannot contain the global optimal solution information, so leading to the local optimal problem; thus, in this paper, uniform design is used to produce the initial population to ensure the diversity of the initial population and the uniformity of the individual step distribution [21].

### *3.3. Fitness Function*

The GA algorithm search target is to find the appropriate $m$, $\gamma$, $\sigma^2$, so that it can improve the prediction precision of the delay; thus, the fitness function is related to the accuracy of the time-delay prediction model. Set the $i$-th group time-delay prediction mean square error as:

$$mse(i) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{10}$$

In the above equation, $n$ is the number of the time-delay sequence, $y_i$ is the actual value and $\hat{y}_i$ is the LS-SVM prediction value of the $i$-th time-delay. For easy calculating, the $i$-th prediction time-delay fitness function directly uses the mean square error:

$$f(i) = mse(i) \tag{11}$$

### *3.4. The Design of the Genetic Operator*

For the selection operators, calculate the individual fitness value and order the fitness from big to small; the smaller fitness shows a smaller error; thus, it is the best individual. Use that individual to keep the strategy, and choose the $R$-th optimal individuals as the next generation. The other $n - R$ individual crossover and mutation operation produces a new individual as the next generation.

For the crossover operator, in the individuals, three parts separately select three intersections; then, the crossover operator can respond to the part of another individual. After this, test the generated individual, and verify whether the choice of parameters is in a certain range; if not, repeat the crossover operation again.

For the mutation operator, for the three parts of an individual, adopt the traditional flip variation method on the variation operation, and randomly select one individual bit, according to the rules that convert zero to one and one to zero. Carry out the variation operation, and then, test whether the individual variation is in the parameter range; if not, repeat the crossover operation again.

### *3.5. GA Optimized LS-SVM Time-Delay Prediction Method*

To sum up, the GA-optimized LS-SVM time-delay prediction method steps are:

Step 1: Confirm the original character space of the optimized parameters, and the initial evolution algebra is zero.

Step 2: Code the parameters $m$, $\gamma$, $\sigma^2$.

Step 3: Generate $h$ individual initial populations.

Step 4: Convert the time-delay sequence, which is the original length, into the input and output matrix, according to $x_i = [d_i \ d_{i+1} \ \cdots \ d_{i+m-1}]$, $y_i = [d_{i+m}]$. The LS-SVM algorithm is used to train and predict according to the parameters $\gamma$, $\sigma^2$, recording every group time-delay prediction accuracy and each individual fitness value.

Step 5: Choose the best $R$ prediction performance individual for the next generation, and for the other individual, carry out the selection, crossover and mutation operators, producing a new population. The evolution algebra is increased one.

Step 6: Judge whether this meets the end conditions, and if it meets them, jump to Step 7 or return to Step 4 otherwise.

Step 7: Output the best individual $m$, $\gamma$, $\sigma^2$, and adopt the optimum $m$, $\gamma$, $\sigma^2$ as the LS-SVM algorithm parameters to build up the time-delay prediction model. Output the best prediction value.

## 4. Improved Implicit GPC Time-Delay Compensation Algorithm

Set the total time delay between the controller and actuator, sensor and controller in the NCS as $d(k)$. The NCS can be expressed as an auto-regression and moving average with exogenous (ARMAX) model with time-varying discrete parameters:

$$A(q^{-1})y(k) = q^{-d(k)}B(q^{-1})u(k) + C(q^{-1})v(k) \tag{12}$$

wherein:

$$\begin{aligned} A(q^{-1}) &= 1 - a_1(k)q^{-1} - a_2(k)q^{-2} - \cdots a_n(k)q^{-n} \\ B(q^{-1}) &= b_0(k) + b_1(k)q^{-1} + b_2(k)q^{-2} + \cdots + b_m(k)q^{-m} \\ C(q^{-1}) &= 1 - c_1(k)q^{-1} - c_2(k)q^{-2} - \cdots - a_l q^{-l} \end{aligned} \tag{13}$$

$d_{\min} \le d(k) \le d_{\max}$ is the whole time-delay of the NCS; $u(k)$ is the input; $y(k)$ is the output; $v(k)$ is the interference; and $a_i(k)$, $b_i(k)$, $c_i(k)$ are the time-varying parameters of the system.

### *4.1. The Identification of the Time-Varying Parameters*

After the random time delay has been predicted through the LS-SVM algorithm, the networked control system can be written as a regression model.

$$y(k) = \phi^T \theta(k) + v(k) \tag{14}$$

In the above equation:

$$\begin{aligned}
&\phi^T(k) = [y(k-1), y(k-2), \cdots, y(k-n);\\
&u(k-\hat{d}(k)), u(k-\hat{d}(k)-1), \cdots, u(k-\hat{d}(k)-m);\\
&v(k-1), v(k-2), \cdots, v(k-l)];\\
&\theta^T(k) = [a_1(k), a_2(k), \cdots, a_n(k); b_1(k), b_2(k), \cdots,\\
&b_m(k); c_1(k), c_2(k), \cdots, c_l(k)]
\end{aligned} \tag{15}$$

Then, the next equations can be obtained:

$$\hat{\theta}(k) = \hat{\theta}(k-1) + \frac{P(k-1)\phi(k-1)}{r + \phi^T(k-1)P(k-1)\phi(k-1)}[y(k) - \phi^T(k-1)\hat{\theta}(k-1)]; \tag{16}$$

$$P(k) = P(k-1) - \frac{P(k-1)\phi(k-1)\phi^T(k-1)P(k-1)}{r + \phi^T(k-1)P(k-1)\phi(k-1)} \tag{17}$$

In the above equations, $r$ is the forgetting factor, $0 \le r \le 1$; the initial value, $P(0) = \delta^2 I$, $\theta(0) = 0$, $\delta^2$ is a sufficiently large constant and $I$ is the unit matrix. Through the above method, the time-varying parameters $a_i(k)$, $b_i(k)$, $c_i(k)$ of the system can be identified.

### *4.2. Improved Implicit GPC*

Scroll the optimization to make the objective function minimum:

$$J = E\{\sum_{j=1}^{P}[y(k+j) - y_r(k+j)]^2 + \sum_{j=1}^{M}\lambda[\Delta u(k+j-1)^2]\} \tag{18}$$

In the equation, $E$ is the mathematical expectation, $P$ is the maximum prediction length, $M$ is the control length, $M \le P$, $\lambda$ are the weighting coefficients, $\Delta u(k+j-1)$ is the control increment and $y_r(k+j)$ is the input reference trajectory. The GPC algorithm calculates $y(k+j)$, needed to solve the Diophantine equation. To avoid complex calculation, an improved implicit GPC algorithm is presented in this paper. This algorithm directly identifies the controller parameters and does not need to solve the inverse matrix of the GPC algorithm. The algorithm, given the $d_{th}$ optimal prediction value, is:

$$y(k+d \mid k) = X(k)w(k) \tag{5}$$

$$X(k) = [\Delta u(k), \Delta u(k+1), \cdots, \Delta u(k+d-1), 1] \tag{20}$$

$$w(k) = [g_{d-1}, g_{d-2}, \ldots g_0, f(k+d)]^T \tag{21}$$

Define the input soft coefficient $\beta$ as:

$$\Delta u(k+n) = (1 + \sum_{i=1}^{n}\beta^{i})\Delta u(k) \quad (n = 1, L, M-1) \tag{22}$$

The above equation can prevent input signal intensity changes, and it also restricts the change of the input signal. The control increment value matrix is:

$$\Delta U = Q\Delta u(k) = [\Delta u(k), (1+\beta)\Delta u(k), L, (1 + \sum_{j=1}^{M-1}\beta^{j})\Delta u(k)]^T \tag{23}$$

Rewrite Equation (18) of the performance index as the vector form:

$$J = (Y - Y_r)^T (Y - Y_r) + \lambda \Delta U^T \Delta U \tag{24}$$

Set $\partial J / \partial \Delta U = 0$; then, Equations (25) and (26) can be obtained:

$$\Delta u(k) = [(GQ)^T (GQ) + \lambda Q^T Q]^{-1} (GQ)^T [W - F_k y(k) - H_k \Delta u(k-1)] \tag{25}$$

$$u(k) = u(k-1) + \Delta u(k) \tag{26}$$

In the above equations:

$$Q = \begin{bmatrix} 1 & 0 & 0 & 0 & \cdots & 0 \\ 0 & -a_2 & -a_3 & -a_4 & \cdots & -a_n \\ 0 & 0 & -a_3 & -a_4 & \cdots & -a_n \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & 0 & \cdots & -a_n \end{bmatrix} \tag{27}$$

$$F_k = \begin{bmatrix} -a_1 & -a_2 & -a_3 & -a_4 & \cdots & -a_n \\ -a_2 & 0 & -a_3 & -a_4 & \cdots & -a_n \\ -a_3 & 0 & 0 & -a_4 & \cdots & -a_n \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ -a_n & 0 & 0 & 0 & \cdots & -a_n \end{bmatrix} \tag{28}$$

$$G = \begin{bmatrix} g_0 & & 0 \\ g_1 & g_0 & \\ & & O \\ g_{N-1} & g_{N-2} & g_0 \end{bmatrix} \tag{29}$$

$$H_k = \begin{bmatrix} b_2 & b_3 & b_4 & \cdots & b_n \\ 0 & b_3 & b_4 & \cdots & b_n \\ \cdot & \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdot & \cdots & \cdot \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix} \tag{30}$$

Due to the existence of the soft change matrix, it can be seen that no matter how the length of the predictive domain changed, the control value always is $\Delta u(k)$. Secondly, the calculation of the inverse matrix is avoided, greatly reducing the calculation time and ensuring the speediness of the system. However, the algorithm still retains the basic characteristics of the generalized predictive control.

In order to accelerate the response speed of the system and to restrain overshooting, this paper proposed an improved implicit GPC algorithm. The next step control increment value is used for the compensation of Equation (26), and the real control value can be obtained as:

$$u'(k) = u(k) + \partial \Delta u(k+1), \ \partial \in (0,1) \tag{31}$$

The above equation is equivalent to an increase of the differential effects on control system; it will accelerate the response speed of the system.

Since the improved implicit GPC algorithm is used in the controller, so the control value sequence of the future *N* cycles (*N* is the length of the control sequence) at time *k* can be calculated, but the specific use of the control sequence that acts on the actuator can be determined according to the relationship between the time delay and the sampling period.

Assuming that the system sampling period is *T*, the time delay of the current control cycle is $\tau$. Let:

$$n=\begin{cases}\left[\tau/T\right], & \text{if } \tau/T \text{ is an integer}\\ \left[\tau/T\right]+1, & \text{if } \tau/T \text{ is not an integer}\end{cases} \tag{32}$$

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

The improved implicit GPC time-delay compensation steps of this paper can be expressed as follows:

Step 1: Set the initial value.

Step 2: Update the input time-delay sequence, and put the newest measurement of the time delay into the time-delay sequence.

Step 3: The GA-optimized LS-SVM algorithm is used to estimate the network time delay at the current time.

Step 4: Identify time-varying parameters through the prediction time delay by Equations (16) and (17).

Step 5: Calculate the optimal prediction value by Equation (18).

Step 6: Calculate the current real control value by Equations (31).

Step 7: According to Equation (32), the control value $\boldsymbol{u}(k+n\,|\,k)$ is calculated and sent to the actuator;

Step 8: Repeat Step 2 to Step 7 until the end of the simulation.

## 5. Simulation

The true time toolbox is used to build an IEEE 802.11b wireless NCS. Sensors are time driven, and controllers and actuators are event driven. 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 10%; and the interference can create the network time-delay random variation. An unstable controlled object is chosen as:

$$G_p(s)=\frac{2}{50s^2-5s-1} \tag{33}$$

The structure of the IEEE 802.11b network is as shown in Figure 5. The whole simulation structure is as shown in Figure 6. In these figures, Snd means send node, Rcv means receive node.

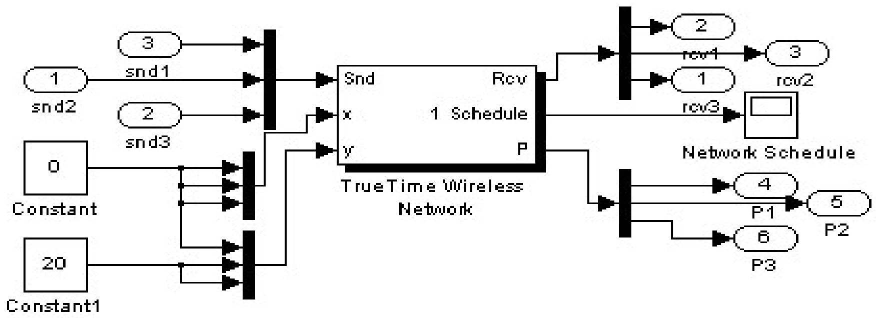

**Figure 5.** The structure of the IEEE 802.11b network.

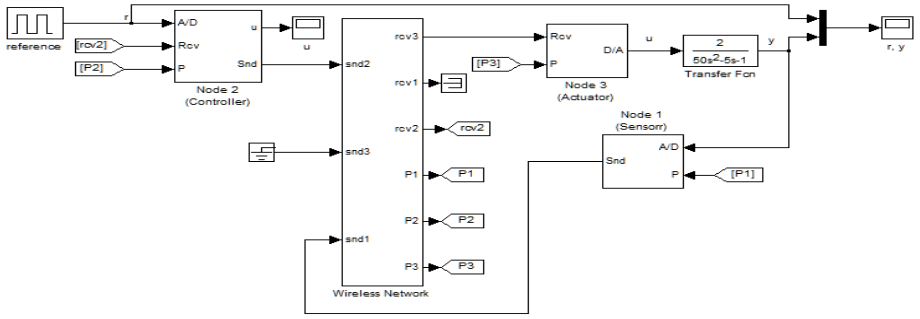

**Figure 6.** The structure of the networked control system (NCS) in this paper.

Then, 300 groups' time-delay test data are collected, and the first 200 groups' data are used to train the GA-optimized LS-SVM time-delay prediction model. The last 100 groups' data are used to verify it. GA algorithm parameters: the number of iterative evolutions is 200; the population quantity is 20; the crossover probability is 0.9; the variation probability is 0.1; the coding length of $m$, $\gamma$, $\sigma^2$ is 10 bits; the $m$ value range is from one to 20; the $\gamma$ value range is from 0.01 to 1000; the$\sigma^2$ value is range from 0.01 to 100; after GA optimized, $m = 9$, $\gamma = 339.8592$ and $\sigma^2 = 2.347$; the MSE is 1.26 for the actual and predicted time delay. Figure 7 is the curve of fitness. Figure 8 is a 100 groups' comparison of the predicted and actual time delay. Figure 9 shows the comparison of the predicted and actual time delay with auto regression (AR) algorithm in the literature [13]; the order of $p$ is nine; the MSE is 10.2791 for the actual and predicted time delay. From the contrast in Figure 8, Figure 9 and the MSE, it can be shown that the time-delay prediction precision of the GA-optimized LS-SVM prediction method is better than the AR algorithm in the literature [13].

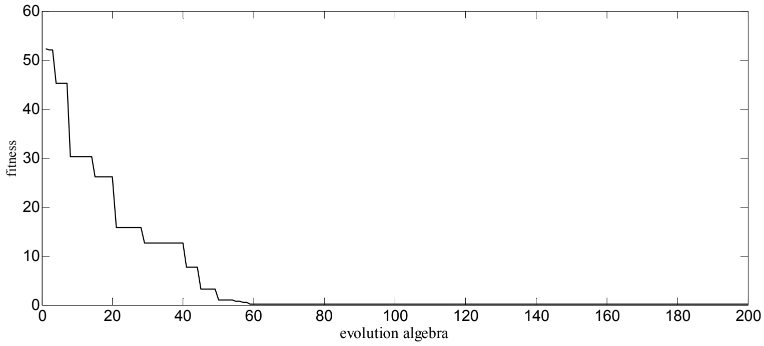

**Figure 7.** The curve of fitness.

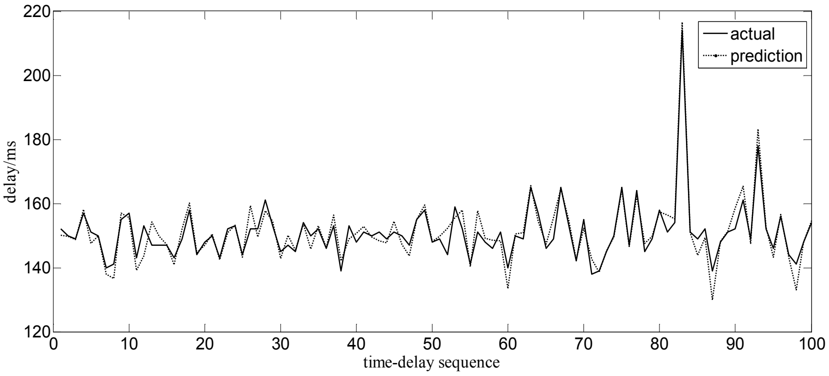

**Figure 8.** GA-optimized least squares (LS-SVM) time-delay prediction result.

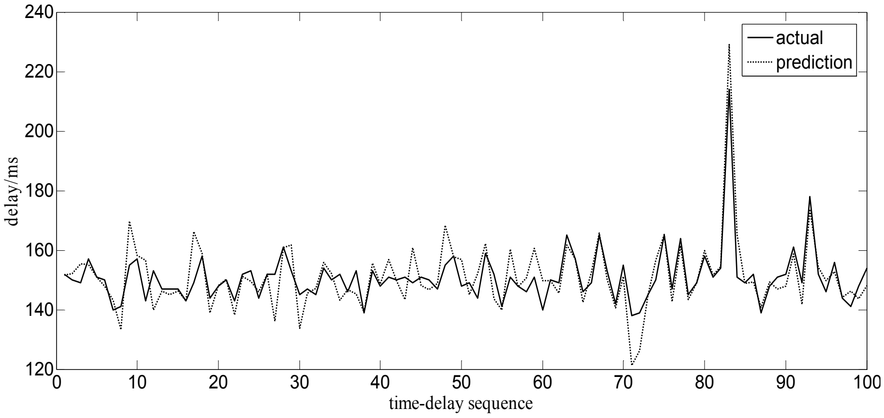

**Figure 9.** Auto regression (AR) time-delay prediction result.

In order to verify the time-delay compensation ability of the improved implicit GPC control algorithm, the simulation is conducted through the built IEEE 802.11b wireless NCS. The improved implicit GPC parameters of the control algorithm are $P = 6$, $M = 6$, $\beta = 0.2$, $\partial = 0.1$. A comparison of the method in the literature [17] is shown in Figure 10 for the square-wave tracking and in Figure 11 for the square-wave tracking with an added white noise amplitude of 0.2 at the output.

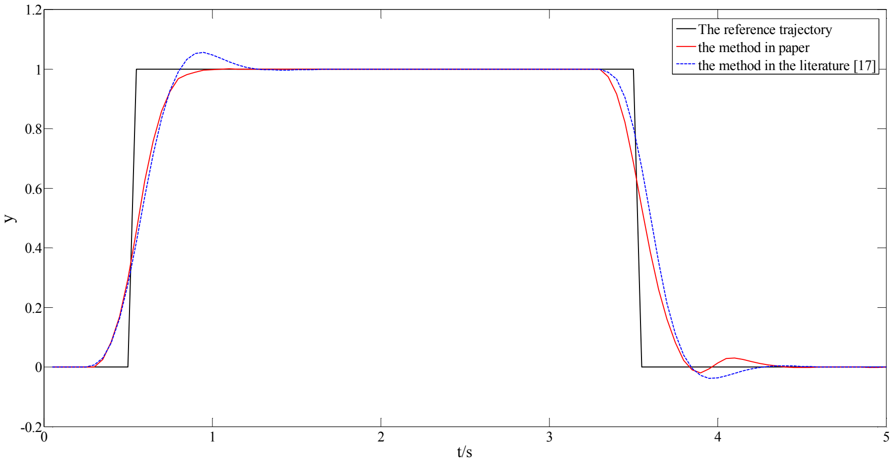

**Figure 10.** Square-wave tracking comparison.

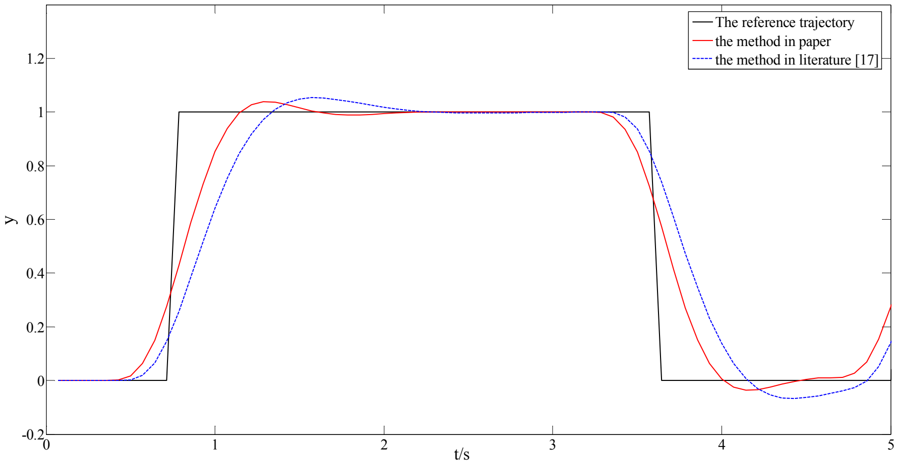

**Figure 11.** Square-wave tracking comparison with added white noise.

The simulation results show that the GA optimized LS-SVM time-delay prediction with improved implicit GPC time-delay compensation method is better than the method in literature [17] in overshoot and response time; it can better track system input and ensure the stability of the system.

A measured control cycle needs 3.4 ms for the method in the literature [17] and 2.1 for the method in this paper. The results showed that the method in this paper is about 1.61-times faster than the method in the literature [17]. Therefore, the method in this paper enhances the time-delay compensation capability and does not increase the execution time; therefore, it is more suitable for the time-delay compensation of a networked control system.

## 6. Conclusions

In the network control system, uncertain factors, such as network delays, can usually deteriorate the control performance and stability of networked control systems. Therefore, a time-delay compensation method based on time-delay prediction and improved implicit GPC is proposed. First, the GA-optimized LS-SVM prediction model is used to estimate the time delay in the networked control system. Then, an improved implicit GPC with predictive time delay is used to obtain the system future control value, to compensate for the time delay of the networked control system. Finally, through the simulation results, the method presented in the paper has a better compensation effect and a smaller calculation time.

In this paper, the parameters of the LS-SVM prediction model are optimized by the genetic algorithm off-line. Additionally, if the time-delay sequence appears to have substantial changes, the prediction accuracy will be reduced. Therefore, the future work is to increase the threshold value; when the cumulative error is larger than the threshold, the optimization process of the parameters is repeated, and more appropriate prediction parameters can be obtained.

## Acknowledgments

This work is partially supported by the National Nature Science Foundation of China (Grant No. 61034005) and the Liaoning Province Doctor Startup Fund (Grant No. 20141070).

## Author Contributions

Zhong-Da Tian participated in the concept, design and interpretation and commented on the manuscript. A substantial amount of Shu-Jiang Li's contribution was to the draft writing and critical revision of this paper. Yan-Hong Wang and Hong-Xia Yu participated in the data collection, analysis and algorithm simulation. All authors read and approved the manuscript.

## Conflicts of Interest

The authors declare no conflict of interest.

## References

1. Tian, Z.-D.; Gao, X.-W.; Li, K. A Hybrid Time-delay Prediction Method for Networked Control System. *Int. J. Autom. Comput.* **2014**, *11*, 19–24.
2. Lü, J.-H.; Chen, G.-R. A time-varying complex dynamical network model and its controlled synchronization criteria. *IEEE Trans. Autom. Control* **2005**, *50*, 841–846.
3. Vatanski, N.; Georges, J.P.; Aubrun, C.; Rondeau, E.; Jämsä-Jounela, S.L. Networked control with delay measurement and estimation. *Control Eng. Pract.* **2009**, *17*, 231–244.
4. Qi, X.-M.; Zhang, C.-J.; Gu, J. Robust fault detection filter for non-linear state-delay networked control system. *Int. J. Autom. Control* **2012**, *6*, 215–230.
5. Shi, Y.; Huang, J.; Yu, B. Robust tracking control of networked control systems: Application to a networked DC motor. *IEEE Trans. Ind. Electron.* **2013**, *60*, 5864–5874.
6. Xu, F.-X.; Cheng, Y.; Ren, H.-L.; Wang, S.-L. Research on Adaptive Neural Network Control System Based on Nonlinear U-Model with Time-Varying Delay. *Math. Probl. Eng.* **2014**, *2014*, 420713.
7. Han, C.-W. Fuzzy Neural Network-based Time Delay Prediction for Networked Control Systems. *Appl. Math Inf. Sci.* **2014**, *8*, 407–413.
8. Yang, R.-N.; Liu, G.-P.; Shi, P.; Thomas, C.; Basin, M.V. Predictive output feedback control for networked control systems. *IEEE Trans. Ind. Electron.* **2014**, *61*, 512–520.
9. Song, H.-B.; Liu, G.-P.; Yu, L. Networked predictive control of uncertain systems with multiple feedback channels. *IEEE Trans. Ind. Electron.* **2013**, *60*, 5228–5238.
10. Yu, W.-W.; Chen, G.-R.; Lü, J.-H. On pinning synchronization of complex dynamical networks. *Automatica* **2009**, *45*, 429–435.
11. Zhou, J.; Lu, J.-A.; Lü, J.-H. Pinning adaptive synchronization of a general complex dynamical network. *Automatica* **2008**, *44*, 996–1003.
12. Yu, W.-W.; Chen, G.-R.; Lü, J.-H.; Kurths, J. Synchronization via pinning control on general complex networks. *SIAM J. Control Optim.* **2013**, *51*, 1395–1416.
13. Shi, W.-G.; Shao, C.; Sun, Z.-Y. Improved GPC network-control algorithm based on AR model time-delay prediction. *Control Decis.* **2012**, *27*, 470–480. (In Chinese)

14. Tang, B.; Liu, G.-P.; Gui, W.-H.; Wang, Y.-L. State-space model based generalized predictive control for networked control systems. In Proceedings of 17th IFAC World Congress, Seoul, Korea, 6–11 July 2008; pp. 13006–13001.
15. Liu, G.-P.; Xia, Y.-Q.; Chen, J.; Rees, D.; Hu, W.-S. Networked predictive control of systems with random network delays in both forward and feedback channels. *IEEE Trans. Ind. Electron.* **2007**, *54*, 1282–1297.
16. Tran, Q.N.; Özkan, L.; Backx, A. Generalized predictive control tuning by controller matching. *J. Process Control* **2015**, *25*, 1–18.
17. Tian, Z.-D.; GAO, X.-W.; Li, K. Networked control system time-delay compensation algorithm based on predictive control. *Syst. Eng. Electron.* **2013**, *35*, 2165–2169. (In Chinese)
18. Pan, J.-S.; Hong, M.-Z.; Zhou, Q.-F.; CAI, J.-Y.; Wang, H.-Z.; Luo, L.-K.; Yang, D.-Q.; Dong, J.; Shi H.-X.; Ren, J.-L. Integrated application of uniform design and least-squares support vector machines to transfection optimization. *BMC Biotechnol.* **2009**, *9*, 52−57.
19. Ayat, N.E.; Cheriet, M.; Suen, C.-Y. Automatic model selection for the optimization of the SVM kernels. *Pattern Recognit.* **2005**, *38*, 1733−1745.
20. Qi, H.-L.; Zhao, H.; Liu, W.-W.; Zhang, H.-B. Parameters optimization and nonlinearity analysis of grating eddy current displacement sensor using neural network and genetic algorithm. *J. Zhejiang Univ. Sci. A* **2009**, *10*, 1205−1212.
21. Xiang, C.-S.; Zhou, Z.-Y. A joint optimization algorithm and its application in chaotic time series prediction. *J. Cent. South Univ. (Sci. Technol.)* **2012**, *43*, 581–587.