Proceedings of the 2007 IEEE Symposium on
Computational Intelligence and Data Mining (CIDM 2007)

Time Series Forecasting Using Multiple Gaussian Process Prior Model

Tomohiro Hachino
Department of Electrical and
Electronics Engineering
Kagoshima University
Kagoshima, 890–0065 Japan
Telephone: +81 99 285 8392
Fax: +81 99 285 8392
Email: hachino@eee.kagoshima-u.ac.jp

Visakan Kadirkamanathan
Department of Automatic Control and
Systems Engineering
The University of Sheffield
Sheffield, S1 3JD, UK
Telephone: +44 114 222 5618
Fax: +44 114 222 5661
Email: visakan@sheffield.ac.uk

Abstract— **Using historical data to forecast future trends in time series is a key application of data mining. This paper deals with the problem of time series forecasting using the non-parametric Gaussian process model. The time series forecasting is accomplished by using multiple Gaussian process models of each step ahead predictor in accordance with the direct approach. The separable least-squares approach is applied to train these Gaussian process models. Hyperparameters of the covariance function are coded into binary bit strings and candidate weighting parameters of the mean function corresponding to each candidate of hyperparameters are estimated by the linear least-squares method. The genetic algorithm is utilized to determine these unknown hyperparameters by minimizing the negative log marginal likelihood of the training data. Simulation results are shown to illustrate the proposed forecasting method and compared with the iterated prediction method.**

I. Introduction

One of the key applications in data mining is the use of historical data for the forecasting of the future. This is particularly important in cases where the data are dynamic and there is little understanding of the underlying process generating the data. Applications in areas such as finance, time series forecasting is a key problem area and where knowledge of underlying models are scarce. The focus of this paper is to address this problem with an emerging machine learning methodology based on the Gaussian process (GP) framework.

In recent years many approaches for multi-step ahead prediction in time series analysis have been proposed by using GP model [1]–[7]. GP model is a non-parametric model and fits naturally into the Bayesian framework. The model was originally utilized for the regression problem by O'Hagan [8] and has recently received much attention for both regression and classification problems [1], [9]. It gives us not only the mean value but also the variance of the conditionally expected value of the output, which is used as a measure of confidence in the predicted output.

There are two approaches to multi-step ahead prediction in time series. One is the direct method which makes multi-step ahead prediction directly, and the other is the iterated method which repeats one-step ahead prediction up to the desired step. Girard *et al* [4] proposed iterated multi-step ahead predictions with propagation of the prediction uncertainty. In [6], [7], dynamic model identifications using GP prior model were proposed and the estimated models were validated by iterated multi-step ahead predictions both with and without propagation of uncertainty. Moreover some model predictive control designs based on iterated multi-step ahead prediction by GP model have been presented in [10], [11]. Although the iterated prediction method is attractive in the context of control problems such as model predictive control, unacceptable prediction errors are gradually accumulated as the prediction step progresses.

In this paper, we propose the direct method for time series forecasting by using the GP framework. For each step ahead GP prior is trained by minimizing the negative log marginal likelihood of the training data. The time series forecasting is directly performed by using every trained Gaussian process model of each step ahead predictor. The GP model has fewer parameters called hyperparameters compared to parametric models such as neural network models and fuzzy models, but this optimization still suffers from the local minima problem. Therefore in this paper, the training is carried out by the genetic algorithm (GA), which has a high potential for finding global optima [12]. In the case when the prior mean is assumed to be represented by a linear combination of the input variables, the weighting parameters for prior mean and hyperparameters of covariance functions can be estimated separately by using the separable least-squares approach [13]. The separable least-squares method has been utilized for linear space model identification [14], bilinear model identification [15] and nonlinear parametrically varying model identification [16]. In our method, the hyperparameters of covariance functions are coded into binary bit strings in GA, and the weighting parameters of the prior mean function corresponding to each candidate hyperparameter, are estimated by the linear least-squares method.

This paper is organized as follows. In section II, the time series forecasting problem is formulated. In section III, the GP prior model is reviewed. In section IV, the training method of

1-4244-0705-2/07/$20.00 ©2007 IEEE

the GP prior model based on separable least-squares approach is proposed including the use of the GA. In section V, the time series forecasting method using the direct approach is presented. In section VI, simulation results are shown to illustrate the effectiveness of the proposed forecasting method and compared to the iterated prediction method. Finally conclusions are given in section VII.

II. Statement of the Problem

The time series data are represented by

$$Y_{1:T} = \{y(1), y(2), \cdots, y(k), y(k+1), \cdots, y(T)\}. \tag{1}$$

The time series forecasting problem is to estimate

$$\hat{Y}_{k+1:k+M} = \{\hat{y}(k+1), \hat{y}(k+2), \cdots, \hat{y}(k+M)\}, \tag{2}$$

given the past data $Y_{1:k}$. Each of the estimates $\hat{y}(k+j)$ can be thought of as a solution of multi-step ahead time series prediction problem. An optimal predictor for $\hat{y}(k+j)$ is given by

$$\hat{y}(k+j) = \mathrm{E}[y(k+j)|y(k), y(k-1), \cdots, y(1)], \tag{3}$$

where $\mathrm{E}[\cdot]$ is the expectation operator.

Typically, the time series data are assumed to be generated by an underlying dynamic model such that the optimal predictor can be rewritten as

$$\hat{y}(k+j) = \mathrm{E}[y(k+j)|\boldsymbol{x}(k)] = g_j(\boldsymbol{x}(k)), \tag{4}$$

where $\boldsymbol{x}(k) = [y(k), y(k-1), \cdots, y(k-L+1)]^{\mathrm{T}}$ is the vector consisting of the L most recent observations of the time series and is linked to the order of the dynamic system.

The problem of time series forecasting then requires the construction of multiple time step ahead predictors $g_j(\boldsymbol{x}(k))$ for which a GP framework is adopted here.

III. Gaussian Process Prior Model

A Gaussian Process (GP) is a Gaussian random function and is completely described by its mean function and covariance function. We can regard it as a collection of random variables which has joint multivariable Gaussian distribution, i.e. for any N, we have

$$f(\boldsymbol{x}_1), f(\boldsymbol{x}_2), \cdots, f(\boldsymbol{x}_N) \sim \mathcal{N}(\boldsymbol{m}(\boldsymbol{X}), \boldsymbol{\Sigma}), \tag{5}$$

where $\boldsymbol{x}_i = \boldsymbol{x}(i+L)$, $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_N]$, $\boldsymbol{m}(\boldsymbol{X})$ is the mean function and $\boldsymbol{\Sigma}$ is the covariance matrix. The mean function $\boldsymbol{m}(\boldsymbol{X})$ is usually represented by a linear combination of the input, i.e.

$$\boldsymbol{m}(\boldsymbol{X}) = \tilde{\boldsymbol{X}}\boldsymbol{\theta}_l \tag{6}$$

where

$$\begin{aligned} \tilde{\boldsymbol{X}} &= [\ \boldsymbol{X}^{\mathrm{T}} \ \vdots \ \boldsymbol{e}\ ] \\ \boldsymbol{e} &= [1, 1, \cdots, 1]^{\mathrm{T}}, \end{aligned} \tag{7}$$

and $\boldsymbol{\theta}_l = [\theta_{l0}, \theta_{l1}, \cdots, \theta_{lL}]^{\mathrm{T}}$ is the unknown weighting parameter vector.

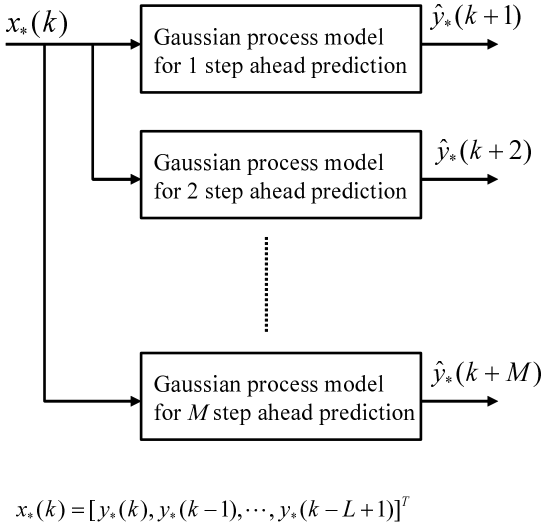
$x_*(k)$
Gaussian process model for 1 step ahead prediction
$\hat{y}_*(k+1)$
Gaussian process model for 2 step ahead prediction
$\hat{y}_*(k+2)$
Gaussian process model for M step ahead prediction
$\hat{y}_*(k+M)$
$x_*(k) = [y_*(k), y_*(k-1), \cdots, y_*(k-L+1)]^T$

Fig. 1. The proposed multi-step ahead predictor based time series forecasting scheme

The covariance $\Sigma_{pq} = Cov(f(\boldsymbol{x}_p), f(\boldsymbol{x}_q)) = C(\boldsymbol{x}_p, \boldsymbol{x}_q)$ is an element of covariance matrix $\boldsymbol{\Sigma}$ which is a function of $\boldsymbol{x}_p$ and $\boldsymbol{x}_q$. Under the assumption that the process is stationary and smooth, the following Gaussian kernel is often utilized for Σ_{pq}:

$$\Sigma_{pq} = C(\boldsymbol{x}_p, \boldsymbol{x}_q) = \sigma_y^2 \exp\left(-\frac{\|\boldsymbol{x}_p - \boldsymbol{x}_q\|^2}{2\ell^2}\right), \tag{8}$$

where $\|\cdot\|$ denotes the Euclidean norm.

In the presence of zero mean Gaussian white noise of variance σ_n^2, we have

$$y_1, y_2, \cdots, y_N \sim \mathcal{N}(\boldsymbol{m}(\boldsymbol{X}), \boldsymbol{K}), \tag{9}$$

where $y_i = y(i+L+j)$ $(j = 1, 2, \cdots, M)$, and $\boldsymbol{K} = \boldsymbol{\Sigma} + \sigma_n^2\boldsymbol{I}$ ($\boldsymbol{I}$: $N \times N$ identity matrix).

$\boldsymbol{\theta}_c = [\sigma_y, \ell, \sigma_n]^{\mathrm{T}}$ contains the hyperparameters. The overall variance of the random function can be controlled by σ_y and the characteristic length-scale of the process can be changed by ℓ.

IV. Training of Gaussian Process Prior Model

To perform time series forecasting, 1 to M step ahead prediction models are needed for the direct approach (see Fig.1). The accuracy of prediction depends on the unknown parameter vector $\boldsymbol{\theta} = [\boldsymbol{\theta}_l^{\mathrm{T}}, \boldsymbol{\theta}_c^{\mathrm{T}}]^{\mathrm{T}}$ and therefore $\boldsymbol{\theta}$ has to be optimized. In this paper, we propose a new training technique using genetic algorithm and linear least-squares method based on the idea of separable least-squares [13]. This training is carried out by minimizing the negative log marginal likelihood

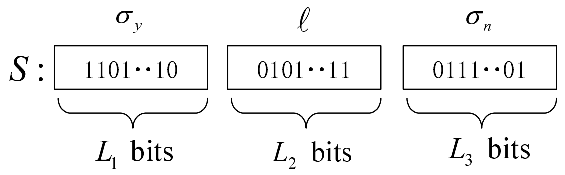

Fig. 2. Coding

of the training data:

$$\begin{aligned} J &= -\log p(\boldsymbol{y}|\boldsymbol{X},\boldsymbol{\theta}) \\ &= \frac{1}{2}\log|\boldsymbol{K}| + \frac{1}{2}(\boldsymbol{y}-\boldsymbol{m}(\boldsymbol{X}))^{\mathrm{T}}\boldsymbol{K}^{-1}(\boldsymbol{y}-\boldsymbol{m}(\boldsymbol{X})) \\ &\quad +\frac{N}{2}\log(2\pi) \\ &= \frac{1}{2}\log|\boldsymbol{K}| + \frac{1}{2}(\boldsymbol{y}-\tilde{\boldsymbol{X}}\boldsymbol{\theta}_l)^{\mathrm{T}}\boldsymbol{K}^{-1}(\boldsymbol{y}-\tilde{\boldsymbol{X}}\boldsymbol{\theta}_l) \\ &\quad +\frac{N}{2}\log(2\pi), \end{aligned} \tag{10}$$

where $\boldsymbol{y} = [y_1, y_2, \cdots, y_N]^{\mathrm{T}}$. Although this problem is a nonlinear optimization one, we can separate the linear optimization part and the nonlinear optimization part. The partial derivative of (10) with respect to the weighting parameter vector $\boldsymbol{\theta}_l$ of the mean function is as follows:

$$\frac{\partial J}{\partial \boldsymbol{\theta}_l} = -\tilde{\boldsymbol{X}}^{\mathrm{T}}\boldsymbol{K}^{-1}\boldsymbol{y} + \tilde{\boldsymbol{X}}^{\mathrm{T}}\boldsymbol{K}^{-1}\tilde{\boldsymbol{X}}\boldsymbol{\theta}_l. \tag{11}$$

Note that if the hyperparameter $\boldsymbol{\theta}_c$ of the covariance function is given, then the weighting parameter $\boldsymbol{\theta}_l$ can be estimated by linear least-squares method from (11),

$$\boldsymbol{\theta}_l = (\tilde{\boldsymbol{X}}^{\mathrm{T}}\boldsymbol{K}^{-1}\tilde{\boldsymbol{X}})^{-1}\tilde{\boldsymbol{X}}^{\mathrm{T}}\boldsymbol{K}^{-1}\boldsymbol{y}. \tag{12}$$

However even if the weighting parameter vector $\boldsymbol{\theta}_l$ is known, the optimization with respect to hyperparameter vector $\boldsymbol{\theta}_c$ is a complicated nonlinear problem and might suffer from the local minima problem. Therefore only the hyperparameter vector $\boldsymbol{\theta}_c$ of the covariance is coded into binary bit strings as shown in Fig.2 and searched by the GA which has a high potential for global optimizations [12].

σ_y is decoded logarithmically as follows:

$$\begin{aligned} &\sigma_y = 10^r \\ &r = \frac{\log_{10}\sigma_{y,max} - \log_{10}\sigma_{y,min}}{2^{L_1}-1}\mathcal{R} + \log_{10}\sigma_{y,min}, \end{aligned} \tag{13}$$

where $\mathcal{R}$ is the decimal value of the binary representation of the first block of the string S and $[\sigma_{y,min}, \sigma_{y,max}]$ is the search range of σ_y. ℓ and σ_n are also decoded logarithmically in the same manner.

The proposed training algorithm is as follows:

step 1: Initialization for training

Set $j = 1$ and let the training input data be $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_N]$.

step 2: Preparation of training output data

Let the training output data be $\boldsymbol{y} = [y_1, y_2, \cdots, y_N]^{\mathrm{T}}$, where $y_i = y(i + L + j)$.

step 3: Initialization for GA

Generate an initial population of Q binary bit strings for $\boldsymbol{\theta}_c$ randomly.

step 4: Decoding

Decode Q strings into real values $\hat{\boldsymbol{\theta}}_{c,i}$ $(i = 1, 2, \cdots, Q)$ by above mentioned decoding method.

step 5: Construction of covariance matrix

Construct Q candidates of the covariance matrix $\boldsymbol{K}_i$ using $\hat{\boldsymbol{\theta}}_{c,i}$ $(i = 1, 2, \cdots, Q)$.

step 6: Estimation of $\boldsymbol{\theta}_l$

Estimate Q candidates of $\hat{\boldsymbol{\theta}}_{l,i}$ corresponding to $\hat{\boldsymbol{\theta}}_{c,i}$ $(i = 1, 2, \cdots, Q)$ from (12)

step 7: Fitness value calculation

Calculate the negative log marginal likelihood of the training data:

$$\begin{aligned} J_i &= -\log p_i(\boldsymbol{y}|\boldsymbol{X},\hat{\boldsymbol{\theta}}_i) \\ &= \frac{1}{2}\log|\boldsymbol{K}_i| + \frac{1}{2}(\boldsymbol{y}-\tilde{\boldsymbol{X}}\hat{\boldsymbol{\theta}}_{l,i})^{\mathrm{T}}\boldsymbol{K}_i^{-1}(\boldsymbol{y}-\tilde{\boldsymbol{X}}\hat{\boldsymbol{\theta}}_{l,i}) \\ &\quad +\frac{N}{2}\log(2\pi) \qquad (i = 1, 2, \cdots, Q) \end{aligned} \tag{14}$$

and the fitness values $F_i = D - J_i$ using $\hat{\boldsymbol{\theta}}_i = [\hat{\boldsymbol{\theta}}_{l,i}^{\mathrm{T}}, \hat{\boldsymbol{\theta}}_{c,i}^{\mathrm{T}}]^{\mathrm{T}}$, where D is a positive constant value.

step 8: Reproduction

Reproduce each of individual strings with the probability of $F_i / \sum_{j=1}^{Q} F_j$.

step 9: Crossover

Pick up two strings randomly and decide whether or not to cross them over according to the crossover probability P_c. Exchange strings at a crossing position if the crossover is required. The crossing position is chosen randomly.

step 10: Mutation

Alter a bit (0 or 1) of string according to the mutation probability P_m.

step 11: Repetition for GA

Repeat **step 4** ~ **step 10** from generation to generation so that the fitness value of the population increases. In simulations, the genetic operations will be repeated until prespecified G-th generation.

step 12: Determination of the GP prior model

Construct the suboptimal prior mean and prior covariance for the j step ahead predictor by using the string with the best fitness value over all the past generations:

$$\boldsymbol{m}(\boldsymbol{x})_j = [\boldsymbol{x}^{\mathrm{T}}, 1]\hat{\boldsymbol{\theta}}_{l,best} \tag{15}$$

$$\begin{cases} C(\boldsymbol{x}_p, \boldsymbol{x}_q)_j = \hat{\sigma}_{y,best}^2 \exp\left(-\dfrac{||\boldsymbol{x}_p - \boldsymbol{x}_q||^2}{2\hat{\ell}_{best}^2}\right) \\ K(\boldsymbol{x}_p, \boldsymbol{x}_q)_j = C(\boldsymbol{x}_p, \boldsymbol{x}_q)_j + \hat{\sigma}_{n,best}^2\delta_{pq}, \end{cases} \tag{16}$$

where $K(\boldsymbol{x}_p, \boldsymbol{x}_q)$ is an element of covariance matrix $\boldsymbol{K}$ and δ_{pq} is a Kronecker delta which is 1 if $p = q$ and 0 otherwise.

step 13: Repetition for the GP prior model

If $j < M$ then $j = j + 1$ and go to **step 2**.

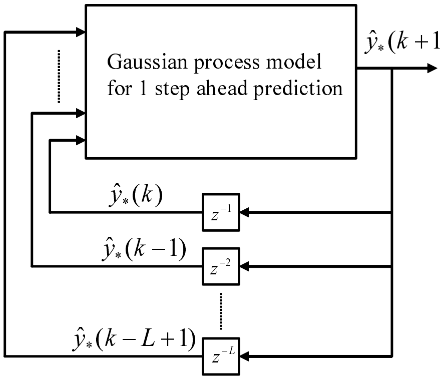

Fig. 3. The iterated multi-step ahead prediction scheme

V. Multi-step Ahead Prediction

For a new given test input $\boldsymbol{X}_* = [\boldsymbol{x}_{*1}, \boldsymbol{x}_{*2}, \cdots, \boldsymbol{x}_{*\tilde{N}}]^{\mathrm{T}}$, we have

$$\begin{bmatrix} \boldsymbol{y} \\ \boldsymbol{y}_* \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \boldsymbol{m}(\boldsymbol{X}) \\ \boldsymbol{m}(\boldsymbol{X}_*) \end{bmatrix}, \begin{bmatrix} \boldsymbol{K} & \boldsymbol{\Sigma}(\boldsymbol{X}, \boldsymbol{X}_*) \\ \boldsymbol{\Sigma}(\boldsymbol{X}_* \boldsymbol{X}), & \boldsymbol{\Sigma}(\boldsymbol{X}_*, \boldsymbol{X}_*) \end{bmatrix} \right). \tag{17}$$

From the formula for conditioning a joint Gaussian distribution, the posterior distribution for a specific test data is

$$p(\boldsymbol{y}_* | \boldsymbol{X}, \boldsymbol{y}, \boldsymbol{X}_*) \sim \mathcal{N}(\bar{\boldsymbol{y}}_*, cov(\boldsymbol{y}_*)), \tag{18}$$

where $\bar{\boldsymbol{y}}_*$ is the predictive mean, $cov(\boldsymbol{y}_*)$ is the predictive covariance, and is given by,

$$\begin{aligned} \bar{\boldsymbol{y}}_* &= \boldsymbol{m}(\boldsymbol{X}_*) + \boldsymbol{\Sigma}(\boldsymbol{X}_*, \boldsymbol{X}) \boldsymbol{K}^{-1} (\boldsymbol{y} - \boldsymbol{m}(\boldsymbol{X})) \\ cov(\boldsymbol{y}_*) &= \boldsymbol{\Sigma}(\boldsymbol{X}_*, \boldsymbol{X}_*) - \boldsymbol{\Sigma}(\boldsymbol{X}_*, \boldsymbol{X}) \boldsymbol{K}^{-1} \boldsymbol{\Sigma}(\boldsymbol{X}, \boldsymbol{X}_*) + \sigma_n^2 \boldsymbol{I}. \end{aligned} \tag{19}$$

In section IV, we obtained GP prior models for j ($j = 1, 2, \cdots, M$) step ahead predictors. Using these models the forecasting up to M step is carried out from (19), where the number of the test input data $\tilde{N} = 1$ and the test input is set to be vector $\boldsymbol{x}_*(k) = [y_*(k), y_*(k-1), \cdots, y_*(k-L+1)]^{\mathrm{T}}$ (see Fig.1).

This direct multi-step ahead prediction method is comparable to iterated multi-step ahead prediction which repeats one step ahead prediction with feedback of predicted previous output [3], [4], [6], [7], [10], [11] (see Fig.3). The advantage with the iterated method is that only one GP model for one step ahead prediction is required, and is therefore computationally efficient. The direct approach on the other hand has the potential for increased accuracy in predictions. In the next section some simulation results by our forecasting method will be given and compared with the iterated prediction method.

VI. Simulations

A. Example 1

Consider the following system:

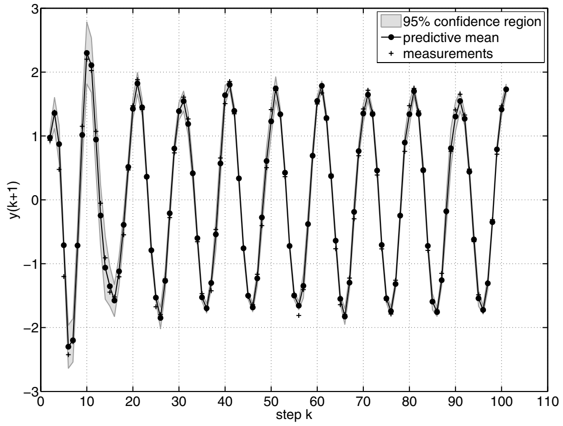

Fig. 4. Prediction result for 1 step ahead predictor (example 1)

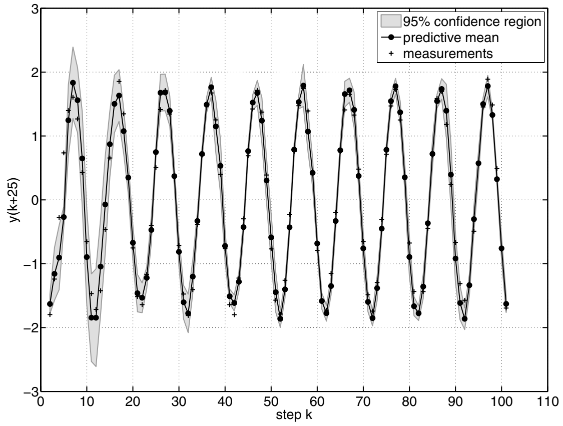

Fig. 5. Prediction result for 25 step ahead predictor (example 1)

$$\begin{cases} y(k+1) = a_1 y(k) + a_2 y(k-1) + \sin(2\pi f k) + e(k) \\ a_1 = 0.9, \ a_2 = -0.8, \ f = 0.1, \quad e(k) : N(0, 0.05^2). \end{cases} \tag{20}$$

The training input is chosen as $\boldsymbol{x}(k) = [y(k), y(k-1)]^{\mathrm{T}}$ ($L = 2$ in (4)) and the time series forecasting up to $M = 50$ step is carried out from the starting step $k = k_0 = 53$. The number of the training input and output data is taken to be $N = 100$ for each j ($j = 1, 2, \cdots, M$) step ahead predictor.

The design parameters of the GA are given as follows:

population size: $Q = 100$
string length: $L_1 = L_2 = L_3 = 10$
crossover probability: $P_c = 0.8$
mutation probability: $P_m = 0.03$
search range of σ_y: $[\sigma_{y,min}, \sigma_{y,max}] = [10^{-3}, 10]$
search range of ℓ: $[\ell_{y,min}, \ell_{y,max}] = [10^{-3}, 10]$
search range of σ_n: $[\sigma_{n,min}, \sigma_{n,max}] = [10^{-6}, 10]$
termination criteria G = 100-th generation

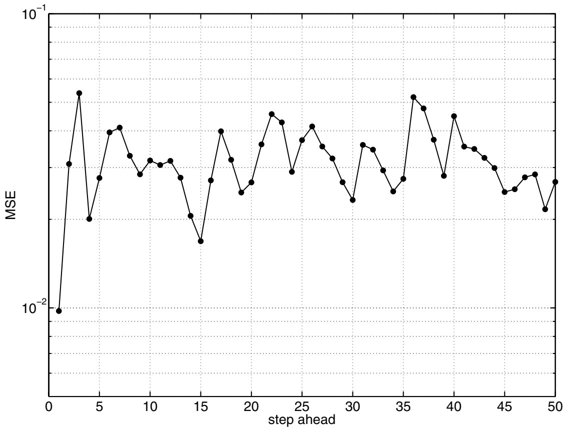

Fig. 6. Mean square error against step ahead (example 1)

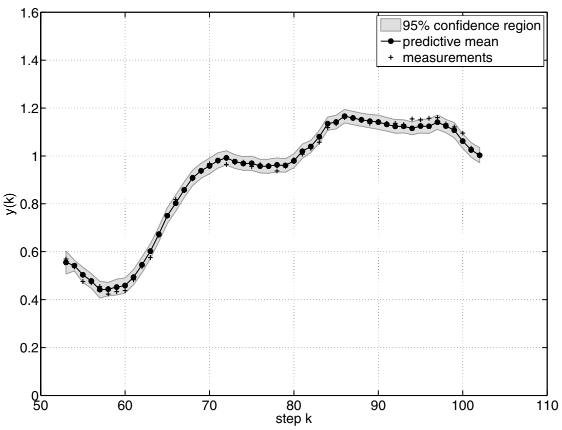

Fig. 8. Forecasting result by the proposed method (example 2)

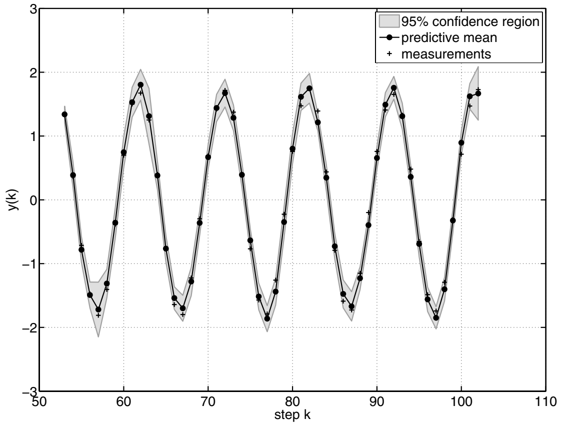

Fig. 7. Forecasting result (example 1)

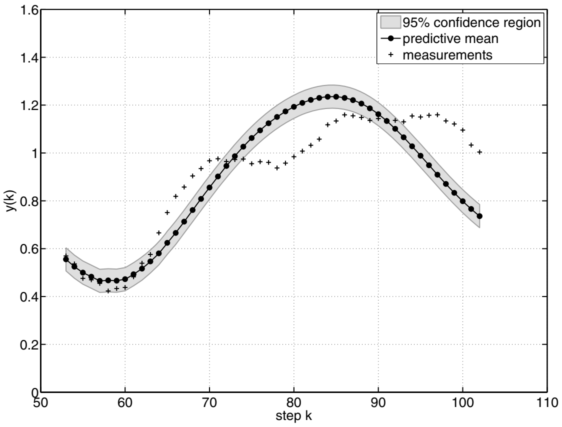

Fig. 9. Forecasting result by the iterated method (example 2)

To validate our training method, the prediction results for 1 and 25 step ahead predictors are shown in Figs.4 and 5, where the number of the test data is $\tilde{N} = 100$. In these figures, the cross symbol shows the test output, the circle symbol with line shows the predictive mean and the shaded area gives the 95% confidence region. Moreover the mean square error MSE=$(\sum_{k=2+j}^{1+j+\tilde{N}}(y_*(k)-\hat{y}_*(k))^2)/\tilde{N}$ against the step ahead j $(j = 1, 2, \cdots, M)$ is depicted in Fig.6. From Figs.4 ~ 6 we can confirm that the error between the test data and the predictive mean is quite small for every step ahead predictors and it does not become large as the step ahead increases.

Fig.7 shows the result of the time series forecasting from the starting step k_0 to $k_0 + M - 1$ by the proposed method. The mean square error $(\sum_{k=k_0}^{k_0+M-1}(y_*(k)-\hat{y}_*(k))^2)/M$ of the time series forecasting is 8.3478×10^{-3}. It is clear that we can carry out the time series forecasting successfully by the proposed method.

B. Example 2

Consider the following system called Mackey-Glass chaotic system which has high nonlinearity:

$$\begin{cases} \dfrac{dz(t)}{dt} = -a_1 z(t) + a_2 \dfrac{z(t-\tau)}{1+z(t-\tau)^{10}} \\ y(t) = z(t) + e(t) \\ a_1 = 0.1,\ a_2 = 0.2,\ \tau = 17,\quad e(t) : N(0, 0.01^2). \end{cases} \tag{21}$$

The time series is sampled with sampling period $T_s = 1$ as $y(k) = y(t = k)$.

The training input is again chosen as $\boldsymbol{x}(k) = [y(k), y(k-1), \cdots, y(k-15)]^{\mathrm{T}}$ ($L = 16$ in (4)). The number of the training input and output data is taken to be $N = 100$ for each j $(j = 1, 2, \cdots, M)$ step ahead predictor.

Fig.8 shows the forecasting result by the proposed method, where the time series forecasting up to $M = 50$ step is carried out from the starting step $k = k_0 = 53$. For comparison the forecasting result by the iterated multi-step ahead prediction is shown in Fig.9. The mean square error $(\sum_{k=k_0}^{k_0+M-1}(y_*(k) - \hat{y}_*(k))^2)/M$ of the time series forecasting is 2.4317×10^{-4} for the proposed method and 2.0139×10^{-2} for the iterated method, respectively. In the iterated method the error of the predicted values increases as the prediction step progresses as shown in Fig.9. On the other hand the proposed method gives accurate predicted values and reasonable confidence region for all step ahead predictions as shown in Fig.8. It is found from these results that the proposed method can be also applied to nonlinear time series forecasting.

VII. Conclusions

In this paper, we have presented a direct approach to the time series forecasting using the GP prior model. The time series forecasting is carried out directly by using every Gaussian process model of each step ahead predictor. Based on the idea of separable least-squares approach, a new training algorithm for GP prior model using GA has been proposed. Although the proposed prediction method is rather computationally demanding in the training, the prediction error is not accumulated as the prediction step increases. In addition, both the predictive mean and the predictive covariance can be directly obtained without any modifications of the prediction algorithm. Simulation results show that the proposed method can be applied to the time series forecasting with high accuracy for both linear and nonlinear time series.

References

[1] C. K. I. Williams, "Prediction with Gaussian Processes: From Linear Regression to Linear Prediction and Beyond", in *Learning and Inference in Graphical Models*, Kluwer Academic Press, pp.599–621, 1998.

[2] R. Murray-Smith and A. Girard, "Gaussian Process Priors with ARMA Noise Models", *Proc. of the Irish Signals and Systems Conference*, pp.147–153, 2001.

[3] G. Gregorčič and G. Lightbody, "Gaussian Processes for Modelling of Dynamic Non-linear Systems", *Proc. of the Irish Signals and Systems Conference*, pp.141–147, 2002.

[4] A. Girard, C. E. Rasmussen, J. Q. Candela and R. Murray-Smith, "Gaussian Process Priors with Uncertain Inputs -Application to Mutiple-Step Ahead Time Series Forecasting", in *Advances in Neural Information Processing Systems*, vol.15, pp.542–552, MIT Press, 2003.

[5] S. Brahim-Belhouari and A. Bermak, "Gaussian Process for Nonstationary Time Series Prediction", *Computational Statistics & Data Analysis*, vol.47, pp.705–712, 2004.

[6] K. Ažman and J. Kocijan, "An Example of Gaussian Process Model Identification", *Proc. of 28th International Convention MIPRO, CIS-Intelligent Systems*, pp.79–84, 2005.

[7] J. Kocijan, A. Girard, B. Banko and R. Murray-Smith, "Dynamic Systems Identification with Gaussian Processes", *Mathematical and Computer Modelling of Dynamical Systems*, vol.11, no.4, pp.411-424, 2005.

[8] A. O'Hagan, "Curve Fitting and Optimal Design for Prediction (with discussion)", *Journal of the Royal Statistical Society B*, vol.40, pp.1–42, 1978.

[9] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*, MIT Press, 2006.

[10] R. Murray-Smith, D. Sbarbaro, C. E. Rasmussen and A. Girard, "Adaptive, Cautious, Predictive Control with Gaussian Process Priors", *Proc. of the 13th IFAC Symposium on System Identification*, pp.1195–1200, 2003.

[11] J. Kocijan, R. Murray-Smith, C. E. Rasmussen and A. Girard, "Gaussian Process Model Based Predictive Control", *Proc. of American Control Conference 2004*, pp.2214–2219, 2004.

[12] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley Publishing Company, Inc., 1989.

[13] G. H. Golub and V. Pereyra, "The Differentiation of Pseudo-inverses and Nonlinear Least Squares Problems Whose Variables Separate", *SIAM Journal of Numerical Analysis*, vol.10, no.2, pp.413–432, 1973.

[14] J. Bruls, C. T. Chou, B. R. J. Haverkamp and M. Verhaegen, "Linear and Non-linear System Identification Using Separable Least-Squares", *European Journal of Control*, vol.5, pp.116–128, 1999.

[15] V. Verdult and M. Verhaegen, "Identification of Multivariable Bilinear State Space Systems Based on Subspace Techniques and Separable Least Squares Optimization", *International Journal of Control*, vol.74, no.18, pp.1824–1836, 2001.

[16] F. Previdi and M. Lovera, "Identification of Non-linear Parametrically Varying Models Using Separable Least Squares", *International Journal of Control*, vol.77, no.16, pp.1382–1392, 2004.