Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

Nonlinear System Modeling by Hybrid Genetic Programming

Nozomi Hashimoto * **Nobuhiko Kondo** **
Toshiharu Hatanaka *** **Katsuji Uosaki** ****

* *Department of Information and Physical Sciences, Osaka University, Suita, Japan; e-mail:n-hashimoto@ist.osaka-u.ac.jp)*
** *Department of Information and Physical Sciences, Osaka University, Suita, Japan; e-mail:nobuhiko@ist.osaka-u.ac.jp)*
*** *Department of Information and Physical Sciences, Osaka University, Suita, Japan; e-mail:hatanaka@ist.osaka-u.ac.jp*
**** *Department of Manegement and Information Sciences, Fukui University of Technology, Fukui, Japan, (e-mail: uosaki@fukui-ut.ac.jp)*

Abstract: Genetic Programming (GP) is a useful tool of nonlinear model building, however a simple use of GP often fails in numeric optimization since GP hangs on random number sampling in searching appropriate constant parameters in individual representing each model candidate. From this viewpoint a hybrid GP based nonlinear system identification method is proposed in this paper. We introduce a simple numerical optimization inspired by Particle swarm in GP operation to improve numeric optimization ability. Then, this hybridization is applied to nonlinear system identification by using GP. The applicability of the proposed method is shown by the results of some numerical experiments.

Keywords: Nonlinear system identification, model structure determination, evolutionary computation, genetic programming.

1. INTRODUCTION

The black box modeling based on the observed input and output data is called system identification. Such black box model of an actual system has an important role in many engineering problems, such as control system design, time series prediction, fault detection and diagnosis and so on. The system identification technique has been developed for the stochastic systems under the assumption of linearity and normal distribution Ljung [1999]. However, almost systems have inherent nonlinear property. The linear system model is not enough to use as a model of such nonlinear system, hence nonlinear system identification methodologies have been studied Olover [2000].

Evolutionary computation, that is generic term of commutation method inspired by organic evolution, is one of the practical ways to nonlinear system optimization and adaptation DeJong [2006]. Hence applications of evolutionary computations to the control engineering field are receiving much attention recently. For example genetic algorithms are applied to optimization of PID control design Mitsukura [1999] and applied to nonlinear system parameter estimation Yao [1994].

Genetic Programming (GP) proposed by Koza is one of the evolutionary computations. It is able to handle structural expression such as hierarchy structures and tree structures Koza [1992]. This means that GP will be able to an autonomous structure selection in the nonlinear system identification, then GP is introduced to the structure determination in nonlinear system identification Gary [1998], Katya [1997], Marenbach [1997] and Uosaki [2000].

However, a simple use of GP tends to give too complex model structures to design control systems. From this point of view, Hashimoto et. al. proposed GP based local modeling Hashimoto [2007], in order to build an identification algorithm that is able to give a precise model but rather than simple representation for the nonlinear systems. This approach is consisted by the clustering of the regression vectors, local system identification by using GP and model fusion of these local models by fuzzy inference to provide one global model. And also a simple use of GP often failed in numeric optimization since GP hangs on random number generation in searching appropriate constant parameters in individual representing each model candidate. It becomes difficult to give a precise model for the complicated system having plural parameters. From this view point, Hatanaka and Uosaki proposed hybrid GP in which the least square is used in numerical optimization, and applied this approach to Hammerstein model identification Hatanaka [2001]. However, applicability of this approach was limited, since this approach is able to apply only a model linear in parameters.

In this paper, we propose an identification technique by GP in combination with numerical adaptation mechanism inspired by particle swarm proposed by Kennedy [1995]. Then the proposed method is applied to GP based local modeling proposed by Hashimoto [2007]. The applicability

of the proposed method is illustrated by the numerical simulation results.

This paper is organized as follows. In section 2 the problem statement is described and in section 3 GP is briefly introduced. GP based system modeling and the proposed method are presented in section 4. The numerical simulation results are shown in section 5. Then, in the last section we mention about concluding remarks and future works.

2. PROBLEM STATEMENT

Let assume that the objective unknown nonlinear dynamic systems is able to be described by the following NARX (Nonlinear Auto-Regressive model with eXogenous input) model.

$$y(t) = g(\phi(t)) + e(t). \quad (1)$$

Here, $g(\cdot)$ is an unknown nonlinear function, $u(t)$ and $y(t)$ denote the input and the output of the system at the time instant t, respectively. An observation noise is represented by $e(t)$. Though the assumption of normal distribution is often presented, In this approach there are no assumptions about probabilistic distribution. However, for simplify we assume that $e(t)$ is identically indecently distributed with 0 mean distribution.

In addition,

$$\phi(t) = (y(t-1), \ldots, y(t-n_y), u_t, \ldots, u(t-n_u))^T$$

is a regression vector consisting of the observed input and output signals. n_u and n_y are the maximum delay, and they are unknown in general.

Then, the identification problem considering in this paper is evolving GP population appropriately to fit observed input and output data and giving the elitist as a identified system model.

3. GENETIC PROGRAMMING

The evolutionary computation is multi points search method based on some kind of probabilistic methodology. It is considered as a kind of meta-heuristics. There are several variations of evolutionary computation. Each evolutionary computation technique is independently developed by some researchers. For example, well known Genetic Algorithms is originated in Hollamd [1975] and Evolution Strategies is developed by Rechenberg [1994]. And then it is pointed out that these techniques have in common, so they became collectively called evolutionary computations and became to be free from boundaries.

The common procedure of evolutionary computation is described in Fig.1. The search process based on the organic evolution such as principle of the survival of the fittest and it commonly used two genetic operations of crossover and mutation are iterated until some terminate conditions are satisfied. The crossover operation makes two offspring from the selected two parents, and the mutation operation brings variation into individual to maintain diversity of the population.

Genetic programming(GP) was developed by Koza to evolve an executable computer programs. While a standard genetic algorithm(GA) deals with data represented by fixed length strings over some finite characters, the

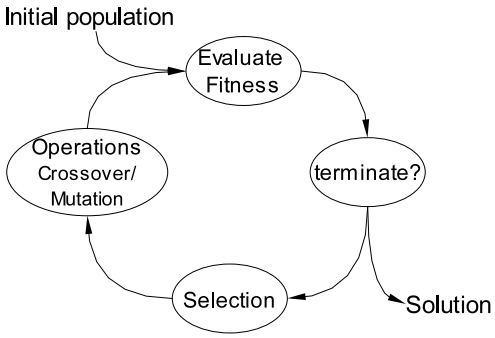

Fig. 1. The procedure of evolutionary computation.

individuals in GP are the expression trees and it can be used for structural optimization rather than numerical parameter optimization.

For example, the polynomial

$$y_t = R_1 x_t + (R_2 - x_t^2)$$

can be written as an expression tree in Fig.7.

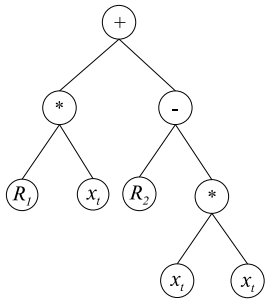

Fig. 2. An example of the individual in genetic programming. This corresponds to $R_1 x_t + (R_2 - x_t^2)$

The elements of trees are called nodes. The nodes are classified into nonterminal nodes and terminal nodes.

- Nonterminal nodes(function nodes) consume one or more inputs and provide one output(e.g. + and $*$ in the Fig.2).
- Terminal nodes represent external inputs or constants. These are the leaves of the expression trees.

The evolution in GP proceeds in similar way to standard GA, i.e. an initial population is generated at random and each individuals is evaluated its fitness value, then it is evolved by means of genetic operators as follows:

- Selection: Pairs of parent trees are selected based on its fitness for reproduction.
- Crossover: This process is performed by selecting a node at random then exchanging the associated subtrees to produce a pair of offspring trees.
- Mutation: This is performed by either replacing a node selected at random with its associated subtrees generated randomly or changing its type.
- Inversion: Inversion is performed by exchanging two nodes which are located in the same level in the genetic programming tree.

System identification by using GP has been studying from the early time of GP. Most of them focused on its ability of automatic function generation. For example, R. V. Katya

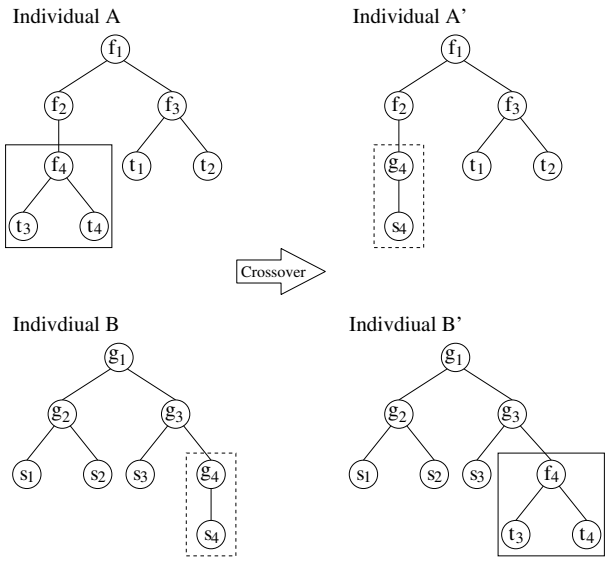

Fig. 3. Crossover in genetic programming

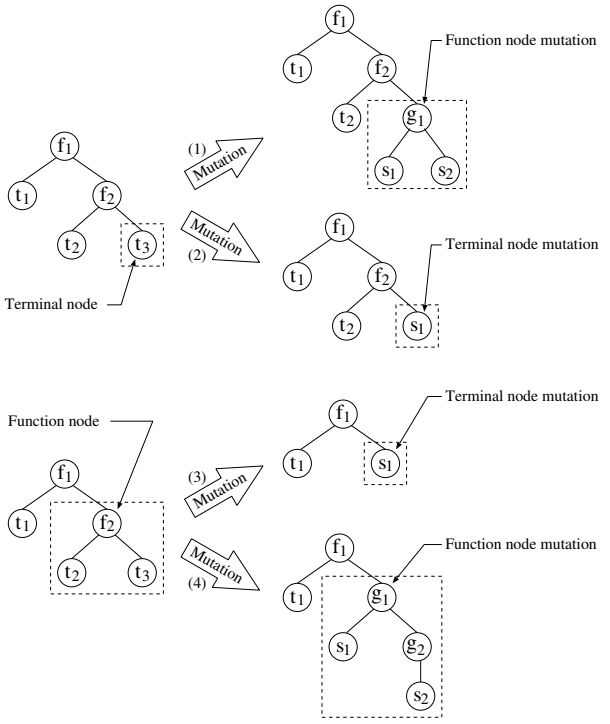

Fig. 4. Mutation in genetic programming

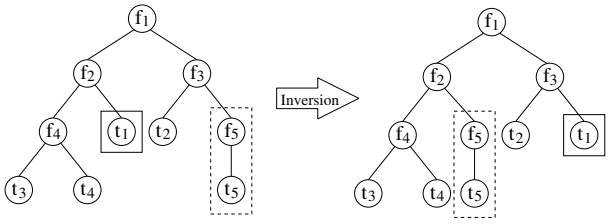

Fig. 5. Inversion in genetic programming

employed GP as a nonlinear polynomials system identification method Katya [1997]. She also proposed multiobjective identification of nonlinear system Katya [2004]. On the other hand, Hashimoto et. al. proposed GP based local modeling Hashimoto [2007]. This approach uses GP as a local nonlinear model builder. The identification problem is to build system model as following weighted sum of local nonlinear models,

$$\hat{y}(t) = \sum_{i=1}^{K} \psi_i(\phi(t)) f_i(\phi(t)), \tag{2}$$

where, $\psi_i(\phi_t), i = 1, \ldots, K$ are coefficients decided based on the membership functions, that is determined by the clustering results by SOM. $f_i(\phi_t), i = 1, \ldots, K$ represent each local model built by using GP. To divide the whole operating regime into several small ones, it needs some clustering technique. Hashimoto et. al. employed Self Organization Map (SOM) which inputs are regression vectors for clustering tool. And then, the fuzzy membership functions are used to merge local models to one global model. The fuzzy membership functions are selected by the prior knowledge and the trained SOM.

However, the parameter estimation step in the system identification procedure is much important in order to give an accurate model. A simple use of GP is not good from this point of view, because numerical search in GP depends on random sampling. In this paper, we propose a hybrid GP, in which a simplified Particle Swarm Optimization is introduced in GP operations.

4. NONLINEAR SYSTEM IDENTIFICATION BY GENETIC PROGRAMMING

In this section, we describe the procedure of GP based system identification briefly. In order to identify the nonlinear dynamic systems, it required suitable configurations of the nonterminal nodes and the terminal nodes in GP. The nonterminal nodes have arguments consisting either the other nonterminal node nor terminal node, to process at the node. The terminal nodes having no arguments denote the variables, constants including the unknown parameters of the system model. In this study, we take on the four arithmetic operands + and × as terminal nodes, see Table 1, in which the arithmetic operand has two arguments. In practice, we can select the other combination of arithmetic operands, mathematical functions, and terminal nodes elements. While we employ the constant represented by symbol R, the input symbol, and the output symbol with time delay as the terminal nodes, also see Table.1. Note that we can employ the other kind of nonterminal node such as mathematical functions i.e. $\sin, \cos, \exp, \log$, logical functions and so on. In this paper, since we focus on improving the search ability by introducing numerical optimization process in GP, we use rather simple setting of GP nodes.

Table 1. Functions for genetic programming in nonlinear model building

symbol	inputs	function
$u_t t = 1, 2, \ldots$	0	input(terminal node)
$y_t, t = 1, 2, \ldots$	0	output(terminal node)
R	0	constant(terminal node)
$+$	2	increment
$*$	2	multiplier

The fitness function is also important in order to give a 'good' model in some senses. The information criterion are

often used to give a model having generalization ability, that is good enough model not only for training data but also for unseen data set. In this study, we use AIC (Akaike Information Criteria) and MSE (Mean Squared Error) as a fitness function. In this way, model structure selection and variable selection are autonomously performed by GP.

To improve numerical search, we propose a hybrid GP by combining conventional GP with numerical optimization procedure inspired by Particle swarm. The overall flow diagram of this idea is shown in Fig.6. In this procedure,

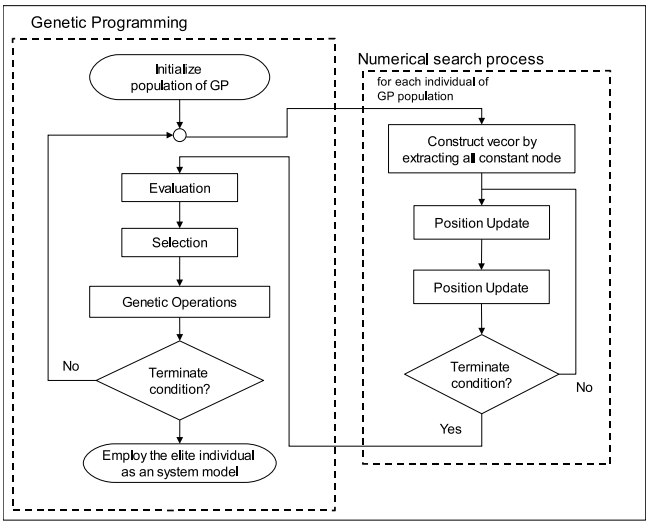

Fig. 6. The overall flow diagram of the proposed hybrid GP.

the constant nodes (represented by R in the Table.1) are extracted from each individual (note that they are symbols at the first generation), then combine them and make N number of vectors by random sampling, i.e. the instance is made at random. $\theta_i^0 = (R_1^{(i)}, R_2^{(i)}, \ldots, R_n^{(i)})^T$. Where, $i = 1, 2, \ldots, N$ denotes an individual index in this search process and n represent the number of constant node.

The population of θ_i^k is updated by the following equation, see Fig.8.

$$\nu_i^{k+1} = w\nu_i^k + c_1(\theta_i^{(k)} - p_i) + c_2(\theta_i^{(k)} - g) \quad (3)$$

$$\theta_i^{(k+1)} = \theta_i^{(k)} + \nu_i^{k+1} \quad (4)$$

Here, p_i indicates the best position found by $i-$th vector in this generation, and g indicates the best position among all vectors in this generation. The coefficient w is the constriction constant and c_1 and c_2 are the random values uniformly distributed over the range $[0, \phi]$.

After that, the elite individual of the population, denoted by g is get in GP population, then each instance of GP individual is processed by GP's genetic operation. This process is iterated until the terminate condition is satisfied. Note that the instance is kept in crossover and inversion operation, however in mutation, the instance is reinitialized by random sampling.

5. NUMERICAL EXAMPLE

To show the validity of the proposed approach to modeling nonlinear dynamic system, numerical simulation study is carried out.

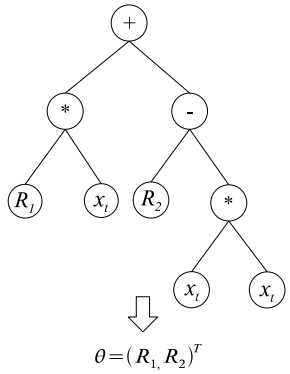

Fig. 7. Terminal nodes that represent constant, denoted by $R_i, i = 1, 2, \cdots$, are extracted from tree.

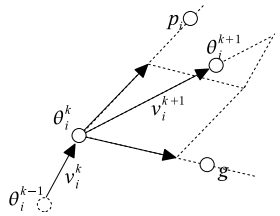

Fig. 8. An illustration of numerical search used here

Simulation 1

Consider the following system as a true system,

$$y(t) = -0.5y(t-2) + 0.7u(t-1)y(t-1) + 0.6u(t-2)^2 + 0.2y(t-1)^3 - 0.7u(t-2)^3 + e(t) \quad (5)$$

Here, $e(t)$ is a white noise normally distributed with mean zero and unknown variance and the input signal $u(t)$ is assumed random input uniformly distributed over [-1, 1].

1000 sets of input and output data were used for model construction using the proposed approach. Table 2 shows the executive settings of the proposed method. According to this setting, the constant nodes are combined into a vector, and 5 copies are generated at random. Then they are updated by (3) and (4) until 40 times iterations. The elite individual in this vector pool is returned to the original tree in the GP population for each individual in the GP population.

Table 2. GP settings

population size	50
terminate generation	200
crossover rate	0.8
mutation rate	0.5
inversion rate	0.3
the number of vectors	5
iterations	40
inertia weight w	0.8
c_1, c_2	$U[0, 0.5]$

The evolution was terminated at 200 generation, an example of the obtained models was, i.e. a example of the elite individual in the final generation was,

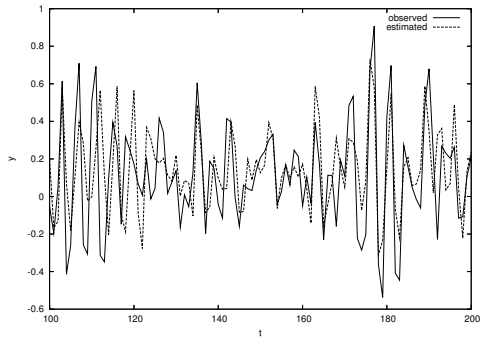

Fig. 9. Observed output data and estimated output by the proposed method

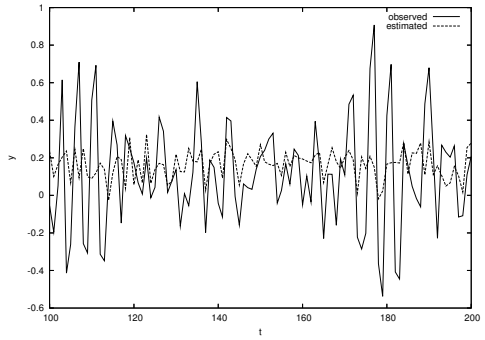

Fig. 10. Observed output data and estimated output by simple GP

$$\begin{aligned}\hat{y}(t) = ((&0.250562 * (((y(t-2) * (y(t-2) + u(t-1))) \\ &* u(t-2)) + ((0.488325 * (0.456085 + u(t-1))) \\ &+ ((2.136705 + y(t-5)) * (u(t-2) * u(t-2)))))) \\ &+ (y(t-2) * (0.646936 + (((-0.703105) \\ &* (u(t-2) * u(t-2))) + (-1.133628))))) \end{aligned} \quad (6)$$

Figure 9 shows observed output data and estimated output from $t = 100$ to $t = 200$ by the proposed method. In order to compare the proposed method, we carried out identification experiment by using the conventional GP so called simple GP.

Figure 10 shows observed output data and estimated output by conventional GP. Table 3. indicates the fitness value at the final generation in 10 times simulations. It was shown that the proposed method was able to provide more accurate model.

Table 3. the elite's MSE in the final generation

	Average	Min	Max
Simple GP	0.1533539	0.147358	0.157292
Proposed GP	0.1323273	0.096288	0.149925

Simulation 2

Them, we applied the proposed method to Hammerstein model identification. Assume that the true system is described by the following equations,

$$\begin{aligned} y(t) = &0.9y(t-1) - 0.7y(t-2) + x(t-1) \\ &+ 0.5x(t-2) - 0.4x(t-3) + e(t) \end{aligned} \quad (7)$$

$$x(t) = \begin{cases} -1.5 & (u(t) < -1.5) \\ u(t) & (-1.5 \le u(t) < 1.5) \\ 1.5 & (1.5 \le u(t)) \end{cases} \quad (8)$$

Where, $u(t)$ and $y(t)$ are the input and output signal of the system, $x(t)$ denotes an intermediate signal that is not able to observe. An observation noise distributed with mean 0 and unknown variance is denoted by $e(t)$. We obtained 1000 identification data by random input $u(t)$ normally distributed with mean 0 and variance 0.1^2.

Using the same configuration as previous simulation, an example of the obtained models is shown by following,

$$\begin{aligned}\hat{y}(t) = (((((&u(t-2) + (y(t-5) * (-0.123543))) \\ &+ (y(t-2) * ((-0.514437) * 1.018631))) * 0.665160) \\ &+ ((0.353379 * y(t-1)) + u(t-2))) + (u(t-1) \\ &+ ((u(t-2) * (-1.621914)) + (u(t-2) + (y(t-3) \\ &* ((-0.195790) * 1.908045)))))) \end{aligned} \quad (9)$$

In order to compare the proposed method, we carried out identification experiment by simple GP and the obtained model is as follows.

$$\hat{y}(t) = u(t-2) \quad (10)$$

Figure 11 and 12 show the observed output and the estimated output of the obtained model by the proposed method and simple GP, respectively. As shown these figures, the proposed method was able to provide more accurate model than conventional GP. The applicability of the proposed approach was shown and the other numerical examples were similar results. Table 4 indicates the fitness value at the final generation in 10 times simulations. It was shown that the proposed method was able to provide more accurate model.

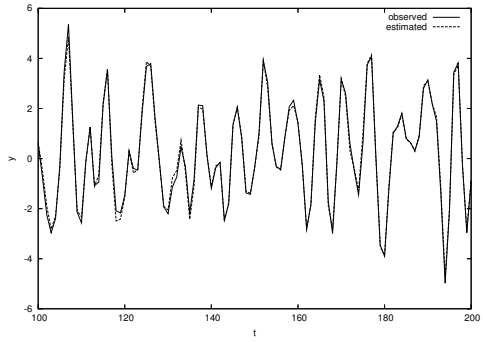

Fig. 11. Observed output data and estimated output by the proposed method

Table 4. the elite's MSE in the final generation

	Average	Min	Max
Simple GP	4.0337678	3.790967	4.645765
Proposed GP	0.2319267	0.027483	1.261841

The proposed method is expected to give more accurate model than simple GP, though these simulations are the specific cases. The MSE criterion used here is a part of information criteria, and the information criterion does not effect on numerical search steps because it is competition of the same model structure. Even if the other

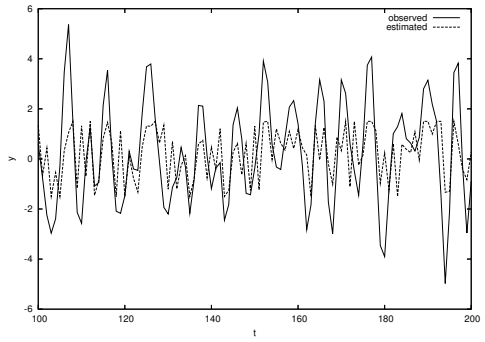

Fig. 12. Observed output data and estimated output by simple GP

criterion such as AIC (Akaike Information Criterion) and MDL (Minimum Description Length) is employed, the proposed method is expected to work well.

6. CONCLUSIONS

In this paper, we have proposed a nonlinear system identification method by using hybrid Genetic Programming, in which numerical search step is inserted in order to enhance parameter estimation ability. The numerical search step is based on the idea of Particle Swarm Optimization. In this step, the parameter vector is extracted from GP individual in each generation, and then the vector is updated by using its personal best and the global best step by step.

We have shown the applicability of the proposed approach by the numerical examples. The proposed approach applied to the nonlinear polynomial model identification and Hammerstein model identification. It was illustrated that the proposed method has an ability to give more accurate model than a simple use of genetic programming.

An application to nonlinear GP based local modeling is now under investigation. The use of the other type node setting is also studying. These results will be presented near future. Then, suitable selection of GP nodes and control parameters and the balance between the GP iterations and numerical search step sizes are the fundamental issues in such approaches. An extension of the proposed method to the multi-input multi-output systems, an application to the actual plant identification and control system design are the future works.

Acknowledgement

The authors thank graduate students, Yuuichi Matsumoto and Takeshi Korenaga for cooperation in developing a system identification tool based on genetic programming.

REFERENCES

K. A. De Jong. *Evolutionary Computation: A Unified Approach*, MIT Press, Cambridge, 2006.

G. J. Gary, D. J. Murray-Smith, Y. Li, K. C. Sharman and T. Weinbrenner, "Nonlinear model structure identification using genetic programming," *Control Engineering Practice*, 6, pages 1341–1352, 1998.

N. Hashimoto, T. Hatanaka, K. Uosaki and A. Kitamura. Nonlinear System Identification by SOM with GP Based Local Modeling, *Proceedings SICE Annual Conference 2007*, pages 2360–2363, 2007

T. Hatanaka and K. Uosaki. Hammerstein model identification method based on geneticprogramming, *Proceedings of the 2001 Congress on Evolutionary Computation, 2001*, Vol 2, pages 1430–1435, 2001.

T. Hatanaka, K. Uosaki and N. Manabe, "Structure identification in Takagi–Sugeno fuzzy modeling," *Proc. of The 2002 IEEE World Congress on Computational Intelligence*, 2002.

J. H. Holland, *Adaptation in Natural and Artificial Systems* The University of Michigan Press, 1975.

R.-V. Katya, C. M. Fonseca and P. J. Flemming : Multiobjective Genetic Programming : A Non-Linear System Identification Application, *Genetic Programming 97 Conference*, pages 207-212, 1997.

R.-V. Katya, C. M. Fonseca and P. J. Fleming : "Identifying the Structure of Nonlinear Dynamic Systems Using Multiobjective Genetic Programming," *IEEE Transactrions on Systems, Man and Cybernetics*, Part A, Vol. 34, No. 4, pages 531–545, 2004.

J. Kennedy, R.C. Eberhart, "Particle Swarm Optimization," *Proceedings of IEEE the International Conference on Neural Networks*, 1995.

J. R. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, MIT Press, Cambridge, 1992.

L. Ljung. *System Identification : Theory for the User*, Prentice-Hall, NJ, 1999.

P. Marenbach, et.al. : Data–driven Structured Modeling of a Biotechnological Fed-batch Fermentation by Means of Genetic Programming, *Proceedings the Institution of Mechanical Engineers Part I*, Vol. 211, No. 5, pages 325–332, 1997.

Y. Mitsukura, T. Yamamoto, and M. Kaneda, "A design of self-tuning PID controllers using a genetic algorithm," *Proceedings of American Control Conference 1999*, pages 1361–1365, 1999.

O. Nelles. *Nonlinear System Identification*, Springer, Berlin, 2000.

Rechenberg, I. Evolution Strategy, in *Computational Intelligence: Imitating Life*, J. M. Zurada, R. J. Marks II, and C. Robinson, Eds., IEEE Press, Piscataway, NJ. 1994

J. Sjöberg, Q. Zhang, L. Ljung and A. Benevite et al. "Nonlinear black–box modeling in system identification: a unified overview," *Automatica*, Vol.31, No.12, pages 1691–1724,1995.

T. Takagi and M. Sugeno : Fuzzy Identification of Systems and Its Applications to Modeling and Control, *IEEE Transactrions on Systems, Man and Cybernetics*, Vol. SMC-15, No. 1, pages 116–132, 1985.

K. Uosaki, T. Hatanaka. Non-linear system identification by using Genetic Programming (in Japanese). *Instrument and Control*, Vol. 39, No. 5, pages 352–357, 2000.

Yao, L. and W.A.Sethares, "Nonlinear parameter estimation via the genetic algorithm," *IEEE Transactions on Signal Processing*, Vol.42, No.4, pages 927–935, 1994.