from eqn. 7 is twice the spread of the *TF* distribution determined from S parameters, which implies that the relative error will vary from +100% to –100% as can be seen from Fig. 3, where the relative error between the two methods has been calculated for each parameter set. However, it can also be seen that in > 80% of all cases the relative error is < 40%. The mean relative error is < 20% (Fig. 3). This calculated *TF* value is therefore a good estimation for the simulation of a great number of circuit applications. In fact, when the circuit simulation (Fig. 1) is performed with a transit time parameter from eqn. 7 the delay time error is 1.7% and the fall time error is ~5% (set 3, last row of Table 1), which is negligible.

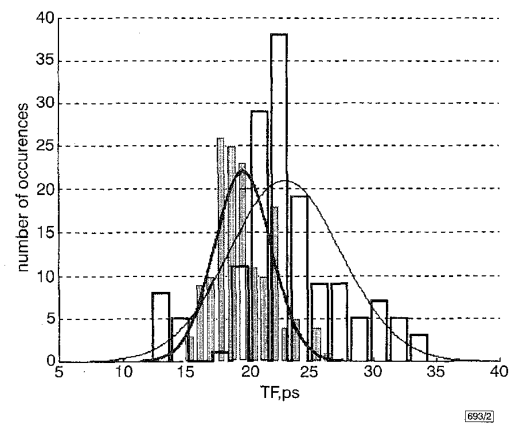

Fig. 2 *TF calculated using eqn. 7 compared to TF determined from S parameters*

□ TF calculated using eqn. 7
□ TF determined from S parameters

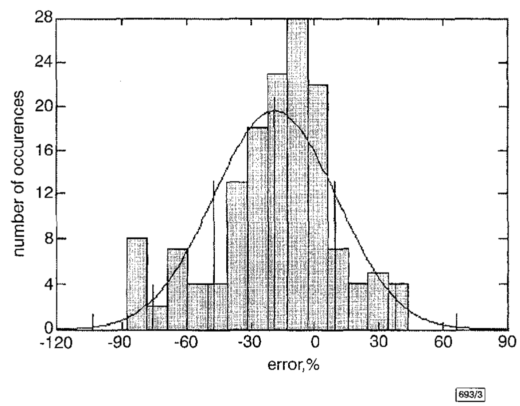

Fig. 3 *Relative error between TF from eqn. 7 and TF from S parameters*

Mean = –18.8, standard deviation = 28.4

Conclusion: For circuit applications operating up to the medium frequency range a rough estimation of the transit time parameter is sufficient for circuit simulation. A method has been presented which permits the evaluation of the forward transit time of the bipolar junction transistor without performing S parameter measurements. The reliability of the method has been proven. The method was applied to a standard 8GHz BJT technology database. The mean error of the *TF* value is < 20% and the relative error is < 40% in 80% of all cases. The presented method gives a reasonably good estimation of the forward transit time.

17 August 1998

Electronics Letters Online No: 19981219

T. Zimmer, J.B. Duluc and N. Lewis (*Laboratoire de Microélectronique IXL, Université de Bordeaux 1, 351 cours de la Libération, 33405 Talence Cedex, France*)

E-mail: zimmer@ixl.u-bordeaux.fr

References

1 GUMMEL, H.K., and POON, H.C.: 'An integral charge control model of bipolar transistors', *Bell Syst. Technol. J.*, 1970, **49**, pp. 827–852

2 CHO, H., and BURK, D.E.: 'A three-step method for the de-embedding of high frequency S-parameter measurement', *IEEE Trans.*, 1991, **ED-38**, (6), pp. 1371–1375

3 SZE, S.M.: 'Physics of semiconductor devices' (John Wiley & Sons, New Jersey, 1981)

GA-based predictive control for nonlinear processes

S.C. Shin and S.B. Park

The control problem of nonlinear processes is handled by using a GA-based optimisation technique. A three-layered neural network is used to generate future predicted outputs of the controlled process. Level control of a simulated tank is presented to demonstrate the effectiveness of the proposed method.

Introduction: In the last two decades, conspicuous progress has been made in predictive control methods both in terms of theoretical understanding and practical applications. Predictive control seems extremely powerful for processes with dead-time or if the set-point is preprogrammed [1]. Many predictive control techniques have been developed, based on the assumption that the plant to be controlled can be regarded as a linear system and that its model is available *a priori*. Such linear model-based approaches show limited control performance. Naturally, several methods involving specific nonlinear models have also been suggested. Recently, some neural network-based predictive control methods have been found to be effective in controlling a wide class of nonlinear processes [2]. Most previous works are, however, based on the nonlinear programming method which provides local optimum values only and in addition these values depend on the selection of the starting point.

In this Letter, a GA-based optimisation technique is adopted to obtain optimal future control inputs for the nonlinear plant. The GA-method is known to have more chances of finding an optimal value than descent-based nonlinear programming methods for optimisation problems [3].

Prediction model: Consider the following general discrete-time nonlinear process:

$$y(k+1) = f(y(k-m_1+1), ..., y(k), u(k-m_2+1), ..., u(k)) \qquad (1)$$

where $f(\cdot)$ is a smooth function with real $(m_1 + m_2)$ arguments. For the above nonlinear process, the future prediction outputs are produced by using a neural network-based n-step ahead predictor, as follows:

$$\hat{y}(k+n) = \sum_i w_i^O \sigma\left(\sum_j w_{ij}^I \mathbf{z}_j^n\right) \qquad (2)$$

The input $\mathbf{z}_j^n$ of the prediction model is the present and past input-output data of the plant.

$$\mathbf{z}_j^n = [y(k-m_1+n) \cdots \hat{y}(k+n-2)\, \hat{y}(k+n-1)\, u(k-m_2+n) \cdots u(k+n-2)\, u(k+n-1)]^T$$

Once a predicted output is obtained, the value is used to provide further prediction outputs by taking it as an element of the input vector $\mathbf{z}_j$.

In eqn. 2, $\sigma(\cdot)$ is a tangent hyperbolic activation function in the hidden layer, and w^I and w^O are the weights of input and hidden layers in the neural network. The overall configuration of the prediction model is shown in Fig. 1.

GA-based predictive control: In the predictive control problem, an evaluation function in a genetic algorithm is replaced by a cost function:

$$J(\mathbf{u}) = \frac{1}{2}\sum_{i=1}^{n}\{y_r(k+i) - \hat{y}(k+i)\}^2 \qquad (3)$$

where $\hat{y}$ is a function of $\mathbf{u}$ as given in eqn. 2 and y_r is the reference. The future control input sequence $\mathbf{u} = [u(k)\ u(k+1) \cdots u(k+n-1)]^T$ is obtained by minimising eqn. 3 via a genetic algorithm. In the genetic algorithm, the population is a set of solutions and can be represented as $P(r) = \{\mathbf{u}_1^r\ \mathbf{u}_2^r \cdots \mathbf{u}_m^r\}$ at iteration step r where m is an arbitrary size of population. The population undergoes natural evolution by mutation and crossover operations according to the probability of mutation p_m and crossover p_c.

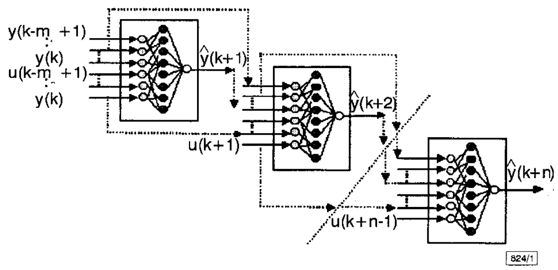

Fig. 1 *Prediction model*

● nonlinear neuron
○ linear neuron

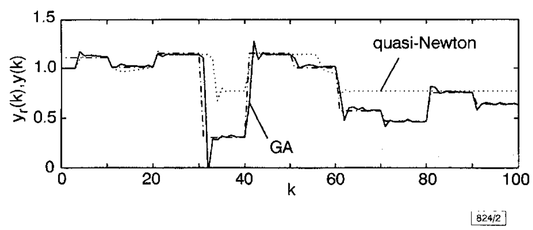

Fig. 2 *GA against quasi-Newton method*

To demonstrate the control performance of the GA-based predictive controller, consider the following discrete-time nonlinear process:

$$y(k+1) = -0.3y(k) + 0.5\sin(0.6\pi \cdot u(k)) \cdot u(k) \qquad (4)$$

In the simulation, the numbers of input, hidden, and output neurons are $n_i = 3$, $n_h = 20$ and $n_o = 1$, respectively. The parameters of the genetic algorithm are set to *pop_size* = 20, $p_c = 0.25$ and $p_m = 0.12$. The range of input is [0 2.0] and the set-point is updated every 10s with a random value selected from $0.3 \le y_r \le 1.2$. The initial value of control input u_0 is set to 0.7. Fig. 2 shows the control results obtained by quasi-Newton and GA methods. As shown in Fig. 2, the plant output does not track the desired trajectory due to the local minimum problem in the quasi-Newton method. However, the controlled output by the GA-based optimiser shows a good tracking result. The GA method is compared with the quasi-Newton method in terms of average process time (T_p) and squared average tracking error (E_{ave}) as shown in Table 1.

Table 1: Comparison GA with quasi-Newton method

		T_p [s]		E_{ave}	
		$n = 1$	$n = 2$	$n = 1$	$n = 2$
NLP	$u_0 = 0.7$	0.19	0.30	0.0505	0.0204
	$u_0 = 1.7$	0.17	0.29	0.0320	0.0201
GA	*pop_size* = 20	2.80	4.75	0.0114	0.0105
	pop_size = 50	3.30	5.25	0.0103	0.0100

The proposed GA-based predictive controller is also applied to a simulated nonlinear tank level control problem. The process, illustrated in Fig. 3, is a head tank containing liquid where the flow rate from the tank f_o is related to the level of liquid l within the tank. A flow rate into the tank f_i is available for adjustment to achieve a level l. The cross-sectional area of the tank varies considerably with level as follows:

$$r = \begin{cases} r_1 & l \le l_1 \\ r_1 + (l - l_1)\frac{r_2 - r_1}{l_2 - l_1} & l_1 < l \le l_2 \\ r_2 & l > l_2 \end{cases} \qquad (5)$$

The dynamics is described by the equations

$$f_o = v\sqrt{l} \qquad (6)$$

$$\frac{dl}{dt} = \frac{f_i - f_o}{\pi r^2} \qquad (7)$$

The radius of the tank at the surface level l of the liquid is r. The model parameters l_1, l_2, r_1, r_2 were set to 1.0, 2.0, 0.5, 2.5m respectively, and v, the flow constant, was set to 2.0. The maximum tank level was constrained to 3.0m. The parameters of the controller were set to *pop_size* = 20, $p_c = 0.25$ and $p_m = 0.12$. The control result of the GA-based predictive controller is shown in Fig. 4 which shows that the control method is available for such a process.

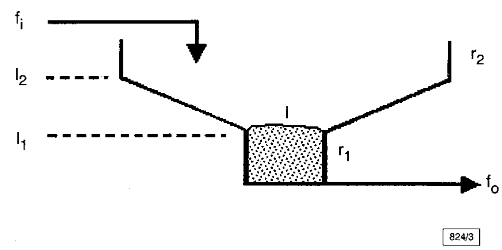

Fig. 3 *Diagram of tank level process*

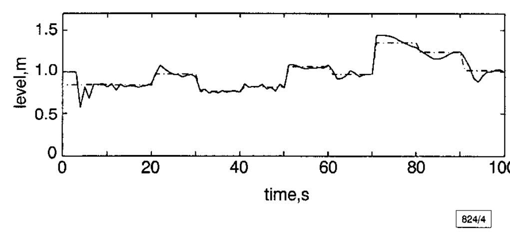

Fig. 4 *Level control of simulated tank by GA*

Conclusions: It is found from extensive simulation studies that the tracking error of the proposed GA-based control system is less than that of a quasi-Newton method, although more processing time is required in the GA-based control system. The GA-based preditive control method is robust to selection of the control input values and can be used to control a class of nonlinear processes in which control actions can be executed slowly.

27 July 1998
Electronics Letters Online No: 19981368

S.C. Shin and S.B. Park (*Department of Electrical Engineering, Korea Advanced Institute of Science and Technology, 373-1 Kusong-dong, Yusung-gu, Taejon 305-701, Korea*)

E-mail: ssc@ctrsys.kaist.ac.kr

References

1 CLARKE, D.W., MOHTADI, C., and TUFFS, P.S.: 'Generalized predictive control - Part I', *Automatica*, 1987, **23**, pp. 137–160

2 SAINT-DONAT, J., BHAT, N., and McAVOY, T.J.: 'Neural net based model predictive control', *Int. J. Control*, 1991, **54**, pp. 1453–1468

3 MICHALEWICZ, Z.: 'Genetic algorithms + data structures = evolution programmings' (Springer-Verlag, 1994)