FACTA UNIVERSITATIS (NIŠ)
Ser.: Elec. Energ. vol. 16, April 2004, xx-xx

An Algorithm Using Genetic Programming for the Compensation of Nonlinear Distortion Based on Wiener System Model

Changsong Xie, Xuhui Li

Abstract: In this paper we presented an iteration algorithm using genetic programming (GP) to get the Wiener model of a nonlinear system and then to compensate the nonlinear distortion. The GP is used to identify the linear time-invariant (LTI) part and memoryless nonlinear (MLNL) part of the Wiener model of the object system. By means of iteration, the identification precision will be improved gradually with the iteration steps. In order to compensate the nonlinearity a distortion compensation function (DCF) will be estimated also by means of GP. If the object system can be well described using Wiener model, this algorithm converges. The experiment results show that the compensation precision is fairly high.

Keywords: Genetic programming, linear time-invariant system, memoryless nonlinear system, Wiener model, distortion compensation

1 Introduction

System identification means model building for the object system so that the model is equivalent to the system concerning their inputs and outputs. As shown in Fig. 1 $x(t)$ is the input of the object system and the system model, $y(t)$ and $\widehat{y}(t)$ are outputs of the object system and system model respectively. If the energy of the error signal $e(t) = y(t) - \widehat{y}(t)$ is small enough, it is said that the model and the system are equivalent. Obviously the precision of the system model depends on the a priori knowledge to the object system. Accordingly there are *black-box*, *grey-box* and *white-box* methods. With

Manuscript received March 16, 2004

The authors are with Institute of Communications Engineering, German Federal Armed Forces University Munich, 85577 Neubiberg, Germany `changsong.xie@unibw-muenchen.de`

a *black-box* method no a prior-knowledge is available, so it is improbable to obtain a good result. With a *Grey-box* method a good result may be obtained. With a *White-box* method it is quite sure that a good result can be obtained [1]. Of course the performance of system identification depends also on the complexity of the object system and the computation amount of the identification process. A LTI system identification can achieve good result using simple methods. Unfortunately, most of the natural systems are not LTI systems but nonlinear time-variant systems (NLTV). Although a NLTV system can be generally described using a volterra model [2], but this model is too complicated to be used in system identification. In practice, nonlinear systems are investigated under some simplified conditions. For example, weak nonlinearity, time-invariance, etc.. The Wiener model can describe a special class of the nonlinear time-invariant system [2]. In this paper, only such systems are considered. A Wiener model is composed of a linear time-invariant (LTI) part cascaded with a memoryless nonlinear time-invariant (MLNL) part. In many applications, the nonlinearity of the system results in signal distortion, this degrades the system performance. A compensator is usually needed to equalize the distortion.

2 Wiener Model and Compensation of Nonlinear Distortion

A finite memory nonlinear system can be modelled by a linear time-invariant system (LTI) with finite memory in cascade with a memoryless nonlinear system (MLNL) [2]. This model is called Wiener model shown in Fig. 1.

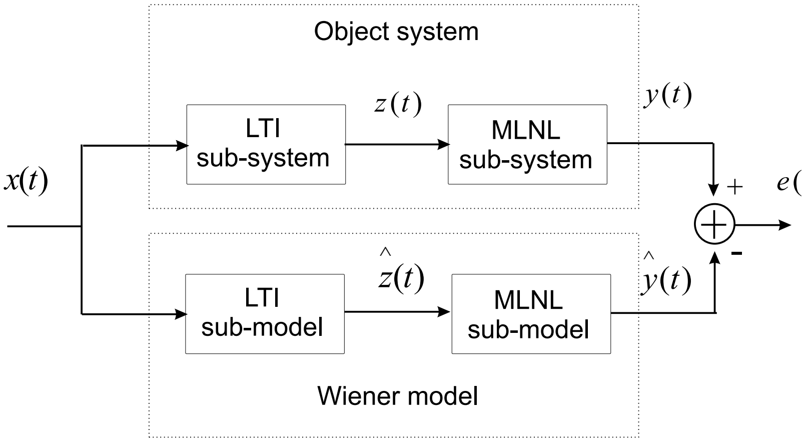

Fig. 1. System identification using Wiener model

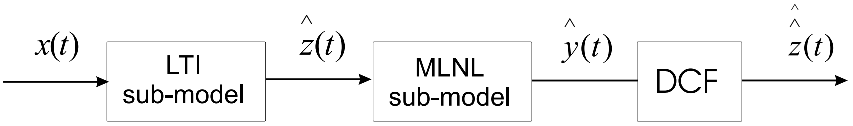

Fig. 2. Identification of compensation function

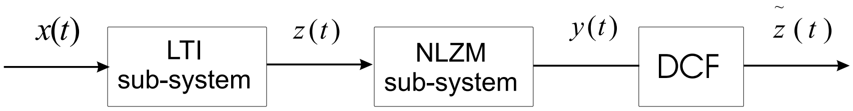

Fig. 3. Compensation of nonlinear distortion using identified DCF

It is quite common to let the dynamics of a system be included in a linear system, and the nonlinearities be static [2]. This will be the case if actuators are nonlinear due to saturation, or if audio amplifiers have a nonlinear characteristic. Using a Wiener model post-compensation will be applied. If the characteristic function of the MLNL sub-model in a Wiener model is invertible, the nonlinear distortion of the whole system is theoretically compensable. In this case the ideal compensation of nonlinear distortion is realized if $\widetilde{z} = k \cdot z$ as in Fig. 3. In practical situation, signa z is unknown, so if $\widehat{\widehat{z}} = k \cdot \widehat{z}$, it is assumed that the distortion is compensated, as shown in Fig. 2. k is a constant, typically, k=1.

3 Precision of System Identification and Distortion Compensation

System identification is the prerequisite of compensation. As long as the LTI sub-model and the MLNL sub-model are precisely identified, an inverse MLNL system can be built to compensate the nonlinear distortion. The compensation precision depends on the precision of the system identification. In order to evaluate the identification precision and compensation precision and for the sake of convergence analysis the following relative errors are defined.

LTI sub-model identification precision in i*th* iteration step:

$$\rho_l(i) = \frac{\int_{\mathcal{T}} [\widehat{z}_i(t) - \widehat{z}_{i-1}(t)]^2 dt}{\frac{1}{2}\int_{\mathcal{T}} [\widehat{z}_i^2(t) + \widehat{z}_{i-1}^2(t)]dt} \quad ; \quad i = 2,3,4... \qquad (1)$$

where $\mathcal{T}$ is time period of the training data, $\widehat{z}_i$ is the output of LTI sub-model in i*th* step.

LTI sub-model identification precision:

$$\rho_l = \lim_{i\to\infty} \rho_l(i) \tag{2}$$

Identification precision of nonlinear sub-model in i*th* iteration step:

$$\rho_{nl}(i) = \frac{\int_{\mathcal{Z}} [\widehat{f}_i(z) - \widehat{f}_{i-1}(z)]^2 dz}{\frac{1}{2}\int_{\mathcal{Z}} [\widehat{f}_i^2(z) + \widehat{f}_{i-1}^2(z)] dz} \quad ; \quad i = 2, 3, 4... \tag{3}$$

where $\widehat{f}_i$ is the identified characteristic function of the MLNL sub-model in i*th* step, $\mathcal{Z}$ is the input value space.

Identification precision of nonlinear sub-model:

$$\rho_{nl} = \lim_{i\to\infty} \rho_{nl}(i) \tag{4}$$

Identification precision of the whole system in i*th* iteration step:

$$\rho_s(i) = \frac{\int_{\mathcal{T}} [y(t) - \widehat{y}_i(t)]^2 dt}{\int_{\mathcal{T}} y^2(t) dt} \quad ; \quad i = 2, 3, 4... \tag{5}$$

where $y(t)$ is the output of the object system, $\widehat{y}_i$ is the output of system model in i*th* step.

Identification precision of the whole system:

$$\rho_s = \lim_{i\to\infty} \rho_s(i) \tag{6}$$

Compensation precision of i*th* iteration step:

$$\rho_c(i) = \frac{\int_{\mathcal{T}} [\widehat{z}_i(t) - \widetilde{z}_i(t)]^2 dt}{\frac{1}{2}\int_{\mathcal{T}} [\widehat{z}_i^2(t) + \widetilde{z}_i^2(t)] dt} \quad ; \quad i = 2, 3, 4... \tag{7}$$

where $\widetilde{z}_i$ the compensated output using DCF in i*th* step.

Compensation precision:

$$\rho_c = \lim_{i\to\infty} \rho_c(i) \tag{8}$$

4 Iterative Identification Algorithm

As shown in Fig. 2, the Wiener model consists of two sub-models. According to our fundamental experiments GP can easily identify the sub-models, if the inputs and outputs of the corresponding sub-systems are known. The problem is that the interstage signal z is unknown. To overcome this problem, an algorithm that can somehow meet in the middle is proposed. The basic idea is that an initial LTI sub-model is at first supposed. With the LTI sub-model we can get the $\widehat{z}_1$ as the first estimate of z. With $\widehat{z}_1$ and y we identify the MLNL sub-model $\widehat{f}_1$. With MLNL sub-model we can identify the DCF $\widehat{f}_1^{-1}$. With DCF we can get the compensated signal $\widehat{\widehat{z}}_1$, which will be assigned as the estimate of z in the second iteration step. So the iteration is running. If the iteration process converges, it is said that the system model is found and the DCF can compensate the nonlinear distortion.

The complete algorithm can be described in the following steps:

1. Initiation of LTI sub-model h_0.
2. With x and LTI sub-model h_i the interstage signal $\widehat{z}_i$ will be identified
3. With $\widehat{z}_i$ and y the MLNL sub-model $\widehat{f}_i$ will be identified.
4. Constants optimization of LTI sub-model and MLNL sub-model.
5. If the identification is precise enough go to 12, otherwise go to 6 .
6. With the identified MLNL sub-model $\widehat{f}_i$ the DCF $\widehat{f}_i^{-1}$ will be identified.
7. Constants optimization for DCF.
8. With the identified DCF $\widehat{\widehat{z}}_i$ will be calculated.
9. Let $\widehat{z}_{i+1} = \widehat{\widehat{z}}_i$.
10. With x and $\widehat{z}_{i+1}$ LTI sub-model h_{i+1} will be identified.
11. Go to 2.
12. Convert the DCF into a simplified form with extended polynomial grammar in GP.
13. Constants optimization of DCF
14. If the precision of DCF is satisfied, go to 15, otherwise go to 12.
15. stop.

The Fig. 4 shows the flowchart of the algorithm. In GP evolution the selection of the surviving LTI sub-models is relatively free. But the selection of the surviving characteristic function of the MLNL sub-models should satisfy the following conditions.

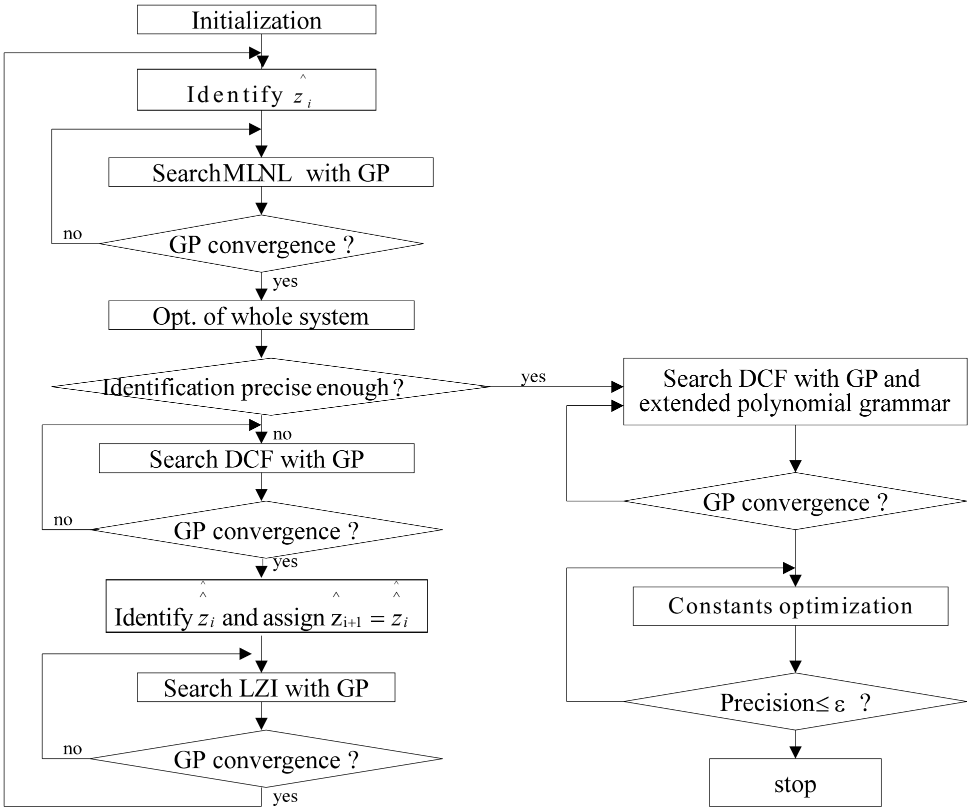

Fig. 4. The flowchart of the algorithm

1. Fitness value is small.
2. Characteristic function is smooth.
3. Characteristic function should be invertible.

The criteria that the identification process to be stopped is that the relative errors $\rho_l(i)$, $\rho_{nl}(i)$ and $\rho_s(i)$ are all small enough, e.g. $< 10^{-3}$.

5 GP Grammar and Constants Optimization

Generally, GP grammar defines the searching space of the GP programs. This searching space includes two sets, namely the operation set and the variable set. For example, the variable set of a LTI GP grammar is the set of all the past inputs and outputs and its operation set includes only addition and constant-multiplication. That means only functions with the form $y(t) = \sum_{i=0}^{M} a_i x(t-\tau_i) + \sum_{j=1}^{N} b_j y(t-\tau_j)$ are allowed. A MLNL GP grammar may be more complicated, since its operation set can include all possible functions, like $\sin, \cos, \log, \exp$, etc.. A simple MLNL grammar is the extended polynomial GP grammar, it limits GP to search the characteristic function only in the form of $f(x) = \frac{\sum_{i=0}^{K} a_i x^i}{\sum_{j=0}^{L} b_j x^j}$. This grammar is acceptable but not unique for the identification of a characteristic function and a DCF. With such a grammar GP may obtain a simple-formed result with less computation than with a grammar containing complicated functions.

Optimization of constants will not change the structure of the system but improve the identification precision and the compensation precision.

The Compensation precision depends also strongly on the assumption of the initial LTI sub-model. The following information can help to determine the initial sub-model.

1. A priori knowledge of the object system.
2. The expected object linear system.
3. The training data.

6 Experiment and Results

In this section a test object system whose LTI and MLNL parts are described by Eq. (9) and Eq. (10) is supposed. The nonlinear function Eq. (10) is invertible in $z \in [-2, 2]$.

$$\begin{aligned} z(t) &= 0.3x(t) - 0.13x(t-1) + 2.5x(t-2) + 0.9x(t-3) + 0.4x(t-4) \\ &\quad +0.2x(t-5) + 0.1x(t-6) + 1.5z(t-1) - 0.7z(t-2) \\ &\quad +0.9z(t-3) - 1.3z(t-4) \end{aligned} \tag{9}$$

$$y = f(z) = z + 0.8z^2 + 0.4z^3 - 0.2z^4 \tag{10}$$

In the experiment the input signal is a zero-mean white noise. The data length is 500 samples. For each GP searching process, 20 independent runs are carried out. Tab. 1 shows the parameters for controlling GP runs.

The characteristic function found by GP algorithm described in section 4 is

$$\widehat{y} = \widehat{f}(\widehat{z}) = 1.0026\widehat{z} + 0.7711\widehat{z}^2 + 0.3998\widehat{z}^3 - 0.1907\widehat{z}^4 \tag{11}$$

Population size	10000
Maximum of generation	60
Prob. of crossover p_c	0.9
Prob. of reproduction p_r	0.1
Prob. of mutation p_m	0.3
Grammar limitation	LTI-Grammar for searching LTI sub-model $\sin, \log, \exp$, polynomial for MLNL sub-model Extended polynomial for simplified DCF
Weighting fitness	For searching MLNL sub-model and DCF

Table 1. Parameters for controlling the GP Run

The simplified compensation function found by GP using extended polynomial grammar:

$$\widetilde{z} = \widehat{f}^{-1}(y) = \frac{17.4423y^3 + 107.9088y}{-1.2951y^3 + 51.7418y^2 + 36.5536y + 92.1343} \tag{12}$$

Fig. 5 shows the convergence process of the identified characteristic function of the MLNL sub-model.

Fig. 6 shows the identification precision of the whole system in different iteration steps. The final relative error is $2.75 \cdot 10^{-4}$.

To verify the algorithm, we compare the output of the real linear system and the compensated system. Fig. 7 shows both the outputs of linear subsystem and the compensated signal with white noise as input. The final relative error is $9.88 \cdot 10^{-4}$.

Fig. 8 shows both the outputs of linear sub-system and the compensated signal with sine-wave as input. The final relative error is $5.06 \cdot 10^{-4}$.

7 Conclusion

A iteration algorithm using genetic programming for the compensation of nonlinear distortion is proposed. The requirements for this algorithm working well are:

1. The object system can be described or approximated using a Wiener model.

2. The identified characteristic function of the MLNL sub-model is invertible.

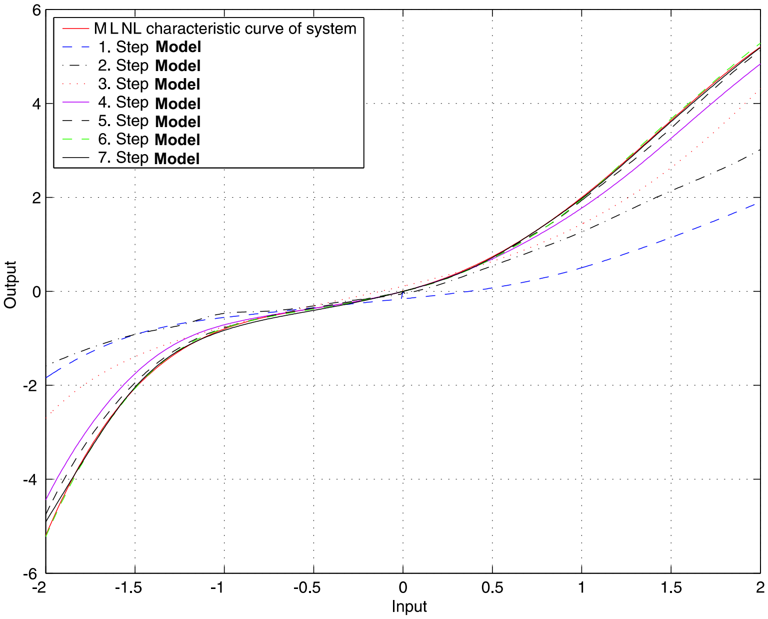

Fig. 5. Iterative convergence of identified MLNL sub-model

The compensation precision depends on the precision of system identification, that will be determined by the following factors:

1. A priori knowledge to the object system.

2. Limitation on the GP grammar.

3. Computation resources.

Compared to the traditional system identification methods and distortion compensation methods, the advantage of this algorithm is the compensation precision. However the compensation functions are usually complicated. To overcome it, again GP is used to simplify the compensation function in a form of extended polynomial. Another advantage is that theoretically GP needs no a priori knowledge to the object system apart from the measurement data, but GP may not converge as a result of unsuitable initialization and searching space. Adequate a priori knowledge to the object system can avoid this disadvantage.[2].

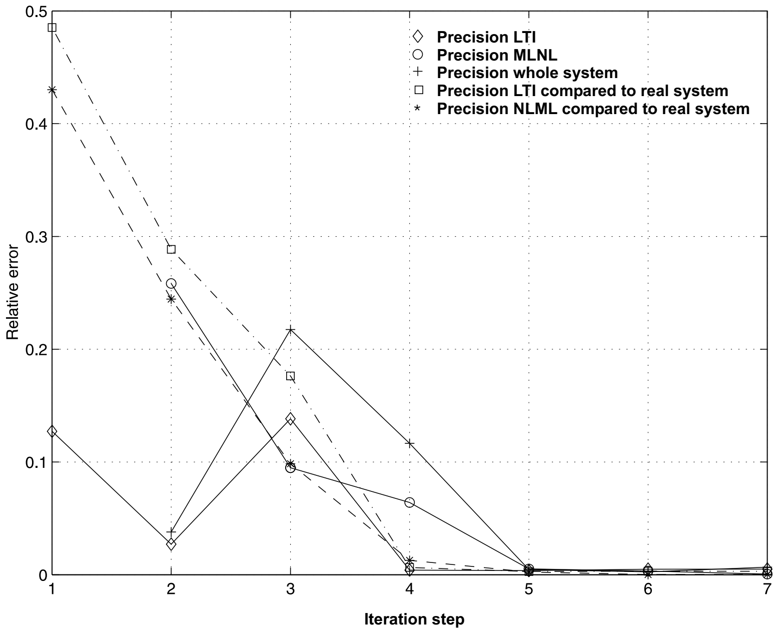

Fig. 6. Identification precision of the whole system at each iteration step

References

[1] X. Li, *Dynamische Kompensation von nichtlinearen Verzerrungen mit genetischer Programmierung.* Dissertation, Universitaet der Bundeswehr Muenchen: Shaker Verlag, 2003.

[2] M. Schetzen, *The Volterra and Wiener theories of nonlinear systems.* New York: Wiley and Sons, 1980.

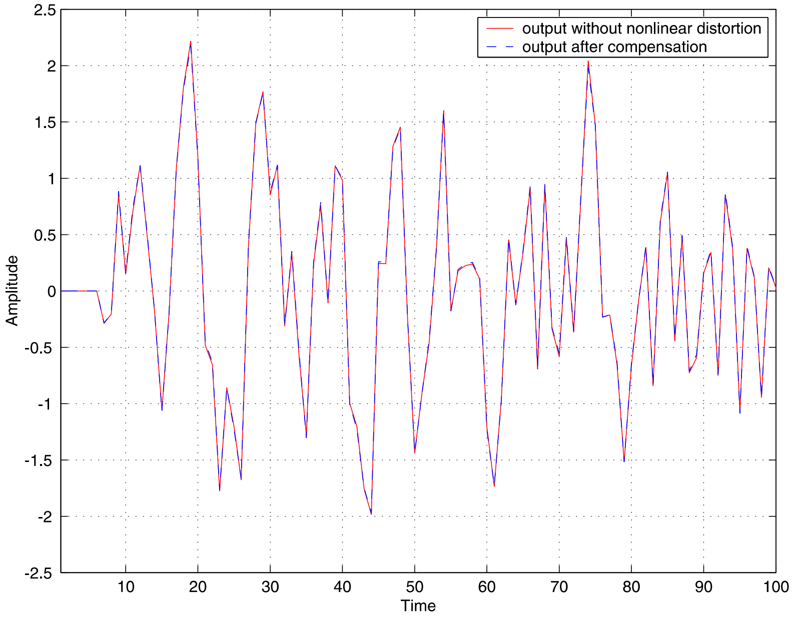

Fig. 7. Compensation effect with white noise input

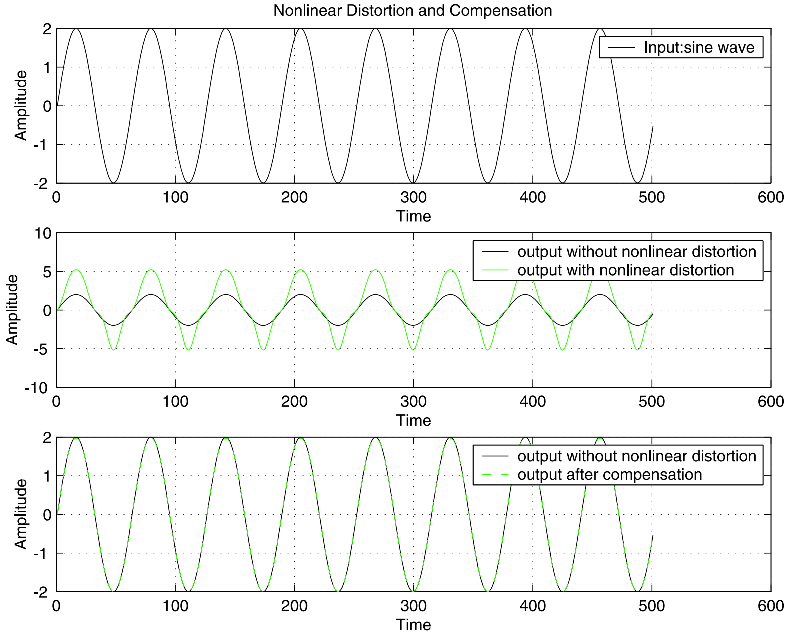

Fig. 8. Compensation effect with sine-wave input