In future research the interaction of 3 different agents: splitting agent, speaker agent and classification agent would be tested. The given system can be extended to the system of speech and emotions recognition if we add some agents for emotions recognition. Also, some aggregation agents can be added to provide the relation between classification and complete meanings, if there is a special need that would appear for classification based on syllables and vowels or consonants. If the further investigation shows that dynamic programming method in classification of speech parts is fast enough and reliable then it should be a good alternative to complex classification technique for real-time speech recognition or play a special role in forming the classification algorithm for multi agent classification systems those were presented in the second part of the article.

It is also important to point out that the programming of MAS with its given properties are possible with using the one of the special programming software, which is available to free download and use all over the internet. What is also important to point out is that some platforms for MAS programming works on Java and on different devices, such as, for example, cellar phone.

All in all we should highlight that the speech recognition problem is the complex problem that requires a lot of different algorithms to carry out the different tasks. These task, as it was pointed out earlier are also complex and the accuracy of every one's depends on the quality of the solution for the previous problem.

References

1. Walsh M., Kelly R., O'Hare G. M. P., Carson-Berndsen J., Abu-Amer T. A Multi-Agent Computational Linguistic Approach to Speech Recognition // IJCAI'03 Proceedings of the 18th Intern. joint Conf. on Artificial intelligence. 2003. P. 1477–1479.

2. Walsh M., O'Hare G. M. P., Carson-Berndsen, J. An agent-based framework for speech investigation // INTERSPEECH 2005. 2005. P. 2701–2704.

3. Taha M., Helmy T., Alez R. A. Multi-agent Based Arabic Speech Recognition // Proceedings of the 2007 IEEE/WIC/ACM Intern. Conf. on Web Intelligence and Intern. Conf. on Intelligent Agent Technology Workshops, Silicon Valley, CA, USA. 2007. P. 433–436.

4. Russell S. J., Norvig P. Artificial Intelligence: A Modern Approach. 2nd ed. : Upper Saddle River. New Jersey : Prentice Hall, 2003.

5. Wooldridge M. An Introduction to MultiAgent Systems : John Wiley & Sons. 2002. P. 366.

6. Furtuna T. F. Dynamic Programming Algorithms in Speech Recognition // Informatica Economica. Vol. XII. Issue 2. 2008. P. 94–98.

И. С. Рыжиков

О ПРИМЕНЕНИИ МУЛЬТИАГЕНТНЫХ СИСТЕМ ДЛЯ ЗАДАЧ РАСПОЗНАВАНИЯ РЕЧИ

Предлагаются две различных многоагентная системы для решения задачи распознавания речи. Многоагентные системы (МАС) становятся достаточно популярными благодаря их функциональности и применимости к решению сложных задач. В основе таких систем лежит функционирование каждого ее элемента – агента, и их активное взаимодействие. Основным преимуществом подобного подхода является возможность использовать в качестве агентов простые подсистемы, гораздо более простые, чем решаемая задача. Таким образом, решение задачи сводится к настройке взаимодействий между агентами.

Ключевые слова: многоагентные системы, распознавание речи, интеллектуальные агенты.

UDC 005; 519.7; 303.732

I. S. Ryzhikov

AUTOMATIC LINEAR DIFFERENTIAL EQUATION IDENTIFICATION IN ANALYTICAL FORM

In this paper we suggest a reduction of linear dynamics identification problem to the global optimization task. The current approach allows automatic determining the structure and parameters of a linear differential equation via the usage of the modified hybrid evolutionary algorithm for extremum seeking. The a priori information algorithm needed is only the dynamic system initial point or an estimation of the initial point and the sample of measurements: system output and, if there is one, system input.

Keywords: evolutionary strategies, identification, structure and parameters, differential equation.

There are many different approaches of linear differential equation parameters estimation. But since there is no a priori information about the system itself most of them become useless. For different tasks there are some special techniques that allow solving the problem. In this paper we consider the situation when the data can be noised and the system structure is unknown. We can use, for example, stochastic difference equations [1], and build a model using the output observations. But there are some restrictions in using this approach: we still need the information about the order of differential equation and we must observe the system output on the unit step function. To simply estimate the reaction of linear dynamic system on different control input or smooth the output data we can use nonparametric methods, neural network of fuzzy output modeling. Also, there is a possibility to estimate the solution of differential equation for current situation [2] via genetic programming. As for nonparametric or neural network approaches it is possible to define the system output for different control function using the Cauchy equation, but the system cannot be presented in an analytical form. As for genetic programming technique we still have a possibility to find the output for different control, but since then, it can be found numerically. Moreover, the models are very complex and the analytical solution for estimation of differential equation seems to be very long and have a superior size. Here we suggest seeking the model in differential equation form. The benefits are the following: it would be easy to estimate the system output numerically for any control function with any desired precision, in some cases it would be easy to define an analytical solution via eigenvalues evaluation, and there are plenty of control methods and analysis techniques for the models in differential equation form.

In article [3] the dynamic system approximation with second order linear differential equation via genetic algorithm is examined. The genetic algorithm is well known as effective global optimization technique. The only problem with it is that seeking works on a compact with given boarders and the real values ought to be quantized. In this paper we suggest to use an evolutionary strategies algorithm with local optimization and some modifications to approximate not only the parameters, but also the structure of a ordinary differential equation (ODE).

Linear differential equations models are useful in filtering, in articulatory identification [4] and [5] – for stochastic ODE that can be identified in the same way. With some modifications, this method can be used for Bessel equations identification. It is also applicable to Markov processes [6] as a stochastic ODE too. That is why the linear differential equation identification can be useful is some fields related to speech recognition problem.

Let us have a sample $\{y_i, u_i, t_i\}, i=\overline{1,s}$, where s is its size, $y_i \in R$ are dynamic system output measurements at t_i, and $u_i = u(t_i)$ are control measurements. It is also known, that the system is linear and dynamic, so it can be described with the ordinary differential equation (ODE):

$$a_k \cdot x^{(k)} + a_{k-1} \cdot x^{(k-1)} + \ldots + a_0 \cdot x = b \cdot u(t),$$
$$x(0) = x_0 . \quad (1)$$

Here x_0 is supposed to be known. In the case of the transition observation, we can put forward a hypothesis about initial point: the system output is known at initial time and the derivative values can be set to zero, because usually the system observation starts in its steady state. In general, the initial point can be approximated. Using the sample data we need to identify parameters and the system order m, which is assumed to be limited, so $m \le M, M \in N$. M is a parameter that is set by the researcher. This value limits the structure of the differential equation, i.e., it limits the ODE order. It is also assumed that there is an additive noise $\xi : E(\xi) = 0, D(\xi) < \infty$, that affects the output measurements:

$$y_i = x(t_i) + \xi_i . \quad (2)$$

Without information on the system order, we would not be able to solve the identification task, but because of the maximum order limitation, the task can be partially parameterized. The maximum order is supposed to be chosen a priori. It would specify the optimization problem space dimension.

Without loss of the generality, let the leading coefficient of ODE be the constant equal to 1, so that

$$x^{(k)} + \frac{a_{k-1}}{a_k} \cdot x^{(k-1)} + \ldots + \frac{a_0}{a_k} \cdot x = \frac{b}{a_k} \cdot u(t), \quad (3)$$

or

$$x^{(k)} + \tilde{a}_k \cdot x^{(k-1)} + \ldots + \tilde{a}_1 \cdot x = \tilde{b} \cdot u(t) . \quad (4)$$

Then we can seek the solution of the identification task as a linear differential equation with the order m,

$$\hat{x}^{(m)} + \hat{a}_m \cdot \hat{x}^{(m-1)} + \ldots + \hat{a}_1 \cdot \hat{x} = \hat{b} \cdot u(t),$$
$$\hat{x}(0) = x_0 , \quad (5)$$

where the vector of equation parameters

$$\hat{a} = (0, \ldots, 0, \hat{a}_m, \ldots, \hat{a}_1, \hat{a}_0) \in R^n , \quad n = M + 1 ,$$

delivers an extremum to the functional

$$I(a) = \sum_{i=1}^{N} \left| y_i - \hat{x}(t_i) \right| \Bigg| \to \min_{a \in R^n} . \quad (6)$$

In general case, the solution $\hat{x}(t)$ is evaluated with a numerical integration method, because the control function has no analytical from, rather is given algorithmically. We prefer the criterion (6) instead of quadratic criteria because of its robustness. For the correct numerical scheme realization, let us have a coefficient restriction for equation (3), $|a_k| > 0.05$. Otherwise, this parameter is going to be equal to zero, so $a_k = 0, m = m - 1$. That condition prevents extra computational efforts of the numerical evaluation scheme and is necessary for the local optimization algorithm effecting on the system structure.

Now let us consider the specific modelling issue. The identification of linear differential equations system is connected with the optimization problem for the system of equations:

$$a^i_{k_i} \cdot x_i^{(k)} + \ldots + a^i_0 \cdot x_i = \sum_{\substack{j=1 \\ j \ne i}}^{n_o} b^i_j \cdot x_j + b^i_0 \cdot u_i(t) , \quad (7)$$

where $x_i, i = \overline{1, n_o}$, is an observed system output; n_o is the number of outputs.

Equation (7) shows that the system is considered not in general way and every system output depends on other outputs but not on their derivatives. Also, there is only one control input for every equation. This can be easily extended to the case with many control inputs.

The identification problem for the system with equation (7) is important and an ability to solve it could be useful. And it is clear, that the functional (6) can be transformed into the functional

$$I(a)=\sum_{j=1}^{n_o}\sum_{i=1}^{N}\left|y_i^j-\hat{x}_j(t_i)\right|\Bigg|_{\hat{a}=a}\to\min_{a} \qquad (8)$$

for system (7). The criterion can easily be extended to matrix form of differential equation.

The reason why the basics of optimization technique was borrowed from evolutionary strategies algorithm [7] is that the identification problem leads to solving the multimodal optimization task. The goal of the given approach is the identification of the parameters and the structure simultaneously. The system structure and its parameters are defined with one vector. The criteria (6) and (8) for this vector is complex and sensitive to the its components, which are changing by stochastic search operators. This is why we have to develop the specific modification for the global optimization technique.

Let every individual be represented with tuple

$$H_i=\left\langle op^i, sp^i, fitness(op^i)\right\rangle, i=\overline{1,N_I},$$

where $op_j^i\in R$, $j=\overline{1,k}$, is the set of objective parameters of the differential equation; $sp_j^i\in R^+$, $j=\overline{1,k}$, is the set of strategic parameters; N_I is the population size;

$$fitness(x):R^k\to(0,1],\ fitness(x)=\frac{1}{1+I(x)}$$

is the fitness function.

As the selection types, proportional, rank-based and tournament-based selections were chosen. The algorithm produces one offspring from two parents and every next population have the same size as previous. Recombination types are intermediate and discrete. The mutation of every offspring's gene happens with the chosen probability p_m. If we have the random value $z=\{0,1\}$, $P(z=1)=p_m$ which is generated for every current objective gene and its strategic parameter then

$$op_i^{offspring}=op_i^{offspring}+z\cdot N(0,sp_i^{offspring});$$

$$sp_i^{offspring}=\left|sp_i^{offspring}+z\cdot N(0,1)\right|,$$

where $N(m,\sigma^2)$ is the normally distributed random value with the mean m and the variance σ^2.

We suggest a new operation that could increase the efficiency of the given algorithm. For every individual, the real value is rounded down to the nearest integer. This provides searching for solutions with near the same structure.

Also for N_1 randomly chosen individuals and for N_2 randomly chosen objective chromosomes we make N_3 iterations of local search with the step h_l to determine the better solution. This is the random coordinate-wise optimization.

To make an investigation 100 systems were generated: 10 for every order from first to tenth. Parameters of the systems were randomly generated: $\widehat{a}_k^i=U(-5,5)$, $\widehat{b}_k=U(-5,5)$, $i=\overline{1,10}$, $k=\overline{1,i}$ where $U(-5,5)$ is the uniform distribution. The time of the process was set to 5. The control function was the step function and we know what was the control for every system, so $u(t)=1$. Let $\{x_i,t_i\}, i=\overline{1,T/h_i}$ be the numerical solution for the system. We take $s<T/h_i, s=100$ points randomly. For every system 10 runs of the algorithm were executed with every combination of its parameters. To estimate the efficiency of different approaches we considered the identification without any noise.

Having different types of the selection and the crossover, we would also vary the probability $p_m\in\left\{\frac{1}{11},\frac{5}{11},\frac{1}{5},1\right\}$ to find out the most effective combination of the algorithm settings. As a pre-set we use the population size in 50, the number of populations in 50, $N_1=50$, $N_2=50$ and $N_3=1$ with $h_l=0.05$.

We compared the efficiency of following algorithms: 1 – the evolutionary strategies (ES) algorithm; 2 – ES with the local optimization, hybrid evolutionary strategies (HES); 3 – HES with modified mutation; 4 – HES with turning real numbers into integer numbers; 5 – HES with modified mutation and turning real numbers to integer ones.

After testing the algorithms on different samples of the systems, the efficient presets were found: modified HES algorithm with turning the real numbers to integer ones, 50 individuals for 50 populations, $N_1=50$, $N_2=50$ and $N_3=1$ with $h_l=0.05$, the tournament selection with the tournament size 25 %, the discrete crossover and the mutation with the probability $p_m=\frac{5}{11}$.

For the proper structure and parameters determination we need an adequate sample that reflects all the transient process. Let us take some stable systems that come into the steady state in time $T=5$. In Table 1 we would make an efficiency investigation for the modified HES algorithm. 20 runs of the algorithm were made for every system. We will say that the algorithm determines the structure and parameters if $\max(\hat{a}-a)<0.05$.

As we can see from Table 1, the high value of fitness does not guarantee the success in identification the real structure. Let us highlight that for the most of solutions found from this study for stable systems, the order was found correctly.

Let us describe the identification problem for hexadecane chemical reaction. The disintegration of the hexadecane gives the following products: the spirits and carbonyl compounds. The initial point is known. There is no control input in this identification problem. We set the maximum order for the first equation to 10. The 50 runs of the algorithm gave us some different solutions that are shown in Table 2.

Table 1

The efficiency of "true" parameters estimation

Order	$\hat{p}(\max(\hat{a}-a)<0.05)$	Fitness
1	0,65	0,9593
2	0,95	0,9979
3	0,90	0,9977
4	0,95	1,0000
5	0,80	0,9961

Table 2

The hexadecane disintegration model

Models and the error (I)
$4.05\cdot x'+0.9\cdot x=1,\quad I=0.3022$
$1.05\cdot x'+0.4\cdot x=1,\quad I=0.2834$
$2.1\cdot x'+0.55\cdot x=1,\quad I=0.1822$
$-1.05\cdot x'''-0.15\cdot x''-6.85\cdot x'-0.9\cdot x=1,\quad I=0.227$
$-3.4\cdot x'-0.45\cdot x=0,\quad I=0.202$

As we can see, the found parameters and system structure forms the first order differential equation, and that fact does not contradict the hypothesis [8], which states that disintegration chemical reactions can be presented as first order linear differential equation.

Knowing the structure of the equations we can identify the system itself in a matrix form. The given optimization procedure is a stochastic algorithm, that is why the best solution from the 20 runs was taken. The system outputs and the sample are shown on figure 1 for hexadecane, spirits and carbonyl compounds. As we can see on figures, the measurement at the point $t = 7$ seems to be an abnormal measurement, but it did not effect on the model.

The solution for the system can be represented in the matrix form

$$A=\begin{pmatrix}-0.1671 & 0.7630 & -0.3625\\ 0.0413 & -0.3428 & 0.1150\\ 0.0026 & 0.4050 & -0.3270\end{pmatrix},$$

$$B=\begin{pmatrix}0.3477\\ 0\\ 0\end{pmatrix}.$$

Modifications of evolutionary strategies algorithm increase the accuracy of model and allow solving two tasks at the same time. The further investigation should be concentrated on the estimation of the performance of algorithm with the different local optimization and mutation parameters. Also, differential equation algorithm or partial swarm optimization are to be tested as basic optimization procedure.

References

1. Zoteev V. Parametrical identification of linear dynamical system on the basis of stochastic difference equations // Matem. Mod. 2008. Vol. 20. No 9. P. 120–128.

2. Evolutionary modeling of systems of ordinary differential equations with genetic programming / H. Cao, L. Kang, Y. Chen, J. Yu // Genetic Programming and Evolvable Machines. Vol. 1 (40). 2000. P. 309–337.

3. Parmar G., Prasad R., Mukherjee S. Order reduction of linear dynamic systems using stability equation method and GA // International J. of computer and Infornation Engeneering. 1:1, 2007.

4. Reimer M., Rudzicz F. Identifying articulatory goals from kinematic data using principal differential analysis // Proceedings of Interspeech 2010, Makuhari Japan. 2010. P. 1608–1611.

5. Mineiro P., Movell J. R., Williams R. J. Modeling path distributions using partially observable diffusion networks: a Monte-Carlo approach //

6. Saerens M. Viterbi algorithm for acoustic vectors generated by a linear stochastic differential equation // Proceedings of the IEEE Intern. Conf. on Acoustics, Speech and Signal Processing (ICASSP), Detroit, 1995. P. 233–236.

7. Schwefel Hans-Paul Evolution and Optimum Seeking : New York : Wiley & Sons., 1995.

8. Romanovskii B.V. The foundations of the chemical kinetics. Moscow : Ekzamen, 1996.

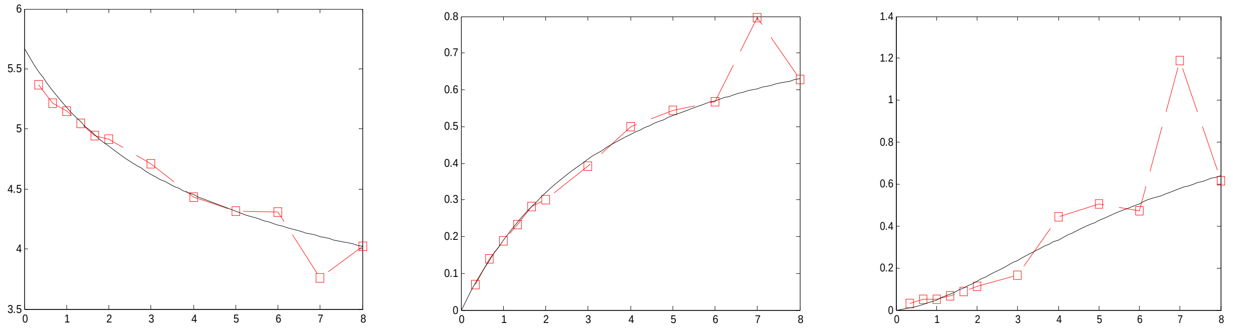

Fig. 1. Hexadecane, spirits and carbonyl compounds concentration measurements and model output, respectively

И. С. Рыжиков

АВТОМАТИЧЕСКАЯ ИДЕНТИФИКАЦИЯ ЛИНЕЙНЫХ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ В АНАЛИТИЧЕСКОМ ВИДЕ

Рассматривается сведение задачи идентификация линейных динамических систем к задаче поиска глобального оптимума. Рассматривается подход, который позволяет автоматически определять структуру и параметры линейного дифференциального уравнения через решение оптимизационной задачи с помощью модифицированного гибридного метода эволюционных стратегий. Располагая априорной информацией такой, как вектор начальных состояний системы или его оценка, выборка измерений выхода динамической системы и входная характеристика.

Ключевые слова: эволюционные стратегии, идентификация, структура и параметры, дифференциальное уравнение.

UDK 519.8

E. S. Semenkin, M. E. Semenkina

INTEGRATION OF INTELLIGENT INFORMATION TECHNOLOGIES ENSEMBLES WITH SELF-CONFIGURING GENETIC PROGRAMMING ALGORITHM*

Self-configuring genetic programming algorithm with the modified uniform crossover operator, that realizes a selective pressure on the recombination stage, is used for the automated integration of the computational intelligence technique ensembles. Ensemble members are the symbolic regression formulas, the artificial neural networks or their combination. They are also designed automatically with the self-configuring genetic programming algorithm. The comparative analysis of the approach performance is given on the benchmark and real world problems.

Keywords: Genetic programming, self-configuration, neural networks, symbolic regression, ensembles, automated design, classification problems.

For many real world problems we can observe the following situation. There is a big data base of the results of the complex system behavior observations but appropriate model of this system is not yet clear. Here we can use intelligent information technologies (IIT) to obtain the first stage model within short time in order to simulate the system and learn its properties that gives us a possibility to develop a full profile model of the system. However, the design of IIT can also be a problem.

Currently, intelligent systems have got wide propagation in various fields of human activity connected with complex systems modeling and optimization. Artificial neural networks [25], fuzzy logic [31], symbolic regression [18], evolutionary algorithms [8] and other techniques and technologies are the popular tools for the system investigation due to their capability to solve complex intelligent problems that are difficult for the classic techniques [17].

The highly increasing computing power and technology made possible the use of more complex intelligent architectures, taking advantage of more than one intelligent technique in a collaborative way. This is an effective combination of intelligent techniques that outperform or compete to simple standard intelligent techniques.

One of the hybridization forms, the ensemble technique, has been applied in many real world problems. It has been observed that the diversity of members, making up a committee, plays an important role in the ensemble approach [5].

Different techniques have been proposed for maintaining the diversity among members by running on the different feature sets [14] or training sets (e. g. bagging [1] and boosting [11]).

Some techniques, such as neural networks, can be run on the same feature and training sets producing the diversity by different structures [20]. Simple averaging, weighted averaging, majority voting, and ranking are common methods usually applied to calculate the ensemble output.

Johansson et al. [16] used genetic programming (GP) for building an ensemble from the predefined number of the artificial neural networks (ANN) where functional set consisted of the averaging and multiplying and the terminal set included the models and constants. In [2], a similar approach was proposed where first a specified number of the neural networks is generated and then a genetic programming algorithms applied to build an ensemble making up symbolic regression from partial decisions of the specific members.

*The study was supported by The Ministry of education and science of Russian Federation, project № 16.740.11.0742, 14.740.12.1341 and 11.519.11.4002.