Automatic differential equations identification by self-configuring genetic programming algorithm

T S Karaseva
Reshetnev Siberian State University of Science and Technology 31, Krasnoyarsky Rabochy Av., Krasnoyarsk, 660037, Russian Federation

E-mail: tatyanakaraseva@yandex.ru

Abstract. The paper considers a reduction of differential equations identification problem to the symbolic regression task. The current approach allows automatic determining the structure of a differential equation via the usage of the self-configuring genetic programming algorithm. The a priori information needed is only the dynamic system initial point and the sample of input and output effects. The stability of the proposed approach to the presence of noise in the sample and the small amount of data is investigated.

1. Introduction

Models of dynamic processes building is a current and demanded field for solving research and practical problems. The paper proposes an approach that is planned to be applied later in identification of a dynamic object based on observations of its output, i.e., in solving so-called inverse problems of mathematical modeling. According to this approach a mathematical model is built on the basis of the initial data concerning the input and output effects. It describes the relationship between them.

A large number of methods based on various approaches has been developed to solve problem of the model of the dynamic object building. A lot of methods are based on the estimation of the transient response [1]. One can use nonparametric methods, a neural network for modeling fuzzy inference to evaluate the response of a dynamic system to different control input data or smooth output data [2-4]. It is also possible to estimate the solution of the differential equation that describes the behavior of the dynamic object using genetic programming [5].

Each proposed method has its advantages and disadvantages. But all of them help to find either the estimation only for the observed period of time, or the estimation whose symbolic representation makes it useless for analytical methods of control or analysis. Modern research papers related to the identification of dynamic objects apply various representations of models. The most common and convenient representation of the model for further application is the representation of an object by a differential equation in the symbolic form [6-7].

In some scientific papers, a model of a dynamic object was built in the form of a differential equation according to its known structure. That is, we are talking about parametric identification. The problem solution of the parametric identification of differential equations is presented in [8-9]. Despite the fact that the papers [10] consider an arbitrary structure of the differential equation, in fact, the search for a symbolic model in the form of a linear differential equation is considered, and the choice of the structure is in determining the order of the equation.
The approach to a model of the dynamic object building as an arbitrary differential equation in a symbolic form is proposed. It obtains a solution with an unknown structure of the equation.

2. Description of the approach

Consider the statement of the identification problem using the sample data. Let be a sample of volume \( n \{y_i, t_i\}, i=1, 2, ..., n \), where \( y_i \) is the measurement of the dynamic system output at time \( t_i \). The control action, which is an input of the dynamic system, is known. The object is described by a differential equation with the known initial condition:

\[
y^{(k)} = F(y^{(k-1)}, ..., y', y, x), \]
\[
y(0) = y_0
\]

The order of the differential equation will be considered to be limited. It is necessary to build a symbolic model in the form of the differential equation that describes the relationship between the input and output of the system. The initial conditions will be considered to be known.

In this statement, the identification problem is reduced to the problem of symbolic regression solved by the genetic programming (GP) algorithm successfully [11-12].

The GP algorithm requires the representation of the individual in the form of a tree [11]. In this paper, a method for encoding a differential equation in the form of a tree is developed and described.

It is necessary to specify the maximum possible order of the differential equation \( K \) to form a tree. Therefore, we will seek for a solution to the identification problem as a differential equation of the order \( k < K \), where \( k \in \mathbb{N} \):

\[
\hat{y}^{(k)} = F(\hat{y}^{(k-1)}, ..., \hat{y}', \hat{y}, x).
\]

The functional and terminal sets are redefined to using the GP algorithm. The terminal set includes a set of all input (\( x \)) and output (\( y \)) variables, a set of constants, derivatives \( y', ..., y^{(k-1)} \). The functional set consists of the functions used by the algorithm to form the solution (+, -, /, sin, cos, etc.).

The root vertex of the tree contains not only an element of the functional set, but also the information about the maximum order of the derivative for the given individual. An example of a tree is shown in figure 1.

![Figure 1. Example of the differential equation in the form of a tree.](image)

The evolutionary steps of the algorithm were also substantially modified for the GP algorithm operating with differential equations.

A starting population is generated randomly from the given number of individuals at the initialization step. An order of the differential equation is not higher than the given value of \( K \) in the root node.
An individual in the form of a tree is converted into a formula. Then the fitness value is calculated for it:

\[
\text{fitness} = \frac{1}{1 + \text{error}},
\]

\[
\text{error} = \frac{1}{n} \sum_{i=1}^{n} \sqrt{\left(\hat{y}_i - y_i\right)^2}
\]

\[
\text{max}(y_i) - \text{min}(y_i)
\]

where \(n\) is a sample size, \(\hat{y}_i\) is a value of the individual at the \(i\)-th point, \(y_i\) is values from the original sample.

The calculating derivative must be performed at the points of the initial sample in calculating an error of correspondence of the found solution to the true one. In this paper, the output estimation of the differential equation \(\hat{y}_i\) from the points of the initial sample \(x_i\) was performed using the fourth-order Runge–Kutta method [13].

The peculiarity of the mutation in the proposed approach is a possible change in the root node containing the maximum order of the differential equation. In case of a change (mutation) of the maximum order of the differential equation, the order of the derivative must also be changed. So, in figure 2, a mutation of the maximum order for the given individual will entail a change (randomly choosing the order of the derivative lower than a new maximum order) and other selected vertices.

![Figure 2. Vertices mutating in changing the root node.](image)

3. Self-configuring of GP

In this paper, a self-configuring type of GP algorithm is used. Self-configuring means the automated selection and application of the existing variants of operators and numerical parameters [14]. Operator groups are presented in table 1.

| Operator groups | Type selected operators | Numerical values elected operators |
|-----------------|-------------------------|----------------------------------|
| Selection       | (Tournament, rank selection, proportionate) | Crossover probability |
| Crossover       | (Standard, single-point, uniform) | Mutation probability |

For operators with a selected type, self-configuring was carried out using the Population-Level Dynamic Probabilities method [15]. The numerical parameters are set according to the Success History Adaptation algorithm [16]. The application of these procedures will reduce span time significantly and will simplify the interaction with algorithms of scientists who are not specialists in the field of evolutionary algorithms.
4. Testing of approach

Differential equations of various orders were used to identify differential equations (table 2). According to the conditions from table 2, samples \( \{x_i, y_i\}, i = 1, s \) were formed, where \( s \) is a sample size, for testing the proposed approach.

| № | Differential equation | Starting points | Interval of \( x \) |
|---|------------------------|-----------------|-------------------|
| 1 | \( y' = -\frac{\cos(x)}{2\sin(y)\cos(y)\sin^2(x)} \) | \( y\left(\frac{\pi}{2}\right) = 0 \) | \([\frac{\pi}{2}; 3\pi]\) |
| 2 | \( y' = -\frac{y}{x+1} \) | \( y(-2) = -7.389 \) | \([-2;5]\) |
| 3 | \( y' = \frac{y + x^2\cos(x)}{x} \) | \( y(0,1) = 0.06 \) | \([0.1;3.5]\) |
| 4 | \( y' = 2e^{x^2} - \frac{y}{x} \) | \( y(1) = e \) | \([1;6]\) |
| 5 | \( y' = -xe^y \) | \( y(0) = 0 \) | \([0;5]\) |
| 6 | \( y'' = \frac{4(y^4 - 1)}{y^3} \) | \( y(0) = \sqrt{2} \) | \([0;5]\) |
| 7 | \( y'' = 2y' + 10y \) | \( y'(0) = 2 \) | \([0;6]\) |
| 8 | \( y'' = e^{3x} + 6y' + 9y \) | \( y(0) = 0 \) | \([0;6]\) |
| 9 | \( y'' = -2y'' - 2y + 2x^2 + 8x + 6 \) | \( y(0) = 1 \) | \([0;6]\) |
| 10 | \( y'' = -\frac{(y')^2}{1 + x^2} \) | \( y'(0) = 4 \) | \([1;6]\) |
| 11 | \( y''' = \frac{6}{x^3} \) | \( y'(0) = 1 \) | \([1;6]\) |
| 12 | \( y''' = -y \) | \( y'(0) = 1 \) | \([0;4]\) |
| 13 | \( y''' = -y' + \frac{1}{\cos(x)} \) | \( y''(0,1) = 0 \) | \([0.1;4]\) |
| 14 | \( y'''' = -4y''' - 4y'' + x - x^2 \) | \( y''(0,1) = -1 \) | \([0;5]\) |
| 15 | \( y'''' = y''' + 2x + 3 \) | \( y''''(0) = 1 \) | \([0;6]\) |

To identify each differential equation, 100 individuals were selected, and the number of generations was 200. The number of runs on each function was 30 some criteria were selected to evaluate the efficiency of the algorithm. They are an error and averaged generation number, where the solution was found. The error is taken as the standard deviation of the output of the differential equation from its estimation. The results are presented in table 3.
Table 3. Test results of the approach to the identification of differential equations with one input variable.

| Task number | Error  | Generation number |
|-------------|--------|-------------------|
| 1           | 0.0000 | 98                |
| 2           | 0.0000 | 8                 |
| 3           | 0.0000 | 97                |
| 4           | 0.0002 | 76                |
| 5           | 0.0000 | 12                |
| 6           | 0.0000 | 34                |
| 7           | 0.0000 | 75                |
| 8           | 0.0001 | 77                |
| 9           | 0.0000 | 114               |
| 10          | 0.0000 | 33                |
| 11          | 0.0002 | 102               |
| 12          | 0.0000 | 74                |
| 13          | 0.0000 | 43                |
| 14          | 0.0000 | 68                |
| 15          | 0.0000 | 81                |

For the resulting models their belonging to the type is determined to be symbolically accurate (solutions that can be reduced to the exact elementary transformations without rounding), symbolically conditionally accurate (solutions that can be reduced to exact elementary transformations using rounding) or approximate (solutions that require more complex transformations and/or have complex superior size tree structure that cannot be reduced to an exact solution) [17]. It is considered that the algorithm has found a solution if the fitness of the individual encoding the solution is not lower than 0.9999. It corresponds to the error of not more than 0.0001. For individuals whose fitness is below this value, symbolic accuracy is not determined. The results are shown in figure 3.

Figure 3. Symbolic accuracy of the received differential equations.

More often, in real problems, a dynamic object is affected by several input variables; only one input action is a rare exception. Therefore, it is necessary to study the operational efficiency of the proposed algorithmic complex with several input variables (table 4). Search results for the differential equation based on the sample data \{x_{ij}, y_i\} (i = 1, s, j = 1, J, where s is a sample size, J is a number of input variables) presented in table 5.
Table 4. Test identification problems with several input variables.

| №  | Differential equation                                                   | Starting points | Interval of \( x_j \) |
|----|------------------------------------------------------------------------|-----------------|------------------------|
| 16 | \( y' = \frac{y + x^2 \cos(x)}{x} \) \( y_0 = 0.06 \) \( x_1 \in [0.1; 6] \) \( x_2 \in [3; 3.9] \) \( x_3 \in [4.5; 10.4] \) |                |                        |
| 17 | \( y' = \frac{3x^2 e^{-x^2} - (x + y)y}{x_2} \) \( y_0 = 0 \) \( x_1 \in [-1; 1] \) \( x_2 \in [3; 8] \) |                |                        |
| 18 | \( y' = 2x_2 \sqrt{y} + \frac{2y}{x_1} \) \( y_0 = 1 \) \( x_1 \in [1; 6] \) \( x_2 \in [-1; 1] \) |                |                        |
| 19 | \( y'' = -2y' - 2y + 2x_1^2 + 8x_2 + 6 \) \( y_0 = 1 \) \( y_0' = 4 \) \( x_1 \in [0; 5] \) \( x_2 \in [2.5; 7.5] \) |                |                        |
| 20 | \( y'' = \frac{y' + x_1^2 yy'}{x_2} \) \( y_0 = 0 \) \( y_0' = 2 \) \( x_1 \in [1; 6] \) \( x_2 \in [27; 512] \) |                |                        |

Table 5. Test results of the approach to the identification of differential equations with several input variables.

| Task number | Error | Generation number |
|-------------|-------|-------------------|
| 16          | 0.0043| 89                |
| 17          | 0.0093| 47                |
| 18          | 0.0091| 75                |
| 19          | 0.002 | 94                |
| 20          | 0.0022| 68                |

5. Investigation of the proposed approach stability to the presence of noise in the sample and the small amount of data

Often in solving real problems in the measurement channels there is noise, unobservable effects. Therefore, it is necessary to study the efficiency of the approach for the case when the input effect is represented by a noisy sample. For the tasks presented in tables 2 and 3, samples were generated with noise equal to 5%, distributed according to the normal law. Figure 4 shows a case where the obtained differential equation describes the cloud of sample data well and almost completely coincides with the known true one.

Initial differential equation \( y' = \frac{y + x^2 \cos(x)}{x} \). Resulted differential equation \( \hat{y}' = \frac{1.1y + x^2 \cos(1.04x)}{x} \).

Figure 4. Solution of the identification problem by data with noise.
Table 6. Dependence of the error of the resulted solution on the noise presence in the data.

| Task number | Error Without noise | Error With noise |
|-------------|---------------------|------------------|
| 1           | 0.0000              | 0.0000           |
| 2           | 0.0000              | 0.0002           |
| 3           | 0.0000              | 0.0001           |
| 4           | 0.0000              | 0.0000           |
| 5           | 0.0000              | 0.0000           |
| 6           | 0.0000              | 0.0000           |
| 7           | 0.0002              | 0.0002           |
| 8           | 0.0004              | 0.0005           |
| 9           | 0.0001              | 0.0001           |
| 10          | 0.0002              | 0.0002           |
| 11          | 0.0003              | 0.0003           |
| 12          | 0.0327              | 0.037            |
| 13          | 0.0624              | 0.0685           |
| 14          | 0.3712              | 0.4011           |
| 15          | 0.0801              | 0.0864           |

The Student’s t-test was used to test hypotheses about the significance of differences in error. Testing the hypothesis of equality of mathematical expectations by the Student's t-test proves the absence of a statistically significant difference between errors resulted in samples with and without noise.

In considering approaches to solving the identification problem, it was noted that a significant drawback of the methods number is a significant decrease in the quality of the solution in reducing the sample size [18]. However, for some real-world processes, measuring the output characteristic is a laborious and expensive process. In this paper, the dependence of the error of the obtained solutions for identifying differential equations on the sample size is investigated. Figure 5 shows a graph of the error variation with a decreasing sample size from 400 to 50 points, averaged over the tasks from tables 2-3.

![Figure 5](image_url)

**Figure 5.** Dependence of the error on the sample size.

On the graph, you can see that the error practically does not change with a decrease in the number of sample points, but significantly increased only with a knowingly extremely small sample size.
6. Conclusion
The proposed approach can be used to identify a dynamic object from its observations automatically.

The significance of this approach is in the automatic determination of the order, structure and coefficients of the differential equation. The efficiency of this approach is considered in the identification of test differential equations of various orders and the number of input variables. The stability of the resulted solutions with noisy data and a small sample size is demonstrated.

Further investigation will be related to the automatic search for the starting point.

References
[1] Medvedev A V 1975 Identification and control for linear dynamic system of unknown order Optimization Techniques IFIP Technical Conference (Berlin–Heidelberg–New York: Springer-Verlag) pp 48–56
[2] Boguslavskiy J A 2006 Polynomial Approximations for Nonlinear Problems of Estimation and Control (Fizmat, MAIK) p 208
[3] Jovanovic O 1997 Identification of Dynamic System Using Neural Network Architecture and Civil Engineering 1 (4) 525–32
[4] Passino K M and Yurkovich S 1998 Fuzzy control (Addison Wesley Longman) p 502
[5] Burakov S V and Semenkin E S 2013 Solving variational and Cauchy problems with self-configuring genetic programming algorithm International Journal of Innovative Computing and Applications 5(3) 152–62
[6] Pashchenko F F 2006 Introduction to consistent methods of systems modeling Mathematical foundations of systems modeling (Finansy i statistika, Moscow) p 328
[7] Reimer M and Rudzicz F 2010 Identifying articulatory goals from kinematic data using principal differential analysis Proceedings of Interspeech 2010 (Makuhari, Japan) pp 1608–11
[8] Al-Duwaish H N 2000 A genetic approach to the identification of linear dynamical systems with static nonlinearities International Journal of Systems Science 31(3) 307–13
[9] Astreon K J and Bohlin T 1965 Numerical Identification of Linear Dynamic Systems from Normal Operating Records IFAC Proceedings Volumes 2(2) 65–111
[10] Ryzhikov I, Semenkin E and Panfilov I 2016 Evolutionary optimization algorithms for differential equation parameters, initial value and order identification Proceedings of the 13th International Conference on Informatics in Control, Automation and Robotics pp 168–76
[11] Koza J R 1992 The Genetic Programming Paradigm: Genetically Breeding Populations of Computer Programs to Solve Problems (Cambridge, MA: MIT Press) p 785
[12] Karaseva T S 2018 Solving classification problems by the self-configuring genetic programming algorithm Computer science, management and systems analysis. Proceedings of the V All-Russian Scientific Conference of Young Scientists with International Participation pp 217–26
[13] Bakhalov N S, Zhidkov N P and Kobelkov G M 2007 Numerical methods (Moscow: Binom) p 636
[14] Semenkin E S and Semenkina M E 2012 Self-configuring Genetic Algorithm with Modified Uniform Crossover Operator Lecture Notes in Computer Science 7331 414–21
[15] Niehaus J and Banzhaf W 2001 Adaption of Operator Probabilities in Genetic Programming Lecture Notes in Computer Science 2038 325–36
[16] Tanabe R and Fukunaga A 2013 Success-History Based Parameter Adaptation for Differential Evolution Graduate School of Arts and Sciences (Tokyo) pp 71–8
[17] Karaseva T S Self-configuring genetic programming algorithm for a Cauchy problem and variational problem in symbolic form Proceedings of the Institute for systems analysis Russian academy of sciences 69(3) 80–90
[18] Ryzhikov I and Semenkin E 2013 Evolutionary strategies algorithm based approaches for the linear dynamic system identification Lecture Notes in Computer Science 7824 477–84