

Polynomial Curve Fitting with Varying Real Powers

B. Cetisli, H. Kalkan

*Department of Computer Engineering, Suleyman Demirel University,
32260 Isparta, Turkey, phone: +90 246 211 1392, e-mail: bayramcetisli@sdu.edu.tr*

crossref <http://dx.doi.org/10.5755/j01.eee.112.6.460>

Introduction

Curve fitting or function approximation is the process of fitting to series of data points with curves [1]. By this way, the data is denoted by any mathematical function. The fitted curves can be created by regression, splines, or interpolation methods.

In general, polynomials are preferred as mathematical functions and the solution of polynomials can be performed by linear equation. Let $\mathbf{X}^{M \times N}$ be the input data and $\mathbf{y}^{M \times 1}$ be the corresponding outputs. The relation between inputs and outputs is given by linear equations [2]

$$\begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1N} \\ x_{21} & x_{22} & \cdots & x_{2N} \\ \vdots & \vdots & \vdots & \vdots \\ x_{M1} & x_{M2} & \cdots & x_{MN} \end{bmatrix} \times \begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_N \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_M \end{bmatrix}. \quad (1)$$

Using matrix notation, Eq. (1) is briefly rewritten as

$$\mathbf{X}\boldsymbol{\theta} = \mathbf{y}, \quad (2)$$

where $\boldsymbol{\theta}$ denotes the unknown parameter vector. The parameter vector can be obtained by linear algebra, if the input vector \mathbf{X} is square ($M=N$) and non-singular. The linear solution of unknown parameter vector $\boldsymbol{\theta}$ is obtained as

$$\boldsymbol{\theta} = \mathbf{X}^{-1}\mathbf{y}. \quad (3)$$

In case of $M \geq N$ where the input vector \mathbf{X} is singular the $\mathbf{X}^T \mathbf{X}$ becomes non-singular and the solution is obtained by

$$\hat{\boldsymbol{\theta}} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}, \quad (4)$$

where $\hat{\boldsymbol{\theta}}$ denotes the approximated unknown parameters. The new linear system is to include an error. Therefore, the linear solution equation in (2) is extended as below;

$$\mathbf{X}\hat{\boldsymbol{\theta}} + \mathbf{e} = \mathbf{y}, \quad (5)$$

where \mathbf{e} and $\mathbf{X}\hat{\boldsymbol{\theta}}$ denotes the error vector and the estimated output, respectively. This system can be solved by least squares estimation (LSE) method [1-2]. The LSE method minimizes the estimation error by using the sum of squared error as cost function

$$E(\boldsymbol{\theta}) = \sum_{k=1}^M (y_k - \mathbf{x}_k \hat{\boldsymbol{\theta}})^2, \quad (6)$$

where \mathbf{x}_k denotes the k^{th} input row vector. In linear networks, the unknown parameters can also be estimated by derivative based methods. The gradient vector of parameters respect to error function can be obtained by chain rule in which the optimum values of parameters are determined by iterative algorithms.

Unlike the linear problems, nonlinear problems require high order polynomials. Let $\mathbf{x}^{M \times 1}$ denotes the one-dimensional input vector and $\mathbf{y} = f(\mathbf{x})$ denotes the output where $f(\mathbf{x})$ is a high order polynomial. The relation is defined as

$$\begin{bmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^N \\ 1 & x_2 & x_2^2 & \cdots & x_2^N \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_M & x_M^2 & \cdots & x_M^N \end{bmatrix} \times \begin{bmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_N \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_M \end{bmatrix}. \quad (7)$$

Similar to linear systems, the nonlinear system can be solved by Eq. (4). In matrix notation, this nonlinear system can also be defined as

$$\mathbf{y} = f(\mathbf{x}) = \theta_0 + \theta_1 \mathbf{x} + \theta_2 \mathbf{x}^2 + \cdots + \theta_N \mathbf{x}^N. \quad (8)$$

The LSE method is commonly used in real problems such as, classification, curve fitting, control, system identification, estimation, smoothing, and so on [2-6]. The classical polynomial curve fitting uses the nonnegative integer exponents which restrict the working space. However, the errors cannot be sufficiently decreased in this restricted space. If the real exponents rather than positive

integer exponents are used in functions, the working space can be transformed from real space to complex space. This new complex polynomial with real exponents is defined as

$$y = f(\mathbf{x}) = \theta_0 + \theta_1 \mathbf{x}^{p_1} + \theta_2 \mathbf{x}^{p_2} \dots + \theta_k \mathbf{x}^{p_k} + \dots + \theta_N \mathbf{x}^{p_N}, \quad (9)$$

where θ_k and \mathbf{y} can be complex vectors and p_k denotes real power. However, this expansion brings new problems. One of them is how to determine the polynomial degrees. The second is how to determine the complex coefficients. The last is how to determine the outputs and constitute the relation between the output and the real world problems. In case of the inputs is negative real numbers and the exponents are also real numbers, the polynomial function outputs can be complex numbers. For example, $(-2)^{2.5} = 0.00 + 5.66i$, $(-2)^3 = -8$, $(-2)^{3.5} = -0.00 - 11.31i$ and $(0)^{p < 0} = \infty$. These concerns are quite important problems to be considered for curve fitting. Therefore, the input space should either be in positive region or should be mapped to positive regions. Royston and Altman were used the fractional polynomials to fit the some medical data [7]. But in their applications, some pre-determined real powers within different curves had been used as constant, and the best curve was selected among the determined curves according to their regression errors. Also, the power ranges was determined within $\{-2, 2\}$.

In this study, a hybrid method with genetic algorithm and LSE is proposed for estimating the real positive exponents for curve fitting.

Materials and Methods

A new hybrid curve fitting method, called varying real powers (VRP), is developed in this study. This method uses genetic algorithm for estimating the more accurate real exponents of non-linear systems. To determine the polynomial coefficients, the LSE is employed. The developed method is compared with traditional LSE for estimating the coefficients of polynomials.

Genetic Algorithms

Genetic algorithms (GAs) are derivative-free stochastic optimization methods based on natural selection and evolutionary process [2,8,9]. They have global search abilities to determine the optimum parameters. Therefore, GA methods have been extensively used as optimization methods in fuzzy systems, neural networks and support vector machines [2, 10, 11].

In GAs, the system parameters are encoded with binary numbers that are called as chromosome. The chromosomes constitute population that is regarded as the collection of solution vectors. For better evolutionary process, more than one chromosome is required for each individual parameter. These chromosomes are regarded as parents and used to generate new generation (new possible solutions) by crossover and mutation operations. The new generation is subjected to survivability test by considering their fitness values and survived chromosomes are regarded as parent chromosomes for the next generation.

The followings are the steps of genetic algorithm [2,8,9]:

1. An initial population is randomly created from input space.
2. Then GA creates a sequence of new populations. At each step, the algorithm uses the chromosomes in the current generation to create the next population.
 - a) The scores of the current population are determined by their fitness values.
 - b) Select the members using their fitness scores.
 - c) Some of the members in the current population that have lower fitness are chosen as elite. These elite members are moved to the next population.
 - d) The children are created from the parents with mutation and crossover operations.
 - e) The current population is exchanged with the children and so, the next generation is obtained.
3. The algorithm is terminated when the number of generations is reached or, the best solution is obtained.

The VRP method follows the below algorithm to fit the given function.

1. Set the input range and the number of powers.
 2. for $g = 1$ to generation number
 - Determine the p_k powers, $k = 1, 2, \dots, N$ using GA.
 - Determine the θ_k coefficients using LSE.
 - Evaluate the VRP function with Eq. (9).
- end*

The p_k powers can also be adapted by any derivative based methods such as steepest descent, conjugate gradient and so on [2,12]. However, there exist boundary problems on computing the gradient vector of powers for derivative

based methods. When $\frac{\partial E(\theta)}{\partial p_k} = \frac{\partial E(\theta)}{\partial y} \frac{\partial y}{\partial p_k}$ is determined

with chain rule, this gives $\frac{\partial y}{\partial p_k} = \theta_k x^{p_k} \ln(x)$. However,

this gradient consist of some boundary problems such as, $x = 0 \Rightarrow \ln(0) = -\infty$ and $x < 0 \Rightarrow \ln(x) \in \square$, where \square denotes the complex numbers. Therefore, GA is more suitable to fit the functions than derivative based methods.

Experimental Studies

The first example has only one exponent. The output function is

$$y_1 = 2.7x^{0.75} + 10, \quad (10)$$

where $x \in [0.1, 2]$. This function can be accepted as linear function.

The polynomial curve of the given function is fitted by proposed method and traditional least squares estimation (TLSE) for comparison. In comparison, adaptive neuro-fuzzy inference system (ANFIS) is also used. ANFIS is a well known function approximator based on fuzzy systems. It divides the input space to sub-regions. Each region is defined by first-order polynomials. The obtained results are plotted in Fig. 1 and given in Table 1, respectively. In Table 1, #pow and #par denotes the number of powers and number of adapted parameters of polynomials, respectively. Also, root mean square error

(RMSE) criterion is used for comparison. ANFIS produces 4 fuzzy rules and 4 polynomials for y_1 . For that reason, ANFIS polynomials were not presented in any tables.

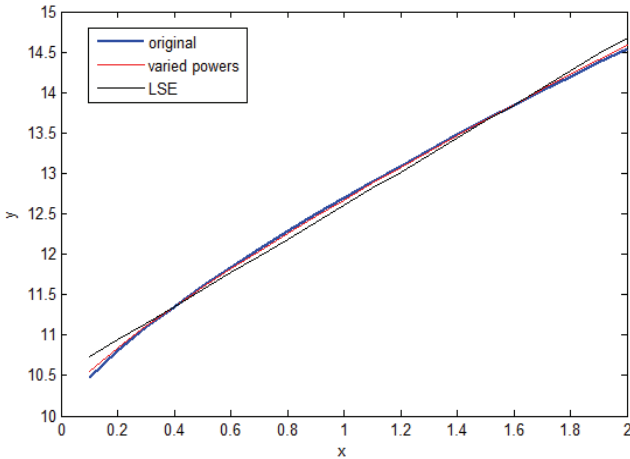


Fig. 1. Function approximations for y_1

It is observed in Table 1 that the proposed method gives the coefficients and exponent close the original values. Also, the RMSE value of VRP is the best among the methods. It can be seen in Fig. 1 that the curve of VRP is overlapped with the original function.

The second function is defined with two real powers as

$$y_2 = 5.4x^{6.7} - 50.6x^{3.49} + 2, \quad (11)$$

where $x \in [0.1, 2]$. The obtained results are given in Table 2 and plotted in Fig. 2, respectively. According to the Table 2 and Fig. 2, the VRP is best among the compared methods with same conditions. When the degree of TLSE is increased to 7, the TLSE gives promising results. However, TLSE reaches that result with high number of parameters when it is compared to VRP.

Table 2. Curve fitting results of methods for y_2

Methods	Polynomials	#powers	#parameters	RMSE
TLSE	$y = 65.46x^2 - 156.77x + 27.77$	2	3	19.49
TLSE	$y = -0.27x^7 + 4.77x^6 - 3.08x^5 + 15.45x^4 \dots$ $-42.99x^3 - 20.31x^2 + 1.23x + 1.99$	7	8	0.0004
VRP	$y = 4.57x^{6.84} - 50.14x^{3.41} + 2.21$	2	5	0.0012
ANFIS	-	1	8	0.5227

Table 3. Curve fitting results of methods for y_3

Methods	Polynomials	#powers	#parameters	RMSE
TLSE	$y = 10^8 \left(\begin{array}{l} -0.003x^{11} + 0.03x^{10} - 0.17x^9 + 0.49x^8 - 0.90x^7 \dots \\ +1.08x^6 - 0.82x^5 + 0.38x^4 - 0.1x^3 + 0.01x^2 \end{array} \right)$	11	12	50.91
VRP	$y = 10^{11} \left(\begin{array}{l} -0.54x^{6.34} + 0.002x^{4.37} - 0.92x^{7.4} + 1.13x^{8.24} + 3.74x^{7.93} \dots \\ -4.3x^{8.1} + 0.52x^{5.6} + 0.79x^{6.74} - 0.03x^{8.7} - 0.4x^{5.5} \end{array} \right)$	11	20	29.39
ANFIS	-	1	8	116.18
ANFIS	-	1	24	1.82

The third example is more complex than the previous examples. This function is defined as

$$y_3 = 1000 \exp(-x^{4.5}) \sin(10x) + 8x^{2.6} + 4.25x^{7.3} - 10, \quad (12)$$

where $x \in [0.1, 2]$. The obtained results are given in Table 3 and plotted in Fig. 3, respectively. According to the results, VRP is better than TLSE. However, the ANFIS is the best.

Table 1. Curve fitting results of methods for y_1

Methods	Polynomials	#pow	#par	RMSE
TLSE	$y = 2.08x + 10.52$	1	2	0.0958
VRP	$y = 2.71x^{0.74} + 9.98$	1	3	0.0020
ANFIS	-	1	8	0.0068

In addition to one-dimensional functions, two-dimensional surface fitting problem is also evaluated in this study. These surfaces are formulated as:

$$\begin{cases} z_1 = \left(1 + 2(x_1 - cx_1)^2 + (x_2 - cy_1)^2 \right)^{-1}, \\ z_2 = \left(1 + 2(x_1 - cx_2)^2 + (x_2 - cy_2)^2 \right)^{-1}, \\ y_4 = -(z_1 + 0.6z_2) \end{cases}, \quad (13)$$

where $x_1, x_2 \in [0.1, 2]$, $cx_1 = 1.5$, $cy_1 = 1$ and $cx_2 = 2$, $cy_2 = 1.5$ are centers of the first and second bowls.

The obtained results are given in Table 4 and plotted in Fig. 4, respectively. In Table 4, the TLSE and VRP polynomials with 21 terms were not presented due to its size. Surface fitting is more challenging than curve fitting. But, VRP is the best among the comparison methods according to RMSE values. The performance of VRP is also validated with the same problem with 20 powers and the obtained surfaces are plotted in Fig. 5.

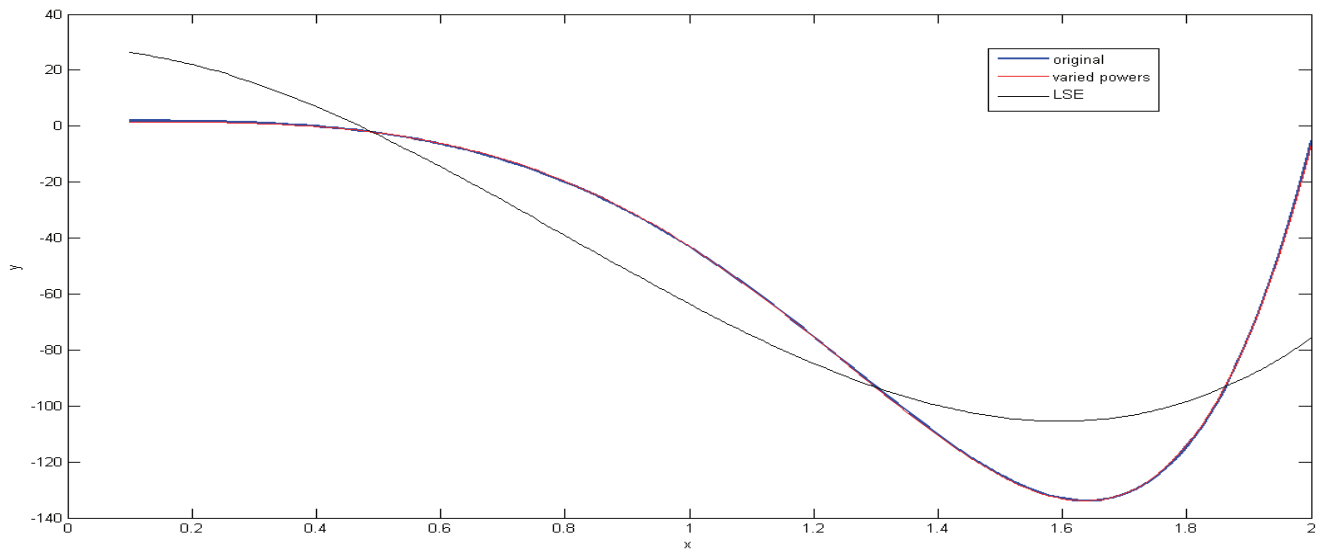


Fig. 2. Function approximations for y_2

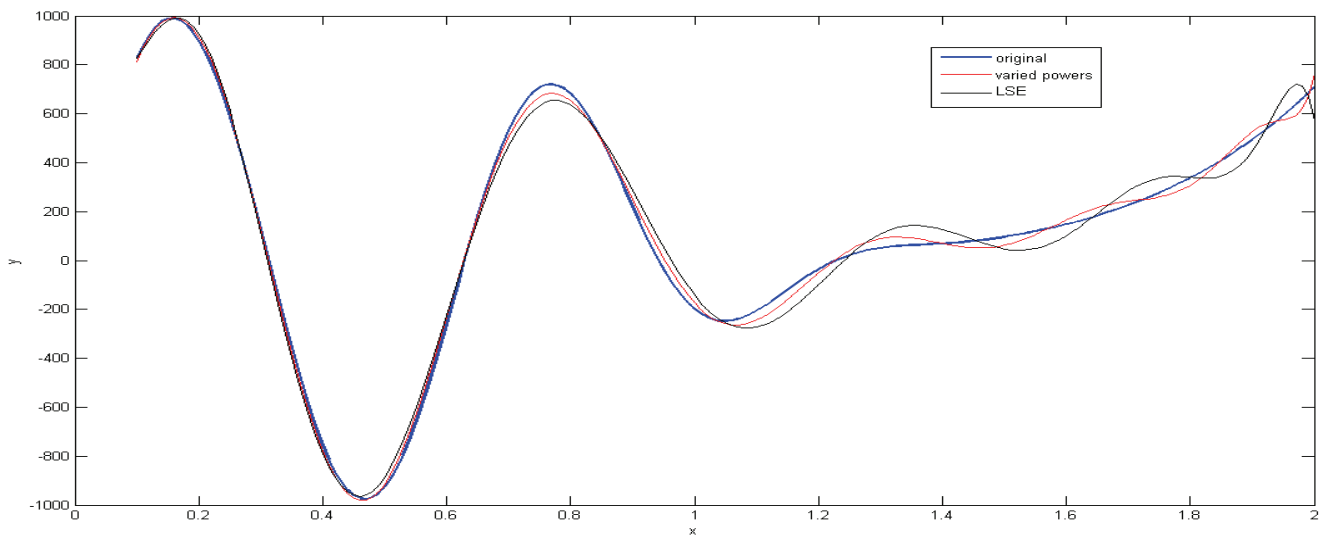


Fig. 3. Function approximations for y_3

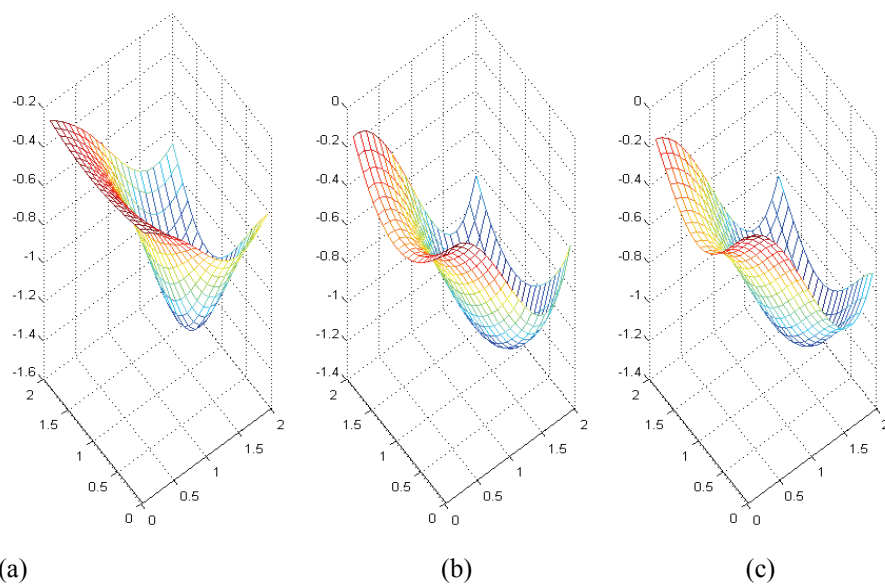


Fig. 4. Surface fitting for y_4 with 8 powers: a – Original surface; b – VRP surface; c – TLSE surface

Table 4. Curve fitting results of methods for y_4

Methods	Polynomials	#powers	#parameters	RMSE
TLSE	$y = 0.12x_1^4 - 0.16x_2^4 - 0.02x_1^3 + 0.76x_2^3 \dots$ $-0.62x_1^2 - 0.78x_2^2 + 0.02x_1 - 0.21x_2 - 0.05$	8	9	0.0748
VRP	$y = 0.04x_1^{5.05} - 1.91x_2^{1.58} - 1.21x_1^{1.44} + 1.60x_2^{1.83} + 0.76x_1^{0.97}x_2^{-0.06} - 0.19$	8	17	0.0629
TLSE	-	20	21	0.0736
VRP	-	20	41	0.0523
ANFIS	-	1	20	0.0702

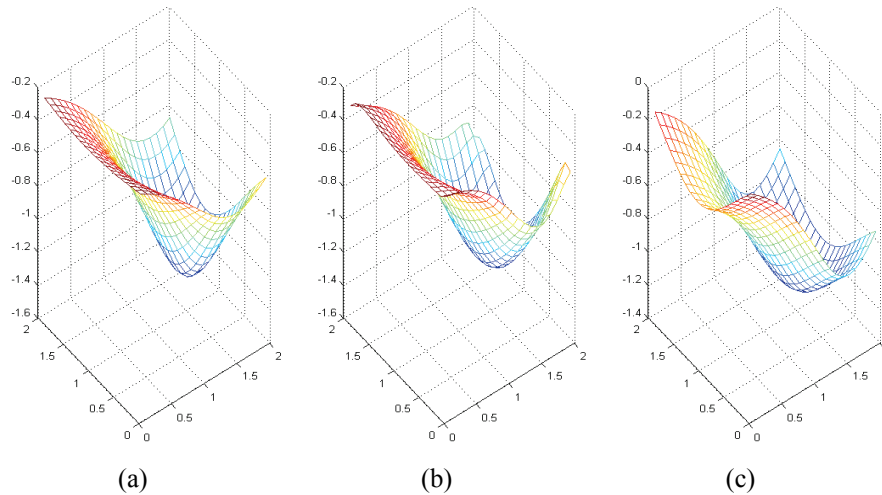


Fig. 5. Surface fitting for y_4 with 20 powers: a – Original surface; b – VRP surface; c – TLSE surface

In this study, the powers are constrained within -50 and +50 values. The extension of this range may lead us to global solution. However, this will significantly increase the computation time if genetic algorithms were used. This problem can be solved by using parallel programming.

Also it is obvious that the using of correlated powers in curves is to improve the fitting success. In this case, the number of tuned parameters is to increase rapidly. For that reason, only uncorrelated powers are used in this study.

Conclusion

A hybrid tuning method, which combines genetic algorithm and LSE, is proposed to best fit to the objected curves by using varying real powers. In this method, the genetic algorithm tunes the powers of polynomial while the LSE is tuning the coefficients of polynomial. A point needed to argue is that the method generates complex numbers when the inputs are negative and the powers are real numbers. Therefore, only the positive inputs were used in the experiments. Experimental studies show that the usage of real powers rather than positive integer numbers significantly improves the approximation performance in curve fitting problems. The proposed method can be used in a wide number of fields including pattern recognition, estimation, signal processing, and control areas, when the obtained complex results are projected to real numbers. The new polynomial classifiers can be proposed. Some new polynomial feature extraction methods or definition methods as Legendre and Zernike polynomials can also be defined. We hope that the proposed polynomial approximation can be flashed on new methods and ideas.

References

1. Daniel C. Fitting equations to data: Computer analysis of multifactor data. – Wiley–Interscience, 1999.
2. Jang J. S. R., Sun C.T., Mizutani E. Neuro–fuzzy and soft computing. – Prentice Hall – Upper Saddle River, 1997.
3. Belinska V., Kluga A., Kluga J. Application of least square method with variable parameters for GPS accuracy improvement // Electronics and Electrical Engineering. – Kaunas: Technologija, 2010. – No. 8(104). – P. 109–112.
4. Taylan O., Karagözoğlu B. An adaptive neuro–fuzzy model for prediction of student’s academic performance // Computers & Industrial Engineering, 2009. – No. 57(3). – P. 732–741.
5. Zarandi M. H. F., Türksen I. B., Sobhani J., Ramezani-pour A. A. Fuzzy polynomial neural networks for approximation of the compressive strength of concrete // Applied Soft Computing, 2008. – No. 8(1). – P. 488–498.
6. Cetisli B., Edizkan R. Estimation of adaptive neuro–fuzzy inference system parameters with the expectation maximization algorithm and extended Kalman smoother // Neural Computing and Applications, 2011. – No. 20(3). – P. 403–415.
7. Royston P., Altman D.G. Regression using fractional polynomials of continuous covariates: Parsimonious Parametric Modeling // J. of The Royal Statistical Society – Series C, 1994. – No. 43(3). – P. 429–467.
8. NirmalaDevi M., Mohankumar N., Arumugam S. Modeling and analysis of neuro–genetic hybrid system on FPGA // Electronics and Electrical Engineering. – Kaunas: Technologija, 2009. – No. 8(96). – P. 69–74.
9. Goldberg D.E. Genetic algorithms in search, optimization, and machine learning. – Addison–Wesley, 1989.
10. Mitra S., Hayashi, Y. Bioinformatics with soft computing // IEEE Transactions on Systems, Man, and Cybernetics, Part

- C: Applications and Reviews, 2006. – No. 36(5). – P. 616–635.
11. **Lauks G., Jelinskis J.** Optimal policy for LSP control in MPLS networks // *Electronics and Electrical Engineering*. – Kaunas: Technologija, 2007. – No. 5(77). – P. 65–68.
12. **Cetišli B., Barkana A.** Speeding up the scaled conjugate gradient algorithm and its application in neuro-fuzzy classifier training // *Soft Computing*. – Springer, 2010. – No. 14(4). – P. 365–378.

Received 2011 04 03

B. Cetisli, H. Kalkan. Polynomial Curve Fitting with Varying Real Powers // *Electronics and Electrical Engineering*. – Kaunas: Technologija, 2011. – No. 6(112). – P. 117–122.

In this study, the polynomial curve fitting is expanded with real powers by combining the genetic algorithm and the traditional least squares estimator. In general, integer values are used as the power of variables in traditional polynomials. However, the usage of real powers decreased the approximation error on polynomial curve fitting. In addition, the number of parameters is also decreased, when it is compared to the traditional polynomials. But, the number of adapted parameters of the proposed method is bigger than the parameters of traditional polynomials with same conditions. Although the improvements in polynomial curve fitting, the algorithm can be used for the positive input space to avoid the complex outputs. Ill. 5, bibl. 12, tabl. 4 (in English; abstracts in English and Lithuanian).

B. Cetisli, H. Kalkan. Polinominių kreivių pritaikymas keičiantis realiai galiai // *Elektronika ir elektrotechnika*. – Kaunas: Technologija, 2011. – Nr. 6(112). – P. 117–122.

Naudojant genetinį algoritmą ir mažiausiųjų kvadratų metodą, polinominės kreivės pritaikytos keičiantis realiai galiai. Taikant realias galias, sumažėja aproksimavimo paklaida. Lyginant su tradiciniu polinomu nustatytas parametų skaičiaus sumažėjimas. Tomis pačiomis sąlygomis taikant pasiūlytą metodą, gautų parametų skaičius yra didesnis nei tradiciniame polinome. Il. 5, bibl. 12, lent. 4 (anglų kalba; santraukos anglų ir lietuvių k.).