University of Ostrava

**Institute for Research and Applications of Fuzzy Modeling**

# SOME METHODS FOR ROBOT MOVEMENT CONTROL AND THEIR COMPARISON

Ondřej Polakovič, Radek Valášek

**Research report No. 111**

2006

*Submitted/to appear:*

Journal of Electrical Engineering

*Supported by:*

Project 1M0572 of the MšMT

**University of Ostrava**
**Institute for Research and Applications of Fuzzy Modeling**
**30. dubna 22, 701 03 Ostrava 1, Czech Republic**

tel.: +420-59-6160234 fax: +420-59-6120 478
e-mail: ondrej.polakovic@osu.cz, radek.valasek@osu.cz

Our problem is to control the movement of a robot. By robot we mean a moving autonomous vehicle which moves within a chosen corridor with a constant speed. It should successfully move through the chosen continuous corridor without obstacles. In this contribution we show some approaches to solving this problem. The first approach is based on classical if rules, the second approach is based on fuzzy if-then rules and extended fuzzy transform [5, 8, 9]. The third approach is based on the neural network and backpropagation algorithm [1, 6]. All of these methods are able to control the robot and its movement successfully.
**Keywords** Fuzzy Sets, Fuzzy Approximation, Neural networks, Backpropagation

*2000 Mathematics Subject Classification:* 68T01

# 1 INTRODUCTION

In this paper we deal with an autonomous vehicle with a pivoted ultrasound rotating sensor. This sensor provides partial information about the environment surrounding our vehicle. Its task is to *drive through any corridor without accident.*

This paper focuses on special methods for the control of vehicle movement which fulfils the above task. First, we form a set of classical (crisp) if-then rules and tune them until successful control of vehicle movement has been reached. Second, we apply the fuzzy transform as a fuzzy approximation method and construct a fuzzy rule base system. Third, we form a three-layer feed-forward neural network, which will be adapted by the backpropagation algorithm.

Perhaps the most important advantage of neural networks is their adaptivity. Neural networks can adjust the weights to optimize their behavior as pattern recognizers, decision makers, system controllers, predictors, etc. While fuzzy logic performs an inference mechanism under cognitive uncertainty, neural networks offer advantages, such as learning, adaptation, fault-tolerance, parallelism and generalization. However, it is generally not possible to extract rules from trained neural networks (black-box), therefore, we can not generally integrate special information about the problem into the neural network. A certain input produces a desired output, but how the network achieves this result is left to a self-organizing process.

A fuzzy rule base can be imagined as a set of fuzzy if-then rules. Fuzzy rules make it possible to use expert knowledge which can be vague. There are several possibilities for obtaining these rules. One of them is to construct rules using the extended fuzzy transform (see Section 2.1).

The paper is organized as follows: in Section 2, we define and describe fuzzy transform, in Section 3, we introduce the problem of autonomous vehicle movement control and describe the vehicle, in Section 4, we form classical crisp if-then rules, create a fuzzy rule base using an extended fuzzy transform and in the last technique we deal with the neural network and backpropagation algorithm. In Section 5 we display our results in singular methods and their basic comparison. We show some data computed from one corridor.

# 2 PRELIMINARIES

In this paper, in addition to a simple expert system based on (crisp) if-then rules, we apply three soft-computing techniques, namely fuzzy if-then rules, fuzzy transforms and neural networks. Because of lack of space, we mention only the fuzzy transform, which is less known. For the neural network, we refer the reader to a wide pool of literature [1, 6].

## 2.1 FUZZY TRANSFORM

As usual, by fuzzy set we understand a function $\mathbf{A} : X \to [0, 1]$ where $X$ is a set. A fuzzy set $\mathbf{A} : \mathbb{R} \to [0, 1]$ is triangular if $\mathbf{A}$ is piecewise linear with one element kernel and bounded connected support. In this section we recall some basic definitions from the theory of the fuzzy transform (f-transform). For other definitions and ideas see [3, 4, 5, 7, 8]. Fuzzy transform approximates a continuous function $f : X \to Y$ where $X$ and $Y$ are real closed intervals.

We deal with triangular basis functions that form fuzzy partition [7]. Each basis function can be viewed as a triangular fuzzy number. We suppose the function to be approximated given in form $(x_p, f(x_p))$ for $p = 1, \ldots, m$. The *discrete fuzzy transform* has two phases: direct and inverse.

Let $\mathbf{A}_1, \ldots, \mathbf{A}_n$ be basis functions. The n-tuple of real numbers $[F_1, \ldots, F_n]$

$$F_i = \frac{\sum\limits_{p=1}^{m} f(x_p)\mathbf{A}_i(x_p)}{\sum\limits_{p=1}^{m} \mathbf{A}_i(x_p)} \tag{1}$$

is discrete *direct fuzzy transform.*

The vector of the fuzzy transform $[F_1, \ldots, F_n]$ provides discrete representation of the original function. The approximation of $f$ is obtained using the inverse *fuzzy transform.*

Let $[F_1, \ldots, F_n]$ be the fuzzy transform of function $f$ w.r.t. $\mathbf{A}_1, \ldots, \mathbf{A}_n$. The function

$$f_n^F = \sum_{i=1}^{n} F_i \mathbf{A}_i(x) \tag{2}$$

is called *inverse fuzzy transform.*

# 3 VEHICLE DESCRIPTION AND TASK

At first, let us describe the vehicle. Simply put, our vehicle is a square metal plane with integrated circuits. At the bottom, there are three wheels. The back wheels are powered by electric engines. At the top, there is a rotating ultrasound sensor. This sensor measures distances from obstacles in its environment. In our case it measures the distance to the corridor borders.

Now we can define our problem. The task is to control the movement such that the vehicle should be able to move through any corridor. By a corridor we mean corridor without inner obstacles and without crossings.

The vehicle measures a distance $d_i^{left}$ and $d_i^{right}$ to the left and right borders of the corridor in step $i$, resp. From these data we obtain the relative distance from the center of the corridor in step $i$

$$d_i^{relative} = \frac{2d_i^{left}}{d_i^{left} + d_i^{right}} - 1. \tag{3}$$

This is the input to control program.

How does the autonomous vehicle work? In one step it scans the distances, after that, the program computes the relative distance and the method computes from this relative distance the required turn degreewhich results in the vehicle turning. In the next step, the autonomous vehicle moves and scans distances again.

# 4 IMPLEMENTATION

## 4.1 CRISP IF-THEN RULES APPROACH

Let us describe first the classical crisp if-then rules approach. By if-then rules we mean standard commands from the programming language. In our case we have used C++ and we have written rules in the form

if relative distance is in concrete interval then turn to required direction. As an example, a part of the code is

```
if ((Values[0] > -0.2) && (Values[0] < -0.1))
{       ATurnDev=VeSm;       ...    }
```

How did we proceed? First, we controlled and drove the vehicle manually and observed its behavior in a few corridors. While we were moving the vehicle, we saved the scanned and computed data to obtain more precise information. After that, we designed the first project and primary rules (commands). Then we observed the vehicle during it's self-acting drive through the corridor. After the first attempt, we modified, deleted and added rules. After that, we observed the vehicle with the new rules.

Using 13 if-then rules, the vehicle has moved through various corridors without any collision. This approach is a relatively simple and easy way to control the vehicle's motion - 13 rules are sufficient to correct the drive. One of the disadvantages is that this method has no online learning advantage as do the other two approaches. It is also applicable only to simple problems like our task of corridors without inner obstacles. For a more complicated problem it would be difficult to set proper expert rules.

## 4.2 FUZZY IF-THEN RULES APPROACH

Fuzzy if-then rules

$$\text{IF } x \text{ IS } \mathcal{A}_i \text{ THEN } y \text{ IS } \mathcal{F}_i \tag{4}$$

form so-called fuzzy rule base(FRB), where $i = 1, \ldots, n$ and $\mathcal{A}_i, \mathcal{F}_i$ are linguistic evaluating expressions [2] represented by fuzzy sets $\mathbf{A}_i \subset X$ and $\mathbf{F}_i \subset Y$.

A simple and effective method - extended fuzzy transform- can be used to learn a linguistic description from the data. As we have mentioned, for some systems, expert knowledge is difficult to obtain.For this and other cases, we have chosen an algorithm with learning (not in the neural systems sense). This algorithm deals with training data. The data was obtained by experimental human control of the vehicle. These were the same data as collected in the crisp if-then command approach. We suppose that the data is in the form $(x_p, f(x_p))$ for $p = 1, \ldots, m$.

FRB systems (4) are interpreted by fuzzy relations $\mathbf{R} : X \times Y \to [0,1]$ which can be viewed as a mapping $\mathbf{R} : X \to [0,1]^Y$ which assigns a fuzzy subset of $Y$ to each node $x \in X$. Let us be given data $(x_p, \mathbf{R}(x_p, y)), p = 1, \ldots, m$. Then the F-transform can be extended as follows

$$\mathbf{F}_i(y) = \frac{\sum_{p=1}^{m} \mathbf{R}(x_p, y)\mathbf{A}_i(x_p)}{\sum_{p=1}^{m} \mathbf{A}_i(x_p)}. \tag{5}$$

Interpretation is given by

$$\mathbf{R}_n^F(x, y) = \bigoplus_{i=1}^{n} (\mathbf{A}_i(x) \odot \mathbf{F}_i(y)) \tag{6}$$

where $\oplus$ is the Lukasiewicz t-conorm and $\odot$ is the product t-norm. The basis functions obviously fulfill the generalized orthogonality condition $\bigoplus_{i=1, i\neq j}^{n} (\mathbf{A}_i(x) = 1 - \mathbf{A}_j(x))$ which implies that the inverse F-transform lies between interpretation of the generalized Mamdani FRB and the Lukasiewicz implicative FRB

$$\bigvee_{i=1}^{n} (\mathbf{A}_i(x) * \mathbf{F}_i(y)) \leq \bigoplus_{i=1}^{n} (\mathbf{A}_i(x) \odot \mathbf{F}_i(y)) \leq \bigwedge_{i=1}^{n} (\mathbf{A}_i(x) \to_L \mathbf{F}_i(y)) \tag{7}$$

and we talk about the additive interpretation of the FRB. For more details we refer the reader to [9].

For our problem, we have used a double input - single output system. The first input variable $d_i^{relative}$ is the relative distance from the center of the corridor and the second input variable $\Delta d_i^{relative}$ is the difference in the distances between step $i$ and $i-1$, i.e. $\Delta d_i^{relative} = d_i^{relative} - d_{i-1}^{relative}$.

The FRB is composed of the rules in the form

$$\text{IF } d_i^{relative} \text{ IS } \mathcal{A}_i \; AND \; \Delta d_i^{relative} \text{ IS } \mathcal{B}_i \text{THEN } y \text{ IS } \mathcal{F}_i \tag{8}$$

where $\mathcal{F}_i$ is represented by $\mathbf{F}_i$ using the extended fuzzy transform, $\mathcal{A}_i$ and $\mathcal{B}_i$ are represented by fuzzy sets $\mathbf{A}_i$ and $\mathbf{B}_i$ - basis function of fuzzy transform.

This FRB system with the center of gravity as the defuzzification was used. Because of unsatisfactory moving, we had to put in additional expert knowledge to create another rule. Then we computed components of the fuzzy transform again to join the experimental data and expert knowledge. After that, the vehicle moved through the testing corridors successfuly.

### 4.3 NEURAL NETWORK APPROACH

In this paper, we apply the backpropagation algorithm to the problem of control of autonomous vehicle movement.

The learning problem $L$ is represented by set of pairs of vectors in the form $\{x_p, f(x)_p\}, p = 1, \ldots, k$ where $x_p, f(x)_p$ represents vectors of learning input and output patterns. In our case, we have used the same data as was used in the crisp if-then rules approach. Let a multilayer perceptron with m-n-1 topology and sigmoidal activation function be given. This supervised learning algorithm modifies the network weights to minimize the network error.

$$E_p = \frac{1}{2} \sum_{v \in U} (f(x)_p^v - a_p^v), \tag{9}$$

where $a_p^v$ is actual network output. For weight changes we assume

$$\Delta W(u,v) = -\frac{\partial E_p}{\partial W(u,v)} \tag{10}$$

for neurons $u$ and $v$ in consecutive layers.

We have used an on-line version of the backpropagation algorithm [6]. We used relative distance $d_i^{relative}$ as the input to the neural network and the output of the neural network was turn degree. To obtain minimal network error, we searched the best topology. We found the best topology and the correct weights. The topology with 8 hidden neurons has been justified as the most appropriate one. After training, our neural network was able to give the correct answer. For more results see Section 5.

## 5 RESULTS AND COMPARISON

Finally, let us to show our results and compare the three methods described above. We suggest a comparison of the methods described above as follows. The test corridor is the same in all cases. Furthermore, we compute the following characteristics:

1. Arithmetical mean of the absolute values of the relative distances in step $i$, $i = 1, \ldots, n$

$$\frac{\sum_{i=1}^{n} |d_i^{relative}|}{n}. \tag{11}$$

2. Arithmetical mean of the relative distances in step $i$, $i = 1, \ldots, n$

$$\frac{\sum_{i=1}^{n} d_i^{relative}}{n}. \tag{12}$$

3. Maximal deviation of the relative distances from the center of the corridor on the left side

$$\max_{i \in left} d_i^{relative}, \tag{13}$$

on the right side

$$\max_{i \in right} d_i^{relative} \tag{14}$$

and maximal deviation of the absolute values of relative distances

$$\max_{i=1,\ldots,n} |d_i^{relative}|. \tag{15}$$

4. The last parameter is the maximal change of the relative distances in the whole run

$$\max_{i=2,\ldots,n} |d_i^{relative} - d_{i-1}^{relative}|. \tag{16}$$

| Method | Clasic IF | Neural Network | Fuzzy Rule Base |
|---|---|---|---|
| (11) | 0.128 | 0.225 | 0.273 |
| (12) | -0.098 | -0.105 | -0.219 |
| (13) | -0.398 | -0.660 | -0.618 |
| (14) | 0.118 | 0.555 | 0.227 |
| (15) | 0.398 | 0.660 | 0.618 |
| (16) | 0.530 | 0.760 | 0.156 |
| (17) | 0.051 | 0.148 | 0.169 |

Table 1: Comparison of the methods

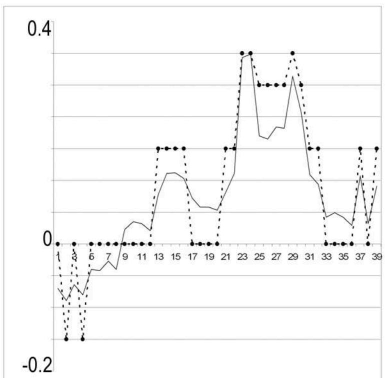


Figure 1: Relative distances and control action for crisp if-then rules. The continuous thin line represents the relative distances and the dashed line represents the corresponding control action of the autonomous vehicle.

For overall criteria, the following *coefficient of usability* has been proposed

$$\max_{i=1,\ldots,n} |d_i^{relative}| \frac{\sum_{i=1}^{n} |d_i^{relative}|}{n}. \tag{17}$$

The results are summarized in Table 1. Let us remark that (15) is not a consequence of (13) and (14) and so it is necessary for characterization of the vehicle's movement. The relative distances and the corresponding control actions are depicted in Figures 1.-3.

We found an appropriate configuration of all approaches which leads to successful control of the autonomous vehicle in all thinkable possibilities of the task. Just recall that the task is to control the movement of the vehicle, and that the vehicle is able to move through any corridor without crossings and inner obstacles. To obtain as general conclusions as possible, a huge amount of tests and experiments in different corridors was realized.

In the case of the crisp if-then rules approach, we set the sufficient number of rules and tuned the interval to obtain the best characteristics defined above. When the vehicle was controlled by crisp if-then commands, we obtained a successful drive. After some additions, changes and "tuning" of intervals in the if commands, the control program was able to drive the robot without any accident. As can be seen in Table 1, this method provides the most accurate results. We should choose the method with the smallest value of the coefficient(17). The smaller coefficient, the better usability, and the vehicle remains nearer to the center of the corridor. This approach is inexpensive from a computional point of view, which is the undoubtable advantage as well as imbuing the possibility of constructing the model expertly. On the other hand, let us stress that this expert approach is not always possible. One of the disadvantages is the lack of an online approach. Another disadvantage is that this approach is applicable only to simple tasks and more advanced tasks cause problems.

The second approach based on a fuzzy rule base (FRB) is a nice demonstration of the fuzzy transform as a tool for the creation of fuzzy if-then rules. From the obtained learning data, we constructed initial

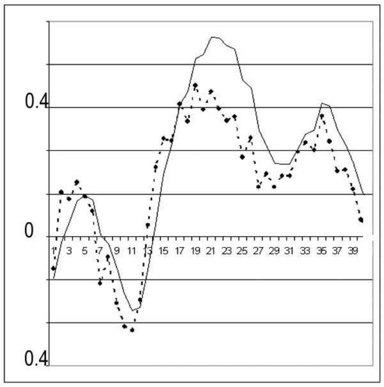


Figure 2: Relative distances and control action for fuzzy if-then rules. The continuous thin line represents the relative distances and the dashed line represents the corresponding control action of the autonomous vehicle.

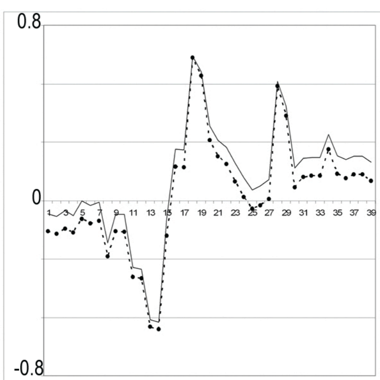


Figure 3: Relative distances and control action for neural method. The continuous thin line represents the relative distances and the dashed line represents the corresponding control action of the autonomous vehicle.

| Crisp if-then | | Fuzzy if-then | | Neural Net | |
|---|---|---|---|---|---|
| Distance | Action | Distance | Action | Distance | Action |
| -0.093 | 0 | -0.129 | -75 | -0.062 | -106 |
| -0.118 | -150 | -0.017 | 103 | -0.073 | -113 |
| -0.085 | 0 | 0.044 | 87 | -0.046 | -96 |
| -0.107 | -150 | 0.111 | 127 | -0.068 | -110 |
| -0.053 | 0 | 0.128 | 94 | -0.001 | -64 |
| -0.055 | 0 | 0.113 | 60 | -0.022 | -79 |
| -0.036 | 0 | 0.005 | -110 | -0.009 | -70 |
| -0.054 | 0 | -0.024 | -48 | -0.194 | -192 |
| 0.031 | 0 | -0.100 | -155 | -0.062 | -106 |
| 0.046 | 0 | -0.176 | -210 | -0.065 | -108 |
| 0.042 | 0 | -0.227 | -218 | -0.305 | -261 |
| 0.028 | 0 | -0.221 | -147 | -0.314 | -267 |
| 0.104 | 150 | -0.105 | 27 | -0.545 | -433 |
| 0.148 | 150 | 0.037 | 161 | -0.555 | -441 |
| 0.149 | 150 | 0.192 | 228 | -0.084 | -121 |
| 0.137 | 150 | 0.280 | 223 | 0.235 | 117 |
| 0.096 | 0 | 0.409 | 307 | 0.230 | 113 |
| 0.077 | 0 | 0.448 | 267 | 0.660 | 489 |
| 0.078 | 0 | 0.551 | 352 | 0.587 | 428 |
| 0.071 | 0 | 0.565 | 295 | 0.339 | 207 |
| 0.108 | 150 | 0.618 | 336 | 0.275 | 151 |
| 0.148 | 150 | 0.616 | 297 | 0.243 | 124 |
| 0.391 | 300 | 0.591 | 270 | 0.172 | 65 |
| 0.398 | 300 | 0.580 | 279 | 0.555 | 400 |
| 0.227 | 250 | 0.485 | 185 | 0.104 | 12 |
| 0.221 | 250 | 0.461 | 230 | 0.050 | -28 |
| 0.245 | 250 | 0.327 | 116 | 0.066 | -16 |
| 0.243 | 250 | 0.276 | 147 | 0.095 | 5 |
| 0.352 | 300 | 0.227 | 115 | 0.545 | 392 |
| 0.275 | 250 | 0.224 | 141 | 0.432 | 291 |
| 0.146 | 150 | 0.223 | 141 | 0.149 | 46 |
| 0.125 | 150 | 0.271 | 197 | 0.192 | 81 |
| 0.056 | 0 | 0.319 | 219 | 0.196 | 85 |
| 0.066 | 0 | 0.328 | 200 | 0.196 | 84 |
| 0.056 | 0 | 0.414 | 281 | 0.302 | 175 |
| 0.038 | 0 | 0.405 | 220 | 0.204 | 91 |
| 0.143 | 150 | 0.330 | 151 | 0.187 | 77 |
| 0.043 | 0 | 0.287 | 156 | 0.203 | 90 |
| 0.121 | 150 | 0.228 | 110 | 0.202 | 90 |

Table 2: Table of relative distances and control action for crisp if-then rules, fuzzy if-then rules and neural networks in the test corridor. Control action is the difference between left wheel speed and right wheel speed.

FRB using fuzzy transform. After the first run, we had to add further rules to cope with all situations of the vehicle's position. Then the vehicle was able to move through the corridor without any collision. The fuzzy approach provided a control with typical "fuzzy" features. This means that a lower accuracy was compensated for by the smooth behavior of the vehicle (Compare Fig 1, 2, 3). From the subjective point of view, the vehicle trajectory controlled by FRB was the closest one to human understanding. Let us stress that the fuzzy if-then approach allows expert identification and automatic generation (approximation) as well as their combination.

The last compared method is based on a neural network which was adapted by a backpropagation algorithm. In the case of the neural network approach, we found the best topology and weight of the feed-forward neural network to obtain not only the minimal network error but also the best characteristics which describe the usability of this method. Only a neural network with one hidden layer has been tested since riches structures would lead to a higher computational complexity in comparison with the other methods. Single-hidden layer neural networks with different numbers of neurons in the hidden layer have been tested and their accuracy measured. The best control has been performed by a neural network with 8 hidden neurons. More neurons lead to the overfitting problem [6]. We have taken advantage of the on-line learning algorithm. Because the learning patterns did not cover all possibilities after first learning, the neural network was not able to make decisions in some situations and the generalization property led to wrong answers and we had to add another learning pattern. Finally, the network was able to control vehicle movement and the vehicle passed through the corridor without any accident.

In this article, we have compared three approaches of movement control of an autonomous vehicle. We have tested the autonomous vehicle described above on many corridors. The computed data and coefficients are from the corridor in the shape of the letter S. It seems that the motion of the vehicle controlled by the crisp if-then rules method is the best. On the corridors with simpler shapes, the differences are smaller and obviously on the corridors with more complicated shapes, the results from all compared methods are worse; the first method is not the best in all cases. If we were able to minimize the passed distances in one time step and speed up the measuring process, we would be able to obtain more precise results, but due to the character of the vehicle and the ultrasound sensor, it is not possible. All methods and their fundamentals, realizations and advantages have been discussed. The final decision of a chosen method for every control task has to be made based on several criteria e.g. identification (automatic - FRB, neural networks, expert - crisp if-then, fuzzy if-then, combination fuzzy if-then FRB), accuracy (crisp if-then), computational expanses (crisp if-then only for simple tasks).

In future we propose to use other neural networks techniques and extend the task of the vehicle.

# References


[1] Fullér, R.(2000). *Introduction to Neuro-Fuzzy Systems*, Advances in Soft Computing Series, Springer-Verlag, Heidelberg, New York.

[2] Novák V. and Perfilieva I. (1999), Evaluating linguistic expressions and functional fuzzy theories in fuzzy logic. In: *L.A. Zadeh, J. Kacprzyk (Eds.) Computing with Words in Information/Intelligent Systems 1*, Springer-Verlag, Heidelberg, 383-406.

[3] Perfilieva, I.(2001). Normal forms for fuzzy logic functions and their approximation ability. *Fuzzy Sets and Systems*, 124, pp. 371-384.

[4] Perfilieva, I.(2003). Fuzzy approach to solution of differential equations with imprecise data: Application to reef Growth Problem. *Fuzzy Logic in Geology*, Academic Press, Amsterdam, pp. 275-300.

[5] Perfilieva, I.(2005). Fuzzy transform: Theory and Appliations. *Fuzzy Sets and Systems*, 157, pp. 993-1023.

[6] Rojas, R.(1996) *Neural Networks. A Systematic Introduction*, Springer–Verlag Berlin Heidelberg.

[7] Ruspini, E.H.(1969). A new approach to clustering. *Inform. and Control*, 15, pp. 22-32.

[8] Štěpnička M. and Valášek R.(2004). Fuzzy transforms and their application to wave equation. *Journal of Electrical Engineering*, 55, pp. 7-10.

[9] Štěpnička M. and Valášek R. (2005), Dynamic robot control based on fuzzy approximation. *Journal of Electrical Engineering.* 56(12/s), pp. 17-20.

**Acknowledgement** This investigation has been supported by project 1M0572 of the MŠMT ČR.

Ondřej Polakovič(Mgr.) is a postgraduate student of Applied mathematics - Fuzzy modeling at the Faculty of Science of the University of Ostrava. His supervisor is Professor Vilém Novák.

Radek Valášek (Mgr.) is a postgraduate student of Applied mathematics - Fuzzy modeling at the Faculty of Science of the University of Ostrava. His supervisor is Professor Irina Perfilieva.