POLITECHNIKA KOSZALIŃSKA

Zeszyty Naukowe
Wydziału Elektroniki i Informatyki

Nr 10

KOSZALIN 2016

Zeszyty Naukowe Wydziału Elektroniki i Informatyki Nr 10

ISSN 1897-7421
ISBN 978-83-7365-443-3

Projekt okładki
Tadeusz Walczak

Skład, łamanie
Maciej Bączek

Wydawnictwo Uczelniane Politechniki Koszalińskiej
75-620 Koszalin, ul. Racławicka 15-17

Koszalin 2016, wyd. I, ark. wyd. 12,26, format B-5, nakład 100 egz.
Druk: INTRO-DRUK, Koszalin

Spis treści

Adam Słowik
Department of Electronics and Computer Science
Koszalin University of Technology
aslowik@ie.tu.koszalin.pl

Modified Fan Roulette Selection Method for Application in Evolutionary Algorithms

Key words: artificial intelligence, evolutionary algorithms, selection methods.

1. Introduction

The phase of selection is an important element among particular steps occurring in genetic or evolutionary algorithms. Due to selection operation, new populations i.e. sets of potential solutions are created. In literature we can find different methods of selection used in genetic algorithms, for example roulette method [1, 2, 3], elitist method [4], deterministic methods [5], random choice method according to the rest [1] (with repetition and without repetition), randomly tournament method [6]. But the most common selection methods used in practice are roulette and elitist methods. According to the schemata theorem, more copies are generated from the best individuals (chromosomes), the same number of copies are generated from average quality individuals, and the worse individuals are dying. However in the roulette selection the best chromosome (solution) can be destroyed and schemata coded in it will stop to spread out. To avoid this situation the elitist selection is used, in which the best individual found is remembered and replaces an individual with the worst fitness, in the next generation (when the best individual did not survive). With such an approach we know for sure that the best solution found will not be destroyed. In papers [7, 8] a modification of the roulette method has been presented. This modification depended on increasing survival probability for the best individual (surviving schemata existing in it) without guarantee that the best individual will pass to the next population for sure (thus, we assure a certain random factor during selection). This modification has been named a fan roulette selection (FRS). The results of test function minimization obtained using fan roulette selection presented in papers [7, 8] have been promising in relation to the results obtained using roulette selection, and elitist selection method. The fan roulette selection described in papers [7, 8] depends on increase of selection probability of the best individual in selection to the next population by simultaneous decrease of

selection chances for other solutions (individuals). In the fan roulette selection the relative fitness values for particular individuals i.e. the probability selection values of individuals passing to the next population have been modified using formula (1), and (2):

* for the best individual

$$rfitness'_{max} = rfitness_{max} + (1 - rfitness_{max}) \cdot a \tag{1}$$

* for other individuals

$$rfitness' = (1 - rfitness'_{max}) \cdot \left(rfitness + \frac{rfitness_{max}}{M-1}\right) \tag{2}$$

where:

rfitness'$_{max}$ – new relative fitness of the best individual; *rfitness*$_{max}$ – relative fitness of the best individual; *a* – parameter causing the "fan expansion" ∈ [0, 1]; *rfitness'* – new relative fitness of chosen individual; *rfitness* – relative fitness of chosen individual; *M* – number of individuals in population.

The value changes of selection probabilities for given individual (potential solution) for different values of parameter *a* are shown in Figure 1.

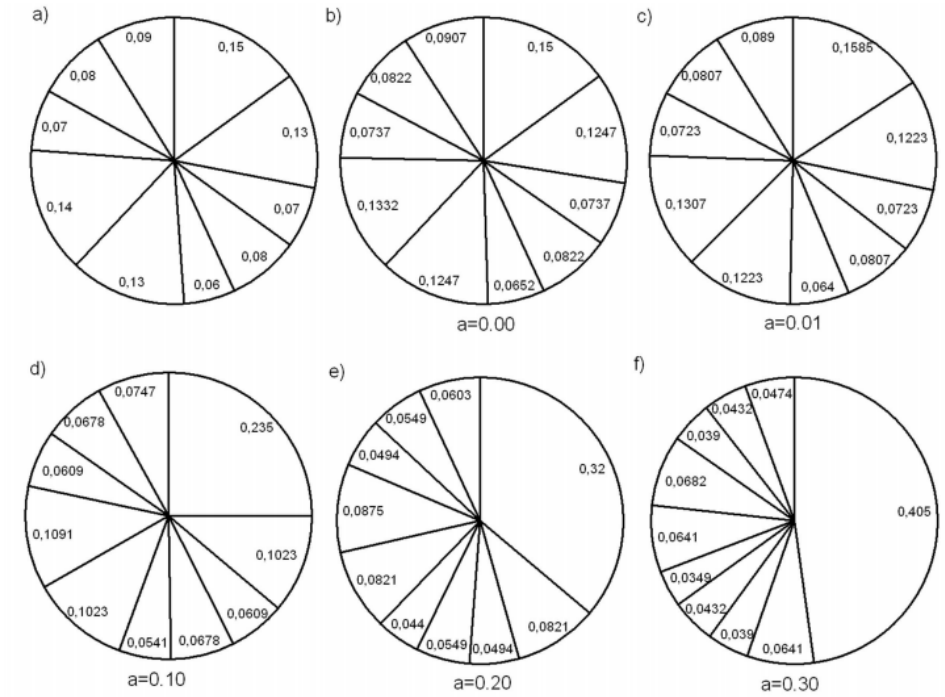


Figure 1. Roulette wheel: roulette selection (a), fan roulette selection for different values of *a* (b, c, d, e, f)

However, the fan roulette selection algorithm presented in papers [7, 8] has a inconvenience, that is for low values of *a* parameter, besides increasing of selection probability for the best individual, the selection probability values of the worse individuals selected to the new population are also increased at the expense of average quality individuals [9]. In this paper a modification of the fan roulette selection is presented. This modification has been named proportional fan roulette selection (PFRS). The proposed method eliminates early described inconvenience. The effectiveness of PFRS method has been checked by minimization of ten test functions chosen from literature. Results obtained using PFRS method have been compared with results obtained using roulette method, elitist method, and the fan roulette selection method (FRS).

2. Proportional Fan Roulette Selection – PFRS

The proportional fan roulette selection depends on increasing the selection probability for the best individual selected to the new population with simultaneous proportional decreasing of selection chances for other individuals selected to the new population. In PFRS method the formula (1) remains not changed, but the formula (2) is changed to the following form:

$$rfitness' = rfitness \cdot \left(\frac{rfitness_{max} - rfitness'_{max}}{\sum_{i=1}^{M} rfitness_i - rfitness_{max}} + 1 \right) \tag{3}$$

where:

$rfitness'_{max}$ – new relative fitness of the best individual; $rfitness_{max}$ – relative fitness of the best individual; $rfitness'$ – new relative fitness of chosen individual; $rfitness$ – relative fitness of chosen individual; M – number of individuals in population.

Depending on the value of parameter *a*, the values of selection probabilities for given individuals are changed as is shown in Figure 2.

It can be seen from Figure 2, that probability of survival of the best solution in the next generation is increasing and the probabilities of survival for all other solutions diminish proportionally in the PFRS method.

In the Figure 3, the relative fitness RF values of individuals shown in Figure 1a, and Figure 2a are presented for the purpose of more careful comparison of both methods. However, the β scaling coefficient values (see Appendix) for different values of *a* parameter are presented in particular columns of Figure 3. The β scaling coefficient is defined as follows:

$$\beta = \frac{rfitness'}{rfitness} \tag{4}$$

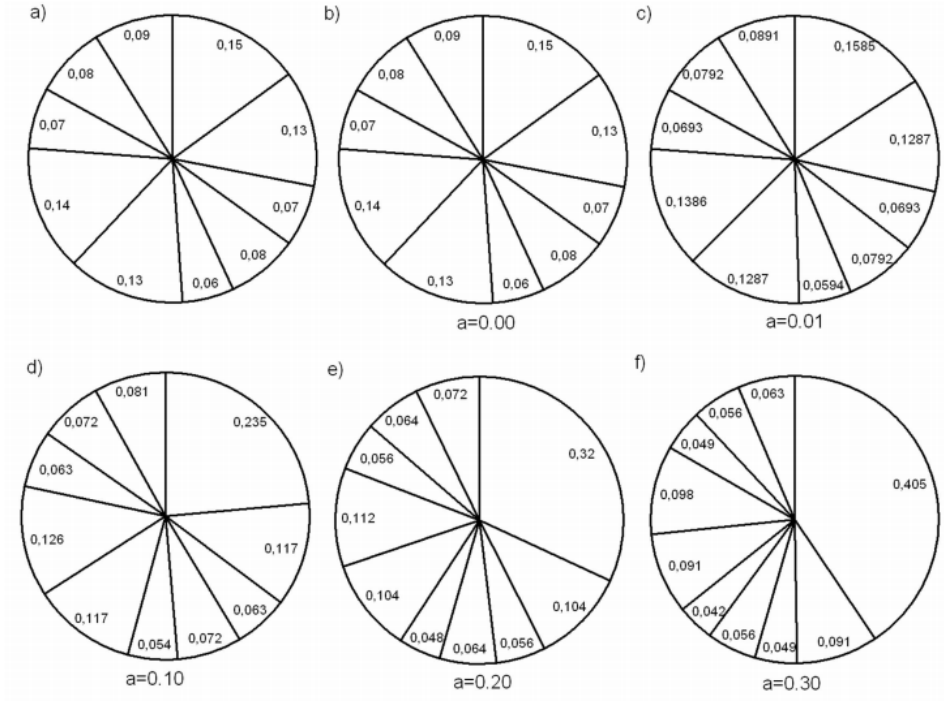


Figure 2. Roulette wheel: roulette selection (a), proportional fan roulette selection for different values of *a* (b, c, d, e, f)

and points out the ratio of relative fitness of individual after scaling (*rfitness'*), and before scaling (*rfitness*).

The symbols used in Figure 3, are the same as in equations (1-3).

It can be seen from Figure 3, PFRS method has better properties than FRS method.

No.	RF	β (for a=0.0)		β (for a=0.01)		β (for a=0.1)	
		FRS	PFRS	FRS	PFRS	FRS	PFRS
1	0.15	1.0	1.0	1.0566	1.0566	1.5667	1.5667
2	0.13	0.9590	1.0	0.9410	0.99	0.7866	0.9
3	0.07	1.0524	1.0	1.0334	0.99	0.8706	0.9
4	0.08	1.0271	1.0	1.0084	0.99	0.8479	0.9
5	0.06	1.0861	1.0	1.0668	0.99	0.9010	0.9
6	0.13	0.9590	1.0	0.9410	0.99	0.7866	0.9
7	0.14	0.9512	1.0	0.9333	0.99	0.7796	0.9
8	0.07	1.0524	1.0	1.0334	0.99	0.8706	0.9
9	0.08	1.0271	1.0	1.0084	0.99	0.8479	0.9
10	0.09	1.0074	1.0	0.9889	0.99	0.8302	0.9
No.	RF	β (for a=0.2)		β (for a=0.3)		β (for a=0.9)	
		FRS	PFRS	FRS	PFRS	FRS	PFRS
1	0.15	2.1333	2.1333	2.7	2.7	6.1	6.1
2	0.13	0.6312	0.8	0.4928	0.7	0.0194	0.1
3	0.07	0.7059	0.8	0.5582	0.7	0.0287	0.1
4	0.08	0.6857	0.8	0.5405	0.7	0.0262	0.1
5	0.06	0.7329	0.8	0.5818	0.7	0.0321	0.1
6	0.13	0.6312	0.8	0.4928	0.7	0.0194	0.1
7	0.14	0.6250	0.8	0.4873	0.7	0.0186	0.1
8	0.07	0.7059	0.8	0.5582	0.7	0.0287	0.1
9	0.08	0.6857	0.8	0.5405	0.7	0.0262	0.1
10	0.09	0.6699	0.8	0.5267	0.7	0.0242	0.1

Figure 3. Comparison of relative fitness RF and β coefficient values for different values of *a* parameter in fan roulette selection FRS and proportional fan roulette selection PFRS methods

3. Description of Experiments

Experiments were performed using evolutionary algorithm with individual representations in the form of lists of real numbers (each gene was represented by a real number from a given range). One point crossover and uniformly distributed mutation are used. The several test functions (from literature [1, 7, 8]) are chosen for verification and comparison of different selection methods (abbreviation GM stands for global minimal value):

a) De Jong function F1

$$\sum_{i=1}^{3} x_i^2 \; ; \; -5.12 \le x_i \le 5.12; \; GM=0 \text{ in } (x_1, x_2, x_3) = (0, 0, 0)$$

b) De Jong function F2

$100 \cdot \left(x_1^2 - x_2\right)^2 + (1 - x_1)^2$; -2.048 ≤ x_i ≤ 2.048; GM=0 in (x_1, x_2) = (1, 1)

c) De Jong function F3

$\sum_{i=1}^{5}$ *integer* (x_i); -5.12 ≤ x_i ≤ 5.12;

GM=–25 for all –5.12 ≤ x_i ≤ -5.0

d) De Jong function F4

$\sum_{i=1}^{30} i \cdot x_i^4$; -1.28 ≤ x_i ≤ 1.28; GM=0 in (x_1, x_2, ..., x_{30}) = (0, 0, ..., 0)

e) De Jong function F5

$$\frac{1}{1/K + \sum_{j=1}^{25} f_j^{-1}(x_1, x_2)}$$, where $f_j(x_1, x_2) = c_j + \sum_{i=1}^{2} \left(x_i - a_{ij}\right)^6$,

and –65.536 ≤ x_i ≤ 65.536, K=500, c_j = j, and

$$\left[a_{ij}\right] = \begin{bmatrix} -32 & -16 & 0 & 16 & 32 & -32 & -16 & \ldots & 0 & 16 & 32 \\ -32 & -32 & -32 & -32 & -32 & -16 & -16 & \ldots & 32 & 32 & 32 \end{bmatrix}$$

GM=0.998 in (x_1, x_2) = (-32, -32)

f) Schaffer function F6

$$0.5 + \frac{\sin^2 \sqrt{x_1^2 + x_2^2} - 0.5}{\left[1.0 + 0.0001 \cdot \left(x_1^2 + x_2^2\right)\right]^2}$$; -100 ≤ x_i ≤ 100;

GM=0 in (x_1, x_2) = (0, 0)

g) Schaffer function F7

$\left(x_1^2 + x_2^2\right)^{0.25} \cdot \left[\sin^2\left(50 \cdot \left(x_1^2 + x_2^2\right)^{0.1}\right) + 1.0\right]$; -100 ≤ x_i ≤ 100;

GM=0 in (x_1, x_2) = (0, 0)

h) Goldstein-Price function F8

$$\left[1 + \left(x_1 + x_2 + 1\right)^2 \cdot \left(19 - 14 \cdot x_1 + 3 \cdot x_1^2 - 14 \cdot x_2 + 6 \cdot x_1 \cdot x_2 + 3 \cdot x_2^2\right)\right] \cdot$$
$$\left[30 + \left(2 \cdot x_1 - 3 \cdot x_2\right)^2 \cdot \left(18 - 32 \cdot x_1 + 12 \cdot x_1^2 + 48 \cdot x_2 - 36 \cdot x_1 \cdot x_2 + 27 \cdot x_2^2\right)\right]$$

-2 ≤ x_i ≤ 2; GM=3 in (x_1, x_2) = (0, -1)

i) Six-humps camel back function F9

$$\left(4-2.1\cdot x_1^2+\frac{x_1^4}{3}\right)\cdot x_1^2+x_1\cdot x_2+\left(-4+4\cdot x_2^2\right)\cdot x_2^2 \quad ;$$

$-3 \le x_1 \le 3$ and $-2 \le x_2 \le 2$

GM=–1.0316 in $(x_1, x_2) = (-0.0898, 0.7126)$ and $(0.0898, -0.7126)$

j) Coldville function F10

$$100\cdot\left(x_2-x_1^2\right)^2+\left(1-x_1\right)^2+90\cdot\left(x_4-x_3^2\right)^2+\left(1-x_3\right)^2+$$
$$+10.1\cdot\left(\left(x_2-1\right)^2+\left(x_4-1\right)^2\right)+19.8\cdot\left(x_2-1\right)\cdot\left(x_4-1\right),$$

$-10 \le x_i \le 10$; GM=0 in $(x_1, x_2, x_3, x_4) = (1, 1, 1, 1)$

In Figures 4-8, the 3D graphical representations of each test function are shown. In the case of test functions having more variables than two (for example function F4 or F10), the graphical function representation based only on their two first variables have been shown. For test function F10, it is assumed, that variables x_3, and x_4 are equal to 1.

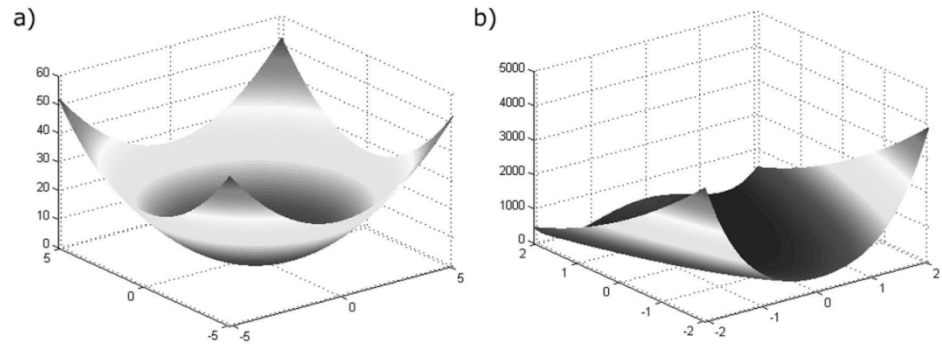


Figure 4. Graphical representation of De Jong function F1 (a), and De Jong function F2 (b)

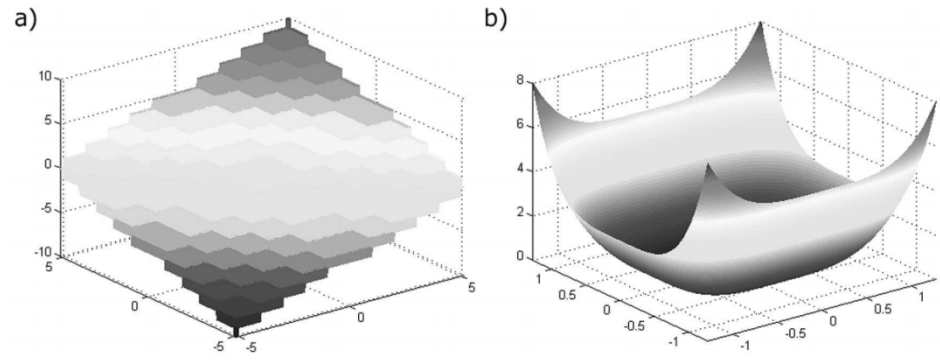


Figure 5. Graphical representation of De Jong function F3 (a), and De Jong function F4 (b)

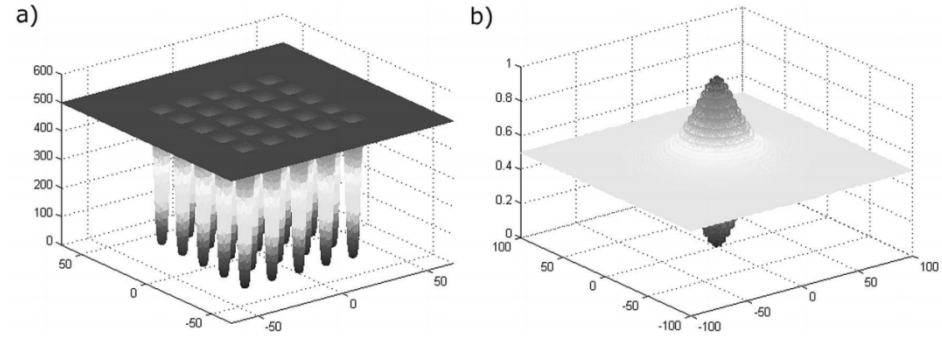


Figure 6. Graphical representation of De Jong function F5 (a), and Schaffer function F6 (b)

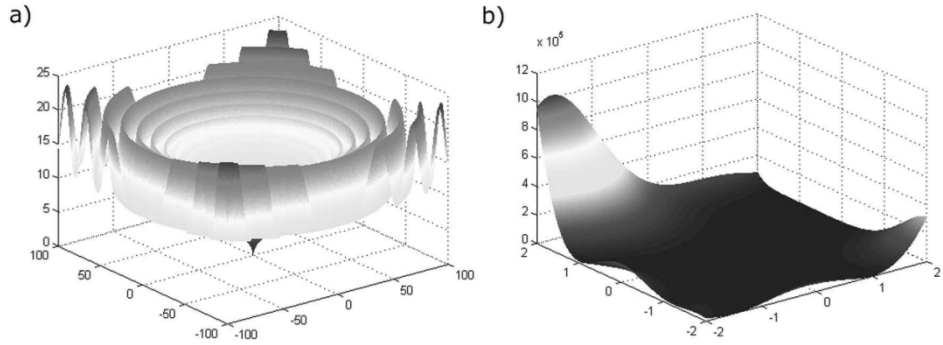


Figure 7. Graphical representation of Schaffer function F7 (a), and Goldstein-Price function F8 (b)

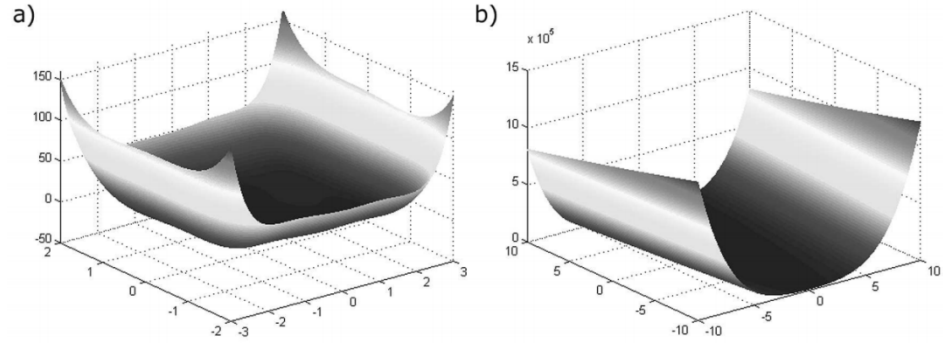


Figure 8. Graphical representation of Six-humps camel back function F9 (a), and Coldville function F10 (for x_3=1 and x_4=1) (b)

The evolutionary algorithms searched extremes of the test functions for different selection methods (roulette, elitist, fan roulette, proportional fan roulette). In the first experiment an evaluation of function minimum values which have been found by algorithms were performed. The evolutionary algorithm used for this purpose had following parameters: cross-over probability 0.5, mutation probability 0.1, number of individuals in population 100, the fan "expansion" *a* parameter value 0.3, number of generations 100. The computation has been repeated 100-fold. In Figure 9 the best function minimum values obtained after 100-fold repetition of evolutionary algorithm are shown.

TF	GM	RS	ES	FRS	PFRS
F1	0	$1.1293 \cdot 10^{-3}$	$4.2520 \cdot 10^{-4}$	$\mathbf{9.2702 \cdot 10^{-7}}$	$9.8702 \cdot 10^{-7}$
F2	0	$9.5024 \cdot 10^{-3}$	$3.3844 \cdot 10^{-4}$	$\mathbf{1.0770 \cdot 10^{-4}}$	$3.1188 \cdot 10^{-4}$
F3	-25	**-25**	**-25**	**-25**	**-25**
F4	0	23.1877	3.7063	0.0911	**0.0664**
F5	0.998	**0.9980**	**0.9980**	**0.9980**	**0.9980**
F6	0	$9.7285 \cdot 10^{-3}$	$9.7160 \cdot 10^{-3}$	$1.7518 \cdot 10^{-3}$	$\mathbf{9.5864 \cdot 10^{-4}}$
F7	0	0.05503244	0.03640455	0.01906307	**0.01650163**
F8	3	3.506820	3.003203	3.000057	**3.000003**
F9	-1.0316	-1.0311873	**-1.0316**	**-1.0316**	**-1.0316**
F10	0	41.1205836	0.91838732	0.13542065	**0.04534423**

Figure 9. The best minimal function values after 100-fold evolutionary algorithm repetition

TF	GM	RS	ES	FRS	PFRS
F1	0	0.26796329	0.01061643	0.00045582	**0.00041223**
F2	0	0.05785460	0.01170290	**0.00466204**	0.00507883
F3	-25	-24.36	**-25**	**-25**	**-25**
F4	0	43.6227861	11.5860708	0.63902525	**0.54373476**
F5	0.998	3.97313152	1.05175371	0.99825404	**0.99824610**
F6	0	0.04817742	0.02215023	**0.01745605**	0.01819210
F7	0	0.22630675	0.12623553	0.12286859	**0.11739955**
F8	3	27.3174015	3.2903303	3.0099151	**3.0098951**
F9	-1.0316	-0.92488558	-1.0300348	**-1.0315323**	-1.0314822
F10	0	863.908505	20.456860	5.24611426	**4.48156882**

Figure 10. The average minimal function values after 100-fold evolutionary algorithm repetition

In Figure 10 the average values of function minima after 100-fold repetition of evolutionary algorithm are shown. In Figure 9, and Figure 10, the symbols are as follows: TF - test function, GM - global minimum value, RS - roulette selection method, ES - elitist selection method, FRS – fan roulette selection method, PFRS - proportional fan roulette selection method. In both Tables the bold fonts represent the best obtained results.

It follows from Figure 9, that the solutions found using the proportional fan roulette selection (after 100 generations) are much better than solutions found in the same run-time using roulette selection, and are better (or comparable) than solutions found using elitist selection. In comparison to fan roulette selection method (FRS), the better or comparable results in 8 cases on 10 possible have been obtained using proposed proportional fan roulette selection (PFRS) method. Also, average values after 100 repetitions (Figure 10) show that the proportional fan roulette selection (for selected parameter a) is more stable, than roulette selection or elitist selection. Described PFRS selection method has the least deviations of obtained results from the best obtained solution. It is understandable, because larger part of the best individuals has a chance to enter to the next population. The highest differences we can find for De Jong function F4, Goldstein-Price function F8, and Coldvill function F10. Those differences refer to both the best solutions found after 100 generations and average values of solutions found in 100 subsequent tests. In the case of De Jong function F4 it is probably caused by the fact, that this function has 30 variables, what with mutation probability of order of 0.1, and population size of order of 100 causes that during one generation, approximately 300 genes can be mutated. This means that each individual in the population will undergo mutation, that is the searching will have more random character. It is possible to conclude from this, that the proportional fan roulette selection gives much better results, than roulette selection, and elitist selection in the case of existence of large number of

mutated genes in population. The results obtained in Figure 10 for PFRS selection are in 7 cases (on 10 possible) better or comparable with results obtained using FRS selection. In the second experiment it has been examined how fan "expanding" parameter *a* influences the solution quality found by evolutionary algorithm. Here only the value of parameter *a* was changed in the range [0; 1], and other algorithm parameters were as in the first experiment. In order to obtain more diverging results, two test functions have been chosen to this experiment: F4, and F10, for which the highest variations of average value have been observed (after 100 generations). The average values of test functions minima obtained after 100 repetitions of evolutionary algorithm (with FS selection, and PFS selection) for different values of *a* parameter are shown in Figure 11. In Figure 11, the bold fonts represent the best obtained results.

Value of a	Test function F10		Test function F4	
	FRS	PFRS	FRS	PFRS
0.00	923.787454	**825.831571**	43.22775589	**41.27742876**
0.05	9.34704799	**6.05478458**	**16.32133735**	17.43088281
0.10	**4.01422242**	4.51124504	7.05015349	**6.75264382**
0.15	**3.42510477**	3.64843891	3.84245710	**3.75743812**
0.20	3.59535739	**3.49873288**	1.87237231	**1.59304921**
0.25	4.47301575	**3.45493020**	1.01711616	**0.96485874**
0.30	**3.93893284**	5.72613486	0.52734645	**0.51963446**
0.35	**4.40365514**	4.93385538	**0.35324468**	0.38431538
0.40	5.47608763	**4.31626349**	**0.16633081**	0.19688269
0.45	7.85601993	**5.40837456**	0.15065552	**0.10133293**
0.50	7.09333457	**6.97917105**	0.10778160	**0.07065058**
0.55	7.67483995	**6.32359892**	**0.05067826**	0.05873887
0.60	**5.88777219**	7.04008018	0.04277946	**0.03591072**
0.65	**8.06077111**	8.54346201	**0.02588992**	0.02944295
0.70	8.31152387	**8.14839001**	**0.02306716**	0.02453017
0.75	10.46178631	**10.32998750**	**0.01562465**	0.01929378
0.80	10.07995614	**6.90307361**	0.01415020	**0.01368041**
0.85	**9.93050283**	10.23908552	0.01122326	**0.01112236**
0.90	11.31274074	**7.20104095**	0.00912869	**0.00857236**
0.95	11.83710844	**11.60276270**	**0.00773982**	0.00774064
1.00	11.16117024	**9.52263277**	0.00632409	**0.00610230**

Figure 11. The value of *a* parameter influence on obtained average values of functions minima

It can be seen from Figure 11, that in the case of function F10 better results in (lower function values) have been obtained using proposed PFRS method compared to FRS method in 14 cases on 21 possible cases. However, for function F4, better results have been obtained using PFRS method compared to FRS method in 13 cases on 21 possible. Also, in the case of function F10, we can determine an approximate range of *ax* parameter values, for which better results have been obtained. This range is between 0.1 and 0.4. In the case of function F4, it has been observed, that the increase of *a* parameter values causes considerable improvement of obtained minimum values, and improves the algorithm convergence (more and more better results are found in the same time period).

4. Conclusions

In this paper the modification of fan roulette selection, named proportional fan roulette selection has been presented. Due to application of formula (3) the disadvantage (occurring in the fan roulette selection) depending on promotion of worst individuals at the cost of average quality individuals has been eliminated. Results obtained using proposed PFRS selection are in all cases better or comparable to results obtained using roulette selection, and elitist selection. The results obtained using proposed proportional fan roulette selection are in more cases (42 obtained results on 62 possible) better or comparable to results obtained using fan roulette selection.

References

1. Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*, WNT, Warsaw, 1999, (in Polish)
2. D. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, WNT, Warsaw, 1998, (in Polish)
3. J. Arabas, *Lectures of evolutionary algorithms*, WNT, Warsaw, 2001, (in Polish)
4. S. Zen, C. T. Zhou Yang, *Comparison of steady state and elitist selection genetic algorithms*, Proceedings of 2004 International Conference on Intelligent Mechatronics and Automation, August 26-31, 2004, pp. 495-499
5. N. Takaaki, K. Takahiko, Y. Keiichiro, *Deterministic Genetic Algorithm*, Papers of Technical Meeting on Industrial Instrumentation and Control, IEE Japan, pp. 33-36, 2003
6. T. Blickle, L. Thiele, *A Comparison of Selection Schemes used in Genetic Algorithms*, Computer Engineering and Communication Networks Lab, Swiss Federal Institute of Technology, TIK Report, No. 11, Edition 2, December 1995

7. A. Slowik, M. Bialko, *Modified Version of Roulette Selection for Evolution Algorithm - The Fan Selection*, Proceedings of Seventh International Conference on Artificial Intelligence and Soft Computing, ICAISC 2004, Lecture Notes in Artificial Intelligence, Volume 3070/2004, pp. 474-479, Springer-Verlag, 2004
8. A. Slowik, *Design and Optimization of Digital Electronic Circuits Using Evolutionary Algorithms*, Ph. D. Thesis, Koszalin University of Technology, Department of Electronics and Computer Science, Koszalin, 2007, (in Polish)
9. Private information from Prof. W. Jedruch from Gdansk University of Technology, Department of Electronics, Telecommunications and Informatics, 2007

Abstract

In the paper modified version of fan roulette selection method named proportional fan roulette selection is presented. This modification depends on increase of survive probability of the best individual at the expense of worse individuals and often gives better results compared to other selections. Test functions chosen from literature are used for determination of quality of proposed method. Results obtained using proportional fan roulette selection are compared with results obtained using roulette selection, elitist selection, and fan roulette selection.

Streszczenie

W artykule przedstawiono proporcjonalną selekcję wachlarzową będącą zmodyfikowaną wersją selekcji wachlarzowej. Wprowadzona modyfikacja polega na zwiększeniu prawdopodobieństwa przeżycia najlepszego osobnika kosztem osobników gorszych, często dając lepsze rezultaty w porównaniu do innych metod selekcji. Do sprawdzenia jakości utworzonej metody zastosowano funkcje testowe wybrane z literatury. Wyniki uzyskane przy użyciu proporcjonalnej selekcji wachlarzowej porównano z wynikami uzyskanymi przy użyciu selekcji ruletkowej, elitarnej oraz wachlarzowej.

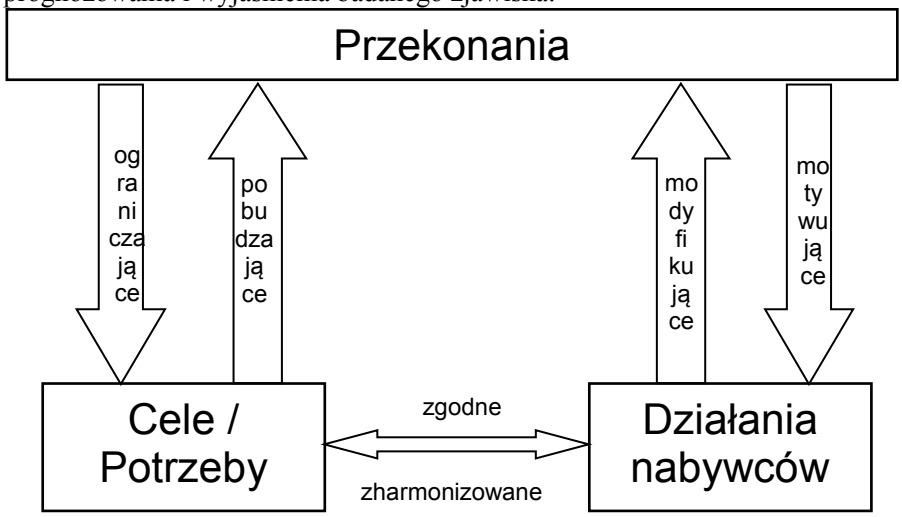
Słowa kluczowe: sztuczna inteligencja, algorytmy ewolucyjne, metody selekcji

Anton Smoliński
Wydział Informatyki
Zachodniopomorski Uniwersytet Technologiczny w Szczecinie
anton.smolinski@zut.edu.pl

Hybrydowy model preferencji konsumenta wykorzystujący selekcję proporcjonalną

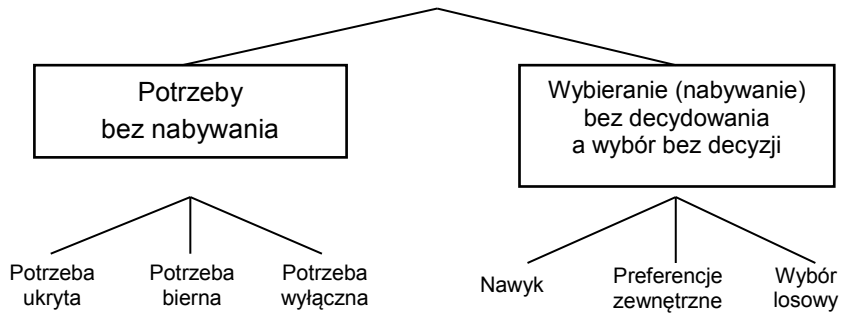
1. Preferencje konsumenta

Jednym z trudniejszych zadań stojących przed naukowcami zajmującymi się modelowaniem zachowań konsumentów jest określenie rodzaju czynników i ich siły mających wpływ na podejmowanie decyzji odnośnie zakupu towarów i usług. Na przestrzeni lat wielu badaczy proponowało własne podejścia do danego tematu, które jednak można sklasyfikować na dwa nurty: Nurt Poznawczy, starający się poprzez analizę przyczynowo-skutkową wyjaśnić zachodzące zjawiska oraz tendencje konsumentów, oraz Nurt Behawioralny, starający się odkryć i zmierzyć czynniki mające wpływ na konsumenta a następnie za ich pomocą aproksymować skalę przyszłej konsumpcji. W swoim artykule Adam Sagan [1] dokonuje dokładnej analizy tych dwóch nurtów konkludując, że nie powinno się ich łączyć przy próbie prognozowania i wyjaśnienia badanego zjawiska.



Rysunek 1. Model zależności wzajemnego oddziaływania celów, potrzeb i przekonań konsumentów [źródło: O'Shaughbessy J.]

W swojej książce Joyce O'Shaughnessy [2] wyprowadza dwa modele zależności potrzeb klienta, na które warto zwrócić uwagę. Pierwszy z nich, ukazany na rysunku 1. przedstawia współzależność potrzeb, przekonań oraz działań nabywców. Model ten wskazuje, że na powstanie decyzji odnośnie zakupu mają wpływ nie tylko potrzeby, lecz także przekonania, które mogą wpływać pozytywnie („wszyscy kupują, więc ja też"), bądź negatywnie („nie wypada"). Istotnym czynnikiem jest także sprzężenie zwrotne pomiędzy decyzjami konsumenta a przekonaniami. Ukazuje to, że im częściej konsument dokonuje danej decyzji, tym bardziej jest do niej przyzwyczajony. Relacja pomiędzy potrzebami a decyzją konsumenta jest oczywista i każdy z modeli go uwzględnia. Interesujące jednak jest sprzężenie zwrotne pomiędzy przekonaniami a potrzebami. Relacje pobudzające i ograniczające jednoznacznie wskazują na fakt, że potrzebami konsumenckimi można sterować, np. za pomocą działań marketingowych, które mogą być skierowane do konkretnego produktu, lub całego segmentu produktów. Przykład takiej zależności można zaobserwować w sektorze rozrywkowym (kluby, lokale, gastronomia) po katastrofie Smoleńskiej w 2010 r. Z powodu ogłoszonej żałoby narodowej wiele lokali nocnych (dyskotek) musiało zawiesić swoją działalność. W tym okresie, ludzie zaczęli odwiedzać bary, które miały inny charakter działalności. Od tego okresu lokale nocne (dyskoteki) odnotowały spadek zainteresowania ze strony odwiedzających. Przykład ten wskazuje, że „przekonania" były w stanie zmienić potrzeby konsumenckie. Wydarzenia tamte miały charakter długotrwały, gdyż nawet 7 lat później wciąż można odczuć ich skutki, dlatego stwierdzenie że potrzeby klientów mogą ulec zmianie pod wpływem różnych czynników jest prawdziwe. W mniejszym zakresie potrzeby kształtują także reklamy, które skłaniają do wyboru określonych towarów i usług niezależnie od potrzeby ich posiadania, a nawet jej braku.



Rysunek 2. Podział potrzeb, które konsument nie zaspokaja poprzez nabywanie, oraz nabywanie, którego konsument dokonuje pomimo braku wyraźnej potrzeby [źródło: O'Shaughbessy J.]

Drugi z modeli wartych wzmianki w dziele Joyce-a O'Shaughbessy-ego [2] przedstawiony na rysunku 2. przedstawia podział potrzeb, których konsument nie zaspokaja poprzez nabywanie, oraz nabywanie, którego konsument dokonuje pomimo braku wyraźnej potrzeby. Model ten wskazuje, że konsument pomimo potrzeby posiadania jakiegoś towaru nie zawsze dokonuje decyzję o jego nabyciu. Wynika to z faktu, że potrzeba może być ukryta, czyli konsument nie zdaje sobie z niej sprawy, jest potrzebą bierną, czyli brak jej zaspokojenia nie wpływa negatywnie na konsumenta, bądź jest potrzebą wyłączoną, gdy konsument świadomie odmawia zaspokojenia danej potrzeby (np. z powodu ograniczeń finansowych). Warte uwagi jest także dokonywanie decyzji o zakupie bez wyraźnej potrzeby do zaspokojenia (pomijając potrzebę nabywania). Takie działanie może być powodowane nawykiem, a więc cykliczną, powtarzalną czynnością wykonywaną nawet, gdy potrzeba zostanie zaspokojona. Kolejnym powodem takich decyzji są preferencje zewnętrzne, pewne czynniki (jak np. presja otoczenia) oddziałują na konsumenta popychając go do zakupu niepotrzebnych mu towarów. Przykładem takiego stanu rzeczy może być obserwowany kilka lat temu trend na telefony komórkowe. W momencie rozprzestrzeniania się tej technologii posiadanie najnowocześniejszej „komórki" było wyznacznikiem statusu społecznego, co skłaniało niektórych do zakupu aparatów, z których nie potrafili korzystać i w rezultacie nie używania tego produktu. Dziś świadomość społeczna oraz edukacja technologiczna znacząco się rozwinęła, tak więc nie można zaobserwować tego zjawiska na przykładzie telefonów komórkowych, których rola ewoluowała do towaru z którego korzysta się na co dzień.

2. Teoria wyboru konsumenta

Ekonomiści Vilfredo Pareto oraz Francis Edgeworth próbując matematycznie opisać sposób zachowania nabywcy, wnieśli ogromny wkład do czegoś, co dziś nazywane jest Teorią wyboru konsumenta [3]. Teoria ta należąca do mikroekonomii opisuje mechanizmy dystrybucji dóbr oraz kształtowania się cen. Teoria ta opisuje zachowania nabywcy za pomocą czterech kategorii:

- Konsument,
- Dochód konsumenta,
- Preferencje konsumenta,
- Użyteczność.

Kategoria konsumenta określa sposób myślenia decydenta podczas nabywania dóbr. Wprowadza ona pewne założenia dotyczące podejmowanych decyzji. Po pierwsze, zakłada się, że konsument dokonuje decyzji suwerennych. Oznacza to, że dana decyzja jest podejmowana wyłącznie w oparciu o własne preferencje w ramach ograniczonych możliwości finansowych. Przytoczony na początku model

współzależność potrzeb, przekonań oraz działań nabywców (przedstawiony na rysunku 1.) wskazuje wyraźnie błąd w danym założeniu, co pokazuje że już u podstaw modelu został niedokładnie sformułowany. Kolejnym założeniem jest racjonalność decyzji, która określa, że konsument wybiera dobra maksymalizując użyteczność nabytych towarów. To założenie także nie wyczerpuje zagadnienia, gdyż jak wskazano na modelu podziału potrzeb (rysunek 2.) istnieją wybory które konsumenci dokonują bez wyraźnej przyczyny, lub z przyczyn innych niż użyteczność. Ostatnim założeniem są nieograniczone potrzeby konsumenta, co oznacza że potrzeby nie można zaspokoić. Próbą naprawienia tego założenia jest stosowanie tzw. Prawa malejącej użyteczności krańcowej (I prawo Gossena)[4], które zakłada, że wraz ze wzrostem konsumpcji danego dobra spada jego użyteczność. Prawo to jest uwzględniane w kategorii Użyteczności, problemem natomiast jest wyznaczenie nachylenia krzywej spadku użyteczności, które za pomocą prostej modyfikacji danego założenia można rozwiązać.

Kategoria preferencji konsumenta zakłada, że każdy decydent posiada pełnię informacji odnośnie nabywanych dóbr, że dobra te można ze sobą porównać pod względem użyteczności (i ustawić w pewnym porządku nierosnącym lub niemalejącym), oraz że preferencje konsumenta cechują się przechodniością (jeżeli decydent woli towar A bardziej, niż towar B, a towar B bardziej niż C, to znaczy, że przy wyborze między A a C, wybierze towar A). Istnieją metody należące do matematyki rozmytej pozwalające na wyznaczenie modelu preferencji konsumenta bez konieczności spełniania tych założeń, jednakże stosowanie ich nie wpływa w znaczący sposób na osiągane wyniku modelowania [5, 6].

Dochód konsumenta jest ograniczeniem jakie posiada konsument na dokonanie zakupu. Przedstawia on wysokość środków pieniężnych pozostających do dyspozycji decydenta, które może spożytkować na zakup potrzebnych mu dóbr.

Ostatnią kategorią jest użyteczność, czyli wprowadzony jednoznaczny system subiektywnej oceny metrycznej danego towaru, dający się wyrazić liczbowo. Dzięki wprowadzeniu tej wielkości model posiada jednoznaczne dane dotyczące oceny poszczególnych dóbr przez konsumenta, co pozwala na określenie jego preferencji uwzględniając pozostałe czynniki.

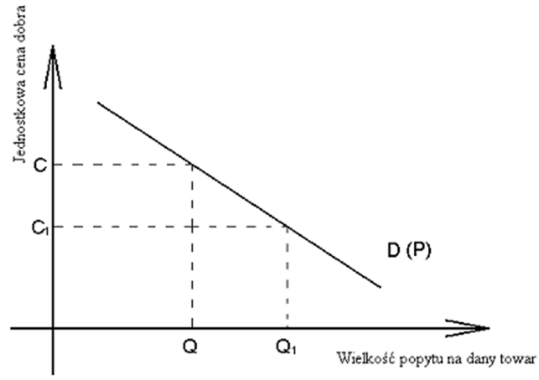
Jak każdy model, jest to jedynie uproszczenie mające opisać pewne zjawisko. Jednym z trudniejszych do opisania zjawisk są te, gdzie czynnik ludzki ma decydujący wpływ, jednak założenia dotyczące konsumenta w danym modelu stanowią pole, w którym model ten można w znaczący sposób ulepszyć (przybliżyć do rzeczywistości).

3. Krzywa popytu

Podstawową doktryną ekonomii jest teoria dotycząca zachowania równowagi pomiędzy popytem i podażą [7]. Konsument, którego zachowania stara się

zamodelować autor tego artykułu kształtuje popyt na towar bądź grupę towarów, dlatego istotną kwestią jest zbadanie zależności, pomiędzy popytem a preferencjami. Opisana w rozdziale 2 Teoria wyboru konsumenta określa czynniki kształtujące popyt, skupiając się na zależności pomiędzy preferencjami konsumenta a dokonywanym wyborem. Jest więc spojrzeniem na dany problem z punktu widzenia kupującego. Patrząc na to samo zjawisko z punktu widzenia towaru, pojawia się pojęcie popytu, w ramach którego istnieje wiele dobrze opisanych matematycznych zależności, o których Teoria wyboru jedynie wspomina marginalnie. Tak więc do konstrukcji poprawnego modelu wymagana jest wiedza na temat zależności dotyczących popytu.

Krzywa popytu, czyli stosunek wielkości popytu do ceny towaru ma nachylenie ujemne, gdyż zgodnie z tą teorią spadek ceny na dane dobro powoduje wzrost popytu (rysunek 3).

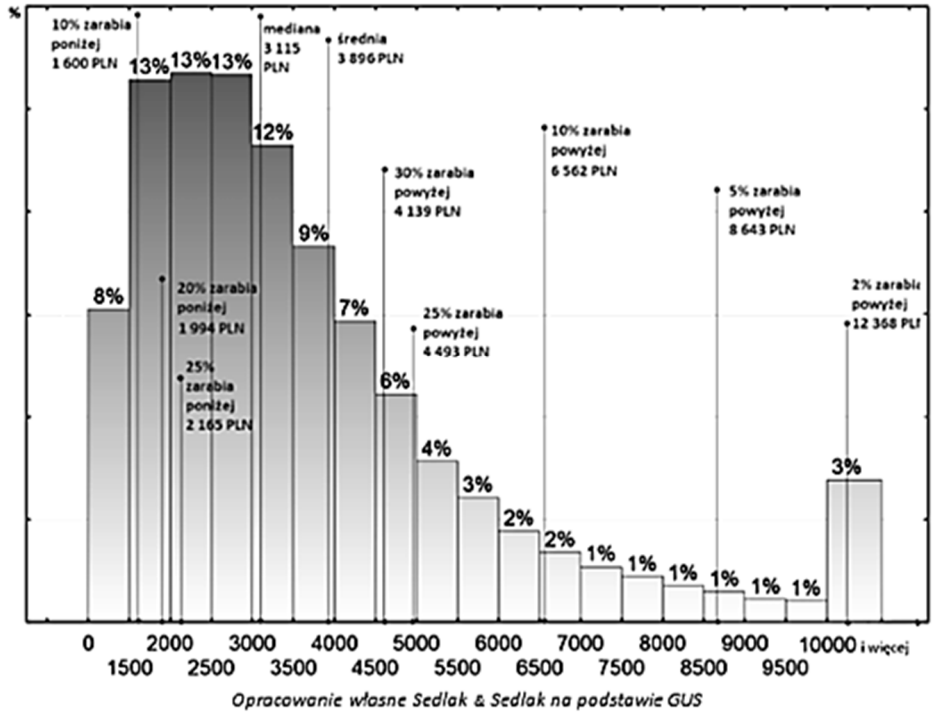


Rysunek 3. Przykładowa krzywa popytu [źródło: http://mfiles.pl/]

Warto jednak zauważyć, że preferencje poszczególnego nabywcy (jego decyzja, czy zakupić dane dobro) uwzględniają cenę jedynie: 1) w ramach użyteczności (jednakże opcjonalnie, w zależności od tego, jak zostanie to zamodelowane), oraz 2) jako ograniczenie do wysokości dochodu, którym dysponuje konsument. Jeżeli w modelu przyjąć, że użyteczność zależy wyłącznie od cech produktu a nie jego ceny, zmiana cen wpłynie na decyzje konsumenta binarnie (jeżeli stać go na zakup, wówczas go dokona, w przeciwnym wypadku nie). Model więc posiada bardzo niską elastyczność popytu względem ceny [7].

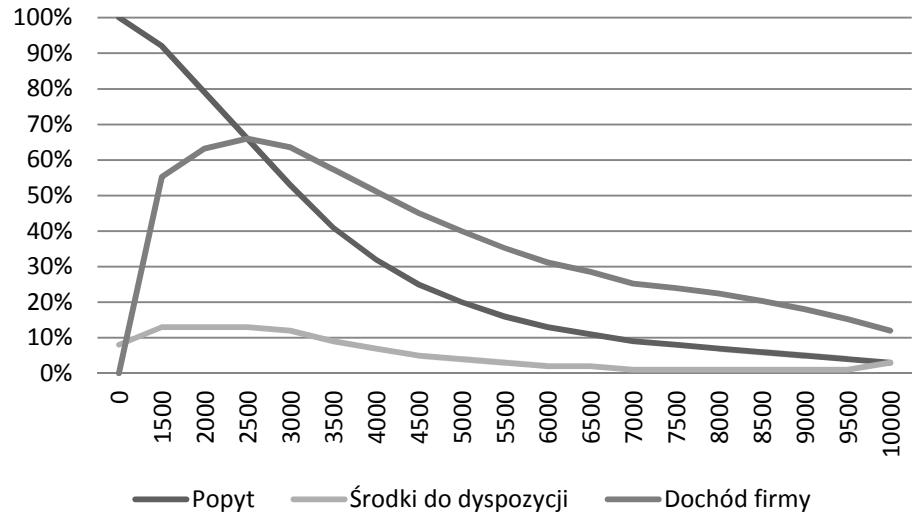
Powstaje więc pytanie w jaki sposób osiągnąć krzywą popytu, która byłaby bardziej zbliżona do rzeczywistej? Pierwszym sposobem jest wprowadzenie modelu użyteczności jako zależność między ceną a cechami danego dobra. Jednakże takie rozwiązanie również sprawi, że krzywa będzie dyskretna (nieciągła), gdyż po przekroczeniu pewnej wartości (dochodu) popyt spadnie do 0. Kolejnym sposobem jest zastosowanie efektu skali, czyli rozpatrywanie problemu z punktu widzenia

wielu konsumentów o różnych dochodach. Wówczas nakładające się na siebie krzywe popytu kształtowane przez pojedynczego konsumenta utworzą wykres ciągły (przy odpowiednio dużej populacji). Istotną obserwacją jest także fakt, że rozkład dochodów w społeczeństwie nie jest równomierny, i przypomina on rozkład normalny (co oznacza, że większość populacji dysponuje podobnym dochodem).



Rysunek 4. Przykładowy rzeczywisty rozkład dochodów (dane za rok 2012) [źródło: http://blog.ucmsgroup.pl/]

Zakładając, że dane z rysunku 4 będą stanowić budżet do dyspozycji, krzywa popytu dla danego rozkładu dochodów będzie kształtować się następująco:



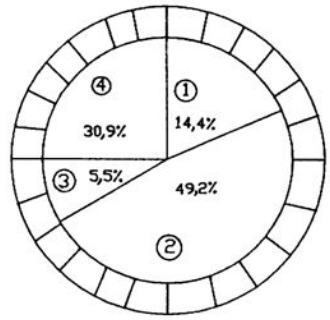
Rysunek 5. Przykładowa rzeczywista krzywa popytu [źródło: opracowanie własne]

Z rysunku 5 można zauważyć więc, że najwyższy dochód ze sprzedaży (rozumiany jako iloczyn popytu i ceny) pojedynczego towaru firma osiągnie, gdy będzie sprzedawać go po cenie 2500 zł, czyli w miejscu, gdzie krzywa popytu osiąga największą ujemną pochodną.

4. Metoda koła ruletki

Teoria wyboru konsumenta zakłada dokonanie zakupu w oparciu o maksymalizację użyteczności dóbr, jednakże jak zostało już wykazane w oparciu o inne teorie, konsumenci nie zawsze dokonują wyborów najbardziej użytecznych. Aby urzeczywistnić dany model można skorzystać ze znanego mechanizmu selekcji metodą koła ruletki, używanego powszechnie w algorytmach genetycznych [8, 9, 10].

Metoda ta polega na utworzeniu tzw. koła ruletki, w którym każde dobro posiada sektor proporcjonalny do wartości jego użyteczności. Oznacza to, że dobra o większej wartości parametru użyteczności będą zajmowały większą powierzchnię koła. Następnie przeprowadza się losowanie mogące symbolizować zakręcenie danym kołem. W ten sposób zostaje wybrane konkretne dobro uwzględniając niedoskonałość ludzkich wyborów (spowodowanych brakiem wiedzy, bądź nieprzemyślanym nabyciem dóbr), a jednocześnie preferując bardziej użyteczne dobra.



Rysunek 6. Zasada działania selekcji metodą ruletki [11]

Metoda koła ruletki w znacznie dokładniejszy sposób obrazuje sposób dokonywania wyboru przez konsumenta i może zostać bezpośrednio wprowadzona jako udoskonalenie Teorii wyboru konsumenta, która zakładała wybór tylko dóbr o najwyższej użyteczności.

5. Model hybrydowy

Zbierając dotychczasowe rozważania można wyprowadzić nowy model preferencji konsumenta bazujący w głównej mierze na kategoriach z teorii wyboru konsumenta. Proponowany w niniejszej pracy model zgodnie z opisanymi w rozdziale 2 argumentami odrzuca założenia kategorii konsument. Elementy niezgodne z założeniami teorii o których wspomniano w 2 rozdziale znajdą swoje odbicie w funkcji użyteczności dobra. Kategoria dochodu konsumenta została zmodyfikowana poprzez uwzględnienie krzywej popytu, o czym traktował rozdział 3. Uzewnętrznia się to poprzez wprowadzenia modelu różnych wysokości dochodów grupy konsumentów, czyli zmiana z dyskretnej wartości (nabycie dobra przy wystarczających środkach bądź nie), na wartości ciągłe (nabycie części dobra odpowiadającej % udziałowi wynikającego z rozkładu dochodów). Kategoria preferencji zostaje zmodyfikowana poprzez wprowadzenie metody selekcji kołem ruletki (opisanej w rozdziale 4). Ostatnią zmianą jest wyprowadzenie funkcji użyteczności dobra, która uwzględniałaby parametry o których pisano w 1 rozdziale, ze szczególnym uwzględnieniem zależności parametrów pomiędzy sobą, oraz innych ograniczeń.

W proponowanej metodzie do wyznaczenia wartości użyteczności dobra wykorzystuje się takie parametry jak:

- preferencje indywidualne – subiektywną ocenę i zależność dóbr między sobą wartość stała dla dobra;

- przyzwyczajenie – wzrost przywiązania do danego dobra w zależności od ilości jakiej się go nabyło;
- podaż – ograniczenie, które zapobiega zakupowi większej ilości dobra, niż jest dostępna na rynku. Ograniczenie to jest maksymalną wartością, jaką może przyjąć funkcja użyteczności;
- nasycenie – zgodnie z I prawem Gossena [4] malejąca wartość wraz ze wzrostem ilości nabytego dobra. Zmiana wartości wyrażona jest jako stosunek nabytego dobra do ograniczenia konsumpcyjnego (ile maksymalnie dobra może dana osoba zużyć);
- moda – należący do czynników zewnętrznych wzrastająca wartość, gdy popyt globalny na dane dobro wzrasta;
- reklama – wpływ reklamy (wydatków na reklamę przedsiębiorcy dystrybuującego dobro) przyczynia się do wzrostu wartości użyteczności;
- jakość dobra – wyrażona ilościowo jakość produktu jest wprost proporcjonalna do wartości użyteczności;
- cena – wrażliwość konsumenta na cenę. Wraz ze wzrostem ceny preferencja do kupna danego towaru maleje.

Wyrażona za pomocą tych parametrów funkcja użyteczności uwzględnia zarówno czynniki wewnętrzne jak i zewnętrzne z różnych płaszczyzn, które mogą wpływać na konsumenta przy wyborze danego produktu. Parametrami tymi można sterować za pomocą odpowiednich im współczynników, które mogą przyjmować wartości zarówno dodatnie, jak i ujemne. Przykładowo jeżeli wartość współczynnika przy parametrze moda będzie miała wartość ujemną, oznacza to, że konsumenci preferują unikalność danego dobra nad jego powszechnością. Wartość odpowiednich współczynników sugeruje także siłę wpływu danego parametru na wybór którego dokonuje konsument.

Dodatkowo należy uwzględnić ograniczenie, że użyteczność nie może być niższa niż zadana, dodatnia wartość ε. Ograniczenie to spełnia dwie funkcje. Po pierwsze ujemna wartość funkcji użyteczności oznaczałaby ujemny popyt, co można zinterpretować jako podaż. Model ten nie uwzględnia możliwości odsprzedaży dóbr dlatego też wartość ta musi być nieujemna. Drugą funkcją ograniczenia ε jest otrzymanie niewielkiej wartości dodatniej, aby dane dobro miało swoją reprezentację na kole ruletki, a więc był możliwy jego wybór (nawet jeżeli jego prawdopodobieństwo jego wystąpienia wynosiłoby ułamki promila). Zabieg ten ma na celu symulację nierozważnych zakupów pod wpływem emocji, lub jakichś czynników losowych.

6. Implementacja modelu

Aby móc wykorzystać opracowany model należy wykonać 4 kroki:

1. Do każdego z wprowadzanych do modelu produktów przygotować zestaw parametrów będących iloczynem siły wpływu parametru w stosunku do innych parametrów oraz oceny danego dobra w kontekście wybranego parametru:

   $Param_{(i,j)} = Siła_{(i)} \cdot Ocena_{(j)}$,

   gdzie:

   i – nazwa parametru,

   j – nazwa dobra.

   W rezultacie otrzymujemy macierz parametrów którą można wykorzystać do dalszych obliczeń. Dla uproszczenia w dalszej części autor będzie posługiwał się oznaczeniem $Param_{(i,j)}$ do określenia pojedynczej wartości parametru.

2. Dokonać wyliczenia preferencji poszczególnych dóbr zgodnie ze wzorem:

   $Preferencja_j = \sum_i^m Param_{(i,j)} \cdot Udział\_rynku_j + \varepsilon$,

   gdzie:

   m – liczba parametrów modelu,

   i – rozpatrywany parametr,

   j – rozpatrywane dobro,

   ε – minimalna wartość preferencji.

   Nie wszystkie parametry modelu są zależne od udziału produktu w rynku, co omówiono w rozdziale 5. W przypadku takich parametrów składnik udziału w rynku można pominąć. Ograniczenie ε zostało opisane w ostatnim akapicie 5 rozdziału. Otrzymana macierz preferencji poszczególnych dóbr pozwala na przeprowadzenie symulacji popytu na zadanej populacji.

3. Wprowadzenie do macierzy Preferencji zmian uwzględniających I prawo Gossena:

   $Preferencja_j = Preferencja_j \cdot \frac{(nasycenie_j - zakupiono_j)}{nasycenie_j}$.

4. Dla każdego osobnika l, w symulowanej populacji należy wykonać kilka kroków celem określenia rodzaju oraz ilości towarów które zamierza nabyć.

   4.1. Należy wylosować liczbę $los_l \in \langle 0, \sum_j^n Preferencja_j \rangle$ zakładając, że losowane liczby będą tworzyły rozkład jednostajny.

   4.2. Należy znaleźć k, spełniające założenie $los < \sum_j^k Preferencja_j$, gdzie k, będzie oznaczać dobro, które zostało wylosowane metodą koła ruletki (rozdział 4) dla danego osobnika.

4.3. Wprowadzenie zaburzenia θ , które odpowiada za irracjonalne decyzje klienta. Zaburzenie to polega losowej zmianie wybranego dobra bez uwzględnienia preferencji dla określonego procenta populacji.

4.4. Na podstawie krzywej rozkładu dochodu wyliczyć ilość towaru nabywanego przez danego osobnika:

$$Towarów_{(l,k)} = siła\_nabywcza_{max} \cdot \int_{\text{siła\_nabywcza}_{min}}^{siła\_nabywcza_{max}} F(l)dl,$$

gdzie:

l – oznaczenie osobnika,

k – wybrany towar,

siła_nabywcza – maksymalna i minimalna wartość dób, które konsument może nabyć,

F(l) – funkcja rozkładu dochodów.

4.5. Wprowadzenie ograniczeń na $Towarów_l$ dotyczących podaży, oraz nasycenia rynku,

$$Towarów_{(l,k)} = \begin{cases} podaż_k, gdy\ Towarów_{(l,k)} > podaż_k \\ nasycenie_k, gdy\ Towarów_{(l,k)} > nasycenie_k \\ Towarów_{(l,k)}, dla\ pozostałych\ przypadków \end{cases}$$

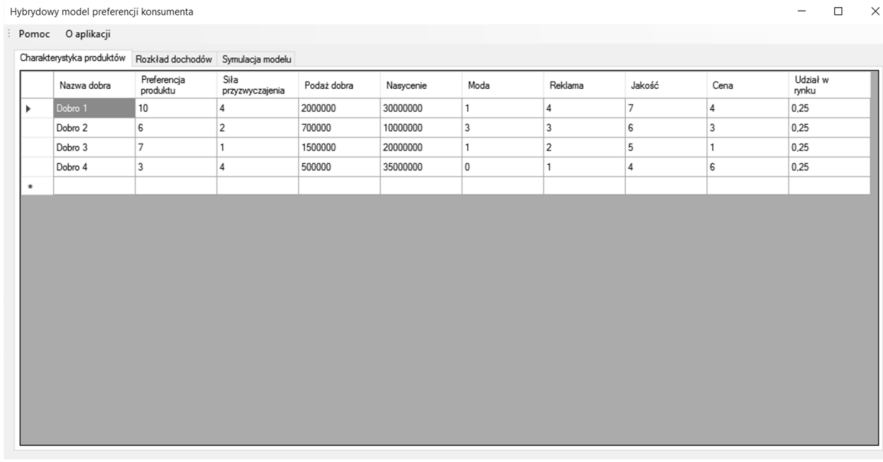
5. Macierz $zakupiono_j = \sum_l^p Towarów_{(l,j)}$ odzwierciedla udział w rynku poszczególnych towarów i wraz z macierzą $Preferencja_j$ stanowi wynik symulacji modelowania za pomocą stworzonego Hybrydowego modelu preferencji konsumenta.

7. Testowanie modelu

Celem przetestowania opracowanego modelu stworzono prostą symulację, pozwalającą na wprowadzenie poszczególnych wartości parametrów modelu, a następnie wyliczenia koła ruletki, które symbolizuje procentowy udział w rynku poszczególnych dóbr. Kolejne iteracje pozwalają na sprawdzenie jak zachowuje się modelowany rynek pod wpływem decyzji podejmowanych przez konsumentów.

Program pozwalający na własne symulacje i przetestowanie opracowanego modelu można znaleźć na witrynie: http://antons.pl/ w dziale dokumenty pod nazwą HybrydowyModelPreferencji.zip. Program został wykonany w technologii Microsoft .NET z wykorzystaniem języka C# oraz WindowsFormApplication.

W celu przetestowania modelu wprowadzono następujące dobra i parametry:

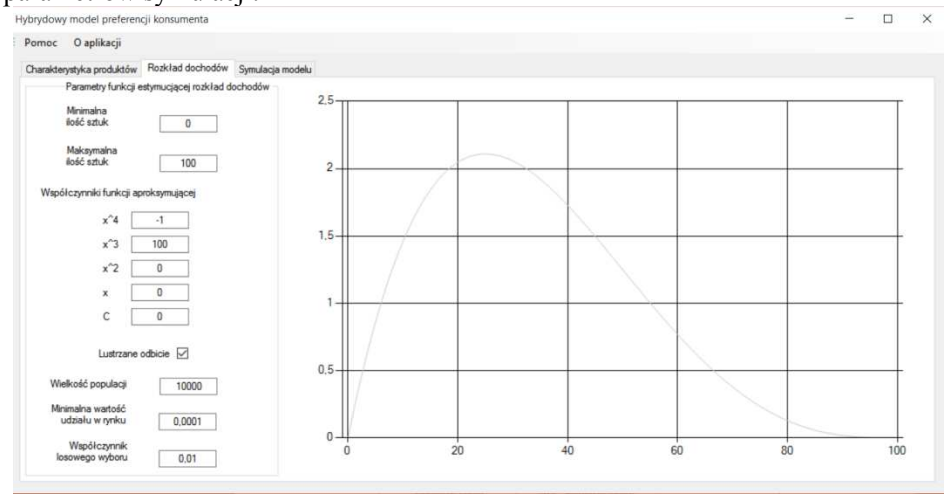


	Nazwa dobra	Preferencja produktu	Siła przyzwyczajenia	Podaż dobra	Nasycenie	Moda	Reklama	Jakość	Cena	Udział w rynku
▶	Dobro 1	10	4	2000000	30000000	1	4	7	4	0,25
	Dobro 2	6	2	700000	10000000	3	3	6	3	0,25
	Dobro 3	7	1	1500000	20000000	1	2	5	1	0,25
	Dobro 4	3	4	500000	35000000	0	1	4	6	0,25
*										

Rysunek 7. Parametry testowanych dóbr [źródło: opracowanie własne]

Parametry te pozwalają zaobserwować wpływ prawa Gossena na nabywców a także dostrzec różnice pomiędzy preferencjami a udziałem w rynku poszczególnych dóbr.

Kolejnym etapem w aplikacji jest ustalenie krzywej rozkładu dochodów, oraz parametrów symulacji:



Rysunek 8. Funkcja rozkładu dochodów [źródło: opracowanie własne]

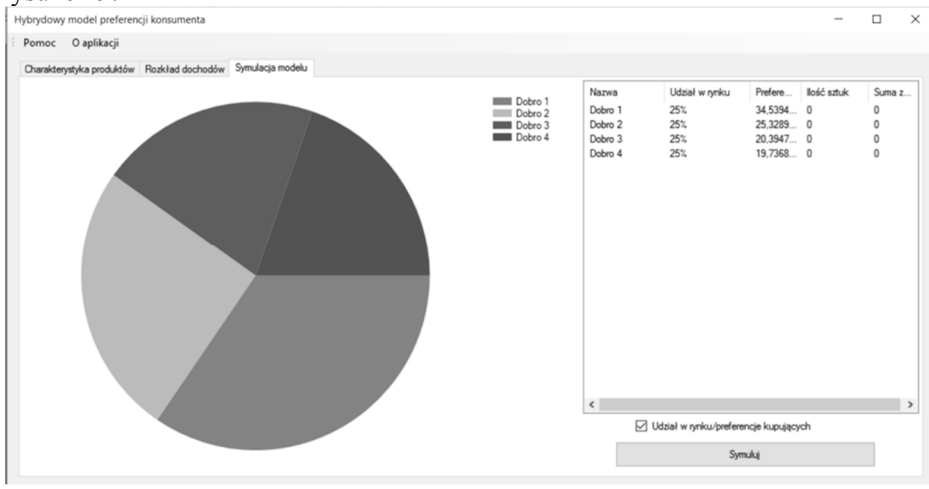
Funkcja rozkładu dochodów estymowana jest wielomianem 4 stopnia na przedziale $\langle 0{,}100 \rangle$, co symbolizuje minimalną i maksymalną liczbę dóbr jakie może zakupić konsument (siła_nabywcza). Na osi rzędnych oznaczony jest procent

populacji którzy posiadają daną siłę nabywczą. Program skaluje oś rzędnych by prawdziwe było wyrażenie:

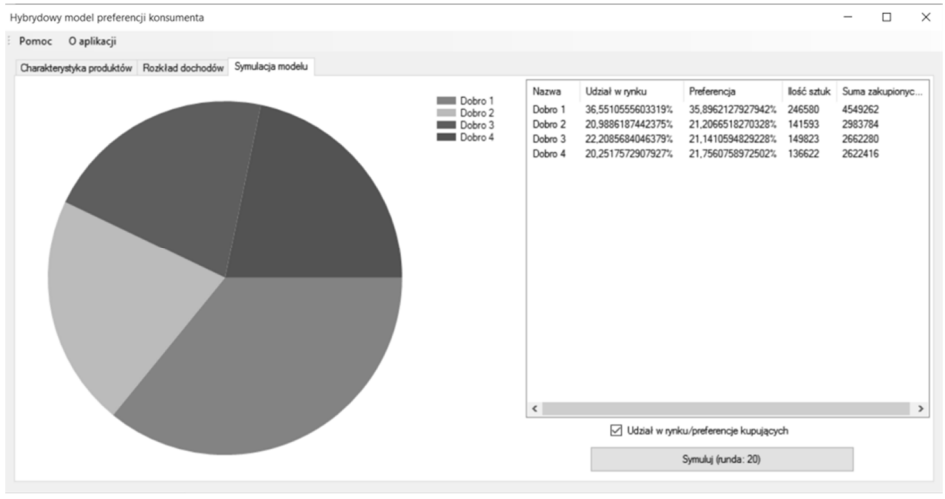
$$\int_0^{100} F(x)dx = 100,$$

Co pozwala interpretować wartości funkcji F(x) jako wartości procentowe.

Początkowe ustawienia preferencji konsumenckich dla przykładu prezentuje rysunek 9:



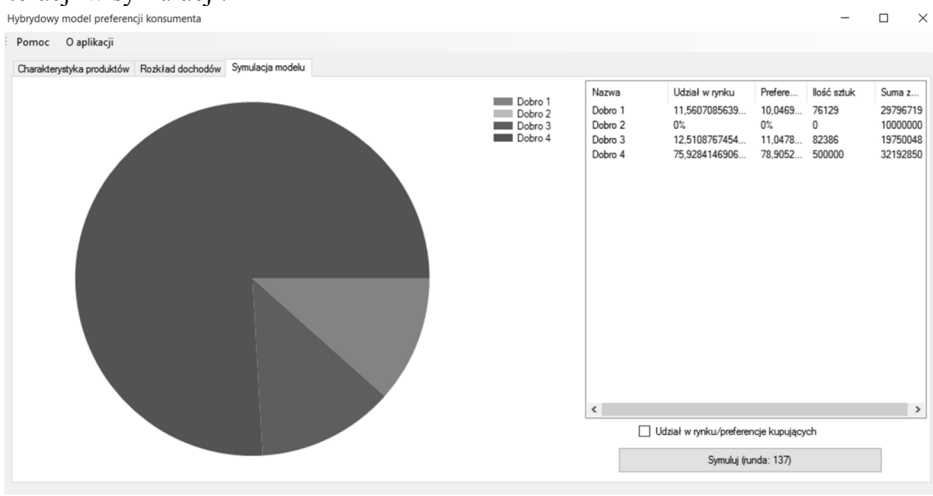
Rysunek 9. Wartość preferencji wyliczonych przez model dla warunków początkowych [źródło: opracowanie własne]



Rysunek 10. Wartość preferencji wyliczonych przez model po 20 iteracjach symulacji [źródło: opracowanie własne]

Kolejne iteracje symulacji pozwalają prześledzić zmianę preferencji oraz udziału w rynku poszczególnych produktów. Zmiana parametru wyświetlania ukazuje też różnice pomiędzy wartościami preferencji a udziałami w rynku danego dobra. Wartości te różnią się między sobą o ok. 0,5%.

Dobrane parametry dla dobra nr 2 zostały dobrane tak, że jest ono atrakcyjne dla konsumentów, lecz rynek szybko się nim nasyca, w rezultacie dobro nr 2 sukcesywnie traci wartość preferencji oraz udział w rynku aż do osiągnięcia granicy nasycenia, przy której udział w rynku wynosi 0%. Sytuacja ta ma miejsce ok. 130 iteracji w symulacji.



Rysunek 11. Wartości modelu po 137 iteracjach symulacji. Sprzedaż dobra 2 osiągnęła wartość nasycenia i zniknęło z rynku. [źródło: opracowanie własne]

8. Podsumowanie

W niniejszym artykule przedstawiono prace nad modyfikacją Teorii Wyboru Konsumenta [3] i na ich podstawie opracowano hybrydowy model preferencji konsumenta wykorzystujący zagadnienia algorytmów genetycznych takie jak selekcja proporcjonalna, czy symulacja osobników w określonej populacji.

Rozwiązanie to wprowadza wiele elementów, które nie były wcześniej uwzględniane przy predykcji decyzji konsumentów łącząc je w spójną całość, której działanie można prześledzić w specjalnej aplikacji którą stworzono na potrzeby tego projektu.

Jak każdy model również tutaj można znaleźć wiele uproszczeń, z których najbardziej znaczące mogą być ustawianie parametrów preferencji dla produktów a nie dla poszczególnych osobników, czy stawianie różnych dóbr na równi bez podziału na kategorie produktów.

Dalsze prace w tej tematyce mogą przynieść rozwiązanie tych problemów, a także umożliwią badanie zależności odwrotnej, czy na podstawie udziałów jakie produkt ma na rynku można określić parametry preferencji konsumentów.

Model ten także może stanowić serce gier decyzyjnych i menadżerskich, w których gracz musi podejmować działania dotyczące dystrybucji posiadanych zasobów celem uzyskania jak największego udziału w rynku, wyliczanego za pomocą niniejszego modelu.

Bibliografia

1. Sagan A. *Modele strukturalne w analizie zachowań konsumenta – ewolucja podejść*, Konsumpcja i Rozwój Nr 1/2011, Instytut Badań Rynku, Konsumpcji i Koniunktur, Warszawa 2011
2. O'Shaughnessy J. *Dlaczego klienci kupują*, PWN, Warszawa 1994
3. Begg D., Fischer S., Dornbusch R. *Ekonomia. Tom 1*, PWE, Warszawa 1995
4. Moroz E., *Podstawy mikroekonomii*, PWE Warszawa 2005
5. Stankiewicz M., *Modelowanie profili klientów w informatycznym systemie wspomagania decyzji*, rozprawa doktorska, Wydział Informatyki ZUT, Szczecin 2013
6. Gawrońska D., *Model rozmyty wyboru samochodu w najwyższym stopniu spełniającego preferencje klienta*, Organizacja i zarządzanie z. 64, Nr kol. 1894, Wrocław 2013
7. Beksiak J., *Ekonomia*, Wydawnictwo Naukowe PWN, Warszawa 2001
8. Turing A. M., *Computing Machinery and Intelligence. „Mind"*, Tom 59, nr 236, X/1950
9. Michalewicz Z., *Algorytmy genetyczne+struktury danych=programy ewolucyjne,* Wydawnictwa Naukowo-Techniczne, 1996
10. Goldberg D. E., *Algorytmy genetyczne i ich zastosowania*, WNT, Warszawa 1998
11. Studniarski M., *Wykłady z algorytmów genetycznych Część 1: Podstawowe informacje o algorytmach genetycznych*, Wydział Matematyki i Informatyki Uniwersytetu Łódzkiego, http://math.uni.lodz.pl/~marstud/AGwyklad1-2.pdf [dostęp: 26.12.2015]

Streszczenie

W niniejszym artykule zaprezentowano model preferencji konsumenta opierający się na popularnej teorii wyboru konsumenta, jednak z kilkoma znaczącymi modyfikacjami. Głównym celem skonstruowania tego modelu jest dokładna symulacja popytu na różnorodne towary. Opracowany model bazuje na takich zmiennych jak: trendy konsumenckie (moda), nawyki, popyt, reklamy czy nasycenie rynku. Dodatkowo został wprowadzony czynnik symulujące nieprzemyślane, losowe decyzje konsumentów oparty na selekcji proporcjonalnej znanej z algorytmów genetycznych.

Przeprowadzone eksperymenty symulacyjne pozwoliły zweryfikować, że opracowany model zachowuje się poprawnie, imitując zachowanie licznej populacji konsumenckiej. Model ten może mieć wiele różnych zastosowań, począwszy od badań nad predykcją sprzedaży, skończywszy na tworzeniu gier decyzyjnych umożliwiających trening umiejętności kadry zarządzającej.

Słowa kluczowe: Preferencje konsumenta, teoria wyboru konsumenta, model podejmowania decyzji

Abstract

This paper presented model of consumer preferences based on consumer choice theorem, but in several significant modifications. The main goal of developed model is the ability to accurate simulate for demand for various goods. Formulated model takes into account several factors like: consumer trends, habits, supply, advertisement, or market saturation. They were also modeled reckless, random buying decisions by proportional selection drawn from genetic algorithms.

Conducted experiments model verification show, that model correctly imitates behavior of multiply client population. Designed model have many applications, not only research customer preferences or sale predictions, but also creating decision making games for managers training.

Keywords: Consumer preferences, Consumer choice theorem, decision making model

Anton Smoliński
Wydział Informatyki
Zachodniopomorski Uniwersytet Technologiczny w Szczecinie
anton.smolinski@zut.edu.pl

Samoadaptacyjna Optymalizacja Genetyczna

1. Algorytmy genetyczne

Algorytmy genetyczne są jedną z najczęściej stosowanych metod heurystycznych do rozwiązywania różnorakich problemów. Ich popularność wywodzi się przede wszystkim z uniwersalności zastosowań, oraz braku ograniczeń, które często występują w innych metodach. Stosowanie algorytmów genetycznych nie wymaga zgłębiania się w rozwiązywany problem, gdyż wystarczy znaleźć odpowiedź na pytanie jak porównać dwa rozwiązania i wybrać „lepsze". Dzięki swoim właściwościom algorytmy genetyczne doskonale sprawdzają się w rozwiązywaniu problemów rzeczywistych, oraz takich, w których przestrzeń rozwiązań jest na tyle duża, że inne metody stają się zbyt czasochłonne [5, 12].

Algorytmy genetyczne nazują na rozwiązaniu naśladującym pewne właściwości spotykane w naturze, takie jak dziedziczenie, czy wymiana genów, oraz dobór naturalny. Można wskazać, że działają one na zasadzie symulacji, w której tworzony jest zbiór pewnych rozwiązań problemu (zwany populacją). Pojedyncze rozwiązanie kodowane jest w tzw. chromosom zawierający informację na temat danego rozwiązania. Kodowanie ma szczególne znaczenie dla problemów rzeczywistych, gdyż prócz przedstawienia samego rozwiązania umożliwia także wprowadzenie pewnych ograniczeń na przestrzeń poszukiwań, dzięki czemu nie trzeba martwić się w kolejnych etapach algorytmu, czy generowane nowe rozwiązania należą do danej przestrzeni. Zagadnienie doboru odpowiedniego kodowania ma znaczący wpływ również na szybkość i dokładność działania algorytmów genetycznych i zostało opisane i przebadane w wielu pracach [14, 15].

W każdym kroku (iteracji) algorytmu genetycznego populacja zostaje przetworzona za pomocą kilku operatorów genetycznych takich jak selekcja, krzyżowanie i mutacja. Pozwalają one na wybranie oraz utworzenie nowych rozwiązań problemu, które w rezultacie dążą do lepszego dopasowania całej populacji. Przez wiele lat, odkąd opracowano ideę algorytmów genetycznych (klasyczną formę tych algorytmów przedstawił John Holland w 1970 r.), opracowywano wiele różnych operatorów, których zadaniem było usprawnienie przeszukiwania przestrzeni rozwiązań dla konkretnych problemów. Powstało wiele

prac oraz badań nad wpływem dobranych operatorów oraz ich parametrów na to, jak szybko i jak dokładnie algorytmy genetyczne wyszukują rozwiązanie quasi optymalne [1, 3, 4, 7, 8, 13].

Jedną z poważniejszych wad algorytmów genetycznych jest wspomniane wyszukanie rozwiązań quasi optymalnych. Oznacza to, że każdy przebieg (iteracja) algorytmu ulepsza pewne rozwiązania znajdujące się w populacji, dzięki czemu populacja jako ogół staje się coraz bardziej dopasowana do problemu, jednakże znalezienie rozwiązania optymalnego jest bardzo rzadkie. Wynika to z faktu w pewnej mierz losowego przeszukiwania przestrzeni rozwiązań i kłopotem z dokładnością reprezentacji rozwiązania. Aby rozwiązać ten problem często tworzy się hybrydy metod genetycznych i klasycznych, gdzie po znalezieniu pewnego rozwiązania quasi-optymalnego, metody klasyczne rozpoczynają dodatkowe dostrajanie celem polepszenia wyniku zwróconego przez algorytm genetyczny (oczywiście pod warunkiem, że znana jest metoda na dokładne rozwiązanie rozpatrywanego problemu). Nie ulega jednak wątpliwości, że to właśnie algorytmy genetyczne w rozsądnym czasie zwracają dobre wyniki, co w wielu przypadkach (szczególnie problemach rzeczywistych) jest wystarczające.

Drugim problemem algorytmów genetycznych jest złożoność obliczeniowa. Jako iż algorytmy te nie są zależne od rozpatrywanego problemu nie sposób określić ile iteracji należy wykonać aby osiągnąć akceptowalne rozwiązanie. Istnieje kilka metod zatrzymania algorytmu, z których najczęściej rozpowszechniona i najprostsza jest określenie z góry ilości iteracji algorytmu. Inne sposoby badają zbieżność populacji bądź najlepszego wyniku w populacji (np. jeżeli nie poprawił się przez zadaną liczbę iteracji, bądź kolejne wyniki są od siebie nie lepsze niż zadana dokładność). Jednak wraz z rozwojem wieloprocesorowych maszyn a także prostym dostępem do systemów rozproszonych problem złożoności przestaje być krytycznym. Struktura algorytmów genetycznych zakłada równoległe wykonywanie większości operacji, tak więc wykonywanie ich sekwencyjnie jest pewnym ograniczeniem ich możliwości [2, 5, 9, 15].

Uogólniając pewne właściwości algorytmów genetycznych powstała nowa klasa rozwiązywania problemów – obliczenia ewolucyjne. W odróżnieniu od swojego pierwowzoru obliczenia ewolucyjnie nie posiadają ściśle określonej struktury operatorów genetycznych ani reprezentacji. Bazują one w luźny sposób na operatorach genetycznych wprowadzając pewne urozmaicenia jak subpopulacje, zmienność operatorów bądź ich parametrów, czy inne zmiany. Natomiast jak w przypadku zwykłych algorytmów genetycznych celem tym zmian jest jedna (lub dwie) z poniższych zmian:

- Zwiększenie dokładności otrzymywanych obliczeń,
- Zwiększenie szybkości zwracania danych obliczeń.

2. Adaptacje w algorytmach genetycznych

Przy rozpatrywaniu algorytmów genetycznych oraz dobierania odpowiednich operatorów i parametrów do rozpatrywanego problemu ważne jest zrozumienie dwóch pojęć: naporu selekcyjnego, oraz różnorodności.

Napór selekcyjny można rozumieć jako szybkość z jaką populacja zbiega do pewnego rozwiązania quasi-optymalnego. Różnorodność określa natomiast różnice w pojedynczej populacji. Oczywistym faktem jest, że te dwa pojęcia posiadają przeciwstawne zwroty. Otóż silny napór selekcyjny powoduje, że populacja staje się jednorodna, natomiast duża różnorodność populacji spowalnia napór selekcyjny. Z jednej strony używając algorytmów genetycznych oczekuje się znalezienia dobrych wyników w krótkim czasie, jednak zbyt szybki napór selekcyjny i mała różnorodność pogarsza dokładność zwracanych rozwiązań, a może wręcz prowadzić do wpadnięcia do pułapki optimów lokalnych [9, 10, 11].

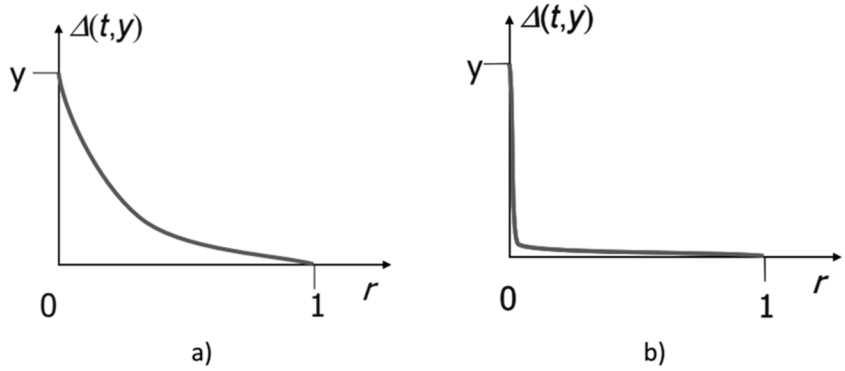
Wielkościami tych zjawisk steruje się za pomocą dobierania parametrów poszczególnych operatorów. Aby odpowiednio dobrać parametry często przeprowadza się kilka(-naście, -dziesiąt) przebiegów algorytmu z różnymi parametrami, gdyż nie odkryto jeszcze sposobu na dobór odpowiednich parametrów do konkretnego problemu. Często pojawia się więc paradoks, gdyż by wykonać optymalizację za pomocą algorytmów genetycznych należy najpierw zoptymalizować same algorytmy, a dokładnie – parametry poszczególnych operatorów.

Sposobem, który w dużej mierze rozwiązuje problem doboru odpowiednich parametrów jest stosowanie metod adaptacyjnych. Dzięki nim w trakcie swego działania programy ewolucyjne same dostrajają potrzebne im parametry. Przykładowo, różnorodność populacji jest ważniejszy na samym początku pracy algorytmów, natomiast im dokładniejsze wyniki są otrzymywane, zaczyna się w większym stopniu liczyć napór selekcyjny i dostrajanie lokalne [1, 6, 9, 10].

Aby osiągnąć adaptację stosuje się kilka różnych metod. Najpopularniejszą jest wprowadzanie zmiennych parametrów, np. mutacji zależnych od numeru iteracji algorytmu. Najpopularniejszą z takich metod jest mutacja nierównomierna, która zakłada, że w początkowych fazach algorytmu mutacji przede wszystkim powinny podlegać najbardziej znaczące bity w bitowej reprezentacji chromosomu, dzięki czemu zapewnia się dużą zmienność w rozwiązaniach, a następnie w trakcie jego działania zmieniać mniej znaczące bity – dzięki czemu kolejne rozwiązania nie różnią się zbytnio od siebie. Przykład zależności zmian można prześledzić na rysunku 1. Wykres a) zawiera zależność funkcji $\Delta(t,y)$ – czyli wartości zmiany chromosomu w wyniku mutacji, od liczby losowej (r) i maksymalnego przedziału zmian (y) [1, 9].

Innym sposobem na adaptacje algorytmów ewolucyjnych (adaptacja powoduje, że algorytmy należą już do wyższej klasy – ewolucyjnych a nie tylko genetycznych), jest stosowanie zmiennych wielkości populacji. Uzależnienie

wielkości populacji od kolejnych iteracji ma podobny wynik jak uzależnienie zmian w operatorze mutacji. Przy dużych populacjach naturalnym jest fakt, że osobniki są zróżnicowane (choćby za sprawą wielkości danej populacji). Jednocześnie powoduje to zwiększenie ilości wykonywanych operacji, gdyż każdy z osobników musi być osobno oceniony, a następnie należy utworzyć ponowną, dużą populację. Małe populacje wymagają znacznie mniej operacji, lecz wadą w tym przypadku jest niska różnorodność osobników (nawet, jeżeli mocno się od siebie różnią, reprezentują niewielką grupę przestrzeni rozwiązań) [9, 10].



Rysunek 1. Zależność zmian wprowadzanych przez mutację nierównomierną w a) początkowej fazie algorytmu ewolucyjnego, b) końcowej fazie algorytmu ewolucyjnego

Niektórzy badacze wykorzystując także metody hybrydowe do tworzenia algorytmów ewolucyjnych nazywając je adaptacją. Najpopularniejsze są w tym przypadku hybrydy pomiędzy algorytmami genetycznymi a metodami klasycznymi (dla danego problemu), które mają za zadanie zwiększyć dokładność rozwiązania wskazanego przez AG. Innym sposobem jest stosowanie metod np. logiki rozmytej jako metaalgorytm doboru parametrów AG [6].

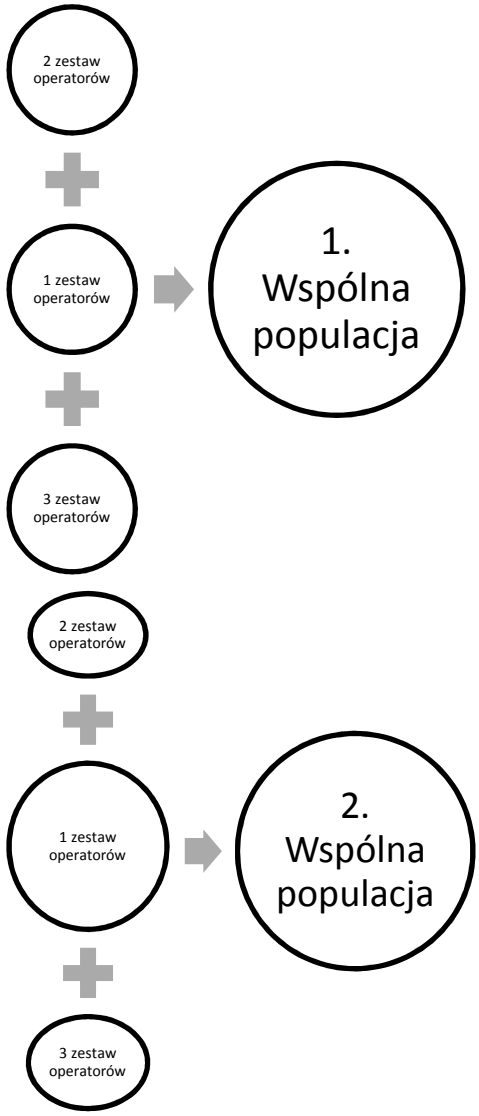
Przy rozpatrywaniu znaczenia adaptacji warto również wspomnieć tzw. No Free Lunch Theorem opracowaną przez Wolpert, Macready [9]. Teoria ta opisuje matematyczne podstawy znanego powszechnie faktu, że każda metoda specjalizacyjna będzie lepsza od metody uniwersalnej w zadanym problemie. Dodatkowo można to rozumieć, że odpowiednio dobrane parametry algorytmu genetycznego tworzą rozwiązanie bardziej wyspecjalizowane, od algorytmu w którym te parametry nie są dobierane – mimo że obydwa powinny po pewnym czasie dać dobre wyniki. Jednak problemy z metodami specjalizowanymi wynikają przy próbie ich zastosowania do innych problemów, niż te dla których zostały dostrojone. W tym przypadku radzą sobie znacznie gorzej od metod uniwersalnych. Oczywiście szczególnym przypadkiem są algorytmy genetyczne, które z założenia są metodami uniwersalnymi, jednak zmieniając parametry można je dostosować. Zmiana

parametrów będzie miała wpływ na szybkość osiąganych wyników, jednak nie tak wielką, jak zmiana całej metody wyszukiwania odpowiedniego rozwiązania. Adaptacje wprowadzane do metod genetycznych dodatkowo pozwalają na osiągnięcie „lepszych" wyników gdy nie znamy dokładnego rozwiązania dla danego problemu, czy klasy problemów.

3. Idea metody SOG

Opracowane przez autora podejście Samoadaptacyjnej Optymalizacji Genetycznej bazuje na kilku założeniach i modyfikacjach przetestowanych przez innych badaczy. Metoda ta ma na celu uzyskania efektu synergii w wyniku połączenia różnych modyfikacji algorytmów genetycznych, a także usprawnienie adaptacji celem samoistnego dostrajania się algorytmu do istoty rozwiązywanego problemu. W tym celu proponuje się wprowadzenie wielopoziomowej adaptacji: Metod, Operatorów oraz Parametrów.

W odróżnieniu do powszechnie stosowanych metod adaptacji, metoda SOG przenosi adaptację na poziom metaalgorytmu. Oznacza to, że poszczególne metody rozwiązywania zadań, bądź operatory także stanowią przedmiot adaptacji i mogą być wzmacniane, bądź wygłuszane w zależności od rozpatrywanego zadania. Aby tego dokonać metoda zakłada wprowadzenia obliczeń ewolucyjnych nie tylko na poziomie rozwiązań, lecz na poziomie poszczególnych algorytmów. Schemat danej adaptacji został przedstawiony na rysunku 2. W ramach danej metody uruchamiane są różne metody oraz operatory genetyczne pracujące na osobnych subpopulacjach. W kroku selekcji oceniane są nie tylko pojedyncze rozwiązania, lecz całe populacje zwracane przez zestawy operatorów. Populacje o najlepszym dopasowaniu powodują że utworzone je operatory otrzymują priorytet w kolejnych obliczeniach, utożsamiany z wielkością subpopulacji którą przetwarza, natomiast pozostałe zestawy operatorów otrzymują zubożenie swojej populacji. W ten sposób schemat doboru naturalnego przekłada się nie tylko na samych rozwiązaniach, lecz także na sposób wyszukiwania nowych rozwiązań. Biorąc pod uwagę, że różne optymalne dla danego problemu zestawy operatorów i parametrów mogą być różne, po każdej selekcji subpopulacje są łączone, a następnie znów dzielone w losowy sposób.



Rysunek 2. Schemat działania metaalgorytmu adaptacyjnego SOG

Prócz wprowadzenia konkurowania pomiędzy metodami, operatorami oraz ich parametrami, metoda SOG wprowadza także najpopularniejsze usprawnienia algorytmów genetycznych, które przyczyniają się do zwiększenia ich dokładności a także szybkości obliczeń. W tym celu korzysta się z dobrze przebadanych poprawek do klasycznego algorytmu genetycznego i spuścizny badaczy, aby

uzyskać nową wartość dodaną wynikającą z ich połączenia. Proponowane modyfikacje można zauważyć na rysunku 3. Na rysunku tym widnieją dwa kolory przedstawiające modyfikacje wspierające szybkość działania programów ewolucyjnych, a także modyfikacje wspierające dokładność znajdowanych rozwiązań.

Modyfikacje wpływające na dokładność znajdowanych rozwiązań, a więc Kodowanie zmiennopozycyjne, Mutacja nierównomierna, oraz adaptacyjna elitarność, to jedne z częściej stosowanych modyfikacji w algorytmach genetycznych. Kodowanie zmiennopozycyjne pozwala na zwiększenie przestrzeni poszukiwań, spowodowane zmianą zapisu poszczególnych liczb. Dodatkowo zmiana ta wprowadza pewną właściwość liczb rzeczywistych, której nie sposób było osiągnąć za pomocą kodowania binarnego – otóż podobne rozwiązania, mają podobną reprezentację, czyli niewiele się od siebie różnią. W przy kodowaniu binarnym, aby osiągnąć dany efekt należało stosować zapis binarny w Kodzie Graya.

Rysunek 3. Modyfikacje AG proponowane w metodzie SOG

Mutacja nierównomierna, wspominana już w niniejszym artykule wprowadza kolejny poziom adaptacyjności na poziomie poszczególnych parametrów, zmiennych w trakcie przeprowadzania obliczeń. Podobnie adaptacyjna elitarność oznacza wprowadzenie elitarności zależnej od ilości iteracji wykonanych przez algorytm. Elitarność jest jednym z najpowszechniej stosowanych środków do zwiększenia naporu selekcyjnego. Zakłada ona porównanie osobników potomnych z osobnikami rodzicielskimi i wybranie „lepszych", bądź w niektórych przypadkach jawne przeniesienie najlepszych osobników z populacji rodzicielskiej do populacji potomnej bez wykonywania na nich innych operacji genetycznych [5, 9, 12, 13].

Przyspieszenie obliczeń w metodzie SOG realizowane jest głównie za pomocą ich zrównoleglania. W tym przypadku bezwzględny czas obliczeń będzie dłuższy, niż obliczenia sekwencyjne (gdyż należy doliczyć czas synchronizacji), jednakowo czas względny, który będzie odczuwalny przez użytkownika jest znacznie krótszy z powodu asynchronicznego przeprowadzania obliczeń i wykorzystywania większej ilości rdzeni komputera.

Dodatkowym usprawnieniem prędkości działania jest propozycja wysunięta przez Goldberga, polegająca na przetwarzaniu tzw. Nieporządnych algorytmów genetycznych (messy genetic algorithms). Reprezentacja taka umożliwia przetwarzanie rozwiązań bez kompletu genów określających dane rozwiązanie. W rezultacie chromosomy mogą posiadać różną długość, a co za tym idzie zużycie pamięci do przechowywania populacji będzie mniejsze. W przypadku tej reprezentacji istotnym czynnikiem jest określenie jak interpretować brakujące geny chromosomu. Nadmiarowość genów jest natomiast redukowana poprzez wyciągnięcie średniej z danych wartości. Najczęściej stosowanymi operatorami przy tej reprezentacji są operatory przecinania i sklejania [5, 9, 11].

Zdaniem autora metody modyfikacje te powinny zapewnić doskonały kompromis pomiędzy uniwersalnością a specjalnością metody do rozwiązywania problemów, szczególnie gdy rozwiązywane są problemy rzeczywiste, gdyż właśnie one stanowią główny cel stosowania opracowanej metody. Podstawy teoretyczne poszczególnych modyfikacji, różne badania, oraz powszechność stosowania pozwalają mieć nadzieję graniczącą z pewnością że metoda ta przyniesie oczekiwane po niej wyniki.

Niestety na chwilę obecną metoda ta dopiero została opracowana koncepcyjnie na podstawie dogłębnych badań literaturowych i badań nad poszczególnymi elementami algorytmów genetycznych. Przetestowanie działania metody i rzeczywiste wyniki, jakie można za jej pomocą osiągnąć zostaną przedstawione w przyszłych artykułach autora na podstawie już prowadzonych badań w danym zakresie, które nie są jeszcze gotowe do publikacji.

W dalszej części niniejszego artykułu przedstawione zostały badania poszczególnych modyfikacji algorytmów genetycznych zastosowanych przy

metodzie SOG. Badania te miały na celu określenie jak dane modyfikacje wpływają na czas oraz szybkość osiąganych wyników.

4. Badania modyfikacji algorytmów genetycznych

Przeprowadzone badania nad poszczególnymi algorytmami genetycznymi wykonano w środowisku .NET. Głównym powodem wyboru tego środowiska była możliwość opracowania algorytmów w paradygmacie obiektowym, oraz uproszczone procedury zrównoleglania obliczeń.

Aplikacja pozwalająca na badanie poszczególnych algorytmów genetycznych składa się z szeregu współdziałających klas, w których najważniejszą rolę odgrywa dziedziczenie. Taki sposób zapisu algorytmu gwarantuje ich kompatybilność – każda badana wersja zawierająca różne modyfikacje posiada taką samą strukturę, interfejs oraz operuje na tych samych danych. Rozszerzając klasę bazową, którą jest klasyczny algorytm genetyczny poszczególne badane metody przeciążały jeden, bądź dwa operatory genetyczne. Dzięki temu zabiegowi badane metody różniły się tylko i wyłącznie modyfikacją, która była przedmiotem tych badań. Dodatkowo obiekt chromosomu pozwalał na konwersję przechowywanych wartości poszczególnych genów zarówno na postać bitową (wymaganą przez pewne operatory), jak i postać zmiennopozycyjną (badaną w innych operatorach).

By dokładniej poznać różnice pomiędzy różnymi modyfikacjami wygenerowano również pojedynczą populację początkową, która została przekazana do wszystkich metod, tak więc wyeliminowano kolejny czynnik wpływający na osiąganie wyniki. Wszystkie modyfikacje zostały również przetestowane na tym samym zestawie parametrów, by również czynnik różnych parametrów nie zakłócał wyników tych badań. W rezultacie badaniu zostały poddane jedynie modyfikacje w strukturze algorytmu, a wszystkie pozostałe elementy zostały ujednolicone dla badanych algorytmów.

Badanie zostało przeprowadzone na 5 funkcjach (1 lub 2 parametrycznych), które reprezentują różne klasy problemów i były używane przez innych badaczy do ukazania zróżnicowania działania algorytmów [9]:

- $x\sin(10\pi x)+1$,
- $21.5+x\sin(4\pi x)+y\sin(20\pi y)$,
- $\frac{8x}{8}$,
- $x\,sgn(x)$,
- $0.5+\frac{\sin^2\sqrt{x^2+y^2}-0.5}{1+0.001(x^2+y^2)^2}$.

Na wskazanych funkcjach przetestowano 5 podejść, algorytmów:

- Klasyczny algorytm genetyczny
- Algorytm genetyczny z reprezentacją zmiennopozycyjną
- Algorytm „nieporządny”
- Elitarność w algorytmach genetycznych
- Mutację nierównomierną

W każdym z badanych podejść wykorzystano selekcję proporcjonalną opartą na kole ruletki i stałą populację. Klasyczny algorytm genetyczny bazując na reprezentacji binarnej korzystał z operatora jednopunktowego krzyżowania, oraz mutacji dla każdego bitu z zadanym prawdopodobieństwem. Przy reprezentacji zmiennopozycyjnej operator krzyżowania tworzył potomków odpowiednio w 1/3 i 2/3 odległościach pomiędzy osobnikami rodzicielskimi, natomiast operator mutacji zmieniał wartość genu losowo +/- o wartość z przedziału od 0 do zadanego prawdopodobieństwa mutacji procentowej wartości danego genu. Nieporządne algorytmy wprowadziły zmianę jedynie w operatorze krzyżowania, który losowo przyjmował wartość albo pierwszego, albo drugiego rodzica, albo średnią z ich wartości i w ten sposób generował dwoje potomków. Elitarność zarówno w operatorze krzyżowania jak i mutacji sprawdzała czy osobnik rodzicielski jest lepszy od osobnika nowo utworzonego, przy czym jako wynik operatory zwracały lepszy zbiór osobników. Natomiast ostatnia modyfikacja, mutacji nierównomiernej wprowadziła dodatkową funkcję, która na podstawie aktualnego numeru populacji wyliczała wartość o jaką zmieni się wartość mutującego genu. Funkcja ta ma postać:

$$x \cdot 1 - r^{1-\frac{itt}{60}},$$

gdzie *r* jest losową wartością przyjmującą wartość 0 lub 1, *itt* jest numerem aktualnej iteracji, a *x* maksymalną zmianą w genie, która w zależności od liczby losowej z równym prawdopodobieństwem wynosi:

$$\begin{cases} max - x \\ x - min \end{cases},$$

gdzie *max* i *min* to odpowiednio górne i dolne ograniczenia przestrzeni rozwiązań.

Każdą z metod uruchomiono 10 krotnie (z tymi samymi parametrami i populacją początkową), by pomimo losowości móc ustalić średnią wartość poszczególnych parametrów. Badaniu podlegały dwie wartości – dopasowanie najlepszego wyniku uzyskanego daną metodą, a także liczbę iteracji potrzebnych do uzyskania najlepszego wyniku. Warunkiem zakończenia poszczególnych algorytmów było osiągnięcie zadanej liczby iteracji bez poprawy rozwiązania.

Aplikacja do testowania równolegle uruchamiała 10 instancji pojedynczej metody w osobnych wątkach. Po wykonaniu wszystkich przebiegów dla jednej z metod uruchamiało kolejne 10 instancji dla kolejnej metody, aż zostały przebadane wszystkie badane modyfikacje.

5. Wyniki

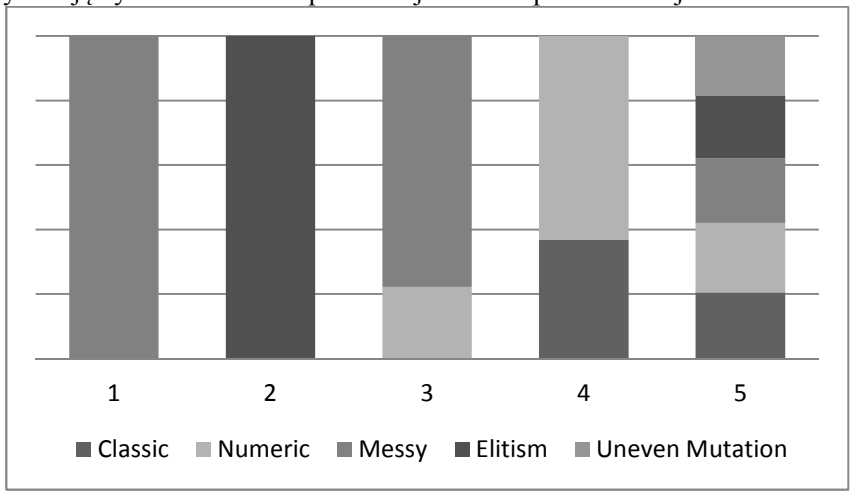
Wyniki dla parametrów:

- Wielkość populacji: 75,
- Iteracji bez polepszenia rozwiązania: 25 (warunek stopu),
- Parametr krzyżowania: 0.65,
- Parametr mutacji: 0.015,

można prześledzić na rysunkach 4 i 5.

Rysunek 4 przedstawia dokładność poszczególnych metod (zaznaczonych różnymi kolorami) dla poszczególnych funkcji (numery funkcji na osi odciętych). Im większy udział metody, tym lepsze otrzymywane były średnie wyniki w porównaniu z innymi metodami. Jak można łatwo zauważyć nie można znaleźć jednoznacznego zwycięzcy. Każda badana modyfikacja poprawiała wyniki w innej klasie problemu. Wyjątkiem jest tutaj Mutacja nierównomierna, która jedynie w ostatniej funkcji osiągnęła wyniki zbliżone do innych funkcji.

Na rysunku 5 można natomiast prześledzić liczbę iteracji, która była potrzebna każdej z metod do osiągnięcia najlepszej dla niej wartości rozwiązania problemu. Interpretując wykres, im mniejszy udział odpowiedniej metody (koloru) w słupku wskazującym na konkretny problem, tym szybciej uzyskała ona wynik. Zgodnie z oczekiwaniami, najbardziej ogólne, uniwersalne, podejście, czyli klasyczny algorytm genetyczny na rozwiązanie każdego z zadań potrzebował mniej więcej podobnej liczby iteracji. Pozostałe metody w zależności od problemu potrzebowały więcej, bądź mniej przebiegów. Najbardziej wrażliwą na różne klasy problemów modyfikacją było stosowanie reprezentacji zmiennoprzecinkowej.

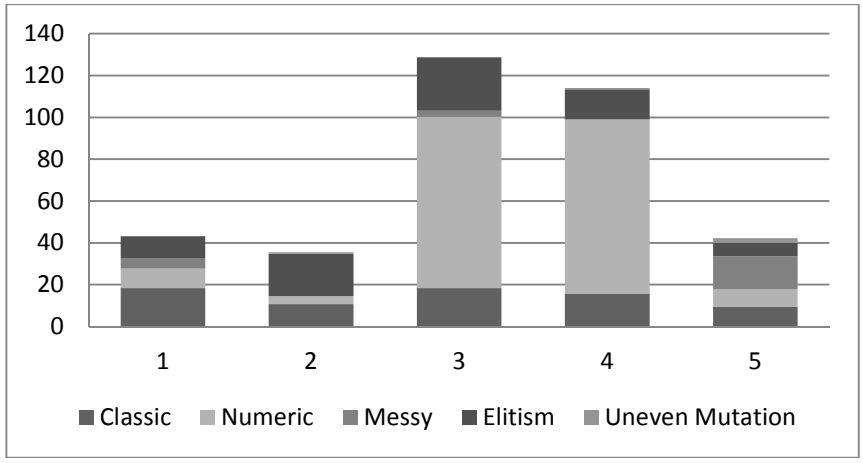


Rysunek 4. Dokładność rozwiązań zwracanych przez metodę

Porównując obydwa wykresy, można wnioskować, że najlepsze wyniki dało zastosowanie nieporządnych algorytmów genetycznych (messy) gdyż zarówno w dokładności zwracanych rozwiązań, jak i szybkości działania ma zadowalające wyniki ogólne. Jednakże dla każdego z rodzajów problemu inne modyfikacje wykazały się lepszymi osiągnięciami.

Metoda klasyczna zachowywała się stabilnie niezależnie od problemu, co jednoznacznie wskazuje na jej uniwersalność i niezależność od rozpatrywanych problemów. Reprezentacja numeryczna w przypadku funkcji 4 (zgodnie z kolejnością podaną w rozdziale 4) przy każdym przebiegu poprawiała swój rezultat znacząco wyprzedzając inne metody. Można zauważyć prawidłowość, że metoda ta działa albo długo i daje bardzo dobre wyniki, lub gdy nie jest w stanie ich poprawić szybko kończy swoją pracę. Nieporządne algorytmy genetyczne, oprogramowane jako nietypowy operator krzyżowania (nie badano właściwości zmiennej długości chromosomu na pamięć potrzebną do wykonania programu), okazała się jedną z lepszych modyfikacji. Wprowadzenie elitarności do algorytmów genetycznych wydłużyło ich czas działania w stosunku do klasycznego podejścia, jednak dla pewnej klasy problemów dało zaskakująco dobre rezultaty. Natomiast największym rozczarowaniem okazało się zastosowanie mutacji nierównomiernej, która jedynie dla ostatniej funkcji przy porównywalnych wynikach do pozostałych metod zakończyła swoje działanie w rekordowo krótkim czasie.

Warto jednak zauważyć, że z powodu dużej losowości występującej przy każdym przebiegu iteracji, a także innej populacji początkowej przy każdym wykonywanym teście (również generowanej losowo) wyniki mogą być różne nawet przy zastosowaniu tych samych parametrów. Przedstawione w niniejszej pracy rezultaty są uśrednieniem trendu widocznego przy dłuższym badaniu tych parametrów i mogą nie być weryfikowalne przy zaledwie kilku powtórzeniach danego badania.

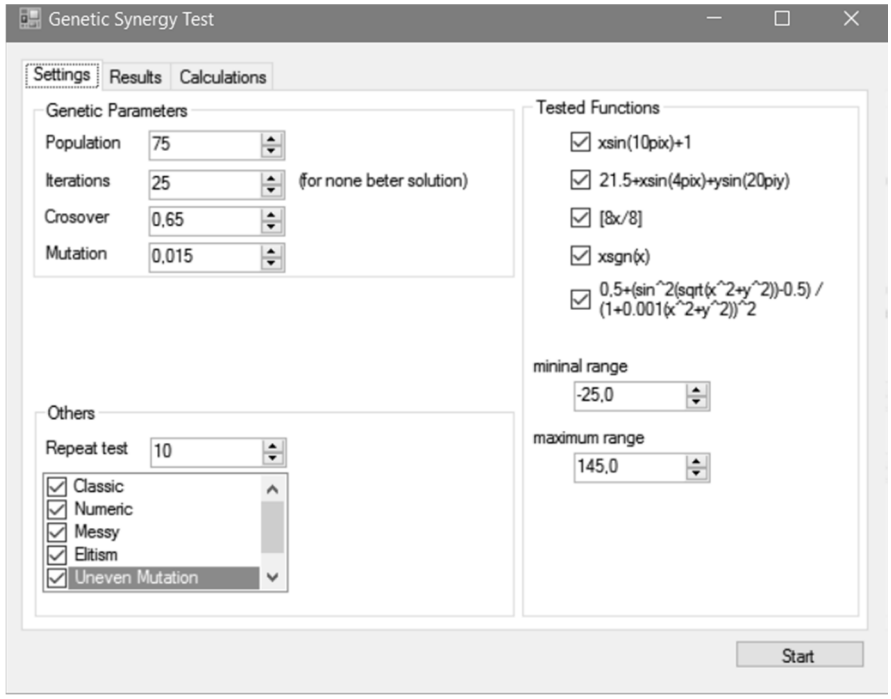


Rysunek 5. Ilość iteracji potrzebna poszczególnym metodom do osiągnięcia najlepszego wyniku.

6. Podsumowanie

Opracowana metoda samoadaptacyjnego algorytmu genetycznego ma na celu wytworzenie uniwersalnego algorytmu, który niezależnie od klasy problemu, a przede wszystkim z myślą o problemach rzeczywistych, będzie zwracać wyniki lepsze zarówno pod względem dokładności, jak i szybkości od klasycznego podejścia algorytmów genetycznych. W tym celu autor powołuje się na przebadane i dobrze udokumentowane modyfikacje, których porównanie w opisanych badaniach wskazało, że każda z nich jest lepsza w pewnej klasie problemów od podejścia klasycznego. Zastosowanie tych modyfikacji umożliwi wykorzystanie mocnych stron każdej z nich, co powinno odbić się korzystnie na wynikach otrzymywanych przez taki algorytm.

W ramach badań wytworzono również aplikację, za pomocą której można badać wpływ poszczególnych operatorów oraz podejść na pracę algorytmu genetycznego, eliminując (bądź minimalizując) wpływ losowości dzięki zapewnieniu jednakowych warunków testowania dla wszystkich badanych przypadków. Rozwój danej aplikacji (przedstawionej na rysunku 6) umożliwi badanie parami, oraz grupami poszczególnych modyfikacji celem maksymalizacji efektu synergii ich stosowania.



Rysunek 6. Aplikacja umożliwiająca testowanie poszczególnych modyfikacji algorytmów genetycznych

Dalsze badania autora skupiają się na określeniu siły wpływu poszczególnych modyfikacji na osiągane wyniki programu ewolucyjnego, a także na metodach związanych z oceną populacji zwracaną przez poszczególne metody (aby umożliwić rywalizację metod i ocenę ich działania w czasie rzeczywistym).

Bibliografia

1. Beluch W.: *Wpływ operatorów mutacji na skuteczność poszukiwań AE*, Obliczenia ewolucyjne, laboratorium 4, Wydział Mechaniczny Technologiczny, Politechnika Śląska,
   http://www.imio.polsl.pl/Dopobrania/Lab%20OE%2004%20(mutacja).pdf [dostęp: 29.05.2016]
2. Deb K.: *Multi-objective Optimization using Evolutionary Algorithms*. John Wiley & Sons Ltd, UK, 2001
3. Fogel D. B., Atmar J. W.: *Comparing Genetic Operators with Gaussian Mutations in Simulated Evolutionary Process Using Linear Systems*, Springer-Verlag, Biological Cybernetics vol. 63, 1990

4. Freisleben B, Merz P.: *New Genetic Local Search Operators for the Traveling Salesman Problem*, Springer Berlin Heidelberg, Applications Of Evolutionary Computation Evolutionary Computation In Computer Science And Operations Research, 2005
5. Goldberg D. E.: *Algorytmy genetyczne i ich zastosowania*, WNT, Warszawa 1998
6. Grygierek K.: *Samoadaptacyjna metoda algorytmów genetycznych w optymalizacji przestrzennych kratownic*, Modelowanie Inżynierskie nr52, Gliwice 2014
7. Gwiazda T. D.: *Algorytmy genetyczne kompendium tom1*, Wydawnictwo Naukowe PWN, Warszawa 2007
8. Gwiazda T. D.: *Algorytmy genetyczne kompendium tom2,* Wydawnictwo Naukowe PWN, Warszawa 2007
9. Michalewicz Z.: *Algorytmy genetyczne+struktury danych=programy ewolucyjne,* Wydawnictwa Naukowo-Techniczne, 1996
10. Paszyńska A.: *Projektowanie wspomagane komputerowo a problemy zbieżności algorytmów genetycznych*, rozprawa doktorska, Instytut Podstawowych Problemów Techniki PAN, 2007
11. Przewoźniczek M.: *Nowy szablon z kodowaniem nieporządnym jako remedium na typowe wady algorytmu genetycznego,* rozprawa doktorska, Wydział Informatyki i Zarządzania, Politechnika Wrocławska, Wrocław 2012
12. Słowik A.: *Właściwości i zastosowania algorytmów ewolucyjnych w optymalizacji*, Metody informatyki stosowanej, Koszalin 2007
13. Studniarski M.: *Wykłady z algorytmów genetycznych Część 1: Podstawowe informacje o algorytmach genetycznych*, Wydział Matematyki i Informatyki Uniwersytetu Łódzkiego, http://math.uni.lodz.pl/~marstud/AGwyklad1-2.pdf [dostęp: 26.12.2015]
14. Turing A. M.: *Computing Machinery and Intelligence. „Mind"*, Tom 59, nr 236, X/1950
15. Zitzler E.: *Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications*, rozprawa doktorska, Swiss Federal Institute of Technology Zurich, 1999

Streszczenie

W artykule przedstawiono nowe podejście do adaptacyjnych Algorytmów genetycznych. Koncepcja samoadaptacyjnej optymalizacji genetycznej opiera się na wprowadzeniu meta-algorytmu, w ramach którego poszczególne algorytmy genetyczne (z różnymi operatorami oraz parametrami) rywalizują między sobą. Artykuł zawiera wstępne badania, ukazujące działanie różnych modyfikacji

algorytmów genetycznych na wybranych problemach. Przeprowadzone eksperymenty wskazują, że użycie strategii samoadaptacji w proponowanym zakresie może przynieść obiecujące rezultaty.

Opisywane w niniejszym dokumencie prace ukazują porównanie modyfikacji takich jak: reprezentacja numeryczna chromosomów, nieporządne algorytmy genetyczne, mutacja nierównomierna czy elitarność. Wyniki różnych podejść zostały również porównane do klasycznego podejścia (reprezentacja binarna, jednopunktowe krzyżowanie).

Słowa kluczowe: Algorytmy genetyczne, Adaptacja genetyczna, reprezentacja numeryczna chromosomów, nieporządne algorytmy genetyczne, mutacja nierównomierna, elitarność

Abstract

This paper presents a new way of adaptive in genetic algorithms. Concept of self-adaptive genetic optimization was based on meta-algorithm, where different operators with different parameters competitive with each other. The paper contains preliminary research, showing how the various genetic algorithms modification react with different problems. Conducted experiments suggest that developed self-adaptive strategy for real problem optimization using genetic algorithms may return promising results.

Described research compare genetic modification as: chromosome numeric representation, messy genetic algorithms, uneven mutation and elitism. The results of different approach have been also compared to result of classic genetic algorithm (with binary representation, one-point crossing).

Keywords: Genetic algorithms, genetic adaptive, real problem optimization, numeric representation, messy genetic algorithms, uneven mutation, elitism.

Bohdan Andriyevsky
Chair of Electronics
Department of Electronics and Computer Sciences
Koszalin University of Technology, Poland

Zbigniew Czapla
Department of Physics[1]
Opole University of Technology, Poland
Institute of Experimental Physics[2]
University of Wrocław, Poland

Band electronic structure and dielectric functions of $(C_3N_2H_5)_2SbF_5$ crystals

Key words: crystals, first principles calculations, electronic structure, optical properties

1. Introduction

The bis(imidazolium) pentafluoroantimonate(III) crystal $(C_3N_2H_5)_2SbF_5$ is one of the known crystal of the general chemical formula A_2SbX_5, where $A = NH_4^+$ and $C_3N_2H_5^+$ and X = F and Cl. In the case of the $(NH_4)_2SbF_5$ a sequence of phase transition was found and ferroelastic properties below 292 K [1 - 5]. In the case of $(C_3N_2H_5)_2SbCl_5$ the para- ferroelectric phase transition was revealed at 180 K [6]. The $(C_3N_2H_5)_2SbF_5$ was found to exhibit first order structural phase transition from orthorhombic room temperature phase (space group *Pnma*) to monoclinic one (space group $P2_1/c$) at about 220 K [7]. Structural studies revealed strong disorder of imdazolium cations at room temperature (phase I). Below phase transitions temperature (phase II) imidazolium cations are completely ordered. Disorder of cations is seen well in phase I, where enhanced values of permittivity (45 - 55) were observed along the a- and c-axes. At the phase transition temperature an abrupt decrease of permittivity connected with freezing of imdazolium cations was observed. Reorientational motion of imdazolium cations were confirmed and described in dielectric dispersion studies in the frequency range 10 kHz – 1 MHz [8]. Above room temperature the crystal reveals the properties, which permit to treat one as the ionic conductor [8].

Until now, no theoretical *ab inito* based studies of the materials properties of $(C_3N_2H_5)_2SbF_5$ are known. Such investigations could however deliver information,

necessary for the better understanding of phase transition observed in $(C_3N_2H_5)_2SbF_5$ on the basis of peculiarities of interatomic interactions in the crystal studied. We present results of the first principles calculations of electronic band structure of $(C_3N_2H_5)_2SbF_5$ obtained for the first time.

2. Methods of investigations

First principles calculations of the band electronic structure and optical properties of $(C_3N_2H_5)_2SbF_5$ were carried out using the CASTEP code (CAmbridge Serial Total Energy Package) [9] based on the density functional theory (DFT) and the plane-wave basis set. The generalized gradient approximation with the Perdew−Burke−Ernzerhof functional (GGA-PBE) for the exchange and correlation effects [10] together with ultrasoft pseudopotentials [11] were used for the calculations. A cutoff energy of 340 eV was assumed in the plane-wave basis set. During the self-consistent electronic minimization, the eigen-energy convergence tolerance was chosen to be 2.4×10^{-7} eV and the tolerance for the electronic total energy convergence during optimization was 1.0×10^{-5} eV. The corresponding maximum ionic force tolerance was $3\cdot10^{-2}$ eV/Å and the maximum stress component tolerance was $5\cdot10^{-2}$ GPa. Subsequently, the electronic properties, such as the band dispersion $E(k)$, partial density of states (PDOS), and dielectric functions $\varepsilon(E)$ were computed at the respective optimized geometry. In view of relatively large unit cell dimensions of the crystal studied ($a = 7.50$ Å, $b = 21.00$ Å, $c = 7.56$ Å, $\beta = 91°45'$ [7]) five k-points in the irreducible Brillouin zone were used (corresponds to the fine k-points setting), as well as a smearing of 0.1 eV. The calculations were performed at the ultrasoft and norm conserving pseudopotentials. The latter one was used for the calculations of phonons using density functional perturbation theory (DFPT).

For the proper reflection of the observable physical properties the considering of the van-der-Waals (vdW) and hydrogen types interactions in theoretical calculations could be indispensible. Therefore, the vdW interaction has been taken into account at presented *ab initio* calculations of $(C_3N_2H_5)_2SbF_5$ using CASTEP package [12, 13].

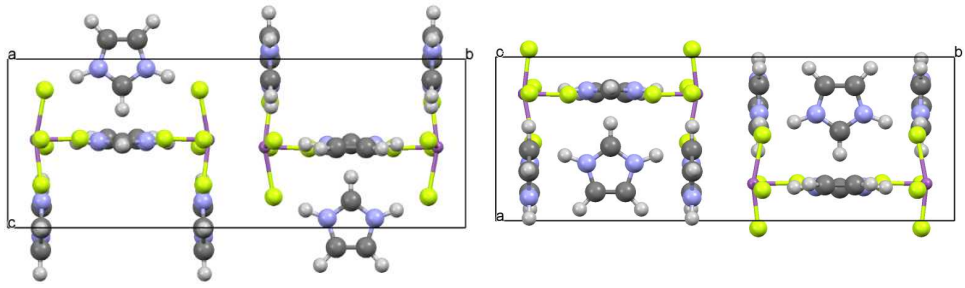
3. Results and discussion

The experimental [7] and computationally optimized crystal unit cell parameters of $(C_3N_2H_5)_2SbF_5$ at the space group of symmetry $P2_1/m$ (no. 11) are presented in Table 1.

Table 1. Unit cell parameters of $(C_3N_2H_5)_2SbF_5$ crystal at the space group of symmetry $P2_1/m$

Unit cell parameters Source	a /Å	b /Å	c /Å	β /degree	V /Å^3
Experiment [7] at T = 85 K	7.50	21.0	7.56	91.45	1190
Calculations without vdW interactions	7.99	25.6	8.10	97.1	1644
Calculations with vdW interactions	7.58	20.9	7.62	92.4	1207

The vdW interaction of the Grimme's form has been taken into account at calculations using CASTEP code. In the case without taking into account the vdW interaction, the optimized structure of the crystal was obtained after 88 iterations, whereas, in case with taking into account the vdW interaction, the necessary number of iterations steps was almost twice smaller (45). The structural data obtained (Table 1) indicate that the consideration of vdW interaction in the *ab initio* structure optimization is rather indispensible to obtain reliable results for $(C_3N_2H_5)_2SbF_5$, that was observed also at studies of similar molecular type crystals [14]. Two views of $(C_3N_2H_5)_2SbF_5$ structure are presented in figure 1, which show clear difference between cuts, perpendicular to *a*- and *c*-axes.



(a) (b)

Figure 1. View of the optimized crystal structure of monoclinic $(C_3N_2H_5)_2SbF_5$ along *a*-axis (a) and *c*-axis (b) (space group no. 11). Hydrogen - white, carbon - grey, nitrogen - blue, fluorine – green, antimony - violet.

Band electronic structure of $(C_3N_2H_5)_2SbF_5$ consists of a number of narrow valence bands that indicates low 'band energy - wave vector' dispersion $E(K)$ (Fig. 2). The

corresponding valence - to - conduction band gap $E_g = 4.362$ eV is found to be indirect, taking place between ZΓ and E points of Brillouin zone (Fig. 2).

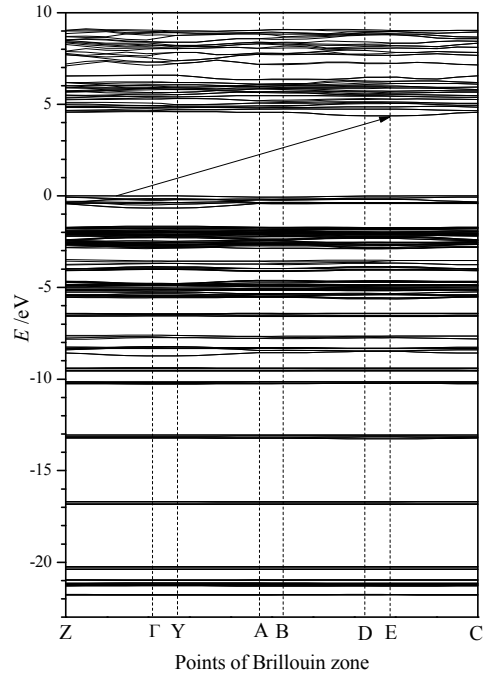


Figure 2. Band structure of $(C_3N_2H_5)_2SbF_5$ at monoclinic symmetry group no. 11. Arrow indicates the electronic transition from the highest valence to the lowest conduction band.

The partial density of electronic states (PDOS) of $(C_3N_2H_5)_2SbF_5$ consists of 14 narrow valence bands of the width 1 – 2 eV (Fig. 3). On the other hand, the degree of hybridization of these bands is relatively high. These narrow bands and their clear hybridization are the main characteristic features of the electronic structure of the crystal.

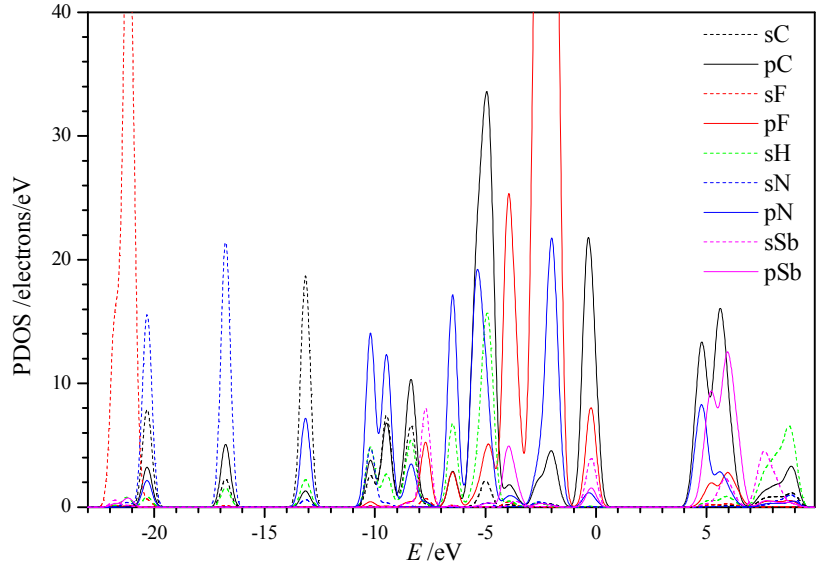


Figure 3. Partial density of electronic states of $(C_3N_2H_5)_2SbF_5$ at the monoclinic symmetry group no. 11 for all chemical elements (H, C, N, F, Sb) and different orbital moments (*s*, *p*).

The calculated dielectric function ε(ηω) in the range of electronic excitations of $(C_3N_2H_5)_2SbF_5$ reveals significant anisotropy (Figs. 4, 5). Namely, the real part of dielectric function ε_1 polarized along the *b*-axis (ε_{1y}) is clearly smaller than the components ε_{1x} and ε_{1z} polarized along the perpendicular axes *a* and *c* (Fig. 4). Also, the low photon energy shoulder of the imaginary part of dielectric function $\varepsilon_{2y}(\eta\omega)$ is shifted near 1 eV into the higher photon energies $\hbar\omega$ (Fig. 5). Thus, $(C_3N_2H_5)_2SbF_5$ reveals peculiarities of the layer structure crystal. In the ranges of the crystal unit cell close to the fractional coordinates $y = 0$ and $y = 0.5$, the smallest distances between the negative fluorine ions of the SbF_5 group and the positive carbon ions of pentagon of the $C_3N_2H_5$ group is equal to 3.26 Å, that is too big to form the relatively strong bond. The total electronic charges of atoms in the crystal $(C_3N_2H_5)_2SbF_5$ are presented in Table 2.

Table 2. The total electronic charges of atoms in the crystal $(C_3N_2H_5)_2SbF_5$ obtained using CASTEP code

Atom	H	C	N	F	Sb
Charge /e	0.57 - 0.71	3.92 - 4.17	5.47	7.64 - 7.67	2.17

The interatomic distances Sb - F are equal to 2.29 Å (within one SbF_5 group) and 3.53 Å (for two neighboring SbF_5 groups). Because of the absence of more shorter interatomic distances between $C_3N_2H_5$ and SbF_5 groups in this region, the value 3.26 Å may be regarded as the interlayer distance. In the ranges [0 - 0.5] and [0.5 - 1.0] of the unit cell *y*-coordinates, the shortest interatomic distances F - H are changed between 1.61 Å and 2.35 Å. So, here the interatomic bonds between the groups $C_3N_2H_5$ and SbF_5 are stronger. The above remarks on the interatomic distances and corresponding bonding in the crystal $(C_3N_2H_5)_2SbF_5$ permit to regard the half of the unit cell dimension $b/2 \approx 10$ Å as the thickness of the layers mentioned.

When taking into account the above remarks on the crystal structure of $(C_3N_2H_5)_2SbF_5$, the characteristic features of PDOS of the crystal (Fig. 3), become understandable: (1) relatively large number of PDOS valence bands are caused by small interatomic interaction in the *b*-direction of the crystal, and (2) clear hybridization of the electronic states within separate PDOS band takes place due to the relatively high interatomic interaction within one structural *b*-layer of the crystal.

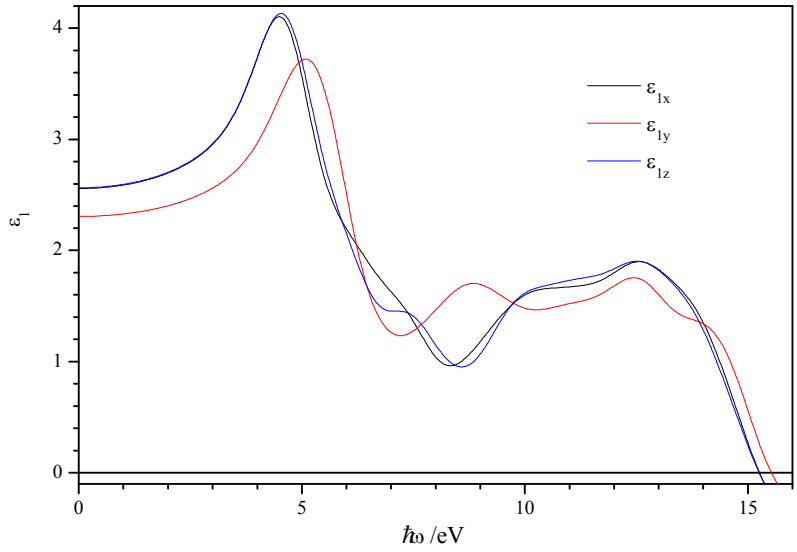


Figure 4. The Cartesian components ε_{1x}, ε_{1y}, and ε_{1z} of the real part of dielectric function ε_1 versus photon energy $\hbar\omega$ for $(C_3N_2H_5)_2SbF_5$ at the monoclinic symmetry group no. 11.

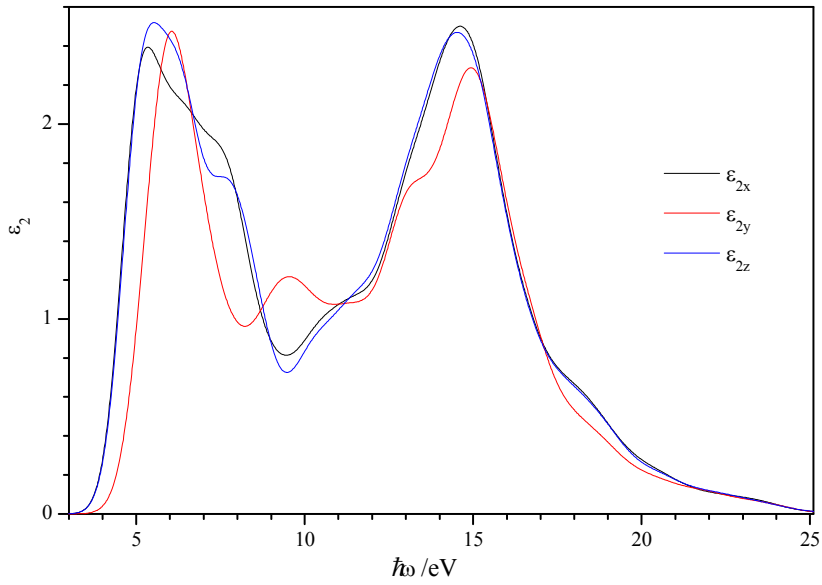


Figure 5. The Cartesian components ε_{2x}, ε_{2y}, and ε_{2z} of the imaginary part of dielectric function ε_2 versus photon energy $\hbar\omega$ for $(C_3N_2H_5)_2SbF_5$ at the monoclinic symmetry group no. 11.

The vibration spectrum of $(C_3N_2H_5)_2SbF_5$ has been calculated also using the CASTEP code of the Materials Studio 8.0 package [9] in the frequency range of 0 cm^{-1} - 3300 cm^{-1}. In the wavenumber range of 1000 cm^{-1} - 3300 cm^{-1}, the calculated partial DOS is associated mainly with the vibrations of H – C and H – N bonds. The calculated Raman spectrum of $(C_3N_2H_5)_2SbF_5$ is presented in figure 6. Unfortunately, we have no possibility to compare the Raman spectrum obtained with the corresponding experimental data because of the absent of the reference experimental studies for $(C_3N_2H_5)_2SbF_5$.

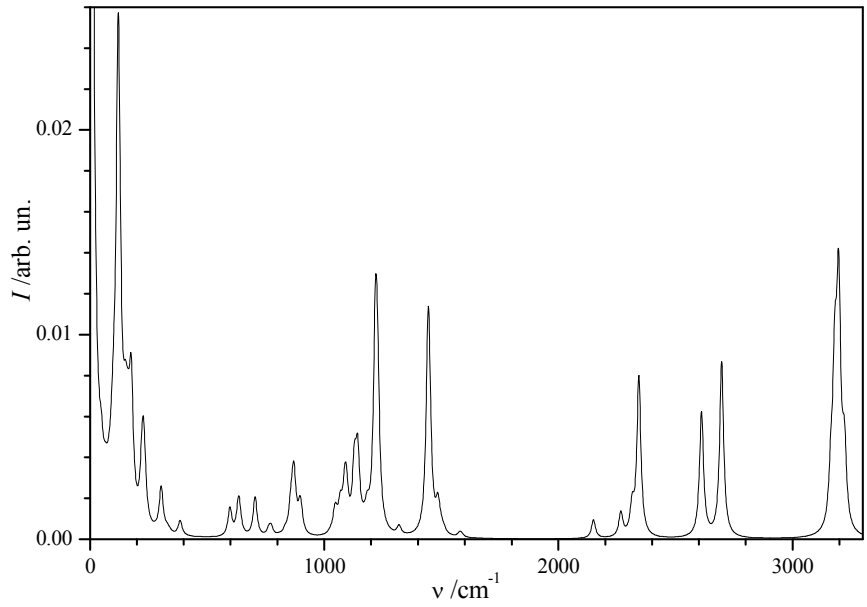


Figure 6. Raman spectrum of $(C_3N_2H_5)_2SbF_5$ at the space group of symmetry no. 11 computed in the framework of the DFPT using the CASTEP code of Materials Studio 8.0 package.

Conclusions

1. The layer-type crystal structure of $(C_3N_2H_5)_2SbF_5$ at the monoclinic symmetry (space group no. 11) has been revealed clearly in the characteristics of band electronic structure and optical functions of the crystal: (1) relatively large number of PDOS valence bands are caused by small interatomic interaction in the *b*-direction of the crystal; (2) clear hybridization of the electronic states within separate PDOS band takes place due to the relatively high interatomic interaction within one structural *b*-layer of the crystal.
2. The crystals $(C_3N_2H_5)_2SbF_5$ are characterized by the relatively large anisotropy of the real part of dielectric function ε_{1i} ($i = x, y, z$) due to the value ε_{1y}, which is smaller than the values ε_{1x} and ε_{1z} by 0.3. The corresponding maximum birefringence $\Delta n_{xy} \approx \Delta n_{zy} = 0.1$ is also relatively large, that makes this crystal perspective for practical use as an active material for the birefringence based optical polarizers.

Acknowledgments

BA acknowledges support from the WCSS computer center of Wrocław Technical University, where the calculations using CASTEP program were performed in the framework of the project no. 053.

References

1. Avkutskij L, Davydovitch R, Zemnukova I, Gordienko D, Urbanovicius V and Grigas J 1983 *Phys. Stat. Sol. (b)* **116** 483
2. Nakamura N 1986 *Naturforsh.* **41**a 243
3. Ryan R R, Cromer D T 1972 *Inorg. Chemistry* **11** 2322
4. Czapla Z and Dacko S 1993 *Ferroelectrics* **140** 271
5. Przesławski J, Furtak J and Czapla Z 2006 *Ferroelectrics* **337** 139
6. Piecha A, Białońska A and Jakubas R 2012 *J. Mater.Chem.* **22** 333
7. Czapla Z, Krawczyk M K, Ingram A and Czupiński O 2015 *J. Phys. Chem. Solids* **87** 233
8. Ingram A, Czapla Z and Wacke S 2016 *Curr. Appl. Phys.* **16** 278
9. Clark S J, Segall M D, Pickard C J, Hasnip P J, Probert M J, Refson K and Payne M C 2005 *Zeitschrift für Kristallographie* **220** 567
10. Perdew J P, Burke K and Ernzerhof M 1996 *Phys. Rev. Lett.* **77** 3865
11. Vanderbilt D 1990 *Phys. Rev. B* **41** 7892
12. McNellis E R, Meyer J and Reuter K 2009 *Phys. Rev. B* **80** 205414
13. Tkatchenko A and Scheffler M 2009 *Phys. Rev. Lett.* **102** 073005
14. Andriyevsky B, Kurlyak V Yu, Stadnyk V Yo, Romanyuk M O and Patryn A 2015 *Mater. Chem. Phys.* **162** 787

Abstract

Structural and electronic properties of the ferroelastic crystal $(C_3N_2H_5)_2SbF_5$ of the molecular type were studied by *ab initio* methods in the framework of the density functional theory. Band electronic structure, density of electronic states and dielectric functions in the range of valence electrons excitations of the crystal in the monoclinic phase (space group no. 11) have been obtained using the plane waves, ultrasoft pseudopotentials and van-der-Waals corrections. The electronic values obtained are discussed from the viewpoint of the layer-type crystal structure of $(C_3N_2H_5)_2SbF_5$.

Streszczenie

Strukturalne i elektronowe właściwości ferroelektrycznego kryształu $(C_3N_2H_5)_2SbF_5$ typu molekularnego zostały obliczone w ramach teorii funkcjonału gęstości (DFT) z wykorzystaniem odpowiedniej metody z pierwszych zasad (*ab initio*). Pasmowa struktura elektronowa, gęstość stanów elektronowych i funkcje dielektryczne w zakresie wzbudzenia elektronów walencyjnych kryształu zostały obliczone dla strukturalnej fazy jednoskośnej (grupa przestrzenna no. 11) z wykorzystaniem płaskich fal, super pseudopotencjałów miękkich i uwzględnienia poprawek na oddziaływania międzyatomowe typu van-der-Waalsa. Otrzymane wielkości elektronowe zostały omówione pod kątem warstwowej struktury krystalicznej $(C_3N_2H_5)_2SbF_5$.

Słowa kluczowe: kryształy, obliczenia z pierwszych zasad, struktura elektronowa, właściwości optyczne

Daniel Pieniak
Jarosław Zubrzycki
Łukasz Wojciechowski
Instytut Technologicznych Systemów Informacyjnych
Wydział Mechaniczny, Politechnika Lubelska
20-618 Lublin, ul. Nadbystrzycka 36
daniel.pieniak@gmail.com
j.zubrzycki@pollub.pl
l.wojciechowski@pollub.pl

Zastosowanie technologii rapid prototyping do projektowania elementów maszyn

Cel pracy

Celem niniejszej pracy jest przedstawienie oraz omówienie zastosowania technologii Rapid Prototyping do projektowania części maszyn i mechanizmów. W artykule przedstawiono autorski proces Szybkiego Prototypowania końcówek chwytnych do chwytaka pneumatycznego KGG 60-40 dedykowanego dla robota przemysłowego. Zakres pracy obejmuje przedstawienie idei Rapid Prototyping, omówienie technik RP oraz szczegółowe zaprezentowanie procesu tworzenia elementów chwytnych – począwszy od zaprojektowania części, badań wytrzymałościowych, kończąc na stworzeniu fizycznej części chwytnej za pomocą druku 3D.

Wstęp

Rapid Prototyping jest zbiorem metod szybkiego wytwarzania modeli fizycznych, części lub prototypów maszyn z pominięciem tradycyjnych technologii mechanicznych, takich jak: odlewanie, obróbka ubytkowa, czy elektroerozyjna. Podstawą wszystkich metod Szybkiego Prototypowania jest model 3D danego elementu. Po odpowiednim przetworzeniu i dostosowaniu do produkcji można wytworzyć go np. za pomocą jednej z technik SP. Główną zaletą RP jest to, że pozwala na stworzenie fizycznego modelu nadającego się do zademonstrowania potencjalnym inwestorom, którzy nie muszą mieć odpowiedniej wiedzy lub doświadczenia, potrzebnego do analizy modeli wirtualnych. Można przeprowadzić wstępne badania, a także zweryfikować to, czy rozwiązania, które wstępnie zastosowano, zadziałają w rzeczywistości. Charakterystyczne jest to, że w sposób

szybki i korzystny ze względu na koszty, można wytworzyć rzeczywisty przedmiot bezpośrednio w oparciu o dane z programów do Komputerowego Wspomagania Projektowania, bez zastosowania form czy narzędzi [1, 4, 5].

Metody Szybkiego Prototypowania to przede wszystkim metody generacyjne - addytywne, budujące dany element za pomocą warstw materiału układanych jedna na drugą. Modele przestrzenne stworzone w systemach CAD są podzielone za pomocą poziomych płaszczyzn na warstwy, z których następnie generuje się łatwiejsze do przetworzenia modele dwuwymiarowe. Większość metod RP bazuje na punktowym utwardzaniu materiału np. za pomocą lasera. Powtórzenie tego procesu dla wszystkich warstw pozwala na stworzenie fizycznego obiektu [6].

Metody Rapid Prototyping pozwalają obecnie na obróbkę szerokiej gamy materiałów takich jak: tworzywa sztuczne, foto-polimery, wosk, nylon, materiały drewnopodobne, materiały ceramiczne, proszki metaliczne. Wspólną cechą wszystkich metod Rapid Prototyping jest to, że tworzenie danego przedmiotu odbywa się nie poprzez usuwanie warstw, lecz przez dodawanie kolejnych warstw materiału. Umożliwia to wytworzenie w relatywnie krótkim czasie modeli o złożonych kształtach, bez zastosowania skomplikowanych narzędzi.

Historia Rapid Prototyping sięga początków lat 80. XX wieku. W 1984 roku Charles Hull, założyciel firmy *3D SYSTEMS* wynalazł stereolitografię, która jest jedną z częściej stosowanych addytywnych metod produkcji elementów. W 1988 roku firma *3D SYSTEMS* stworzyła maszynę stereolitograficzną o nazwie SLA-250, która została później uznana za pierwszą drukarkę 3D w historii. W 1994 roku dr James Bredt i Tim Anderson wykupili od MIT patent dotyczący techniki drukowania trójwymiarowego i założyli firmę *Z-Corporation*. W tym samym czasie firma Hulla stworzyła nową technologię Szybkiego Prototypowania, którą była SLS - modelowanie przy użyciu lasera o wysokiej mocy. W 1989 roku kolejna z firm *Stratasys* wynalazła inną metodę, jaką była FDM - modelowanie ciekłym tworzywem sztucznym. Należy dodać, iż w latach 80. XX wieku nikt nie używał terminu druk 3D. Użyto go po raz pierwszy w 1996 roku, podczas prezentacji pierwszych szeroko dostępnych modeli maszyn Szybkiego Prototypowania, których rozwiązania techniczne i technologiczne stały się wzorcem dla późniejszych produktów. W 2005 roku *Z-Corporation* stworzyło pierwszą na świecie kolorową drukarkę 3D do druku w wysokiej rozdzielczości 600x540 dpi. Wszystkie te firmy do dzisiaj pracują nad udoskonalaniem technologii druku przestrzennego. Jednym z przykładów jest RepRap, czyli idea samoreplikującej się drukarki. Za jej pomocą można wyprodukować praktycznie każdy element, w cenie przystępnej dla przeciętnego człowieka [8].

Szybkie Prototypowanie znajduje szerokie zastosowanie w wielu gałęziach przemysłu i życia codziennego. W przemyśle wykorzystywane jest głównie do budowy prototypów maszyn dla celów weryfikacyjnych i badawczych, oraz budowy fizycznych modeli w celach projektowych lub prezentacyjnych [2, 3]. Wraz

z rozwojem technologii RP zaczęto stosować je również do wytwarzania części i wyrobów jednostkowych i małoseryjnych. Rapid Prototyping wykorzystuje się także w medycynie np. przy tworzeniu protez, jak również w architekturze do stworzenia makiet budynków [5, 7]. Metody RP pozwalają na relatywnie tanią i szybką produkcję spersonalizowanych przedmiotów np. części maszyn, lub elementów protez. Dzięki edycji trójwymiarowego modelu każdy element można dostosować do indywidualnych potrzeb, a sama produkcja nowej części nie wymaga zmian form i narzędzi produkcyjnych [8].

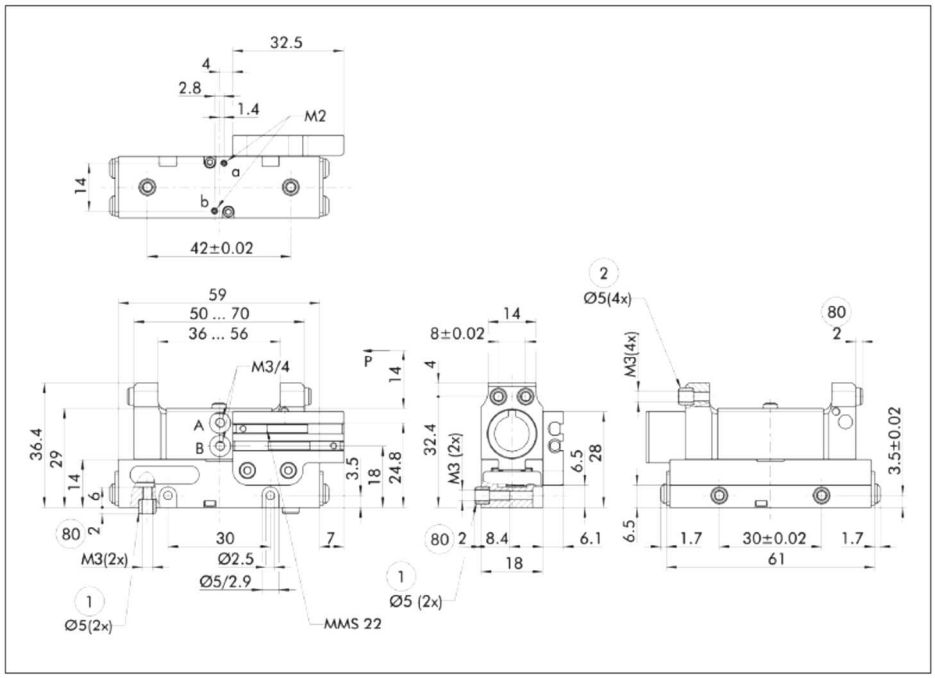
1. Projekt części chwytnej

W artykule opisano przeprowadzony proces Szybkiego Prototypowania. Na przykładzie elementów chwytnych do chwytaka pneumatycznego KGG 60-40 firmy SHUNK stworzono wirtualny prototyp części. Następnie przeprowadzono analizę wytrzymałościową modelu, oraz sprawdzono dokładność teselacji jego geometrii. Kolejnym etapem było odpowiednie ułożenie modelu na płaszczyźnie pod modelowej, oraz wygenerowanie supportu. Dzięki temu otrzymano plik z G-codem, sterującym ruchami głowicy urządzenia. Ostatnim etapem było stworzenie fizycznego elementu za pomocą urządzenia opartego na technologii FDM.

1.1. Wstępne założenie

Danymi wejściowymi, które posiadano, były rysunki techniczne chwytaka KGG 60-40, oraz jego specyfikacja techniczna, którą pozyskano ze strony producenta [10].

Na podstawie powyższych informacji wstępnie nakreślono ogólną koncepcję elementów chwytnych: ich kształt, zasadę działania, oraz sposób wykorzystania. Za pomocą technologii Rapid Prototyping chciano stworzyć dwa identyczne ramiona, które zamocowano by w chwytaku pneumatycznym. Następnie zamierzano przeprowadzić analizę wytrzymałościową modelu, oraz obliczenia wytrzymałościowe i symulację odkształceń Metodą Elementów Skończonych. Ostatecznie za pomocą urządzenia FDM marki uPrint założono wykonanie dwóch zaprojektowanych elementów. Celem było stworzenie w pełni funkcjonalnego chwytaka do robota Kawasaki RS03N.



Technical data

Description		KGG 60-20	KGG 60-40
ID		0303050	0303051
Stroke per finger	[mm]	10	20
Closing force	[N]	45	45
Opening force	[N]	53	53
Weight	[kg]	0.09	0.11
Recommended workpiece weight	[kg]	0.23	0.23
Air consumption per double stroke	[cm³]	3	6
Min./max. operating pressure	[bar]	2/8	2/8
Nominal operating pressure	[bar]	6	6
Closing/opening time	[s]	0.03/0.03	0.04/0.04
Max. permitted finger length	[mm]	42	42
Max. permitted weight per finger	[kg]	0.04	0.04
IP class		40	40
Min./max. ambient temperature	[°C]	-10/90	-10/90
Repeat accuracy	[mm]	0.02	0.02

Rysunek 1.1. Dokumentacja techniczna chwytaka KGG 60-40 [10]

1.2. Programy komputerowe wykorzystane w procesie modelowania

Pierwszym programem, z którego skorzystano, był Solid Edge – wersja studencka firmy Siemens PLM Software (licencja w posiadaniu Wydziału Mechanicznego Politechniki Lubelskiej). W tym systemie CAD-3D stworzono wirtualny model ramienia chwytnego, przeprowadzono analizę i symulację wytrzymałościową naprężeń i odkształceń MES przy zadanym obciążeniu wynikającym z warunków pracy. W późniejszym etapie stworzono złożenie z modelem

chwytaka dostępnym na stronie producenta, w celu sprawdzenia poprawności stworzonego mechanizmu. Na końcu wygenerowano plik w formacie STL, potrzebny w dalszych etapach procesu.

W procesie wzięto pod uwagę to, że przy zapisie modelu w formacie STL geometria jego powierzchni zostaje uproszczona i mogą nastąpić przerwania struktury. Drugim programem, z którego skorzystano był Netfabb – program na licencji freeware, za pomocą, którego sprawdzono dokładność wygenerowanego pliku STL. Przy Szybkim Prototypowaniu nie jest to punkt obowiązkowy, jednakże dzięki niemu najłatwiej jest wychwycić potencjalne błędy w modelu. Co więcej, program ten zawiera funkcję automatycznej naprawy w razie jakichkolwiek nieprawidłowości w bryle modelu. Etap ten można było w zasadzie pominąć, jednak przy niepoprawnych plikach mogą wystąpić puste warstwy, lub swobodne wypełnienia poza obszarem części.

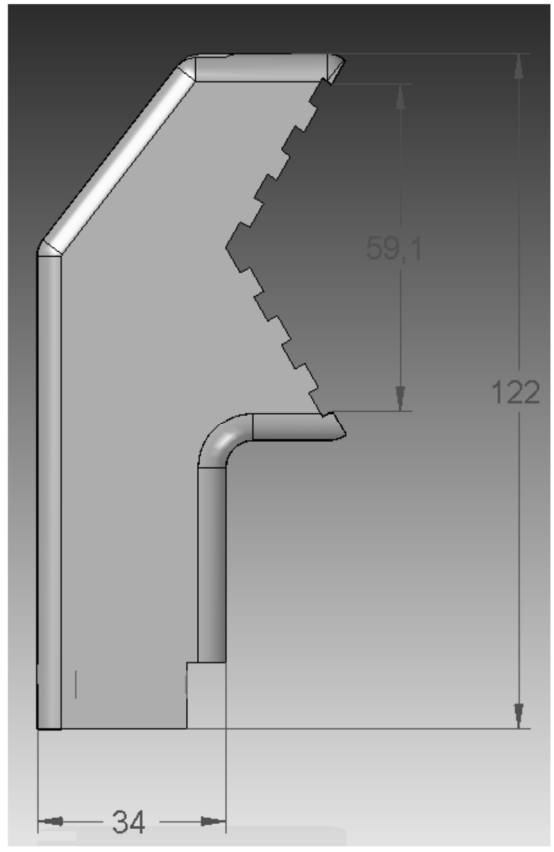
Następnym oprogramowaniem był Slic3r, będące kolejnym darmowym oprogramowaniem, pozwalającym wygenerować G-code potrzebny do stworzenia rzeczywistej części. Jak wspomniano wcześniej, Szybkie Prototypowanie opiera się na addytywnym tworzeniu rzeczywistej części, poprzez podzielenie modelu wirtualnego na warstwy, które są później nakładane na siebie. Slic3r jest programem, przetwarzającym model części poprzez podzielenie go na przekroje i wygenerowanie G-Codu do urządzenia RP. Do programu można wprowadzić wszystkie parametry drukarki, takie jak wymiary platformy i parametry druku. Następnie program generuje wirtualne urządzenie, w którym umieszcza się opracowane elementy. Slic3r uwzględniając konfigurację druku sam generuje materiał podporowy oraz plik z G-codem. Potrafi on również wizualizować model trójwymiarowo w taki sposób, że każda warstwa może być wyświetlana jako oddzielny obiekt. Pozwala to bardzo łatwo zidentyfikować niechciane elementy modelu.

Następnym oprogramowaniem wykorzystanym w procesie jest CatalystEX. Jest to dedykowane oprogramowanie sterujące urządzeniem uPrint. Łączy ono funkcję programów Slic3r oraz Netfabb. Pozwala w przystępny sposób określić położenie części, oraz wizualizuje poszczególne ścieżki głowicy. Posiada też funkcje komunikowania się z urządzeniem, dzięki czemu przesyła kod sterujący bezpośrednio do urządzenia.

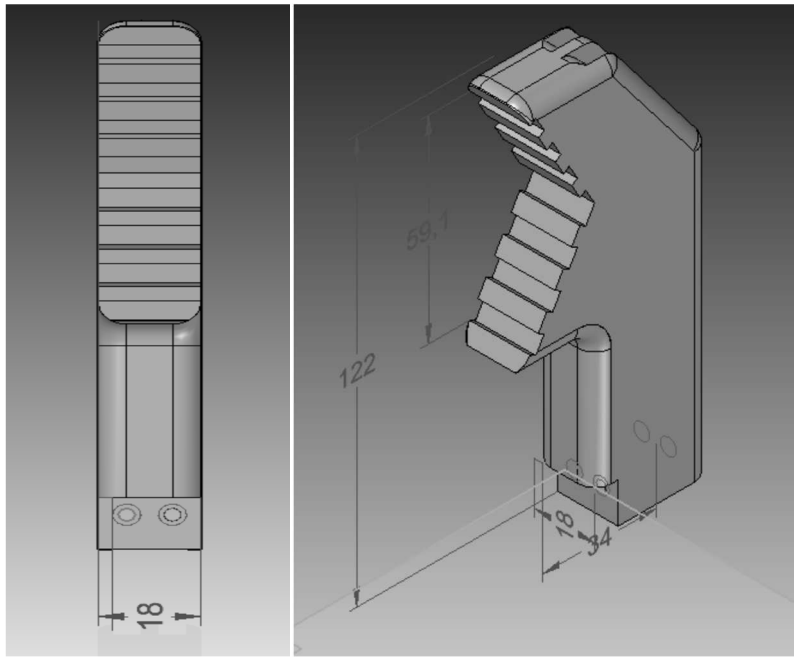
2. Przebieg procesu

2.1. Model 3D

Pierwszym etapem procesu Szybkiego Prototypowania było stworzenie modelu końcówki chwytnej w programie CAD. Został on opracowany na podstawie posiadanych danych o wymiarach, oraz siłach działających na element dzięki temu nakreślono wstępny zarys części. Wzorując się na rzeczywistych końcówkach zaprojektowano poniższy element.



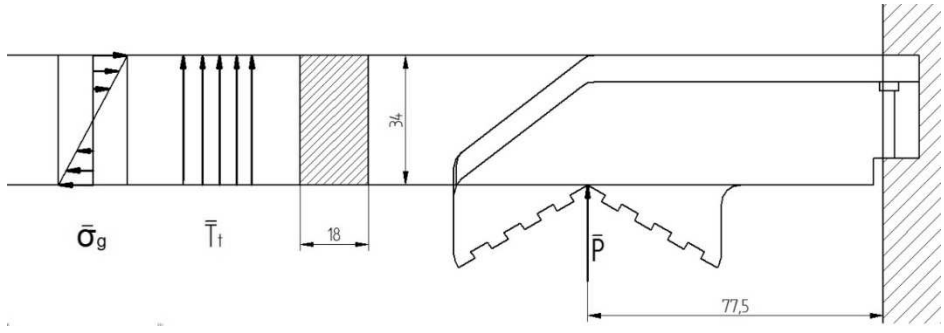
Rysunek 2.1. Projekt końcówki chwytnej (źródło – opracowanie własne)



Rysunek 2.2. Projekt końcówki chwytnej (źródło – opracowanie własne)

2.2. Badania wytrzymałościowe

W drugim etapie procesu RP przeprowadzono analizę wytrzymałościową chwytaka. Stwierdzono, że najlepsze będą dwa sposoby. Pierwszy z nich to klasyczna metoda obliczeniowa, w której policzono naprężenia zredukowane występujące w elemencie i porównano je z dopuszczalnymi. Uwzględniając fakt, że sposób zamocowania elementu do chwytaka ogranicza jego ruch, postanowiono przeprowadzić obliczenia jak dla zginania utwierdzonej belki (rys. 2.3).



Rysunek 2.3. Uproszczony model wytrzymałościowy (źródło – opracowanie własne)

Obliczenia:

$$\tau_t = \frac{P}{A} = \frac{22{,}5\ N}{18\ mm * 34\ mm} = 0{,}037\ MPa \tag{1}$$

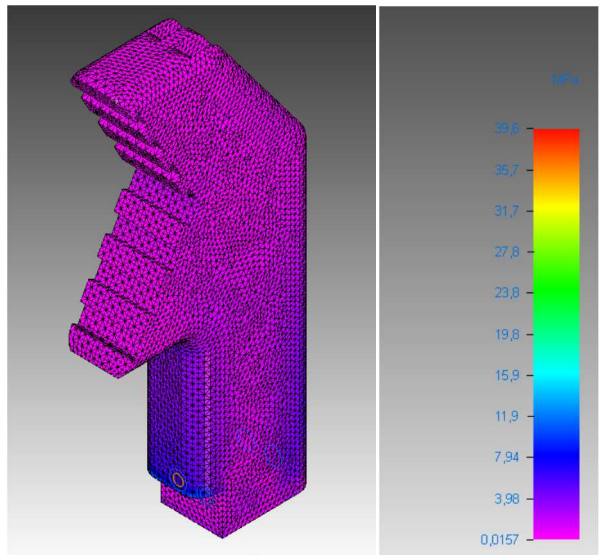
$$\sigma_g = \frac{M_g}{W_x} = \frac{P*l}{\frac{b*h^2}{6}} = \frac{22{,}5\ N * 122\ mm}{\frac{18\ mm*(34\ mm)^2}{6}} = 0{,}564\ MPa \tag{2}$$

$$\sigma_z = \sqrt{{\sigma_g}^2 + 3 * {\tau_t}^2} = \sqrt{0{,}564^2 + 3 * 0{,}037^2} = 0{,}568\ MPa \tag{3}$$

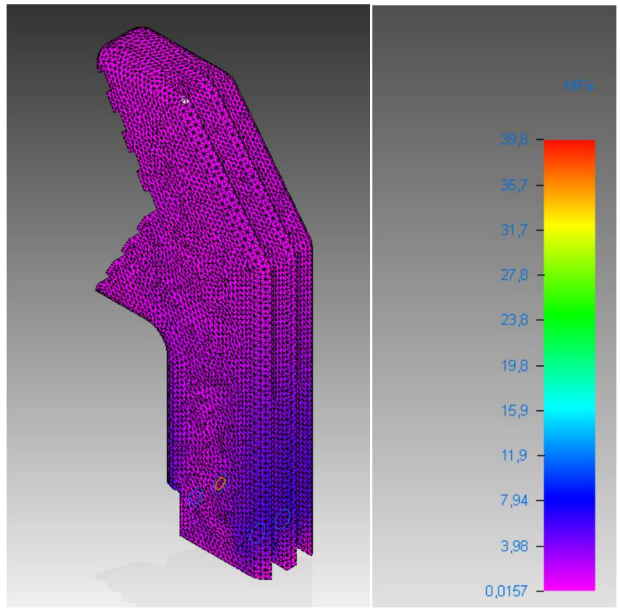
Model rzeczywisty wykonano z materiału P430 firmy ABS Plus. Materiał ten jest termoplastycznym tworzywem polimerowym wykorzystywanym w technologii FDM. Według producenta charakteryzuje się on wytrzymałością na zginanie na zakresie 35-58 MPa w zależności od kierunku zginania. Za współczynnik wytrzymałości na zginanie przyjęto dolną wartość, wówczas:

$$k_g = 35\ MPa \gg \sigma_z = 0{,}568\ MPa \tag{4}$$

Z obliczeń wynika jednoznacznie, że zaprojektowana część przeniesie zadane jej obciążenie. Po metodzie analitycznej, przeprowadzono analizę numeryczną za pomocą MES w celu otrzymania mapy rozkładu naprężeń i odkształceń. Program Solid Edge posiada wbudowany moduł numeryczny, zatem po wprowadzeniu geometrii modelu oraz odpowiednich parametrów (zamocowanie, obciążenia, materiał badanego elementu) otrzymano odpowiednie rozkłady (rys. 2.4 i 2.5).



Rysunek 2.4. Wyniki analizy MES (źródło – opracowanie własne)

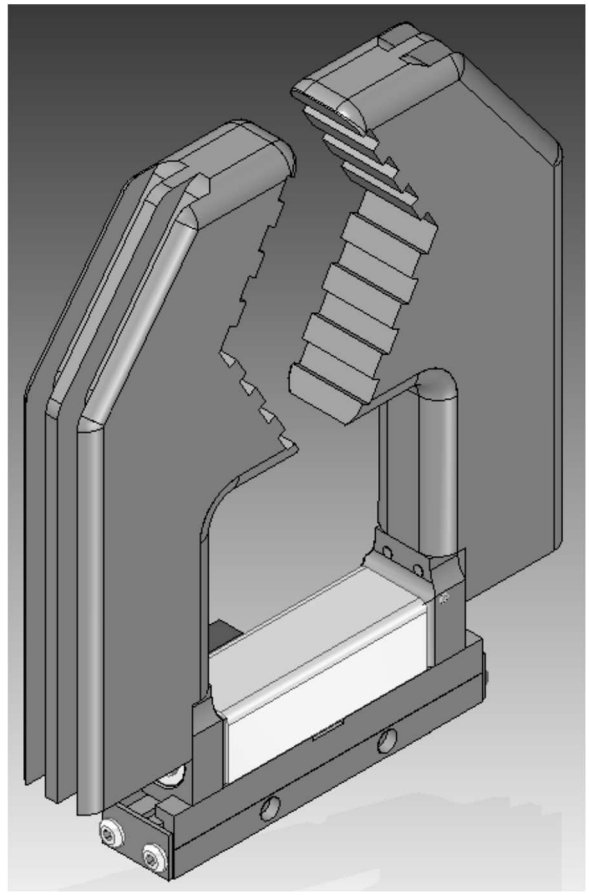


Rysunek 2.5. Wyniki analizy MES (źródło – opracowanie własne)

2.3. Złożenie

Po wykonaniu analiz i obliczeń jednoznacznie stwierdzono, że zaprojektowany model części przeniesie założone obciążenia występujące w czasie pracy. Następnym etapem całego procesu było sprawdzenie zgodności elementu poprzez wykonanie złożenia z modelem 3D chwytaka KGG60-40, który pozyskano ze strony producenta.

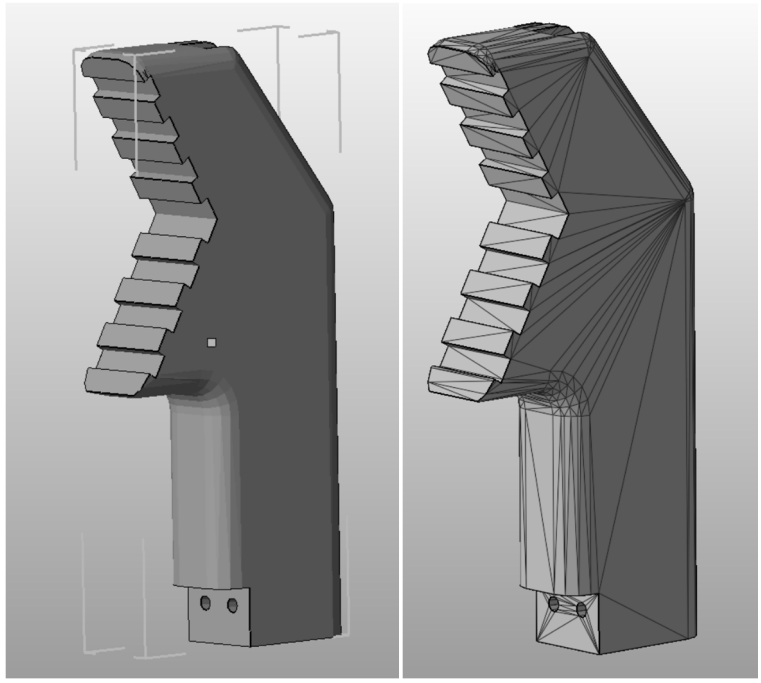
Gotowe złożenie całego mechanizmu pokazuje, że część została zaprojektowana poprawnie i pasuje do gotowego chwytaka. Żaden z elementów nie nachodzi na siebie oraz zachowany został zaplanowany odstęp między ramionami chwytnymi.



Rysunek 2.6. Złożenie chwytaka (źródło – opracowanie własne)

2.4. Model STL i jego dokładność

Następnym etapem był zapis modelu w formacie STL. Dokładność wygenerowanej części, oraz uproszczeń wynikających z przekonwertowania pliku na format STL zweryfikowano za pomocą programu Netfabb. Program ten umożliwia również pokazanie efektu teselacji, czyli odwzorowania geometrii powierzchni za pomocą siatki trójkątów.

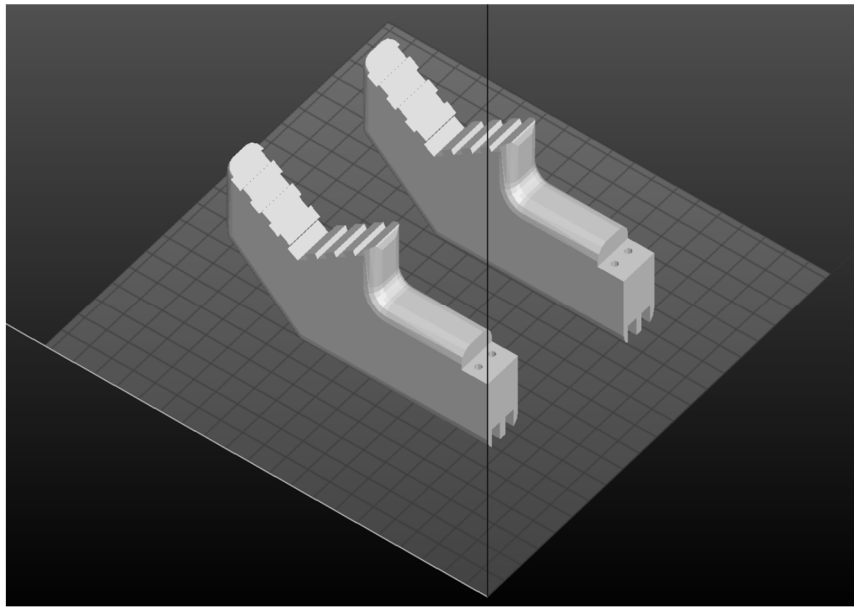


Rysunek 2.7. Model chwytaka w formacie STL (źródło – opracowanie własne)

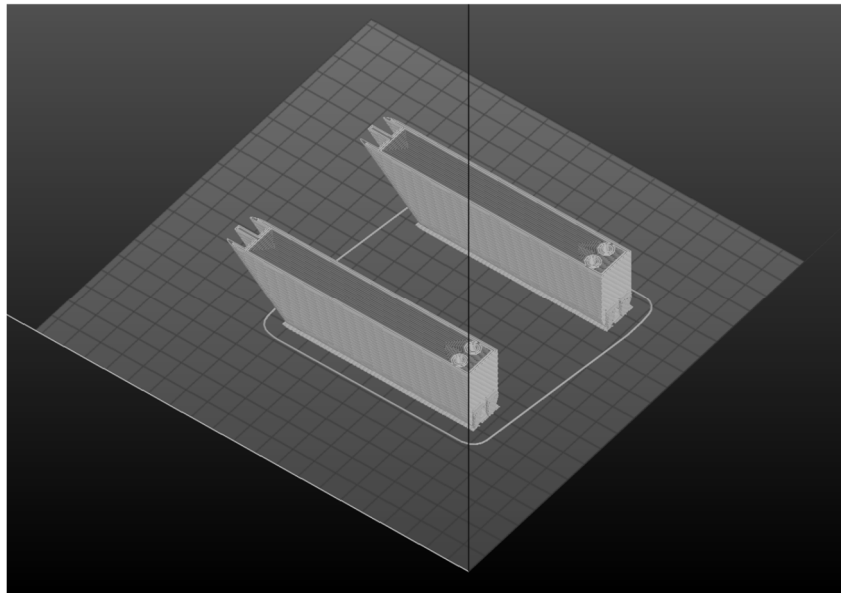
Program Netfabb nie wykazał żadnych nieprawidłowości w modelu. Geometria powierzchni modelu jest zamknięta i nie wymaga naprawy.

2.5. Podział na warstwy i generowanie G-codu

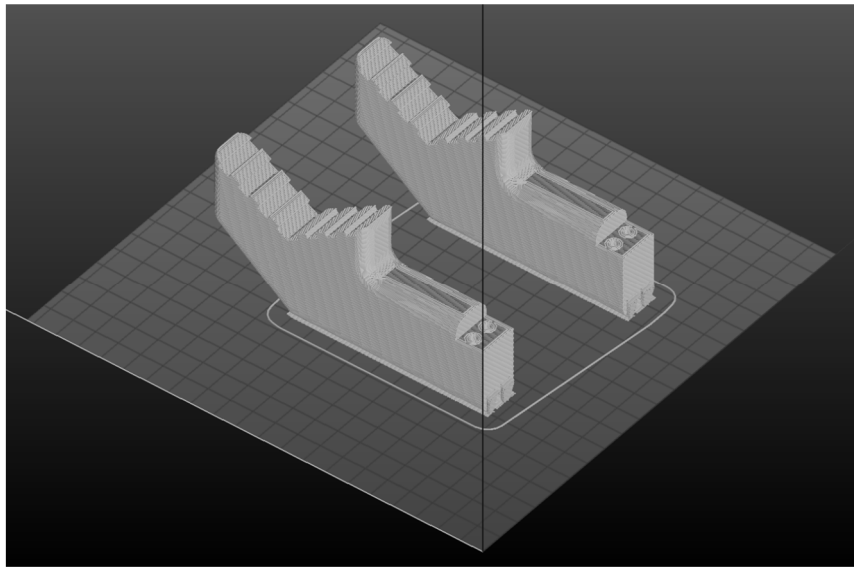
Ostatnim etapem modelowania numerycznego był podział modelu na przekroje. Jak już wcześniej wspomniano, do podziału elementu na warstwy i wygenerowania G-Codu użyto darmowego oprogramowania Slic3r. Do programu wprowadzono parametry urządzenia, którego zamierzano użyć do wytworzenia części tj. wymiary platformy, grubość warstwy, szybkość i temperaturę druku.



Rysunek 2.8. Umiejscowienie elementów chwytnych na platformie drukarki (źródło – opracowanie własne)



Rysunek 2.9. Przekrój warstwy modelu (źródło – opracowanie własne)



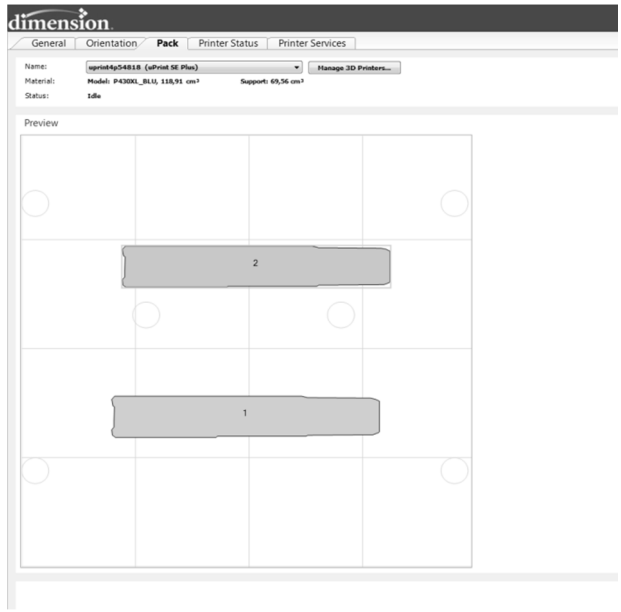
Rysunek 2.10. Modele chwytaka z zaznaczonymi warstwami materiału (źródło – opracowanie własne)

Po ułożeniu modeli na platformie i sprawdzeniu poprawności warstw wygenerowano plik z G-codem do sterowania głowicą drukarki 3D.

2.6. Wykonanie rzeczywistych modeli

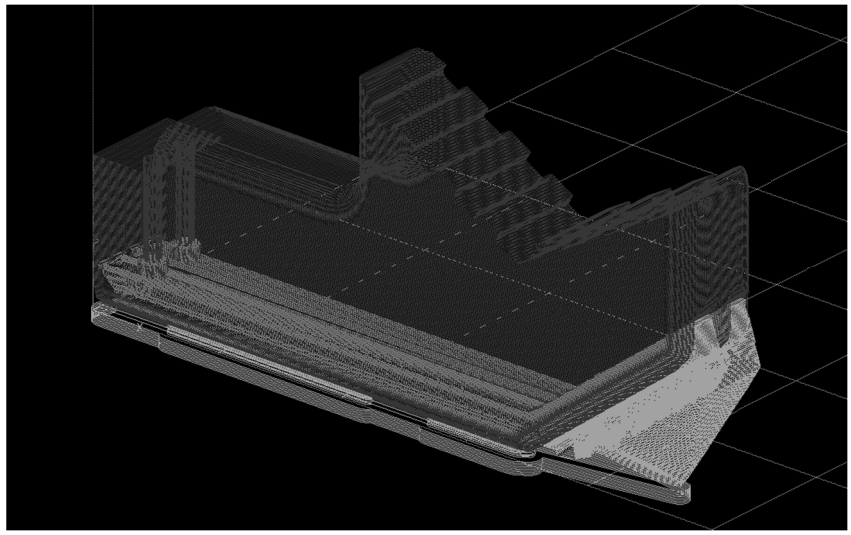
Po sprawdzeniu poprawności konstrukcji, następnie sprawdzono, czy nie występują żadne nieprawidłowości w strukturze modelu, oraz czy jest on stworzony poprawnie i część zostanie wydrukowana w zakładany sposób. Po tym procesie wygenerowano plik z G-codem do sterowania głowicą urządzenia.

Urządzenie RP wykorzystane do wytworzenia elementów posiada dedykowane przez producenta oprogramowanie sterujące CatalystEX. Program ten umożliwia automatyczne czytanie parametrów urządzenia. Wyświetlone zostaje również pole robocze, na którym można określić położenie części do druku.



Rysunek 2.11. Ułożenie części w programie CatalystEX (źródło – opracowanie własne)

Oprogramowanie zawiera funkcję wygenerowania ścieżek głowicy urządzenia tworzących poszczególne warstwy przedmiotu wraz z określeniem ilości oraz warstw materiału wsporczego.



Rysunek 2.12. Ustawienie elementu w przestrzeni roboczej z zaznaczeniem ścieżek głowicy (źródło – opracowanie własne)

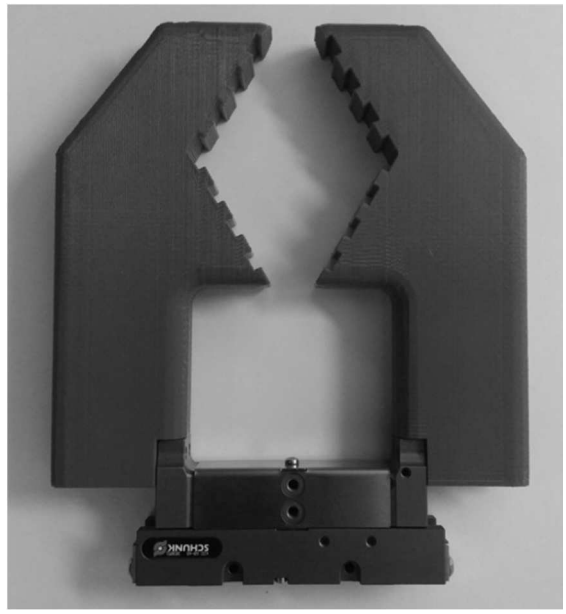
Po odpowiednim ułożeniu części w przestrzeni roboczej urządzenia można uruchomić proces drukowania, program samodzielnie skomunikuje się z urządzeniem przesyłając kod sterujący. Przewidywany czas tworzenia zaprojektowanej części wyniósł siedem godzin.

3. Wyniki procesu

Proces budowy elementów zakończono po siedmiu godzinach. Części zostały wykonane prawidłowo, a sam proces przebiegał bez problemów. Wynikiem procesu były dwa ramiona chwytne do chwytaka pneumatycznego KGG 60-40. Elementy te razem z chwytakiem będą stanowiły kiść chwytną robota Kawasaki RS03N. Na rys. 3.1 oraz 3.2 przedstawiono otrzymane produkty, natomiast na rys. 3.3 złożenie kompletnego chwytaka.

Rysunek 3.1. Zdjęcie gotowego elementu (źródło – opracowanie własne)

Rysunek 3.2. Zdjęcie gotowego elementu (źródło – opracowanie własne)



Rysunek 3.3. Złożenie z chwytakiem KGG 60-40 (źródło – opracowanie własne)

4. Podsumowanie i wnioski

Rapid Prototyping są to metody szybkiego wytwarzania części składowych, prototypów maszyn i gotowych mechanizmów. Pozwalają one na wytworzenie gotowego wyrobu o właściwościach użytkowych i estetycznych z pominięciem tradycyjnych technologii mechanicznych. Wszystkie technologie Szybkiego Prototypowania opierają się na jednej podstawowej zasadzie. Bazują na odpowiednio przetworzonych modelach 3D. Po uproszczeniu geometrii, oraz późniejszym transformacjom modelu, można w prosty sposób z obiektu wirtualnego wytworzyć fizyczną część. Dzięki temu, w sposób czytelny istnieje możliwość zaprezentowania projektu osobom nie mającym doświadczenia w analizie modeli generowanych za pomocą programów CAD.

Generacyjne metody oparte na warstwowej budowie części pozwalają na tworzenie obiektów o dużym stopniu złożoności. Poprzez punktowe utwardzanie warstw materiału można w krótkim czasie wytworzyć model o praktycznie dowolnym kształcie. Istotne jest to, że nie zawsze jest on do uzyskania w konwencjonalnej obróbce. Szeroka gama obrabianych materiałów, oraz ciągły rozwój metod RP pozwalają na wykonywanie pełnowartościowych części bez stosowania specjalnych narzędzi i form nieodbiegających od tych wytworzonych za pomocą obróbki ubytkowej. Zapewnia to małe koszty wykonania elementów przy produkcji jednostkowej i małoseryjnej. Dzięki zastosowaniu RP można skrócić cykl

rozwojowy produktu, oraz zmniejszyć nakłady na jego opracowanie, zmniejszając przy tym ryzyko inwestycyjne związane z wprowadzaniem nowego produktu na rynek.

Jedną z wad RP są ograniczone wymiary budowanych części. Obiekty większe od pola roboczego urządzenia należy złożyć z poszczególnych części, które zostaną później połączone. Wyklucza to jednak zachowanie pełnych właściwości wytrzymałościowych gotowego elementu. Kolejną z wad jest dokładność wykonania części, która zależy od parametrów urządzenia. Wysokość przekroju zależy tutaj od grubości warstwy materiału. Natomiast jakość powierzchni uzależniona jest przede wszystkim od wyboru metody. Często technologie o wysokiej jakości wykonania nie zapewniają wymaganej wytrzymałości elementu, należy więc wybrać wtedy technologię o niższej dokładności.

Powoduje to, że potrzebna jest dodatkowa obróbka wygładzająca. Sam proces budowy części metodami RP uchodzi za dość powolny, dlatego nie są one stosowane w produkcji masowej.

W przypadku prezentowanych w pracy elementów chwytowych za pomocą technologii Rapid Prototyping w prosty sposób można przejść z zaprojektowanego modelu do w pełni funkcjonalnego chwytaka. Znalazła tu zastosowanie opłacalność wykorzystania RP do produkcji jednostkowej. Dzięki komputerowym narzędziom wspomagającym projektowanie, można na podstawie posiadanych informacji zaprojektować model końcówki. Później w module numerycznym MES sprawdzić jego wytrzymałość na obciążenia, które mogą wystąpić podczas rzeczywistej pracy. Wykorzystanie metod RP niesie ze sobą pewne uproszczenia stworzonego modelu, jednakże szeroka gama oprogramowania kontrolującego poprawność uproszczonej geometrii pozwala zapobiec uszkodzeniom modelu wynikającym z teselacji. Za pomocą oprogramowania komputerowego możliwa była wizualizacja całego procesu budowy części. Po wprowadzeniu parametrów urządzenia i wygenerowaniu zaprojektowanych elementów można sprawdzić każdą warstwę w celu uniknięcia niechcianych obiektów, czy nieciągłości modelu. Oprogramowanie wykorzystywane do produkcji gotowych części samo wygenerowało G-cod sterujący głowicą urządzenia, dzięki czemu maszyna zbudowała gotowe części.

Podsumowując, technologie Szybkiego Prototypowania pozwalają w znacznym stopniu przyśpieszyć proces projektowania części i mechanizmów maszyn. Dzięki nim można w prosty sposób wytworzyć poszczególne części, złożyć je i zaprezentować gotowy fizyczny prototyp urządzenia. Ciągły rozwój metod RP, a także wykorzystanie nowych materiałów, pozwala na jednostkową produkcję gotowych części, oraz mechanizmów o praktycznie dowolnym kształcie bez wykorzystania specjalistycznych narzędzi. Niestety technologia ta wiąże się z pewnymi ograniczeniami. Podstawowym jest wielkość tworzonego elementu, gdyż nie można jednorazowo wytworzyć przedmiotu większego od pola roboczego urządzenia. Kolejnymi ograniczeniami są dokładność wykonania, oraz jakość

powierzchni, jednak dzięki zastosowaniu dodatkowej obróbki istnieje możliwość zniwelowania ich wpływu na przedmiot.

Literatura

1. Chlebus E., *Innowacyjne technologie rapidprototyping-rapidtooling w rozwoju produktu,* Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław, 2003.
2. Chlebus E., *Techniki komputerowe CAx w inżynierii produkcji,* Wydawnictwo Naukowo-Techniczne, Warszawa, 2000.
3. Rudnicki Z., *Nowoczesne techniki przyśpieszające wytwarzanie,* Wykład, AGH w Krakowie, http://www.kkiem.agh.edu.pl/dydakt/Wyklady/RapidProt12.pdf – grudzień 2015.
4. Dyrda J., Dyrda R., *O metodach RapidPrototyping słów kilka*, FORUM NARZĘDZIOWE OBERON, Nr 3/2010, s. 42-45.
5. Kulawiuk A., Penkała P., Sobaszek Ł., *Komputerowe wspomaganie zabiegów alloplastycznych na przykładzie endoprotezoplastyki stawu kolanowego*, Informatyczne systemy zarządzania, t. 6, Wydawnictwo Uczelniane Politechniki Koszalińskiej, 2015, s. 41-56.
6. Walczak M., Gąska D., Guzik M., *Characteristics of Products Made of 17-4PH Steel by means of 3D Printing Method*, Applied Computer Science, vol. 12, no. 3, pp. 29-36.
7. Zubrzycki J., Smidova N., *Computer-aided design of human knee implant,* W: Industrial and service robotics: 13th International Conference on Industrial, Service and Humanoid Robotics (ROBTEP): Conference [WOS]; [Red:] Hajduk Mikulas, Koukolová Lucia- Stafa-Zurich; Switzerland: Trans Tech Publications, 2014, s. 172-181 .- (Applied Mechanics and Materials, ISSN 1660-9336; nr 913)
8. Instrukcja obsługi drukarki 3D uPrint SE Plus (dołączona do zestawu).
9. http://gadzetomania.pl/14355,drukarki-3d-skad-sie-wlasciwie-wziely – grudzień 2015.
10. http://www.schunk.com/schunk_files/attachments/KGG_gesamt_EN.pdf – grudzień 2015.

Streszczenie:

Technologie Rapid Prototyping (RP) obejmują wiele metod szybkiego wytwarzania modeli fizycznych – prototypów lub części maszyn – z pominięciem tradycyjnych technologii wytwarzania. Odlewanie, obróbka ubytkowa czy elektroerozyjna zastępowane są zaawansowanymi technikami RP. Obiekty rzeczywiste uzyskane za pomocą technologii Rapid Prototyping znajdują zastosowanie podczas prezentacji wyrobów potencjalnym inwestorom, w badaniach opinii klienta, analizie gotowego wyrobu, a także mogą być w pełni funkcjonalnymi przedmiotami.

Słowa kluczowe: Rapid Prototyping, modelowanie 3D, drukowanie 3D, FDM

Abstract:

The purpose of the article is to discuss Rapid Prototyping techniques in machine parts designing. The article presents the authorial RP process of industrial robots gripping elements designing. First of all, the idea of Rapid Prototyping and typical RP techniques are described. Moreover, the designing process, the strength analysis and modeled parts printing process were outlined. In the final part of the paper the authors discussed the results of works.

Keywords: Rapid Prototyping, 3D modeling, 3D printing, FDM

Daniel Czyczyn-Egird
Rafał Wojszczyk
Katedra Inżynierii Komputerowej
Wydział Elektroniki i Informatyki
Politechnika Koszalińska
ul. J.J. Śniadeckich 2
75-453 Koszalin

Zastosowanie technik eksploracji danych na przykładzie badania popularności wzorców projektowych w serwisie społecznościowym Stackoverflow.com

Słowa kluczowe: sieci społecznościowe, eksploracja danych, wzorce projektowe

1. Wstęp

Od wieków ludzie poszukiwali sposobów komunikacji na odległość. Każdy krok postępu technologicznego, często związany z wieloma dziedzinami nauki, sprzyjał łatwiejszej wymianie danych i informacji. Sposoby komunikacji zmieniały się z biegiem lat, pierwsze urządzenia elektromechaniczne np. telegraf umożliwiały przesyłanie niewielkiej ilości danych na odległość. Kolejne lata przyniosły wzrost przepustowości i zakresu przesyłanych danych, włączając w to dźwięk i wizję, aż po współczesny Internet. Podobna historia ewolucji spotkała pionierów, którzy dali początek współczesnemu programowaniu. Niewątpliwie pierwszym sposobem komunikacji na odległość pomiędzy praktykami programowania było zwyczajne pisanie listów, ewentualnie wymiana ówczesnych nośników danych. Dopiero dalsze lata i pojawienie się Internetu pozwoliło na komunikację: poprzez pocztę email, grupy usenet oraz publikowanie informacji na stronach internetowych (w przeszłości na stronach domowych, współcześnie popularne są blogi). Wymienione sposoby komunikacji nadal dotyczą wymiany danych, pomijając przy tym szybkość i dostępność łączy internetowych, warto zastanowić się nad sposobem wykorzystania – już nie danych a informacji. W tym przypadku kierunkiem rozwoju wykorzystania Internetu jest tzw. Web 2.0 [12]. Popularne serwisy społecznościowe bazują na założeniach Web 2.0 i znacząco wpływają na szybkość i skalę rozpowszechnianych informacji. Angażując do tego podstawowe założenia sieci semantycznych (poprzez oznaczanie informacji metadanymi

w postaci tagów) można otrzymać połączenie dające więcej niż dostęp do informacji, wynikiem tego połączenia jest wiedza pozyskiwana z eksploracji danych.

Współczesne techniki informatyczne nie istniałby gdyby nie rozwój rzemiosła programowania. Programowanie nieodłącznie wiąże się z korzystaniem z różnych wzorców: architektonicznych (łączą różne warstwy aplikacji), projektowych (obejmują zakres poziomu klas), implementacyjnych (nazywane również idiomami, występują na poziomie wierszy kodu). Wzorce projektowe przedstawione w [6] powstały na podstawie doświadczenia społeczeństwa, które wówczas zajmowało się zagadnieniami programowania obiektowego i dobrych praktyk. Zasadniczym przeznaczeniem wzorców jest rozwiązywanie problemów projektowania i programowania obiektowego [14]. Wzorce projektowe są uznawane również za pewnego rodzaju język komunikacji. Język ten sprawia, że kod programu, który zawiera wzorce będzie łatwiejszy w poznaniu i zrozumieniu niż ten, który ich nie zawiera. Zatem osoby znające wzorce samoistnie tworzą społeczeństwo porozumiewające się językiem wzorców [1].

Celem artykułu jest zbadanie popularności oraz tendencji zmian wykorzystania wzorców projektowych, w oparciu o wyspecjalizowane sieci społecznościowe. Wyniki badania będą wykorzystane do wskazania kierunku dalszych prac nad jakością implementacji wzorców projektowych, natomiast metodyka przeprowadzonych badań przyczyni się do wybrania kierunku prac w dziedzinie data miningu.

W drugim rozdziale artykułu przedstawiono przegląd wybranych zagadnień związanych z wykorzystaniem sieci komputerowych do wspierania pracy programistów oraz opisu informacji o wzorcach projektowych. Trzeci rozdział zawiera opis środowiska badawczego oraz uzasadnienia dokonanych wyborów. W rozdziale czwartym przedstawiono wyniki badań oraz stosowne wnioski. Ostatni, piąty rozdział, stanowi podsumowanie pracy.

2. Przegląd rozwiązań sieciowych

2.1. Ogólne wsparcie deweloperów

W dzisiejszych czasach do dyspozycji deweloperów (programistów) oddawane są cały czas nowe platformy z dokumentacją produktowo-techniczną, które charakteryzują się obszerną bazą informacji przydatnych podczas projektowania nowych rozwiązań informatycznych. Wiele firm dostarczających własne rozwiązania informatyczne oraz produkty świadomie i celowo stara się udostępnić wszystkim zainteresowanym dobrze udokumentowane biblioteki czy też interfejsy programistyczne. Takie działanie pozwala osiągnąć pewne korzyści, dokładniej zapewnia produktom odpowiednią reklamę oraz szerokie rozpowszechnienie na rynku, chociażby przez zapewnienie swoim odbiorcom możliwości tworzenia

spersonalizowanych dodatków czy nowych funkcji do już istniejących systemów komputerowych.

W XXI wieku ilość dokumentacji w formie papierowej drastycznie maleje na rzecz dokumentacji online, która dzięki powszechnemu dostępowi do globalnej sieci internetowej, jest rozpowszechniona na cały świat i dostępna przez całą dobę.

Deweloperzy wyposażeni w dostęp do globalnej sieci internetowej są w stanie w łatwy sposób dotrzeć do opisów technicznych, obliczeń konstrukcyjnych, planów, rysunków, harmonogramów i kosztorysów. Dzięki powszechnemu dostępowi do dokumentacji mogą powstawać nowe rozwiązania, bez potrzeby bezpośredniego kontaktu dewelopera z firmą udostępniającą produkt.

Dodatkowym elementem wspierającym pracę deweloperów są sieci społecznościowe, które pozwalają na wspólne analizowanie dokumentacji technicznych a następnie zaproponowanie nowych rozwiązań. Gotowe półprodukty, pakiety oraz biblioteki, udostępnione są w sieciowych repozytoriach, które łączą deweloperów poprzez wymianę zaufanych fragmentów oprogramowania. Jednym z konkretnych narzędzi, o którym warto wspomnieć w kontekście pracy z repozytoriami tego typu, jest NuGet. Jest to tak zwany menedżer pakietów lub też systemem zarządzania pakietami, dla platformy deweloperskiej firmy Microsoft, w tym platformy .NET. Narzędzia klienckie NuGet'a umożliwiają wygodne korzystanie z pakietów lub też przygotowanie własnych, a całość w łatwy sposób integruje się ze środowiskiem programistycznym Visual Studio. Zawartość pakietów zawartych w repozytorium NuGet'a jest stale aktualizowana przez autorów pakietów zrzeszonych w sieci społecznościowej. Podobnymi rozwiązaniami dostępnymi na rynku są na przykład: GetIt dla narzędzi deweloperskich z rodziny Delphi, czy Maven dla Javy.

Sieciami społecznościowymi możemy także określić takie platformy jak Github, CodePlex czy też SourceForge, które są portalami gromadzącymi oraz przechowującymi projekty informatyczne. Udostępniają one darmowy hosting programów open source oraz płatne prywatne repozytoria. Portale tego typu służą deweloperom także do wymiany informacji oraz kodów źródłowych, przyczyniając się tym samym do rozwoju oprogramowania, zwłaszcza na licencji otwartej i bezpłatnej.

Społeczność połączona w sieć przyczyniła się także do rozwoju i spopularyzowania repozytoriów plikowych dostępnych online, jako systemy kontroli wersji. Takimi narzędziami są przykładowo Git oraz SVN, których przeznaczeniem jest m.in. śledzenie zmian w kodzie źródłowym oraz zapewnienie pomocy programistom w łączeniu różnych wersji plików, edytowanych przez wielu deweloperów w różnych momentach czasowych. Systemy kontroli wersji można podzielić na:

- lokalne, umożliwiające zapisanie danych wyłącznie na lokalnym komputerze (np. SCCS oraz RCS),
- scentralizowane, cechujące się architekturą klient-serwer (np. CVS, SVN),
- rozproszone, oparte na architekturze peer-to-peer (np. BitKeeper, Git).

Pierwszy rodzaj systemów zapisuje jedynie wersje plików na lokalnym komputerze. Jest to bardzo wygodne, jednak mało bezpieczne rozwiązanie (w sensie ochrony przed utratą danych) oraz blokujące możliwość dzielenia się swoimi kodami z innymi deweloperami. W przypadku rozwiązań scentralizowanych istnieje jedno centralne repozytorium, za pomocą którego wszyscy użytkownicy systemu synchronizują swoje zmiany. Rozwiązania rozproszone pozwalają na równoczesne prowadzenie niezależnych, ale też równoprawnych gałęzi, które można wzajemnie synchronizować, np. poprzez pocztę elektroniczną. Typ wykorzystywanego systemu kontroli wersji zależy od założeń projektu i powinien być dopasowany do potrzeb deweloperów.

Doskonałym wykorzystaniem sieci społecznościowych są także fora internetowe zrzeszające użytkowników, których zainteresowania pokrywają się z tematyką owych portali. Przykładowo, największe w Polsce forum programistyczne 4programmers.net przyciąga programistów, administratorów, webmasterów, słowem – ludzi związanych z branżą IT. Tysiące tematów i komentarzy stanowią o sile tego typu portali, gdzie użytkownicy mogą zakładać nowe tematy a zainteresowani mogą pisać nowe komentarze dotyczące zadanej sprawy. Wraz z upływem czasu pojawiło się zapotrzebowanie na portale jeszcze bardziej ściśle techniczne, gdzie dzielenie się wiedzą przez specjalistów opierało się na schemacie Q&A, czyli „pytanie-odpowiedź". Jednym z takich portali jest polski serwis devpytania.pl, który umożliwia zadawanie pytań czy też udzielanie odpowiedzi do istniejących pytań. Obecnie w bazie portalu jest blisko 4000 pytań. Zdecydowanie lepiej jest skorzystać z portalu o zasięgu światowym jakim jest serwis stackoverflow.com. Baza pytań tego serwisu jest imponująca i sięga blisko 11 milionom, a także odpowiedzi do pytań udzielane są przez specjalistów z całego świata. Serwisy tego typu oferują pomoc w rozwiązywaniu problemów, które są napotykane przez deweloperów każdego dnia, dlatego ich rosnąca popularność nie powinna dziwić. Niniejsza publikacja skupia się także na analizie i eksploracji danych [4] z wyżej wspomnianego portalu. Dokładniej na przebadaniu popularności wzorców projektowych w odniesieniu do wiedzy zgromadzonej w bazie danych serwisu, który w 2008 roku zaistniał w Internecie. Do tej pory serwis zbudował ogromną sieć społecznościową, która przyczynia się do ciągłego rozwoju portalu – na chwilę obecną pytających i odpowiadających użytkowników jest razem około 4,8 miliona.

2.2. Informacje o wzorcach projektowych

Podstawową literaturą dotyczącą wzorców projektowych jest wymienione wcześniej [6]. Na katalogu wzorców [6] bazują często inni autorzy proponując inne warianty lub implementacje dopasowane do konkretnego języka, np. [11], [3]. Forma opisu wzorców projektowych przedstawiona w wymienionej literaturze jest z założenia przeznaczona do nauki ich wykorzystania, zawiera: opis słowny w języku naturalnym, diagramy klas oraz przykładowy kod implementacji oparty o proste przykłady. Podobny cel jest realizowany przez repozytorium wiedzy o wzorcach projektowych przedstawione w [9]. Celem tego repozytorium jest szerzenie wiedzy o wzorcach, nauka oraz wsparcie w ich implementacji. Autorska aplikacja internetowa z [9] umożliwia dialog pomiędzy użytkownikiem a systemem, na zasadzie pytanie-odpowiedź. Wiedza o wzorcach oraz pytania są wstępnie predefiniowane z możliwością dodania nowych zasobów.

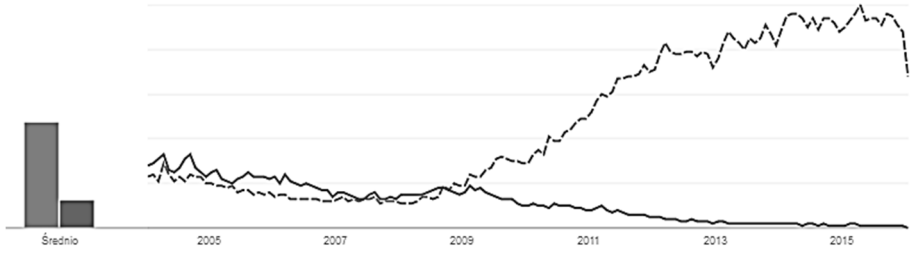
Niektóre z formalnych sposobów reprezentacji wzorców projektowych opierają się o sieci semantyczne, dokładniej wykorzystują ontologie [8], [2], [5]. W filozofii otologia opisuje byty świata rzeczywistego [10], natomiast w informatyce jest wykorzystywana do formalnej reprezentacji wiedzy w postaci zbioru pojęć z danej dziedziny i relacji między tymi pojęciami. Bardzo ważną zaletą ontologii jest możliwość transformacji danych „z" oraz „do" różnych źródeł. W [2] oraz [5] zostały zaproponowane ontologie opisujące podstawową wiedzę o wzorcach projektowych. Wiedza ta może być przetransformowana do innych podejść, np. opisane wcześniej [9] czy [8] służącego do wyszukiwania wystąpień wzorców projektowych w kodzie źródłowym.

3. Przygotowania do badań

3.1. Portal stackoverflow.com

Serwis stackoverflow.com jest jednym z największych portali typu Q&A z zakresu szeroko pojętych tematów związanych z inżynierią oprogramowania. Jest to platforma pozwalająca użytkownikom zadawać pytania i uzyskiwać na nie odpowiedzi. Społeczność zgromadzona na portalu ma możliwość oceniania „w dół" lub „w górę" zadawanych pytań i udzielanych odpowiedzi, jest to rodzaj naturalnej selekcji, w której odrzucane są „złe" rozwiązania a wyróżniane „dobre". Użytkownicy oddają głosy na podstawie posiadanego stanu wiedzy i preferencji, dlatego należy pamiętać, że te wybory mogą nie być obiektywne. Jednakże głos społeczeństwa jest bardzo dobrym wyznacznikiem popularności, co zostało wykorzystane w dalszych badaniach. Użytkownicy za swoją działalność są odpowiednio nagradzani punktami lub odznakami, które służą do wewnętrznej, co ważne – pozytywnej rywalizacji między nimi. Cała zawartość wygenerowana przez użytkowników jest oparta na licencji Creative Commons.

Wybór serwisu stackoverflow.com jako źródła informacji jest uzasadniony popularnością zmierzoną przez Google Trends oraz wykorzystaniem w innych badaniach [13]. Rysunek 1 przedstawia wykres popularności stackoverflow.com w stosunku do bezpośredniego konkurenta experts-exchange.com.



Rys. 1. Wykres popularności konkurencyjnych serwisów, linia niebieska to popularność zapytań stackoverflow.com, czerwona experts-exchange.com, stan na 06.01.2016.

3.2. Wzorce projektowe jako przedmiot badań

Wzorce projektowe należy traktować jako uniwersalne i sprawdzone w praktyce rozwiązania często pojawiających się problemów projektowych. Pokazują występujące powiązania oraz zależności klas i obiektów, ułatwiają również tworzenie, modyfikację i utrzymanie kodów źródłowych systemów informatycznych. Stosowanie wzorców w małych projektach zazwyczaj przyczynia się zwiększenia pracochłonności procesu wytwórczego. Natomiast w rozbudowanych projektach poprawia ogólną jakość tworzonych systemów oraz ułatwia konserwację. Wnioskując na podstawie korzyści wynikających z wykorzystania wzorców projektowych [15] należałoby założyć, że popularność wzorców powinna cały czas wzrastać. W dalszej części publikacji, zostały przedstawione wyniki badań nad popularnością wzorców projektowych wypromowanych przez tzw. Bandę Czworga (Erich Gamma, Richard Helm, Ralph Johnson oraz John Vlissides) [6]. Badanie zostało zrealizowane poprzez analizę tematów zebranych w portalu stakcoverflow.com, na podstawie wystąpień słów kluczowych związanych ze wzorcami projektowymi, z wykorzystaniem technik data mining.

3.3. Przebieg badań

Badania zostały przeprowadzone na trzech odrębnych zbiorach danych, które zostały przygotowane przy pomocy autorskiej aplikacji pobierającej informacje. Następnie zbiory zostały odpowiednio przetworzone, w celu uzyskania konkretnych odpowiedzi oraz reprezentacji ich za pomocą określonych miar.

W celu wydobycia konkretnych danych z portalu stackoverflow.com został wykorzystany webowy interfejs programistyczny StackExchange API w najnowszej wersji 2.2. Za pomocą odpowiednich metod interfejsu możliwe było budowanie

sparametryzowanych zapytań do serwera, a następnie uzyskiwanie konkretnych odpowiedzi spełniających zadane kryteria. Rysunek 2 przedstawia przykładowe zapytanie interfejsu, w tym wypadku poszukiwanie informacji dotyczącej wzorca projektowego „singleton".

Rys. 2. Przykładowe zapytanie interfejsu API

Wszystkie odpowiedzi zwrotne były otrzymywane w formacie JSON [7]. Format ten świetnie nadaje się do wymiany danych pomiędzy usługami programistycznymi, ponieważ jest stosunkowo lekki oraz łatwy do dalszego przetwarzania i obróbki. Rysunek 3 przedstawia przykładowe dane wyjściowe programu w formacie JSON.

Rys. 3. Przykładowa odpowiedź serwera w formacie JSON

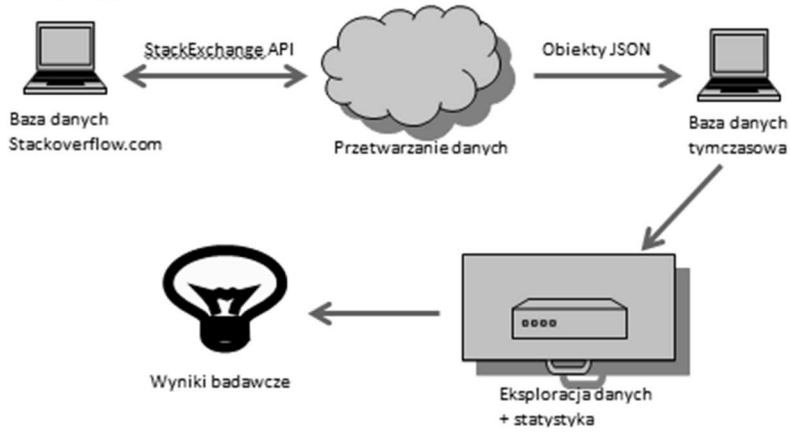
Dane, przetwarzane za pomocą autorskiego oprogramowania, zostały umieszczone jako rekordy w bazie danych, rysunek 4 przedstawia schemat ideowy środowiska badawczego. Warto zauważyć, że API znacząco ułatwia dostęp do bazy informacji w portalu. Niestety posiada również pewne ograniczenia dla deweloperów. Serwis wykorzystuje walidację adresu IP tj. w ciągu jednej doby można wywołać około 300 zapytań, które zwrócą jednorazowo maksymalnie 100 obiektów zwrotnych. W przypadku chęci pobrania jednorazowym zapytaniem zbioru pytań dotyczących wybranego słowa kluczowego, możliwe jest uzyskanie maksymalnie 100 wyników. Samo preparowanie zapytania sprowadza się do odpowiedniego zarządzania parametrami oraz filtrami, np. możliwe jest odfiltrowanie danych po dacie utworzenia, wyrażeniach występujących w tytule pytań czy też jako słowa kluczowe, a także sortowanie według ilości odpowiedzi czy popularności.

Najpoważniejsza niedogodność dotycząca aktualnej wersji API jest związana z brakiem możliwości przeszukiwania bazy serwisu według wystąpienia zadanego

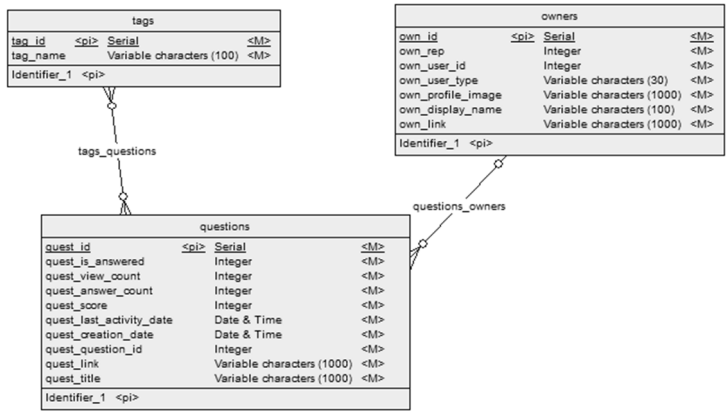
hasła jednocześnie w tytule oraz w treści pytania. Problem jest dość istotny, ponieważ wstępne rozeznanie pokazało, że niektórzy użytkownicy nie zawsze podają konkretne sformułowanie w tytule pytania lub jako słowo kluczowe (tag) – szukana fraza występuje dopiero w treści pytania, do którego nie ma obecnie dostępu z poziomu API. Rozwiązaniem tego problemu jest manualne wykorzystanie wyszukiwarki serwisu, która w wyszukiwaniu uwzględnia treść pytań. Niestety to rozwiązanie jest nie do przyjęcia w postawionym celu, dlatego w ten sposób zostało wykonane tylko jedno badanie. W pozostałych przypadkach wyszukiwanie odbywało się wyłącznie po obecności wybranej frazy w tytule pytania, gdyż okazało się, że wyszukiwanie po słowach kluczowych (tagach) było nieskuteczne – pomijało zbyt dużą ilość pytań.

Wszystkie uzyskane wyniki zostały przetworzone i umieszczone w relacyjnej bazie danych PostgreSQL 9.3, dzięki której możliwa jest łatwa eksploracja i klasyfikacja danych [4]. Rysunek 5 przedstawia model danych, który został zaprojektowany w taki sposób, aby zapewnić łatwy dostęp do danych oraz ich bezproblemowe zrozumienie.

Encje odpowiadają obiektom takim jak: pytania (question), autorzy pytań (owners) oraz słowa kluczowe (tags). Każda encja zawiera odpowiednie pola, które ją charakteryzują.



Rys. 4. Schemat środowiska badawczego

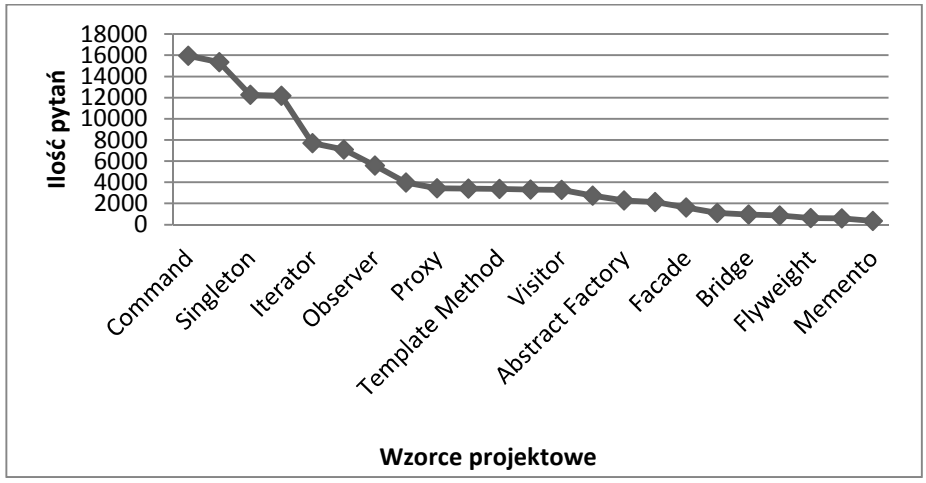


Rys. 5. Model encja-związek bazy danych dla obiektów z portalu stackoverflow.com

4. Wyniki badań

4.1. Ogólna popularność wzorców projektowych

Pierwszy zbiór danych został zasilony poprzez manualne zadanie pytań w wyszukiwarce serwisu. Uzyskany wynik, wyrażony w postaci wykresu liniowego, przedstawia rysunek 6.



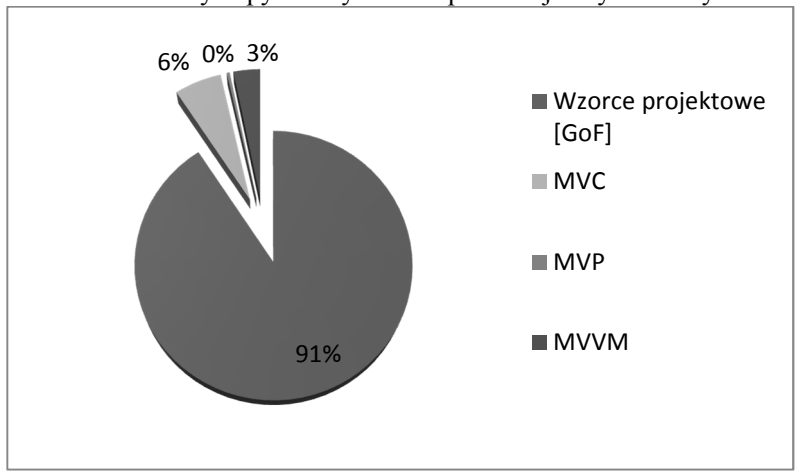
Rys. 6. Wykres liniowy pokazujący ilości pytań dla poszczególnych wzorców projektowych w serwisie stackoverflow.com, wraz z linią trendu (średnia ruchoma), stan na okres grudzień 2015

Analizując otrzymane wyniki można zauważyć dominującą grupę czterech wzorców: Command, State, Singleton oraz Factory. Badanie ogólnej popularności nie dostarcza dodatkowych informacji, które wynikałyby z kontekstu zadawanych pytań w serwisie stackoverflow.com. Wysoka popularność oznacza, że programiści podejmują próby implementacji danych wzorców projektowych, jednakże napotykają w tym problemy lub ich wiedza jest niewystarczająca. Niska popularność wskazuje na mniejszą potrzebę stosowania danych wzorców lub też łatwą implementację. Aczkolwiek zaprzecza temu popularność wzorca Singleton, który jest uznawany za jeden z najłatwiejszych. W celu doszczegółowienia wniosków zostały wykonane następne badania.

4.2. Względna popularność wzorców projektowych

Wzorce projektowe należą do tzw. dobrych praktyk, które często są tworzone na podstawie doświadczenia społeczeństwa. Toteż uzasadnione jest porównanie aktualnej popularności wzorców projektowych [6] względem podobnych dobrych praktyk, w tym przypadku do wzorców architektonicznych [9] takich jak MVC, MVP oraz MVVM.

Drugi zbiór danych składał się z 13000 pytań, które pojawiły się w grudniu 2015 roku. Pytania zostały pobrane z serwisu za pomocą API. Bazy danych została zasilona z pominięciem duplikatów, co poskutkowało zmniejszeniem rozmiaru zbioru do 12095 unikalnych pytań. Rysunek 7 prezentuje uzyskane wyniki.



Rys. 7. Diagram kołowy obrazujący rozkład pytań dla wzorców projektowych [6] oraz architektonicznych

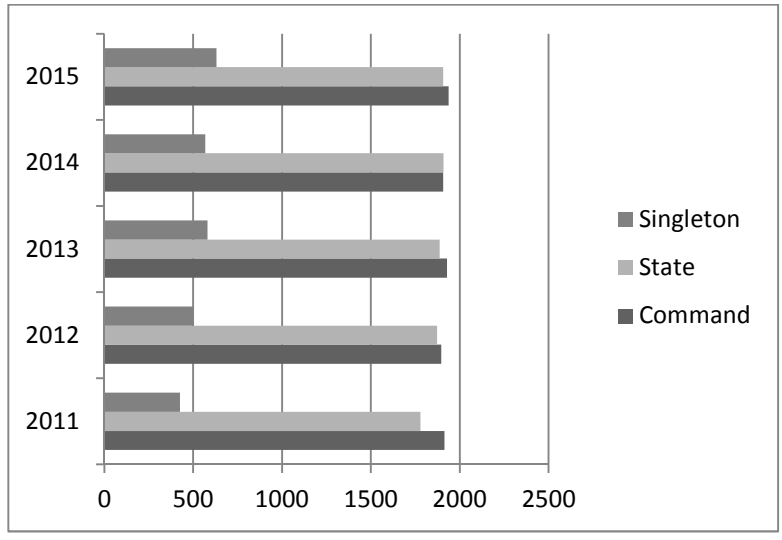
Otrzymany wynik wskazuje na znacząco większą popularność względem podobnych dobrych praktyk. Przyczyną tego może być fakt, że wzorce projektowe przy każdej implementacji wymagają zrozumienia oraz własnego wkładu od

programisty. Popularne środowiska programistyczne nie oferują automatycznej implementacji wzorców. Przeciwna sytuacja występuje wobec wzorców architektonicznych, w przypadku których zautomatyzowane narzędzia integrują się ze środowiskami programistycznymi, często niezauważalnie dla dewelopera. Zatem brak narzędzi automatyzujących implementację badanych wzorców wzmaga zainteresowanie tym tematem.

4.3. Popularność wzorców projektowych w funkcji czasu

Celem ostatniego badania opierającego się o miary popularności było wykazanie tendencji zmian dla wybranych wzorców w zadanej funkcji czasu.

Zbiór danych, przygotowany do tego badania, składał się z pytań pobranych dla trzech najpopularniejszych wzorców projektowych wykazanych we wcześniejszych badaniach, tj. dla: Command, State, Singleton. Dla każdego wzorca zostały pobrane pytania od początku roku 2011 aż do końca 2015 roku. Początkowa suma wynosiła zatem blisko 25000 pytań, jednak po odrzuceniu duplikatów pozostało 21640 pytań w bazie danych. Wyszukiwanie odbywało się ponownie według kryterium obecności wybranej frazy w tytule pytania. Uzyskane wyniki przedstawia rysunek 8.



Rys. 8. Popularność wybranych wzorców w funkcji czasu

Analiza uzyskanych wyników pozwala na wyciągnięcie następujących wniosków:

- wybrane do badania wzorce utrzymują stałą tendencję popularności, nieznaczne wahania wynikają z różnych czynników losowych, których występuje bardzo wiele w aktywnych społecznościach,

- stosunek popularności pomiędzy poszczególnymi wzorcami jest również stały, co potwierdza wynik badania przedstawiony przez miarę ogólnej popularności,
- można przewidywać, że popularność wybranych wzorców projektowych nie zmniejszy się w 2016 roku.

5. Podsumowanie

W pracy przedstawiono wyniki badań popularności wzorców projektowych. W opracowaniu wyników zostały wykorzystane techniki data miningu, wspierane dodatkowo przez autorskie oprogramowanie. Dane do badań zostały pozyskane z serwisu stackoverflow.com, który jest jednym z przykładów sieci społecznościowych. Ważną zaletą wykorzystania tego rodzaju sieci jest fakt, że społeczność programistów obligatoryjnie dokona selekcji treści pojawiających się w serwisie.

Przeprowadzone badania dotyczą: ogólnej popularności wzorców projektowych, popularności wzorców względem podobnych dobrych praktyk oraz próby przewidywania popularności w 2016 na podstawie wcześniejszych tendencji. Uzyskane wyniki pokazują, że najpopularniejszymi wzorcami projektowymi są: Command, State, Singleton oraz Factory. Na przykładzie wymienionych wzorców wykazano stałą popularność na przestrzeni 5 ostatnich lat, co pozwala przewidywać, że popularność pozostanie bez zmian. Również w przypadku popularności względem wzorców architektonicznych wykazano, że wzorce projektowe [6] są popularniejsze.

Dalsze prace w zakresie technik eksploracji danych przewidują rozbudowę metody i wykorzystanie narzędzi Business Intelligence. Następnie przewidywana jest zmiana przedmiotu badań na inne zagadnienia inżynierii oprogramowania. Wyniki otrzymane w powyższych badaniach przyczyniają się do wyboru kierunku dalszych prac w zakresie oceny jakości implementacji wzorców projektowych.

BIBLIOGRAFIA

1. Alexander C.: *Język wzorców. Miasta, budynki, konstrukcja*, GWP Gdańskie Wydawnictwo Psychologiczne, Sopot 2008
2. Alnusair A., Zhao T., Yan G., *Rule-based detection of design patterns in program code*, International Journal on Software Tools for Technology Transfer, Springer Berlin Heidelberg, 2013
3. Bernd Bruegge, Allen H. Dutoit, *Inżynieria oprogramowania w ujęciu obiektowym*. UML, wzorce projektowe i Java, Helion

4. Czyczyn-Egird D.: *Eksploracja danych na podstawie szacowania informacji z wykorzystaniem regresji liniowej*, w: Modele inżynierii teleinformatyki 10, Wydawnictwo Uczelniane Politechniki Koszalińskiej, ISBN 978-83-7365-365-8, Koszalin 2015
5. Dietrich J., Elgar C., *A formal description of design patterns using OWL*, Software Engineering Conference, Australian, 2005
6. Gamma E. i inni, *WZORCE PROJEKTOWE Elementy oprogramowania wielokrotnego użytku*, Helion, Gliwice, 2010
7. Kasprowski P.: *Choosing a persistent storage for data mining task*, Studia Informatica, Vol. 33, No. 2B, 2012
8. Kirasić D., Basch D., *Ontology-Based Design Pattern Recognition*, Knowledge-Based Intelligent Information and Engineering Systems, Zagreb, Croatia, 2008
9. Pavlic L. i inni, *Improving design pattern adoption with Ontology-Based Design Pattern Repository*, Informatica An International Journal of Computing and Informatics, Vol 33, Ljubljana, Slovenia, 2009
10. Plecka P., Bzdyra K.: *Wykorzystanie ontologii w wymiarowaniu projektów informatycznych*, w: Innowacje w Zarządzaniu i Inżynierii Produkcji, Tom I, strony od 230 do 240, Politechnika Opolska, Opole 2014
11. Steven John Metsker, *C#. Wzorce projektowe*, Helion
12. Szewczyk A., *Popularność funkcji serwisów społecznościowych*, Studia Informatica 2011
13. Wang S., Lo D., Jiang L.: *An Empirical Study on Developer Interactions in StackOverflow*, Research Collection School Of Information Systems, Singapore 2013
14. Wojszczyk R.: *Porównanie sposobów reprezentacji wzorców projektowych*, w: Modele inżynierii teleinformatyki 9, strony od 133 do 145, Wydawnictwo Uczelniane Politechniki Koszalińskiej, ISSN 2353-6535, Koszalin 2014
15. Wojszczyk R.: *The model and function of quality assessment of implementation of design patterns*, w: Applied Computer Science, Vol. 11, No. 3, strony od 44 do 55, Politechnika Lubelska, ISSN 1895-3735, Lublin 2015

Streszczenie

Idea sieci społecznościowych jest znana od wielu lat. Jednak dopiero od niedawna nabrały nowego znaczenia, do czego przyczyniła się popularność współczesnych serwisów społecznościowych. Generowana treść przez społeczności jest ogromnym zasobem wiedzy do przeanalizowania. W artykule przedstawiono wyniki badań nad popularnością wzorców projektowych w oparciu o dane zgromadzone w wyspecjalizowanych sieciach społecznościowych. Wyniki badań uzyskano poprzez wykorzystanie technik eksploracji danych.

Abstract

The idea of social networks has been known for many years. However, only recently took on a new meaning, which was due to the popularity of today's social networks. Social service user-generated content constitutes tremendous stores of knowledge to be analysed. The article presented results of research on the popularity of design patterns on the basis of data gathered in the specialised social networks. The research results were obtained thanks to using data mining techniques.

Keywords: social networks, data mining, design patterns

Jacek Kaczmarek
Wydział Elektroniki i Informatyki
Politechnika Koszalińska

Przetwornica Buck sterowana metodą napięciową - dobór transmitancji analogowego układu sterowania

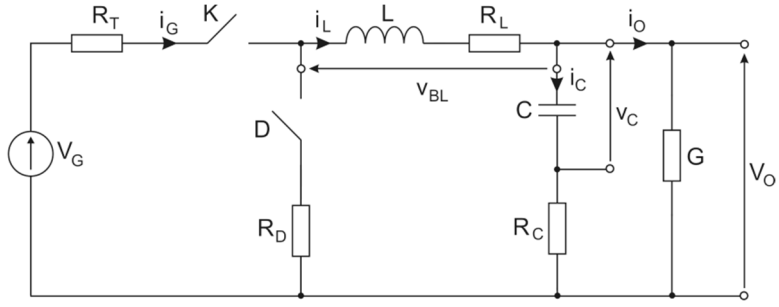
Słowa kluczowe: Buck, metoda napięciowa, obciążenie przetwornicy, analogowy układ sterowania, sprzężenie zwrotne

1. Wprowadzenie

Przetwornice Buck są powszechnie stosowane ponieważ blok główny przetwornicy zapewnia wysoką sprawność energetyczną, małe wymiary i małą masę układu zasilania, w porównaniu do liniowego regulatora napięcia. Napięcie wyjściowe bloku głównego zależy między innymi od punktu pracy, dlatego w przetwornicach stosuje się układ sterujący pracą bloku głównego. Największe, niepożądane zmiany napięcia wyjściowego w stanie nieustalonym bloku głównego przetwornicy Buck są wywoływane przez: zmiany napięcia zasilającego oraz zmiany obciążenia przetwornicy [1]–[3]. W pracy przedstawiono sposób kształtowania transmitancji układu sterującego pracą bloku głównego pozwalający zminimalizować wpływ ww. czynników na napięcie wyjściowe przetwornicy.

2. Model małosygnałowy bloku głównego przetwornicy

Schemat elektryczny bloku głównego przetwornicy Buck przedstawia rysunek nr 1 [5]:



Rysunek 1. Schemat elektryczny bloku głównego przetwornicy Buck

Wartość napięcia wyjściowego bloku głównego przetwornicy Buck reguluje się poprzez odpowiednie przełączanie kluczy K i D – rysunek 1. W pracy przyjęto, że blok główny przetwornicy Buck jest sterowany sygnałem PWM (Pulse Width Modulation) o stałej częstotliwości – f_S. W każdym cyklu pracy bloku głównego przetwornicy występują dwie fazy: ON i OFF. Zawsze na początku cyklu jako pierwsza występuje faza ON, potem następuje faza OFF. W czasie fazy ON klucz K jest załączony a klucz D jest wyłączony, w czasie fazy OFF jest odwrotnie. Blok główny przetwornicy Buck jest układem o zmiennej topologii, do analizy pracy bloku głównego stosuje się model małosygnałowy opisany w dziedzinie Laplace'a [6]–[11]. Model małosygnałowy bloku głównego przetwornicy jest także wykorzystywany do doboru parametrów transmitancji H_S układu sterującego pracą bloku głównego. Powszechnie stosowane modele małosygnałowe wyznaczają wpływ zmiany obciążenia bloku głównego przetwornicy na napięcie wyjściowe pośrednio, za pomocą impedancji wyjściowej bloku głównego przetwornicy [12]. W pracy wykorzystano model małosygnałowy bloku głównego przetwornicy opisany w publikacjach [5], [12]–[16]. Wybrany model małosygnałowy umożliwia bezpośrednio, bez korzystania z wzoru na transmitancję impedancji wyjściowej bloku głównego, wyznaczenie wpływu zmiany obciążenia bloku głównego na napięcie wyjściowe. Zastosowany model małosygnałowy bloku głównego przetwornicy upraszcza proces kształtowania transmitancji układu sterowania.

Przyjmując oznaczenia:

d_A –sygnał PWM sterujący blokiem głównym ,

d_a – składowa małosygnałowa sygnału d_A,

Θ – sygnał d_a w dziedzinie Laplace'a,

g_t – składowa małosygnałowa konduktancji obciążenia G bloku głównego,

Γ – sygnał g_t w dziedzinie Laplace'a

v_g – składowa małosygnałowa napięcia zasilającego blok główny v_G,

V_g – sygnał v_g w dziedzinie Laplace'a

v_o – składowa małosygnałowa napięcia wyjściowego bloku głównego v_O,

V_o – sygnał v_o w dziedzinie Laplace'a

Transmitancje małosygnałowe opisujące wpływ: Θ, V_g, i Γ na V_o mają postać:

$$H_d = \left.\frac{V_o}{\Theta}\right|_{V_g = 0, \Gamma = 0} \tag{1}$$

$$H_g = \left.\frac{V_o}{V_g}\right|_{\Theta = 0, \Gamma = 0} \tag{2}$$

$$H_\Gamma = \left.\frac{V_o}{\Gamma}\right|_{V_g = 0, \Theta = 0} \tag{3}$$

Transmitancje nieidealnego bloku głównego przetwornicy Buck: H_d, H_g, H_Γ wyrażają się zależnościami [5], [12]–[16]:

Transmitancja H_d nieidealnego bloku głównego przetwornicy Buck:

$$H_d = \frac{\left[V_G - I_L \cdot \left(R_T - R_D\right)\right] \cdot \left(1 + C \cdot R_C \cdot s\right)}{M_0 + M_1 \cdot s + M_2 \cdot s^2} \tag{4}$$

Transmitancja H_g nieidealnego bloku głównego przetwornicy Buck:

$$H_g = \frac{D_A \cdot \left(1 + C \cdot R_C \cdot s\right)}{M_0 + M_1 \cdot s + M_2 \cdot s^2} \tag{5}$$

Transmitancja H_Γ nieidealnego bloku głównego przetwornicy Buck:

$$H_\Gamma = -\frac{\left[D_A \cdot \left(R_T - R_D\right) + R_D + R_L + L \cdot s\right] \cdot \left(1 + C \cdot R_C \cdot s\right) \cdot V_O}{M_0 + M_1 \cdot s + M_2 \cdot s^2} \tag{6}$$

Współczynniki transmitancji (4) – (6) określają zależności:

$$M_0 = 1 + G \cdot \left[D_A \cdot \left(R_T - R_D\right) + R_D + R_L\right] \tag{7}$$

$$M_1 = G \cdot L \cdot C \cdot \left[R_Z + R_C \cdot \left(1 + G \cdot R_Z\right)\right] \tag{8}$$

$$R_Z = D_A \cdot \left(R_T - R_D\right) + R_D + R_L \tag{8.1}$$

$$M_2 = C \cdot L \cdot \left(1 + G \cdot R_C\right) \tag{9}$$

$$V_O = \frac{D_A \cdot V_G}{1 + G \cdot R_Z} \tag{10}$$

$$I_L = G \cdot V_O \tag{11}$$

V_O, D_A, V_G I_L G oznaczają składowe spoczynkowe napięcia wyjściowego, wypełnienia sygnału sterującego, napięcia zasilającego przetwornicę, prądu cewki oraz konduktancji obciążenia bloku głównego przetwornicy.

W modelach małosygnałowych przetwornic Buck sterowanych metodą napięciową transmitancja H_d opisuje wpływ sygnału sterującego d_a na napięcie wyjściowe bloku głównego przetwornicy v_o. Stabilna praca przetwornicy z zamkniętą pętlą sprzężenia zwrotnego wymaga uwzględnienia właściwości transmitancji H_d przy doborze transmitancji układu sterowania H_s. Właściwości transmitancji H_g i H_Γ pomagają określić dodatkowe wymagania jakie powinna spełnić transmitancja układu sterowania H_s by zapewnić skuteczne tłumienie wpływu v_g i g_t na v_o. Przy doborze współczynników transmitancji układu sterowania H_s istotną rolę odgrywa rozmieszczenie zer i biegunów transmitancji sterowania H_d [7]–[9], [11], [17], [18] oraz wartości współczynników:

f_r – Częstotliwość rezonansowa modelu małosygnałowego bloku głównego przetwornicy informuje o położeniu biegunów transmitancji H_d w dziedzinie częstotliwości. W praktyce wartości f_r przybliża się wzorem na częstotliwość rezonansową idealnego obwodu szeregowego L C [7], [8], [11], [17] lub stosuje się dokładniejszy wzór uwzględniający pasożytnicze wartości R_L i R_C oraz obciążenie przetwornicy [6]. Wartości f_r można wyznaczyć empirycznie na podstawie pomiaru $|H_d|$ [18].

f_{ESR} – Częstotliwość odpowiada położeniu zera transmitancji H_d w dziedzinie częstotliwości. O wartości f_{ESR} decyduje pojemność i pasożytnicza rezystancja szeregowa kondensatora w bloku głównym przetwornicy. f_{ESR} odgrywa istotną rolę przy rozmieszczaniu biegunów transmitancji H_S [6], [8], [11], [19], [20]. Wartość f_{ESR} jest przybliżana wzorem [6], [8], [11], [19], [20] lub wyznaczana na podstawie pomiaru $|H_d|$ [18].

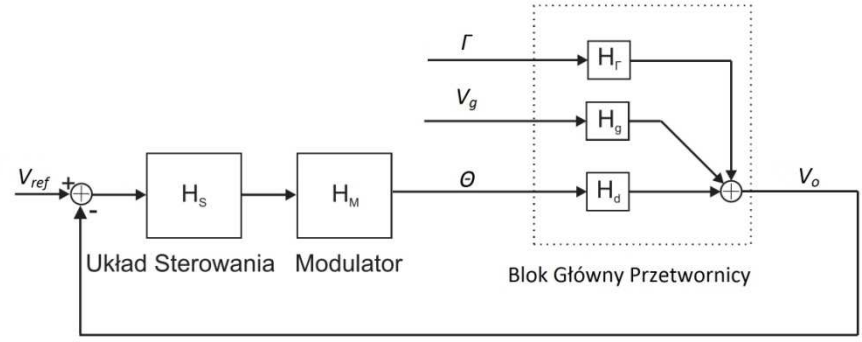
f_S –Elementy kluczujące w bloku głównym przetwornicy są przełączane ze stała częstotliwością f_S. Nawet w idealnym bloku głównym przetwornicy na skutek przełączania elementów kluczujących napięcie wyjściowe v_O ulega okresowym zmianom. W chwili przełączania rzeczywiste elementy kluczujące powodują powstawanie dodatkowych zmian chwilowej wartości napięcia wyjściowego v_O bloku głównego przetwornicy. W widmie częstotliwościowym zmiany napięcia wyjściowego v_O są widoczne w pobliżu w pobliżu częstotliwości f_S oraz harmonicznych i subharmonicznych f_S [11]. Układ sterowania nie powinien reagować na ww. zmiany v_O, dlatego górna granica pasma pętli sprzężenia zwrotnego powinna być odpowiednio mniejsza od wartości f_S. Wartość częstotliwości f_S przetwornicy Buck nie jest zdeterminowana żadnym wzorem, projektant dobiera wartość f_S kierując się swoim doświadczeniem i wartościami elementów bloku głównego. Wyżej wymienione zmiany napięcia wyjściowego bloku głównego przetwornicy v_O nie są uwzględniane w modelach małosygnałowych.

W praktyce spełniona jest poniższa zależność [21] wpływająca między innymi na złożoność i rozmieszczenie zer i biegunów transmitancji układu sterowania H_S:

$$f_r << f_{ESR} < f_S \tag{12}$$

3. Metoda napięciowa

Metoda sterowania określana jako napięciowa jest podstawową metodą stabilizacji napięcia wyjściowego przetwornicy DC/DC [6]–[11], [16]–[18], [20], [22]–[24]. Zaletami tej metody są: skuteczność stabilizacji napięcia wyjściowego oraz łatwość realizacji w praktyce. Model małosygnałowy przetwornicy sterowanej ww. metodą ilustruje rysunek 2.



Rysunek 2. Model małosygnałowy przetwornicy DC/DC sterowanej metodą napięciową

Blok główny przetwornicy Buck opisuje model małosygnałowy przedstawiony w rozdziale 2 [5], [12]–[16]. Sygnał sterujący pracą bloku głównego generuje modulator PWM. Pracę modulatora PWM przybliża liniowa transmitancji H_M [7], [9], [11], [20]:

$$H_M = \frac{1}{V_x} \tag{13}$$

gdzie:

V_X - amplituda sygnału piłokształtnego modulatora PWM o częstotliwości f_S.

Transmitancje opisujące wpływ sygnałów wejściowych na napięcie wyjściowe przetwornicy definiują poniższe wzory:

Wpływ sygnału V_g na V_o opisuje transmitancja H_{gsz}:

$$H_{gsz} = \frac{H_g}{1 + H_{OL}} \tag{14}$$

gdzie:

$H_{OL} = H_S \cdot H_M \cdot H_d$ - transmitancja pętli sprzężenia zwrotnego (14.1)

Wpływ sygnału Γ na V_o opisuje transmitancja $H_{\Gamma sz}$:

$$H_{\Gamma sz} = \frac{H_\Gamma}{1 + H_{OL}} \tag{15}$$

Wpływ pętli sprzężenia zwrotnego na stabilizację napięcia wyjściowego przetwornicy V_o zostanie pokazany na przykładzie sygnału V_g. Układ sterowania zapewni skuteczną stabilizację napięcia wyjściowego przetwornicy jeżeli będzie spełniona zależności:

$1>>|H_{gsz}|$ (16)

Z nierówności (16) wynika, że skuteczną stabilizację napięcia wyjściowego przetwornicy zapewni spełnienie poniższej nierówności [9], [10], [16]:

$|H_{OL}|>>1$ (17)

Transmitancja H_{OL} pozwala nie tylko ocenić skuteczność stabilizacji napięcia wyjściowego na podstawie zależności (17), ale przede wszystkim jest wykorzystywana do określenia stabilność przetwornicy z zamkniętą pętlą sprzężenia zwrotnego. Ocenę stabilności i skuteczność stabilizacji napięcia wyjściowego dokonuje się w dziedzinie częstotliwości za pomocą oceny wykresów Bodego. Analizę wykresów Bodego transmitancji H_{OL} powszechnie wykorzystuje się do ukształtowania transmitancji H_S układu sterowania [6]–[11], [17]–[20], [25], [26]. Znacznie rzadziej stosuje się dobór transmitancji H_S w dziedzinie czasu [27]. W dalszej części pracy przedstawiono sposób doboru parametrów funkcji H_S w dziedzinie częstotliwości za pomocą wykresów Bodego transmitancji H_{OL}.

4. Transmitancja układu sterowania H_S

Układ sterowania stosowany w przetwornicy sterowanej metodą napięciową powinien zapewnić:

- Stabilną pracę przetwornicy DC/DC.
- Skuteczne tłumienie wpływu zmiany wartości v_g, i g_t na v_o:
  - w stanie ustalonym v_o ma wymaganą wartość,
  - stan nieustalony v_o spowodowany zmianami v_g, g_t charakteryzuje się krótkim czasem trwania i małą amplitudą maksymalną.
- Odporność na zakłócenia generowane przez blok główny przetwornicy.
- Układ sterowania powinien spełniać wymagania ekonomiczne:
  - niski koszt elementów
  - prosta implementacja układu sterowania w praktyce.

By sprostać powyższym wymaganiom układ sterowania musi być dopasowany do bloku głównego przetwornicy i jego przewidywanego punktu pracy. Wartości nieidealnych elementów bloku głównego, obciążenie przetwornicy oraz częstotliwości przełączania kluczy K i D [11], [16], [17], [20], [28], [29] istotnie wpływają na właściwości bloku głównego przetwornicy. Im większa jest sprawność energetyczna bloku głównego, tym większe wymagania stawiane są układowi sterowania pod względem zapewnienia stabilnej pracy przetwornicy oraz skutecznej stabilizacji napięcia wyjściowego. Przyjęto, że blok główny przetwornicy pracuje w trybie CCM i charakteryzuje się wysoką sprawnością energetyczną, większą od 75%. Odzwierciedlają to charakterystyki Bodego transmitancji H_d, H_g i H_Γ. Dla przykładu, wykresy Bodego transmitancji H_d charakteryzują się zmianą fazy większą od ~120° w pobliżu częstotliwości rezonansowej f_r, dobroć obwodu rezonansowego bloku głównego przetwornicy Q jest nie mniejsza niż ~1.1 [9], [10]. Wpływ wielkości pasożytniczych elementów bloku głównego przetwornicy Buck na wykresy Bodego transmitancji H_g i H_Γ pokazano w pracach [12], [28].

Powyższe wymagania i właściwości bloku głównego przekładają się na złożoność transmitancji układu sterowania H_S [8]–[11], [17], [20]:

- Pojedynczy biegun H_S należy umieścić w zerze zmiennej zespolonej s. Obecność członu całkującego zapewnia, że w stanie ustalonym v_o osiąga żądaną wartość.
- Dwa zera H_S są ulokowane w pobliżu częstotliwości rezonansowej bloku głównego przetwornicy f_r. Obecność zer skompensuje zmiany fazy transmitancji H_d (w pobliżu częstotliwości f_r) i opóźnienie fazowe członu całkującego H_S.
- Drugi biegun H_S powinien być umieszczony w pobliżu częstotliwości f_{esr}, biegun ten kompensuje zero transmitancji H_d.
- H_S powinna posiadać trzeci biegun ograniczający pasmo układu sterowania. Zmniejszona zostanie podatność układu sterowania układu sterowania na zakłóceń generowane przez blok główny przetwornicy.

Z przedstawionych wyżej rozważań wynika, że funkcja transmitancji H_S powinna być trzeciego rzędu i posiadać dwa zera oraz trzy bieguny – oznaczenie 2Z3P. Transmitancję H_S typu 2Z3P określa wzór:

$$H_{s2z3P} = K_{dc} \cdot \frac{(s-z_1)\cdot(s-z_2)}{(s-p_1)\cdot(s-p_2)\cdot(s-p_3)} \tag{18}$$

Wszystkie zera i bieguny transmitancji H_S 2Z3P powinny być rzeczywiste [6]–[11], [17]–[20], [25], [26], [30]. Rozmieszczenie zer i biegunów transmitancji H_S w dziedzinie częstotliwości przybliża nierówność (12). Stosowanie wyższych rzędów transmitancji H_S albo zer, biegunów sprzężonych, wielokrotnych pozwala co najwyżej nieznacznie zwiększyć skuteczność stabilizacji napięcia wyjściowego. Istotną przeszkodą w wykorzystaniu bardziej złożonych funkcji transmitancji jest

konieczność zapewnienia stabilnej pracy przetwornicy - zmiany fazy H_S nie mogą być zbyt duże. Następnym ważnym ograniczeniem przy stosowaniu zer i biegunów sprzężonych albo wielokrotnych w transmitancji H_S jest stosunkowo mała różnica wartości pomiędzy częstotliwościami f_r i f_S bloku głównego przetwornicy. W porównaniu do transmitancji 2Z3P wpływ dodatkowych zer albo biegunów transmitancji H_S na właściwości transmitancji H_S w przedziale częstotliwości $< f_r, f_S >$, przynajmniej częściowo, wzajemnie się znosi. Typowo częstotliwość f_S jest od ~15 do ~60 razy większa od częstotliwości f_r [6]–[8], [11], [17], [18], [31], [32]. Ponadto, dodatkowe zera, bieguny transmitancji H_S zwiększają ekonomiczne koszty doboru i implementacji transmitancji H_S w analogowym układzie sterującym pracą przetwornicy. Funkcja transmitancji 2Z3P posiadająca wyłącznie zera i bieguny rzeczywiste zapewnia jednoczesne spełnienie wymagań stawianych jakości napięcia wyjściowego przetwornicy [10], [11] i wymagań ekonomicznych. W literaturze funkcja 2Z3P oznaczana jest jako funkcja typu III [6]–[8], [11], [17], [18], [20], [30] lub zmodyfikowana funkcja PID [4], [8], [25]. Funkcja 2Z3P nie jest jedyną powszechnie stosowaną funkcją transmitancji H_S. Dla bloków głównych przetwornic charakteryzujących się mniejszą sprawnością albo przy mniejszych wymaganiach dotyczących stanu nieustalonego v_o stosuje się funkcje niższego rzędu. Zazwyczaj są to funkcje transmitancji oznaczane w literaturze jako typ I i typ II [6]–[11], [17]–[20], [25], [26], [30].

5. Transmitancja pętli sprzężenia zwrotnego H_{OL}

Teoria sprzężenia zwrotnego pozwala określić stabilność układu z zamkniętą pętlę sprzężenia zwrotnego na podstawie właściwości transmitancji H_{OL} (14.1). By wykorzystać transmitancje H_{OL} do zbadania stabilności układu z zamkniętą pętlą sprzężenia zwrotnego, muszą być spełnione dwa warunki [9]:

- Część rzeczywista biegunów H_{OL} nie jest dodatnia.
- Tylko dla jednej wartości częstotliwości f_c spełnione jest równanie:

$$\left|H_{OL}\left(f_c\right)\right| = 1 \tag{19}$$

Jeżeli ww. warunki nie są spełnione, wówczas analizę stabilności należy przeprowadzić za pomocą np. twierdzenia Nyquista. W tym celu należy wyznaczyć trajektorię amplitudowo fazową transmitancji H_{OL} i zbadać jak trajektoria H_{OL} zachowuje się w pobliżu punktu (-1, j0). W praktyce, warunek dotyczący położenia biegunów transmitancji H_S i H_d jest zawsze spełniony. Układ z zamkniętą pętlą sprzężenia zwrotnego jest stabilny jeżeli spełniony jest warunek:

$$\left|H_{OL}\right| \geq 1 \wedge \varphi(H_{OL}) \in (-180^\circ, 0^\circ)$$

Do zbadania czy funkcja H_{OL} spełnia powyższy warunek stabilności wykorzystuje się wykresy Bodego [6]–[11], [17]–[20], [25], [26]. Zakładając, że

transmitancja H_S spełnia wymagania przedstawione w części 4, krzywą modułu H_{OL} można przybliżyć na wykresie Bodego linią prostą o nachyleniu -20 dB/ dekadę częstotliwości [8], [33], [34]. Układ sterowania będzie odporny na zakłócenia powstające w czasie pracy bloku głównego przetwornicy [11] jeżeli wartość częstotliwości f_c (19) będzie odpowiednio mniejsza od częstotliwości kluczowania bloku głównego f_S. Ponieważ częścią składową H_{OL} jest H_S, to korzystając z nierówności (17), można wykorzystać wykresy Bodego H_{OL} do ukształtowania H_S tak by zapewnić także skuteczną stabilizację napięcia wyjściowego przetwornicy [6]–[11], [17]–[20], [25], [26]. Poniżej zamieszczono parametry transmitancji H_{OL} wykorzystywane przy doborze współczynników transmitancji układu sterowania H_S:

f_c – Częstotliwość odcięcia definiuje wzór (19). Wartość f_c wpływa istotnie na pracę przetwornicy sterowanej napięciowo. Im większa wartość f_c tym lepsza dynamika i stabilizacja napięcia przetwornicy [6]–[8], [10], [11], [25], [34]–[36], ale pogarsza się odporność układu sterowania na zakłócenia generowane w czasie pracy bloku głównego [11]. Z tych względów przyjmuje się, że wartość f_c, dla analogowych układów sterowania, powinna mieścić się w zakresie $<0.1\ f_S, 0.2\ f_S>$ [8], [11], [18], [25]. Obecnie, zaleca się by górna granica tego przedziału wzrosła do $0.3\ f_S$ [23], [25], [34], [36]. Przybliżenie $|H_{OL}|$ prostą o nachyleniu -20 dB/ dekadę częstotliwości [8], [33], [34] ilustruje znaczenie wartości częstotliwości f_c, im większa jest wartość f_c tym większy jest przedział częstotliwości, dla których nierówność (17) jest prawdziwa. Niekiedy zwiększenie wartości f_c pozwala na zmniejszenie pojemności kondensatora bloku głównego przetwornicy bez pogorszenia parametrów dynamicznych stabilizowanego napięcia wyjściowego [37]. Przekłada się to korzystnie, między innymi, na miniaturyzację przetwornicy.

PM - Margines fazy definiuje poniższy wzór:

$$PM = 180 + \varphi\left(H_{OL}\left(f_c\right)\right) \tag{20}$$

Układ z pętlą sprzężenia zwrotnego jest stabilny, jeżeli $PM \geq 0$ W praktyce, wartość PM jest większa od zera, z uwagi na rozrzut wartości elementów elektronicznych oraz rosnącą amplitudę stanu nieustalonego v_o gdy $\varphi(H_{OL}(f_c))$ dąży do -180° [9], [10], [16]. Typowo wartość PM mieści się w przedziale <40°,100°> [7], [19], [34].

GM - Margines wzmocnienia, którego wartość określa wzór:

$$GM = 20 \cdot \log\left(\left|H_{OL}\left(f_{GM}\right)\right|\right) \tag{21}$$

gdzie:

f_{GM} - częstotliwość, dla której faza transmitancji H_{OL} przyjmuje wartość -180°.

Układ z zamkniętą pętlą sprzężenia zwrotnego jest stabilny, jeżeli $\varphi(H_{OL}) \leq -180^{\circ} \wedge |H_{OL}| < 1$. Z uwagi na rozrzut elementów przetwornicy GM przyjmuje wartości z przedziału <-5,-20>[dB] [18], [25], [34]. Parametry GM i PM łączy zależność: $f_c < f_{GM}$. Parametr GM nie zawsze jest wykorzystywany, w książce [9] poświęcono cały rozdział kształtowaniu pętli sprzężenia zwrotnego dla przetwornic: Buck, Buck-Boost i Boost, ale nawet nie wspomniano o GM.

6. Kształtowania transmitancji układu sterowania H_S

Wyznaczenie stabilności przetwornicy sterowanej metodą napięciową analitycznie jest zadaniem prostym. Znacznie trudniejsze jest wykazanie za pomocą analizy matematycznej, jaka transmitancja H_S spełni najlepiej wymagania dotyczące skutecznego tłumienia wpływu sygnałów v_g i g_t na v_o. Opis analityczny tego zagadnienia staje się zbyt skomplikowany i dlatego nie jest stosowany. Praktycznym sposobem okazuje się graficzne rozwiązanie problemu za pomocą oceny wykresów Bodego transmitancji H_{OL}.

Przyjmując że:

- można transmitancje HOL wykorzystać do badania stabilności,
- układ z zamkniętą pętlą jest stabilny,
- parametry HOL tj. fc, PM mieszczą się w granicach podanych w rozdziale 5,
- spełniona jest nierówność (17) dla częstotliwości mniejszych od ~0.5 fc

to dobrana transmitancja H_S zapewni stabilizację napięcia wyjściowego przetwornicy sterowanej metodą napięciową. Pomimo, że wymagania stawiane H_S związane z tłumieniem wpływu sygnałów v_g i g_t na v_o nie są identyczne [16], [36], [38], [39]. Zwykle za pomocą szeregu iteracji modyfikujących położenie zer i biegunów transmitancji H_S dochodzi do znalezienia funkcji H_S zapewniającej stabilność oraz skuteczną stabilizację napięcia wyjściowego przetwornicy [16]. Dobór funkcji transmitancji H_S jest procesem wieloetapowym, najważniejszymi etapami procesu kształtowania H_s są:

1. Określenie wymagań projektowych.
2. Wyznaczenie modelu małosygnałowego bloku głównego przetwornicy.
3. Wybranie typu funkcji transmitancji HS.
4. Wyznaczenie wartości współczynników funkcji HS.
5. Weryfikacja spełnienia przyjętych założeń projektowych.

W pracy przyjęto, że przed przystąpieniem do doboru transmitancji układu sterowania H_S znane są parametry i punkt pracy bloku głównego przetwornicy. Jednocześnie zdefiniowano zakres dopuszczalnych zmian v_o pod wpływem zmian v_g i g_t.

Pierwszym etapem jest zdefiniowanie założeń projektowych:

- wartości parametrów stanu nieustalonego vo pod wpływem zmian vg i gt,
- wartość marginesu fazy PM,
- wartość marginesu wzmocnienia GM (opcjonalnie),
- wartość częstotliwość odcięcia fc,
- wartość częstotliwość fS (opcjonalnie).

Drugim etapem jest wyznaczenie modelu małosygnałowego bloku głównego przetwornicy. Trzecim etapem jest wybranie funkcji H_S, przyjmując, że blok główny przetwornicy charakteryzuje się dużą sprawnością i jakość stabilizacji napięcia v_o powinna być jak najlepsza, funkcja H_S powinna być postaci 2Z3P (18) tzw. typ III [11]. Ostatnie dwa etapy kształtowania funkcji H_S tj. dobór wartości współczynników H_S i weryfikacja spełnienia przyjętych założeń są zazwyczaj wykonywane wielokrotnie, do momentu spełnienia przyjętych założeń. Spełnienie założeń projektowych zazwyczaj kończy prace nad kształtowaniem transmitancji H_S. Osiągnięcie założeń projektowych przez dobraną funkcję H_S nie oznacza, że otrzymana transmitancja H_S zapewnia najlepsze parametry stabilizacji napięcia wyjściowego. Również nie można założyć, że jest to jedyny typ transmitancji H_S zapewniający spełnienie przyjętych założeń projektowych. Ilość iteracji potrzebnych do wyznaczenia parametrów transmitancji H_S i sprawdzenia czy spełniono założenia projektowe zależy głównie od przyjętych założeń oraz doświadczenia i umiejętności projektanta. Zmieniając położenie zer i biegunów transmitancji należy kierować się wynikami poprzednich iteracji i właściwościami zer i biegunów w dziedzinie częstotliwości [9], [10]. Najczęściej jako punkt wyjścia do pierwszej iteracji przyjmuje się, że bieguny i zera H_{S2z3p} powinny być rozmieszczone w sposób podany w części 4. Takie położenie zer i biegunów H_S zapewnia zazwyczaj już w pierwszej iteracji stabilność układu z zamkniętą pętlą sprzężenia zwrotnego. Dla pokazania, że nie tylko funkcja transmitancji typu 2Z3P zapewnia skuteczne tłumienie wpływu sygnałów v_g i v_g na v_o w pracy pokazano wyniki badań symulacyjnych dla transmitancji H_S posiadającej dwa zera i dwa bieguny. Transmitancja H_S typu 2Z2P jest opisana wzorem:

$$H_{s2z2P} = K_{dc} \cdot \frac{(s-z_1)(s-z_2)}{(s-p_1)(s-p_2)} \tag{22}$$

Wykorzystując transmitancje typu 2Z2P przyjęto, że układ analogowy zapewni właściwości zaprojektowanej transmitancji 2Z2P w przedziale częstotliwości $<0, 0.5\ f_S)$.

Dobór funkcji H_S za pomocą oceny wykresów Bodego transmitancji H_{OL} pozwala na pewną dowolność w rozmieszczeniu zer i biegunów transmitancji H_S. Wynika to głównie z przyjętego sposobu rozwiązania problemu kształtowania transmitancji H_S oraz z różnorodności i rozrzutu elementów użytych do budowy bloku głównego przetwornicy. W literaturze i notach aplikacyjnych poświęconych

sterowaniu przetwornicami za pomocą metody napięciowej jedynie położenie bieguna p_1 w środku układu współrzędnych Laplace'a jest niezmienne. Zalecenia dotyczące rozmieszczenia pozostałych zer i biegunów transmitancji H_{S2z3p} albo wartości parametrów: f_c, PM, GM są w każdej z cytowanych publikacji inne. Ogólnie, oba zera transmitancji H_{S2z3p} mają skompensować zmiany fazy H_d spowodowane biegunami transmitancji H_d i opóźnienie fazowe wnoszone przez człon całkujący H_{S2z3p} – biegun p_1. Najczęściej są stosowane dwa sposoby rozmieszczenia zer H_{S2z3p} względem częstotliwości rezonansowej f_r bloku głównego przetwornicy. Rozwiązanie pierwsze to umieszczenie zer H_{S2z3p} tak, żeby częstotliwości odpowiadające położeniu zer były rozmieszczone poniżej i powyżej częstotliwości rezonansowej f_r np. f z_1=0.65 f_r i fz_2=2.19 f_r [6]. Drugim rozwiązaniem jest umieszczenie jednego z zer H_{S2z3p} tak by odpowiadająca mu częstotliwość pokrywała się z częstotliwością rezonansową bloku głównego przetwornicy f_r [8], [17], [20]. Zaletą takiego rozwiązania jest możliwość osiągnięcia dużej wartość marginesu fazy GM, wadą zmniejszenie wartość $|H_{OL}|$ w pobliżu częstotliwości rezonansowej bloku głównego. Zazwyczaj skutkuje to pogorszeniem parametrów dynamicznych stabilizowanego napięcia wyjściowego przetwornicy. Rzadko spotykanym rozwiązaniem jest umieszczenie obu zer transmitancji H_{S2z3p} tak, by odpowiadały częstotliwości rezonansowej bloku głównego przetwornicy [11]. Należy unikać umieszczenia jednego z zer zbyt blisko bieguna p_1, w takim wypadku efekt całkowania transmitancji H_{S2z3p} zostanie częściowo ograniczony. Może pojawić się w stanie ustalonym zbyt duża różnica pomiędzy napięciem v_o a zadaną wartością v_o. Zadaniem bieguna p_2 jest skompensowanie zmian transmitancji H_d spowodowanych pasożytniczą rezystancją szeregową kondensatora C w bloku głównym przetwornicy. Najczęściej zaleca się umieścić biegun p_2 na osi częstotliwości tak, by pokrywał się z położeniem zera transmitancji H_d - f_{ESR}. [6], [7], [11], [17], [25]. W pracach [19], [20] zaleca się by częstotliwość odpowiadająca położeniu bieguna p_2 była kilka razy większa od częstotliwości f_c. Uwzględniając nierówność (12) oraz to, że f_c zawiera się w przedziale częstotliwości $<0.1\ f_S, 0.3\ f_S >$ biegun p_2 także znajdzie się w pobliżu zera transmitancji H_d. Biegun p_3 transmitancji H_{S2z3p} umożliwia zwiększenie odporności układu sterowania na zakłócenia generowane przez blok główny przetwornicy i polepszenie parametrów dynamicznych przetwornicy sterowanej metodą napięciową (poprzez zwiększenie wartości f_c). O położeniu bieguna p_3 decydują elementy bloku głównego przetwornicy, częstotliwość f_S oraz wartości parametrów: f_c, PM, GM. Zazwyczaj częstotliwość odpowiadająca położeniu bieguna p_3 należy do przedziału częstotliwości $< 0.5\ f_{ESR}, 0.5\ f_S >$ [8], [11], [20], [25].

Podsumowując, rozmieszczenie zer i biegunów transmitancji H_{S2z3p} w dziedzinie Laplace, powinno odpowiadać w dziedzinie częstotliwości następującym wartościom:

$$\begin{aligned} f_{p1} &= 0 \\ f_{z1} &\in \left\langle 0.6 \cdot f_{LC}, 0.9 \cdot f_{LC} \right\rangle \\ f_{z2} &\in \left\langle 2 \cdot f_{LC}, 5 \cdot f_{LC} \right\rangle \\ f_{p2} &= f_{ESR} \\ f_{p3} &\in \left\langle 0.5 \cdot f_{ESR}, 0.9 \cdot f_{PWM} \right\rangle \end{aligned} \tag{19}$$

gdzie:

fPx - częstotliwość odpowiadająca położeniu bieguna px,

fZy - częstotliwość odpowiadająca położeniu zera zy,

Po rozmieszczeniu zer i biegunów transmitancji H_{S2z3p} należy wyznaczyć wartość współczynnika k_{dc} (18) tak, by $|H_{OL}|=1$ dla zakładanej wartości częstotliwości f_c. Następnie analizuje się wykresy Bodego H_{OL} oraz wyznacza zmiany v_o w dziedzinie czasu na skokowe zmiany v_g i g_t. W przypadku transmitancji H_S typu 2Z2P początkowe rozmieszczenie zer i biegunów jest identyczne jak dla H_S typu 2Z3P z pominięciem bieguna p_3.

By zmniejszyć wpływ g_t na v_o należy, zdaniem autora, zwrócić uwagę na różnicę pomiędzy właściwościami transmitancji H_Γ i H_g [5], [12]. Stan nieustalony v_o spowodowany skokową zmianą g_t charakteryzuje się większą dynamiką w porównaniu do dynamiki zmian v_o wywołanych skokową zmianą v_g. Dla idealnego bloku głównego zmiana obciążenia, po zaniku stanu nieustalonego, nie powoduje zmiany średniej wartości napięcia wyjściowego. W przypadku skokowej zmiany v_g, po zaniku stanu nieustalonego, wartość v_o uległa zmianie proporcjonalnie do zmiany v_g. Dla bloku głównego charakteryzującego się dużą sprawnością zmiana obciążenia powoduje jedynie nieznaczną zmianę v_o. Prowadzi to do wniosku, że dla tłumienia wpływu g_t na v_o największe znaczenie ma wartość $|H_{OL}|$ dla częstotliwości większych od ~0.5 f_r, inaczej jest w przypadku tłumienia wpływu v_g na v_o. Wymagania stawiane funkcji H_{OL} dla tłumienia wpływu sygnału v_g na v_o są inne od wymagań jakie są stawiane dla tłumienia wpływu sygnału g_t na v_o [16], [36], [38], [39]. Dla lepszego tłumienia wpływu g_t na v_o zaleca się by przy określaniu wymagań projektowych przyjąć jak największą wartość f_c z dopuszczalnego przedziału wartości f_c [23], [25], [34], [36], [39], [40] oraz użyć w bloku głównym kondensatora o jak najmniejszej wartości rezystancji szeregowej R_C [38]. Według autora, powyższe zalecenia należy uzupełnić:

1. Dobrać położenie zer i biegunów HS tak, by osiągnąć jak największe maksimum |HOL| w pobliżu fr.
2. Na wykresie Bodego pole powierzchni pod krzywą |HOL| powinno być jak największe dla częstotliwości < 0.5 fr, fc >. Krzywa |HOL| dla częstotliwości < fr, fc > powinna być funkcją wypukłą.

Podsumowując, graficzne rozwiązanie zagadnienia doboru transmitancji H_S dla przetwornicy sterowanej metodą napięciową pozwala zapewnić skuteczną stabilizację napięcia wyjściowego v_o. Nie można założyć, że otrzymana transmitancja H_S zapewnia najlepsze tłumienie wpływu v_g i g_t na v_o. Bardzo duży wpływ na jakość stabilizowanego napięcia wyjściowego, oprócz właściwości bloku głównego i sposobu rozmieszczenia zer i biegunów H_s mają parametry: PM, f_c i GM. Ich wartości są przyjmowane a priori, w przypadku gdy mimo wielu prób nie udało się spełnić założeń projektowych dotyczących parametrów dynamicznych stanu nieustalonego v_o spowodowanego zmianami v_g i g_t należy zmienić w pierwszej kolejności wartości f_c i PM.

7. Badania symulacyjne

7.1. Wprowadzenie

Wpływ rozmieszczenia zer i biegunów transmitancji analogowego układu sterowania na właściwości przetwornicy Buck sterowanej metodą napięciową zaprezentowano na przykładzie najczęściej stosowanej transmitancji układu sterowania H_S typu 2Z3P. Dla wykazania, że inny typ funkcji transmitancji układu sterowania może zapewnić bardzo podobne parametry stabilizowanego napięcia wyjściowego zamieszczono także przykład transmitancji układu sterowania H_S typu 2Z2P. Wyniki badań symulacyjnych ilustrują wykresy Bodego oraz wykresy w dziedzinie czasu.

7.2. Założenia projektowe dla transmitancji H_{OL}

Do projektowania transmitancji układu sterowania przystępuje się po zaprojektowaniu bloku głównego przetwornicy oraz zdefiniowaniu punktu pracy przetwornicy [9]. Kolejnym krokiem jest zdefiniowanie wymagań stawianych transmitancjom H_S oraz H_{OL}.

Przyjęto następujące założenia projektowe:

A. margines fazy PM nie może być mniejszy niż 40°
B. częstotliwość f_c wynosi $0.2 \cdot f_{WPM}$
C. funkcja transmitancji H_S jest typu 2Z2P lub 2Z3P i posiada wyłącznie rzeczywiste zera i bieguny.

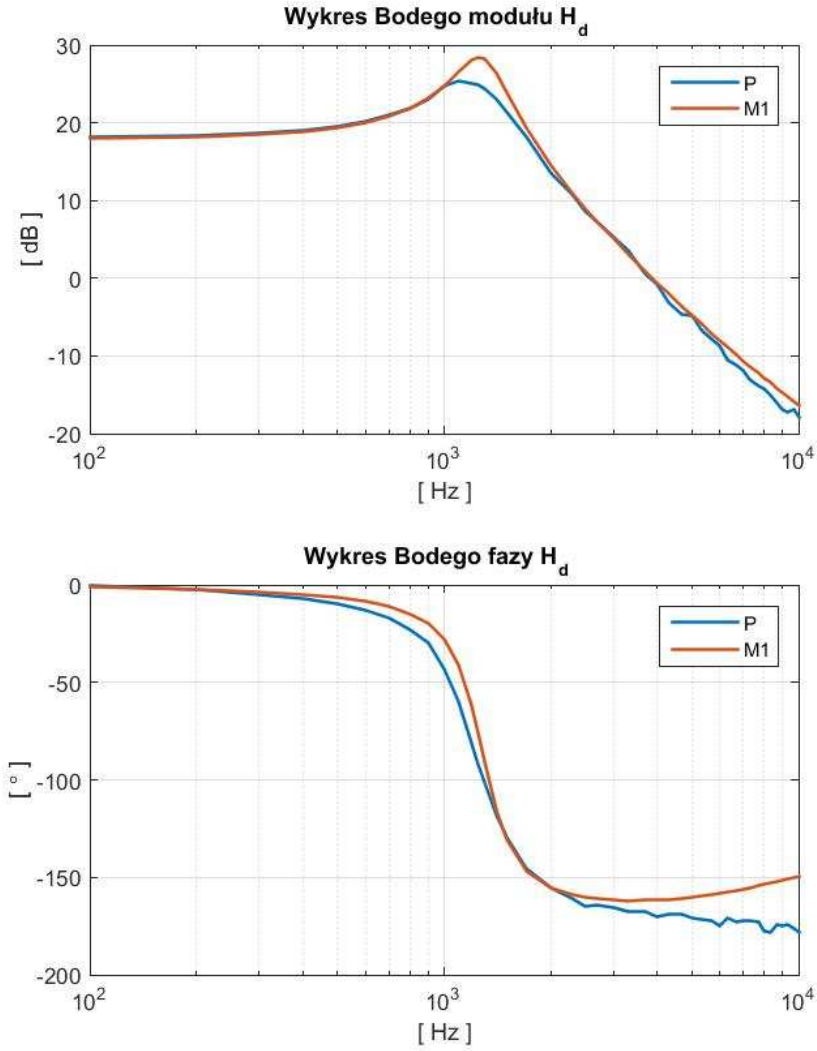
7.3. Przetwornica synchroniczna Buck

Model małosygnałowy przetwornicy Buck wykorzystywany w badaniach symulacyjnych powstał na podstawie zmierzonych wartości elementów bloku głównego synchronicznej przetwornicy Buck. Funkcje transmitancji H_d, H_g i H_r wyznaczono na podstawie wzorów (6) - (8). Zmierzone wartości elementów bloku głównego przetwornicy oraz punkt pracy zamieszczono poniżej:

V_G=7.99 V, D_A=0.5, f_S =100 kHz, G=1 S, R_T=7 mΩ, R_D=7 mΩ, L=47 μH, R_L=12 mΩ, C=325.35 μF, R_C=26 mΩ

Dla pokazania różnic pomiędzy modelem małosygnałowym przetwornicy Buck a zbudowaną przetwornicą dokonano pomiaru transmitancji H_d badanej przetwornicy. Na podstawie pomiaru amplitudy oraz fazy sygnału wejściowego d_a i odpowiedzi przetwornicy v_o wyznaczono punkty charakterystyki Bodego transmitancji H_d badanej synchronicznej przetwornicy Buck. Wyniki pokazano na wykresie 3.

Dla zapewnienia stabilności układu z zamkniętą pętlą sprzężenia zwrotnego najważniejsza jest różnica pomiędzy krzywą fazy transmitancji H_d modelu małosygnałowego badanej przetwornicy a zmierzoną charakterystyką fazy transmitancji H_d dla częstotliwości większych od f_r. Różnica ta wynika z nieuwzględnienia w modelu małosygnałowym bloku głównego przetwornicy opóźnienia wnoszonego przez modulator PWM oraz klucze elektroniczne [39], [41]. W żadnej z prezentowanych publikacji opóźnienie fazowe wprowadzane przez modulator PWM i klucze elektroniczne nie jest uwzględniane przy doborze transmitancji układu sterowania H_S. Wpływ tego opóźnienia na stabilność przetwornicy z zamkniętą pętlą sprzężenia zwrotnego jest neutralizowany przez przyjęcie odpowiednio dużej wartości marginesu fazy PM. Dlatego przyjęto, że margines fazy PM powinien być nie mniejszy od 40°. Różnice pomiędzy krzywymi modułu H_d modelu małosygnałowego przetwornicy a rzeczywistą przetwornicą wynikają przede wszystkim z nieuwzględnienia rezystancji źródła zasilającego przetwornicę, kabli oraz strat związanych z przełączaniem elementów elektronicznych bloku głównego przetwornicy i nie są istotne dla stabilności układu z zamkniętą pętlą sprzężenia zwrotnego. Zwiększenie rezystancji R_T i R_D modelu małosygnałowego z wartości 7 mΩ (zmierzona rezystancja kanału tranzystora MOS w stanie włączenia) do wartości 35 mΩ powoduje praktyczne nałożenie się krzywych modułu H_{OL} badanej przetwornicy i jej modelu.



Rysunek 3. Wykres Bodego transmitancji H_d: P –zmierzona transmitancja H_d badanej synchronicznej przetwornicy, M1 – obliczona transmitancja H_d modelu małosygnałowego badanej przetwornicy.

7.4. Transmitancja układu sterowania

Podane poniżej przykłady analogowej transmitancji układu sterowania H_S obrazują zakres zmian parametrów funkcji transmitancji H_{OL}. Dobierając przykłady, autor starał się by istotnej zmianie uległ wyłącznie jeden parametr, np. wartość częstotliwości f_c. Pozostałe parametry, powinny być jak najbardziej zbliżone do wzorcowej transmitancji układu sterowania H_S. Nie wszystkie przedstawione transmitancje układu sterowania spełniają wymagania projektowe, przykładem jest transmitancja Hs dla której wartość fc wynosi $0.1 \cdot f_S$ zamiast $0.2 \cdot f_S$.

Oznaczenia badanych transmitancji układu sterowania H_s:

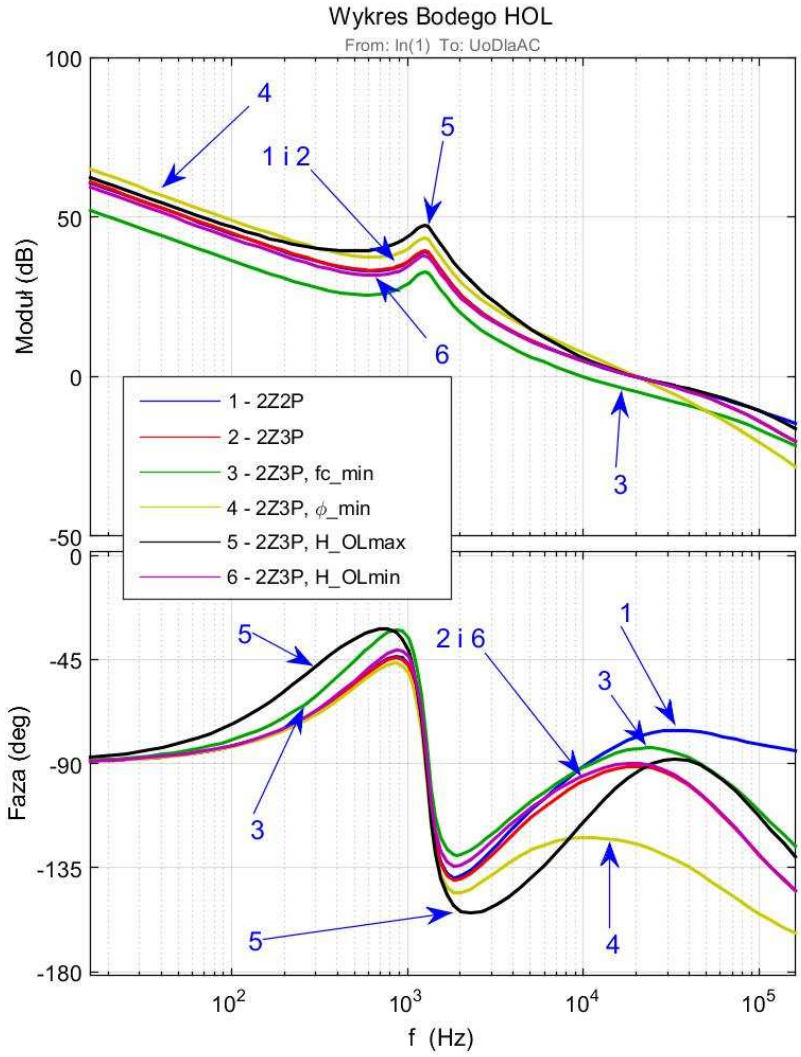
1. 2Z3P - transmitancja będąca punktem odniesienia w ocenie właściwości badanych układów sterowania. Spełnia założenia projektowe.
2. 2Z2P - transmitancja spełnia założenia projektowe.
3. 2Z3P,fCmin - transmitancja o zmniejszonej wartość fc z $0.2 \cdot$ fS do wartości $0.1 \cdot$ fS. Transmitancja 2Z3P, fCmin nie spełnia założeń projektowych.
4. 2Z3P, Φmin - transmitancja dla której PM został zmniejszony do wartości ~35°. Wartość PM nie spełnia założeń projektowych.
5. 2Z3P, HOLmin - transmitancja mająca najmniejszą wartość modułu HOL w lokalnym ekstremum w pobliżu częstotliwości fr. Transmitancja spełnia założenia projektowe.
6. 2Z3P, HOLmax - transmitancja mająca największą wartość modułu HOL w lokalnym ekstremum w pobliżu częstotliwości fr. Transmitancja nie spełnia założenia projektowego – PM wynosi ~26°. Dla funkcji HOLmax stan nieustalony vo spowodowany zmianą gt ma najmniejszą amplitudę – rysunek 6.

Wartości współczynników ww. transmitancji H_S układu sterowania zamieszczono w tabelach 1-3 zamieszczonych w dalszej części pracy.

7.5. Wykresy

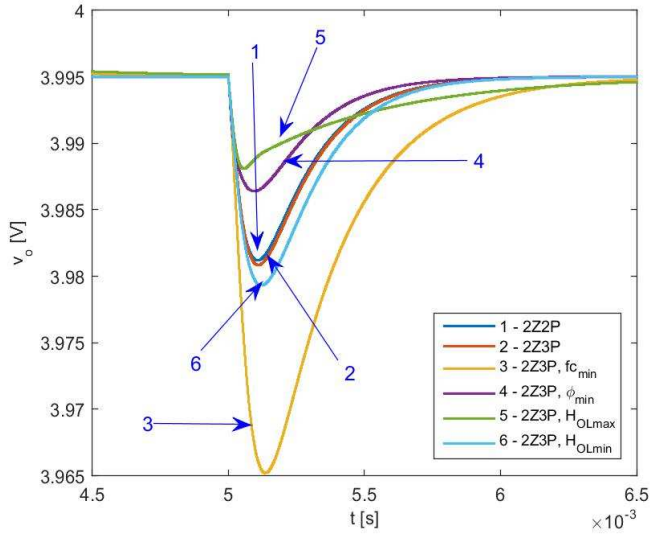
W dalszej części pracy przedstawiono wykresy Bodego transmitancji H_{OL} odpowiadające badanym funkcjom transmitancji układu sterowania H_S. Dla zbadania odpowiedzi przetwornicy sterowanej metodą napięciową wyznaczono na podstawie wzorów (2) i (3) zmianę sygnału v_o pod wpływem jednostkowej skokowej zmiany sygnału v_g lub g_t.

7.5.1. Wykresy Bodego

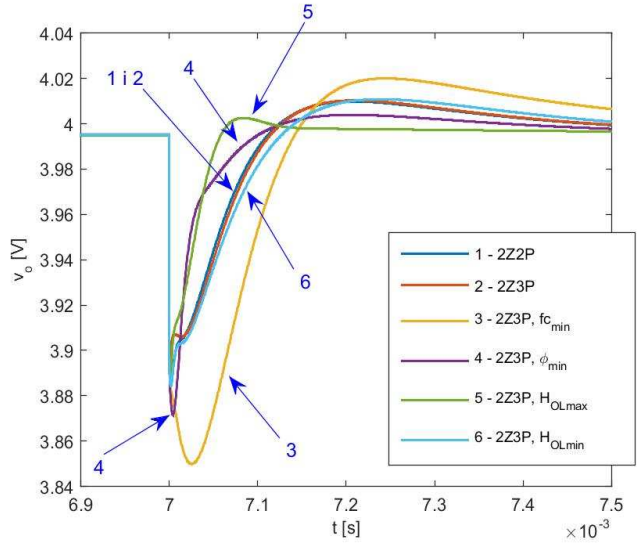


Rysunek 4. Wykresy Bodego transmitancji H_{OL}

7.5.2. Wykresy w dziedzinie czasu



Rysunek 5. Odpowiedź przetwornicy sterowaną metodą napięciową na skokową zmianę v_g



Rysunek 6. Odpowiedź przetwornicy sterowaną metodą napięciową na skokową zmianę g_t

8. Wnioski

Przegląd literatury poświęconej projektowaniu pętli sprzężenia zwrotnego dla przetwornicy Buck oraz przedstawiona w pracy analiza doboru transmitancji układu sterowania H_S prowadzą do konkluzji, że liczba transmitancji Hs spełniających założenia projektowe jest nieskończona. Dotyczy to zarówno wartości parametrów funkcji Hs jak i jej typów. Z względów praktycznych i ekonomicznych stosuje się najczęściej funkcje transmitancji H_S rzędu trzeciego. Najtrudniejszym obecnie zadaniem projektowym jest zapewnienie skutecznej stabilizacji napięcia wyjściowego przetwornicy v_o mimo dużych zmian obciążenia przetwornicy g_t. Dynamika zmian napięcia v_o pod wpływem zmian g_t jest znacznie większa od dynamiki zmian v_o pod wpływem zmian v_g - ilustrują to rysunki 5 i 6. Przeprowadzone badania symulacyjne potwierdzają, że funkcja transmitancji H_S typu 2Z3P zapewnia spełnienie wymagań projektowych stawianych transmitancji układu sterowania. Zaproponowany sposób rozmieszczenia zer i biegunów transmitancji układu sterowania zapewnia spełnienie założeń projektowych oraz skutecznie tłumi wpływ v_g i g_t na v_o – rysunki 5 i 6.

Tabela 1. Wartości współczynników analogowej transmitancji H_S typu 2Z2P i 2Z3P

	H_S2Z2P	H_S2Z3P
k_{dc}	≈43.373	$\approx 2.5125 \cdot 10^7$
Z_1	$2 \cdot \pi \cdot 0.75 \cdot f_r \approx 5910$ Hz	$2 \cdot \pi \cdot 0.75 \cdot f_r \approx 5910$ Hz
Z_2	$2 \cdot \pi \cdot 1.6 \cdot f_r \approx 1.261 \cdot 10^4$ Hz	$2 \cdot \pi \cdot 1.6 \cdot f_r \approx 1.261 \cdot 10^4$ Hz
P_1	0 Hz	0 Hz
P_2	$2 \cdot \pi \cdot 2 \cdot f_{ESR} \approx 2.328 \cdot 10^5$ Hz	$2 \cdot \pi \cdot 2 \cdot f_{ESR} \approx 2.328 \cdot 10^5$ Hz
P_3		$2 \cdot \pi \cdot 3 \cdot f_c \approx 5.655 \cdot 10^5$ Hz

Tabela 2. Transmitancje H_S analogowego układu sterowania o zmniejszonej wartości f_c - 2Z3P, $_{fcmin}$ i o zmniejszonej wartości marginesu fazy - 2Z3P, $_{\Phi min}$

	H_S2Z3P,f_{Cmin}	H_S2Z3P, $_{\Phi min}$
k_{dc}	$\approx 2.8324 \cdot 10^7$	$\approx 9.3333 \cdot 10^6$
Z_1	$2 \cdot \pi \cdot 0.55 \cdot f_r \approx 4334$ Hz	$2 \cdot \pi \cdot 0.75 \cdot f_r \approx 5910$ Hz
Z_2	$2 \cdot \pi \cdot 1.4 \cdot f_r \approx 1.103 \cdot 10^4$ Hz	$2 \cdot \pi \cdot 1.6 \cdot f_r \approx 1.261 \cdot 10^4$ Hz
P_1	0 Hz	0 Hz
P_2	$2 \cdot \pi \cdot 2 \cdot f_{ESR} \approx 2.328 \cdot 10^5$ Hz	$2 \cdot \pi \cdot 3 \cdot f_{ESR} \approx 3.493 \cdot 10^5$ Hz
P_3	$2 \cdot \pi \cdot 6 \cdot f_C \approx 1.131 \cdot 10^6$ Hz	$2 \cdot \pi \cdot 0.7 \cdot f_C \approx 8.796 \cdot 10^4$ Hz

Tabela 3. Transmitancje H_S analogowego układu sterowania o najmniejszej wartości lokalnego maksimum modułu H_{OL}- oznaczenie 2Z3P, H_{OLmin} i o największej wartości lokalnego maksimum modułu H_{OL} - oznaczenie 2Z3P, H_{OLmax}

	H_S2Z3P,H_{OLmin}	H_S2Z3P,H_{OLmax}
k_{dc}	$\approx 2.518 \cdot 10^7$	$\approx 4.8147 \cdot 10^7$
Z_1	$2 \cdot \pi \cdot f_r \approx 7880$ Hz	$2 \cdot \pi \cdot 0.25 \cdot f_r \approx 1970$ Hz
Z_2	$2 \cdot \pi \cdot f_r \approx 7880$ Hz	$2 \cdot \pi \cdot 6.6 \cdot f_r \approx 5.201 \cdot 10^4$ Hz
P_1	0 Hz	0 Hz
P_2	$2 \cdot \pi \cdot 2 \cdot f_{ESR} \approx 2.328 \cdot 10^5$ Hz	$2 \cdot \pi \cdot 3 \cdot f_{ESR} \approx 3.493 \cdot 10^5$ Hz
P_3	$2 \cdot \pi \cdot 3 \cdot f_C \approx 5.655 \cdot 10^5$ Hz	$2 \cdot \pi \cdot 6.7 \cdot f_C \approx 8.419 \cdot 10^5$ Hz

Bibliografia

1. L. Ibarra, H. Bastida, P. Ponce, i A. Molina, „Robust control for buck voltage converter under resistive and inductive varying load", w *2016 13th International Conference on Power Electronics (CIEP)*, 2016, ss. 126–131.
2. K. Sato, T. Sato, i M. Sonehara, „Transient response improvement of digitally controlled buck-type dc-dc converter with feedforward compensator", w *2015 IEEE International Telecommunications Energy Conference (INTELEC)*, 2015, ss. 1–5.
3. U. Nasir, Z. Iqbal, M. T. Rasheed, i M. K. Bodla, „Voltage mode controlled buck converter under input voltage variations", w *2015 IEEE 15th International Conference on Environment and Electrical Engineering (EEEIC)*, 2015, ss. 986–991.
4. S. Seshagiri, E. Block, I. Larrea, i L. Soares, „Optimal PID design for voltage mode control of DC-DC buck converters", w *2016 Indian Control Conference (ICC)*, 2016, ss. 99–104.
5. W. Janke, „Equivalent circuits for averaged description of DC-DC switch-mode power converters based on separation of variables approach", *Bull. Polish Acad. Sci. Tech. Sci.*, t. 61, nr 3, ss. 711–723, sty. 2013.
6. R. Miftakhutdinov, „Designing for Small-Size, High-Frequency Applications Using TPS546xx DC / DC Converters", *Texas Instruments Application Report SLVA107*. 2001.
7. B. D. Mitchell i B. Mammano, „Designing Stable Control Loops", *Texas Instruments*. 2002.
8. M. Qiao, P. Parto, i R. Amirani, „Application Note AN-1043 Stabilize the Buck Converter with Transconductance Amplifier", *International Rectifier*. 2002.

9. R. W. Erickson, *Fundamentals of power electronics*, 2nd ed. Norwell Mass.: Kluwer Academic Publishers, 2001.
10. M. Kazimierczuk, *Pulse-width modulated DC-DC power converters*, First Edit. A John Wiley and Sons, Ltd, Publication, 2008.
11. W. H. Lei i T. K. Man, „AND8143/D A General Approach for Optimizing Dynamic Response for Buck Converter", *ON Semiconductor*. 2004.
12. W. Janke, M. Bączek, i M. Walczak, „Output characteristics of step-down (Buck) power converter", *Bull. Polish Acad. Sci. Tech. Sci.*, t. 60, nr 4, ss. 751–755, sty. 2012.
13. W. Janke, „Averaged models of pulse-modulated DC-DC power converters. Part II. Models based on the separation of variables", *Arch. Electr. Eng.*, t. 61, nr 4, ss. 633–654, sty. 2012.
14. W. Janke, „The extension of small signal model of switching DC-DC power converters", *XII Symp. PPEEm*, 2007.
15. W. Janke, „Averaged models of pulse-modulated DC-DC power converters. Part I. Discussion of standard methods", *Archives of Electrical Engineering*, t. 61, nr 4. ss. 609–631, 01-sty-2012.
16. W. Janke, *Impulsowe Przetwornice Napięcia Stałego*. Koszalin: Politechnika Koszalińska, 2014.
17. D. Mattingly, „Designing Stable Compensation Networks for Single Phase Voltage Mode Buck Rregulators", *Intersil*. 2003.
18. R. Miftakhutdinov, „Compensating DC / DC Converters with Ceramic Output Capacitors", *Texas Instruments*. 2005.
19. Texas Insturments, „SLVP101,SLVP102, and SLVP103 Buck Converter Design Using the TL5001", *Texas Instruments*. 1998.
20. STMicroelectronics, „L5981", *STMicroelectronics Datasheet*. STMicroelectronics, 2009.
21. T. Siew-Chong, Y. M. Lai, i M. K. H. Cheung, „An adaptive sliding mode controller for buck converter in continuous conduction mode", *Ninet. Annu. IEEE Appl. Power Electron. Conf. Expo. 2004. APEC '04.*, ss. 1395–1400.
22. L. Guo, J. Y. Hung, i R. M. Nelms, „Digital controller design for buck and boost converters using root locus techniques", w *IECON'03. 29th Annual Conference of the IEEE Industrial Electronics Society*, 2003, ss. 1864–1869.
23. A. Prodić, D. Maksimović, i R. W. Erickson, „Design and implementation of a digital PWM controller for a high-frequency switching DC-DC power converter", w *IECON'01. 27th Annual Conference of the IEEE Industrial Electronics Society*, 2001, t. 2, ss. 893–898.

24. S. Choudhury, „Designing a TMS320F280x based digitally controlled dc-dc switching power supply", *Texas Instruments Application Report SPRAAB3*. 2005.
25. International Rectifier, „IR3637SPBF - 1 % Accurate Synchronous PWM Controller. Data Sheet No. PD94713", *Internationl Rectifier*. ss. 1–21, 2005.
26. Texas Insturments, „SLVP088 20 V to 40 V Adjustable Boost Converter Evaluation Module User ' s Guide", *Texas Instruments*. 1997.
27. M. M. Peretz i S. Ben-Yaakov, „Time-Domain Design of Digital Compensators for PWM DC-DC Converters", *IEEE Trans. Power Electron.*, t. 27, nr 1, ss. 284–293, sty. 2012.
28. W. Janke, M. Walczak, i M. Bączek, „Charakterystyki wejściowe i wyjściowe przetwornic napięcia BUCK i BOOST z uwzględnieniem rezystancji pasożytniczych", *Przegląd Elektrotechniczny*, ss. 291–294, 2012.
29. W. Janke, M. Bączek, i M. Walczak, „Output characteristics of step-down (Buck) power converter", t. 60, nr 4, ss. 1–5, 2012.
30. S. W. Lee, „Demystifying Type II and Type III Compensators Using Op- Amp and OTA for DC / DC Converters", nr July, ss. 1–16, 2014.
31. Texas Insturments, „PTD08A015W", *Texas Instruments*. 2010.
32. Texas Insturments, „PTD08A020W", *Texas Instruments*. 2010.
33. P. A. Amir M. Rahimi, Parviz Parto, „Application Note AN-1162". ss. 1–36.
34. Intersil, „Digital-DC^{TM} Control Loop Compensation - AN2016.0", *Intersil*. ss. 1–10, 2009.
35. S. Raghunath, „Digital Loop Exemplified", *Texas Instruments Application Report*, nr December. Texas Instruments, ss. 1–20, 2011.
36. T. Takayama i D. Maksimović, „Digitally controlled 10 MHz monolithic buck converter", *2006 IEEE Work. Comput. Power Electron.*, ss. 154–158, 2006.
37. Intersil, „ISL6580", *Intersil*, nr September. 2003.
38. R. Redl, B. P. Erisman, i Z. Zansky, „Optimizing the load transient response of the buck converter", *APEC '98 Thirteen. Annu. Appl. Power Electron. Conf. Expo.*, ss. 170–176, 1998.
39. M. Hagen i V. Yousefzadeh, „Applying Digital Technology to PWM Control-Loop Designs", *Power Supply Design Seminar - SEM1800*. Texas Instruments, s. 7.1-7.28, 2009.

40. B. Prakash i S. Prakash, „Analysis of High DC Bus Voltage Stress in the Design of Single Stage Single Switch Switch Mode Rectifier", *Proc. IEEE Int. Symp. Ind. Electron. 2005. ISIE 2005.*, ss. 505–512, 2005.
41. B. Bryant i M. K. Kazimierczuk, „Voltage-Loop Power-Stage Transfer Functions With MOSFET Delay for Boost PWM Converter Operating in CCM", t. 54, nr 1, ss. 347–353, 2007.

Streszczenie

W pracy przedstawiono sposób kształtowania analogowej transmitancji układu sterowania H_S dla przetwornicy Buck sterowanej metodą napięciową w trybie CCM (Continuous Conduction Mode). Proponowany sposób kształtowania transmitancji H_S zapewnia skuteczne tłumienie wpływu zmiany obciążenia przetwornicy na napięcie wyjściowe przetwornicy. Dobór parametrów transmitancji H_S oraz badanie stabilności układu z zamkniętą pętlą sprzężenia zwrotnego przeprowadzono w dziedzinie częstotliwości za pomocą wykresów Bodego. Na wybranych przykładach pokazano wpływ rozmieszczenia zer i biegunów funkcji transmitancji H_S na pracę przetwornicy.

Abstract

The article presents a method of shaping the transfer function H_S of the analog control circuit for Buck converter in voltage control mode. Buck converter is operating in continuous conduction mode. The proposed method of shaping H_S provides an effective damping of load changes to the output voltage of the Buck converter. Parameters of H_S transmittance were chosen in frequency domain using Bode diagrams. Stability of close loop system was tested by using Bode diagrams of open loop transmittance.

Keywords: Buck, voltage mode, load converter, analog control circuit, feedback

Krzysztof Stolc
Wydział Elektroniki i Informatyki
Politechnika Koszalińska

Zastosowanie algorytmu genetycznego do tworzenia portretów pamięciowych

Słowa kluczowe: algorytm genetyczny, tworzenie portretu pamięciowego

1. Wstęp

Początkowo przestępców rozpoznawano na podstawie twarzy, postawy oraz innych charakterystycznych cech. Z powodu braku technik utrwalających wygląd sprawców, policjanci musieli polegać na własnej spostrzegawczości i zdolności zapamiętywania. W latach 80. XIX wieku Alfonse Bertillon stworzył antropometryczną kartotekę, składającą się z danych takich jak wzrost, wysokość osoby siedzącej, rozwartość ramion, długość i szerokość głowy, szerokość twarzy, długość i szerokość prawego uda, długość lewego przedramienia, lewej stopy, średniego i małego palca lewej ręki. W Lyonie w 1952 r. powstał pierwszy portret składany. Zdjęcia o wymiarach 13 cm x 18 cm cięto w poprzek tak, by uzyskać obraz poszczególnych części twarzy, takich jak włosy, czoło, brwi, oczy, nos, usta i broda. Z powstałych w ten sposób fragmentów twarzy świadek składał wizerunek sprawcy [1].

Mimo iż portret pamięciowy jest jedną z najstarszych metod badawczych stosowanych w kryminalistyce, to w dalszym ciągu stworzenie takiego, który w odpowiednim stopniu odzwierciedla rzeczywisty wygląd opisywanej osoby, sprawia dużą trudność.

Podczas identyfikacji sprawców przestępstw przez naocznych świadków najważniejszą rolę odgrywa:

- rozpoznawanie twarzy – umiejętność odpowiedzi na pytanie: czy oraz kiedy i gdzie dana osoba została poznana,

- reprodukcja twarzy – zdolność do słownego opisania wyglądu każdej z jej części.

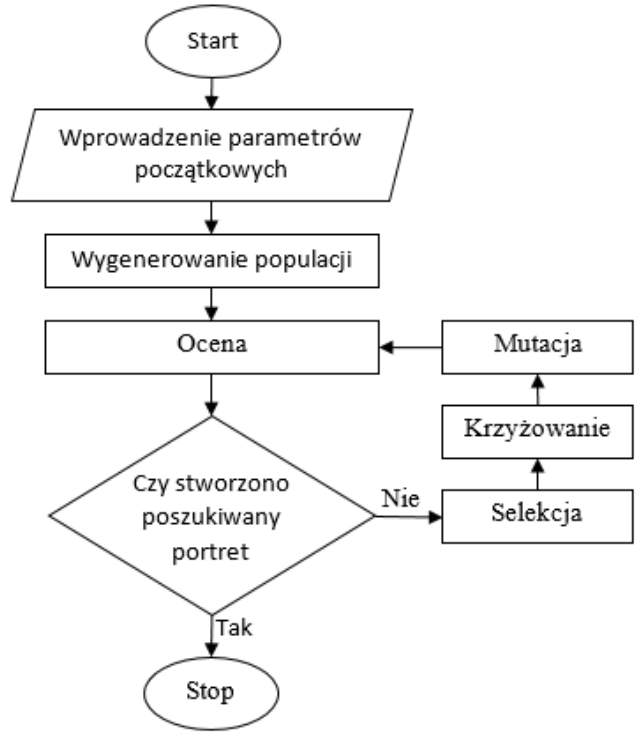
W codziennym funkcjonowaniu znacznie częściej stajemy przed koniecznością rozpoznania osoby niż przed zwerbalizowaniem jej wyglądu. Opis sprawcy pozwalający na reprodukcję wizerunku z perspektywy poznawczego funkcjonowania człowieka jest bardzo trudnym zadaniem pamięciowym, ponieważ wymaga od świadka przywołania z pamięci wyglądu sprawcy i jego słownego wyrażenia. Reprezentacja umysłowa twarzy ma charakter wizualny, zaś reprodukcja wymaga zamiany informacji wyrażonych w kodzie obrazowym, na werbalny opis elementów wyglądu. Jest ona z natury holistyczna i konfiguracyjna. To coś więcej niż prosta suma poszczególnych części. Reprodukcja wymaga przełożenia wizerunku na odrębne, niezależne elementy, co jest sprzeczne z regułami jego kodowania i przechowywania [2].

Zastosowanie algorytmu genetycznego umożliwia stworzenie portretu pamięciowego nawet przez świadka, który nie jest w stanie określić, który spośród listy wzorców określających konkretną część twarzy jest najbardziej odpowiadający. Jest to możliwe, ponieważ od użytkownika nie wymaga się przełożenia zapamiętanego wizerunku na odrębne, niezależne elementy. Świadkowi zaprezentowane zostają przykładowe portrety, a jego zadaniem jest jedynie rozpoznawanie twarzy i określanie stopnia jej podobieństwa [3, 4, 5].

2. Opis zaimplementowanego algorytmu genetycznego

Aby stworzyć portret pamięciowy bez konieczności wyboru poszczególnych elementów twarzy niezbędne jest stworzenie algorytmu, który zminimalizuje liczbę ocenianych portretów konieczną do wygenerowania wizerunku poszukiwanej osoby. Przy odpowiednio małej przestrzeni rozwiązań, można zastanowić się nad klasyczną metodą, czyli sprawdzeniem każdego możliwego rozwiązania. Zakładając, że obrazy z których tworzony jest portret pamięciowy zostaną podzielona na 9 grup (owal twarzy, włosy, brwi, oczy, uszy, nos, wąsy, usta, broda) i każda z nich będzie liczyć zaledwie po 5 elementów, to aby wybrać najlepsze odwzorowanie wizerunku twarzy poszukiwanej osoby należy ocenić 5^9 (czyli niemal 2 miliony) portretów pamięciowych. Nawet przy tak nielicznej bazie wzorców efektywność metody

iteracyjnej okazuje się zbyt mała. Do rozwiązania tego problemu zastosowany zastał algorytmu genetyczny. Umożliwił on przeszukanie przestrzeni alternatywnych rozwiązań i wyszukanie odpowiednio dobrego rozwiązania, czyli wygenerowanie portretu pamięciowego który w wystarczająco dużym stopniu odwzorowuje wygląd poszukiwanej osoby. Schemat zaimplementowanego algorytmu został przedstawiony na poniższym rysunku (Rysunek 1).



Rysunek 1. Schemat algorytmu genetycznego

Podstawowe pojęcia:

Gen – pojedyncza cecha osobnika reprezentowana przez obraz przedstawiający część twarzy.

Chromosom – uporządkowany ciąg genów. Zbiór wszystkich części twarzy.

Osobnik – pojedyncza propozycja rozwiązania problemu. Jest on zakodowany przez 1 chromosom. Przedstawia portret pamięciowy.

Populacja – zbiór osobników o określonej liczebności.

W parametrach początkowych ustalane są cechy generowanego portretu pamięciowego oraz parametry algorytmu genetycznego. Użytkownik wprowadza płeć poszukiwanej osoby oraz określa czy portret pamięciowy ma zawierać elementy twarzy takie jak włosy, wąsy, broda. W zależności od wartości ustalonych na tym etapie, chromosom zawiera od 6 do 9 genów, które reprezentują poszczególne elementy twarzy. Podczas ustawiania parametrów algorytmu genetycznego określany jest sposób selekcji, liczebność populacji, procentową wartość współczynnika krzyżowania oraz procentowy udział osobników powstałych w wyniku mutacji.

Funkcja oceny ma na celu przypisanie wartości liczbowej, która określa przystosowanie każdego osobnika populacji. Użytkownik przy użyciu dziesięciostopniowej skali, ocenia podobieństwo wygenerowanego portretu do poszukiwanego wizerunku sprawcy.

Selekcja polega na wybraniu z populacji tych osobników, które będą brały udział w krzyżowaniu oraz mutacji. Podobnie jak w naturalnej selekcji, osobniki z większą wartością funkcji przystosowania, mają większe szanse na przejście do etapu krzyżowania. Selekcja w zależności od ustawień początkowych odbywa się jedną z trzech metod:

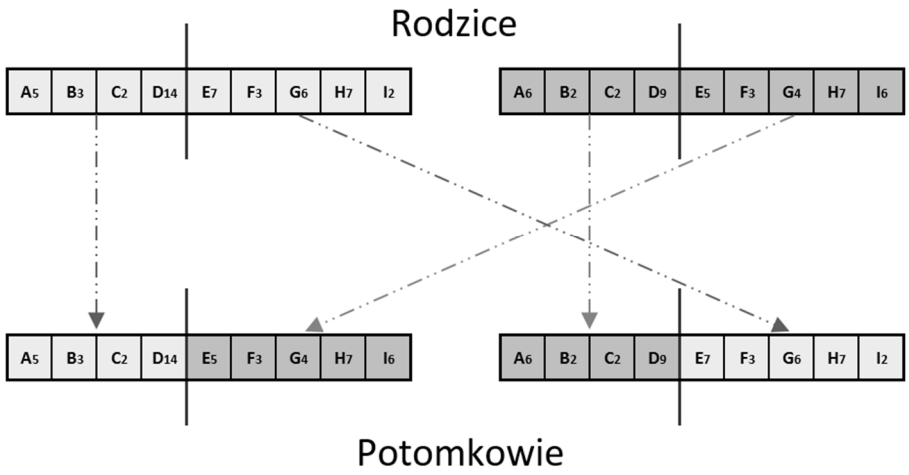
- Ranking – osobniki sortowane są malejąco według wartości jaką otrzymały w funkcji oceny. Pierwszy na liście przypisany jest do grupy biorącej udział w krzyżowaniu.
- Turniej – w sposób losowy wybieranych jest kilka osobników z populacji. Najlepszy z nich przypisywany jest do grupy biorącej udział w krzyżowaniu.
- Ruletka – losowany jest osobnik z prawdopodobieństwem równym ilorazowi funkcji oceny do sumy wartości funkcji oceny wszystkich osobników populacji.

Proces powtarzany jest do osiągania żądanej liczby osobników, która wynika z iloczynu współczynnika krzyżowania i liczebności populacji.

Zadaniem operacji krzyżowania jest wymiana informacji zapisanych w genach pomiędzy osobnikami. W opisywanym algorytmie zastosowane zostało

krzyżowanie jednopunktowe. Proces krzyżowania można podzielić na następujące etapy:

- Losowy wybór par chromosomów.
- Wyznaczenie punktu krzyżowania – w sposób losowy, w tym samym miejscu dla obydwu chromosomów.
- Wymiana genów pomiędzy chromosomami



Rysunek 2. Schemat krzyżowania jednopunktowego

Chromosomy biorące udział w krzyżyżowaniu wybierane są w sposób losowy z grupy osobników powstałej w wyniku selekcji. W wyniku procesu krzyżowania tworzona jest nowa populacja składająca się z unikalnych osobników [6]. W przypadku gdy urozmaicenie populacji jest na tyle niskie, że w wyniku krzyżowania nie da się stworzyć określonej liczby unikalnych osobników potomnych, losowy osobnik z grupy biorącej udział w krzyżowaniu zostaje poddany mutacji.

Zadaniem operatora mutacji jest zapewnienie zmienności chromosomów, czyli stworzenie możliwości wyjścia procedury optymalizacji z ekstremów lokalnych funkcji przystosowania [7].

Proces mutacji można podzielić na następujące etapy:

- Losowy wybór osobnika z populacji powstałej w wyniku krzyżowania

- Wylosowanie genu
- Przypisanie nowego wzorca wylosowanego z bazy.

Liczba osobników poddawanych mutacji jest wyliczana na podstawie danych wprowadzonych w parametrach początkowych.

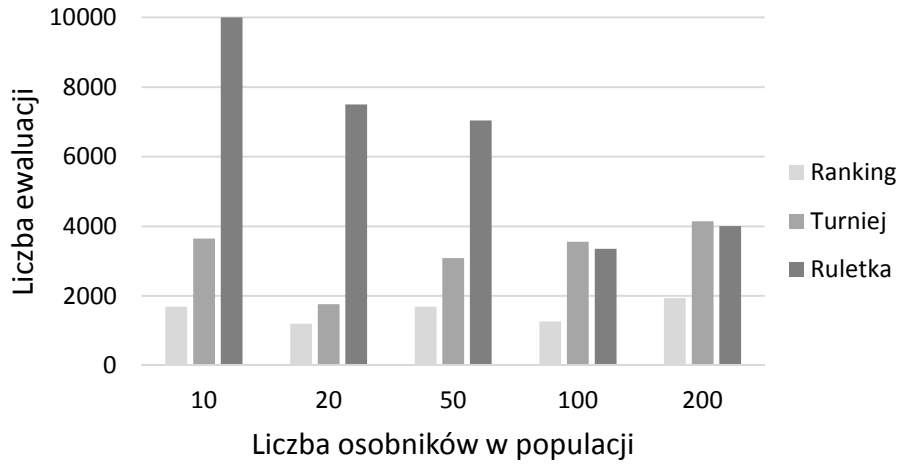
3. Badania ustawień algorytmu genetycznego

Celem przeprowadzonych badań jest minimalizacja liczby ewaluacji niezbędnej do wygenerowania poszukiwanego portretu pamięciowego. W opracowanej metodzie komputerowej wykorzystującej algorytm genetyczny do odtwarzania portretu pamięciowego każdy wygenerowany osobnik jest oceniany przez człowieka. Z tego powodu bardzo istotne jest ustalenie parametrów algorytmu genetycznego umożliwiających stworzenie portretu pamięciowego przy jak najmniejszej liczbie ocenianych wizerunków.

Poszukiwany portret pamięciowy został ustalony w sposób losowy z elementów znajdujących się w bazie wzorców. Przedstawia on twarz mężczyzny nieposiadającego wąsów i brody. Liczebność bazy wzorców umożliwia stworzenie ponad 450 milionów unikalnych wizerunków. W przeprowadzonych badaniach każdy z wygenerowanych osobników został poddany funkcji oceny. Jej zadaniem jest przydzielenie wartości liczbowej z przedziału od 0 do 9 każdemu wygenerowanemu osobnikowi proporcjonalnie do liczby genów odpowiadających cechom poszukiwanego portretu pamięciowego. Najmniejsza wartość przydzielona jest w przypadku gdy żaden z genów ocenianego osobnika nie odpowiada genom poszukiwanego portretu pamięciowego, zaś największa osobnikowi którego wszystkie cechy są tożsame z genami szukanego portretu. Dla każdego z ustawień algorytmu genetycznego przeprowadzonych zostało 5 badań. Przedstawione wyniki przyjmują uśrednione wartości z przeprowadzonych badań.

3.1. Wyznaczenie sposobu selekcji

Poniższy wykres (Rysunek 3) przedstawia zależność liczby ewaluacji od liczebności populacji dla trzech rodzajów selekcji. Współczynnik krzyżowania wynosi 50%, wartość mutacji 20%.



Rysunek 1. Wykres zależności liczby ewaluacji od liczebności osobników w populacji

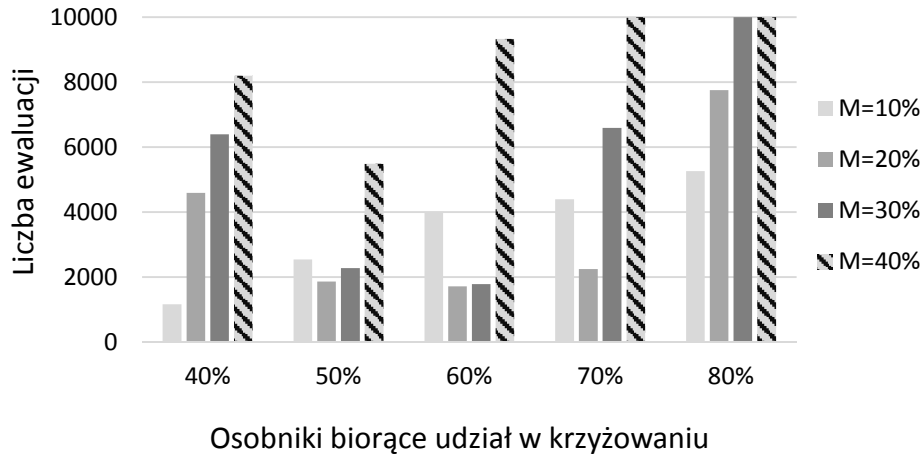
Wnioski:

1. Spośród trzech typów selekcji liczba ocenionych wizerunków niezbędna do wygenerowania poszukiwanego portretu pamięciowego jest najmniejsza dla selekcji rankingowej.
2. Optymalna liczebność populacji dla selekcji rankingowej wynosi 20 osobników.

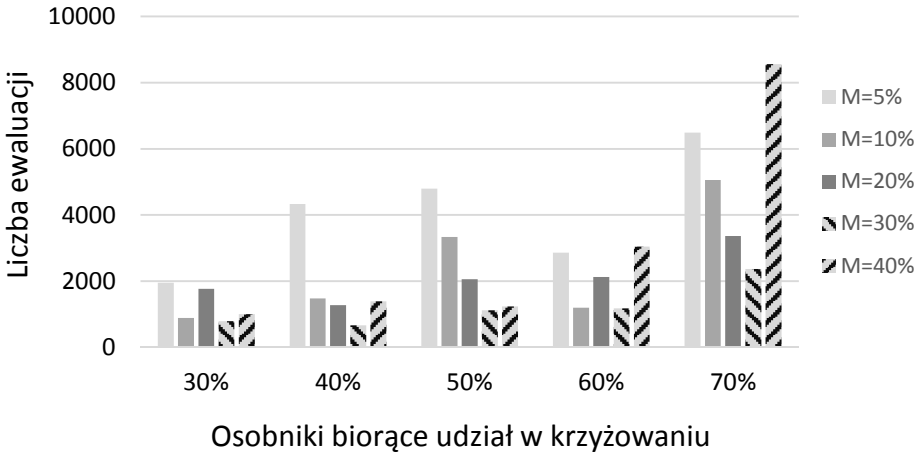
3.2. Wyznaczenie liczebności populacji, współczynnika krzyżowania oraz mutacji

Po analizie wyników badań przedstawionych w punkcie 3.1 badania mające na celu wyznaczenie współczynnika krzyżowania oraz procentowy udział w populacji osobników powstałych w wyniku mutacji zostały przeprowadzone dla selekcji rankingowej oraz populacji liczącej 10, 20 oraz 50 osobników. Większa liczebność populacji mogłaby doprowadzić do obciążenia poznawczego.

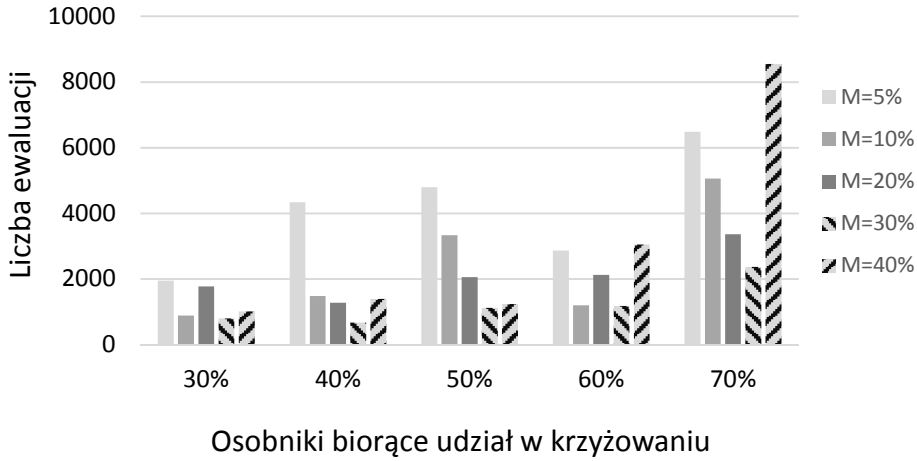
Poniżej w formie wykresów przedstawione zostały wyniki przeprowadzonych badań.



Rysunek 4. Wykres zależności liczby ewaluacji od współczynnika krzyżowania dla populacji liczącej 10 osobników



Rysunek 5. Wykres zależności liczby ewaluacji od współczynnika krzyżowania dla populacji liczącej 20 osobników

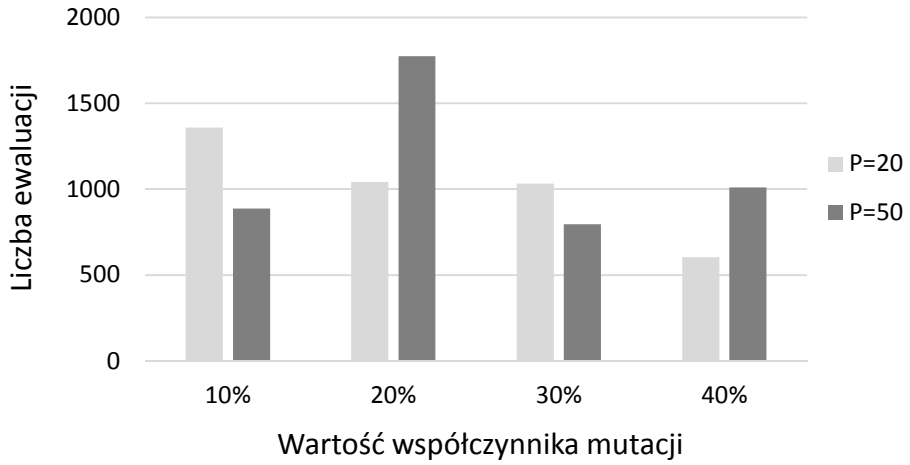


Rysunek 6. Wykres zależności liczby ewaluacji od współczynnika krzyżowania dla populacji liczącej 50 osobników

Wnioski:

1. Liczba ewaluacji przyjmuje najmniejsze wartości dla współczynnika krzyżowania wynoszącego 30%.
2. Najlepsze wyniki osiągane są dla populacji liczącej 20 i 50 osobników.
3. Współczynnik mutacji powinien przyjmować wartość z zakresu 10-40%.

Na poniższym wykresie (Rysunek 7) przedstawione zostały wyniki badań dla ustawień algorytmu genetycznego opisanych we wnioskach.



Rysunek 7. Wykres zależności liczby ewaluacji od wartości współczynnika mutacji

Analiza przeprowadzonych badań umożliwiła ustalenie optymalnych ustawień algorytmu genetycznego:

- Liczebność populacji – 20 osób
- Wartość współczynnika mutacji – 40%
- Wartość współczynnika krzyżowania – 30%
- Rodzaj selekcji – Ranking

4. Podsumowanie

Do tworzenia portretu pamięciowego coraz częściej wykorzystywane są techniki komputerowe. Zastosowanie w nich algorytmu genetycznego umożliwia przeszukanie przestrzeni alternatywnych rozwiązań, w wyniku czego możliwe jest stworzenie poszukiwanego wizerunku bez konieczności wyboru poszczególnych elementów twarzy. Celem niniejszego artykułu było ustalenie ustawień algorytmu genetycznego umożliwiające wygenerowanie poszukiwanego portretu pamięciowego przy jak najmniejszej liczbie ewaluacji. Przedstawiony został sposób i wyniki przeprowadzonych badań. Obecnie trwają prace nad przeprowadzeniem

bardziej szczegółowych badań które umożliwią dokładniejsze zoptymalizowanie ustawień zaimplementowanego algorytmu.

Bibliografia

1. Krawczyńska A., *Odtworzyć wygląd*, Policja 997, nr 5(50), 05.2009r, s. 16-17
2. Kabzińska J., *Przeszłość, teraźniejszość i przyszłość obrazowego portretu pamięciowego,* w: *III Dni Kryminalistyki Wydziału Prawa i Administracji Uniwersytetu Rzeszowskiego.*, Rzeszów 09.2009r, s. 131-140
3. Schreiber P., Kovac M., Moravcik O., *Using Genetic Algorithms for Identikit Creation*, w: Proceedings of the World Congress on Engineering and Computer Science 2012, Newswood Limited, 2012r, s. 363–368
4. Gibson S., Bejarano A., Solomon C., *Synthesis of Photographic Quality Facial Composites using Evolutionary Algorithms*, *Proceedings of the* British Machine Vision Conference 2003r, s. 221-230
5. Frowd, C., Skelton, F., Hancock, J., *Evolving an identifiable face of a criminal,* The Psychologist vol 25, February 2012r, s. 116 – 119
6. Beasley D., Bull D. R., *An Overview of Genetic Algorithms: Part 1, Fundamentals*, w: University Computing, volume 15, 1993r, s. 58–69
7. Narayana Rao T. V., Madiraju S., *Genetic Algorithms and Programming-An Evolutionary Methodology,* w: International Journal of Computer Science and Information Technologies, volume 1, Tech science publications, 2010r, s. 427–437

Streszczenie

W artykule przedstawiony został sposób wykorzystania algorytmu genetycznego do tworzenia portretu pamięciowego metodą komputerową. Zastosowanie metod sztucznej inteligencji umożliwiło wygenerowanie portretu pamięciowego bez konieczności wyboru przez użytkownika poszczególnych elementów twarzy. Zadanie świadka odtwarzającego portret pamięciowy sprowadza się w tym wypadku do określenia stopnia podobieństwa przedstawianych wizerunków do poszukiwanej osoby. Szczegółowo opisane zostały najistotniejsze etapy algorytmu, czyli

określenie cech populacji początkowej, funkcja oceny, selekcja oraz operatory genetyczne – krzyżowanie oraz mutacja. Przeprowadzone zostały badania na przykładowej bazie wzorców zawierającej takie elementy jak: owal twarzy, włosy, brwi, oczy, uszy, nos i usta. Analiza ich wyników umożliwiła optymalizację ustawień parametrów algorytmu genetycznego.

Abstract

The article shows the way of implementation of the genetic algorithm for computer-based creation of photofits. The usage of artificial intelligence has made it possible to generate photofits without the user needed to be requested for selecting particular elements of a face. Determining the degree of similarity between the images shown and the actual appearance of the wanted person can now be considered as the one sole task of a witness who is trying to reproduce a photofit picture. The fundamental steps of the algorithm are described in the article, that is the determination of the attributes of the original population, evaluation function, selection and genetic operators – crossing and mutation. Research has been conveyed on an exemplary set of patterns which include following elements: oval of a face, ears, eyes, eyebrows, hair, lips and nose. The analysis of the outcome of the investigation has made it possible to optimize the settings of the genetic algorithm.

Keywords: Genetic algorithm, creation of photofits

Małgorzata Śliwa
Instytut informatyki i zarządzania produkcją,
Wydział Mechaniczny, Uniwersytet Zielonogórski
ul. prof. Z. Szafrana 4, 65-516 Zielona Góra
M.Sliwa@iizp.uz.zgora.pl

Model konwersji wiedzy ukrytej w wiedzę jawną przy zastosowaniu algorytmu Bayes'a na przykładzie działu badawczo-rozwojowego w przedsiębiorstwie produkcyjnym

1. Wstęp

Wiedza firmy jest zasobem przedsiębiorstwa, który może stanowić o jego sukcesie na rynku w dynamicznej i konkurencyjnej gospodarce. Według Druckera wiedza to zasób, będący podstawą do podejmowania działań przedsiębiorstwie, w większości przypadku innowacyjnych [1]. Różnice wartości księgowej, a rynkowej przedsiębiorstwa występują właśnie w firmach innowacyjnych [2], w których sama infrastruktura i zaplecze technologiczne bez zastrzeżonych receptur czy specjalistycznej wiedzy pracowniczej nie mogą funkcjonować. Wartość organizacji zależy ściśle od pracowników i ich wiedzy.

Pozyskanie wiedzy jawnej (ang. explicite knowledge) to zbiór działań polegający na gromadzeniu, przetwarzaniu i przekazywaniu informacji wiedzy zapisanej w postaci reguł, procedur, instrukcji. Natomiast systematyzacja i zapis wiedzy ukrytej (ang. tacit knowledge) jest procesem wymagającym zaangażowania i motywowania pracowników firmy [3]. Trudności związane z pozyskiwaniem wiedzy ukrytej, utożsamianej głównie z doświadczeniem pracownika wiedzy, zachęcają do opracowywania przez przedsiębiorstwo strategii wspomagających eksternalizację.

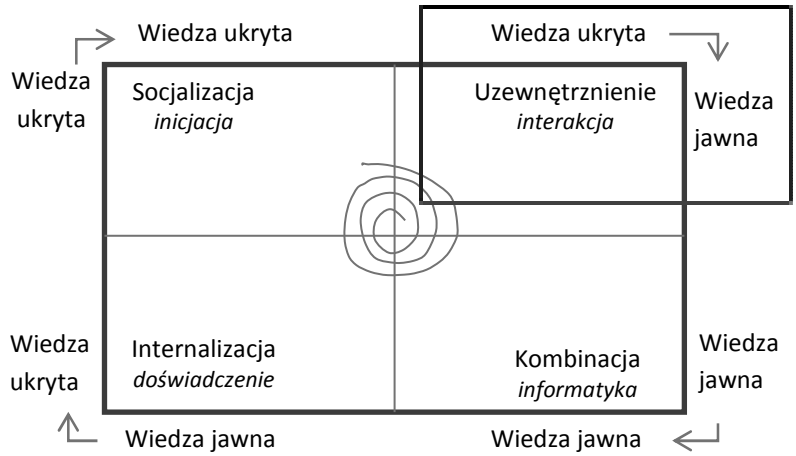
W artykule podjęto próbę zbudowania modelu konwersji wiedzy ukrytej w wiedzę jawną przy zastosowaniu algorytmu Bayes'a dla działu badawczo-rozwojowego w przedsiębiorstwie produkcyjnym. Na podstawie literatury przedmiotu dokonano charakterystyki procesu konwersji wiedzy. W tym celu zidentyfikowano źródła wiedzy ukrytej w dziale badawczo-rozwojowym w przedsiębiorstwie produkcyjnym, następnie zaproponowano mechanizmy jej pozyskiwania. Sformułowany model zilustrowano na przykładzie z praktyki gospodarczej. W podsumowaniu pokazano kierunki dalszych prac obejmujące

implementację informatyczną przedstawionego modelu oraz jego weryfikację w przedsiębiorstwach produkcyjnych.

2. Proces konwersji wiedzy ukrytej w wiedzę jawną

W literaturze przedmiotu można znaleźć wiele definicji konwersji wiedzy i dzielenia się nią. Zakładają one zależności między wiedzą jawną a ukrytą związane z charakterystyką pracowników wiedzy: ich edukacją, doświadczeniem zawodowym oraz życiowym [4]. Dalej, w modelu SECI, według I. Nonaki i H. Takeuchi (2000) wyróżnia się cztery sposoby przemiany wiedzy (rysunek 1) [5]:

- wiedza ukryta do ukrytej (socjalizacja),
- od wiedzy ukrytej do dostępnej (eksternalizacja),
- od wiedzy dostępnej do dostępnej (kombinacja),
- od wiedzy dostępnej do ukrytej (internalizacja).



Rys. 1. Przekształcanie wiedzy – model SECI [6]

Proces konwersji wiedzy można zdefiniować analogicznie do procesu eksternalizacji wiedzy, który polega na wyrażeniu wiedzy ukrytej w formie wiedzy jawnej (np. w postaci słowników pojęć, procedur i innych) [7]. Istotne są także czynniki, które mogą pomóc i zachęcić pracownika do udostępnienia wiedzy ukrytej na zewnątrz i podziału nią z organizacją oraz jej członkami. Dzięki identyfikacji wiedzy ukrytej, a następnie próbom jej formalizacji mógłby wzrosnąć kapitał intelektualny firmy, a tym samym jej konkurencyjność na rynku. Usprawniając system pozyskiwania wiedzy, który pozwalałby na zachowanie jej w trwałej formie z punktu widzenia organizacji, można pozytywnie wpłynąć na zoptymalizowanie

nakładów pracy przy prowadzeniu projektów lub badań. Sukcesem w tym wypadku byłyby korzyści płynące m.in. z: redukcji nakładów finansowych przeznaczonych na projekt, optymalizacji zasobów ludzkich, mniejszej liczby poprawek lub reklamacji czy szybsze zakończenie projektu (przed czasem).

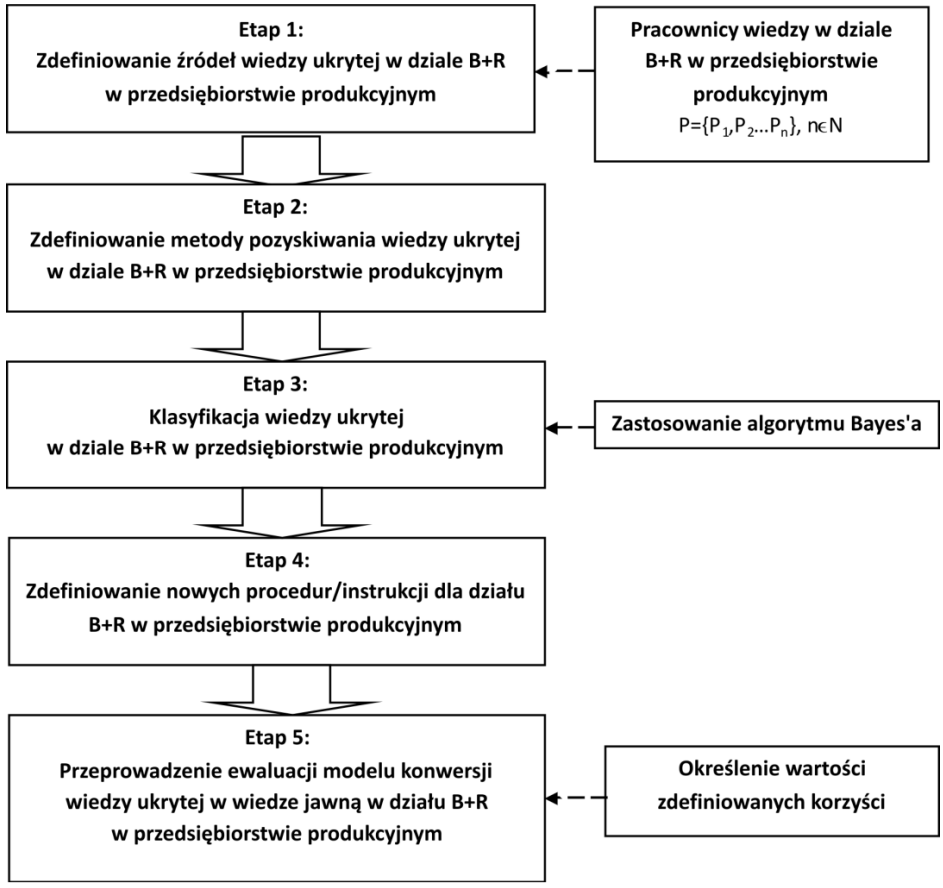
Należałoby w tym wypadku skupić się na trafnej ocenie przydatności danego zespołu pracowniczego przy kluczowym dla organizacji projekcie. Wtedy można twierdzić, że ciągłe doskonalenie wynikające z zachowania wiedzy, wpłynęłaby pozytywnie na dalszą współpracę:

- z nowymi pracownikami: głównie przy wchłanianiu pracownika do zespołu, zapoznaniu z podstawowymi kierunkami działania, z „powszechnie znanym" podejściem do określonych problemów, zgodnie z zasadami organizacji; również przy kompletowaniu potencjalnej kadry umysłowej przez dział HR;
- z podobnymi projektami: przy przekazywaniu podstawowej wiedzy eksperckiej wewnątrz grupy i na zewnątrz (w organizacji), usystematyzowanie sposobu postępowania a także mniej powtarzających się działań i „odkrywania" czegoś na nowo;
- z nowa kadrą zarządzającą: w momencie kompletownia zespołu dedykowanego przedsięwzięciu, tym bardziej przy kluczowych projektach z krótkim czasem realizacji.

M. Morawski podkreśla różnice między źródłem sukcesu w organizacji starego typu (przestrzeganie reguł i zasad, unikanie błędów), a nowego – gdzie liczy się pomysłowość, samorozwój i zaangażowanie [8]. W swoim opracowaniu T. Kowalski powołuje się na twierdzenie, że pracownicy wiedzy niechętnie otrzymują polecenia, mają trudny do jednoznacznego usystematyzowania tryb pracy, a najlepsze rezultaty osiągają przy współpracy z innymi w tzw. sieciach kontaktów. Istotne aby pracownik wiedzy [9]:

- uzyskiwał nowe umiejętności, doświadczenia i kontakty,
- pozyskiwał wiedzę specjalistyczną w wyniku szkoleń, kursów, wymian zagranicznych, potwierdzonych odpowiednim certyfikatem,
- posiadał dostęp do wysoko specjalistycznych informacji i danych,
- wykonywał samodzielnie zadania, dobierając odpowiednie metody,
- opierał pracę zespołową głównie na nieformalnych relacjach i swobodnej dyskusji.

Na podstawie przeprowadzonej analizy literatury przedmiotu zdefiniowano model konwersji wiedzy ukrytej w wiedzę jawną przy zastosowaniu algorytmu Bayes'a dla działu badawczo-rozwojowego w przedsiębiorstwie produkcyjnym:



Rys. 2. Model konwersji wiedzy ukrytej w wiedzę jawną przy zastosowaniu algorytmu Bayes'a dla działu badawczo-rozwojowego w przedsiębiorstwie produkcyjnym

W dalszej części artykułu opisano szczegółowo każdy etap modelu (pp. rys. 2) oraz pokazano praktyczny przykład jego zastosowania.

3. Model konwersji wiedzy ukrytej w wiedzę jawną przy zastosowaniu algorytmu Bayes'a dla działu badawczo-rozwojowego w przedsiębiorstwie produkcyjnym

W ramach etapu pierwszego zdefiniowanego modelu (pp. rys. 2) należy określić źródła wiedzy ukrytej w dziale B+R w przedsiębiorstwie produkcyjnym. Podstawowym źródłem wiedzy w przedsiębiorstwie jest niezmiennie nauka w działaniu, na którą składa się głównie doświadczenie czyli trening i obserwacja

[10]. Stwierdza się, że podstawowym źródłem wiedzy cichej pracownika przedsiębiorstwa jest:

- wiedza zdobyta podczas samodzielnych działań przy realizacji zadań i projektów,
- wiedza i obserwacje nabyte podczas testów i badań,
- analiza reklamacji (wyrobu, procesu, który wydaje się być zgodny),
- feedback, wywiady oraz inne formy sprzężenia zwrotnego z klientem i użytkownikiem produktu, usługi,
- wspólne burze mózgów i przyglądanie się problemowi z innej strony.

Istnieje kilka grup pracowników firmy produkcyjnej, pośród których można zdefiniować pracowników wiedzy. Tworzą je członkowie działu B+R lub biura konstrukcyjnego, ale także działu kontroli jakości, utrzymania ruchu, oraz kadra managerska. Proponowany model opiera się na wyborze pracowników wiedzy według następujących charakterystyk (określonych jako dziedzina X0) [11, 12, 13]:

- altruizm – x_1,
- zaangażowanie – x_2,
- wspólne poznanie – x_3,
- doświadczenie – x_4,
- wykształcenie kierunkowe – x_5,
- dostęp do procedur – x_6,
- dostęp do baz danych – x_7.

Drugi etap modelu (pp. rys. 2) obejmuje zdefiniowanie metody pozyskiwania wiedzy ukrytej w dziale B+R w przedsiębiorstwie produkcyjnym. Poniżej przedstawiono propozycje pozyskania wiedzy cichej od zdefiniowanych pracowników wiedzy:

- nagranie działań doświadczonego pracownika,
- rozmowa ze specjalistą odnośnie tematu/ zagadnienia i kroków postępowania,
- zaangażowanie do działania specjalistów z firm zewnętrznych,
- stworzenie miejsca, w którym pracownicy wiedzy mogą zapisywać swoje spostrzeżenia,
- przygotowanie formularzy do pozyskiwania wiedzy [4].

Przeszkodę w pozyskiwaniu wiedzy od specjalistów stanowi indywidualizm i chęć zachowania autonomii przez pracowników wiedzy [14].

W etapie trzecim proponuje się zastosowanie algorytmu Bayes'a w celu klasyfikacji zebranej wiedzy ukrytej. Zasadnicza weryfikacja poprawności założeń kreowanego modelu nastąpi przy pomocy metody probabilistycznej. Proponowana klasyfikacja

bayesowska stanowi prosty i efektywny system bazujący na statystyce. Dzięki niemu można przewidzieć prawdopodobieństwo warunkowe przynależności badanego obiektu do określonej klasy czy wartości decyzyjnej.

Twierdzenie Bayes'a: $P(C\backslash X) = \frac{P(C\backslash X)P(C)}{P(X)}$

Gdzie: P(C|X) jest tzw. prawdopodobieństwem warunkowym (a posteriori) zdarzenia C: danej klasy wiedzy ukrytej (należenie do klasy C) pod warunkiem zajścia zdarzenia X (posiadanie właściwości X – charakterystyki pracownika wiedzy).

$P(Ci) = s_i / s$

s oznacza liczbę obiektów w zbiorze treningowym: liczba pracowników w dziale B+R przedsiębiorstwa produkcyjnego,

s_i oznacza liczbę obiektów w klasie C_i : liczba pracowników wiedzy w dziale B+R przedsiębiorstwa produkcyjnego,

Dla $X = (x_1, x_2, \ldots, x_n)$,

wartość $P(X\backslash Ci) = P(x_1\backslash Ci) * P(x_2\backslash Ci) * \ldots * P(x_n\backslash Ci)$,

$P(x_k\backslash Ci) = s_{ik}/s_i$,

s_{ik} oznacza liczbę obiektów klasy C_i, dla których wartość atrybutu A_k jest równa x_k,

s_i oznacza liczbę wszystkich obiektów klasy C_i w zadanym zbiorze treningowym.

W modelu założono, że przy określaniu zbioru treningowego należy posłużyć się tabelą ocen (TAK/NIE, gdzie TAK oznacza się przez 1, natomiast NIE oznacza się jako 0) dla poszczególnych parametrów x_n (tab. 1). Układając zbiór treningowy w celu zdefiniowania pracowników wiedzy kierownik projektu bądź wyznaczona osoba (ekspert), ocenia pracownika lub zespół zgadzając się z występowaniem danego parametru, odpowiadając twierdząco (logiczne oznaczenie „1") lub przecząco (logiczne oznaczenie „0").

Tab. 1. Zestawienie ocen dla poszczególnych parametrów

Źródło: opracowanie własne

	X_1	X_2	X_3	X_4	X_5	X_6	X_7
Pracownik Wiedzy 1							
Pracownik Wiedzy 2							
…							
Pracownik Wiedzy n							

Skompletowanie satysfakcjonującej liczby informacji w zbiorze treningowym będzie stanowiło wzorcową bazę danych dla gotowego modelu. Projektowane narzędzie po określeniu odpowiednich parametrów wsadowych mogłoby wspomagać zarządzanie kluczowym projektem i zespołem pracowników wiedzy, przez mniej doświadczoną lub też nową dla danego środowiska osobę zarządzającą.

W etapie czwartym zdefiniowano przekształcenie pozyskanej wiedzy ukrytej w jej formalne opracowanie w postaci materiałów m.in.:

- procedur,
- instrukcji działania,
- skryptu/broszury,
- bazy danych,
- skrypty szkoleń,
- biblioteczki (papierowej, elektronicznej),
- prezentacji multimedialnej.

Według raportu „Sharing knowledge in the Corporate Hive" przytoczonego przez J. Fazlagić'a [4] można wyróżnić następujące korzyści dla firmy w aspekcie pozyskiwania i dzielenia się wiedzą (w nawiasie odsetek respondentów wskazujących na proces):

- efektywniejsza wymiana informacji wśród pracowników – 56%,
- lepsza obsługa klienta – 40%,
- mniej dublujących się działań – 36%,
- mniej wąskich gardeł – 34%,
- integracja jednostek biznesowych – 33%,
- większa produktywność pracowników – 29%,
- wydajniejsze podejmowanie decyzji – 7%.

Należy jednak pamiętać, że organizacja musi być przygotowana na utratę pracownika i strać się pozyskać jak najwięcej wartości dodanej, próbując przechwycić jego wiedzę, a następnie sformalizować. W tym celu podejmuje się strategie zarządzania wiedzą. Gruczyńska-Malec i Rutkowska opisują je między innymi ze względu na rodzaj wiedzy, i tak wyróżnia się [15]:

- strategię kodyfikacji – jest ona nastawiona na pozyskiwanie, archiwizacje, przetwarzanie i transmisję wiedzy, dotyczy wiedzy jawnej, panuje podejście wielokrotnego zastosowania wiedzy, a dominującą relacją jest człowiek-technologia;
- strategię personalizacji – jej celem jest usprawnienie komunikacji, współpracy oraz wzajemnego wsparcia pracowników, usprawnienie

dzielenia się wiedzą i rozwoju; strategia tyczy się wiedzy niejawnej, a dominującą relacją jest człowiek – człowiek.

- strategię pomostową – łączy ona cechy dwóch powyższych strategii, jej celem jest udoskonalenie dostępu do wiedzy jawnej i ukrytej, udokumentowana wiedza jest łączona z wiedzą ludzi, wiedzę udokumentowaną łączy się z wiedzą ludzi, a dominują relacje socjotechniczne.

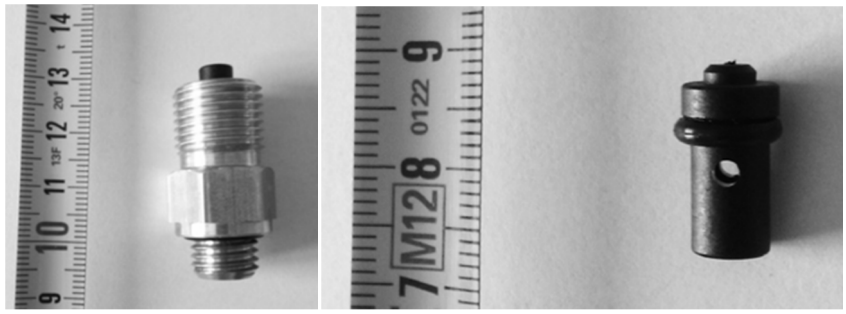
W etapie piątym proponowanego modelu, na podstawie poniższej analizy literatury zdefiniowano korzyści wynikające z jego implementacji w przedsiębiorstwie:

- redukcja kosztów,
- Szybsze zakończenie projektu/ zadania,
- Mniej poprawek i reklamacji do zakończonego projektu,
- Redukcja niezbędnej, skierowanej do zadania Kadry.

Poniżej zaprezentowano praktyczną implementację proponowanego modelu konwersji wiedzy ukrytej w wiedzę jawną przy zastosowaniu algorytmu Bayes'a w dziale badawczo-rozwojowym w przedsiębiorstwie produkcyjnym.

4. Studium przypadku

Rozważany dział badawczo-rozwojowy jest działem funkcjonalnym w przedsiębiorstwie produkcyjnym sektora MSP branży motoryzacyjnej, zajmującej się produkcją elementów pneumatyki stosowanej w układach hamulcowych i zawieszeniach pojazdów użytkowych. Cyklicznie dział B+R otrzymuje nowy projekt od kluczowego klienta. Obecnym problemem są nowe wymagania funkcjonalne (jakościowe), które obejmują produkowany w skali masowej zawór bezpieczeństwa, zwany inaczej test pointem. Wewnątrz mosiężnego korpusu znajduje się tłoczek, którego gumowa (NBR) uszczelka typu o-ring współpracuje z wewnętrzną ścianką korpusu, zapewniając mu tym samym wymaganą szczelność (rys. 3). W niskich temperaturach pracy, guma twardnieje i nie wykazuje wysokiego poziomu odkształcenia, zatem zachowanie normowego poziomu szczelności w obwodzie stanowi problem.



Rys. 3. Zawór bezpieczeństwa będący przedmiotem nowego projektu w dziale B+R w przedsiębiorstwie produkcyjnym

W dziale zatrudnionych jest 6 pracowników. Zakłada się również warunki brzegowe (elementy zbioru D0), w których zawarto również charakterystykę projektu realizowanego w dziale B+R:

D_1: budżet projektu: 150 000 PLN,

D_2: czas realizacji projektu: 6 miesięcy,

D_3: założenia dotyczące projektu: inżynieria odwrotna na podstawie niekompletnego prototypu wraz z poglądowymi rysunkami, ograniczenia normatywne, wytyczne funkcjonalne od klienta,

D_4: tematyka projektu: pneumatyczny zawór kontrolny,

D_5: reprezentant projektu: 15 lat pracy w zespole,

D_6: reprezentant projektu: 16 lat doświadczenia na podobnym stanowisku,

D_7: ilość realizowanych nowych projektów w przedsiębiorstwie: 3-4 /rok.

Zgodnie z przyjętym modelem konwersji wiedzy ukrytej w wiedzę jawną przy zastosowaniu algorytmu Bayes'a w dziale badawczo-rozwojowym w przedsiębiorstwie produkcyjnym przeprowadzono następujące etapy tego modelu:

Etap 1:

Zdefiniowanie źródeł wiedzy ukrytej w dziale B+R w przedsiębiorstwie produkcyjnym

Podczas realizacji nowego projektu pojawią się niespotykane wcześniej sytuacje oraz problematyczne zagadnienia. Zakłada się projekt nowych węzłów konstrukcyjnych, kojarzenie niewspółpracujących do tej pory materiałów, klient zadeklarował nowe wymagania funkcjonalne. Z powodu wcielania nowych rozwiązań technologicznych, rozwiązania generowane problemy nie są opisane w literaturze, w związku z tym, wiedza ta jest pozyskiwana w sposób ciągły. Posługiwanie się narzędziami CAD/CAM pozwoli na stworzenie symulacji i prototypu w pracowni B+R.

Firma w praktyce organizuje i przeprowadza międzywydziałowe spotkania, zatem większość powyższych problemów zostanie rozwiązana w grupie pracowników nie biorących bezpośredniego udziału we wdrażaniu projektu, z działu technologii i jakości, podsuwających swoje pomysły i alternatywne rozwiązania.

Etap 2

Zdefiniowanie metody pozyskiwania wiedzy ukrytej w dziale B+R w przedsiębiorstwie produkcyjnym

Przy pracy nad projektem zaworu, zespół B+R będzie współpracował z grupą naukowców uniwersyteckich specjalizujących się w badaniach tworzyw sztucznych. Prowadzone konsultacje miały za zadanie optymalny dobór materiałów konstrukcyjnych użytych w projekcie. Analizowane przedsiębiorstwo może korzystać równolegle z pomocy firm dostarczających projektowane komponenty, w celu właściwego doboru mieszanki. Kluczowymi parametrami zadanymi przez klienta jest odporność materiału na szeroki zakres temperatur oraz obecność substancji ropopochodnych.

Etap 3

Klasyfikacja wiedzy ukrytej w dziale B+R w przedsiębiorstwie produkcyjnym

Na podstawie doświadczenia w pracy z podobnymi projektami, kierownik działu B+R, wraz z managerem jest w stanie ocenić kadrę inżynierską pod kątem przydatności do realizacji nowego zadania i generacji nowej wiedzy w podobnej dziedzinie – tu: z materiałoznawstwa o tworzywach sztucznych. Został zatem stworzony zbiór danych treningowych przedstawiony w tab. 2, składający się z 10 obiektów. Każdy z badanych pracowników wiedzy musi odnieść się twierdząco lub przecząco do atrybutów x_1-x_7, które określały indywidualne parametry jednostki oraz jej dostęp do informacji. Następnie, doświadczony przełożony stwierdzał czy pracownik był niezbędny przy realizacji projektu, wskazując na przynależność obiektu do klasy.

Tab. 2. Zbiór danych treningowych

Źródło: opracowanie własne

Obiekt	x_1	x_2	x_3	x_4	x_5	x_6	x_7	Przynależność
Pracownik Wiedzy 1	1	1	0	1	1	0	0	1
Pracownik Wiedzy 2	0	1	1	0	0	1	1	1
Pracownik Wiedzy 3	0	0	1	1	1	1	0	0
Pracownik Wiedzy 4	1	1	0	1	0	1	1	1
Pracownik Wiedzy 5	1	1	1	0	1	0	0	0
Pracownik Wiedzy 6	1	0	1	0	1	1	1	0
Pracownik Wiedzy 7	0	1	1	1	1	1	0	1
Pracownik Wiedzy 8	0	1	1	0	1	1	1	1
Pracownik Wiedzy 9	1	1	0	1	1	0	0	0
Pracownik Wiedzy 10	0	1	1	1	0	1	1	1

W momencie oceny przydatności nowego pracownika, należy określić jego atrybuty. Badany obiekt charakteryzuje się poniższymi parametrami:

X=(altruizm: x_1="1", zaangażowanie: x_2="1", wspólne poznanie: x_3="0", doświadczenie: x_4="0", wykształcenie: x_5="1", dostęp do procedur: x_6="1", dostęp do baz danych: x_7="1")

Określenie przynależności bezwarunkowej obiektu do klasy C_i, i=1,2:

$P(C_1)=0,6$

$P(C_2)=0,4$

Obliczanie prawdopodobieństwa $P(X\backslash C_i)$, jest równe iloczynowi poszczególnych prawdopodobieństw warunkowych:

$P(X\backslash C_1)=(2/6)*(6/6)*(2/6)*(2/6)*(3/6)*(5/6)*(4/6)=0,0102$

$P(X\backslash C_1)*P(C_1)=0,0102*0,6=0,00617$

$P(X\backslash C_2)=(3/4)*(2/4)*(1/4)*(2/4)*(4/4)*(2/4)*(1/4)=0,0059$

$P(X\backslash C_2)*P(C_2)= 0,0059*0,4=0,00234$

Pracownik wiedzy został zakwalifikowany do klasy C_1, czyli jego kandydatura jest brana pod uwagę przy kompletowaniu zespołu pracującego nad zleconym zadaniem.

Etap 4:
Zdefiniowanie nowych procedur/instrukcji dla działu B+R w przedsiębiorstwie produkcyjnym

W ramach nowego zadania pozyskano istotne dane z dziedziny materiałoznawstwa. Zakłada się stworzenie formularza, w którym należałoby umieścić kluczowe informacje na temat stosowanej grupy współpracujących materiałów, jak: rozwinięcie składu mieszanki, dopuszczalne tolerancje wymiaru, maksymalną wypływkę, poziom szczelności występujący przy niskich temperaturach pracy, uzupełniony o graficzne przedstawienie zależności.

Etap 5:
Przeprowadzenie ewaluacji modelu konwersji wiedzy ukrytej w wiedze jawną w dziale B+R w przedsiębiorstwie produkcyjnym

Należy zweryfikować przydatność pozyskanej wiedzy jawnej przy pracy nad podobnymi zadaniami. W analizowanym przypadku, grupa wyselekcjonowanych pracowników B+R, winna pozyskać nową wiedzę, która zostanie zachowana w postaci formularzy i odpowiednio zaklasyfikowana w wewnętrznej bazie danych. Na postawie obowiązujących kontraktów z klientem, można twierdzić, że w najbliższym czasie przedsiębiorstwo otrzyma podobne zlecenia. Potwierdzeniem płynących korzyści byłoby oszacowanie: redukcji czasu pracy w wypadku prowadzenia podobnego projektu, oszczędność w budżecie przeznaczonym na projekt, oraz środków poświęconych na badania i konsultacje ze specjalistami.

5. Podsumowanie i wnioski

Wartość organizacji stanowi nie tylko jej zaplecze infrastrukturalne, ale w podobnej mierze posiadana przez nią wiedza wpływająca na wartość rynkową i renomę marki. Przedsiębiorstwa starają się pozyskać wiedzę ukrytą tkwiącą w pracownikach wiedzy, poprzez stosowanie strategii wspomagających konwersję wiedzy. Prócz zapewnienia odpowiednich warunków rozwojowych wspierających podział wiedzy oraz dostępu do baz danych należałoby zwrócić uwagę na nieformalne otoczenie i relacje wewnątrz zespołu pracowniczego. Organizacja poprzez stymulację pracowników umysłowych do wymiany wiedzy cichej, powiększa swoją wartość know-how sformalizowaną wiedzą grupy specjalistów, ujętą w dostępne dla innych procedury. Działania te są istotne choćby przy stracie tzw. „kluczowego specjalisty", doboru nowego zespołu pracowniczego do strategicznego projektu, ale również przy pracy z podobnymi projektami. Po stworzeniu odpowiednio zasobnej bazy danych i optymalizacji proponowanego modelu wspomagającego transfer wiedzy cichej, można by stosować go z powodzeniem w organizacjach komercyjnych opartych na wiedzy. W efekcie

zakłada się otrzymanie wymiernych korzyści dla przedsiębiorstwa produkcyjnego, t.j.: redukcja kosztów, szybsze zakończenie podobnego projektu, redukcję poprawek, reklamacji oraz optymalny dobór kadry. Dalsze badania dotyczą weryfikacji proponowanych modeli w rzeczywistości gospodarczej. Przyjmuje się również, że powyższe działania mogłyby pozytywnie wpłynąć na wzrost konkurencyjności przedsiębiorstw produkcyjnych.

Bibliografia

1. Drucker P., *Managing for Results*, Harper & Row, New York, 1964
2. Bombiak A., *Wycena kapitału intelektualnego na przykładzie Wawel S.A. – studium przypadku*, Zeszyty Naukowe Uniwersytetu Przyrodniczo-Humanistycznego w Siedlcach, Seria: Administracja i Zarządzanie, Nr 95 (2013), s. 229-244
3. Beyer K., *Wiedza jako kluczowy zasób w nowej gospodarce*, Studia i prace Wydziału Nauk Ekonomicznych i Zarządzania Nr 21 (2011), s. 7-16
4. Fazlagić J., *Innowacyjne zarządzanie wiedzą*, Difin, Warszawa, 2014
5. Nonaka I., Takeuchi H., *The knowledge-Creating company. How Japanese Companies Create the Dynamic of Innovation*, Oxford University Press, New York, 1995
6. A. Jashapara, *Zarządzanie wiedzą*, Polskie Wydawnictwo Ekonomiczne, Warszawa 2006, s. 254
7. Bogdanienko J., *W pogoni za nowoczesnością. Wybrane aspekty tworzenia i wprowadzania zmian*, Towarzystwo Naukowe Organizacji i Kierownictwa, Toruń, 2008
8. Kowalski T., *Pojęcie i cechy pracownika wiedzy*, STUDIA LUBUSKIE, Tom VII, Sulechów 2011, dostęp 30.11.2016: http://www.seipa.edu.pl/s/p/artykuly/91/910/Pracownicy%20wiedzy%202011%20GOOD.pdf
9. Morawski M., *Zarządzanie profesjonalistami,* PWE, Warszawa, 2009
10. Mendryk I., *Źródła wiedzy organizacyjnej – wyniki badań polskich przedsiębiorstw, Zeszyty naukowe: Współpraca w łańcuchach dostaw a konkurencyjność przedsiębiorstw i kooperujących sieci*, Nr 32, 2011, s. 315-331
11. Yua Y., Haob J.-X., Dongc X.-Y., Khalifa M., *A multilevel model for effects of social capital and knowledge sharing in knowledge-intensive work team,* International Journal of Information Management 33, 2013, s. 780-790
12. Haua Y.-S., Kimb B., Leec H., Kimc Y.-G., *The effects of individual motivations and social capital on employees' tacit and explicit knowledge sharing intentions*, International Journal of Information Management 33, 2013, s. 356-366

13. Hunga S.-Y., Durcikova A., Lai H.-M., Lin W.-M., *The influence of intrinsic and extrinsic motivation on individuals' knowledge sharing behavior,* Int. J.Human-Computer Studies 69, 2011, s. 415-427
14. Miroński J., *Wyzwania zarządzania wiedzą w zespołach wirtualnych*, e-mentor nr 5 (57), 2014, dostęp [30.11.2016]
http://www.e-mentor.edu.pl/artykul/index/numer/57/id/1142
15. Gruczyńska-Malec G., Rutkowska M., *Strategie zarządzania wiedz. Modele teoretyczne i praktyczne*, Polskie wydawnictwo ekonomiczne, Warszawa, 2013

Abstrakt

W artykule przedstawiono tematykę konwersji wiedzy ukrytej w wiedzę jawną na podstawie działu badawczo-rozwojowego w przedsiębiorstwie produkcyjnym. Sformułowano otoczenie problemu badawczego: w dziale B+R średniego przedsiębiorstwa produkcyjnego istnieje wiedza jawna i ukryta – zgromadzona w pracownikach umysłowych. W dziale realizowane są procesy biznesowe. Należy odpowiedzieć na pytanie: czy zastosowanie narzędzia wspomagającego podział wiedzy pomoże osiągnąć wymierne korzyści dla przedsiębiorstwa?

Zidentyfikowano źródła wiedzy ukrytej w dziale badawczo-rozwojowym w przedsiębiorstwie produkcyjnym, następnie zaproponowano mechanizmy jej pozyskiwania. Zbadano wpływ charakterystyki pracownika na podział wiedzy ukrytej, co wpływa na wzrost know-how przedsiębiorstwa. W konsekwencji zaimplementowano algorytm Bayes'a. Model zilustrowano na przykładzie z praktyki gospodarczej. W efekcie zakłada się otrzymanie wymiernych korzyści z wynikające z podziału wiedzy, jak: redukcję kosztów, poprawek, reklamacji, szybsze zakończenie podobnego projektu i optymalny dobór kadry. W podsumowaniu pokazano kierunki dalszych prac obejmujące implementację informatyczną przedstawionego modelu oraz jego weryfikację.

Słowa kluczowe: wiedza ukryta, algorytmu Bayes'a, konwersja wiedzy, dział badawczo-rozwojowy, przedsiębiorstwo produkcyjne

Abstract:

Based on the reference works, in article have been showed knowledge's conversion characteristic, based on the own research and development department in manufacturing company. Formulated surrounding the research problem: in section B+R medium manufacturing company, there is tacit and explicit knowledge - gathered in the white-collar workers. In this department they are implemented business processes. It should answer the question: whether the use of a tool to support the knowledge sharing will help achieve tangible benefits for the company?

It studied the effect of the characteristics of an employee on the tacit knowledge sharing, which increases on its know-how value in organization. The sources of tacit knowledge in the research and development department in a manufacturing company were identified, and then mechanisms of its collection were proposed. Consequently, Bayes' algorithm was implemented. The model is illustrated by the example of business practice. As a result, it is assumed to receive measurable benefits, i.e. cost reduction, corrections, complaints and faster completion of a similar project, optimal selection of workers. In summary it presents directions for further work, including the IT implementation of the presented model and its verification.

Key words: tacit knowledge, Bayes algorithm, knowledge's conversion, research and development, manufacturing company

Marcin Piłat
Łukasz Sobaszek
Łukasz Wojciechowski
Instytut Technologicznych Systemów Informacyjnych
Wydział Mechaniczny, Politechnika Lubelska
20-618 Lublin, ul. Nadbystrzycka 36
marcin.pilat@gmail.com
l.sobaszek@pollub.pl
l.wojciechowski@pollub.pl

Porównanie rezultatów tworzenia modeli cyfrowych za pomocą skanerów 3D

1. Wstęp

Obrazowanie 3D obejmuje wszechstronne oraz nowoczesne rozwiązania przestrzennego odwzorowywania obiektów rzeczywistych znajdujące zastosowanie w wielu obszarach – m.in. w przemyśle maszynowym do inżynierii odwrotnej, w archiwizowaniu zbiorów dzieł sztuki, w medycynie do planowania zabiegów chirurgicznych i innych [1, 6, 9, 11, 13]. Wraz z rosnącą popularnością technologii 3D coraz częściej spotyka się producentów oferujących szeroki asortyment skanerów 3D. Duża konkurencyjność sprawia, iż firmy produkujące sprzęt tego typu proponują coraz nowocześniejsze rozwiązania [10].

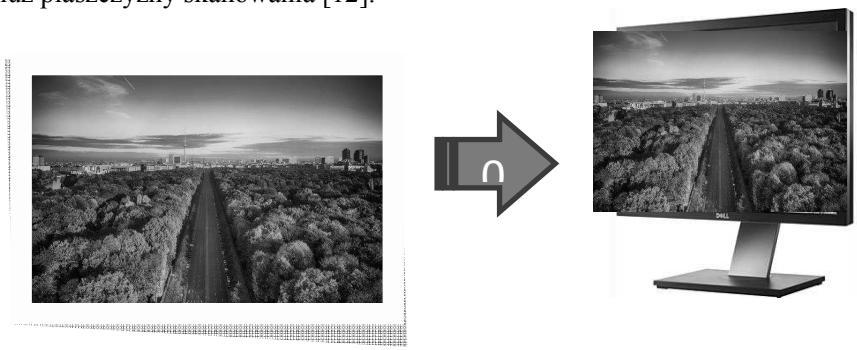
Celem niniejszej pracy jest ocena rezultatów procesu skanowania 3D, a także analiza parametrów wykorzystanych skanerów. W artykule przedstawiono oraz przeanalizowano rezultaty tworzenia modeli cyfrowych za pomocą ręcznych skanerów 3D. Zakres pracy obejmuje omówienie idei skanowania 3D, przedstawienie charakterystyk wykorzystanych skanerów firmy Artec oraz szczegółowe zaprezentowanie metodyki procesów skanowania, obróbki modeli cyfrowych oraz analizę otrzymanych rezultatów.

2. Skanowanie 3D

Idea skanowania 3D jest rozwinięciem idei skanowania 2D. Skanowanie 2D działa na zasadzie odbicia światła od płaszczyzny. Odbite od obrazu światło pada na elementy optyczne głowicy skanera i jest następnie rekonstruowane przez procesor na postać cyfrową (rysunek 1) [12].

Do zamiany obrazu rzeczywistego na jego cyfrowy odpowiednik skaner musi posiadać odpowiednie elementy budowy: źródło światła białego (diody lub

świetlówka), lustra potrzebne do obijania padającego światła, skupiające promienie soczewki, elementy CCD (Charge-Coupled Device) przechwytujących odbite światło, przetwornik ADC (Analog-Digital Converter) zamieniający sygnał z elementów CCD na cyfrowy i silnik krokowy, przesuwający wszystkie elementy wzdłuż płaszczyzny skanowania [12].



Rys. 1. Idea skanowania 2D [opracowanie własne]

Jednym z ograniczeń klasycznych skanerów 2D jest możliwość przenoszenia do postaci cyfrowej jedynie płaskich obrazów. Skanery 3D oprócz przenoszenia obrazu przenoszą również model skanowanego przedmiotu. Tylko przy użyciu skanerów 3D możliwe jest zobaczenie na monitorze komputera rzeczywistego przedmiotu w postaci trójwymiarowego modelu. Skaner tworzy cyfrowy obraz pobierając dane geometryczne z powierzchni przedmiotu. W zależności od zasady działania skanery 3D dzieli się na dotykowe oraz bezdotykowe [8].

Skanery dotykowe tworzą cyfrowy model przesuwając głowicę pomiarową po powierzchni skanowanego elementu. Jest więc to metoda inwazyjna, mogąca uszkodzić skanowany przedmiot. Wykorzystuje się do tego sondę pomiarową umieszczoną na maszynie współrzędnościowej lub ramieniu pomiarowym. Jest to zalecana metoda skanowania, jeżeli konieczne jest bardzo dokładne zeskanowanie powierzchni i ewentualne uszkodzenie powierzchni nie będzie posiadało istotnego wpływu [8].

Skanery bezdotykowe wysyłają wiązkę promieni na skanowany obiekt, a następnie za pomocą kamery rejestrują punkty odbicia światła (rys. 2). Punkty te tworzą chmurę punktów, która jest przetwarzana na część cyfrowego modelu. Jest to metoda bezinwazyjna nie wymagającą kontaktu skanera ze skanowanym przedmiotem. Jest ona niezastąpiona w przypadku skanowania bardzo delikatnych przedmiotów [1, 7, 8].

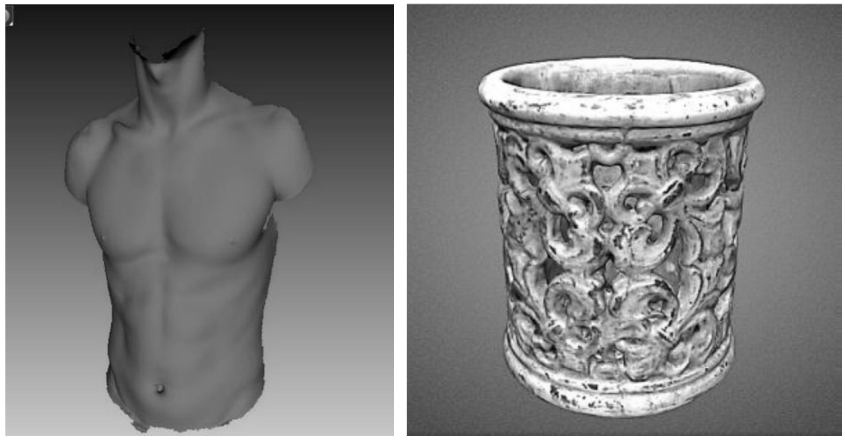
Rys. 2. Skanowanie 3D metodą bezdotykową [5]

3. Specyfikacja skanerów firmy Artec

Artec EVA prezentowany jest przez producenta jako najszybszym i bezkonkurencyjnym skaner do trójwymiarowego odwzorowywania przedmiotów, a zwłaszcza ludzi. Klatki skanowania są wyrównane automatycznie i sprawiają, że skanowanie jest łatwe i szybkie. Cały proces odbywa się w czasie rzeczywistym. Jest to szczególnie ważne podczas badań medycznych i biomechanicznych. Ze względu na wysoką jakość odwzorowania tekstury, modele mogą być wykorzystawane w takich branżach jak animacja komputerowa, kryminalistyka czy medycyna [3, 10]. Jest również najlepszym narzędziem do bezpiecznej i bezdotykowej digitalizacji zbiorów muzealnych. Pozwala opracować dokładną dokumentację mierzonego obiektu, a następnie stworzyć jego kopię za pomocą techniki druku 3D [2, 3]. Pozwala on również tworzyć trójwymiarowe wizualizacje, które następnie można umieścić na stronie internetowej. Wyniki skanowania mogą być dalej przetwarzane za pomocą popularnych narzędzi służących do edycji i obróbki modeli 3D [3].

Artec Spider jest skanerem skonstruowanym z myślą o technice CAD. Jest przeznaczony do inżynierii odwrotnej, projektowania produktu, kontroli jakości i produkcji masowej. Skaner Spider jest najbardziej efektywny podczas odzwierciedlania obiektów małej wielkości (charakteryzujących się skomplikowanymi szczegółami, ostrymi krawędziami, cienkimi żebrami). Świetnie nadaje się do kontroli jakości produktów. Wysoka rozdzielczość i dokładność sprawiają, że jest to idealne narzędzie pomiarowe pozwalające spełnić wymagania dotyczące jakości i wyeliminować błędy produkcyjne [4, 10]. Artec Spider umożliwia także tworzenie scen grafiki komputerowej dzięki znakomitemu poziomowi odwzorowania kolorów i tekstur (rys. 3). Uzyskane w procesie skanowania skanerem Spider modele mogą być wykorzystane w wielu gałęziach przemysłu [2, 11].

Porównanie parametrów technicznych zastosowanych skanerów zostało przedstawione w tabeli 1.



Rys. 3. Zastosowania skanerów podane przez producenta [10]

Tab. 1. Specyfikacja skanerów firmy Artec [2, 3, 4]

	Artec Eva	Artec Spider
Zapis tekstury	TAK	TAK
Rozdzielczość 3D, do	0.5 mm	0.1 mm
Dokładność punktu 3D, do	0.1 mm	0.05 mm
Dokładność 3D na odległość, do	0.03% na 100 cm	
Rozdzielczość tekstury	1.3 mp	1.3 mp
Kolory	24 bpp	24 bpp
Źródło światła	Lampa LED – światło białe	Lampa LED – światło niebieskie
Liniowe pole widzenia (najmniejszy zakres)	214 mm x 148 mm	90 mm x 70 mm
Liniowe pole widzenia (największy zakres)	536 mm x 371 mm	180 mm x 140 mm
Kątowe pole widzenia	30x21°	30x21°
Odległość pracy	0.4 – 1 m	0.17 – 0.3 m
Klatek video, do	16 fps	7.5 fps
Czas ekspozycji	0.0002 s	0.0005 s
Prędkość akwizycji danych, do	2 000 000 punktów/s	1 000 000 punktów/s

4. Stanowisko badawcze

W celu uzyskania jak najbardziej miarodajnych wyników pomiaru, zastosowano kilka rozwiązań mających na celu wyeliminowanie negatywnego wpływu ewentualnych czynników losowych podczas przeprowadzania eksperymentu (rys. 4.).

Pierwszym z nich było zastosowanie statywu na którym umieszczane były skanery. Oba skanery są skanerami ręcznymi co oznacza, że wprawa operatora skanera ma istotny wpływ na późniejsza jakość zeskanowanych modeli. Najdrobniejsze wahania bądź drżenia przekładają się w istotny sposób na rezultaty procesu skanowania.

Drugim z nich było wykorzystanie obrotowego stolika pomiarowego. Zarówno skaner Spider jak i Eva nie posiadają wbudowanego stolika obrotowego. Niezbędne zatem było jego wykonanie w celu zapewnienia osiowego obrotu skanowanych elementów i uzyskania miarodajnych rezultatów. Zaprojektowano oraz wytworzono metodą Rapid Prototyping stolik obrotowy na którym umieszczane były skanowane przedmioty. Stolik został zaprojektowany w programie Solid Edge, a następnie przekonwertowany do formatu STL, dzięki czemu możliwe było jego wydrukowanie na drukarce 3D. Skanery zostały umieszczone w takiej odległości od skanowanych elementów, aby zapewnić jak najlepsze odwzorowanie elementów podczas procesu skanowania (podgląd na monitorze komputera).

Rys. 4. Statyw oraz obrotowy stolik do skanowania [opracowanie własne]

5. Elementy poddane procesowi skanowania

W celu zbadania rezultatów tworzenia modeli cyfrowych przy pomocy skanerów Eva oraz Spider wykorzystano dwa osiowosymetryczne obiekty.

Pierwszym z nich jest tłok samochodowy o metalicznej teksturze. Posiada on otwór przelotowy oraz trzy rowki pod pierścienie. W celu uzyskania jak najlepszych rezultatów został on pokryty talkiem w celu zapobiegnięcia odbijania padających na jego powierzchnie promieni lasera, co powodowało niemożliwość uzyskania obiektu cyfrowego. Skanowany tłok został poddany procesowi odwzorowywania wyłącznie zewnętrznej część elementu, gdyż wnętrze obiektu charakteryzowało się zbył złożoną budową.

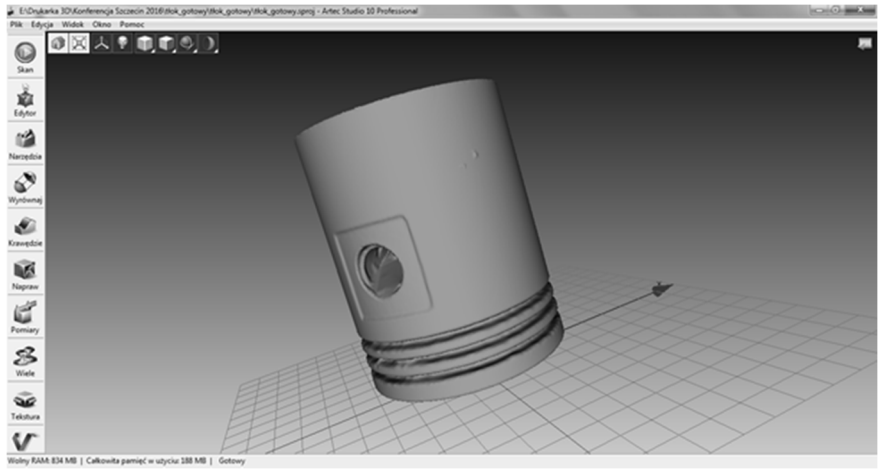
Drugim ze skanowanych obiektów jest drewniany totem do gry, posiadający nadrukowany napis na powierzchni czołowej. Miał on budowę walca o zmiennych średnicach (rys. 5).

Rys. 5. Skanowane przedmioty: tłok oraz totem do gry [opracowanie własne]

6. Porównanie otrzymanych rezultatów

Do uzyskania cyfrowych modeli wykorzystano dostarczone wraz ze skanerami oprogramowanie Artec Studio 10. Pozwala ono na bieżące monitorowanie zbieranych punktów pomiarowych, doprecyzowanie zakresu pracy skanerów, wybranie rodzaju skanowania. Po zakończonym skanowaniu narzędzie pozwala na obróbkę cyfrowego modelu bez konieczności korzystania z dodatkowego oprogramowania (rys. 6). Wśród wielu narzędzi warto wymienić: narzędzia przekształcania i wygładzania, wyrównywanie i łączenie wielu modeli, wygładzanie i zamykanie krawędzi, czy dokonywanie pomiarów. Oprogramowanie umożliwia także zastosowanie odpowiednich algorytmów w celu wstępnej oraz zaawansowanej obróbki uzyskanych modeli. Do najczęściej stosowanych należy zaliczyć algorytmy:

dokładnej rejestracji, szybkiego łączenia, wygładzania, scalania oraz usuwania dziur.



Rys. 6. Okno robocze oprogramowania Artec Studio 10 Professional [opracowanie własne]

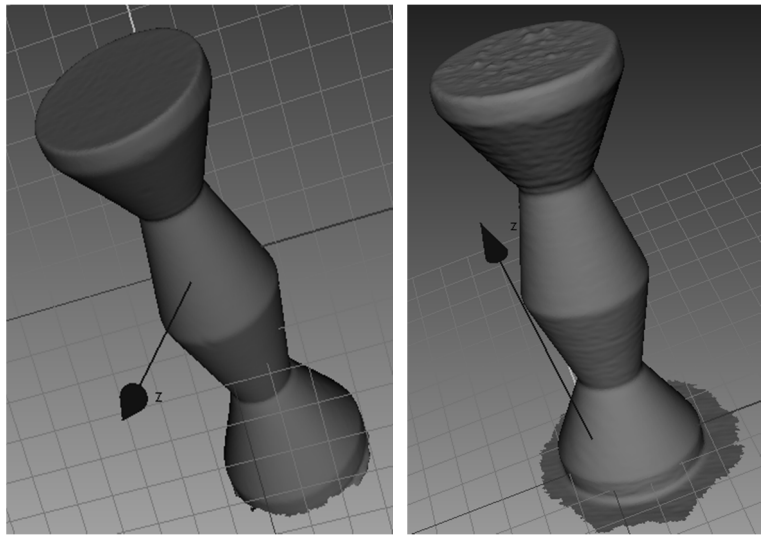
Pierwszym etapem badań była analiza włącznie cyfrowej geometrii zeskanowanych obiektów (rys. 7).

Rys. 7. Jakość modeli tłoka bez nałożonej tekstury: z lewej – rezultat skanowania skanerem Spider, z prawej – rezultat skanowania skanerem Eva [opracowanie własne]

Oba modele zostały po skanowaniu poddane działaniu algorytmu upraszczania siatki w celu wyznaczenia znaczących różnic w stosunku do modeli rzeczywistych. Oba modele nie posiadały otworów przelotowych, natomiast ich położenie po obu stronach walca znajdowały się w osi otworu. Geometria rowków pod pierścienie

znacząco odbiegała od rzeczywistych wymiarów. Kształt główny jak i powierzchnia czołowa na obu modelach nie posiadały znaczących odchyłek. W przypadku modelu uzyskanego za pomocą skanera Eva zaobserwowano delikatne braki w rejestracji dolnej części tłoka.

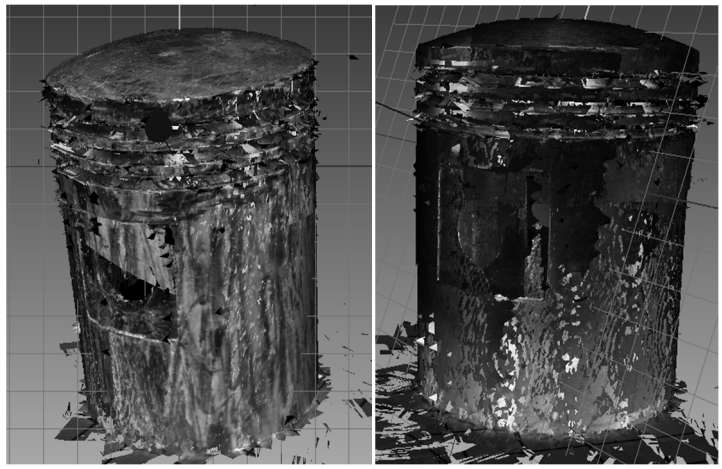
Analogicznie zeskanowano i przeanalizowano rezultaty procesu dla obiektu totem (rys. 8).



Rys. 8. Jakość modeli totemu bez nałożonej tekstury: z lewej – Spider, z prawej – Eva [opracowanie własne]

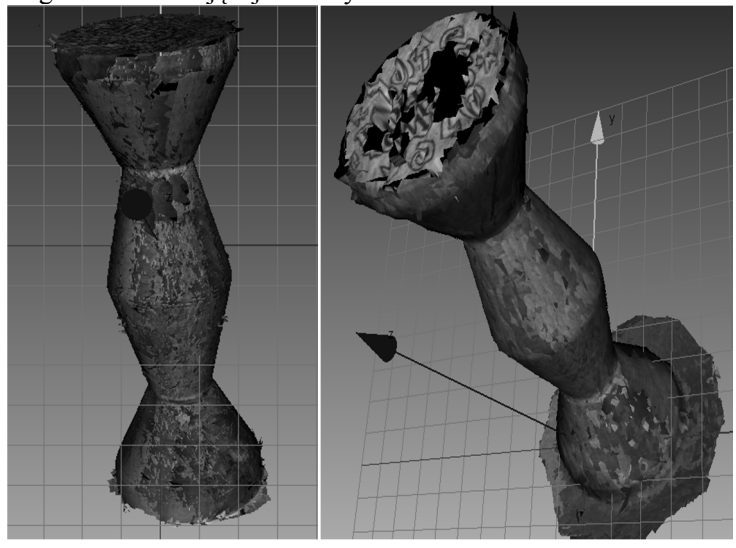
Model uzyskany po zeskanowaniu przy użyciu skanera Spider posiadał dokładnie odwzorowaną powierzchnię boczną oraz czołową. Po skanowaniu skanerem Eva można zauważyć znaczącą chropowatość powierzchni zarówno na płaszczyźnie czołowej jak i na powierzchni bocznej totemu. Wymiary oraz kształt główny modeli nie posiadał znaczących odkształceń.

Kolejnym etapem było sprawdzenie jakości modeli wraz z teksturą (rys. 9). Analizie poddano rezultaty skanowania bez wykorzystania jakichkolwiek algorytmów obróbki modeli.



Rys. 9. Jakość modeli tłoka wraz z teksturą: z lewej – Spider, z prawej – Eva [opracowanie własne]

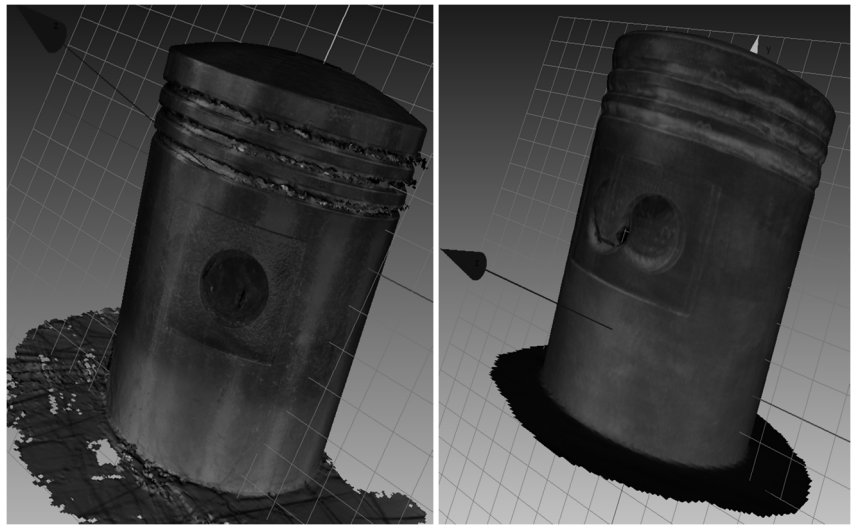
Oba modele posiadały zniekształcenia w okolicy otworów przelotowych. Rowki pod pierścienie posiadały liczne nieciągłości w geometrii. Tekstura po zeskanowaniu skanerem Eva była znacznie ciemniejsza od tej zeskanowanej skanerem Spider. Kształt główny w obu modelach nie posiadał znaczących różnic w stosunku do elementu rzeczywistego. Widoczne były zarejestrowane „szumy" w postaci fragmentów odstającej tekstury.



Rys. 10. Jakość modeli totemu wraz z teksturą; z lewej – Spider, z prawej – Eva [opracowanie własne]

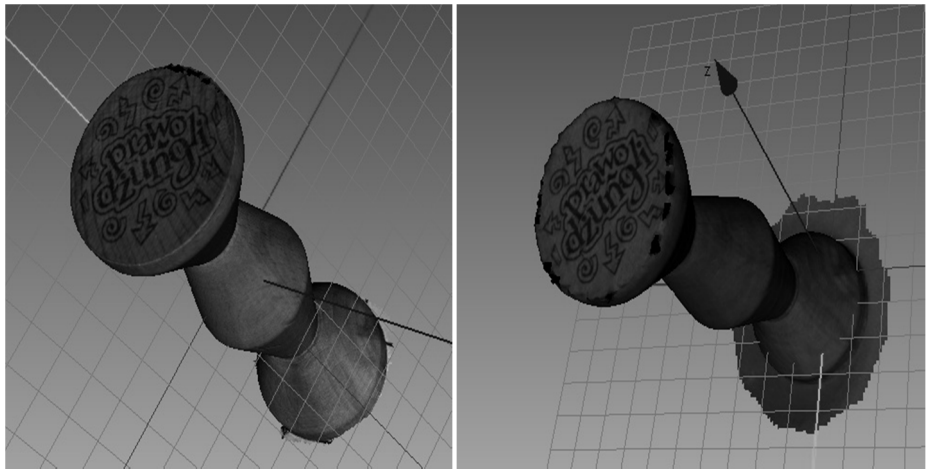
W przypadku totemu oba modele posiadały liczne braki przy krawędzi łączącej powierzchnię czołową totemu z powierzchnią boczną (rys. 10). Po zeskanowaniu elementu skanerem Eva powierzchnia czołowa nie została dobrze odwzorowana. Kształt główny modeli cyfrowych nie odbiegał od modeli rzeczywistych.

Ostatnim etapem badania było uproszczenie siatki zeskanowanego obiektu, a następnie nałożenie na niego zeskanowanej tekstury. W tym celu zastosowano odpowiednie algorytmy obróbki modeli oraz tekstur.



Rys. 11. Jakość uproszczonych modeli z nałożoną teksturą: z lewej – Spider, z prawej – Eva [opracowanie własne]

W przypadku tłoka, po uproszczeniu siatki obu modeli oraz nałożeniu na gotowy model tekstury, element zeskanowany za pomocą skanera Spider posiadał wyraźne nieciągłości w geometrii w miejscu rowków pod pierścienie. Tekstura była wyraźna, dobrze nałożona na model cyfrowy. W przypadku skanera Eva, model posiadał ciągłość na całej powierzchni. Tekstura była gorszej jakości, obrócona w stosunku do modelu rzeczywistego, co było szczególnie widoczne w miejscu otworu przelotowego (rys. 11).



Rys. 12. Jakość uproszczonego modelu z nałożoną teksturą: z lewej – Spider, z prawej – Eva [opracowanie własne]

Model totemu otrzymany po skanowaniu skanerem Spider posiadał jedynie niewielką nieciągłość geometrii w miejscu styku powierzchni czołowej z powierzchnią boczną. Tekstura została doskonale nałożona na model oraz była w bardzo dobrej jakości, co można z łatwością stwierdzić po jakości napisu znajdującego się na powierzchni czołowej. Po zeskanowaniu obiektu skanerem Eva model cyfrowy posiadał liczne braki na krawędzi styku czoła totemu z powierzchnia boczną. Dodatkowo powierzchnia czołowa posiadała niewielkie nierówności. Tekstura była ciemniejsza oraz gorszej jakości (rys. 12).

7. Analiza otrzymanych rezultatów

Dokonując analizy otrzymanych rezultatów skanowania obiektów rzeczywistych skanerami firmy Artec, można stwierdzić, że modele cyfrowe odpowiadają w znacznym stopniu obiektom rzeczywistym. Natomiast posiadają one pewne istotne błędy, które mogą dyskwalifikować tę metodę odwzorowywania przy tworzeniu bardziej skomplikowanych, bądź wrażliwych na niedoskonałości przedmiotów.

Geometria tłoka w miejscu rowków pod pierścienie nie została dobrze odwzorowana przez skanery. Przyczyną takiego efektu mógł być zakres pracy skanera oraz kąt nachylenia skanera w stosunku do skanowanego przedmiotu. Skaner Spider posiada wąski zakres pracy w niewielkiej odległości od siebie. Powinien więc znacznie lepiej poradzić sobie z tym zadaniem w stosunku do skanera Eva, którego zakres pracy jest szeroki oraz znacznie oddalony od skanera.

Mimo to skaner Spider nie był w stanie prawidłowo zeskanować geometrii w dość kluczowym w przypadku tłoka miejscu.

Powierzchnia czołowa oraz boczna totemu po zeskanowaniu skanerem Eva była chropowata. Mogło to wynikać z dużej odległości skanera od skanowanego przedmiotu oraz faktu że zarówno totem jak i tłok posiadają niewielkie gabaryty, do których skanowania skaner Spider jest zalecany przez producenta.

Oba modele po zeskanowaniu skanerami firmy Artec posiadały nieciągłości w geometrii. Skanery podczas pracy znajdowały się na statywie, czyli pozostawały w stałej pozycji w stosunku do obracającego się na stoliku obiektu. W czasie skanowania ich odległość oraz kąt nachylenia nie ulegały zmianie, co mogło doprowadzić do niezdolności skanerów do zebrania wszystkich punktów z przedmiotu. W wyniku tego modele cyfrowe posiadały braki, szczególnie przy krawędziach geometrii.

Tekstura zeskanowana przez skaner Eva była gorszej jakości od tej zeskanowanej przez skaner Spider oraz była ciemniejsza. W tym przypadku również znaczenie mogła mieć odległość skanera od obiektu rzeczywistego, a także rodzaj światła emitowanego przez skanery. Skaner Spider emitował niebieskie światło diodowe, natomiast skaner Eva błyskowe światło białe.

Należy stwierdzić, iż uzyskane rezultaty skanowania są w dużej mierze zależne od wprawy operatora, złożoności elementów, warunków oświetleniowych oraz wydajności wykorzystanego sprzętu komputerowego. Ponadto w przeprowadzonym badaniu elementy skanowano w całości, co było dodatkową trudnością. Z pewnością zastosowanie dodatkowego oprogramowania graficznego byłoby pomocne w procesie uzyskania modeli cyfrowych o lepszych parametrach.

8. Podsumowanie

Skanowanie 3D pozwala przyspieszyć proces projektowania, udoskonalania produktów oraz ułatwia pomiary. W wyniku tego procesu otrzymujemy chmurę punktów, którą w łatwy sposób możemy przekształcić w siatkę trójkątów. Na tej podstawie, dzięki procesowi inżynierii odwrotnej możemy uzyskać edytowalny model CAD.

Obiekty, które są skanowane nie powinny samoistnie odbijać, bądź rozpraszać padających na nie promieni światła. Powoduje to negatywny wpływ na proces skanowania, co może skutkować zniekształconą geometrią modelu cyfrowego, jego brakami bądź w najgorszym wypadku, zgubienie ścieżki skanowania przez skaner. Efektem tego ostatniego będzie konieczność powtórzenia całego procesu od nowa.

Skaner Artec Spider doskonale spisał się przy skanowaniu zarówno tłoka jak i totemu. Oba zeskanowane tym skanerem cyfrowe modele posiadały najbardziej

zbliżoną do rzeczywistej geometrię, nie posiadały odchyłek kształtu, a tekstura została nałożona prawidłowo oraz była bardzo dobrej jakości.

Skaner Artec Eva sprawdził się nieco gorzej, ale nie był on wykorzystany do skanowania zalecanych przez producenta wielkogabarytowych elementów. Mimo to, skaner był w stanie zeskanować obiekt i stworzyć możliwie zbliżony model cyfrowy posiadający odchyłki oraz braki.

Ręczne korzystanie ze skanerów wymaga dużej wprawy operatora. Ruchy skanerem muszą być bardzo płynne oraz spokojne. Zbyt szybki, bądź nerwowy ruch skanerem powoduje zgubienie przez niego ścieżki skanowania. W celu wyeliminowania czynnika ludzkiego z procesu skaner można umieścić na statywie bądź ramieniu robota, który w kontrolowany sposób dokona procesu skanowania.

Dzięki wykorzystaniu technologii skanowania 3D możliwe jest rekonstruowanie różnorodnych elementów dla których nie posiadamy dokumentacji technicznej, dopasowywanie modelu do istniejącego kształtu czy przeprowadzenie procesu kontroli jakości.

Literatura

1. Bouzakis K.-D., Pantermalis D., Mirisidis I., et al.: *3D-Laser Scanning of the Parthenon West Frieze Blocks and Their Digital Assembly Based on Extracted Characteristic Geometrical Details.* Journal of Archaeological Science: Reports, Vol. 6, 2016, pp. 94-108.
2. Develop3D Magazine: Review: Artec Eva & SpaceSpider – http://www.develop3d.com/reviews/review-artec-eva-spacespider-3D-scanning [data dostępu: 15-05-2016]
3. Dokumentacja techniczna skanera ARTEC 3D Eva – https://www.artec3d.com/hardware/artec-eva [data dostępu: 20-05-2016]
4. Dokumentacja techniczna skanera ARTEC 3D Spider – https://www.artec3d.com/hardware/artec-spider [data dostępu: 20-05-2016]
5. http://www.3ders.org/articles/20130908-rubicon-3d-scanner-on-indiegogo.html [data dostępu: 18-05-2016]
6. Kościuk J.: *Skanowanie 3D puka do drzwi*. Geodeta : magazyn geoinformacyjny, 2007, nr 1, s. 14-19.
7. Montusiewicz J., Czyż Z., Kayumov R.: *Selected Methods of Making Three-Dimensional Virtual Models of Museum Ceramic Objects*. Applied Computer Science, Vol. 11, No. 1, 2015, pp. 51-65.
8. Portal „Świat Obrazu": *Czym są skanery 3D i jak działają?* – http://www.swiatobrazu.pl/mobile/czym-sa-skanery-3d-i-jak-dzialaja-32416.html [data dostępu: 22-05-2016]

9. Psikuta A., Frackiewicz-Kaczmarek J., Mert E., Bueno M.-A., Rossi R. M., *Validation of a Novel 3D Scanning Method for Determination of the Air Gap in Clothing*, Measurement, Vol. 67, 2015, pp. 61-70.
10. Strona internetowa firmy 3D MASTER – http://www.manipulatory3d.pl/polecane-3d/skanery-3d [data dostępu: 18-05-2016]
11. VISION & SENSORS: *Artec Scanners used by Hyundai Europe to Create, Modify Automobile Seats.* December, 2013.
12. Wojtuszkiewicz K.: *Urządzenia techniki komputerowej 2 – Urządzenia peryferyjne i interfejsy.* Wydawnictwo Naukowe PWN, Warszawa, 2012.
13. Zgórniak P., Stachurski W.: *Wykorzystanie laserowego skanera 3D oraz współrzędnościowej maszyny pomiarowej do budowy i oceny modelu koła zębatego*. Mechanik, 2014, R. 87, nr 8-9 CD1, s. 438-450.

Abstrakt

Obrazowanie 3D obejmuje wszechstronne oraz nowoczesne rozwiązania przestrzennego odwzorowywania obiektów rzeczywistych znajdujące zastosowanie w wielu obszarach – m.in. w przemyśle maszynowym do inżynierii odwrotnej, w archiwizowaniu zbiór dzieł sztuki, czy w medycynie do planowania zabiegów chirurgicznych. Wraz z rosnącą popularnością technologii 3D coraz częściej spotyka się producentów oferujących szeroki asortyment skanerów 3D. Duża konkurencyjność sprawia, iż firmy produkujące sprzęt tego typu proponują coraz nowocześniejsze rozwiązania.

Celem niniejszej pracy jest ocena rezultatów procesu skanowania 3D, a także analiza parametrów wykorzystanych skanerów. W artykule przedstawiono oraz przeanalizowano rezultaty tworzenia modeli cyfrowych za pomocą ręcznych skanerów 3D. Zakres pracy obejmuje omówienie idei skanowania 3D, przedstawienie charakterystyk skanerów firmy Artec oraz szczegółowe zaprezentowanie metodyki procesu skanowania, obróbki modeli cyfrowych oraz analizę otrzymanych rezultatów.

Słowa kluczowe: modelowanie 3D, skanowanie 3D, inżynieria odwrotna, porównywanie skanerów

Abstract

3D visualization includes modern and comprehensive real objects representation solutions. It is used in engineering industry (in Reverse Engineering), archiving of artworks collection or in a medicine (in surgery planning). Nowadays, there are a lot

of producers which offer modern 3D scanners. High competitiveness stimulates grow of proposed solutions.

The paper presents the evaluation of the real object scanning results by means of two 3D Artec scanners. First of all, the idea of creating digital models using handheld 3D scanners was described. Moreover, the specification of used scanners, methodology of visualization process and 3D models processing were outlined. In the final part of the paper the authors discussed the research results.

Keywords: 3D modeling, 3D scanning, Reverse Engineering, scanners comparison

Marek Grobelny
Sebastian Jarosiński
Marcin Piłat
Daniel Pieniak
Łukasz Sobaszek
Instytut Technologicznych Systemów Informacyjnych
Wydział Mechaniczny, Politechnika Lubelska
20-618 Lublin, ul. Nadbystrzycka 36
marek.grobelny92@gmail.com
sebastian.jarosinski2912@gmail.com
marcin.pilat@gmail.com
daniel.pieniak@gmail.com
l.sobaszek@pollub.pl

Projekt aplikacji komputerowej umożliwiającej sterowanie robotem przemysłowym Kawasaki RS003N

1. Wstęp

We współczesnym przemyśle roboty znajdują coraz szersze zastosowanie [1, 5]. Popularne staje się tworzenie zrobotyzowanych gniazd obróbczych czy montażowych, wykorzystanie robotów do spawania, bądź malowania. Kluczowym celem takich rozwiązań jest zastąpienie ludzi w pracy na stanowiskach uciążliwych, bądź charakteryzujących się niebezpiecznymi warunkami [7]. Roboty są w stanie wykonywać pracę, która jest ryzykowna dla ludzi, albo wymaga dużej siły fizycznej lub precyzji.

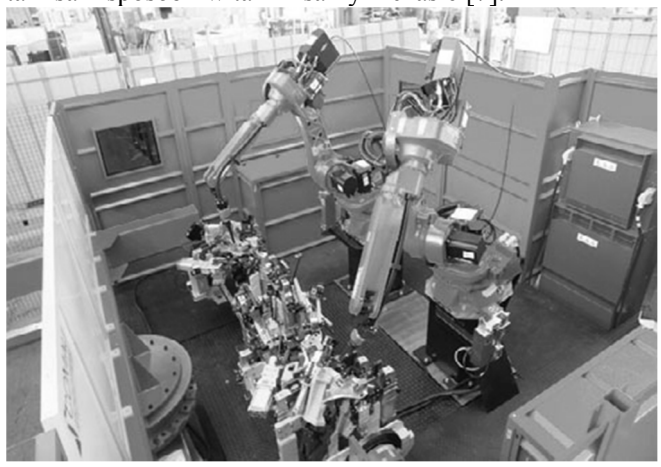
Niemiej jednak, nowym wyzwaniem w dziedzinie robotyki staje się współpraca robota z człowiekiem – jest to zauważalny i popularny trend w tej dziedzinie. Coraz częściej myśli się o tym, aby nie tylko tworzyć autonomiczne stanowiska zrobotyzowane, ale także zapewnić pracę robota „ramię w ramie" z człowiekiem [14, 17]. Zmienia to między innymi podejście do bezpieczeństwa – usuwane są wygrodzenia i bariery bezpieczeństwa, a ich działanie zastępuję się coraz nowszymi rozwiązaniami, włącznie z inteligentnym oprogramowaniem sterującym robotem [14, 18].

2. Zastosowanie robotów w przemyśle

Współczesny rynek produkcyjny wymusza na przedsiębiorcach ciągłe doskonalenie procesów produkcyjnych [4]. Głównym kryterium celu staje się czas realizacji zleceń. Zakład produkcyjny chcąc zaspokoić potrzeby konsumentów musi sprawnie realizować procesy produkcyjne [3, 4]. Dlatego też coraz częściej w przedsiębiorstwach zastosowanie znajdują roboty przemysłowe. Jako główne korzyści wynikające z ich wdrożenia i wykorzystania podaje się [2, 5]:

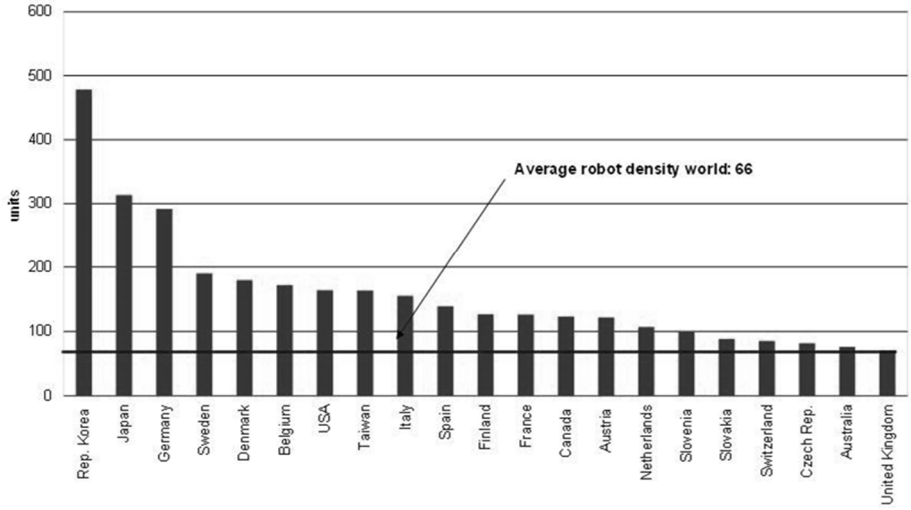
- szybkość – robot potrafi pracować szybciej od człowieka, a przy tym nie potrzebuje czasu na przerwy,
- precyzję i powtarzalność – współczesne roboty osiągają dokładność sięgającą nawet do tysięcznych części milimetra, a powtarzalność pozycji może wynosić ± 0,02 mm.
- niezawodność – szacunkowy czas niezawodnej pracy robota to kilka lat,
- zwiększenie wydajności – jest to wynikiem szybkości pracy robota oraz wydłużeniem czasu pracy (praca bez przerw),
- pracę w trudnych warunkach – roboty do prac specjalnych mogą pracować bezpiecznie w warunkach szkodliwych dla człowieka (wysoka temperatura, duże ciężary, wysokie zapylenie, hałas, środki chemiczne).

Klasycznie roboty wykorzystywane są w zakładach produkcyjnych do wykonywania jednej, wyspecjalizowanej w wykonywaniu zazwyczaj jednej czynności (rysunek 1). Przykładami takich czynności są: paletyzacja, spawanie, lakierowanie, klejenie, montowanie, sortowanie. Zastąpienie człowieka robotem zwiększa wydajność na danym stanowisku, ponieważ robot może pracować nawet 24 godziny na dobę. Robot wykonujący swoją pracę, praktycznie za każdym razem wykona ją w taki sam sposób i w takim samym czasie [7].



Rys. 1. Zrobotyzowana stacja spawalnicza [1]

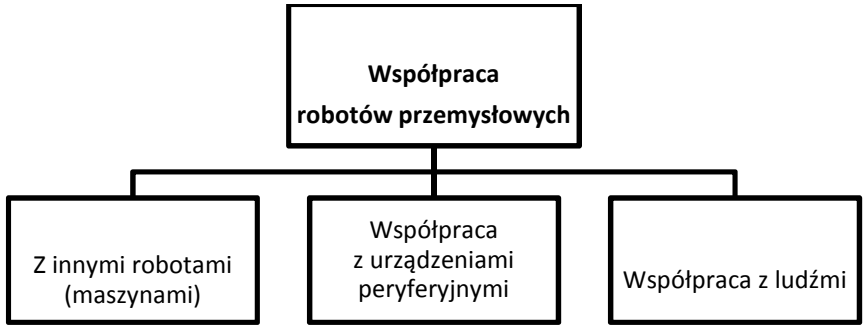
Międzynarodowa Federacja Robotyki opublikowała dane, z których wynika, że w roku 2015 nastąpił wzrost sprzedaży robotów przemysłowych o 8% (jest to prawie 240 000 maszyn) [11]. W roku 2014 zostały także przedstawione dane dotyczące najbardziej zrobotyzowanych przemysłów na świecie (wyrażonych w liczbie robotów przypadających na jednego pracownika przemysłu). Z opublikowanych danych wynika, iż najbardziej zrobotyzowany przemysł posiada Korea Południowa (478 maszyn na pracownika), następna jest Japonia (314 maszyn na pracownika), a kolejne są Niemcy (292 maszyn na pracownika). Średnia światowa wynosiła 66 robotów na pracownika (rys. 2) [12].



Rys. 2. Państwa z najwyższym stopniem zrobotyzowania przemysłu [12]

3. Współpraca robotów z ludźmi

Pojęcie „współpracy" w odniesieniu do robotów przemysłowych zazwyczaj rozumiane jest jako wymiana informacji pomiędzy poszczególnymi maszynami w zrobotyzowanym gnieździe wytwórczym. Niemiej jednak wyróżnić można inne kluczowe obszary współpracy robotów przemysłowych (rys. 3) [17].



Rys. 3. Rodzaje współpracy robotów przemysłowych [17]

Współpraca pomiędzy robotami występuje zazwyczaj w zrobotyzowanych gniazdach wytwórczych. Wówczas niezbędne jest zapewnienie odpowiedniej komunikacji, która realizowana jest przez typowe, przemysłowe protokoły komunikacyjne (PROFIBUS, Interbus S, MODBUS [8]. Czasami kilka robotów może być obsługiwanych za pomocą jednego kontrolera – wówczas nie jest koniecznie stosowanie zewnętrznych protokołów [15]. Niejednokrotnie wyposażenie stanowiska zrobotyzowanego obejmuje także inne maszyny posiadające odpowiednie porty komunikacyjne. Wówczas robot może także współpracować z takimi urządzeniami. Mowa tu o stołach pozycjonujących, obrabiarkach CNC, czy maszynach dedykowanych dla konkretnych procesów (np. owijarki do palet).

Komunikacja robota przemysłowego z urządzeniami peryferyjnymi może być także rozpatrywana w aspekcie współpracy. Różnego rodzaju czujniki czy specjalne oprzyrządowanie wymaga niejednokrotnie komunikacji z robotem. Ponadto elementy chwytne robota (mechaniczne, pneumatyczne czy elektryczne) sterowane są z kontrolera robota, a więc wymagają odpowiedniej komunikacji [17].

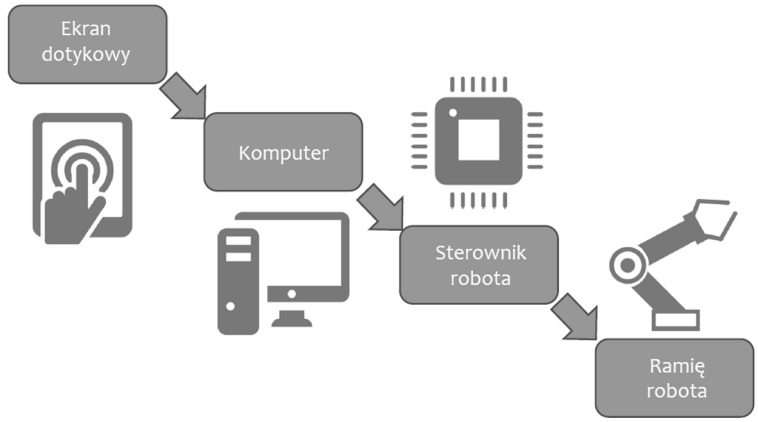
Trzeci rodzaj współpracy (kooperacja z człowiekiem) jest najnowszym kierunkiem rozwoju robotyki. Pomysłodawcy chcą odejść od barier bezpieczeństwa, zastępując je czujnikami wbudowanymi w roboty, systemami wizyjnymi oraz odpowiednim oprogramowaniem. Niektóre roboty można programować metodą uczenia przez demonstrację, co znacznie przyspiesza i ułatwia ich programowanie. Roboty tworzone z myślą o współpracy z człowiekiem często są mniejsze gabarytowo od typowych robotów przemysłowych. Ich kompaktowa wielkość pozwala na przenoszenie robota pomiędzy fragmentami linii produkcyjnej. Istnieje wiele gotowych rozwiązań proponowanych przez licznych producentów robotów. Przykładem mogą tu być roboty: ABB's YuMi, Baxter czy UR3 Universal Robots. Jednak roboty współpracujące z człowiekiem mają też wady, jak na przykład mniejsza maksymalna prędkość ruchów (podyktowana kwestiami bezpieczeństwa).

Dlatego też coraz częściej analizowane ją możliwości adaptacji typowo przemysłowych robotów do pracy z ludźmi [1, 17, 18].

Sugerując się najnowszymi kierunkami rozwoju robotyki, autorzy prezentowanej pracy postanowili wykonać aplikację komputerową, która zapewni komunikację oraz elementy współpracy typowego robota przemysłowego Kawasaki RS003N z użytkownikiem.

4. Sterowanie robotem przemysłowym za pomocą dedykowanej aplikacji

Celem prezentowanej pracy było stworzenie aplikacji komputerowej, która pozwalałaby na sterowanie typowym robotem przemysłowym. Ponadto, w celu weryfikacji działania proponowanego rozwiązania, zbudowano odpowiednie stanowisko wyposażone w: komputer z ekranem dotykowym, router (zapewniający komunikację pomiędzy komputerem i robotem poprzez złącze Ethernet) oraz robota przemysłowego Kawasaki RS003N. Schemat działania proponowanego rozwiązania został przedstawiony na rys. 4.



Rys. 4. Schemat działania proponowanego rozwiązania [opracowanie własne]

Użytkownik poprzez wybór polecenia na ekranie dotykowym komputera aktywował odpowiednią funkcję opracowanej aplikacji. Następnie z wykorzystaniem bibliotek DLL oraz łącza Ethernet do robota wysyłane było odpowiednie polecenie, które po przetworzeniu przez sterownik robota, realizowało odpowiednie ruchy ramienia (odpowiedni program robota). Całość projektu została przygotowana z myślą o Dniach Otwartych Politechniki Lubelskiej. Działanie robota polegało na wybraniu odpowiedniej płyty CD, umieszczeniu jej w odtwarzaczu oraz uruchomieniu muzyki. Aplikacja umożliwiała zatrzymanie odtwarzania oraz zmianę

muzyki. Dodatkowo każdy z operatorów mógł otrzymać od robota upominek – zdarzenie to również realizowane było za pomocą opracowanej aplikacji.

4.1. Opis zastosowanego robota

Prace przedstawione w niniejszej publikacji zostały przeprowadzone na stanowisku laboratoryjnym wyposażonym w robota przemysłowego Kawasaki RS003N wyposażonego w narzędzie pneumatyczne (przyssawkę). Jest to stanowisko wykorzystywane podczas typowych zajęć dydaktycznych (rys. 5). Ponieważ wykonana aplikacja miała zapewniać pełną interakcję z użytkownikiem – świetnie bariery bezpieczeństwa w które wyposażone jest stanowisko nie były podłączone do kontrolera robota.

Rys. 5. Stanowisko badawcze wyposażone w robota przemysłowego Kawasaki [opracowanie własne]

Robot Kawasaki RS003N jest przykładem manipulatora, a więc ma za zadanie odwzorowywać ruchy kończyny górnej człowieka. Manipulator przemysłowy jest to stacjonarnym układ ramion połączonych ze sobą przegubami, umożliwiającymi ich ruch [7]. Ruch wykorzystanego robota można programować za pomocą ręcznego programatora (Teach Pendant'a) lub przy pomocy dedykowanego oprogramowania K-ROSET. Teach Pendant ma charakter panelu operatorskiego, który umożliwia sterowanie oraz programowanie robota (za pomocą zapamiętywania pozycji lub poleceń języka AS). K-ROSET jest natomiast zaawansowanym oprogramowaniem, które pozwala symulować trajektorię ruchu robota, analizować czasy cykli, wyszukiwać potencjalne kolizje oraz analizować pozycje umieszczenia robota na stanowisku. K-ROSET podobnie jak inne programy tego typu umożliwia programowanie robota w trybie off-line. Odbywa się ono w języku AS. AS jest językiem programowania robotów stworzonym przez firmę Kawasaki. Jest on zaimplementowany w programie K-ROSET w postaci zbioru poleceń ruchu, obsługi

sygnałów, sterowania chwytakiem, itp. W aplikacji został zastosowany emulator kontrolera robota, co pozwala na praktycznie 100% zgodność wyników [16]. Za sterowanie ramieniem robota odpowiedzialny jest kontroler, który jednocześnie odpowiada za możliwość komunikacji z robotem poprzez odpowiednie protokoły komunikacyjne. Specyfikacja robota oraz kontrolera została przedstawiona w tabeli 1.

Tab. 1. Specyfikacja robota oraz kontrolera [10]

Specyfikacja robota	Specyfikacja kontrolera
Model: RS003N-A Typ: robot przegubowy Liczba stopni swobody: 6 Powtarzalność: ±0,05 mm Maksymalne obciążenie: 3 kg Maksymalna prędkość: 6000 mm/s Jednostka zasilająca: bezszczotkowy serwonapęd zasilany prądem zmiennym	Model: E70/E71 Konstrukcja: samo-wspierający Liczba sterowanych osi: 6 Typ sterowania: teach/repeat Sposób nauczania: język AS Napięcie zasilające: 200-240 V (50/60 Hz)

Proces programowania robota Kawasaki może odbywać się poprzez naukę „krok po kroku" lub napisanie programu za pomocą poleceń języka AS. Pierwsza z metod polega na wprowadzeniu do pamięci kontrolera danych dotyczących takich parametrów jak: współrzędne kartezjańskie położenia końcówki ramienia (lub wartości kątów wychylenia poszczególnych osi), prędkość ruchu robota, sygnału włączenia/wyłączenia narzędzia, precyzja ruchu ramienia, ewentualny czas oczekiwania na kolejny ruch. W rzeczywistości proces ten polega na zapisywaniu kolejnych położeń robota. Zapisane punkty łączą się w trajektorię ruchu robota. Programowanie robota za pomocą języka AS przypomina klasyczne programowanie wysokopoziomowe. Wykorzystując szereg poleceń (rozumianych przez kontroler) programista określa kolejne zachowania robota. Podobnie jak w większości języków programowania dane mają postać: stałych, zmiennych lokalnych lub globalnych. Standardowe są również typy danych: dane numeryczne, ciągi znaków alfanumerycznych w kodzie ASCII oraz wartości logiczne. Instrukcje języka AS obejmują polecenia sterujące wykonaniem programu, definiowania pozycji robota, ruchowe, sterujące prędkością i dokładnością osiągania pozycji docelowej oraz narzędziem [10]. Możliwe są następujące rodzaje interpolacji:

- liniowa (LMOVE),
- kołowa (C1MOVE),
- liniowa w przestrzeni konfiguracyjnej manipulatora (JMOVE).

Instrukcje sterujące prędkością i dokładnością umożliwiają ustawienie: prędkości (SPEED) czy dokładności osiągania pozycji (ACCURACY). Sterowanie narzędziem odbywa się za pomocą instrukcji uruchomienia (OPEN) i wyłączenia (CLOSE) narzędzia. Do sterowania wykonaniem programu wykorzystuje się również standardowe instrukcje wyboru (IF, THEN, ELSE), pętle (FOR, DO, WHILE) i instrukcję skoku (GOTO). Na rys. 6 został przedstawiony przykład prostego programu polegającego na rozpoczęciu ruchu z punktu *#start*, dojechaniu do punktu pobrania *#p001* i pobraniu elementu za pomocą chwytaka.

```
.PROGRAM point1()

JMOVE #start
JAPPRO #p001,50
JMOVE #p001
CLAMP 1
JAPPRO #pb_1,50

.END
```

Rys. 6. Fragment kodu napisanego w języku AS [opracowanie własne]

Autorzy niniejszej pracy wykorzystali metodę programowania robota za pomocą ręcznego programatora. Dla każdego ze zdarzeń (np. „weź płytę nr 1 i umieść ją w odtwarzaczu", „podaj element dla operatora") za pomocą Teach Pendant'a został opracowany program robota zapisany w pamięci kontrolera.

Ważnym elementem programowania ruchu robota jest weryfikacja wykonanego programu. Ma ona na celu sprawdzenie poprawności pracy robota, wprowadzeniu ewentualnej korekty, uniknięciu kolizji lub zoptymalizowaniu ruchy robota. Proces ten realizowany jest za pomocą ręcznego programatora, który umożliwia zmianę sterowania pomiędzy trybami „TEACH/REPEAT" (z ang. „naucz/powtórz"). Testowanie programu może odbywać się krokowo lub płynnie. Ponadto możliwe jest wykonywanie opracowanego programu w pętli. Podczas realizowanych prac funkcja ta była bardzo pomocna, gdyż precyzyjna praca robota na przedstawionym stanowisku wymagała niejednokrotnie wprowadzania wielu korekt w trakcie programowania ramienia.

4.2. Środowisko programistyczne

W celu stworzenia aplikacji sterującej robotem wykorzystano środowisko Microsoft Visual Studio 2010 w wersji Express. Visual Studio to wszechstronne zintegrowane środowisko programistyczne firmy Microsoft, przeznaczone do tworzenia zróżnicowanych aplikacji, pozwalające na kompleksową realizację

projektu (w ramach jednego zagadnienia istnieje możliwość realizowania kilku odrębnych zadań). Środowisko to pozwala na wykonanie zintegrowanych projektów, będących kombinacją aplikacji okienkowych, internetowych, wykonywanych z linii poleceń czy web serwisów [6]. Cechy charakterystyczne Visual Studio to [13]:

- Jeden IDE (zintegrowany interfejs użytkownika) dla kilku języków i różnych typów projektów,
- pozwala na połączenie z przeglądarką Internetową,
- dostarcza narzędzi do debugowania,
- posiada dostosowany do oczekiwań użytkownika interfejs.

Aplikacja została wykonana w języku Visual Basic, który charakteryzuje się względnie prostą składnia, a ponadto posiada wsparcie ze strony firmy Kawasaki.

W aplikacji zastosowano bibliotekę DLL dedykowaną dla robotów Kawasaki. Jest to biblioteka AsKCT składająca się z odpowiednich funkcji. W opracowanej aplikacji zastosowanie znalazły [9]:

- funkcje AsKCTInit() i AsKCTDestroy() – inicjujące oraz kończące komunikację pomiędzy robotem a aplikacją,
- funkcje AsKCTConnect() i AsKCTDisconnect() – pozwalające nawiązać połączenie za pomocą łącza Ethernet,
- funkcja AsKCTCommand() – umożliwiająca przesyłanie do robota odpowiedniego polecenia (uruchomienie odpowiedniego programu/polecenia).

4.3. Budowa programu sterującego robotem

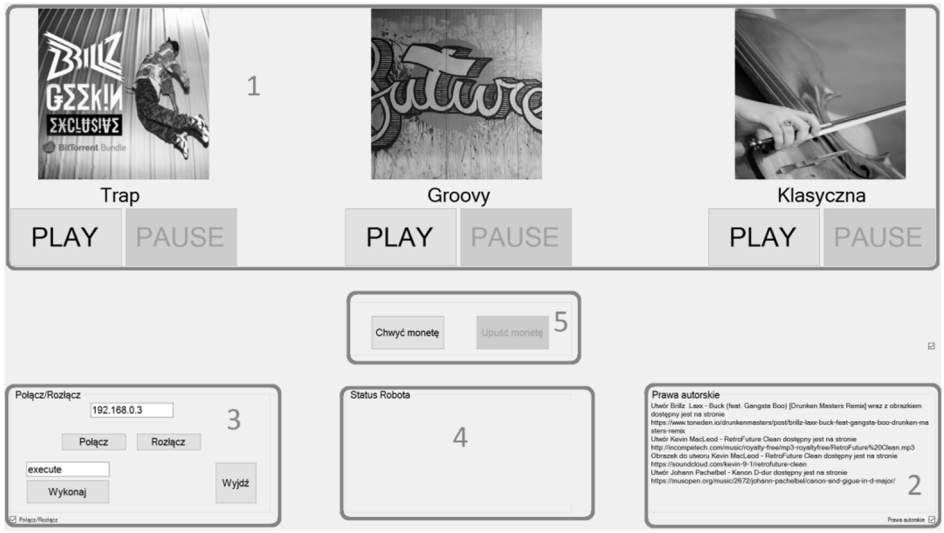
Wygląd aplikacji został opracowany tak, aby ułatwić jego wyświetlanie na dużym ekranie dotykowym oraz zapewnić łatwy i prosty dostęp do poszczególnych funkcji programu (rys. 8). W górnej części ustawiono obrazki przedstawiające dany styl muzyczny oraz przyciski do wydania poleceń – uruchamiania, bądź zatrzymywania muzyki (1). Pod każdym z przycisków umieszczono odpowiednie polecenia języka Visual Basic oraz biblioteki AsKCT (rys. 7)

```
Dim ret As Integer
ret = AsKCTCommand(0, "execute cd1")

If ret <> 0 Then
     MessageBox.Show("Connect Error : " & ret)
Else
     MessageBox.Show("Odtwarzam CD1", "STATUS")
End If
```

Rys. 7. Fragment kodu – uruchomienie programu odtwarzania płyty nr 1 [opracowanie własne]

W prawym dolnym rogu umieszczone zostały informacje dotyczące wykorzystanych materiałów źródłowych (obrazów oraz muzyki) (2). W lewym dolnym rogu zamieszczono przyciski przeznaczone do użytku przez osobę nadzorującą pracą robota (3) – jest to m.in. obszar do manualnego przekazywania poleceń oraz okno wprowadzania adresu IP robota i ustanawiania połączenia. W dolnej części menu (4) znajduje się kontrolka GroupBox w której wyświetlany jest obecny status robota. W centralnej części aplikacji znajdują się dwa przyciski odpowiadające za program do podawania upominków w czasie prezentacji (5). Obszary 2, 3 oraz 5 są domyślnie ukryte i mogą zostać pokazane przez naciśnięcie odpowiedniej kontrolki CheckBox.



Rys. 8. Wygląd programu do sterowania robotem przemysłowym Kawasaki [opracowanie własne]

Ważnym aspektem pracy programu była dezaktywacja jego przycisków w trakcie wykonywania poleceń przez ramie robota. Zbyt duża liczba przesyłanych danych powoduje zawieszenie połączenia z robotem, a także błędy pracy kontrolera. Zastosowano zatem czasową blokadę wybranych przycisków poprzez zastosowanie kontrolek typu Timer, które czasowo dezaktywowały większość przycisków (rys. 9).

```
Private Sub TPlay1_Tick(ByVal sender As System.Object, e As
System.EventArgs) Handles TPlay1.Tick

   i += 1
   If i = 55 Then
        STOP1.Enabled = True
        TPlay1.Stop()
        i = 0
   End If
End Sub

Private Sub TStop1_Tick(ByVal sender As System.Object, e As
System.EventArgs) Handles TStop1.Tick

   i += 1
   If i = 50 Then
        PLAY1.Enabled = True
        PLAY2.Enabled = True
        PLAY3.Enabled = True
        TStop1.Stop()
        i = 0
   End If
End Sub
```

Rys. 9. Fragment kodu – zastosowanie kontrolek Timer [opracowanie własne]

5. Podsumowanie

Roboty przemysłowe znacząco zwiększają produktywność oraz elastyczność produkcji zakładów przemysłowych, więc prognozowany wzrost ich sprzedaży w kolejnych latach nie jest bezzasadny. Czynnik ludki jest i dalej będzie potrzebny w każdym przedsiębiorstwie, niezależnie od jego zrobotyzowania, ponieważ każdy robot potrzebuje operatora. Znajomość języków programowania oraz umiejętne sterowanie robotem pozwala na zaprojektowanie oraz optymalizację aplikacji dedykowanych dla konkretnych procesów (takich jak spawanie, montaż, czy sortowanie). Odpowiednio wykonana aplikacja umożliwia intuicyjną, sprawną i bezpieczną komunikację pomiędzy operatorem, a robotem.

Autorzy w niniejszej pracy zaprezentowali wykorzystanie typowego środowiska programistycznego do stworzenia aplikacji sterującej dla robota przemysłowego Kawasaki RS003N. Przedstawione oprogramowanie realizowało tylko kilka z wielu dostępnych zastosowań robota, jednak niewątpliwie istnieją możliwości rozbudowy proponowanego rozwiązania o chociażby możliwość sterowania robotem za pomocą urządzeń mobilnych z wykorzystaniem połączenia WiFi czy budowę układów sterowania opartych o platformę Arduino, bądź Raspberry PI.

Literatura

1. AutomatykaOnline.pl: Robotyzacja i automatyzacja procesów wytwórczych – http://automatykaonline.pl/Artykuly/Robotyka/Robotyzacja-i-automatyzacja-procesow-wytworczych
2. EVOTEC: Dlaczego robot? – http://evotec.com.pl/dlaczego-robot.html [data dostępu: 17-11-2015]
3. Gola A., Sobaszek Ł., Świć A.: Selected Problems of Modern Manufacturing Systems Design and Operation. Robotics and Manufacturing Systems, Lublin, 2014, p. 56–68.
4. Gola A.: Economic Aspects of Manufacturing Systems Design. Actual Problems of Economics, 156, 6 (2014), pp. 205–212.
5. Hajduk M., Koukolová L.: Trends in Industrial and Service Robot Application. Applied Mechanics and Materials, Vol. 791, 2015, pp. 161–165.
6. Halvorson M.: Microsoft Visual Basic 2010 Step by Step, Microsoft Press, Washington, 2010.
7. Honczarenko J.: Roboty Przemysłowe. Budowa i zastosowanie, Wydawnictwa Naukowo-Techniczne, Warszawa, 2004.
8. Kaczmarek W.: Elementy Robotyki Przemysłowej. WAT, Warszawa, 2008.
9. Kawasaki Robot Materials, KCwinTCP-dll – Instruction Manual.
10. Kawasaki Robot Materials, RS003NFE70 – Robot Specification.
11. Materiały Międzynarodowej Federacji Robotyki: http://www.ifr.org/news/ifr-press-release/president-s-report-807 [data dostępu: 15-05-2016]
12. Materiały Międzynarodowej Federacji Robotyki: http://www.ifr.org/news/ifr-press-release/survey-13-million-industrial-robots-to-enter-service-by-2018-799 [data dostępu: 15-05-2016]
13. Piłat M.: Projekt multimedialnego serwisu internetowego programowania w C#. Politechnika Lubelska, 2016.
14. Portal gospodarczy WNP.PL: Współpraca człowiek-robot w procesie produkcyjnym Audi – http://motoryzacja.wnp.pl/wspolpraca-czlowiek-robot-w-procesie-produkcyjnym-audi,244424_1_0_0.html
15. Semjon J., Baláž V., Vagaš M.: Project multirobotic systems with KUKA robots in cooperation with VW Slovakia, in: L. Koukolová, A. Świć (Eds.), Robotics and manufacturing systems, Lublin, 2014, pp. 33-38.
16. Sobaszek Ł., Gola A., Varga J.: Virtual Designing of Robotic Workstations. Applied Mechanics and Materials, Vol. 844, pp. 31-37.

17. Sobaszek Ł., Gola A.: Perspective and methods of human – industrial robots cooperation. APPLIED MECHANICS AND MATERIALS, 2015, vol. 791, pp. 178-183.
18. The Engineer: Robot revolution: Humans and droids, working together – http://www.theengineer.co.uk/manufacturing/automation/robot-revolution-humans-and-droids-working-together/1019569.article

Abstrakt

Roboty znajdują szerokie zastosowanie w dzisiejszym przemyśle. Najczęściej wykorzystuje się je, aby zastąpić ludzi w pracy na stanowiskach niebezpiecznych, bądź uciążliwych. Niemiej jednak, coraz bardziej popularnym trendem w robotyce staje się współpraca robota z człowiekiem. Zagadnienie to stanowi nowe wyzwanie w tej dziedzinie – usuwane są bowiem bariery bezpieczeństwa, a ich działanie zastępowane jest inteligentnym oprogramowaniem robota.

Celem niniejszego artykułu jest przedstawienie projektu zespołowego dotyczącego stworzenia aplikacji komputerowej, która umożliwia sterowanie robotem przemysłowym. W pracy omówiono także ideę współpracy robotów z ludźmi, przedstawiono stanowisko badawcze wyposażone w robota Kawasaki RS003N na którym realizowano prace, zaprezentowano aplikację opracowaną w języku Visual Basic, jak i metodykę programowania robota.

Słowa kluczowe: robot przemysłowy, programowanie robotów, Visual Basic, współpraca człowiek-robot

Abstract

Nowadays, robots are widely used in the industry. Mainly purpose of its implementation is to replace human at the danger and oppressive work stations. However, an increasingly popular trend in robotics is a human-industrial robots cooperation. This issue represents a new challenge in this area.

The paper presents the results of group project on creating software which helps to industrial robot control. First of all, the idea of a human-industrial robots cooperation was presented. Moreover, the laboratory station equipped with a Kawasaki RS003N robot was outlined. In the final part of the paper the authors discussed the prepared software and industrial robot programming methodology.

Keywords: industrial robot, robots programming, Visual Basic, human-robot cooperation

Mateusz Stasiełowicz
Wydział Elektroniki i Informatyki
Politechnika Koszalińska
mateusz.stawicz@gmail.com

Kompresja obrazów z wykorzystaniem kompresji fraktalnej i systemu funkcji iterowanych

1. Wprowadzenie

Przetwarzanie i przechowywanie informacji towarzyszyło człowiekowi od zawsze w każdej dziedzinie życia. Obecnie większość danych zapisana jest na nośnikach cyfrowych jako ciągi bitów a ilość przechowywanych informacji jest ogromna i rośnie w bardzo szybkim tempie. W każdym obszarze działalności ludzkiej gdzie istnieje potrzeba magazynowania i przesyłania informacji znajdują zastosowanie algorytmy kompresji danych. Dzięki efektywnemu wykorzystaniu, algorytmy te są w stanie zmniejszyć ilość przechowywanych danych kilkukrotnie co przekłada się na duże oszczędności związane ze składowaniem i przesyłaniem danych. Obecnie kompresja obrazów zdominowana jest przez algorytm JPEG, alternatywę do niego może stanowić kompresja fraktalna, która szersze zastosowanie może znaleźć właśnie w kompresji obrazów. Metoda ta może dawać zaskakująco dobre wyniki w kompresji obrazu zachowując jednocześnie dobrą jakość skompresowanego obrazu.

2. Zastosowania kompresji fraktalnej

Istnieją dwie główne kategorie podziału metod kompresji – stratne i bezstratne. Wynikiem działań metod bezstratnych jest obraz który po dekompresji będzie identyczny jak obraz oryginalny. Stratne metody kompresji różnią się pod tym względem. Obraz skompresowany nie jest identyczny z oryginałem, jest tylko w pewnym stopniu podobny. Największą przewagą algorytmów stratnych nad bezstratnymi jest poziom kompresji który jest kilkukrotnie bądź kilkunastokrotnie większy na korzyść algorytmów stratnych. Kompresja fraktalna zaliczana jako metoda kompresji stratnej może być zastosowana wszędzie tam gdzie dopuszczalna jest utrata część informacji podczas procesu kompresji. Z tego względu wykorzystanie metod fraktalnych jest możliwe dla tych informacji, które mogą zostać zniekształcone, a ludzkie zmysły ze względu na swoje niedoskonałości nie są w stanie wychwycić różnic pomiędzy zniekształconą informacją skompresowaną

a oryginalną. Algorytmy kompresji stratnej znajdują w większości zastosowanie w kompresji obrazu, filmu i dźwięku.

Kompresja fraktalna bazuje na pojęciu fraktala. Fraktale zostały wprowadzone do matematyki w latach 70. XX wieku przez Benoîta Mandelbrota i do dzisiaj są przedmiotem wielu badań na całym świecie. Kompresja fraktalna wykorzystuje jedną z głównych cech fraktali – samopodobieństwo – jest to własność zbioru polegająca na tym, że dowolny mały fragment tego zbioru jest podobny w danej skali do większego fragmentu samego siebie. Bazując na powyższym stwierdzeniu można dojść do wniosku, że kompresja fraktalna może być wykorzystywana wyłącznie tam gdzie w kompresowanym obiekcie występują elementy samopodobne. W większości obrazów i filmów występuje bardzo duża ilość powtórzeń afinicznych, dzięki czemu można zastosować względem nich metody kompresji fraktalnej.

3. Algorytm kompresji fraktalnej

Algorytm wykorzystany w przeprowadzonych badaniach bazuje na podstawowych założeniach kompresji fraktalnej. Do wyszukiwania elementów podobnych w obrazie służą przekształcenia afiniczne, składają się one z przekształceń liniowych i translacji.

$$w = \begin{bmatrix} a & b \\ c & d \end{bmatrix} * \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} e \\ f \end{bmatrix} \quad (1)$$

$$w(x, y) = (ax + by + e, cx + dy + f) \quad (2)$$

Przy czym:

- w jest przekształceniem afinicznym,
- a, b, c, d odpowiadają za przekształcenia liniowe takie jak skalowanie i obrót,
- e, f odpowiadają za translację odpowiednio punktów x i y

3.1. Założenia wstępne

Przed przeprowadzeniem badań zostały zdefiniowane ograniczenia i wartości początkowe niektórych parametrów. Zdecydowano się na zastosowanie obrazu w niskiej rozdzielczości i skali szarości ze względu na dużą złożoność obliczeniową algorytmu. Wykorzystany obraz wejściowy jest przedstawiony na rysunku poniżej (Rys. 1). Założenia wstępne:

- Obraz kompresowany jest w skali szarości,
- Obraz kompresowany jest w rozdzielczości 256x256 co daje w sumie 65536 pikseli,
- Obraz wymaga 65 536 bajty pamięci,

- Współczynnik skalowania jest równy 0,5 dla wszystkich przekształceń afinicznych,
- Zostały zdefiniowane cztery przekształcenia obrotu (0˚, 90˚, 180˚, 270˚),
- Obrazem kompresowanym jest obraz "lenna",
- Bloki i obszary są o kształcie kwadratu.

Rys. 1. Obraz "Lenna" z przykładowymi elementami samopodobnymi

3.2. Opis algorytmu

Pierwszym krokiem działania algorytmu jest podział obrazu wejściowego na nienachodzące na siebie bloki. Bloki te mogą być dowolnego kształtu i rozmiaru. W algorytmie zostały przyjęte bloki o rozmiarze 8x8 pikseli w ten sposób zostały zdefiniowane 1024 bloki. Następny krok to zdefiniowanie obszarów o większym rozmiarze niż bloki. Dla uproszczenie współczynnik skalowania został zdefiniowany jako 0,5 dzięki czemu wszystkie obszary mają rozmiar 16x16 pikseli. Obszary mogą zachodzić na siebie, nie jest wymagane również aby pokrywały one cały obraz. W przypadku definiowania obszarów z krokiem co 1 piksel zostanie zdefiniowanych 58 081 obszarów. Zmniejszenie ilości obszarów skutkowało by pogorszeniem jakości skompresowanego obrazu lecz znacznie przyspieszyłoby czas działania algorytmu.

W kolejnym kroku algorytm porównuje wszystkie bloki z każdym z obszarów w każdej z 4 sekwencji obrotu. Jest to najbardziej czasochłonny etap działania algorytmu ze względu na dużą ilość niezbędnych do wykonania porównań. W przypadku obrazu w skali szarości, przekształcenie afiniczne można wyrazić w postaci:

$$W = \begin{bmatrix} a & b & 0 \\ c & d & 0 \\ 0 & 0 & s \end{bmatrix} * \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} e \\ f \\ o \end{bmatrix} \tag{3}$$

Przy czym:

- w jest przekształceniem afinicznym,
- a, b, c, d odpowiadają za przekształcenia liniowe takie jak skalowanie i obrót,
- e, f odpowiadają za translację odpowiednio punktów x i y. $e, f = [-241{,}241], \quad e, f \in Z$
- z jest poziomem szarości, gdzie $z = [0{,}255], \ z \in Z$
- $s = [0{,}1], \ s \in R$
- $o = [-255{,}255], \ o \in Z$

Porównanie bloku R i obszaru D odbywa się poprzez wyznaczenie współczynników s i o dla każdego z wariantów obrotu obszaru D, a następnie obliczenia średniego błędu kwadratowego g_{rms}. Znalezienie najlepiej pasującego obszaru D do bloku R polega na minimalizacji g_{rms}.

$$g_{rms} = \sqrt{\sum_{k=1}^{N^2} (s * d_k + o - r_k)^2} \tag{4}$$

$$s = \frac{N^2 \sum_{k=1}^{N^2} d_k r_k - \sum_{k=1}^{N^2} d_k \sum_{k=1}^{N^2} r_k}{N^2 \sum_{k=1}^{N^2} d_k^2 - \left(\sum_{k=1}^{N^2} d_k\right)^2} \tag{5}$$

$$o = \frac{\sum_{k=1}^{N^2} r_k - s \sum_{k=1}^{N^2} d_k}{N^2} \tag{6}$$

Przy czym:

- d_k jest poziomem szarości k-tego punktu w bloku D
- r_k jest poziomem szarości k-tego punktu w obszarze R
- N jest długością boku bloku R

Działanie algorytmu kończy się wraz z wyznaczeniem całego wektora przekształceń afinicznych W. W przypadku tego wariantu algorytmu wektor W będzie zawierał 1024 przekształcenia afiniczne.

3.3. Wynik działania algorytmu

Cała skompresowana informacja zawiera się w wektorze przekształceń W. Na tym etapie możliwe jest obliczenie ilości bitów niezbędnych do zakodowania niniejszego wektora. Wszystkie niezbędne informacje do przeprowadzenia dekompresji zawierają się w przekształceniach afinicznych i współczynnikach s i o co zostało przedstawione w poniższej tabeli (Tab. 1).

Tab. 1. Zbiór danych przedstawiający ilość bitów potrzebnych do zakodowania jednego bloku

Operacja	**Ilość bitów potrzebnych do zapisu**
Translacja x	9 bitów
Translacja y	9 bitów
Skalowanie (stała wartość)	0 bitów
Rotacja (4 warianty obrotu)	2 bity
Współczynnik s	5 bitów
Współczynnik o	9 bitów
Suma	34 bity

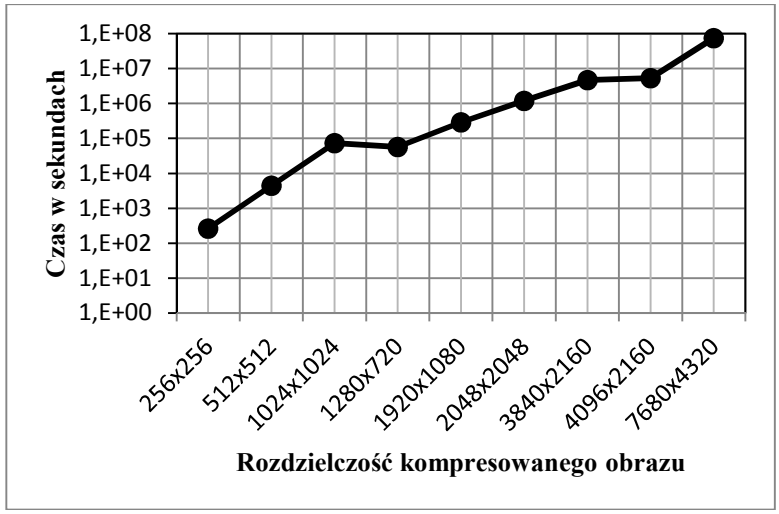
W przypadku tego wariantu algorytmu zostały zdefiniowane 1024 bloki R. Dany obraz wejściowy który zajmował początkowo 65535 bajty, po skompresowaniu będzie reprezentowany przez 34816 bitów czyli 4351 bajty. Dzięki tym informacjom w łatwy sposób można obliczyć stopień kompresji, który w tym przypadku wynosi 15. Na rysunkach (Rys. 2 i 3) zostały przedstawione obraz przed kompresją i po kompresji. Można zauważyć, że obraz oryginalny jest lepszej jakości niż skompresowany, jednakże jest to widoczne dopiero po dokładniejszej analizie obu obrazów. Działanie algorytmu było testowane na jednostce komputerowej z procesorem Intel i7 o taktowaniu 2,7GHz, gdzie czas działania tej wersji algorytmu wyniósł około 263 sekundy. Szczegółowe informacje na temat złożoności obliczeniowej oraz czasu działania w zależności od rozdzielczości obrazu wejściowego zostały przedstawione w poniższej tabeli (Tab. 2).

Rys. 2. Obraz wejściowy (65536 bajty)

Rys. 3. Obraz skompresowany (4351 bajty)

Tab. 2. Zbiór danych przedstawiający ilość porównań i szacunkowy czas kompresji dla wybranych rozdzielczości obrazu w najbardziej czasochłonnym etapie algorytmu z założonymi wcześniej parametrami

Rozdzielczość obrazu	**Ilość bloków**	**Ilość obszarów**	**Ilość porównań**	**Szacunkowy czas działania w sekundach**	**Szacunkowy czas działania**
256x256	1024	58081	2,37900E+08	263	4,4 minuty
512x512	4096	247009	4,04700E+09	4474	74,5 minuty
1024x1024	16384	1018081	6,67210E+10	73761	20,5 godziny
1280x720	14400	891825	5,13691E+10	56789	15,8 godziny
1920x1080	32400	2028825	2,62936E+11	290677	3,36 dnia
2048x2048	65536	4133089	1,08346E+12	1197778	13,86 dnia
3840x2160	129600	8204625	4,25328E+12	4702031	54,42 dnia
4096x2160	138240	8753745	4,84047E+12	5351177	61,93 dnia
7680x4320	518400	32997825	6,84243E+13	75643570	875,50 dnia



Rys. 4. Wykres zależności czasu kompresji od rozdzielczości kompresowanego obrazu. w skali logarytmicznej

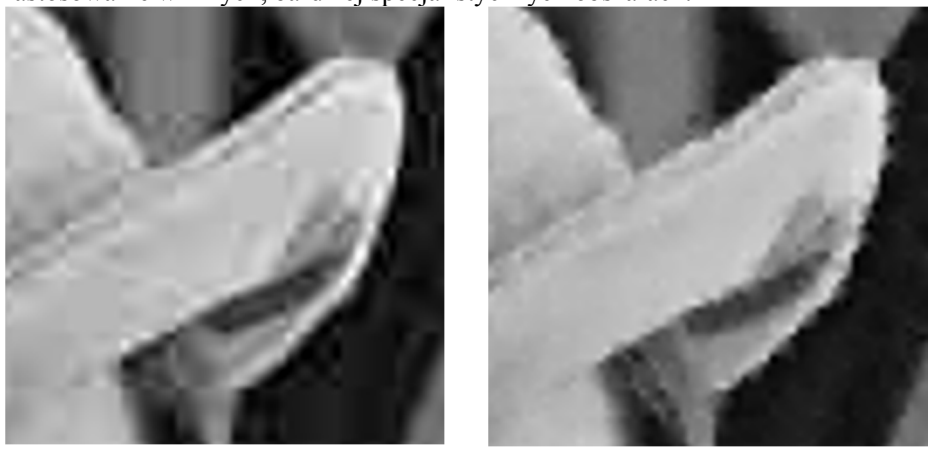
4. Kompresja fraktalna a JPEG

Najpopularniejszym obecnie sposobem kompresji obrazów jest algorytm JPEG. Bazuje on na dyskretnej transformacie kosinusowej, która jest stosowana między innymi w kompresji obrazów i filmów. JPEG charakteryzuje się dużą szybkością zarówno procesu kompresji jak i dekompresji. Jakość kompresji jest bardzo dobra przy niskim stopniu kompresji. Przy wyższym stopniu kompresji mogą pojawiać się tzw. artefakty, w obrazie mogą być widoczne poszczególne bloki pikseli o wymiarach 8x8 co powoduje znaczne pogorszenie jakości lecz skutkuje znacznie zmniejszoną ilością pamięci potrzebną do zapisu obrazu. JPEG wykazuje gorszą jakość obrazu skompresowanego przy kompresji obrazu na którym widnieje tekst, lub obrazów z nagłymi zmianami kontrastu i wyraźnymi podziałami na obrazie.

Kompresja fraktalna zalicza się do asymetrycznych algorytmów kompresji, oznacza to iż czas procesu kompresji różni się w znacznym stopniu od czasu dekompresji. W przypadku kompresji fraktalnej, czas procesu kompresji nieporównywalnie większy od czasu dekompresji, spowodowane jest to złożonością obliczeniową i ilością porównań które trzeba przeprowadzać pomiędzy blokami i obszarami obrazu. Czas jest główną wadą kompresji fraktalnej. Jednakże posiada ona wiele zalet. Jakość skompresowanego obrazu w zależności od parametrów algorytmu może być bardzo dobra przy dużym stopniu kompresji, lecz przekłada się to na dłuższy czas działania algorytmu. W odróżnieniu od JPEG, nie mają tutaj znaczenia duże, nagłe zmiany kontrastu w obrazie, ponadto kompresja fraktalna jest niezależna od rozdzielczości – jest to jedna z cech obiektów fraktalnych, oznacza to

iż obraz może być dekompresowany do dowolnej wielkości bez strat na jakości obrazu, czego nie można zrobić za pomocą algorytmu JPEG. Na poniższym rysunku (Rys. 5.) można zaobserwować różnice pomiędzy kompresją fraktalna i JPEG. Aby różnice były lepiej uwidocznione, obraz został skompresowany algorytmem JPEG z parametrem jakości 20 tak aby ilość bajtów niezbędna do zapisu była taka sama jak obrazu skompresowanego metodami fraktalnymi. Zostały również powiększone odpowiadające sobie elementy obu obrazów.

Algorytm JPEG jest obecnie najpopularniejszym standardem kompresji stratnej obrazów na świecie. Jest obsługiwany przez wiele urządzeń, stron internetowych, aplikacji. W wielu miejscach stał się standardowym formatem za pomocą którego reprezentowane są obrazy, jak choćby w telefonach komórkowych, wielu aparatach fotograficznych i stronach internetowych. Wszystko to sprawia, że bardzo trudno byłoby zastąpić to rozwiązanie innym, jednakże kompresja fraktalna może znaleźć zastosowanie w innych, bardziej specjalistycznych obszarach.



Rys. 5. Fragment obrazu "Lenna" przybliżenie 4 krotne. Obraz po lewej skompresowany za pomocą JPEG o jakości 20 (4417 bajtów). Obraz po prawej skompresowany kompresją fraktalną (4351 bajtów)

5. Podsumowanie

W pracy zaprezentowano metodę stratnej kompresji obrazów z wykorzystaniem metod fraktalnych. Przedstawiono również ogólny sposób działania algorytmu kompresji fraktalnej oraz bardziej obfitujący w szczegóły opis badanego wariantu algorytmu wraz z wynikami jego działania.

Kompresja obrazów metodami fraktalnymi daje bardzo dobre rezultaty zarówno w jakości kompresowanego obrazu jak i stopniu kompresji. Porównanie z algorytmem JPEG pokazało iż fraktalne metody kompresji obrazów wolne są od

wad jakie niesie ze sobą kompresja JPEG i wykorzystana w niej dyskretna transformata kosinusowa. Ponadto metody fraktalne dają dodatkowo możliwość dekompresji obrazu do dowolnej wielkości bez utraty jakości co jest cechą wszystkich obiektów fraktalnych. Mimo swoich zalet metody kompresji fraktalnej posiadają jedną bardzo znaczącą wadę, co zostało wykazane w badaniu. Metody kompresji fraktalnej posiadają bardzo dużą złożoność obliczeniową potrzebną do wykonania procesu kompresji obrazu co przekłada się bezpośrednio na bardzo długi czas działania algorytmu. Czas kompresji wyklucza obecnie algorytmy fraktalne z zastosowania tam gdzie kompresja obrazu powinna odbywać się w czasie rzeczywistym.

Aby poprawić wydajność algorytmu kompresji fraktalnej możliwe jest przeprowadzenie jego optymalizacji. Algorytm posiada wiele parametrów takich jak wielkość bloków i obszarów oraz ich kształt, odległości między blokami, warianty obrotu obszarów, współczynnik skalowania obszaru do bloku. Parametry te mogą zostać dobrane w sposób optymalny tak aby jak najbardziej skrócić czas działania algorytmu bądź też poprawić jakość skompresowanego obrazu, jednakże optymalizacja wieloparametrowa jest niezwykle trudnym zagadnieniem. Innym sposobem optymalizacji jest wstępna analiza obrazu i jego segmentacja na obszary podobne, które będą kompresowane niezależnie od siebie, równolegle z pozostałymi obszarami. Kolejnym sposobem optymalizacji jest odpowiednie zawężenie obszaru poszukiwań bloku podobnego tak aby jak najbardziej zmniejszyć złożoność obliczeniową algorytmu pozostawiając ten sam stopień kompresji obrazu. Możliwe jest również zastosowanie algorytmów genetycznych, które dobrze sprawdzają się przy wyszukiwaniu rozwiązań optymalnych w dużych zbiorach rozwiązań. Dodatkowo algorytmy genetyczne mogą być w łatwy sposób zrównoleglone co może mieć znaczący wpływ na szybkość działania algorytmu. Dalszy rozwój algorytmu kompresji fraktalnej związany jest głównie z badaniami nad metodami jego optymalizacji, w szczególności nad zastosowaniem algorytmów genetycznych oraz wstępnej analizy obrazu, które znacząco mogą wpłynąć zarówno na szybkość działania algorytmu jak i polepszyć stopień kompresji.

Literatura

1. Yuval Fisher: *Fractal Image Compression Theory and Application*. Springer-Verlag 1995
2. Stephen Welstead: *Fractal and Wavelet Image Compression Techniques*. SPIE Washington 1999
3. William B. Pennebaker, Joan L. Mitchell*: JPEG Still Image Data Compression Standard*. Kluwer Academic Publishers 2004

Streszczenie

Pod pojęciem kompresji obrazów kryją się najróżniejsze algorytmy kompresji danych. Najczęściej są to algorytmy kompresji stratnej, które charakteryzują się znacznym stopniem kompresji, lecz ich wadą jest utrata informacji podczas procesu kompresji. W niniejszym artykule zostanie opisany algorytm kompresji fraktalnej, który zalicza się do algorytmów kompresji stratnej i najczęściej wykorzystywany jest przy kompresji obrazów. Zostanie również przedstawiony opisu działania algorytmu z wykorzystaniem systemu funkcji iterowanych oraz jego wady i zalety. Celem artykułu jest porównanie możliwości kompresji metodą fraktalną z najpopularniejszą obecnie metodą kompresji stratnej obrazów JPEG. Uzyskane dane posłużą do dalszej analizy możliwości algorytmu kompresji fraktalnej oraz metod jego optymalizacji.

Słowa kluczowe: kompresja fraktalna, kompresja obrazów, kompresja stratna, fraktal

Summary

The term image compression hides various data compression algorithms. Usually these are lossy compression algorithms, which characterize high degree of compression, but their disadvantage is the loss of information during compression process. The article describes fractal compression algorithm, which is one of the lossy compression algorithms and is used mostly in image compression. It will be also described algorithm details using iterated function system and its advantages and disadvantages. The aim of the article is to compare the capabilities of fractal compression method with JPEG which is currently the most popular method of lossy image compression. The obtained data will be used to further analyze possibilities of fractal compression and methods of its optimalization.

Keywords: fractal compression, iamge compression, lossy compression, fractal

Paweł Poczekajło
Wydział Elektroniki i Informatyki
Politechnika Koszalińska
ul. JJ Śniadeckich 2, 75-453 Koszalin

Badanie dokładności rotatora opartego na algorytmie CORDIC w systemie o skończonej precyzji obliczeń

Słowa kluczowe: rotator, algorytm CORDIC, skończona precyzja obliczeń, dokładność

1. Wstęp

Rotator (z matematycznego punktu widzenia) jest to funkcja realizująca obrót punktu lub obiektu o danych współrzędnych wokół określonego początku układu współrzędnych. Dla tradycyjnego dwuwymiarowego (2D) układu współrzędnych kartezjańskich, rotator opisany jest w następujący sposób [1]:

$$\begin{aligned} x_1 &= x_0 \cos(\alpha) - y_0 \sin(\alpha) \\ y_1 &= x_0 \sin(\alpha) + y_0 \cos(\alpha) \end{aligned} \tag{1}$$

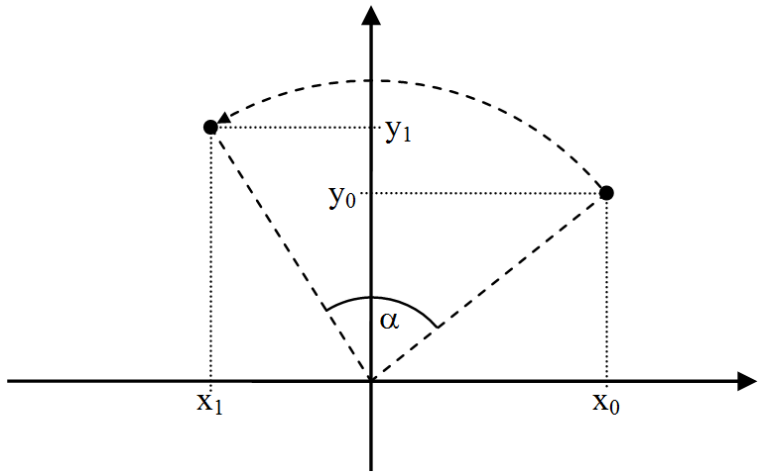
gdzie:

x_0, y_0 – współrzędne punktu przed obrotem,

x_1, y_1 – współrzędne punktu po obrocie,

α – kąt obrotu punktu wokół początku układu współrzędnych.

Graficzną interpretację rotacji punktu (1) przedstawiono na rysunku 1.



Rys. 1. Graficzna interpretacja rotacji punktu w układzie współrzędnych

Rotatory jako funkcje matematyczne są znane i wykorzystywane już od wielu lat. Nawet w dobie powszechnej digitalizacji, cyfrowe odpowiedniki rotatorów są powszechnie używane, np. w systemach graficznych [2] lub układach cyfrowego przetwarzania sygnałów (CPS) [3, 4]. W kolejnych punktach przedstawiono możliwość realizacji sprzętowej rotatora przy wykorzystaniu algorytmu CORDIC [5].

2. Realizacja sprzętowa rotatora

Niniejszy artykuł skupia się na problemie implementacji rotatorów w układach FPGA, gdzie bardzo istotnym uwarunkowaniem jest skończona precyzja zapisu zmiennych. Przy realizacji rotatorów w bezpośredni sposób niezbędne jest wykonanie aż czterech mnożeń i dwóch sumowań. Problemem są tu zwłaszcza układy mnożące, które wymagają użycia znacznych zasobów lub specjalnych bloków DSP procesora FPGA.

Alternatywą jest tu możliwość zastosowania algorytmu CORDIC, który oparty jest na zasadzie kolejnych przybliżeń i wykorzystuje jedynie multipleksery i sumatory. Poniżej przedstawiony został iteracyjny algorytm CORDIC wykorzystany do wykonywania rotacji w systemach CPS [6]:

$$
\begin{aligned}
a_{i+1} &= a_i / 2 \\
p_{i+1} &= p_i / 2 \\
r_{i+1} &= r_i / 2 \\
s_{i+1} &= \begin{cases} s_i + a_i \text{ dla } s_i < 0 \\ s_i - a_i \text{ dla } s_i > 0 \\ s_i \text{ dla } s_i = 0 \end{cases} \\
c_{i+1} &= \begin{cases} c_i + a_i \text{ dla } c_i < 0 \\ c_i - a_i \text{ dla } c_i > 0 \\ c_i \text{ dla } c_i = 0 \end{cases} \\
x_{1i+1} &= \begin{cases} x_{1i} + p_i \text{ dla } c_i \geq 0, s_i < 0 \\ x_{1i} - p_i \text{ dla } c_i < 0, s_i \geq 0 \\ x_{1i} - r_i \text{ dla } c_i < 0, s_i < 0 \\ x_{1i} + r_i \text{ dla } c_i > 0, s_i \geq 0 \text{ lub } c_i \geq 0, s_i > 0 \\ x_{1i} \text{ dla } c_i = 0, s_i = 0 \end{cases} \\
y_{1i+1} &= \begin{cases} y_{1i} + r_i \text{ dla } c_i < 0, s_i \geq 0 \\ y_{1i} - r_i \text{ dla } c_i \geq 0, s_i < 0 \\ y_{1i} - p_i \text{ dla } c_i < 0, s_i < 0 \\ y_{1i} + p_i \text{ dla } c_i > 0, s_i \geq 0 \text{ lub } c_i \geq 0, s_i > 0 \\ y_{1i} \text{ dla } c_i = 0, s_i = 0 \end{cases}
\end{aligned}
\tag{2}
$$

gdzie:

$a_0 = 1$,

$s_0 = \sin(\alpha)$,

$c_0 = \cos(\alpha)$,

$p_0 = x_0 + y_0$,

$r_0 = x_0 - y_0$,

$w_0 = 0$,

$z_0 = 0$,

x_{1i+1}, y_{1i+1} – kolejne przybliżenia współrzędnych punktu po rotacji.

Na potrzeby niniejszej publikacji założono, że system o ograniczonej precyzji zapisu liczb oparty jest na publikacji [6] i dopuszcza zapis zmiennych stałoprzecinkowych na 16-bitach, przy czym zachowana musi być możliwość zapisu wartości funkcji trygonometrycznych w zakresie <-1,1>. Stąd przyjęto zapis stałopozycyjny Q2.14 w systemie U2 dla zmiennych a_i, s_i i c_i oraz zapis Q10.6 dla zmiennych p_i, r_i, w_i i z_i. Dane wejściowe systemu to wartości całkowite z zakresu od -128 do 127 (zgodnie z systemem U2 8-bit). Pobudzenie o takiej postaci zostaje

rozszerzone do formatu Q10.6 [6] oraz w odpowiednio analogiczny sposób dane wyjściowe przekształcane są ponownie do formatu U2 8-bit uwzględniając zaokrąglenie do wartości całkowitej.

W kolejnym rozdziale przedstawiono analizę działania oraz wyniki obliczeń rotatora w systemie o skończonej precyzji obliczeń przy zastosowaniu algorytmu CORDIC (2).

3. Badanie i analiza dokładności rotatora

Zbadanie dokładności działania rotatora opartego na algorytmie CORDIC opiera się głównie na analizie współrzędnych wyjściowych po kolejnych iteracjach. W związku z tym dokonano dokładnej analizy pojedynczych rotatorów oraz zbadano wyniki końcowe dla większej losowej grupy. W tabeli 1 zebrano wartości wyjściowe dla trzech przykładowych rotatorów o losowo dobranych parametrach początkowych.

Jak widać z tabeli 1, przy uwzględnieniu zaokrągleń, już po 8-9 iteracjach algorytmu uzyskujemy wynik zgodny z oczekiwaną wartością. Warto tu jednak zaznaczyć, że minimalna ilość iteracji, koniecznych do uzyskania poprawnego wyniku, jest różna dla różnych danych wejściowych x_0, y_0, i α. Może to skutkować uzyskaniem prawidłowych wartości na wyjściu rotatora nawet dopiero po 10-11 iteracjach. Zauważyć można niewielką niedokładność wartości x_1 w 11-tej iteracji rotatora nr 3. Wynika ona z faktu, że w 10-tej iteracji osiągnięto już prawidłowy wynik dla przyjętego systemu zapisu liczb. Pomimo tego, zgodnie z (2), system w kolejnej iteracji dokonał następnego „przybliżenia" wyniku, co realnie skutkowało jego pogorszeniem. Uniknięcie takiego błędu dla (2) wymagałoby przekształcenia i wyprowadzenia na nowo całego algorytmu oraz uzupełnienia o dodatkowe warunki, co znacznie zwiększyło by złożoność obliczeniową systemu. W efekcie miałoby to również wpływ na znacznie większą zajętość struktury w docelowym układzie sprzętowym. Biorąc pod uwagę, iż dane wyjściowe sprowadzane są do wartości całkowitych z zakresu <-128,127>, dla niedokładności {-1,1}, błąd wynosi ok. 0.39% i jest praktycznie pomijalny.

Dodatkowa próba dokładności polegała na przebadaniu wyników rotacji dla 1000 rotatorów o losowo dobranych parametrach x_0, y_0 i kącie rotacji α. Próba ta wykazała, że dla wartości wynikowych x_1 i y_1 przekształconych i zaokrąglonych do 8-bitowego systemu U2, 816 próbek miało prawidłowe wyniki, natomiast 184 próbki miały błąd o wartość ±1 (w porównaniu do wartości przewidywanej). Podobnie jak we wcześniejszym przypadku, daje to błąd na poziomie ok. 0.39% dla 18,4% odpowiedzi systemu. Należy tu podkreślić, iż są to wyniki otrzymane na podstawie grupy losowej, która zależnie od konkretnych wartości może dawać mniejsze lub większe błędy.

Tabela 1. Wartości wyjściowe dla kolejnych iteracji algorytmu CORDIC

	Parametry rotatora nr 1: $x_0 = 57$, $y_0 = -92$, $\alpha = -37.672354°$		Parametry rotatora nr 2: $x_0 = 12$, $y_0 = 105$ $\alpha = 79.384715°$		Parametry rotatora nr 3: $x_0 = -65$, $y_0 = -108$ $\alpha = -43.563295°$	
Numer iteracji	x_1 po danej iteracji	y_1 po danej iteracji	x_1 po danej iteracji	y_1 po danej iteracji	x_1 po danej iteracji	y_1 po danej iteracji
1	-35.000000	-149.000000	-93.000000	117.000000	-173.000000	-43.000000
2	-17.500000	-74.500000	-46.500000	58.500000	-86.500000	-21.500000
3	-26.250000	-111.750000	-75.750000	35.250000	-129.750000	-32.250000
4	7.625000	-116.125000	-90.375000	23.625000	-108.125000	-26.875000
5	-5.437500	**-106.812500**	-96.187500	**30.937500**	-118.937500	-29.562500
6	-10.093750	-105.718750	-99.843750	28.031250	-124.343750	-30.906250
7	**-10.640625**	-108.046875	**-101.296875**	29.859375	-123.671875	-33.609375
8	-11.812500	**-107.765625**	-100.375000	**30.593750**	**-122.312500**	**-33.265625**
9	**-11.218750**	**-107.906250**	**-100.750000**	**31.062500**	**-121.625000**	**-33.093750**
10	**-11.140625**	**-107.609375**	**-100.937500**	**31.296875**	**-121.531250**	**-33.437500**
11	**-10.984375**	**-107.656250**	**-101.062500**	**31.203125**	-121.359375	**-33.390625**
Wartość oczekiwana (w systemie U2 Q10.6)	-11.109375	-107.65625	-101	31.140625	-121.53125	-33.46875
Wartość oczekiwana (w systemie U2 8-bit)	**-11**	**-107**	**-101**	**31**	**-122**	**-33**

4. Wnioski

Niniejszy artykuł skupia się na problemie realizacji rotatora w systemach o skończonej precyzji obliczeń z wykorzystaniem algorytmu CORDIC. W efekcie przeprowadzonych badań i analiz (rozdział 3) można stwierdzić, że wyniki są bardzo dobre i taka realizacja rotatorów jest idealna do implementacji w słabszych układach scalonych czy mikroprocesorach. Dodatkowym atutem jest tu możliwość regulacji dokładności poprzez zmianę ilości iteracji lub nawet korektę dokładności zapisu różnych parametrów. Oczywiście zmniejszenie iteracji lub rozdzielczości bitowej ma też wpływ na zmianę rozmiarów struktury i zajętości w układzie docelowym. Możliwość stosowania rotatorów do realizacji filtrów FIR [3, 4, 6], daje ciekawą alternatywę wobec klasycznej realizacji splotu, która często wymaga użycia wielu bloków mnożących, co zazwyczaj uniemożliwia potokową realizację przetwarzania. Struktury rotatorowe bazujące na CORDIC'u, ze względu na organizację obliczeń, od razu przetwarzają dane w sposób potokowy, tzn. za każdym cyklem zegara pobierana jest jedna próbka wejściowa i jednocześnie zwracana jest jedna próbka wyjściowa. Dalsze prace i badania będą skupiały się na

modyfikacji (2) oraz zmianie rozdzielczości bitowej, tak aby otrzymać jeszcze mniejsze struktury sprzętowe oraz w miarę możliwości bardziej dokładne rotatory sprzętowe bazujące na algorytmie CORDIC.

Bibliografia

1. Downing D.: *Trigonometry*, Barron's Educational Series, 2001, ISBN-10: 0764113607
2. Liu Z. Y., Zhao X. Z.: *Research and Implementation of Image Rotation Based on CUDA*, Advanced Materials Research, vol. 216, str. 708-712, 2011
3. Wirski R., Wawryn K., Strzeszewski B.: *State-space approach to implementation of FIR systems using pipeline rotation structures*, ICSES 2012
4. Wawryn, K., Wirski, R.T., Strzeszewski, B.: *Implementation of finite Impulse Response Systems Using Rotation Structures*, ISITA 2010, Taiwan
5. Volder J. E.: *The CORDIC trigonometric computing technique*, IRE Trans. Electron. Comput., vol. EC-8, nr. 3, str. 330-334, wrzesień 1959
6. Wawryn K., Poczekajło P., Wirski R.: *FPGA implementation of 3-D sepa-rable Gauss filter using pipeline rotation structures*, MIXDES 2015, Toruń

Streszczenie

W artykule dokonano pomiaru dokładności algorytmu CORDIC stosowanego do realizacji rotatora używanego m.in. w dedykowanych systemach CPS. Badania dotyczyły implementacji struktury w układzie o skończonej precyzji obliczeń. Wykonane zostały szczegółowe pomiary wyników algorytmu dla poszczególnych iteracji oraz dokonano ogólnej analizy dla większej grupy losowej. Przedstawione wyniki dały podstawę do oceny prawidłowego działania algorytmu oraz wykazały zalety i wady takiego podejścia do realizacji sprzętowej rotatorów.

Abstract

In this paper, the accuracy of the CORDIC algorithm is measured and present. This algorithm is used to the rotation realization, which is utilized e.g. in dedicated DSP systems. The research is related to the structure implementation in a system with finite-precision arithmetic. Detailed measurements of the results of the algorithm for each iteration are made and also general analysis of a larger random group is presented. The results allow to rate CORDIC algorithm and show pros and cons this approach to hardware realization of rotation.

Rafał Wojszczyk
Zakład Podstaw Zarządzania i Informatyki
Wydział Elektroniki i Informatyki
Politechnika Koszalińska
rafal.wojszczyk@tu.koszalin.pl

Włodzimierz Khadzhynov
Katedra Inżynierii Komputerowej
Wydział Elektroniki i Informatyki
Politechnika Koszalińska

Wykorzystanie modeli danych do weryfikacji implementacji wzorców projektowych

Słowa kluczowe: wzorce projektowe, model danych, ERD, weryfikacja oprogramowania

1. Wstęp

Ward Cunningham to znana postać w społeczeństwie programistów. Cunnigham uzasadniając potrzebę refaktoryzacji oprogramowania wprowadził metaforę finansową nazywaną długiem konstrukcyjnym [11] [6]: niespłacany dług prowadzi do narastających odsetek, im dłużej nie jest spłacany tym większe szanse, że urośnie do przytłaczającej kwoty, z którą przedsiębiorstwo sobie nie poradzi. Analizując tą metaforę można wywnioskować, że niespłacony kredyt jest niepożądaną sytuacją w przedsiębiorstwie. Podobnie oprogramowanie zawierające zbyt skomplikowany kod prowadzi do niepożądanych konsekwencji zarówno dla użytkowników programu jak również dla przedsiębiorstwa odpowiedzialnego za jego wytworzenie. Istnieje wiele metod i ogólnych zaleceń mających na celu ulepszenie każdego z etapów procesu wytwórczego oprogramowania, są to: przygotowanie specyfikacji wymagań systemowych zidentyfikowanych we wstępnej analizie, następnie podczas projektowania niezwykle popularne jest zamodelowanie najważniejszych fragmentów lub całego systemu za pomocą diagramów UML, w implementacji wykorzystywane są odpowiednie narzędzia i dobre praktyki programowania, ostatecznie procedury testowania i konserwacji oprogramowania.

Dobre praktyki w programowaniu obiektowym to m. in. różnego rodzaju wzorce, a w tym wzorce [22]:

- architektoniczne (np. MVC) łączące różne warstwy aplikacji,
- projektowe [7] obejmujące swoim zakresem poziom klas,
- implementacyjne, nazywane również idiomami, występujące na poziomie wierszy kodu.

Jedne z najpopularniejszych wzorców projektowych zostały przedstawione w [7], są to za [16]: szkielety gotowych mechanizmów, które można wykorzystywać przy rozwiązywaniu typowych problemów projektowania i programowania. Opracowany w 1995 roku katalog wzorców [7] powstał na podstawie doświadczenia praktyków programowania [11] i do dziś wykorzystywany jest przez wielu programistów. Ze względu na dużą popularność i brak formalnej kontroli pojawia się potrzeba weryfikacji implementacji wzorców projektowych w powstającym oraz istniejącym oprogramowaniu.

Celem artykułu jest prezentacja modeli danych wykorzystywanych w procesie weryfikacji implementacji wzorców projektowych. Wynik weryfikacji jest niezbędny do podjęcia próby oceny jakości implementacji wzorców projektowych, jest to dalekosiężny cel prowadzonych prac. Kontekst zagadnienia oraz problem weryfikacji przedstawia rozdział 2. W rozdziałach 3 i 4 zawarto prezentację wykorzystywanych modeli danych, natomiast 5 rozdział to podsumowanie artykułu.

2. Weryfikacja implementacji wzorców projektowych

2.1. Kontekst problemu

Forma opisu wzorców projektowych przedstawiona w [7] jest z założenia przeznaczona do celów dydaktycznych, zawiera: opis słowny w języku naturalnym, diagramy klas w notacji OMT oraz przykładowy kod implementacji oparty o proste przykłady. Wzorce opisane w ten sposób przedstawiają zazwyczaj jeden z wariantów implementacji, który nie jest optymalnym rozwiązaniem [11]. Dlatego programista bazując na informacjach przedstawionych w literaturze powinien przy każdej implementacji danego wzorca wzbogacić wytwarzany przez siebie kod o wiele czynników związanych m.in. z logiką biznesową, aby lepiej dopasować implementację wzorca do kontekstu programu. Te czynniki zwiększają złożoność i różnorodność implementacji wzorców, czego efektem jest występowanie wielu wariantów implementacji oraz zacieranie się granic wzorców projektowych. Ostatecznie prowadzi to do wzrostu pracochłonności podczas inspekcji oraz konserwacji kodu.

Wzorce projektowe bardzo często wiązane są wyłącznie z etapem projektowania oprogramowania a nie z kodem programu [11]. Jest to wąski punkt

widzenia, szczególnie gdy za implementację oprogramowania odpowiedzialny jest mały zespół pracujący według metodyki Scrum [15]. W metodykach zwinnych, z których wywodzi się Scrum, projektowanie oprogramowania oraz dokumentowanie jest mniej ważne niż dostarczenie działającego oprogramowania. Często decyzje o rozwiązaniu pewnego problemu poprzez implementację wzorca są podejmowane na szybko podczas porannych spotkań (tzw. poranny Scrum). Następnie praca nad implementacją może być oddelegowana do członka zespołu, który nie miał wcześniej związku z danym wzorcem. Efektem jego pracy może być implementacja, w której wyłącznie pozornie występuje wzorzec, tj. oprogramowanie może działać prawidłowo i przechodzić wymagane testy, jednakże implementacja samego wzorca będzie niezgodna z zaleceniami i nie będzie spełniała postawionego celu. Problemy wynikające z tego faktu będą najbardziej zauważalne przy próbie rozbudowy lub modyfikacji danego fragmentu oprogramowania. Opisany przykład jest tylko jednym z wielu, które wskazują potrzebę weryfikacji implementacji wzorców projektowych.

Istnieją różne metody i narzędzia przeznaczone do testowania oraz analizy kodu źródłowego oprogramowania. Często dostarczone są razem ze zintegrowanym środowiskiem programistycznym, które zadba o zgodność elementarnych rzeczy z syntaktyką języka i założeniami danego paradygmatu programowania. Możliwości środowisk programistycznych można rozszerzyć dzięki wielu zautomatyzowanym narzędziom, które nie uwzględniają wzorców projektowych [14]. Programista wykorzystując elementy bazowe tworzy własne artefakty oprogramowania (np. biblioteki.dll) zawierające wyspecyfikowaną logikę biznesową, która nie może być sprawdzona przez uniwersalny algorytm. Większość artefaktów wymaga indywidualnego podejścia i przygotowania niezbędnych danych oraz funkcji testujących, jest to często realizowane przez testy jednostkowe. Pozytywny wynik testu jednostkowego dla danego artefaktu nie jest równoznaczny z prawidłową implementacją wzorców projektowych w tym artefakcie.

W metodach związanych z wzorcami projektowymi bardzo często podejmowany jest problem wyszukiwania wystąpień wzorców projektowych w oprogramowaniu [20] [18] [1]. Miarą wymienionych metod jest przedstawienie ilości wystąpień wzorców, jest to niewystarczająca informacja do realizacji stawianej potrzeby. Inne metody [2] skupiają się głównie na wykazaniu poprawności strukturalnej, niestety wymaga to nauki dedykowanego języka programowania do opisu danych implementacyjnych. Podejmowane są również próby stosowania znanych metryk oprogramowania [12] do implementacji wzorców projektowych, jednakże w [9] wykazano, że występowanie wzorców w oprogramowaniu może niekorzystnie wpływać na wyniki metryk.

W przypadku praktyków programowania, np. opisanych wcześniej członkach zespołach Scrumowych, wybór odpowiedniego rozwiązania do weryfikacji wzorców projektowych podyktowany jest odpowiednimi walorami użytkowymi.

Jest to niezbędne, żeby nowe rozwiązanie zostało pozytywnie przyjęte. Rozwiązanie takie musi być dostosowane do umiejętności potencjalnych odbiorców oraz nie powinno wymagać nauki nowych koncepcji. Formalne reprezentacje danych, takie jak wykorzystanie ontologii w informatyce, są stosunkowo nowe i wydają się być mało popularne. Powszechnie znane UML (oraz podobne, np. OMT) w zastosowaniu do wzorców projektowych, jest uznawane za pół-formalną reprezentację [17]. Wynika to z pominięcia w diagramach UML niektórych aspektów implementacji wzorców [17], ograniczonych możliwości weryfikacji implementacji. W [18] [20] autorzy bazując na diagramach klas wykazali, że w takim podejściu nie możliwe jest odróżnienie wzorca strategii od stanu [20] oraz identyfikacja wzorca Singleton [18].

2.2. Weryfikacja implementacji

Weryfikacja implementacji danego wzorca projektowego to zespół czynności, które należy wykonać aby wykazać zgodność badanego kodu programu z regułami i zasadami implementacji przyjętymi dla danego wzorca. Proces weryfikacji realizowany jest w oparciu o autorski model oceny jakości implementacji wzorców projektowych [24].

Badany kod programu, dokładniej fragment kodu źródłowego oprogramowania (lub kodu pośredniego) zawiera zaimplementowany konkretny wzorzec projektowy. Kod badanego oprogramowania, w proponowanym podejściu, jest przekształcany do formalnej reprezentacji (ekwiwalent kodu) i wyłącznie ta postać jest wykorzystywana w weryfikacji. Formalna reprezentacja oprogramowania to pierwszy z omawianych modeli danych.

Drugi proponowany model danych to repozytorium implementacji wzorców projektowych. Bazując częściowo na rozwiązaniu przedstawionym w [17] zaproponowano repozytorium, które zakłada opisanie ogólnych definicji wzorców jako zbiór cech tworzących dany wzorzec. Każda z cech jest następnie opisana przez szczegółowe dane implementacyjne, które uwzględniają różne warianty występowania wzorców. Szczegóły implementacyjne w proponowanym podejściu zostały opisane w modelu referencyjnym, jednakże cechy wzorców mogą być uszczegółowione na różne sposoby [21]: poprzez dodatkowy opis w postaci reguł lub odpowiednie operacje wykonywane na kodzie badanego programu.

Przed pierwszym wykonaniem procesu weryfikacji należy jednokrotnie uzupełnić repozytorium opisami wzorców, następnie nowe warianty wzorców mogą być dodawane na podstawie badanego oprogramowania. W uproszczonym ujęciu, dla jednego zaimplementowanego wzorca projektowego, proces weryfikacji rozpoczyna się od przekształcenia kodu programu do formalnej reprezentacji, następnie wykonywana jest weryfikacja zgodnie z występującymi cechami. W przypadku wykorzystania modelu referencyjnego proces weryfikacji sprowadza się do porównania fragmentu ekwiwalentu kodu do danych implementacyjnych,

które wskazywane są przez opis każdej cechy występującej w danym wzorcu projektowym.

Modele danych omówione w dalszej części pracy mogłyby zostać przedstawione za pomocą diagramów klas UML. Jednakże po uwzględnieniu wcześniej wymienionych informacji o zastosowaniu UML do wzorców projektowych, proponowane modele danych zostały opracowane w oparciu o diagramy związków-encji, co dodatkowo pozwoliło na wyróżnienie przewidzianych encji. ERD są jednymi z popularniejszych diagramów przeznaczonych do modelowania danych. Diagramy przedstawione na rysunkach 1, 2, zostały przygotowane z wykorzystaniem programu narzędziowego Power Designer 15 z użyciem notacji Barkera. Wykorzystane narzędzie pozwoliło na przystępną implementację tych modeli jako bazy danych.

3. Formalna reprezentacja kodu źródłowego oprogramowania

3.1. Motywacja

Najwięcej informacji o programowaniu jest zawartych w kodzie źródłowym, którego jednocześnie wadą jest fizyczna reprezentacja – są to odpowiednio skatalogowane pliki tekstowe, które trudniej analizować niż ustrukturyzowaną reprezentację formalną. W kilku podejściach [8], [13], związanych z analizowaniem wzorców projektowych, wykorzystano transformację kodu programu do formalnej reprezentacji. Uzyskany w ten sposób ekwiwalent kodu programu pozwala na zautomatyzowane przetwarzanie danych z występującą implementacją, a dodatkowo umożliwia usunięcie zbędnego kodu i niepotrzebnych informacji (np. komentarzy zapisanych w języku naturalnym, kodu testów jednostkowych). Niestety istniejące sposoby reprezentacji oprogramowania czy też wzorców projektowych nie spełniają postawionych dalej wymagań, co przyczyniło się do opracowania nowych rozwiązań.

3.2. Wymagania

Wzorce projektowe [7] implementowane są w oprogramowaniu obiektowym, toteż modele pozwalające na weryfikację implementacji powinny bazować (przynajmniej częściowo) na paradygmacie programowania obiektowego. Zasadniczą zaletą tego podejścia jest intuicyjność i przystępność dla osób dobrze znających paradygmat programowania obiektowego, szczególnie dla programistów ale również dla badaczy zajmujących się dziedziną inżynierii oprogramowania. Koleją zaletą jest niezależność od konkretnego języka programowania, ważne aby był to język zorientowany obiektowo.

Dodatkowo przy opracowywaniu modelu danych formalnej reprezentacji badanego oprogramowania, towarzyszyły następujące wymagania:

- reprezentacja struktury obiektowej oprogramowania, w tym oddzielenie elementów składowych klas (osobne encje dla pól, właściwości, metod itd.),
- reprezentacja instancji i zmiany stanów obiektów,
- zasilenie danymi na podstawie kodu zarządzanego [23] oraz możliwość rozbudowy o inne źródła danych,
- realizacja poprzez relacyjną bazę danych.

Jednocześnie zostały nałożone ograniczenia zmniejszające szczegółowość danych względem kodu źródłowego. W proponowanym modelu danych nie zostały odzwierciedlone wartości pól i zmiennych (np. zawartość łańcuchów znaków, dane binarne). Szczegółowość danych została zmniejszona również w przypadku specyficznych elementów nowoczesnych języków programowania, np. typy anonimowe są reprezentowane jako nowe klasy, w tym opracowaniu jest to zgodne ze standardem ECMA-355 [19]. W konsekwencji postawionych wymagań oraz nałożonych ograniczeń, opracowany model formalnej reprezentacji oprogramowania nie jest bezpośrednim odpowiednikiem żadnego języków programowania i nie mógł powstać automatycznie.

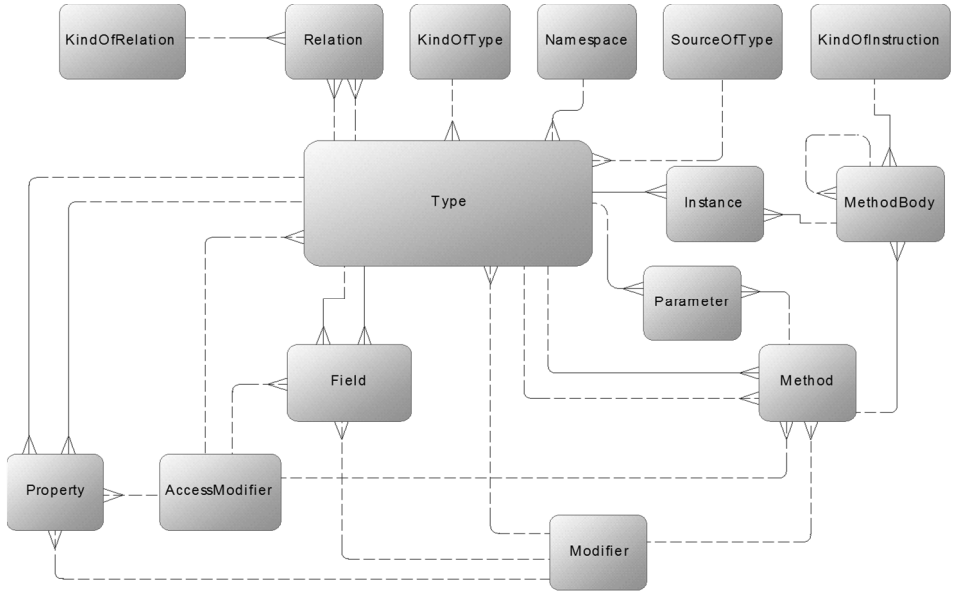
3.3. Model danych

Proponowany model danych oraz algorytm transformacji oprogramowania zostały wstępnie przedstawione w [24]. Rysunek 1 przedstawia rozbudowaną wersję modelu. Elementem o największej odpowiedzialności jest encja *Type*, odpowiada za reprezentację typów (np. konkretna klasa) w paradygmacie programowania obiektowego. Encja ta zawiera:

- kolekcja *Field* – pojedynczy element kolekcji reprezentuje pole (tzw. zmienna globalna) w danym typie, scharakteryzowany jest przez nazwę, typ (określa typ danych przechowywanych przez pole, realizowane przez osobną relację do encji *Type*) oraz encje słownikowe modyfikatora dostępu i modyfikatora,
- kolekcja *Property* – pojedynczy element kolekcji reprezentuje właściwość (tzw. setter lub getter), scharakteryzowany jest przez analogiczne pola i encje jak *Filed*, a dodatkowo określenie funkcji Set i Get tj. zapis i odczyt danych,

- kolekcja *Method* – pojedynczy element kolekcji reprezentuje metodę (funkcję) występującą w danym typie, scharakteryzowany jest przez nazwę, określenie czy metoda jest konstruktorem, typem (określa typ danych zwracanych przez metodę) oraz encje słownikowe modyfikatora dostępu i modyfikatora, dodatkowe encje zostały opisane w dalszej części,
- kolekcja *Relation* – pojedynczy element kolekcji reprezentuję zależność rozpatrywanego typu od innego typu wraz z dookreśleniem rodzaju zależności (np. dziedziczenie, realizacja) poprzez encję słownikową *KindOfRelation*,
- *KindOfType* – określa rodzaj danego typu (np. klasa, interfejs),
- *SourceOfType* – źródło pochodzenia (z badanego oprogramowania, typ systemowy lub zaślepkowy [23]),
- *Namespace* – określenie przestrzeni nazw, w której znajduje się dany typ.

Dodatkowo każdy typ scharakteryzowany jest modyfikatorem i modyfikatorem dostępu.



Rysunek 1. Model danych formalnej reprezentacji oprogramowania

Encja *Method* zawiera kolekcję *Parameter* definiującą parametry przyjmowane przez daną metodę, każdy element kolekcji scharakteryzowany jest przez nazwę oraz oczekiwany typ danych (osobna relacja do encji *Type*). Kolekcja *MethodBody*

określa instrukcje występujące w danej metodzie. Złożone instrukcje są dekomponowane do pojedynczych instrukcji i charakteryzowane odpowiednim rodzajem instrukcji (zgodnie z OpCode [19] kodu zarządzanego, np. tworzenie obiektu) oraz kolejnością występowania. Relacja zwrotna oznacza powiązanie ze sobą poszczególnych instrukcji w instrukcję złożoną. Dodatkowo kolekcja *Instance* grupuje ze sobą instrukcje, które odnoszą się do tej samej instancji danego typu. Instancja danego typu (np. wystąpienie konkretnego egzemplarza danej klasy) scharakteryzowana jest przez nazwę instancji oraz unikatowy Guid (wymagane jeśli nie można uzyskać nazwy instancji).

Encja słownikowa modyfikator (*Modifier*) opisuje modyfikatory (np. abstract, sealed, static, virtual, itd.) występujące przy typach oraz wybranych elementach składowych typów. Encja modyfikator dostępu (*AccessModifier*) określa zasięg widoczności (np. public, private, internal, protected) wybranych elementów. Zawartość encji *Modifier*, *AccessModifier*, *KindOfRelation*, *SourceOfType*, *KindOfInstruction* jest predefiniowana w zależności od języka programowania jaki jest reprezentowany przez model danych (w tym opracowaniu C#). W szczególnym przypadku dopuszczalne jest występowanie wartości „empty" w wymienionych encjach, jeśli pozwala na to dany język programowania.

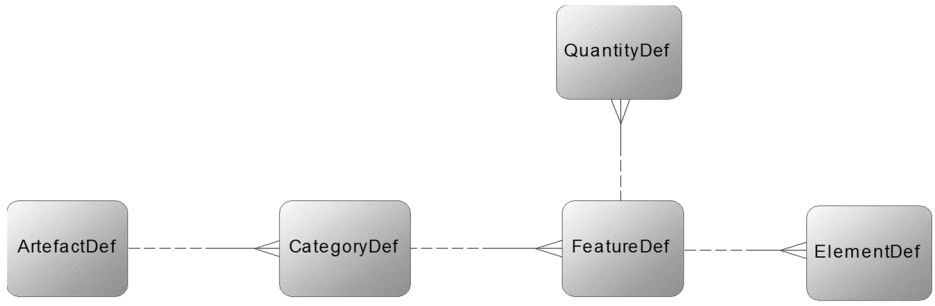
4. Repozytorium implementacji wzorców projektowych

Repozytorium implementacji wzorców projektowych to składnica informacji o możliwych sposobach implementacji rozpatrywanych wzorców projektowych. Z powodu dużej różnorodności implementacyjnej podjęto decyzję o rozdzieleniu ogólnej definicji wzorców od szczegółów implementacyjnych. Definicja wzorca to pewnego rodzaju spis treści, który zawiera odpowiednio dobrane cechy wzorców w oderwaniu od szczegółowej implementacji. Cecha wzorca projektowego to jeden z koniecznych elementów, które charakteryzują dany wzorzec. Bazując na przykładzie wzorca Singleton jedną z cech jest udostępnianie instancji, co można zaimplementować na kilka sposobów (np. pole, właściwość, metoda). Fakt wystąpienia udostępniania instancji jest cechą konieczną tego wzorca, natomiast sposób implementacji jest zależny od różnych czynników (np. kontekst kodu programu, język programowania). Podobne rozdzielenie występuje w normie ISO 9126 określającej jakość produktu programowego. Norma opisana jest przez drzewiastą strukturę, która zawiera za [4] charakterystyki a w nich rekursywnie podcharakterystyki. Charakterystykom liściom przyporządkowane są metryki – funkcje, które wyznaczają pewne wartości na podstawie mierzalnych atrybutów oprogramowania. Definicja charakterystyk jest nieformalna – wyraża pewną intencję, natomiast metryki są sformalizowane. W proponowanym podejściu charakterystykom z ISO 9126 odpowiada model definicji a metrykom szczegółowa implementacja wzorców projektowych.

Odseparowanie cech od implementacji wzorców projektowych pozwala na zwiększenie zakresu weryfikacji. W zależności od badanego aspektu można wykorzystać odpowiednie rozwiązanie: aspekty strukturalne implementacji wzorców projektowych można zweryfikować poprzez porównanie badanego oprogramowania do modelu referencyjnego, natomiast aspekty związane z dynamiką i zachowaniem oprogramowania mogą wymagać weryfikacji opartej o śledzenia zmian stanów obiektów (np. poprzez realizację odpowiednich zapytań na formalnej reprezentacji oprogramowania).

4.1. Model definicji

W proponowanym rozwiązaniu opis cech został zrealizowany jako hierarchiczna struktura danych. Metamodel tej struktury przedstawia rysunek 2. Korzeniem hierarchii i jednocześnie najbardziej ogólna jest encja *ArtefactDef*, która reprezentuje konkretne wzorce projektowe. Jak już zostało wspomniane, definicja każdego wzorca projektowego składa się z odpowiednich cech, które dla ułatwienia zostały zgrupowane w encji *CategoryDef* zawartej w *ArtefactDef*. Cechy (kolekcja encji *FeatureDef*) zgrupowane są w kategoriach według kryterium przynależności do aspektów: struktury tworzącej wzorzec, wykorzystania, zachowania itp. Każda cecha może być zależna od innej cechy, rodzaj zależności określa pole *RelatedType* (niewidoczne na rysunku) będące typem wyliczeniowym. Możliwe rodzaje zależności cech to: koniunkcja, alternatywa oraz zawieranie cech podrzędnych. Cecha scharakteryzowana jest przez nazwę oraz określenie liczebności (encja *QuantityDef*) występowania danej cechy (w szczególnym przypadku liczebność występowania minimum 1 oznacza konieczność wystąpienia cech). Każda cecha zawiera szczegółowe elementy (encja *ElementDef*), które pozwalają na dodatkowe zdekomponowanie cechy np.: cechą struktury tworzącej wzorzec jest konkretna klasa, która dodatkowo jest zdekomponowana na: zależność od innej klasy, modyfikator dostępu określający zasięg widoczności. Każdy element scharakteryzowany jest przez nazwę oraz możliwość zanegowania (dopuszczalne jest wtedy wszystko inne niż dany element). W szczególnym przypadku może występować wyłącznie jeden element danej cechy, co skraca hierarchię do poziomu cech. Encja *ElementDef* jest jednocześnie łącznikiem pomiędzy modelem definicji a opisem szczegółowej implementacji wzorców projektowych.



Rysunek 2. Model definicji

Zaproponowany model danych pozwala na bardzo dużą dowolność w opisie definicji wzorców projektowych. W zależności od wzorca możliwe jest opisanie ogólnej cechy, której implementacja możliwa jest na wiele różnorodnych sposobów, lub wyszczególnienie elementów danej cechy, gdy występuje stała implementacja.

4.2. Model Referencyjny

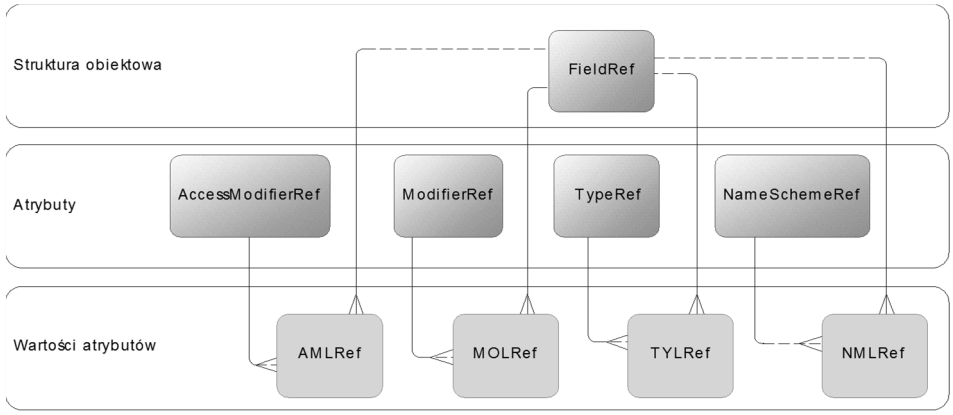
Model referencyjny jest jednym z rozwiązań przeznaczonych do opisu szczegółowej implementacji wzorców projektowych, inaczej danych implementacyjnych. Szczegółowa implementacja to uzupełnienie modelu definicji o zbiór drobnoziarnistych informacji (inaczej atrybutów) [3] związanych bezpośrednio z kodem programu. Drobnoziarnistość informacji pozwala na opis realizacji wzorców w konkretnym języku programowania. Jednocześnie możliwy jest opis generycznych implementacji (inaczej szablonowych). Oznacza to, że występujące dane określają ogólny charakter opisywanych atrybutów bez narzucania konkretnych wartości, np. określenie typu danych pola w klasie jako to dowolny typ numeryczny.

Proponowany model referencyjny jest przewidziany do opisu implementacji struktury tworzącej wzorce projektowe. Struktura tworząca wzorzec to na ogół zbiór klas i interfejsów połączonych ze sobą, np.: we wzorcu Strategia jest to interfejs deklarujący strategię oraz klasy, które implementują ten interfejs. Zgodnie z wcześniejszymi założeniami, badane oprogramowanie, dokładniej dane zawarte w reprezentacji formalnej, jest porównywane z zawartością modelu referencyjnego. Model referencyjny również spełnia wymagania zostały narzucone na formalną reprezentację oprogramowania. Aby zautomatyzować proces porównania, model referencyjny zawiera analogiczną dekompozycję struktury obiektowej na encje jak model formalnej reprezentacji oprogramowania, tj. *TypeRef* zawiera kolekcję *FieldRef* itd. (zgodnie z PascalCase został dodany przyrostek „Ref” do encji modelu referencyjnego w celu łatwiejszego odróżnienia encji pomiędzy modelami, Ref to skrót od Reference). Jednakże dane występujące w tych encjach zostały dodatkowo

zdekomponowane do osobnych encji, które umożliwiają opisanie różnych wariantów drobnoziarnistych atrybutów. Każdy z wariantów oznaczony jest odpowiednim poziomem dopasowania. Poziom dopasowania to określenie stopnia dostosowania danego atrybutu do konkretnego wariantu implementacji. W przyjętym opracowaniu jest to zakres od „0" do „2", gdzie wartość „2" oznacza najwyższy poziom dopasowania. Możliwe jest występowanie wielu atrybutów o jednakowym poziomie dopasowania.

Dane opisywane przez model referencyjny można podzielić na trzy grupy:

1. dane tworzące strukturę obiektową,
2. dane określające atrybuty występujące z danym elementem struktury obiektowej,
3. dane reprezentujące konkretne wartości atrybutów, w tej grupie występuje również określenie poziomu dopasowania.



Rysunek 3. Wybrane encje i relacje opisujące pole w modelu referencyjnym

Podział na wymienione wcześniej trzy grupy danych został zaznaczony na rysunku nr 3, który przedstawia fragment modelu referencyjnego. Na rysunku widoczne są wyłącznie encje opisujące pole. Dla uproszczenia nie widoczne są niektóre relacje, tj. każde pole może być zawarte w konkretnym typie (np. jako element składowy klasy), natomiast każdy typ może być scharakteryzowany przez różne atrybuty. Całość modelu referencyjnego przedstawia rysunek 4. Tabela 1 przedstawia encje tworzące grupę struktury obiektowej w odniesieniu do charakteryzujących je encji atrybutów, co jest zgodne z paradygmatem programowania obiektowego. Dekompozycja pozostałych encji należących do grupy struktury obiektowej została zrealizowana analogicznie jak przedstawiony przykład encji *Field*. Encje *KindOfTypeRef*, *KindOfRelRef*, *ModifierRef*, *AccesModifierRef* to słowniki stałych atrybutów, których wystąpienie wartości (dla danego elementu

struktury obiektowej) przechowywane jest odpowiednio w encjach: *KTLRef*, *KRLRef*, *MOLRef*, *AMLRef*. Znaczenie tych encji jest analogiczne jak w modelu formalnej reprezentacji oprogramowania. Nieco inaczej jest wykorzystywana encja *NameSchemeRef*, której wartości (encja *NMLRef*) pozwalają na opisanie konkretnych nazw zgodnie ze schematami: zaczyna się od, kończy się na, zawiera. Encja *TypeRef* pełni podwójną rolę, zawiera w sobie elementy struktury obiektowej (elementy składowe typów) oraz występuje jako słownik atrybutu typ. Ostatecznie grupa encji reprezentujących konkretne wartości atrybutów jest połączona encją asocjacyjną z *ElementDef* modelu definicji.

Tabela 1. Zestawienie atrybutów występujących z encjami („+" atrybut występuje w danej encji)

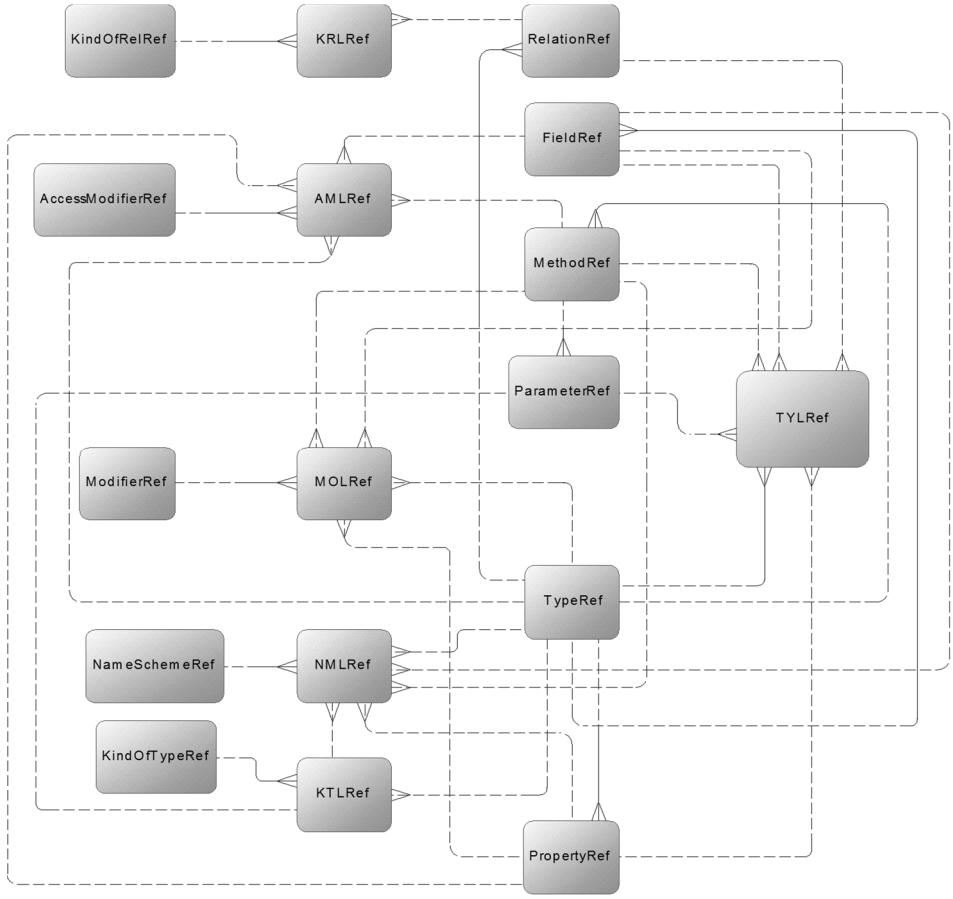
	KindOf-Type-Ref	*KindOf-Rel-Ref*	*Modifier-Ref*	*Acces Modifier-Ref*	*Name Scheme-Ref*	*Type-Ref*
TypeRef	+		+	+	+	
FieldRef			+	+	+	+
Property-Ref			+	+	+	+
MethodRef			+	+	+	+
Parameter-Ref					+	+
Relation-Ref		+				+

4.3. Weryfikacja w oparciu o proponowane modele

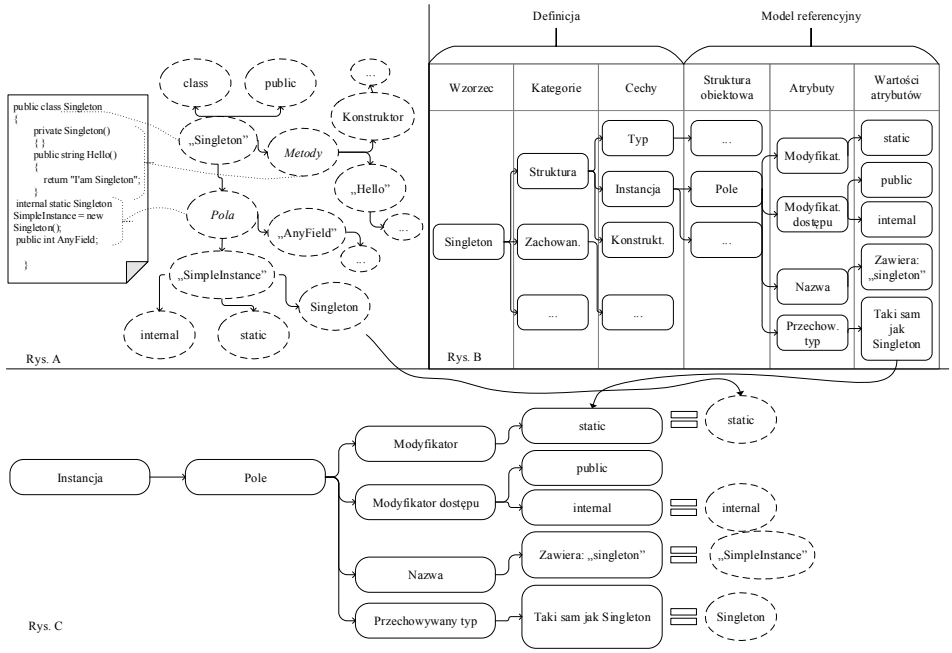
Weryfikacja wzorców projektowych jest złożonym procesem, stąd dążenie do automatyzacji. Nawet dla prostego wzorca, jakim jest Singleton, proces weryfikacji składa się z wielu kroków, których nie można zwięźle opisać. Dotychczasowe prace, tj. przykłady weryfikacji oraz oceny jakości implementacji wzorców projektowych wraz z interpretacją wyników zostały opisane w [24].

Rysunek 5 przedstawia diagram opisujący ideę weryfikacji w oparciu o model referencyjny. W części rys. 5a widoczny jest przykładowy kod źródłowy implementacji wzorca projektowego Singleton oraz wizualizacja kodu (kształty otoczone przerywaną linią) aby lepiej przybliżyć dalsze porównanie. W części rys. 5b widoczna jest uproszczona definicja wzorca projektowego Singleton, jest to wizualizacja danych z modelu referencyjnego. Dodatkowo zaznaczone zostały podziały na hierarchię modelu definicji oraz grupy danych modelu referencyjnego. W celu uproszczenia rys. 5b zostały pominięte mniej znaczące wartości atrybutów (np. brak modyfikatorów) oraz zerowe poziomy dopasowania (są niedozwolone, np.

modyfikator dostępu konstruktora inny niż prywatny). Diagram przedstawia wyłącznie wybrane elementy struktury tworzącej wzorzec, a w omówionym dalej przykładzie ogranicza się to do pola udostępniającego instancję Singleton. Miejsca wykropkowane oznaczają dalsze rozwinięcia danych [24]. W części rys. 5c przedstawiony został wynik porównania. Kształty z wizualizacji formalnej reprezentacji zostały zestawione z kształtami modelu referencyjnego. W przedstawionym przykładzie wystąpił wewnętrzny modyfikator dostępu (niższy poziom dopasowania), którego konsekwencje wyjaśnia tabela 2.



Rysunek 4. Model referencyjny



Rysunek 5. Diagram opisujący ideę weryfikacji w oparciu o model referencyjny

Tabela 2. Konsekwencje wystąpienia wybranych modyfikatorów dostępu w instancji wzorca Singleton

Modyfikator dostępu	Poziom dopasowania	Konsekwencje wystąpienia
public	2	Zalecane, udostępnia instancję Singletonu wszystkim klasom, nie ogranicza wykorzystania.
internal	1	Dopuszczalne, jednakże ogranicza zasięg widoczność wyłącznie do biblioteki. Może powodować problemy przy próbie integracji z badanym oprogramowaniem.
private	0	Niedopuszczalne, uniemożliwi dostęp do instancji, wzorzec nie będzie spełniał swojego przeznaczenia.

5. Podsumowanie

W pracy omówiono krótko potrzebę weryfikacji implementacji wzorców projektowych oraz przytoczono związane z tym problemy. Następnie opisano ogólne założenia autorskiego procesu weryfikacji oraz występujące w nim modele danych. Wynik weryfikacji jest niezbędny do realizacji dalszych prac w kontekście oceny jakości implementacji wzorców projektowych.

Pierwszy z zaprezentowanych modeli danych przeznaczony jest do przechowywania ekwiwalentu kodu źródłowego badanego oprogramowania, drugi przechowuje definicje wzorców projektowych. Oba modele oraz opisany proces zostały zrealizowane jako prototypowe narzędzie. Uzyskane wyniki potwierdziły słuszność przyjętych rozwiązań i stanowią podstawę do dalszych badań dotyczących rozbudowy repozytorium implementacji wzorców o możliwość weryfikacji aspektów behawioralnych, np. w oparciu o diagramy UML, a następnie zaproponowania kryteriów jakości implementacji wzorców projektowych.

Bibliografia

1. Binun A.: *High Accuracy Design Pattern Detection*. Dysertacja doktorska, Rheinischen Friedrich Wilhelms Universitat Bonn, 2012
2. Blewitt A.: *HEDGEHOG: Automatic Verification of Design Patterns in Java*. Dysertacja doktorska, University of Edinburgh, 2006
3. De Lucia A., i inni: *Design pattern recovery through visual language parsing and source code analysis*, Journal of Systems and Software archive, Vol.: 82, Issue 7, Elsevier Science Inc, New York, 2009
4. Dubielewicz I., Hnatkowska B., Huzar Z., Tuzinkiewicz L.: *Wykorzystanie analizy wielokryterialnej w ocenie modeli baz danych*, w: Bazy Danych: Nowe Technologie, Politechnika Śląska, WKŁ, 2007
5. Fowler M.: *UML Distilled: A Brief Guide to the Standard Object Modeling Language*, Wydanie III, Addison Wesley, 2003
6. Fowler M. i inni: *Refaktoryzacja. Ulepszanie struktury istniejącego kodu*, Helion, Gliwice, 2011
7. Gamma E. i inni: *Wzorce projektowe. Elementy oprogramowania obiektowego wielokrotnego użytku*, Helion, Gliwice, 2010
8. Grzanek K.: *Realizacja systemu wyszukiwania wystąpień wzorców projektowych w oprogramowaniu przy zastosowaniu metod analizy statycznej kodu źródłowego*, Dysertacja doktorska, Politechnika Częstochowska, Łódź, 2008
9. Hernandez J. i inni: *Selection of Metrics for Predicting the Appropriate Application of design patterns*, 2nd Asian Conference on Pattern Languages of Programs, 2011

10. Kan S. H.: *Metryki i modele w inżynierii jakości oprogramowania*, PWN SA, Warszawa, 2006
11. Kerievsky J.: *Refaktoryzacja do wzorców projektowych*, Helion, Gliwice, 2005
12. Khaer Md. A. i inni: *An Empirical Analysis of Software Systems for Measurement of Design Quality Level Based on Design Patterns*, Computer and Information Technology, IEEE, 2007
13. Kirasić D., Basch D.: *Ontology-Based Design Pattern Recognition*, Knowledge-Based Intelligent Information and Engineering Systems, Zagreb, Croatia, 2008
14. Kornatka A.: *Projekt i konstrukcja systemu generującego elementy bazodanowych aplikacji biznesowych*, Studia Informatica Vol.: 33, No. 2B, s. 201-215, Wydawnictwo Politechniki Śląskiej, Gliwice, 2012
15. Martin R., Martin M: *Agile. Programowanie zwinne: zasady, wzorce i praktyki zwinnego wytwarzania oprogramowania w C#*, Helion, Gliwice, 2008
16. McConnell S.: *Kod Doskonały*, Helion, Gliwice, 2010
17. Rasool G.: *Customizable Feature based Design Pattern Recognition Integrating Multiple Techniques*, Dysertacja Doktorska, Technische Universitat Ilmenau, Ilmenau, 2010
18. Singh Rao R., Gupta M.: *Design Pattern Detection by Greedy Algorithm Using Inexact Graph Matching*, International Journal Of Engineering And Computer Science, Vol. 2, Issue 10, s. 3658-3664, 2013
19. Troelsen A.: *Język C# i platforma .NET 3.5*, PWN, Warszawa, 2009
20. Tsantalis N. i inni: *Design Pattern Detection Using Similarity Scoring.* IEEE Transactions on Software Engineering, Volume: 32, Issue: 11, s. 896-908, 2006
21. Wojszczyk R.: *Koncepcja hybrydowej metody do oceny jakości zaimplementowanych wzorców projektowych*, w: Zeszyty Naukowe Wydziału Elektroniki i Informatyki nr 7, s. 17-26, Wydawnictwo Uczelniane Politechniki Koszalińskiej, Koszalin, 2015
22. Wojszczyk R.: *Porównanie sposobów reprezentacji wzorców projektowych*, w: Modele inżynierii teleinformatyki 9, s. 133 - 145, Wydawnictwo Uczelniane Politechniki Koszalińskiej, Koszalin, 2014
23. Wojszczyk R.: *Pozyskiwanie struktury obiektowej z kodu zarządzanego przy wykorzystaniu metod inżynierii odwrotnej*, w: Inżynieria oprogramowania: badania i praktyka, s. 199-213, Zeszyty Rady Naukowej Polskiego Towarzystwa Informatycznego, Warszawa, 2014
24. Wojszczyk R.: *The model and function of quality assessment of implementation of design patterns*. Applied Computer Science, Vol. 11, No. 3, s 45-56, Institute of Technological Systems of Information, Lublin University of Technology, Lublin, 2015

Streszczenie

Wzorce projektowe to zagadnienie szeroko opisywane w uznanej literaturze i wykorzystywane przez wielu programistów, ale mimo to nie ma nad nimi formalnej kontroli. W artykule poruszony został problem weryfikacji implementacji wzorców projektowych stosowanych w programowaniu obiektowym. W procesie weryfikacji wyróżniono dwa modele danych: formalną reprezentację będącą ekwiwalentem badanego oprogramowania oraz repozytorium implementacji wzorców zawierające informacje opisujące implementację wzorców projektowych. Opracowane rozwiązanie pozwoli wykazać błędy i potencjalne problemy w implementacji.

Abstract

Although the design patterns constitute the issue that has been widely discussed in the literature and used by many software developers, there is no formal control over them. The article discussed the problem of verifying the implementation of design patterns applied in object-oriented programming. Two following data models were distinguished in the process of verification: a formal representation that is an equivalent of the analysed software, and a repository of implementation of patterns containing information describing the implementation of design patterns. The proposed solution will make it possible to show implementation errors and potential problems.

Keywords: design patterns, data model, ERD, verifying implementation

Sebastian Jarosiński
Łukasz Sobaszek
Łukasz Wojciechowski
Magdalena Gregorczyk
Instytut Technologicznych Systemów Informacyjnych
Wydział Mechaniczny, Politechnika Lubelska
20-618 Lublin, ul. Nadbystrzycka 36
sebastian.jarosinski2912@gmail.com
l.sobaszek@pollub.pl
l.wojciechowski@pollub.pl
m.gregorczyk@pollub.pl

Ocena wybranych technik TCT w odwzorowywaniu rzeczywistych obiektów

1. Wstęp

Techniki przyspieszające procesy prototypowania oraz wytwarzania są obiektem dużego zainteresowania zarówno we współczesnym przemyśle, jak i nauce. Prowadzonych jest wiele badań nad zagadnieniem zastosowania technologii TCT (*Time Compressing Technologies*) w zakresie skrócenia czasu projektowania i wytwarzania gotowych elementów [3, 4, 10].

Celem niniejszej pracy jest przedstawienie oraz ocena wybranych technik skracania czasu wytwarzania. Zakres pracy obejmuje przedstawienie ogólnego podziału technik TCT oraz ich krótkiej charakterystyki, a także analizę możliwości technik Rapid Prototyping oraz Rapid Manufacturing w procesie odwzorowywania rzeczywistych obiektów. Ponadto dokonano porównania parametrów modeli wytworzonych za pomocą technik TCT i parametrów elementów rzeczywistych.

2. Przegląd technik TCT

Techniki TCT (ang. *Time Compressing Technologies*) służą skróceniu czasu projektowania oraz wytwarzania elementów dzięki wykorzystaniu najnowszych technologii, takich jak druk 3D oraz komputerowo wspomagane projektowanie [13]. Wśród technik tych wyróżnia się [12]:

- Virtual Prototyping – wirtualne prototypowanie,
- Rapid Prototyping – szybkie prototypowanie,

- Rapid Manufacturing – szybkie wytwarzanie,
- Rapid Tooling – szybkie wykonywanie narzędzi,
- Reverse Engineering – inżynieria wsteczna (odwrotną).

Wirtualne prototypowanie to proces polegający na tworzeniu oraz badaniu wirtualnego prototypu elementu. Proces ten obejmuje komputerowe zaprojektowanie obiektu, symulację procesu wytwarzania (np. odlewania czy obróbki mechanicznej), a następnie badania symulacyjne jego własności (wytrzymałościowych, funkcjonalnych, ergonomicznych, itp.). Badania modelu komputerowego mogą obejmować sprawdzanie wielu wariantów rozwiązań, analizę wykonalności, badanie możliwości montażu i wykrywanie kolizji, badania wytrzymałościowe (zarówno statyczne jak i dynamiczne), wyznaczanie przepływów ciepła i rozkładów temperatury, a także kinematyczną i dynamiczną symulację pracy. Do wirtualnego prototypowania wykorzystuje się oprogramowanie, które pozwala zbudować realistyczny model prototypu i prowadzić na nim badania dotyczące kinematyki ruchu, a także dynamiki z uwzględnieniem: mas, sił, tarcia, tłumienia, odkształceń, naprężeń, drgań, itd. Pozwala to optymalizować projekt przed rzeczywistym wykonaniem prototypu [11, 12].

Przykładowe środowiska do prowadzenia badań w zakresie Virtual Prototyping, to [12]:

- NASTRAN – symulacje obiektów trójwymiarowych,
- ADAMS – analiza kinematyki i dynamiki,
- MATLAB i SIMULINK – modelowanie i symulacja układów dynamicznych (w tym układów sterowania),
- FIDAP – analiza przepływów.

Rapid Prototyping, czyli szybkie wytwarzanie prototypów, to proces automatycznego wytwarzania elementów maszyn lub innych przedmiotów za pomocą urządzeń sterowanych z komputera na podstawie opracowanego wcześniej modelu bryłowego [12]. W odróżnieniu od metod ubytkowych, stosowanych podczas obróbki na klasycznych obrabiarkach, metody RP są addatywne. Oznacza to, iż polegają na stopniowym dodawaniu kolejnych warstw materiału przez: klejenie, stapianie, spiekanie czy utwardzania różnych materiałów za pomocą różnorodnych wiązek promieniowania. Urządzenia i technologie stosowane do realizacji szybkiego prototypowania określa się mianem systemów szybkiego wytwarzania prototypów (ang. RPS – Rapid Prototyping Systems) [1]. Pierwsze systemy szybkiego prototypowania powstały w latach 80. Początkowo stosowane były tylko do produkcji prototypów. Obecnie znajdują coraz szersze zastosowanie, także do produkcji narzędzi lub krótkich serii wysokiej jakości elementów. Szybkość wytwarzania obejmuje zazwyczaj okres od kilku do kilkudziesięciu godzin. Czas ten zależy głównie od metody i zastosowanego sprzętu oraz złożoności

modelu. Zastosowanie znajdują różne materiały – od papieru, po materiały ceramiczne, aż do tworzyw polimerowych [9, 12].

W procesie RP wykorzystywane są różnorodne metody. Jedną z pierwszych była stereo-litografia, która polega na warstwowym utwardzaniu żywic epoksydowych lub akrylowych pod wpływem promieniowania ultrafioletowego (jego źródłem jest laser o dużej mocy). Wytwarzany obiekt umieszczony jest na platformie zanurzanej w wannie wypełnionej płynnym fotopolimerem. Stopniowe obniżanie wanny po utwardzeniu każdej kolejnej warstwy umożliwia budowanie kolejnych warstw modelu. Jest ona uznawana za technikę charakteryzującą się największą dokładnością odwzorowania [2, 12].

Kolejną z metod Szybkiego Prototypowania jest nakładanie stopionego materiału (ang. FDM – Fused Deposition Modeling). Polega ona na wtapianiu w model kolejnych porcji materiału termoplastycznego. Ma on zazwyczaj postać cienkiej nitki wykonanej z takiego materiału jak ABS, PC, itp. W metodzie tej materiał modelowy jest wykorzystywany naprzemiennie z materiałem podporowy, który jest następnie usuwany [5]. Zaletami produktów wytwarzanych za pomocą FDM są m. in. duża wytrzymałość elementów, niski koszt ich wytwarzania, możliwość dalszej obróbki mechanicznej, a także szczelność i odporność na działanie wody.

Inną z metod jest Selektywne Spiekanie Laserowe (ang. SLS – Selective Laser Sintering), gdzie zastosowanie znajdują lasery dużej mocy. Wykorzystywane są one do warstwowego spiekania małych cząstek tworzyw polimerowych, metalu, ceramiki lub szkła. Pomimo większej złożoności procesu w porównaniu do pozostałych metod, jego zaletą jest większy zakres dostępnych materiałów [7].

Metoda stapiania metali wiązką elektronową w próżni (ang. EBM – Electron Beam Melting) jest metodą podobną technologią do metody SLS. Prototypy wykonane tą metodą charakteryzują się dużą trwałością. Jest to spowodowane charakterystyką procesu wytwarzania – polega on na warstwowym stapianiu cienkich warstw metalu w komorze próżniowej za pomocą wiązki elektronowej. Temperatura w trakcie procesu osiąga od 700 do 1000°C. Produkowane mogą być w ten sposób m.in. implanty, wykonane ze stopów tytanowych [12].

Sklejanie modelu z wycinanych laserowo warstwo papieru (ang. LOM – Laminated Object Manufacturing) to kolejna z metod RP. Polega ona na wycinaniu laserem przez maszynę poszczególnych warstw z podawanego z rolki samoprzylepnego papieru i naklejaniu ich na siebie. Opracowane modele mogą znaleźć zastosowanie w procesie tworzenia form odlewniczych – wówczas wykonać można metalowy odlew prototypu. Elementy wytworzone za pomocą metody LOM są łatwo obrabialne, a rozmiary prototypu są niemal nieograniczone – istnieje bowiem możliwość budowy prototypu w częściach i ich dokładnego sklejenia [12].

Rapid Manufacturing jest to zastosowanie powyższych metod RP do wytwarzania serii elementów z danego materiału, z których każdy z nich jest indywidualnie kształtowany [1].

Szybkie tworzenie form i narzędzi, nosi miano **Rapid Tooling**. Na podstawie modeli wykonanych za pomocą metod Rapid Prototyping wytwarza się formy do powielania kształtu wytworzonego modelu, np. poprzez wykonanie form silikonowych do odlewania próżniowego, napylanie skorupy metalowej bądź budowę formy dla żywicy epoksydowej. Służą one do wytwarzania serii wyrobów z tworzyw sztucznych, ale możliwe jest także zastosowanie prototypów z materiału podobnego do wosku jako wzorców w odlewaniu metodą traconego rdzenia [9, 12].

Tworzenie modelu komputerowego na podstawie istniejącego przedmiotu z zastosowaniem skanerów 3D to tak zwana **Inżynieria Odwrotna** (ang. RE – Reverse Engineering). Celem zastosowania tej metody jest otrzymanie modelu wirtualnego CAD, który następnie może być użyty do wytwarzania technologią szybkiego prototypowania. Metoda ta jest wykorzystywana m. in. podczas prac nad archiwizacją lub rekonstrukcją przestrzennych obiektów takich jak rzeźby, bądź w medycynie do uzyskiwania modeli części ciała pacjenta [8, 9].

3. Metodyka procesu odwzorowywania

Proces odwzorowywania obiektów rzeczywistych odbywa się w kilku etapach: najpierw wybrane elementy są skanowane za pomocą skanerów 3D, a wynik skanowania zostaje przesłany do komputera i przetworzony za pomocą odpowiedniego oprogramowania. Z powstałego w ten sposób modelu CAD tworzony jest plik STL, który jest zbiorem danych dla urządzenia wytwarzającego, czyli drukarki 3D.

Do przeprowadzonych analiz zostały wybrane dwa elementy, które charakteryzował odmienny kształt i właściwości ich powierzchni. Pierwszym z nich był totem do gry, wykonany z drewna, który ma kształt walca o zmiennej średnicy (rys. 1). Drugim obiektem był tłok samochodowy, wykonany ze stopu aluminium, o powierzchni noszącej delikatne ślady zużycia (rys. 2). Ponieważ drugi z obiektów charakteryzował się dość złożoną budową – procesowi odwzorowywania poddano wyłącznie zewnętrzną część elementu.

Rys. 1. Obiekt rzeczywisty – totem

Rys. 2. Obiekt rzeczywisty – tłok

Do procesu skanowania 3D zastosowano urządzenie firmy ARTEC – model Spider, który jest ręcznym skanerem do skanowania małych obiektów w bliskiej odległości. Skaner ten używa światła niebieskiego, generowanego za pomocą lampy LED. Urządzenie odczytuje kształt skanowanych elementów z dokładnością do 0,05 mm. Szczegółowe parametry skanera przedstawiono w tabeli 1 [6].

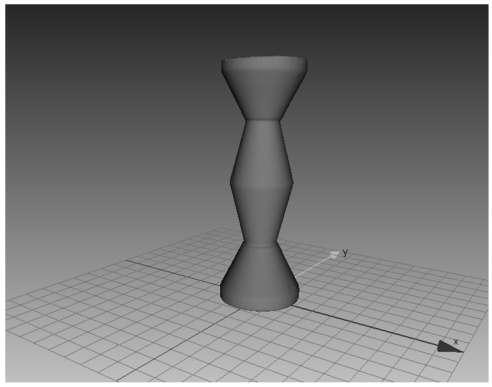
Tab. 1. Parametry techniczne skanera ARTEC Spider [6]

Parametr	**Wartość**
Zapis tekstury	Tak
Rozdzielczość 3D, do	0,1 mm
Dokładność punktu 3D, do	0,05 mm
Dokładność 3D na odległość, do	0,03% na 100 cm
Rozdzielczość tekstury	1.3 MP
Kolory	24 BPP
Źródło światła	Lampa LED – światło niebieskie
Liniowe pole widzenia (najmniejszy zakres)	90 mm x 70 mm
Liniowe pole widzenia (największy zakres)	180 mm x 140 mm
Kątowe pole widzenia	30 x 21°

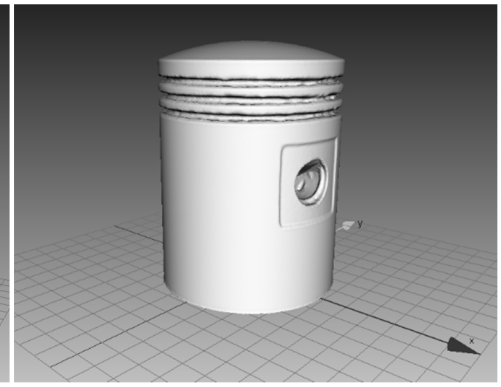
Odległość pracy	0,17–0,3 m
Klatek video, do	7,5 fps
Czas ekspozycji	0,0005 s
Prędkość akwizycji danych, do	1 000 000 punktów/s
Formaty wyjściowe	OBJ, STL, PLY, WRML, PTX, ASCII, XYZRGB, AOP, E57,CSV, XML, DXF
Zdolność przetwarzania	40 000 000 trójkątów/1GB RAM
Wymiary	190 mm x 130 mm x 140 mm
Waga	0,85 kg
Zużycie energii	12V, 24W
Interfejs	1 x USB 2.0

Podczas skanowania obiektów zastosowano statyw fotograficzny oraz stolik obrotowy własnego projektu w celu zmniejszenia błędów spowodowanych drżeniem ręki operatora oraz nie osiowym obracaniem skanowanego przedmiotu. Podczas procesu odwzorowywania tłoka jego powierzchnia została pokryta warstwą talku tak, aby zmniejszyć odbijanie promieni świetlnych przez oszlifowaną powierzchnię. Dzięki temu zredukowano błędy wynikające z nieprawidłowego odczytu sygnałów świetlnych przez skaner.

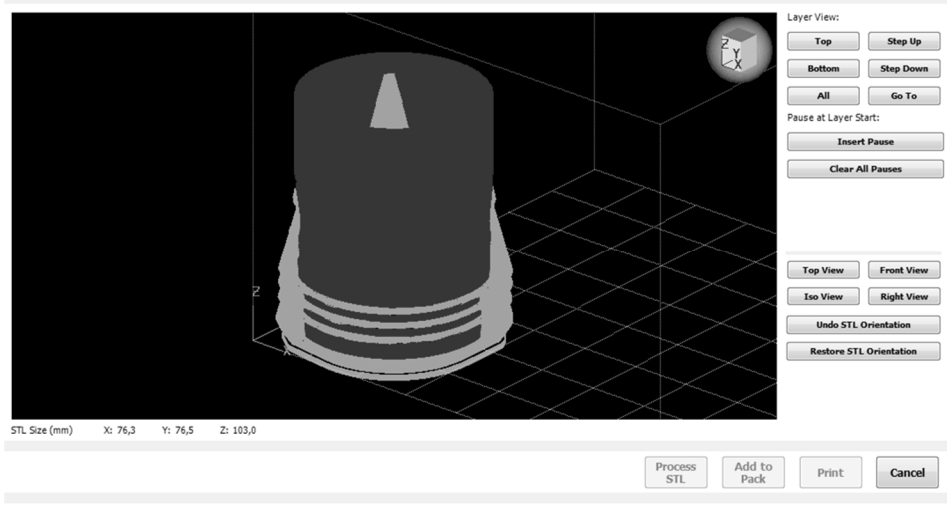
Po zeskanowaniu obydwu obiektów otrzymane modele zostały odpowiednio przetworzone w środowisku ARTEC Studio 10, które jest oprogramowaniem dedykowanym przez dystrybutora do tego typu skanerów 3D. Obróbka polegała na usunięciu punktów błędnych, uzupełnieniu ubytków w modelu, a także usunięciu części zeskanowanego podłoża. Celem wykonanych prac było jak najszybsze odwzorowanie obiektów stąd też zastosowanie znalazły podstawowe algorytmu obróbki modeli (rys. 3 i 4). Zastosowanie znalazły m.in. algorytmy: dokładnej rejestracji, szybkiego łączenia, wygładzania, scalania oraz usuwania dziur. Parametry algorytmów nie były modyfikowane. Dla każdego z obiektów posiadały takie same wartości. W modelu totemu wykorzystano dodatkowo możliwości edycji modelu poprzez skopiowanie zeskanowanej płaszczyzny podstawy obiektu i umieszczenie jej w miejscy drugiej z płaszczyzn, która w trakcie procesu nie została zeskanowana. Następnie uzyskane modele CAD przetworzono na pliki STL, które są roboczym formatem wykorzystanej drukarki 3D (rys. 5).



Rys. 3. Wirtualny model skanowanego totemu



Rys. 4. Wirtualny model skanowanego tłoka



Rys. 5. Przetwarzanie pliku STL w celu wyznaczenia ścieżek przejścia głowicy drukarki 3D

Do wytworzenia odwzorowanego modelu wykorzystano drukarkę 3D marki uPrint SE Plus, która wykorzystuje tworzywo polimerowe ABS, zaś materiałem pomocniczym jest rozpuszczalny SR-30. Głowica drukarki pozwala na wytwarzanie elementów za pomocą dwóch grubości nici tworzywa: 0,254 mm i 0,330 mm, co ma wpływ na dokładność odwzorowania. Do wydruku modelu totemu wykorzystano większą dokładność, zaś przy modelu tłoka postawiono na szybkość wydruku – kosztem jego dokładności. Czas wydruku wyniósł odpowiednio – 4 godziny dla

obiektu totem oraz 8 godzin dla tłoka samochodowego. W tabeli 2 znajdują się parametry użytej drukarki 3D [5].

Tab. 2. Parametry techniczne drukarki 3D uPrint SE Plus [5].

Parametr	Wartość
Materiał	ABS+, kolor: kość słoniowa, biały, niebieski, fluorescencyjny żółty, niebieski, czarny, czerwony, pomarańczowy, oliwkowy, szary
Materiał suportu	SR-30 rozpuszczalny
Rozmiar wydruków	200 x 200 x 150 mm
Grubość warstwy	0,254 mm 0,330 mm

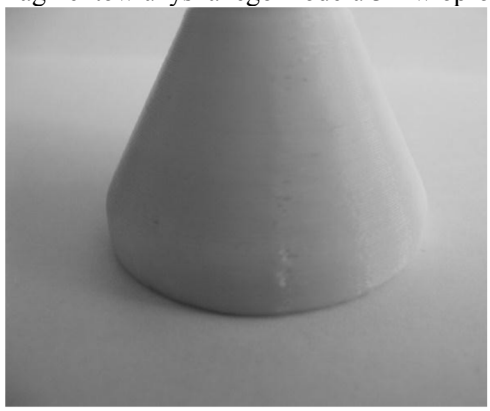
4. Rezultaty badań

Po otrzymaniu wydruków 3D odwzorowujących zeskanowane obiekty poddano je ocenie wizualnej oraz pomiarom, mającym na celu określenie dokładności odwzorowania. Na ilustracjach poniżej (rys. 6 i 7) przedstawiono fotografie uzyskanych wydruków.

Rys. 6. Wydruk 3D modelu totemu

Rys. 7. Wydruk 3D modelu tłoka

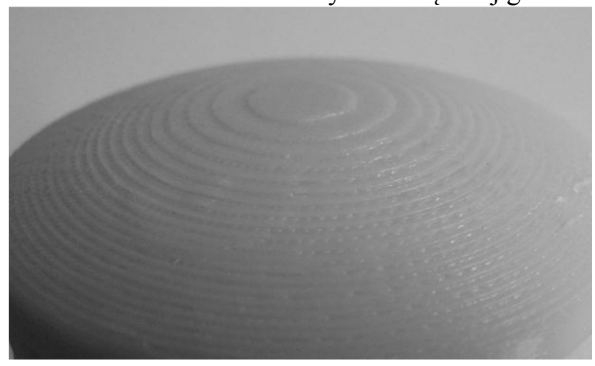
Dzięki analizie otrzymanych wydruków stwierdzono istnienie pewnych niedokładności wykonania modeli. Na powierzchni totemu pojawiło się zgrubienie równoległe do osi modelu, zaś na jednej z podstaw zauważono nierówność powierzchni w postaci charakterystycznych „schodków" (rys. 8 i 9). Pierwsze z niedoskonałości powstało prawdopodobnie w wyniku zużycia się narzędzia – głowicy drukarki 3D. Nierówność powierzchni jest natomiast rezultatem łączenia fragmentów uzyskanego modelu 3D w oprogramowaniu Artec.



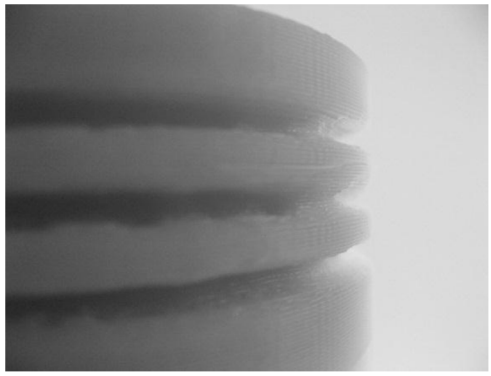
Rys. 8. Zgrubienie na powierzchni bocznej

Rys. 9. Nierówność powierzchni podstawy

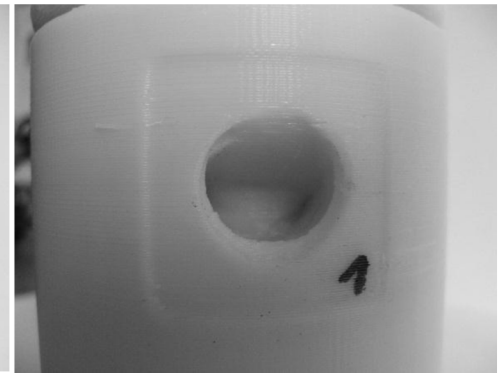
Na wydrukowanym modelu tłoka również zaobserwowano niedokładności odwzorowania geometrii obiektu rzeczywistego. Widoczne są niedokładnie odwzorowane otwory na powierzchni bocznej tłoka, nierówność powierzchni i krawędzi rowków na pierścienie uszczelniające, a także niejednolitość powierzchni czołowej elementu (rys. 10, 11 i 12). Błędy odwzorowania są spowodowane głównie przez odbijającą promienie świetlne gładką powierzchnię przedmiotu skanowanego, jego złożoną budowę oraz zastosowanie warstw tworzywa o większej grubości.



Rys. 10. Nierówność powierzchni czołowej



Rys. 11. Złe odwzorowanie rowków



Rys. 12. Niedokładność otworów

Oprócz wizualnej oceny uzyskanych modeli przeprowadzono także pomiary sprawdzające dokładność odwzorowania. W celu zmierzenia wybranych wartości dla obiektów rzeczywistych jak i wydruków 3D zastosowano suwmiarkę o dokładności 0,02 mm. Pomiarów modeli wirtualnych dokonano za pomocą narzędzia dostępnego w oprogramowaniu Artec Studio 10.

Dla totemu zmierzono całkowitą długość przedmiotu oraz średnice w pięciu charakterystycznych punktach (zgrubieniach i przewężeniach). Rezultaty przeprowadzonych pomiarów przedstawiono w tabeli 3.

Tab. 3. Rezultaty pomiarów wybranych wielkości – obiekt totem

	Obiekt rzeczywisty	**Model wirtualny**	**Wydruk 3D**
Długość [mm]	122,36	121,12	121,68
Średnica 1 [mm]	41,68	41,74	41,58
Średnica 2 [mm]	17,00	17,41	17,00
Średnica 3[mm]	32,26	32,25	32,06
Średnica 4 [mm]	16,60	17,31	17,26
Średnica 5 [mm]	41,88	41,69	41,96

Na podstawie uzyskanych wyników pomiarów ocenić można, że różnica pomiędzy obiektem rzeczywistym a modelem wirtualnym wynosi maksymalnie 1,24 mm. Różnica ta spowodowana jest niedokładnością pomiarów przeprowadzonych za pomocą oprogramowania dołączonego do skanera – punkty pomiarowe wskazywane są przez użytkownika i nie zawsze są one współosiowe. Natomiast

maksymalna różnica pomiędzy modelem uzyskanym metodą druku 3D a obiektem rzeczywistym wynosi zaledwie 0,68 mm.

Podobne pomiary zostały przeprowadzone dla obiektu rzeczywistego typu tłok, gdzie zmierzono długość powierzchni bocznej elementu (wysokość ścianki), średnice w trzech charakterystycznych przekrojach, a także średnice otworów na powierzchni bocznej tłoka (tabela 4).

Tab. 4. Pomiary wybranych wielkości – obiekt tłok

	Obiekt rzeczywisty	**Model wirtualny**	**Wydruk 3D**
Długość [mm]	95,20	96,35	95,70
Średnica 1 [mm]	75,90	75,84	75,78
Średnica 2 [mm]	75,84	75,82	75,82
Średnica 3[mm]	75,78	75,77	75,72
Średnica otworu 1 [mm]	17,90	17,99	17,64
Średnica otworu 2 [mm]	17,94	17,81	17,70

Należy zauważyć, że największa różnica pomiędzy modelem wirtualny a obiektem rzeczywistym wynosi 1,15 mm. Jestem ona spowodowana (podobnie jak w przypadku totemu) niedokładnością podczas pomiarów liniowych wymiarów za pomocą oprogramowania. Różnica pomiędzy drukiem 3D a elementem bazowym wynosi zaledwie 0,50 mm.

Dokonując analizy przeprowadzonych pomiarów należy pamiętać, iż główne różnice wynikały z problemowym wykorzystaniem komputerowego narzędzia do pomiarów, użytego na modelu wirtualnym. Pod uwagę należy także wziąć ewentualny błąd popełniony przez operatora podczas pomiaru obiektu rzeczywistego, a także skurcz materiału z którego został wykonany wydruk 3D.

Niemiej jednak należy stwierdzić, iż uzyskane różnice pomiędzy obiektami są niewielkie, przez co pod kątem odwzorowania wymiarów elementów proces ten należy uznać za zadowalający. Dyskusyjne pozostaje odwzorowanie szczegółów elementów (podstawy totemu, otworów oraz rowków na pierścienie tłoka). Dokładność ich wykonania można jednak poprawić poddając obróbce graficznej uzyskane modele wirtualne.

5. Podsumowanie i wnioski

Wykorzystane techniki niewątpliwie skracają czas procesów prototypowania oraz wytwarzania. Modele wytworzone za pomocą drukarki 3D powstały w czasie o wiele krótszym, niż za pomocą konwencjonalnych metod oraz nie wymagały obróbki wykończeniowej. Wykorzystanie skanowania oraz druku 3D pozwala uzyskiwać obiekty o właściwościach zbliżonych do obiektów pierwotnych. Po przeprowadzonych badaniach wywnioskować można, że obiekty rzeczywiste zostały odzwierciedlone z dużą dokładnością wymiarów i kształtu. Dokładne odwzorowanie wymaga wprawy w procesie skanowania oraz zastosowania dodatkowego oprogramowania. Zastosowanie statywu oraz stolika obrotowego pomogło w uzyskaniu modeli lepszej jakości. Rezultaty wykorzystanych technik mogą być pomocne w innych procesach, np. wyznaczaniu ścieżek obróbczych maszyn NC, projektowaniu form odlewniczych, itd. Poprzez zastosowanie tych technik możliwe jest zarówno zmniejszenie kosztów, jak i czasu potrzebnych na zrealizowanie procesu prototypowania. Niewątpliwie, techniki TCT to techniki przyszłości.

Literatura

1. Chua C.K., Leong, K.F.: Rapid Prototyping: Principles and Applications in Manufacturing. World Scientific, 2000.
2. Dziubek T., Filip M.: Analiza i porównanie dokładności wybranych przyrostowych metod wytwarzania. Mechanik Nr12/2015, s. 54-61, Stowarzyszenie Inżynierów i Techników Mechaników Polskich, 2015 r.
3. Fernanda C. Godoi, Sangeeta Prakash, Bhesh R. Bhandari: 3D Printing Technologies Applied for Food Design: Status And Prospects. Journal of Food Engineering, Vol. 179, 2016, pp. 44-54.
4. Gosselin C., Duballet R., Roux Ph., Gaudillière N., Dirrenberger J., Morel Ph.: Large-scale 3D printing of ultra-high performance concrete – a new processing route for architects and builders, Materials & Design, Volume 100, 15 June 2016, pp. 102-109.
5. Instrukcja obsługi drukarki 3D uPrint SE Plus (dołączona do zestawu).
6. Instrukcja obsługi skanera 3D ARTEC 3D Spider (dołączona do zestawu).
7. Mager A., Moryson G., Cellary A., Marciniak L.: Zastosowanie technik Rapid Prototyping do wytwarzania wyrobów metalowych. Postępy Nauki i Techniki, nr 8, s. 174-182, Oddział SIMP w Lublinie, 2011r.
8. Markowska O., Budzik G.: Innowacyjne metody wytwarzania implantów kostnych za pomocą inżynierii odwrotnej (RE) oraz technik szybkiego prototypowania (RP). Mechanik R. 85, nr 2CD, Stowarzyszenie Inżynierów i Techników Mechaników Polskich, 2012r.

9. Miecielica M.: Techniki szybkiego prototypowania – Rapid Prototyping. Przegląd Mechaniczny, nr 2, s. 39-45, Instytut Mechanizacji Budownictwa i Górnictwa Skalnego, 2010r.
10. Nitish Kumar, Hemant Kumar, Jagdeep Singh Khurmi: Experimental Investigation of Process Parameters for Rapid Prototyping Technique (Selective Laser Sintering) to Enhance the Part Quality of Prototype by Taguchi Method. Procedia Technology, Vol. 23, 2016, pp. 352-360.
11. Pater Z., Tofil A.: Wirtualne prototypowanie w budowie maszyn. http://kis.pwszchelm.pl/publikacje/III/pater.pdf [data dostępu: 24-05-2016]
12. Rudnicki Z.: Nowoczesne techniki przyspieszające wytwarzanie. http://www.kkiem.agh.edu.pl/dydakt/wyklady/rapidprot12.pdf [data dostępu: 24-05-2016]
13. Ruszaj A., Gawlik J., Skorczypiec S.: Stan badań i kierunki rozwoju wybranych niekonwencjonalnych procesów wytwarzania. Inżynieria maszyn, R. 14, z. 1, s.7-19, Wydawnictwo Wrocławskiej Rady FSNT NOT, 2009 r.

Abstrakt

Techniki przyspieszające procesy prototypowania oraz wytwarzania we współczesnym przemyśle oraz nauce są obiektem dużego zainteresowania. Zwiększająca się dostępność nowoczesnych technik skracania czasu sprawia, iż coraz częściej prowadzi się szereg badań mających na celu analizę efektywności zastosowania technologii TCT (Time-Compressing Technologies) w różnorodnych procesach.

Celem niniejszej pracy jest przedstawienie oraz ocena wybranych technik przyspieszających wytwarzanie. Zakres pracy obejmuje przedstawienie ogólnego podziału technik TCT oraz ich krótką charakterystykę, omówienie technik Rapid Prototyping oraz Rapid Manufacturing, a także ocenę wykorzystania skanera 3D oraz druku 3D w procesie odwzorowywania rzeczywistych elementów. W pracy przedstawiono rezultaty badań mające na celu porównanie parametrów elementu wytworzonego za pomocą technik TCT z parametrami obiektu rzeczywistego.

Słowa kluczowe: Rapid Prototyping, Rapid Manufacturing, analiza modeli 3D

Abstract

Rapid Prototyping and Rapid Manufacturing techniques are the subject of great interest in current industry and science. Increasing amount of modern Time-Compressing Technologies causes needs of conduct a lot of research in TCT efficiency analysis area in various processes.

The article presents discussing and evaluation of selected Time-Compressing Technologies. First of all, the TCT – Rapid Prototyping and Rapid Manufacturing was outlined. Moreover, the evaluation of using 3D scanner and 3D printer in real objects modelling are presented. In the final part of the paper the authors presented and discussed parameters comparison of real object and model prepared by means of TCT.

Keywords: Rapid Prototyping, Rapid Manufacturing, 3D models analysis

Łukasz Cieszyński
Wydział Informatyki
Zachodniopomorski Uniwersytet Technologiczny
lcieszynski@zut.edu.pl

Generowanie stymulacji świetlnych za pomocą diody LED na potrzeby interfejsu mózg-komputer

Słowa kluczowe: aparatura stymulująca, SSVEP, interfejs mózg komputer, LED

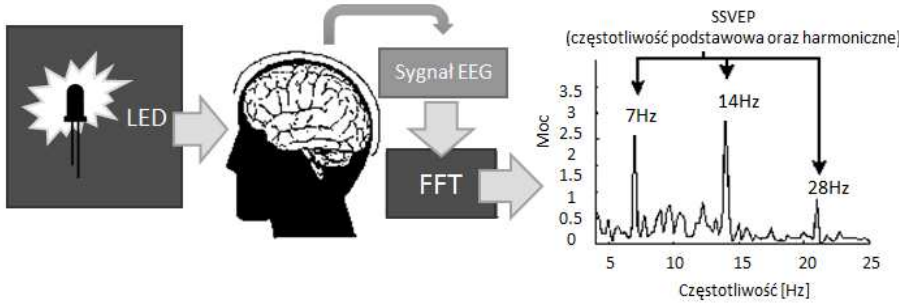
Wstęp

Interfejs mózg-komputer BCI (Brain Computer Interface) stanowi układ do sterowania/komunikowania się człowieka z maszyną. Założeniem takiego systemu jest przekazywanie poleceń do urządzenia bezpośrednio za pomocą fal mózgowych. Ujmując to inaczej, interfejs ten jest w stanie zapewnić komunikację pomiędzy mózgiem użytkownika a komputerem, umożliwiając przekazywanie poleceń /instrukcji do maszyny bez pośrednictwa układu mięśniowego [1]. W ostatnich latach widać zintensyfikowanie prac na rozwojem tego typu rozwiązań. Należy zaznaczyć, że powstało wiele ich odmian opartych na różnych procesach zachodzących w mózgu człowieka. Jedną z nich stanowi BCI oparte na zjawisku występowania i detekcji wzrokowych potencjałów wywołanych stanu ustalonego SSVEP (Steady State Visually Evoked Potentials).

SSVEP można zdefiniować jako typ potencjałów wywołanych, generowanych przez centralny układ nerwowy w odpowiedzi na powtarzający się bodziec wzrokowy (np. migającą z daną częstotliwością diodę LED) [2]. W konsekwencji zainicjowania takiej stymulacji w sygnale EEG (Elektroencefalogram) rejestrowanym znad kory wzrokowej można zaobserwować wyraźny wzrost mocy w paśmie częstotliwości odpowiadającym częstotliwości bodźca stymulującego [3].

Wyobraźmy sobie następujący eksperyment (Rys. 1). Podmiot badany podłączony jest do aparatury EEG i obserwuje migającą z częstotliwością 14 Hz diodę LED (stymulator). Rejestrowany sygnał znad kory wzrokowej zostaje następnie oczyszczony z szumów i artefaktów (filtracja) oraz przetworzony (np. metodą FFT – Fast Fourier Transform). Tak przygotowany sygnał przedstawiony w formie wykresu (zależności mocy sygnału od częstotliwości) ukazuje występowanie maksymalnej amplitudy mocy sygnału dla częstotliwości zgodnej z częstotliwością stymulatora – w naszym przypadku będzie to 14 Hz. Ponadto, co

jest specyficzne dla SSVEP, możliwe jest również zaobserwowanie wysokiego poziomu mocy nie tylko dla częstotliwości podstawowej (tutaj 14 Hz), ale również przy częstotliwościach harmonicznych (28 Hz) i subharmonicznych (7 Hz) [4-5] .



Rys. 1. Schemat powstawania SSVEP

Dysponując układem stymulującym (np. migającą z daną częstotliwością diodą LED lub ekranem LCD) w połączeniu z aparaturą do pomiaru EEG możliwe jest zrealizowanie interfejsów mózg komputer w oparciu o SSVEP.

Powyższy przykład jest najprostszym z możliwych, w którym układ stymulujący dostarcza tylko jednej częstotliwości. Niemniej jednak, w konstruowaniu interfejsów BCI, w praktyce wykorzystuje się klika/kilkanaście częstotliwości. Liczba dostępnych częstotliwości zależy od liczby poleceń, która ma zostać dostarczona użytkownikowi interfejsu. Wykorzystanie kilku diod, gdzie każda pulsuje z inną częstotliwością, pozwala na przygotowanie np. układu sterowania robotem lub klawiaturą.

Zaletą systemów korzystających z SSVEP jest to, iż oparte o potencjały wzrokowe układy działają poza percepcją użytkownika i są skuteczne dla większości osób. Natomiast zastosowanie diod LED daje znaczący zbiór częstotliwości możliwych do wykorzystania w sterowaniu za pomocą interfejsu. Przy uwzględnieniu standardowego użytecznego zakresu częstotliwościowego sygnału EEG (5-30 Hz), diody LED pozwalają uzyskać około 80 możliwych częstotliwości stymulacji.

Autor w niniejszym artykule w części pierwszej przedstawi ogólny opis budowy i zasady działania interfejsu BCI opartego na SSVEP. W kolejnej części zwróci uwagę na zasadność stosowania diod LED jako źródła generowania stymulacji w takich systemach i wskaże przykładowe realizacje. Na koniec podsumuje prezentowany temat.

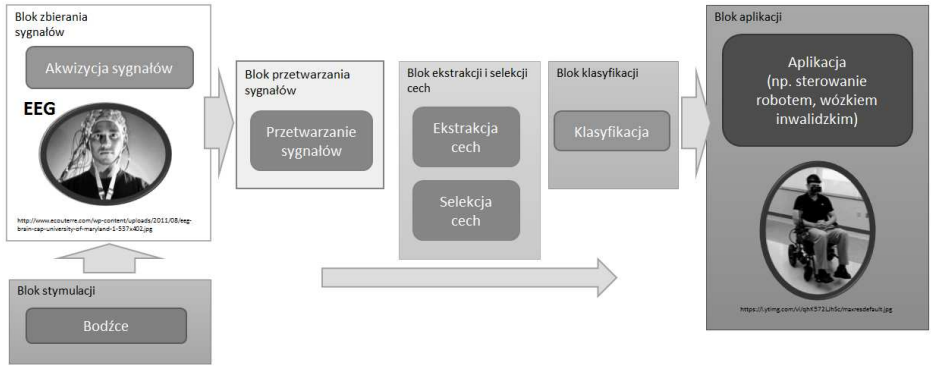
Opis zagadnienia

W budowie interfejsu mózg-komputer wykorzystującego potencjały SSVEP możemy wyróżnić kilka bloków (Rys. 2): blok stymulacji (bodźce); blok zbierania sygnałów (EEG, akwizycja sygnałów); blok przetwarzania sygnałów (przetwarzanie sygnałów); blok ekstrakcji i selekcji cech; blok klasyfikacji (klasyfikacja) oraz blok aplikacji (aplikacja).

Blok stymulacji stanowi źródło generowania bodźców. Stosuje się głównie dwie metody stymulacji pozwalające na obserwację SSVEP: generowanie stymulacji na ekranie monitora (CRT lub LCD) oraz generowanie impulsów świetlnych za pomocą diod LED. Jednakże, badania pokazują, iż odpowiedzi mózgu na sygnały generowane za pomocą LED są znacznie silniejsze niż na sygnały generowane na ekranie monitora [6]. W konsekwencji zastosowania jednego czy też drugiego rodzaju stymulacji możliwe jest – np. przy wykorzystaniu sprzętu do nieinwazyjnych badań EEG – zarejestrowanie, z powierzchni głowy badanego podmiotu, aktywności elektrycznej mózgu zawierającej SSVEP (blok zbierania sygnałów).

Kolejnym etapem jest odpowiednie przetworzenie zebranych sygnałów. Tutaj pierwszym krokiem jest zwykle użycie filtracji dolnoprzepustowej. W dalszej części następuje detekcja i usuwanie artefaktów, które mogą wystąpić jako następstwa takich czynników jak np. ruchów głową, szyją bądź oczami; aktywności serca; interferencji z siecią; szumów od otaczających urządzeń elektrycznych; ruchów kabli lub szumów termicznych. Blok wstępnego przetwarzania sygnału odgrywa istotną rolę dla prawidłowego działania całego BCI. Jego poprawne wykonanie warunkuje w znacznym stopniu kolejne etapy.

W dalszej części – na tak przygotowanym materiale – przeprowadza się proces ekstrakcji i selekcji cech. Pierwszy z nich skupia się na wydobyciu z sygnału cyfrowego, przy użyciu odpowiednich algorytmów np. filtracji przestrzennej, analizy widma, cech w zwarty sposób opisujących ten sygnał. Celem tych operacji jest głównie dokonanie próby jak najefektywniejszego opisu ilościowego właściwości sygnału EEG. Niemniej jednak, w zależności od użytej metodologii ekstrakcji cech liczba wyselekcjonowanych cech może się znacząco różnić. W celu ich redukcji, czy inaczej ujmując na potrzeby wyboru najbardziej istotnych cech przeprowadza się proces selekcji cech. Istotą selekcji jest pozostawienie cech istotnych, czyli takich, które wnoszą najwięcej informacji do procesu klasyfikacji, a usunięcie tych, które nie powodują znaczącego zwiększenia dokładności klasyfikatora. Do poprawnego działania systemu mózg-komputer niezbędna jest skuteczna metoda selekcji cech sygnału EEG [7].



Rys. 2. Schemat budowy interfejsu BCI

Przedostatnim modułem BCI jest klasyfikacja. Tutaj następuje przydzielenie wybranym cechom odpowiednich poleceń/instrukcji – odpowiednich klas. Wykorzystuje się klasyfikatory różnego rodzaju: jak klasyfikatory liniowe (LDA, liniowy SVM) oraz nieliniowe (sieci neuronowe, klasyfikatory Bayesa czy też klasyfikatory kNN) [8, 9]. Literatura nie wskazuje na jakiś konkretny klasyfikator, który dawałby najlepsze wyniki działania systemu BCI. Pamiętać trzeba jedynie, iż większość klasyfikatorów wymaga ręcznej kalibracji w celu dopasowania parametrów do użytkownika. Poprawne przeprowadzenie procesu klasyfikacji pozwala na wygenerowanie z systemu BCI instrukcji/poleceń wprost do aplikacji.

Blok ostatni – aplikacja – odpowiedzialny jest za wykonywanie poleceń. Pisząc wprost to ten element systemu pozwala na bezpośrednią komunikację z urządzeniem przekazując mu jedną z instrukcji wypracowanej w drodze klasyfikacji. Odnosząc to do przykładu BCI opartego na SSVEP pozwalającego na sterowanie wózkiem inwalidzkim, blok aplikacji przekazuje komendę np. skrętu w prawo w chwili gdy użytkownik obserwuje diodę LED, dla której przypisana jest ta instrukcja.

Należy zauważyć, że aby interfejs BCI działał prawidłowo, pomimo zastosowania się do powyższych zaleceń dla każdego z bloków, konieczne jest każdorazowe skalibrowanie interfejsu przed jego użyciem przez użytkownika. Kalibracja polega na dostrojeniu całego systemu do indywidualnego podmiotu np. poprzez zidentyfikowanie częstotliwości dających najwyższą synchronizację przy SSVEP. Pamiętać należy, że wykrywalność SSVEP jest sprawą indywidualną dla każdej osoby. Ponadto zidentyfikowane najlepsze częstotliwości stymulacji dla tego samego podmiotu mogą być różne w zależności np. od pory dnia, stanu psychofizycznego badanego oraz od minimalnie innego umiejscowienia elektrod na głowie użytkownika [10].

W trakcie długotrwałych sesji użytkownika z interfejsem może też być konieczne dodatkowe dopasowanie systemu do pojawiających się zmian

w aktywności mózgowej użytkownika. Po pomyślnie przeprowadzonej kalibracji użytkownik może przejść do trybu użytkowania i rozpocząć praktyczne korzystanie z systemu.

Przegląd literatury

Jednym z istotniejszych elementów BCI opartego o wywoływane potencjały wzrokowe jest blok stymulacji, czy też inaczej rzecz biorąc – aparatura stymulująca. Pod tym pojęciem należy rozumieć urządzenie (bądź zestaw urządzeń) generujących bodźce.

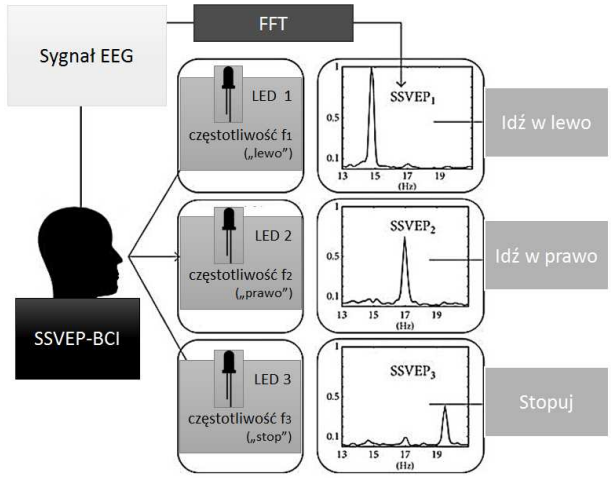
Rozpatrzmy przypadek BCI opartego na SSVEP, w którym bodźce są generowane przez migające ze stałą częstotliwością źródło światła. Jak zostało to już wspomniane i potwierdza to literatura [6], lepszymi stymulatorami są urządzenia mające jako źródło światła diody LED.

Chcąc wygenerować stymulację przy pomocy jednej diody LED w zakresie częstotliwości od 5 do 30 Hz możemy uzyskać około 80 różnych częstotliwości migania diody. Jest to znaczna liczba, która z powodzeniem może być wykorzystana do budowy interfejsu BCI.

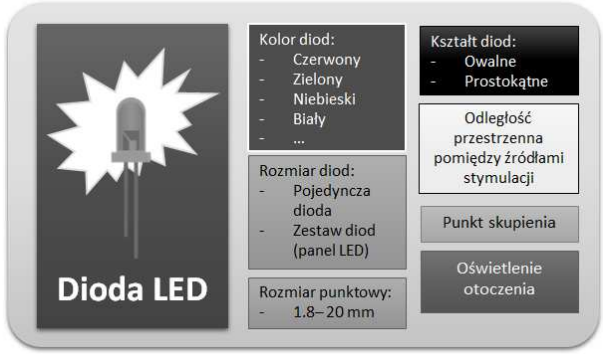
Rysunek 3 przedstawia schematyczną konstrukcję systemu do sterowania (np. robotem czy małym pojazdem zdalnie sterowanym) wykorzystującego diody LED. Użytkownik ma zaprezentowane 3 diody LED – każda mrugająca z inną częstotliwością (f_1, f_2, f_3). Dodatkowo każda z diod ma przypisane inne polecenie sterujące: „lewo", „prawo", „stop". W momencie obserwacji diody LED_1 migającej z częstotliwością f_1 w sygnale rejestrowanym znad kory wzrokowej (za pośrednictwem urządzenia do pomiaru EEG), po odpowiednim przetworzeniu (filtracja, FFT), zaobserwować możemy wyraźny wzrost mocy w paśmie częstotliwości odpowiadającym częstotliwości f_1 - czyli $SSVEP_1$. Detekcja $SSVEP_1$, dla omawianego interfejsu, tożsama jest z przekazaniem do sterowanego urządzenia komendy: „idź w lewo". Analogicznie dzieje się to w momencie obserwacji przez użytkownika innej diody.

Wykorzystane w powyższym przykładzie diody LED, jako generatory stymulacji świetlnych w układach BCI, stanowią najprostszą konstrukcję. Niemniej jednak przy tworzeniu takich aparatów do stymulacji konstruktor musi zwrócić uwagę na kilka parametrów (Rys. 4). Najbardziej istotnym z nich jest kolor. Diody LED dostępne są w kilku kolorach – najczęściej spotkać możemy diody czerwone, zielone, niebieskie i białe. Z łatwością możemy zmienić ten parametr zastępując źródło światła jednego koloru źródłem światła o innej barwie. Każda z nich w innym stopniu pobudza ośrodek wzrokowy, dając w odpowiedzi mózgu odmienną wielkość amplitudy mocy. W literaturze [11-14] czytamy, iż najwyższą amplitudą sygnału cechują się wyniki dla barwy czerwonej (Autorzy rozpatrywali jedynie kolory: czerwony, zielony, niebieski oraz żółty). Jeśli do eksperymentu dodamy

diodę o białej barwie okazuje się, iż daje ona najlepsze wyniki [15]. Autorzy tłumaczą, że spektrum światła białego pokrywa spektra trzech kolorów podstawowych, a więc jednocześnie uaktywnia obecne w oku człowieka receptory światła zielonego, czerwonego oraz niebieskiego.



Rys. 3. BCI oparte na SSVEP wykorzystujące diody LED jako bodźce stymulacji



Rys. 4. Parametry urządzeń stymulujących opartych na diodach LED

Kolejnym parametrem jest rozmiar elementu generującego stymulację świetlne. Możemy go rozpatrywać jako rozmiar punktowy, czyli rozmiar pojedynczej diody. Tutaj mamy do dyspozycji LED o wymiarach od 1.8 do nawet 20 mm. Innym rozumieniem rozmiaru może być tutaj fakt, czy w aparaturze stymulującej wykorzystujemy pojedynczą diodę czy też zestaw diod w postaci panelu (np. 3x3 diody). Korzysta się także z rozwiązań, w których pojedynczą diodę umieszcza się

w obudowie o zadanych rozmiarach (np. 2x2 cm). W rezultacie daje to pole światła o rozmiarach obudowy. Badania pokazują, iż rozmiar powierzchni emitującej światło w układzie do stymulacji ma znaczenie dla detekcji SSVEP. W literaturze [16] możemy znaleźć prace potwierdzające fakt, że większy rozmiar bodźca daje większe pole światła, a co za tym idzie więcej światła dociera do podmiotu badanego. W odpowiedzi uzyskujemy odpowiednio większą amplitudę SSVEP. Ponadto, stwierdza się, że powierzchnia świecenia dla danego podmiotu może osiągać swoją wartość graniczą – powyżej której nie będzie znaczącej zmiany w amplitudzie SSVEP [17].

W badaniach prowadzonych nad kształtem obszaru świetlnego nie stwierdzono znaczącego jego wpływu na zarejestrowany paradygmat SSVEP [16] – jednak mogło to mieć również związek z całkowitą powierzchnią świecenia punktu.

Odległość przestrzenna pomiędzy źródłami stymulacji została również zbadana przez zespół Duszyk i in.. Badacze nie stwierdzili znaczącego wpływu odległości pomiędzy diodami LED na liczbę wykrywanych częstotliwości SSVEP. Jednakże inny zespół [18], prowadzący prace nad skonstruowaniem klawiatury opartej na paradygmacie SSVEP, rozpatrzył ten parametr pod kątem praktycznym. W eksperymentach przez nich prowadzonych użytkownicy zasugerowali odsunięcie od siebie poszczególnych klawiszy (zbudowanych z migających LED), co miało znacznie poprawić komfort pracy z takim interfejsem.

W literaturze [19] rozpatrywane są również konstrukcje wykorzystujące tak zwany „punkt skupienia". „Punkt skupienia" jest realizowany poprzez umieszczenie w układzie stymulującym jednego dodatkowego obiektu, na którym użytkownik ma skupić swoją uwagę, co pozwala na zmniejszenie niepożądanych ruchów oczu. Przekłada się to na poprawę wyników w badaniach nad SSVEP.

Jako ostatni parametr należy rozpatrzyć warunki środowiskowe. Najbardziej istotnym wydaje się parametr oświetlenia otoczenia. Badania pokazują, iż zaciemnienie pomieszczenia ma pozytywny wpływ na liczbę identyfikowanych częstotliwości SSVEP [20].

Podsumowanie

Układy korzystające z interfejsów BCI są coraz bardziej powszechne. Wykorzystanie tutaj potencjałów SSVEP daje zadowalające wyniki. Urządzenia służące do generowania bodźców świetlnych stosowane w takich BCI w większości przypadków oparte są na migających z daną częstotliwością diodach LED. Użycie LED w takich sytuacjach pozwala na uzyskanie stymulacji z dużego zakresu częstotliwości (80 możliwych do uzyskania częstotliwości z zakresu 5-30 Hz).

Ponadto, możliwe jest odpowiednie skonfigurowanie aparatury stymulującej w zależności od zastosowanego koloru, rozmiaru czy kształtu. Odległość

przestrzenna pomiędzy źródłami stymulacji, występowanie punktu skupienia czy oświetlenie otoczenia są kolejnymi czynnikami, które należy uwzględnić w eksperymentach nad SSVEP.

Cytowana literatura pokazuje, że najlepsze rezultaty otrzymuje się stosując biały i czerwony kolor światła przy jednoczesnym wykorzystaniu odpowiednio dużych obiektów (zestawów LED). Również zastosowanie punktu skupienia poprawia rezultaty. Ponadto w badaniach stwierdza się, iż zaciemnienie otoczenia badanego podmiotu także pozwala na uzyskanie lepszych wyników. Co więcej rozmieszczenie przestrzenne źródeł światła nie ma znaczącego wpływu na detekcję SSVEP. Odpowiednie rozmieszczenie wpływa jednak na zwiększenie komfortu użytkownika.

Dodatkowo połączenie kilku-kilkunastu pojedynczych diod w układzie stymulującym pozwala na zestawienie złożonych interfejsów (np. klawiatury).

Literatura

1. Wolpaw JR, Birbaumer N, Mcfarland DJ, Pfurtscheller G, Vaughan TM (2002) Brain-computer interfaces for communication and control. Clinical Neurophysiology 113: 767–79
2. Fernandez-Vargas J, Pfaff HU, Rodriguez FB, Varona P (2013) Assisted closed loop optimization of SSVEP-BCI efficiency. Frontiers in Neural Circuits, vol. 7
3. Vialatte FB, Maurice M, Dauwels J, Cichocki A (2010) Steady-state visually evoked potentials: focus on essential paradigms and future perspectives. Progress in neurobiology, vol. 90(4), pp. 418-438
4. Regan D (1989) Human brain electrophysiology: evoked potentials and evoked magnetic fields in science and medicine. Elsevier Press
5. Herrmann S (2001) Human EEG responses to 1-100 Hz flicker: resonance phenomena in visual cortex and their potential correlation to cognitive phenomena. Experimental Brain Research, vol. 137(3-4), pp.346-353
6. Zhenghua W et al. (2008) Stimulator selection in SSVEP-based BCI. Medical engineering & physics.30(8): 1079-1088.
7. Graimann B, Allison B, Pfurtscheller G (2010) Brain-Computer Interfaces: Non-Invasive and Invasive Technologies. The Frontiers Collection. Springer
8. Aloise F, Schettini F, Arico P, Salinari S, Babiloni F, Cincotti F (2012) A comparison of classification techniques for a gaze-independent P300-based brain–computer interface. Journal of Neural Engineering. 9(4): 045012
9. Lotte F, Congedo M, Lécuyer A, Lamarche F, Arnaldi B et al. (2007) A review of classification algorithms for EEG-based brain–computer interfaces. Journal of Neural Engineering. Vol. 4, 2/2007

10. Paulus, W (2005) Elektroretinographie (ERG) und visuell evozierte Potenziale (VEP). In: Buch-ner, H., Noth, J. (eds.) Evozierte Potenziale, neurovegetative Diagnostik, Okulographie: Methodik und klinische Anwendungen, Thieme, Stuttgart - New York, pp. 57–65
11. Regan D (1966) An effect of stimulus colour on average steady-state potentials evoked in man. Nature 210, 1056-1057
12. Drew P, Sayres R, Watanabe K, Shimojo S (2001) Pupillary response to chromatic flicker. Experimental Brain Research. 136(2): 256-62
13. Gregory R, (1997) Eye and brain the psychology of seeing. Princeton: Princeton University Press
14. Tello RJMG, Müller SMT, Ferreira A, Bastos TF (2015) Comparison of the influence of stimuli color on Steady-State Visual Evoked Potentials. Research on Biomedical Engeneering vol.31 no.3
15. Aljshamee M, Mohammed MQ, Choudhury RUA, Malekpour A and Luksch P (2014) Beyond Pure Frequency and Phases Exploiting: Color Influence in SSVEP Based on BCI. Computer Technology and Application 5,111-118
16. Duszyk A, Bierzyńska M, Radzikowska Z, Milanowski P, Kuś R, Suffczyński P, Michalska M, Łabęcki M, Zwoliński P, Durka P (2014) Towards an Optimization of Stimulus Parameters for Brain-Computer Interfaces Based on Steady State Visual Evoked. PLoS One. 2014 Nov 14; 9(11) :e11 2099
17. Ng KB, Bradley AP, Cunnington R (2012) Stimulus specificity of a steady-state visual-evoked potential-based brain–computer interface. Journal of Neural Engineering, vol 9 (3)
18. Hwang HJ, Lim JH, Jung YJ, Choi H, Lee SW, Ima CH (2012) Development of an SSVEP-based BCI spelling system adopting a QWERTY-style LED keyboard. Journal of Neuroscience Methods 208 59–65
19. Meese TS, Summers RJ, Neuronal convergence in early contrast vision: Binocular summation is followed by response nonlinearity and area summation, Journal of Vision 9 (4), 7-7, 2009
20. Wang N, Qian T, Zhuo Q, Gao X (2010) Discrimination between idle and work states in BCI based on SSVEP. In proc. 2nd International Advanced Computer Control Conference 4: 355–358

Streszczenie

Odpowiedź mózgu na bodziec powtarzany ze stałą częstotliwością (np. migające światło diody LED) nazywana jest potencjałem stanu ustalonego (SSVEP ang. Steady State Visually Evoked Potential). W konsekwencji takiej stymulacji w sygnale EEG (Elektroencefalogram) rejestrowanym znad kory wzrokowej następuje wyraźny wzrost mocy w paśmie częstotliwości odpowiadającym częstotliwości bodźca stymulującego.

Posiadając układ stymulujący, wyposażony w migającą z daną częstotliwością diodę LED oraz wykorzystując aparaturę do pomiaru EEG (elektrody pomiarowe umiejscowione na czaszce podmiotu badanego) możliwe jest skonstruowanie interfejsu mózg-komputer (BCI ang. Brain-Computer Interface), który może być z powodzeniem wykorzystany np. jako układ sterujący wózkiem inwalidzkim dla osób niepełnosprawnych.

Użycie rozwiązania opartego na diodach LED, przy uwzględnieniu standardowego użytecznego zakresu częstotliwościowego sygnału EEG (5-30Hz), daje około 80 możliwych częstotliwości stymulacji. Stanowi to znaczny zbiór częstotliwości możliwych do wykorzystania na etapie uczenia się interfejsu BCI. Etap ten jest konieczny, aby wybrać charakterystyczne dla badanego podmiotu częstotliwości stymulacji dające jak najsilniejszą odpowiedź SSVEP.

W artykule autor przedstawi metodę komunikacji w interfejsie BCI opartą na SSVEP z wykorzystaniem diody LED ze wskazaniem na najbardziej istotne parametry budowy układów stymulacyjnych.

Abstract

The response of the brain to a stimulus repeated with a constant frequency (eg. flashing LED), is called a Steady State Visually Evoked Potential (SSVEP). As a consequence of the stimulation, the EEG signal (electroencephalogram) recorded from the visual cortex shows a significant power increase in the frequency band corresponding to the stimulus frequency.

That means that using a stimulation equipment (with LED flashing with the given frequency) and EEG device recording signals from electrodes placed on the subject's skull, it is possible to construct the brain-computer interface (BCI). It can be used successfully e.g., as a control system for a wheelchair for disabled people.

BCI based on LEDs provides a high number of possible stimulation frequencies. Considering the classic EEG frequency band (5-30 Hz) at least 80 different stimulation frequencies can be delivered by a single LED. This large set of frequencies is used at the BCI learning stage. This stage is necessary in order to select specific stimulation frequencies, which give the strongest SSVEP for a specific subject.

In the article the author will present the method of communication in BCI interface based on the SSVEP using LEDs. The most important parameters of the stimulating systems will be indicated.

Keywords: stimulating equipment, SSVEP, brain-computer interface, LED