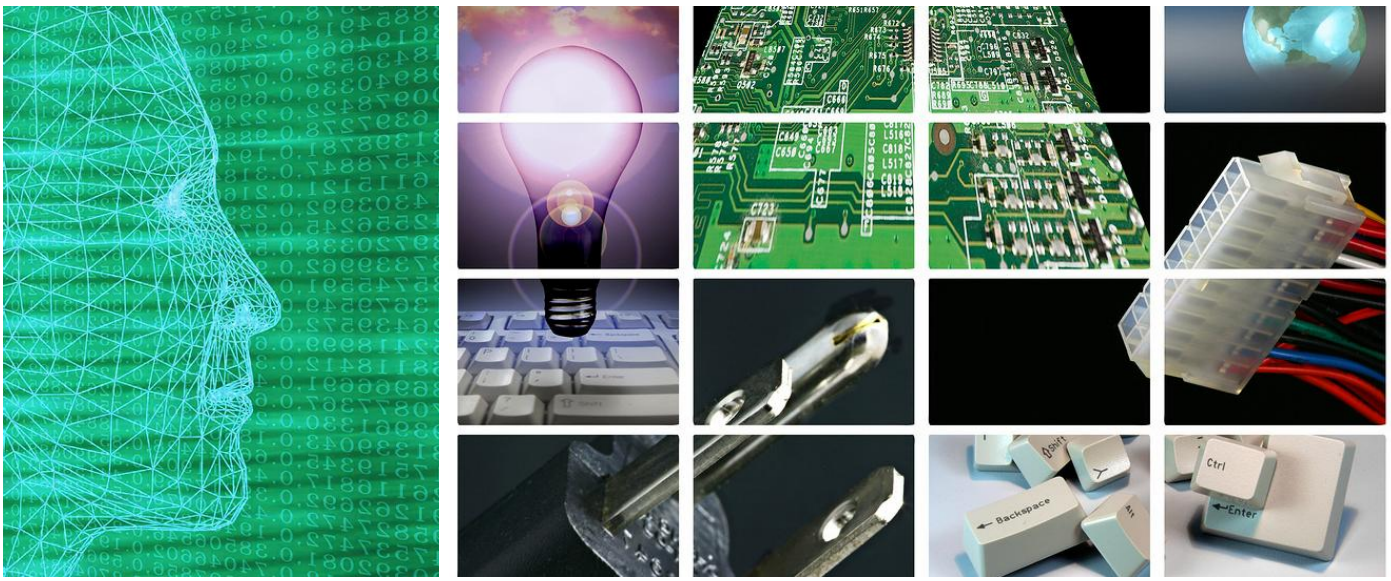
ТЕХНИЧЕСКИ УНИВЕРСИТЕТ - ВАРНА
TECHNICAL UNIVERSITY OF VARNA

Година XVI, Брой 2/2018

КОМПЮТЪРНИ НАУКИ И ТЕХНОЛОГИИ



IEEE Bulgaria Communications Chapter

FCA Faculty of Computing & Automation

COMPUTER SCIENCE AND TECHNOLOGIES

Пета научна конференция с международно участие
"Компютърни науки и технологии"
28 - 29 септември 2018 г., Варна, България

Year XVI, Number 2/2018

Компютърни науки и технологии

Издание
на Факултета по изчислителна техника и автоматизация
Технически университет - Варна

Редактор: доц. д-р Ю. Петкова
Гл. редактор: доц. д-р Н. Николов

Редакционна колегия:

проф. д.н. Л. Личев (Острава)
проф. д.н. М. Илиев (Русе)
проф. д-р М. Лазарова (София)
проф. д-р Г. Спасов (Пловдив)
доц. д-р П. Антонов (Варна)
доц. д-р Р. Вробел (Вроцлав)
доц. д-р Т. Ганчев (Варна)
доц. д-р Н. Атанасов (Варна)
доц. д-р М. Маринов, (Русе)

Печат: ТУ-Варна

За контакти:

Технически университет - Варна
ФИТА
ул. „Студентска” 1, 9010 Варна,
България
тел./факс: (052) 383 320
e-mail: ned.nikolov@tu-varna.bg
yulka.petkova@tu-varna.bg

ISSN 1312-3335

Computer Science and Technologies

Publication
of Computing and Automation Faculty
Technical University of Varna

Editor: Assoc. Prof. Y. Petkova, PhD
Chief Editor: Assoc. Prof. N. Nikolov, PhD

Editorial Board:

Prof. L. Lichev, DSc (Ostrava)
Prof. M. Iliev, DSc (Ruse)
Prof. M. Lazarova, PhD (Sofia)
Prof. G. Spasov, PhD (Plovdiv)
Assoc. Prof. P. Antonov, PhD (Varna)
Assoc. Prof. R. Wrobel (Wroclaw)
Assoc. Prof. T. Ganchev, PhD (Varna)
Assoc. Prof. N. Atanasov, PhD (Varna)
Assoc. Prof. M. Marinov, PhD (Ruse)

Printing: TU-Varna

For contacts:

Technical University of Varna
Faculty of Computing and Automation
1, Studentska Str., 9010 Varna,
Bulgaria
Tel/Fax: (+359) 52 383 320
e-mail: ned.nikolov@tu-varna.bg
yulka.petkova@tu-varna.bg

ISSN 1312-3335

СПОНСОРИ НА КОНФЕРЕНЦИЯТА

ResearchMetrics Powering the research datascape http://www.researchmetrics.com	**Рисърчметрикс** е международна компания с офиси в Северна Америка, Европа и Азия, в които работят над 100 инженери и специалисти. Фирмата разработва иновационни решения, работещи в областта на набора и анализа на данни и усъвършенстване на потребителското обслужване.
ACT BG http://www.actbg.bg	**АКТ БГ ООД** е създадена в началото на 1997г. като 100% частна фирма с предмет на дейност: •Доставка, проектиране, инсталиране, абонаментно обслужване и сервиз на компютърно-комуникационно оборудване и офис техника; •Проектиране и внедряване на ИТ решения за високонадеждни информационни системи; •Информационна сигурност и защита на личните данни; •Изграждане на локални и глобални мрежи; •Проектиране и изграждане на системи за контрол на достъп; •Персонализиране на апаратно-програмни решения и системи.
ADASTRA Academy https://adastracorp.com/academy	**Adastra Academy** е уникалният бранд, който отразява визията на Адастра България за успешни служители чрез непрестанно обучение в рамките на вътрешни и външни инициативи, алтернативни начини за споделяне на знания и иновации.
EuroRisk Systems Ltd. http://www.eurorisksystems.com	**EuroRisk Системи ООД** предоставя софтуерни системи и услуги за банки, застрахователни и финансови институции и други доставчици на финансови услуги.
Beehive Varna Co-Working Space http://beehive.bg	**Beehive** е първото във Варна споделено работно място с 5 години опит в организирането на културни събития и проекти. Провежда над 100 безплатни обучения ежегодно. На фокус са мащабни и международни конференции, startup, предприемачески формати и арт проекти. Успешно реализира проекти и в други градове, сред които са TEDx, Startup Weekend, Europe Code Week. Събира и обединява идейни и креативни личности, като им предоставя среда за ефективно развитие на бизнес и творчески интереси. Като менторска програма обучава младежи в началото на своята кариера и спомага за по-лесната им бъдеща реализация.
ADASTRA http://www.bg.adastragrp.com	**АДАСТРА** предлага информационен мениджмънт - пълна гама услуги, насочени към рационализация и ускоряване на бизнес процеси или обновяване на фирмена технологична инфраструктура.
The Taxback Group http://www.taxback.com	**Taxback Group** предлага пакет от услуги, сред които специализирани данъчни декларации, финансови и туристически консултантски услуги за физически лица и корпоративни клиенти. Компанията предлага възможности за развитие за специалисти в сферата на информационните технологии и програмирането.

http://smartpro-bg.com/	Създадена преди 20 години в град Варна, **SmartPro Ltd.** е софтуерна компания, специализирана в развитието и внедряването на софтуерни решения в помощ на бизнеса. Основен на продукт е ERP системата „Интегра“ - успешно внедрена в над 200 компании.
https://www.flatrocktech.com/	**Flat Rock Technology Ltd.** е софтуерна компания, развиваща се в две направленията разработка на софтуерни и мобилни приложения и предоставяне на аутсорсинг услуги. Основана преди 10 години в Лондон към момента в централния офис в град Варна работят над 130 специалисти. Flat Rock Technology дава възможност на младите да започнат своя кариерен път като програмисти, маркетинг специалисти, анализатори на данни и още много други, ръководени от професионалисти в сферата.
http://www.immedis.com	**Immedis** е най-бързо развиващата се международна пейрол технологична компания в света. За офиса си в град Варна Immedis набира служители на следните позиции: Net Developers, Business Intelligence and Data Analysts, Business Analysts, Quality Assurance Specialists, Wintel Server Administrators, UI/UX Design Specialists, AWS Administrators and Database Administrators.

acmbul

Bulgarian ACM Chapter

Computer Sciences and Engineering

is the fifth international scientific conference organized by the Computer Science and Engineering and Software and Internet Technologies Department, Technical University of Varna, Bulgaria in association with the Bulgarian ACM Chapter - **acmbul**. This Conference is the 32nd **acmbul** event during the last 28 years

СБОРНИК ДОКЛАДИ

Пета научна конференция с международно участие
"Компютърни науки и технологии"
28 – 29 септември 2018 г.
Варна, България

Fifth International Scientific Conference
Computer Sciences and Engineering
28 – 29 September, 2018
Varna, Bulgaria

PROCEEDINGS

Председатели:	**Chairmen:**
Христо Вълчанов	Hristo Valchanov
Виолета Божикова	Violeta Bozhikova

Организационен комитет	**Organizing committee**
Председател:	**Chairman:**
Венета Алексиева	Veneta Aleksieva
Членове:	**Members:**
Ивайло Пенев	Ivaylo Penev
Диян Желев	Diyan Zhelev
Николай Дуков	Nikolay Dukov
Мая Тодорова	Maya Todorova
Айдън Хъкъ	Aydan Haka
Антоанета Иванова	Antoaneta Ivanova
Илиян Бойчев	Iliyan Boychev
Стефка Попова	Stefka Popova

Програмен комитет	**Programme committee**
Председател:	**Chairman:**
Марияна Стоева	Mariyana Stoeva
Членове:	**Members:**
Валентина Кукенска	Valentina Kukenska
Никола Николов	Nikola Nikolov
Радослав Вробел	Radoslaw Wrobel
Розалина Димова	Rozalina Dimova
Милена Лазарова	Milena Lazarova
Синиша Илич	Sinisa Ilic
Гриша Спасов	Grisha Spasov
Румен Трифонов	Rumen Trifonov
Делян Генков	Delyan Genkov
Ирена Вълова	Irena Valova
Милко Маринов	Milko Marinov
Марта Сийбауер	Marta Seebauer

СЪДЪРЖАНИЕ **CONTENTS**

СЕКЦИЯ 3
АВТОМАТИКА И КОМПЮТЪРНИ СИСТЕМИ ЗА УПРАВЛЕНИЕ
КОМУНИКАЦИОННИ СИСТЕМИ И ТЕХНОЛОГИИ

Пета научна конференция с международно участие
"Компютърни науки и технологии"
28 – 29 септември, 2018 г.
Варна, България

Fifth International Scientific Conference
Computer Sciences and Engineering
28 – 29 September, 2018
Varna, Bulgaria

SECTION 3
AUTOMATION AND COMPUTER CONTROL SYSTEMS
COMMUNICATION SYSTEMS AND TECHNOLOGIES

FPGA BASED CANNY: EXPLORING MATHEMATICAL ACCURACY AND APPROXIMATION

Dimitre Zh. Kromichev

Abstract: Exact Canny algorithm's mathematics is generally considered very difficult for FPGA hardware implementation. To circumvent this situation, the available realizations resort to various approximation approaches. They are focused on Sobel, gradient magnitude and direction modules. So far, there has been no in-depth analysis of the approximations' impact on the detected contours' precision. This paper embarks on the task of thoroughly exploring and assessing mathematical accuracy and its deviations in the FPGA based Canny computations.
Keywords: Canny, FPGA, mathematics, accuracy, approximation, methodology, synthetic images, real life images

1. Introduction

The available FPGA based Canny implementations [5], [6], [7], [10] rely on approximations [3], [4] without a thorough analysis of their cost in terms of the mapped contours' quality [1], [2], [8], [9]. The total number of combinations among all available approximation variants in Sobel, gradient magnitude and direction modules is 35. For the evaluation to be comprehensive and objective, the approximation approaches need to be tested in combination with exact mathematics by targeted modules so that each computational stage's individual contribution can be properly ascertained and assessed. For the purpose, all the possible combinations are scrutinized to evaluate the impact of accuracy and its deviations on detected contours' precision. The targeted hardware is Intel (formerly Altera) FPGAs. The software tool used to conduct tests is Scilab programming environment, Scilab Image and Video Processing Toolbox, Scilab Image Processing Design Toolbox. The prformed analyses and conclusions arrived at are relevant only for gray-scale images.

2. Combinations of approximation variants to be tested

The following designations are utilized to represent the combinations:

1) For exact mathematics: Sobel non-scaling down, Magnitude exact, Gradient exact.
2) For approximation variants: Sobel #1, Sobel #2, Magnitude #1, Magnitude #2, Direction #1, Direction #2, Direction #4.

Thus, the complete list of combinations to be tested is:

Table 1. Total number of possible approximation combinations

Total number of possible combinations: approximation variants, exact mathematics and approximation variants
1. Sobel #1 - Magnitude #1 - Direction #1
2. Sobel #1 - Magnitude #1 - Direction #2
3. Sobel #1 - Magnitude #1 - Direction #4
4. Sobel #1 - Magnitude #2 - Direction #1
5. Sobel #1 - Magnitude #2 - Direction #2
6. Sobel #1 - Magnitude #2 - Direction #4
7. Sobel #2 - Magnitude #1 - Direction #1
8. Sobel #2 - Magnitude #1 - Direction #2

9. Sobel #2 - Magnitude #1 - Direction #4
10. Sobel #2 - Magnitude #2 - Direction #1
11. Sobel #2 - Magnitude #2 - Direction #2
12. Sobel #2 - Magnitude #2 - Direction #4
13. Sobel non-scaling down – Magnitude #1 – Direction #1
14. Sobel non-scaling down – Magnitude #1 – Direction #2
15. Sobel non-scaling down – Magnitude #1 – Direction #4
16. Sobel non-scaling down – Magnitude #2 – Direction #1
17. Sobel non-scaling down – Magnitude #2 – Direction #2
18. Sobel non-scaling down – Magnitude #2 – Direction #4
19. Sobel #1 – Magnitude exact – Direction #1
20. Sobel #1 – Magnitude exact – Direction #2
21. Sobel #1 – Magnitude exact – Direction #4
22. Sobel #2 – Magnitude exact – Direction #1
23. Sobel #2 – Magnitude exact – Direction #2
24. Sobel #2 – Magnitude exact – Direction #4
25. Sobel #1 – Magnitude #1 – Direction exact
26. Sobel #1 – Magnitude #2 – Direction exact
27. Sobel #2 - Magnitude #1 – Direction exact
28. Sobel #2 - Magnitude #2 – Direction exact
29. Sobel non-scaling down – Magnitude exact - Direction #1
30. Sobel non-scaling down – Magnitude exact – Direction #2
31. Sobel non-scaling down – Magnitude exact – Direction #4
32. Sobel #1 – Magnitude exact – Direction exact
33. Sobel #2 – Magnitude exact – Direction exact
34. Sobel non-scaling down – Magnitude #1 – Direction exact
35. Sobel non-scaling down – Magnitude #2 – Direction exact

The approximation represented by fixed predefined pair of values replacing the calculation of high and low thresholds is not explicitly included. This is due to the fact that for each combination a complete set of high and low threshold pairs is tested to select the most appropriate values.

3. Image database, methodology and results

Two groups of gray-scale images are used in the experiments – synthetic and real life. All are sourced from Internet accessible databases. All synthetic images are selected to represent geometric fugures: square, rectangle, circle, semicircle, regular hexagon, ellipse, symmetrical double T shape. The real life images encompass many spheres of the sorronding world, with preference for machine gear. All test programs are written in Scilab.

Synthetic images have the advantage of representing a distinctive difference between background and object with respect to both pixel magnitudes and their relative homogeneity. They are generally characterized by a single level discontinuity.The reference contour defined for each tested image is computed by using the following staple requirements:

1) The comparative assessment of accurate mathematics and its deviations must be carried out under the same test conditions. Therefore, the smoothing mask should be a constant. The appropriate option is Gaussian filter of size 5x5, $\sigma = 1.4$ and normalization factor 159.
2) Employ only exact mathematics.
3) Experimentally select the optimal high and low thresholds for each particular test.

On that basis, to define the reference contour, all of the conducted experiments encompass the following steps (pixels are utilized for all numerical measures):

1) In synthetic images object boundaries are usually distinctively marked, so all the pixels pertaining to the object and bordering on pixels belonging to the background are numerically evaluated by counting them prior to applying Gaussian smoothing.
2) Numerically measured is the exact distance between the object boundaries and the outermost columns and rows of the image before Gaussian smoothing.
3) The perimeter of each figure is calculated before applying Gaussian filter.
4) The area of each figure is calculated prior to Gaussian smoothing
5) Calculate the distance between mapped contours and the outermost columns and rows in the hysteresis thresholded image. Compare it with the distance calculated before Gaussian filtering.

The reference contours for each geometric shape having been defined, the targeted comparative evaluation focused on all 35 combinations (Table 1) has the required platform to be realized. For the purpose, the following methodology and tools are utilized:

1) A set of programs is designed to test all combination in Table 1. Contours resulting from each combination are numerically compared with the respective reference contours.
2) Add noise to the input images for both reference contours and combinations. With each type of moise, images are tested for 10 different noise parameter values..
3) Add gradients to the input images for both reference contours and combinations.
4) All experiments are conducted on the basis of selecting the most appropriate high and low thresholds by iterative calculations.

For real life images, the following research methodology is applied. All contour detected images - these obtained by employing only exact mathematics and those achieved by using approximation combinations (Table1) are divided into a number of small sized blocks. Each block contains portion of the entire contour detected image and is scrutinized for the targeted parameters – localization, thinness, false negatives and positives. Numerical evaluation includes these steps:

1) Count the detected contour pixels in the images processed by using both exact mathematics and approximation combinations. Compare their number for each block.
2) Calculate the distance between mapped contours and the outermost columns and rows in the block. Compare their precise localization for exact mathematics and approximation combinations within each block.
3) Count the number of contour pixels along each block's columns and rows to secure information on the mapped contours' thinness. Compare the results for exact mathematics and approximation.
4) Determine the number of false negatives within each block. This is achieved on a pixel-wise basis by comparing the data achieved in steps *1), 2)* and *3)*
5) Define the number of false positives by utilizing the data accumulated by performing steps *1), 2), 3)* and *4)* to evaluate the differences between exact mathematics and approximation.
6) Combine the data obtained for each block in steps *1), 2), 3), 4)* and *5)* in a single database representing the entire contour detected image being analyzed.
7) On the basis of steps *1), 2), 3), 4), 5)* and *6)* comparatively assess detected contours achieved by employing only exact mathematics with those obtained by utilizing the approximation with respect to the following points of interest: lines, curves (concave, convex), circles, ellipses, junctions (T, L,Y, X), angles (acute, right, obtuse).

All experiments are conducred on the basis of selecting the most appropriate high and low thresholds by iterative calculations.

The results achieved for synthetic and real life images are exhibited in the tables below (Tables 2 to 6).

3.1. Synthetic images

The presentation of numerically evaluated differences is in percentage terms for each of the targeted category. The tables' layout in terms of sequential ordering of the analyzed combinations follows strictly the order of combinations in Table 1.

Table 2. Largest cumulative differences between the geometric figures' reference contours defined utilising only exact mathematics and detected contours obtained by using combinations in Table 1

Combinations between approximation variants	Values						
	Largest cumulative numerical differences between the geometric figures' reference contours defined utilising only exact mathematics and detected contours obtained by using approximation variants in percentage terms						
	Square	Rectangle	Circle	Semicircle	Regular hexagon	Ellipse	Symmetrical double T shape
1.	14%	14%	21%	21%	19%	22%	20%
2.	13%	14%	21%	20%	19%	22%	21%
3.	14%	14%	22%	21%	18%	21%	21%
4.	14%	14%	21%	20%	18%	22%	20%
5.	13%	14%	22%	21%	19%	22%	22%
6.	13%	14%	21%	20%	19%	21%	22%
7.	14%	14%	22%	21%	20%	21%	21%
8.	13%	14%	22%	21%	19%	21%	20%
9.	14%	14%	21%	20%	20%	22%	21%
10.	13%	14%	22%	21%	19%	21%	20%
11.	13%	14%	21%	21%	19%	22%	20%
12.	13%	14%	22%	22%	18%	21%	21%
13.	10%	10%	17%	17%	17%	17%	15%
14.	9%	9%	16%	16%	16%	17%	15%
15.	9%	9%	17%	16%	16%	16%	16%
16.	10%	10%	18%	17%	17%	16%	16%
17.	9%	10%	17%	16%	16%	17%	15%
18.	10%	10%	17%	17%	17%	16%	15%
19.	9%	9%	16%	16%	15%	14%	14%
20.	9%	9%	17%	16%	15%	15%	15%
21.	9%	9%	16%	15%	14%	15%	14%
22.	10%	10%	16%	16%	14%	16%	14%
23.	9%	9%	17%	17%	15%	15%	15%
24.	10%	10%	16%	16%	14%	14%	14%
25.	10%	10%	17%	17%	14%	16%	14%
26.	9%	10%	16%	17%	15%	15%	16%
27.	9%	10%	16%	15%	14%	15%	15%
28.	10%	10%	16%	17%	15%	16%	14%
29.	4%	4%	10%	10%	11%	10%	10%
30.	5%	5%	10%	10%	10%	10%	10%
31.	5%	5%	9%	10%	10%	11%	9%
32.	4%	4%	10%	9%	11%	10%	10%
33.	4%	4%	10%	9%	9%	11%	10%
34.	5%	5%	9%	10%	9%	11%	9%
35.	4%	4%	10%	9%	10%	10%	9%

Table 3. Largest cumulative differences between the geometric figures' reference contours defined utilizing only exact mathematics and detected contours obtained by using combinations in Table 1

Combinations between approximation variants	Values						
	Largest cumulative numerical differences between the geometric figures' reference contours defined utilizing only exact mathematics and detected contours obtained by using approximation variants with Gaussian additive noise added to the input images. The results are in percentage terms						
	Square	Rectangle	Circle	Semicircle	Regular hexagon	Ellipse	Symmetrical double T shape
1.	21%	22%	30%	31%	30%	32%	30%
2.	21%	22%	34%	32%	32%	33%	33%
3.	21%	22%	31%	33%	30%	32%	31%
4.	20%	21%	30%	32%	30%	34%	30%
5.	22%	22%	31%	33%	32%	32%	32%
6.	22%	23%	35%	34%	31%	34%	34%
7.	20%	21%	31%	31%	32%	33%	31%
8.	21%	21%	32%	31%	33%	34%	30%
9.	20%	20%	30%	32%	32%	32%	32%
10.	20%	20%	31%	33%	32%	34%	32%
11.	21%	22%	32%	34%	31%	34%	30%
12.	20%	21%	30%	32%	30%	34%	31%
13.	16%	17%	26%	30%	25%	27%	25%
14.	17%	17%	27%	31%	23%	27%	23%
15.	16%	16%	25%	28%	23%	27%	25%
16.	16%	17%	26%	28%	25%	26%	25%
17.	17%	17%	27%	30%	23%	27%	25%
18.	16%	16%	27%	30%	23%	26%	23%
19.	15%	16%	21%	22%	23%	25%	25%
20.	15%	15%	22%	23%	23%	26%	23%
21.	16%	16%	23%	25%	25%	27%	25%
22.	15%	17%	21%	22%	23%	26%	25%
23.	15%	16%	23%	25%	23%	25%	23%
24.	16%	16%	20%	23%	25%	27%	25%
25.	16%	17%	23%	25%	23%	26%	25%
26.	15%	15%	22%	23%	25%	25%	23%
27.	16%	17%	21%	22%	23%	26%	25%
28.	16%	16%	20%	21%	25%	26%	25%
29.	10%	10%	15%	15%	15%	17%	15%
30.	11%	11%	14%	14%	15%	16%	14%
31.	11%	11%	15%	14%	14%	15%	15%
32.	10%	10%	16%	12%	15%	15%	14%
33.	10%	11%	15%	10%	14%	17%	14%
34.	10%	10%	15%	11%	14%	16%	15%
35.	9%	10%	14%	9%	15%	17%	14%

Table 4. Largest cumulative differences between the geometric figures' reference contours defined utilizing only exact mathematics and detected contours obtained by using combinations in Table 1

Combinations between approximation variants	Values						
	Largest cumulative numerical differences between the geometric figures' reference contours defined utilizing only exact mathematics and detected contours obtained by using approximation variants with salt and pepper noise added to theinput images. The results are in percentage terms						
	Square	Rectangle	Circle	Semicircle	Regular hexagon	Ellipse	Symmetrical double T shape
1.	19%	20%	30%	28%	30%	32%	30%
2.	21%	22%	32%	32%	31%	31%	33%
3.	20%	21%	31%	32%	30%	32%	31%
4.	19%	20%	32%	32%	30%	30%	29%
5.	21%	22%	32%	32%	31%	31%	32%
6.	20%	21%	31%	30%	32%	32%	34%
7.	20%	21%	31%	32%	32%	30%	28%
8.	21%	21%	32%	31%	30%	32%	30%
9.	20%	20%	33%	31%	32%	31%	28%
10.	20%	21%	31%	32%	32%	30%	28%
11.	21%	21%	32%	32%	31%	31%	30%
12.	21%	22%	33%	32%	30%	30%	28%
13.	17%	17%	26%	25%	26%	25%	25%
14.	16%	17%	27%	26%	25%	25%	25%
15.	16%	16%	25%	25%	25%	25%	23%
16.	17%	17%	26%	25%	26%	23%	23%
17.	17%	17%	27%	26%	25%	25%	22%
18.	16%	17%	27%	25%	25%	23%	23%
19.	16%	16%	26%	25%	23%	22%	20%
20.	16%	17%	26%	26%	23%	23%	22%
21.	15%	16%	27%	25%	25%	23%	21%
22.	16%	16%	26%	26%	23%	25%	20%
23.	15%	16%	27%	25%	23%	25%	21%
24.	16%	16%	27%	26%	25%	23%	20%
25.	16%	17%	26%	25%	23%	22%	16%
26.	15%	16%	26%	25%	25%	23%	17%
27.	16%	17%	27%	26%	25%	23%	16%
28.	15%	15%	26%	25%	23%	22%	15%
29.	10%	10%	16%	15%	16%	16%	10%
30.	10%	10%	15%	14%	15%	15%	11%
31.	10%	10%	16%	14%	15%	15%	11%
32.	9%	9%	13%	12%	15%	15%	10%
33.	10%	10%	10%	10%	12%	16%	10%
34.	9%	9%	11%	11%	10%	15%	10%
35.	9%	9%	10%	9%	10%	16%	10%

Table 5. Largest cumulative differences between the geometric figures' reference contours defined utilizing only exact mathematics and detected contours obtained by using combinations in Table 1

Combinations beteen approximation variants	Values						
	Largest cumulative numerical differences between the geometric figures' reference contours defined utilizing only exact mathematics and detected contours obtained by using approximation variants with gradients added to the input images. The results are in percentage terms						
	Square	Rectangle	Circle	Semicircle	Regular hexagon	Ellipse	Symmetrical double T shape
1.	20%	21%	32%	30%	31%	33%	30%
2.	23%	23%	32%	32%	32%	32%	30%
3.	21%	22%	31%	31%	30%	32%	31%
4.	20%	20%	31%	32%	28%	33%	30%
5.	22%	22%	31%	31%	32%	31%	31%
6.	20%	21%	34%	33%	30%	32%	32%
7.	20%	21%	31%	32%	32%	32%	30%
8.	21%	21%	32%	32%	30%	31%	30%
9.	20%	20%	33%	31%	30%	31%	28%
10.	20%	20%	31%	30%	31%	32%	28%
11.	21%	22%	32%	31%	32%	33%	30%
12.	20%	20%	30%	32%	30%	31%	28%
13.	16%	16%	26%	30%	25%	26%	23%
14.	15%	16%	27%	31%	23%	25%	22%
15.	16%	16%	25%	28%	23%	25%	23%
16.	15%	15%	26%	28%	25%	26%	22%
17.	15%	16%	27%	30%	23%	25%	22%
18.	16%	16%	27%	30%	25%	25%	23%
19.	15%	15%	23%	22%	23%	22%	21%
20.	16%	16%	25%	23%	23%	23%	22%
21.	15%	15%	23%	25%	22%	25%	20%
22.	16%	16%	25%	22%	23%	23%	22%
23.	15%	15%	23%	25%	23%	22%	21%
24.	16%	16%	25%	23%	25%	21%	20%
25.	15%	16%	23%	25%	22%	22%	23%
26.	16%	17%	25%	23%	23%	23%	21%
27.	15%	15%	25%	22%	23%	21%	20%
28.	16%	16%	23%	21%	22%	22%	20%
29.	10%	10%	12%	15%	14%	16%	14%
30.	9%	9%	12%	14%	14%	14%	13%
31.	9%	9%	16%	14%	13%	15%	14%
32.	8%	8%	12%	12%	12%	16%	13%
33.	8%	8%	10%	10%	14%	14%	13%
34.	9%	9%	11%	11%	13%	14%	14%
35.	9%	9%	10%	9%	13%	14%	13%

3.2. Real life images

With respect to the aforesaid methodology for testing the impact of approximation combinations (Table 1) on detected contours' precision, in the table below summarized are all the results achieved by tracking the points of interest - lines, curves (concave, convex), circles, junctions (T, L,Y, X), ellipses, angles (acute, right, obtuse) taking into account the staple targeted parameters: localization, thinness, false negatives and positives. The exhibited numerical evaluation results show the deviations from exact mathematics based contours. They are presented in percentage terms.

Table 6. Largest cumulative numerical differences between contour detected real-life images utilising only exact mathematics and detected contours obtained by using combinations in Table 1

Combinations beteen approximation variants	**Values**						
	Largest cumulative numerical differences between contour detected real-life images utilising only exact mathematics and detected contours obtained by using approximation variants in percentage terms						
	Lines	**Curves (concave, convex)**	**Circles**	**Junctions**		**Ellipses**	**Angles (acute, obtuse)**
				T, L	**Y, X**		
1.	15%	22%	21%	34%	35%	22%	34%
2.	14%	21%	22%	33%	35%	20%	33%
3.	15%	23%	21%	34%	35%	20%	35%
4.	14%	22%	21%	35%	36%	21%	33%
5.	16%	20%	21%	33%	35%	21%	32%
6.	14%	21%	20%	34%	34%	22%	32%
7.	15%	22%	21%	34%	37%	20%	34%
8.	16%	23%	21%	35%	35%	21%	33%
9.	14%	22%	22%	33%	36%	21%	34%
10.	15%	21%	22%	35%	34%	22%	35%
11.	15%	21%	21%	33%	35%	22%	33%
12.	14%	22%	22%	34%	36%	21%	32%
13.	10%	16%	16%	28%	30%	16%	27%
14.	9%	15%	17%	28%	31%	16%	27%
15.	9%	17%	15%	27%	30%	15%	26%
16.	10%	16%	18%	26%	30%	15%	28%
17.	9%	15%	17%	28%	31%	15%	27%
18.	10%	17%	16%	27%	30%	15%	28%
19.	9%	15%	16%	28%	30%	16%	26%
20.	10%	14%	17%	30%	28%	14%	28%
21.	9%	16%	16%	30%	31%	15%	28%
22.	10%	10%	17%	27%	30%	14%	26%
23.	10%	16%	18%	28%	30%	14%	27%
24.	10%	14%	17%	28%	28%	14%	28%
25.	9%	15%	16%	27%	31%	16%	26%
26.	9%	15%	15%	27%	30%	15%	26%
27.	9%	15%	17%	28%	31%	16%	27%
28.	10%	14%	16%	27%	30%	14%	26%
29.	5%	10%	9%	18%	21%	10%	17%
30.	4%	9%	9%	19%	20%	9%	16%

31.	5%	10%	10%	18%	21%	10%	17%
32.	4%	10%	9%	19%	22%	9%	16%
33.	5%	9%	9%	18%	21%	10%	15%
34.	4%	10%	9%	19%	21%	10%	16%
35.	5%	10%	10%	19%	22%	9%	17%

4. Analysis of results

All the data exhibited in Tables 2 - 6 shows that the larger the deviation from exact mathematics, the poorer the detected contours' precision. The worst results in terms of contours' quality are achieved when combinations do not include exact mathematics. In view of the fact that the largest cumulative difference between accurate mathematics based reference contour and contours obtained by utilizing approximation reaches a hefty 34%, the inherent incompatibility between approximation and precision is peremptorily proven.

The data obtained from contour detected real life images shows that combinations 1 - 12

(Table 1) have the strongest negative effect on precision. And it is these 12 combinations that are the most widely used computational approaches in the FPGA based Canny.

In terms of points of interest, the largest cumulative differences between exact mathematics and approximation combinations are ascertained for T, L, Y and X junctions, and the smallest differences – for lines. The individual contribution of each approximation variants is clearly trackable on the basis of the data exhibited in Table 6. When approximation variants are combined the result is deviations in the comparatively assessed targeted parameters of up to 37%.

5. Conclusions

The numerical results presented in percentage terms definitely confirm the effectiveness of conducted analyses. In total, the data exhibited in Tables 2 - 6 leads to the following recapitulation: all available approximation variants of FPGA based Canny's essential mathematics, working individually or cooperating among each other, impact detected contours' quality in a way which is definitely incompatible with the very concept of precision.

References

[1]. Aasiya Anjum, Sanjay Asutkar (2015), FPGA Implementation of Efficient Edge Detection Using Canny Algorithm, International Journal on Recent and Innovation, Trends in Computing and Communication, Vol. 3 (2), pp. 20-22

[2]. Aravindh G., Manikandababu C. S. (2015), Algprithm and Implementation of Distributed Canny Edge Detector on FPGA, ARPN Journal of Engineering and Applied Sciences, Vol. 10 (7), pp. 3208-3216

[3]. Divya. D, Sushmap S, FPGA Implementation of a Distributed Canny Edge Detector, International Journal of Advanced Computational Engineering and Networking, Vol. 1, 5, 2013 pp. 46-51

[4]. Dhanalakshmi Renganathan, Riyas K S, Anu George, Implementation of Canny Edge Detection Algorithm in FPGA Using HDL, International Journal of Innovative Research in Science, Engineering and Technology, Volume 6 (4), 2017

[5]. D. Narayana Reddy, Vaijanath V.Yerigeri, Harish Sanu, Design and Simulation of Canny Edge Detection, International Journal of Engineering Sciences & Research Technology IJESRT, 3 (11), 2014, pp. 304-310

[6]. D. Sangeetha, P. Deepa, An Efficient Hardware Implementation of Canny Edge Detection Algorithm, International Conference on Embedded Systems (VLSID)., 2016, pp. 512-518

[7]. Fangxin Peng, Xiaofeng Lu, Hengli Lu, Sumin Shen, An improved high-speed Canny edge detection algorithm and its implementation on FPGA, Proceedings of SPIE, Fourth International Conference on Machine Vision (ICMV 2011): Computer Vision and Image Analysis; Pattern Recognition and Basic Technologies, Conference Volume 8350, 2012, pp. 802 - 808
[8]. K. Supraya, Hardware Implementation Of Canny Edge Detection Algorithm With FPGA, Journal of Engineering Science and Technology Vol. 12, No. 9, 2017, pp. 2536-2550
[9]. Ramgundewar, Pallavi, Hingway, S .P. and Mankar, K. (2015), Design of modified Canny Edge Detector based on FPGA for Portable Device, Journal of the International Association of Advanced Technology andScience, Volume 16 (2), pp. 210-214
[10]. Rupalatha, G.Rajesh, K.Nandakumar, Implementation of Distributed Canny Edge Detector on FPGA, International Journal of Computer Engineering Science (IJCES) Volume 3 (5), 2013, pp. 22-28

For contacts:
Ass. Prof. Dimitre Zh. Kromichev
Department of Marketing and International Economic Relations
University of Plovdiv
E-mail: dkromichev@yahoo.com

FPGA BASED CANNY: IMPACT OF NON-MAXIMUM SUPPRESSION MATHEMATICS ON DETECTED CONTOURS' PRECISION

Dimitre Zh. Kromichev

Abstract: This paper targets a specific aspect of FPGA based Canny algorithm's mathematics with respect to its impact on detected contours' precision. Non-maximum suppressoion is a typical characteristic of Canny algorithm. It permits computational variations which are crucial to the quality of mapped contours.
Keywords: Canny, FPGA, non-maximum suppression, mathematics, contours' precision

1. Introduction

A widely spread opinion is that Canny is computationally complicated and difficult to implement in FPGA hardware [3], [4]. Very often, the attempts to enhance performance are at the expense of neglecting essential specifics of Canny mathematics [1], [2], [5], [6]. Non-maximum suppression requires special attention with respect to its crucial impact on detected contours' precision. The objective of this paper is to analyze the influence of a very subtle aspect in local maxima calculations on FPGA based Canny in terms of contour detection results. The targeted hardware is Intel (formerly Altera) FPGAs. The software tool used to conduct tests is Scilab programming environment, Scilab Image and Video Processing Toolbox, Scilab Image Processing Design Toolbox. The prformed analyses and conclusions arrived at are relevant only for gray-scale images.

2. Canny computational model

In functional terms, Canny displays downward dependency, the latter being defined by typologically determined image processing steps and reflected in a definite number of intermediate computational results, each of them marking the completion of a specific task. The generalized flow of computations comprises five distinctive consecutive modules:

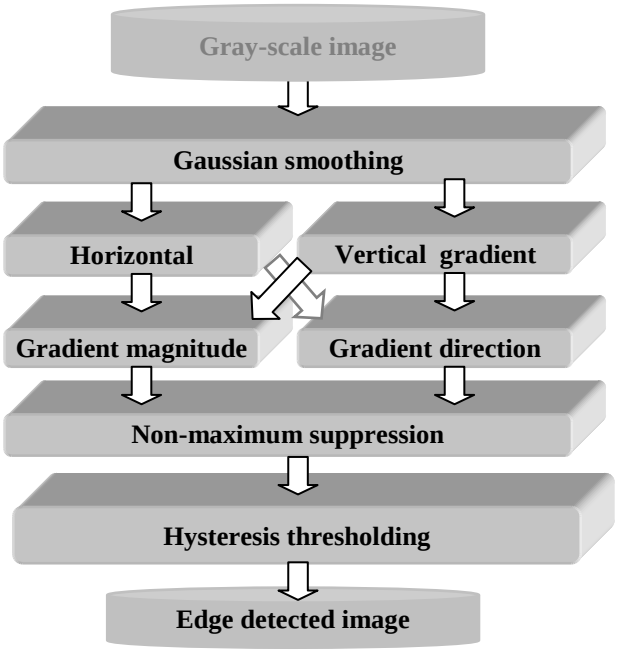


Fig. 1. Canny computational model

3. Non-maximum suppression mathematics

Non-maximum suppression module is a typical characteristic of Canny algorithm's mathematics. Calculating local maxima is based on:

$$M(x,y) = M(x,y) \quad \text{only for}$$
$$[M(x,y) > M\_1(x1,y1) \;\&\; M(x,y) > M\_2(x2,y2)];$$
$$M(x,y) = 0 \text{ in all the other cases,} \qquad (1)$$

where
M(x,y) is a pixel possibly belonging to the edge,
M_1(x1,y1) and M_2(x2,y2) are the magnitudes of the two adjacent pixels positioned along the gradient direction.

4. Exploring non-maximum suppression mathematics. Results and analysis

To test the aforesaid mathematics, two main versions of (1) are to be analyzed and assessed in terms of their impact on the detected contours' precision. A practical illustration is presented in the figures below.

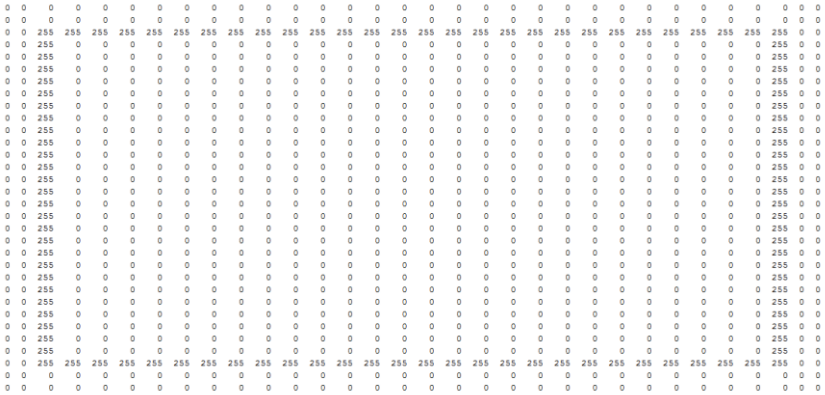


Fig. 2. Contour detected image matrix of a square (Source: Scilab Image and Video Processing Toolbox, Scilab Image Processing Design Toolbox)

And the resultant contour detected image is

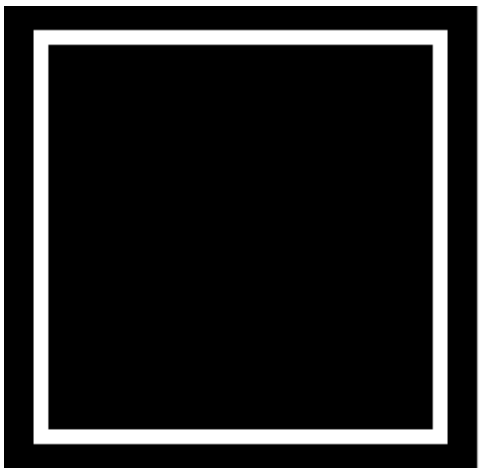


Fig. 3. Resultant contour detected image for matrix in Figure 2 (Source: Scilab Image and Video Processing Toolbox, Scilab Image Processing Design Toolbox)

The important fact here is that this contour can be achieved by not literally implementing (1) for the non-maximum suppression computations, as it is shown in the Scilab program piece below:

```
for i = 2: row-1
  for j =2: col-1
NM = a_new_X_Y_D(i,j);
Mm = 0;
if NM == 0
if (a_new_X_Y_M(i,j) >= a_new_X_Y_M(i,j-1)) & (a_new_X_Y_M(i,j) >=
a_new_X_Y_M(i,j+1))
        Mm = a_new_X_Y_M(i,j);
     else
        Mm = 0;
end
end
if NM == 45
if  (a_new_X_Y_M(i,j) >= a_new_X_Y_M(i+1,j-1)) & (a_new_X_Y_M(i,j) >= a_new_X_Y_M(i-
1,j+1))
     Mm = a_new_X_Y_M(i,j);
     else
     Mm = 0;
end
end
if NM == 90
if  (a_new_X_Y_M(i,j) >= a_new_X_Y_M(i-1,j)) & (a_new_X_Y_M(i,j) >=
a_new_X_Y_M(i+1,j))
     Mm = a_new_X_Y_M(i,j);
     else
     Mm = 0;
end
end
if NM == 135
  if  (a_new_X_Y_M(i,j) >= a_new_X_Y_M(i-1,j-1)) & (a_new_X_Y_M(i,j) >=
a_new_X_Y_M(i+1,j+1))
       Mm = a_new_X_Y_M(i,j);
  else
       Mm = 0;
end
end
a_new_XYLM(i,j) = Mm;
   end
 end
```

Listing 1. Non-maximum suppression programming version #1

Thus, Listing 1 is used instead of the mathematics peremptorily dictated by (1) which, programmed in Scilab, must be

```
for i = 2: row-1
  for j =2: col-1
NM = a_new_X_Y_D(i,j);
Mm = 0;
if NM == 0
if (a_new_X_Y_M(i,j) > a_new_X_Y_M(i,j-1)) & (a_new_X_Y_M(i,j) > a_new_X_Y_M(i,j+1))
        Mm = a_new_X_Y_M(i,j);
```

```
      else
         Mm = 0;
end
end
if NM == 45
if  (a_new_X_Y_M(i,j) > a_new_X_Y_M(i+1,j-1)) & (a_new_X_Y_M(i,j) > a_new_X_Y_M(i-
1,j+1))
      Mm = a_new_X_Y_M(i,j);
      else
      Mm = 0;
end
end
if NM == 90
if  (a_new_X_Y_M(i,j) > a_new_X_Y_M(i-1,j)) & (a_new_X_Y_M(i,j) > a_new_X_Y_M(i+1,j))
      Mm = a_new_X_Y_M(i,j);
      else
      Mm = 0;
end
end
if NM == 135
  if  (a_new_X_Y_M(i,j) > a_new_X_Y_M(i-1,j-1)) & (a_new_X_Y_M(i,j) >
a_new_X_Y_M(i+1,j+1))
       Mm = a_new_X_Y_M(i,j);
  else
       Mm = 0;
end
end
a_new_XYLM(i,j) = Mm;
   end
 end
```

Listing 2. Non-maximum suppression programming version #2

The essential difference between these two versions of programming the non-maximum suppression module is in substituting ‘’>=’’ in version #1 for ‘’>’’ in version #2 . The results from applying literally (1) as in Listing #2 (with all other terms of implementing Canny being the same) is shown below:

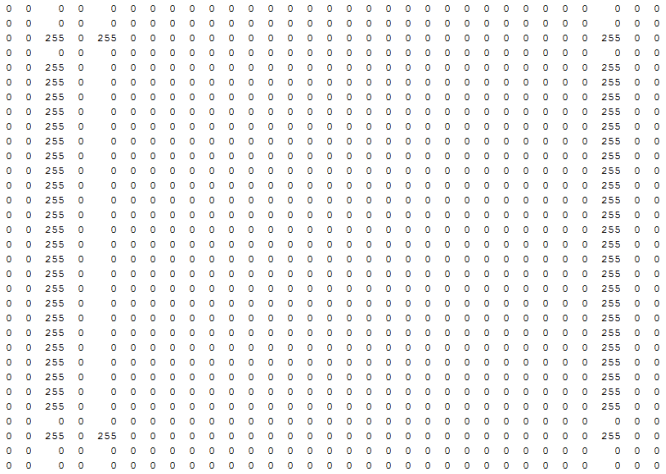


Fig. 4. Contour detected image matrix obtained by applying Listing 2 to execute non-maximum suppression (Source: Scilab Image and Video Processing Toolbox, Scilab Image Processing Design Toolbox)

And the resultant contour detected image is

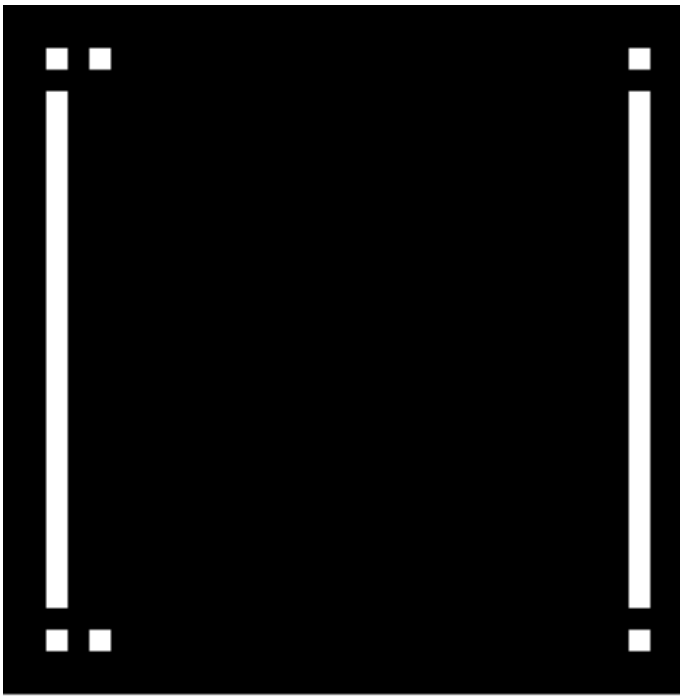


Fig. 5. Resultant contour detected image for matrix in Figure 4 achieved by using the program piece in Listing 2 to execute non-maximum suppression (Source: Scilab Image and Video Processing Toolbox, Scilab Image Processing Design Toolbox)

Thus, the resultant contour detected image demonstrates 50% false nagatives.

For the experiments to be complete, and the conducted analysis and the conclusions arrived at to be reliable, the testing of non-maximum suppression mathematics can go further in terms of:
1) Replacing ‘’>=’’ with ‘’>’’ in Listing 2 only with respect to the left hand neigbouring pixel along the gradient direction, thus leaving ‘’>=’’ comparison with the right hand neighbouring pixel unchaged. The results are:

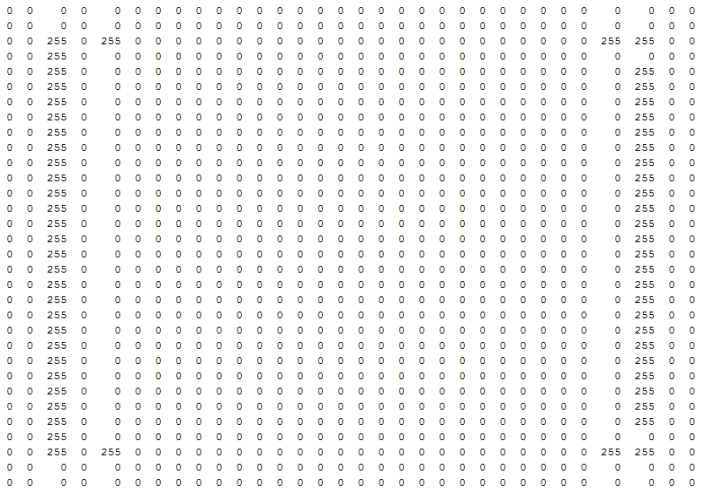


Fig. 6. Contour detected image matrix obtained by applying Listing 2 to execute non-maximum suppression with ‘’>=’’ only with respect to the right hand neighbouring pixel (Source: Scilab Image and Video Processing Toolbox, Scilab Image Processing Design Toolbox)

And the resultant contour detected image is (see Fig. 7):

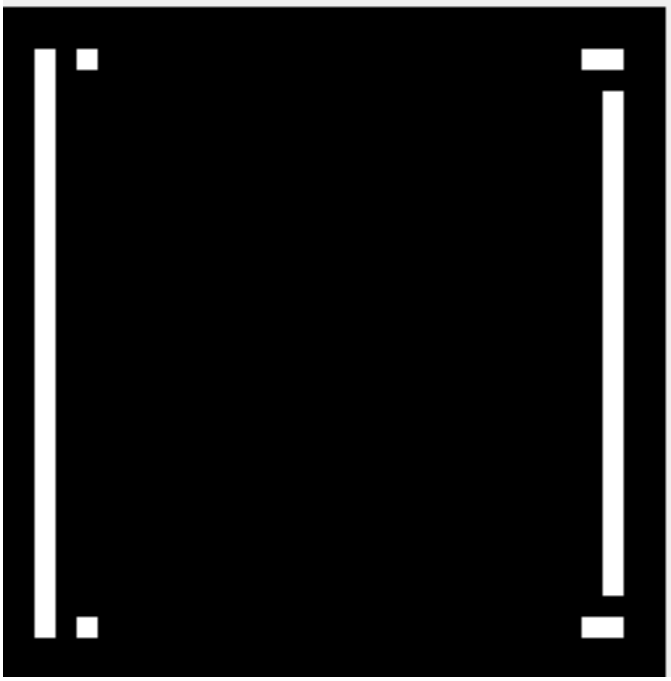


Fig. 7. Resultant contour detected image for the matrix in Figure 6 achieved by using the program piece in Listing 2 to execute non-maximum suppression with ‘’>=’’ only with respect to the right hand neighbouring pixel (Source: Scilab Image and Video Processing Toolbox, Scilab Image Processing Design Toolbox)

The resultant contour detected image demonstrates 46% false nagatives. 2) Replacing ‘’>=’’ with ‘’>’’ in Listing 2 only with respect to the right hand neigbouring pixel along the gradient direction, thus leaving ‘’>=’’ comparison with the left hand neighbouring pixel unchaged. The obtained image matrix is (see Fig. 8):

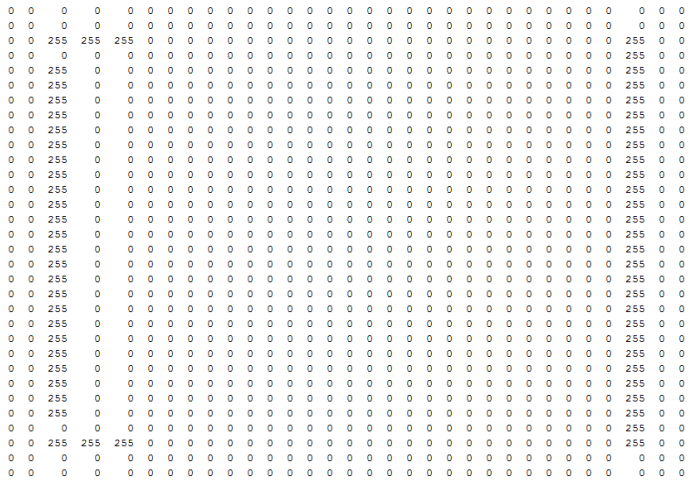


Fig. 8. Contour detected image matrix obtained by applying Listing 2. to execute non-maximum suppression with ‘’>=’ only with respect to the left hand neighbouring pixel (Source: Scilab Image and Video Processing Toolbox, Scilab Image Processing Design Toolbox)

And the resultant contour detected image is (see Fig. 9):

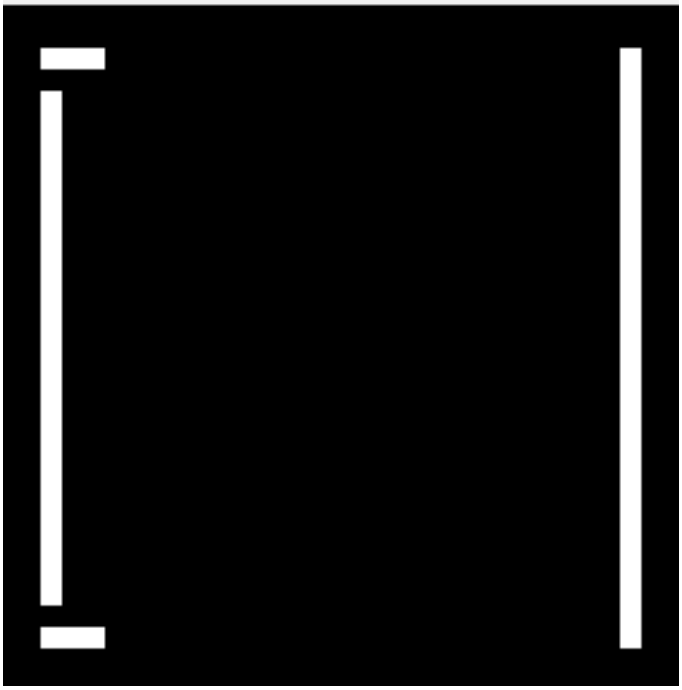


Fig. 9. Resultant contour detected image for matrix in Figure 8 achieved by using Listing 2. to execute non-maximum suppression with ‘’>=’’ only with respect to the left hand neighbouring pixel (Source: Scilab Image and Video Processing Toolbox, Scilab Image Processing Design Toolbox)

The resultant contour detected image demonstrates 46% false nagatives.

5. Analysis of results

To recapitulate, tests show that there are subtleties in the non-maximum suppression mathematics which are to be reckoned with in view of detected contours’ precision, especially regarding the utilization of FPGA based Canny in demanding applications. It is (1) that is the exact mathematics to be employed in the non-maximum suppression module. It implements the very essence of selecting only the local maxima at the expense of the two neighbouring pixes along the gradient direction. Yet, comprehensively exploring all the aspects of Canny’s computational functionality requires that it should be pointed out that (1) is far from guaranteeing reliable results in certain cases.

6. Conclusions

In this paper, presented is an analysis of the non-maximum suppression mathematics in FPGA based Canny. Explored are two main versions of local maxima computations with respect to their impact on detected contours’ precision. The conducted tests and the assessment of achieved results prove that non-maximum suppression calculations are characterized by subtleties which are crucial to the mapped contours’ quality.

References

[1]. D. Sangeetha P. Deepa, FPGA implementation of cost-effective robust Canny edge detection algorithm, Journal of Real-Time Image Processing (JRTIP), Vol. 12, (1), 2016, pp. 1-14
[2]. Fangxin Peng, Xiaofeng Lu, Hengli Lu, Sumin Shen, An improved high-speed canny edge detection algorithm and its implementation on FPGA, Proceedings of SPIE, Fourth International

Conference on Machine Vision (ICMV 2011): Computer Vision and Image Analysis; Pattern Recognition and Basic Technologies, Conference Volume 8350, 2012
[3]. Hao, Geng, Luo Min, and Hu Feng, Improved Self-Adaptive Edge Detection Method Based on Canny, 5th International Conference on Intelligent Human Machine Systems and Cybernetics (IHMSC), Vol. 2, 2013, pp. 527-530
[4]. Heena S. Shaikh, Narayan V. Marathe, Implementation Of Distributed Canny Edge Detection Technique, International Research Journal of Engineering and Technology (IRJET), Volume: 04 (05), 2017, pp. 2926-2928
[5]. T. Rupalatha, C.Leelamohan, M.Sreelakshmi, Implementation of distributed Canny edge detection on FPGA. International Journal of Innovative Research in Science, Engineering and Technology, Vol. 2, (7) 2013, pp. 2618-2626
[6]. Veeranagoudapatil, Chitra Prabhu (2015). Distributed Canny Edge Detector: Algorithm & FPGA Implementation, International Journal for Research in Applied Science & Engineering Technology (IJRASET), Vol. 3 (5), pp. 586-588

For contacts:
Ass. Prof. Dimitre Zh. Kromichev
Department of Marketing and International Economic Relations
University of Plovdiv
E-mail: dkromichev@yahoo.com

Bulgaria Communications Chapter

СРАВНЯВАНЕ НА АЛГОРИТМИТЕ ЗА ДЕКОМПОЗИЦИЯ

Матьо Ст. Динев, Валентина Ст. Кукенска

Резюме: В настоящия доклад са разгледани алгоритми за декомпозиция на обекти, представени чрез графови модели. Показани са резултати от изследване на последователни и итерационни алгоритми. Направено е сравнение на алгоритмите по брой на използваните операции и по бързодействие.
Ключови думи: алгоритъм, граф, декомпозиция

Comparison of decomposition algorithms

Matyo St. Dinev, Valentina St. Kukenska

Abstract: This paper deals with algorithms for decomposition of objects, visualised through graph models. It presents the results of a study into sequential and itinerary algorithms. A comparison of the algorithms is made regarding their speed of action and number of operations used.
Keywords: algorithm, graph, decomposition

1. Въведение

Една от основните задачи, чието решение се налага при моделиране на обекти, е задачата за декомпозиция. Като обекти в настоящия доклад се разглеждат електронни схеми и системи. Декомпозицията на схемите е оптимизационна задача, чието решение позволява отделянето на подсхеми.

Всяка схема (система) се състои от определен брой елементи и връзки между тях. Структурата на дадена схема (система) може да се представи чрез неориентиран граф G(V,E) на базата на следните две аналогии:

1) множеството на върховете $V=(v_1,v_2,..,v_n)$ на графа се съпоставя с множеството на елементите на схемата, а множеството на ребрата $E=(e_1,e_2,\ldots,e_n)$ – с множеството връзките между тях;
2) множеството на върховете $P=(p_1,p_2,\ldots p_n)$ на графа се съпоставя на множеството на потенциалите на схемата, а множеството на ребрата $A=(a_1,a_2,\ldots a_n)$ – на множеството на елементите на схемата.

Чрез предложените аналогии задачата за структурна декомпозиция на дадената схема е приведена към задача за разделяне на топологичния граф G(V,E) на подграфи $G_i(V_i,E_i)$ [2], [4], [5].

Обекти на настоящия доклад са алгоритми за декомпозиция на схеми. Представени са резултати от изследването им. Направено сравнение на алгоритмите по следните критерии:

- брой на използваните операции
- времето за изпълнение.

2. Постановка на задачите

За целта на изследването са формулирани следните задачи:

Постановка на задача 1.

Зададена е принципна схема с определен брой елементи и връзки между тях. Да се раздели схемата на подсхеми при определящ критерий минимален брой външни връзки

между подсхемите. Като допълнително условие се поставя ограничение на броя елементите във всяка подсхема.

На базата на аналогия 1) тази задача е приведена към задача за разделяне на топологичния граф G(V,E) на подграфи. Определящ критерий е минимален брой външни връзки (ребра) за подграфите. Допълнително условие е броят на върховете във всеки подграф да бъде по-малък или равен на m_{max}.

Постановка на задача 2.

Зададена е принципна схема с определен брой елементи и възли между тях. Да се раздели схемата на подсхеми при определящ критерий минимален брой възли между подсхемите. Като допълнително условие се поставя ограничение на броя възли във всяка подсхема.

На базата на аналогия 2) тази задача е приведена към задача за разделяне на топологичния граф G(V,E) на подграфи. Паралелно свързаните елементи в схемата се представят с едно ребро. Определящ критерий е минимален брой външни върхове за подграфите. Допълнително условие е броят на върховете във всеки подграф да бъде по-малък или равен на m_{max}.

3. Алгоритми

За решаването на поставените задачи са използвани четири алгоритъма. Два от тях, алгоритмите с добавяне и отделяне, са представени в [3]. Останалите два са алгоритмите с припокриване и неприпокриване.

Алгоритъм с припокриване

При изпълнението на алгоритъма се построява таблица с три колони. В първата колона I_k се записва избраният връх. Във втората колона $J(I_k)$ се записва списък на съседните върхове на I_k. В третата колона $|J(I_k)|$ се посочва тегловният приоритет на избрания връх. Той се използва като критерий за определяне на върховете в отделните подграфи. Броят на тези върхове не е зададен предварително.

1. Избира се връх с най-малка степен на свързаност и се записва в I_k;
2. В $J(I_k)$ се записват съседните върхове на I_k;
3. На всеки връх в $J(I_k)$ се определя броя на върховете съседни на върховете на подграфа, изграден от подграфа I_k и върха j. Това число се записва като степен към съответния връх.
4. В $|J(I_k)|$ се записва тегловния приоритет на върха в I_k
5. От върховете записани в $J(I_k)$ се избира връх с най-малка степен и се записва в I_k;
6. Проверява се дали са записани всички върхове на графа в I_k. Ако да - преход към стъпка 7, ако не - преход към стъпка 2;
7. Определят се отделните подграфи;
8. Описват се общите върхове за всеки подграф;
9. Край.

Алгоритъм с неприпокриване

При изпълнението на алгоритъма се построява таблица с три колони. В първата колона IS се записва избрания връх *k*. Във втората колона AS се записва списък на съседните върхове на IS. В третата колона CN се посочва тегловния приоритет на избрания връх.

1. Избира се начален итерационен връх k_1. Записва се неговият номер в IS:

$$IS(1) = k_1 \tag{1}$$

2. Записват се в AS(1) номерата на съседните върхове на k_1.
3. Записва се в CN(1) тегловния приоритет на върха k_1:

$$CN(1) = |AS(1)| \quad (2)$$

4. Ако тегловния приоритет е 0 т.е. $CN(i) = 0$, то преход към стъпка 9. Ако не е 0 продължававме със стъпка 5.
5. От $AS(i)$ се избира следващия итерационен връх k_{i+1} и се записва в IS:

$$IS(i+1) = k_{i+1} \quad (3)$$

6. В $AS(i+1)$ се записват върховете $AS(i) - k_{i+1}$ и се обединяват със съседите на k_{i+1}, които не са включени във IS.
7. В *CN(i+1)* се записва тегловния приоритет на избрания връх по следната формула:

$$CN(i+1) = |AS(i+1)| \quad (4)$$

8. Увеличаваме *i* с единица и преминаавме към стъпка 4.
9. Извеждане на получените резултати
10. Край

Най-често в качеството на начален итерационен връх се избира този, който има най-малък тегловен приоритет. За следващ итерационен връх от масива AS(i) се избира върхът, за който следващият ред на *AS* ще има минимален брой върхове $|AS(i+1)|=min$.

При определяне на даден подграф се използва тегловния приоритет (CN). Търси се *min CN(i),* при което $\alpha m_{max} \leq i \leq m_{max}$, където $\alpha=0.6 \div 0.8$, а m_{max} е максималният брой върхове за един подграф, т.е. разглеждат се последните стойности на тегловия приоритет (CN) и разделятно се извършва когато CN(i) приема минимална стойност.

4. Резултати и изводи от изследането

За целта на изследването е разработено програмно приложение с потребителски интерфейс. То е реализирано на Borland Delphi 7. Като входни данни за приложението се използват списъчно и матрично представяне на графови модели. Работата с приложението е показана в [1]. Чрез него са решени примерни задачи за декомпозиция на графови модели. Приложението изчислява необходимия брой операции за всеки алгоритъм, както и времето за неговото изпълнение. Резултатите от изследванията са отразени в таблици 1 и 2.

Алгоритмите с добавяне и отделяне решават задача 1. Извършени са изследвания върху графови модели с брой на върховете от 14 до 600. Получените резултати са отразени в таблица 1.

Таблица 1. Резултати за алгоритмите с добавяне и отделяне

Брой върхове	**Алгоритми**							
	С добавяне				**С отделяне**			
	N	**T (*ms*)**	**n**	**K**	**N**	**T (*ms*)**	**n**	**K**
14	563	536	2	2	373	354	2	2
22	2187	1785	2	2	1029	968	2	3
63	11465	9865	5	7	5935	4523	4	9
100	49377	45678	10	11	11543	9654	9	13
600	294206	258369	20	25	66365	55645	18	28

В първата колона от таблицата е посочен броят на върховете за съответния граф. Означението на буквите е следното:

N – брой на използваните операции за съответния алгоритъм;

T – време за изпълнение на съответния алгоритъм в ms;

n – брой на отделните подграфи след прилагането на алгоритъма;

K – брой на външните връзки;

От резултатите, представени в таблица 1, могат да се направят следните изводи:

- При използване на алгоритъма с добавяне, с нарастване броя на върховете в графа многократно нараства и броят на използваните операции.
- Алгоритъмът с отделяне е по-бърз от алгоритъма с добавяне.
- Алгоритъмът с добавяне дава по-добро решение по поставения критерий.

За решаването на задача 2 се използват алгоритмите с припокриване и неприпокриване. Извършени са изследвания върху графови модели с брой на върховете от 14 до 600. Получените резултати са отразени в таблица 2.

Таблица 2. Резултати за алгоритмите с неприпокриване и припокриване

Брой върхове	**Алгоритми**							
	С неприпокриване				**С припокриване**			
	N	**T (*ms*)**	**n**	**F**	**N**	**T (*ms*)**	**n**	**F**
14	865	895	2	2	1269	1303	2	2
22	2718	3067	2	2	2536	2790	2	2
63	14256	15656	7	7	8865	9459	7	6
100	56561	65634	10	9	17648	19034	10	9
600	345363	389468	20	19	85698	95636	20	18

В първата колона от таблицата е посочен броят на върховете за съответния граф. Означението на буквите е следното:

N – брой на използваните операции за съответния алгоритъм;

T – време за изпълнение на съответния алгоритъм в ms;

n – брой на отделните подграфи след прилагането на алгоритъма;

F – брой на граничните върхове.

От резултатите представени в таблица 2 могат да се направят следните изводи:

- При използване на алгоритъма с неприпокриване, с нарастването на броя на върховете в графа многократно нараства и броят на операциите
- При по-голям брой на върхове в графа значително нараства и времето за изпълнение на алгоритъма с неприпокриване, отколкото при алгоритъма с припокриване
- При декомпозиция на графови модели с голям брой на върховете е по-удачно да се използва алгоритъмът с припокриване, а при по-малък брой на върховете – алгоритъмът с неприпокриване.
- Алгоритъмът с припокриване дава по–добро решение по поставения критерий.

5. Заключение

В този доклад са разгледани част от алгоритмите за декомпозиция на схеми, представени с графови модели. Избрани алгоритми са тествани с програмно приложение. За всеки от тях е определен броят на необходимите операции и времето за изпълнение.

Получените резултати от изследването на алгоритмите могат да се използват при избор на алгоритъм.

Литература

[1]. Динев М., В. Кукенска, Програмно приложение за структурна декомпозиция, Международна научна конференция "Техника, Технологии, Образование" - ICTTE 2017, Ямбол, 19-20 октомври 2017, стр. 190-197, ISSN 1314-9474.

[2]. Манев К. (2013). Алгоритми в графи. Основни алгоритми, Изкуства, 2013.

[3]. Dinev M., V.Kukenska, Matlab application for decomposition of graphs, XII International Conference Strategy of Quality in Industry and Enducation, Bulgaria, Varna, May 30 - June 2, 2016, pp 537-542, ISBN 978-966-2752-71-7

[4]. Golumbic M., Algorithmic Graph Theory and Perfect Graphs Second Edition, Academic Press, 2004.

[5]. Gould R.,C. Goodrich, Graph Theory, Department of Mathematics and Computer Science Emory University, 2012

За контакти:
Инж. Матьо Стефанов Динев
катедра „Компютърни системи и технологии”
Технически университет - Габрово
E-mail: mat@abv.bg

ИЗВЛИЧАНЕ НА ОПИСАТЕЛИ ОТ ФИЗИОЛОГИЧНИ СИГНАЛИ ЗА РАЗПОЗНАВАНЕ НА ЕМОЦИИ И СТРЕС

Калин Б. Калинков, Валентина И. Маркова, Тодор Д. Ганчев

Резюме: Настоящият доклад представя резултати от изследвания, свързани с извличане на описатели от физиологични сигнали за разпознаване на емоции и стрес. Разгледани са множество статистически описатели, изчислени от ЕКГ и фотоплетизмографски (ФПГ) сигнали. Извършен е сравнителен анализ на описателите, получени в среда на Matlab и LabView, при различна дължина на записа. Резултатите сочат за възможна взаимозаменяемост между параметрите, извлечени от ЕКГ и ФПГ. Отчетено е силно влияние на дължината на записа върху оценката на изменчивостта на сърдечния пулс в честотната област.
Ключови думи: Биомедицински сигнали, описатели, ЕКГ, пулс, вариантност на сърдечен ритъм, Matlab, LabVIEW, Consensys

Feature extraction for emotion and stress recognition

Kalin B. Kalinkov, Valentina I. Markova, Todor D. Ganchev

Abstract: The current paper presents experimental results related to feature extraction from physiological signals for emotion and stress recognition. Multiple statistical features were calculated from ECG and PPG signals. Features from different record lengths were extracted in Matlab and LabView, on which a comparative analysis was conducted. The results indicate possibility for interchangeability of ECG and PPG parameters. The record's length strongly influences the appraisement of the heart rate variability, observed in the frequency domain.
Keywords: Biomedical signals, features, ECG, pulse, heart rate variability, Matlab, Labview, Consensys

1. Увод

Всеки човек реагира по различен начин на значими събития в живота си. Емоцията е реакция на организма по отношение на приятно и неприятно усещане. Много често негативните емоции и стресови състояния водят до заболявания, които имат психосоматичен характер. Нараства броят на изследванията, свързани с разпознаване на стресови и емоционални състояния чрез обработка и анализ на биомедицински сигнали. Известни са голям брой описатели както във времева, така и в честотна област, като само част от тях са съществени за разпознаването на емоции и стрес.

Основни биомедицински сигнали, използвани при изследвания от подобен тип, са Електрокардиограмата (ЕКГ), Фотоплетизмограмата (ФПГ) и Кожно-Галваничната Реакция (КГР), Кръвно налягане, Респираторна дейност, Електроенцефалограма (ЕЕГ).

Информационната част на ЕКГ, аналогично на ФПГ, се съдържа в параметрите на QRS комплексите и в частност на разстоянието между два R максимума (пика). Тези разстояния, наречени още R-R интервали, са основният изходен параметър при извличане на описатели от биомедицински запис на сърдечната дейност.

Кожно-Галваничната Реакция се базира на измененията в проводимостта на повърхностния слой на кожата. От спецификите на КГР произлиза необходимостта от

разделяне на сигнала на двете му основни компоненти, за да може да се обработва полезната фазична реакция.

В следствие на изследванията в областта са изградени бази от данни, като ASCERTAIN [1], DEAP [2], MAHNOB [3] и DECAF [4]. Различията се коренят в целите на изследванията и от там - в броя на стимулите, тяхната продължителност, емоционална натовареност и други. В създаването на ASCERTAIN вземат участие 58 участника, които се подлагат на 36 видео стимула с продължителност от 51 до 128 секунди. Изследват се ЕКГ, КГР, ЕЕГ и лицеви изражения. В DEAP от друга страна вземат участие 32 доброволци, изложени на 120 видео стимула, всеки от които с продължителност от 1 минута. При DEAP се изследват по-малък брой биомедицински сигнали – 32 канално ЕЕГ, КГР, Респираторна дейност, Кожна температура, ЕКГ, ФПГ, Електромиограма, Електроокулограма.

В разнообразието от бази данни и изследвания се използват различни прагови стойности за извличане на описатели. Характерни амплитудни прагови стойности, съответно за детекция на R пикове при ЕКГ е 0.2V[5], а за детекция на пикове в SCR при GSR е 0.01μS [6]. Резултатите от изследванията сочат като препоръчителна продължителност на стимулите между 2 и 5 минути [7].

Текущото изследване цели изграждането на база от биомедицински сигнали с помощта на безжични блутут сензори – Consensys Development Kit, да се имплементират алгоритми за извличане на описатели, за изследване на емоции и стрес. Предвид ограниченията на системата за събиране на биомедицински сигнали, базата от данни включва записи на ЕКГ (електрокардиограма/ECG), Плетизмограма (ФПГ/пулс) и Кожно-Галванична реакция (КГР).

След обработката на сигналите се извършва сравнение между ЕКГ описателите, извлечени от Matlab приложение и аналогично такова в LabVIEW, разработено на по-ранен етап. Изследва се взаимозаменяемостта на описателите от ЕКГ и ФПГ, от там и взаимозаменяемостта на самите биосигнали (ЕКГ и ФПГ). Проверява се влиянието на дължината на записа върху описателите от ЕКГ сигнал и оценката на изменчивостта на сърдечния ритъм (HRV/Heart Rate Variability).

2. Системи за събиране и анализ на биомедицински сигнали

Първата част от системата е обвързана със създаване на база данни от биомедицински сигнали. Участниците в изследването се подлагат на аудио-визуална стимулация. Емоционалните стимули представляват музикални клипове, с различна продължителност, емоционален заряд и интензивност. Безжичните модули препращат сигналите от електродите, поставени на участника, към компютър, събиращ данните.

За извършването на записа се използва “Consensys Development Kit”, състоящ се от безжични (блутут) сензори на “Shimmer” и софтуерното приложение “Consensys Pro”. Системата осигурява свобода и комфорт на участника, които са ограничени в изцяло стационарните системи. Приложението Consensys Pro извършва паралелен запис с множество сензори и автоматична синхронизация, като по време на изследването координира модулите ECG и GSR+.

В изследването ЕКГ сигналът се записва с честота на дискретизация от 512 Hz. Електродите са поставени по схемата: сигнални електроди - лява ръка – дясна ръка и неутрален електрод на дясна китка.

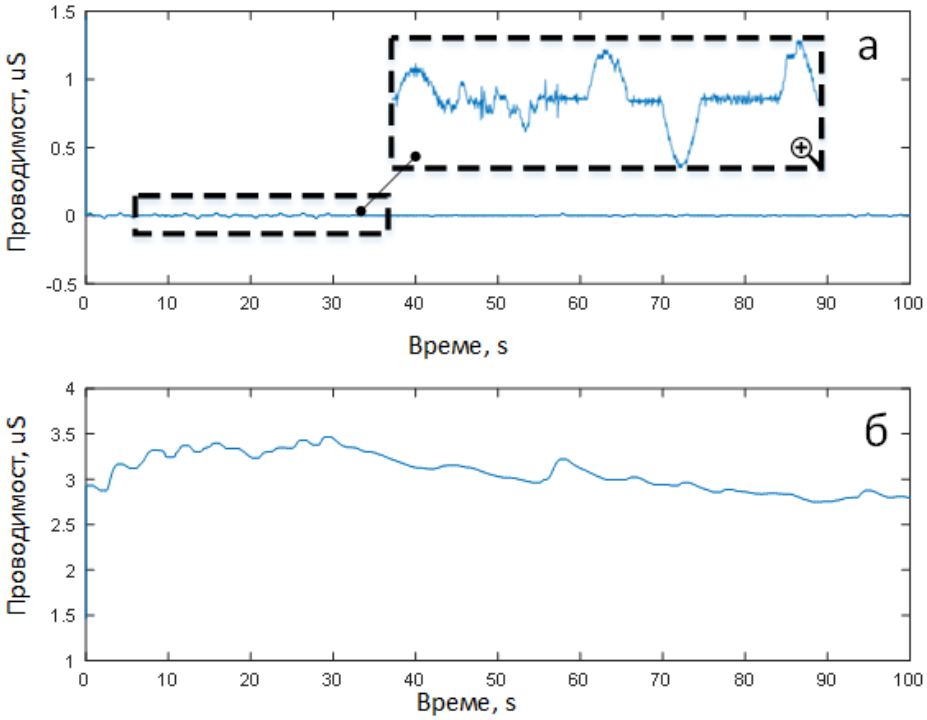
Кожно – Галваничната Реакция и пулсът се записват от GSR+ модула. Записът на двата сигнала става паралелно, с честота на дискретизация от 128 Hz, което трябва да се отчете при по-нататъшните изследвания. Електродите се следват схемата: лява ръка – положителен и отрицателен електрод на GSR, поставени на втори и четвърти пръст, сензор за пулс – трети пръст.

Втората част на системата отговаря за обработка на сигналите и извличане на описателите. Извършва се филтрация за премахване на постоянна съставна и на смущения от електрическата мрежа.

За изследване на КГР се извършва разделяне на сигнала на два компонента:

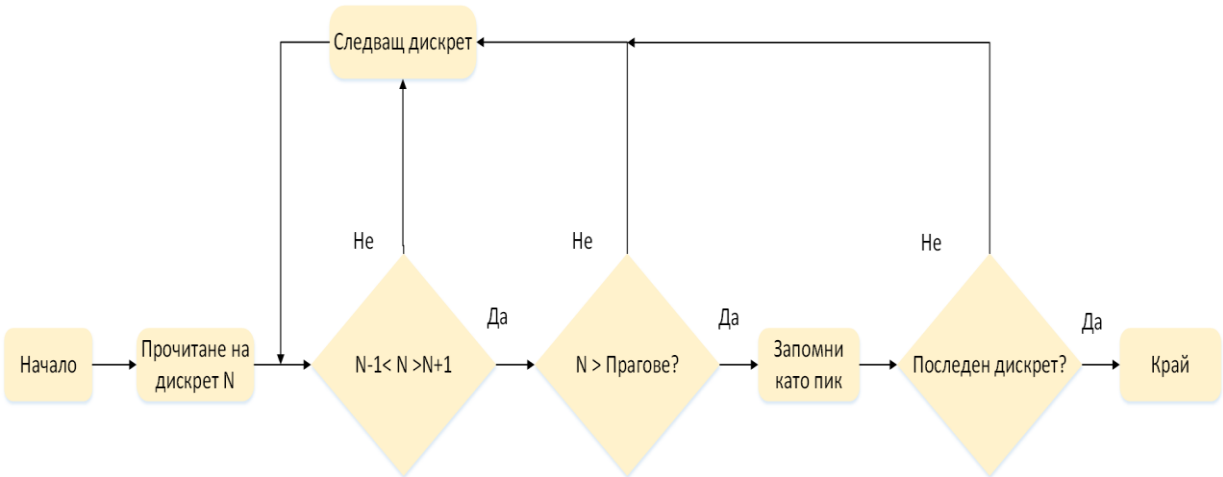
1. Допълнителна филтрация на сигнала с медианен филтър за извличане на тоничното ниво (Skin Conductance Level – SCL).
2. Изваждане на тоничното ниво от оригиналния сигнал, остатък при което е фазичната реакция (Skin Conductance Response – SCR).

На фигура 1 са показани разделените компоненти на Кожно-Галваничната Реакция.

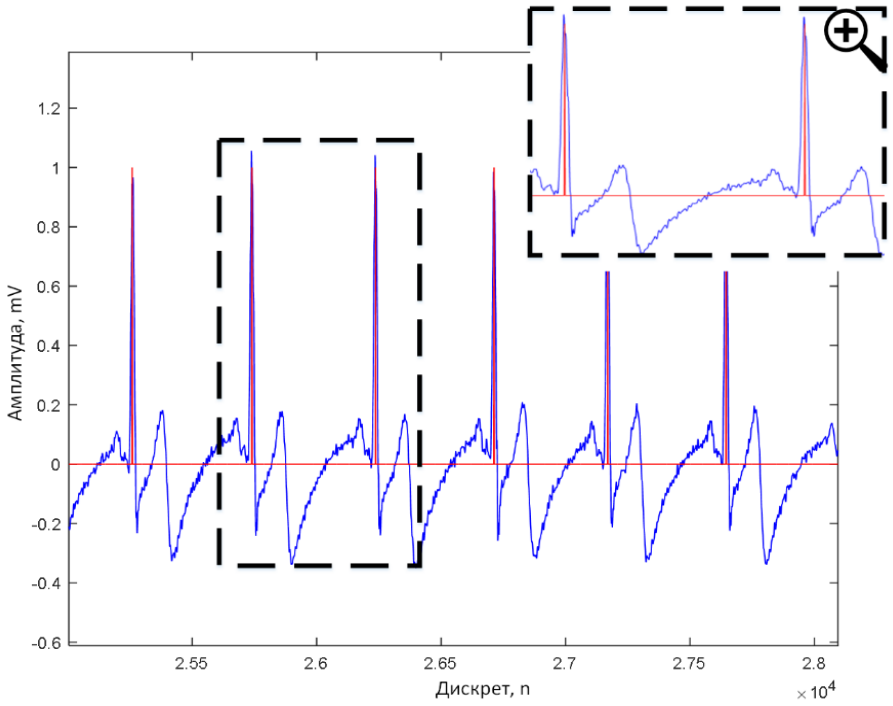


Фиг. 1. Компоненти на GSR сигнал – (а) Фазична реакция и (б) Тонично ниво

Използваният алгоритъм за детекция на пикове е идентичен и при трите биомедицински сигнала и блоковата му схема е показана на фигура 2. При него всеки дискрет се сравнява по амплитуда с предходния и следващия, като най-голямата амплитуда се отчита като пик, при условие че отговаря на амплитудната и времева прагови стойност. На фигура 3 е показана примерна детекция на пикове при ЕКГ сигнал.



Фиг. 2. Алгоритъм за детекция на пикове в биомедицински сигнали



Фиг. 3. Детекция на R пикове в QRS комплексите на ЕКГ сигнал

В таблица 1 и 2 са поместени извлечените описатели от ЕКГ/ФПГ и Кожно-Галванична Реакция.

Таблица 1. Описатели от ЕКГ и ФПГ

№	**Описател**
1	Среден сърдечен ритъм
2	Вариация на сърдечния ритъм
3	Мощност в Много-ниско честотна лента (0-0.04) Hz
4	Мощност в Ниско честотна лента (0.04-0.15) Hz
5	Мощност във Високо честотна лента (0.15-0.4) Hz
6	Мощност в Ниско честотна лента (нормирани единици)
7	Мощност във Високо честотна лента (нормирани единици)
8	Мощност в Много-ниско честотна лента (%)
9	Мощност в Ниско честотна лента (%)
10	Мощност във Високо честотна лента (%)
11	Среден интервал между ударите (Среден RR интервал)
12	Среден NN интервал
13	Максимална дължина на NN интервал
14	Минимална дължина на NN интервал
15	Процент от последователни NN интервали, с разлика над 50ms
16	Стандартно отклонение на NN интервалите
17	Средно квадратично на разликата на последователните NN интервали
18	Стандартно отклонение на RR интервалите

Таблица 2. Описатели от КГР

№	**Описател**
1	Общ брой пикове
2	Средна проводимост за целия запис
3	Максимална амплитуда на пиковете
4	Минимална амплитуда на пиковете
5	Средна проводимост на пиковете
6	RMS на проводимосттан на пиковете
7	Стандартно отклонение на пиковете (СО)
8	Средна абсолютна стойност на пиковете
9	Средно съпротивление за целия запис
10	Асиметрия на пиковете
11	Ексцес на пиковете
12	Разлика от Първи Ред (РПР)
13	Разлика от Втори Ред (РВР)
14	Отношение на РПР и СО
15	Отношение на РВР и СО
16	Мощност в лентата 0 - 2.4Hz

3. Резултати от изследванията

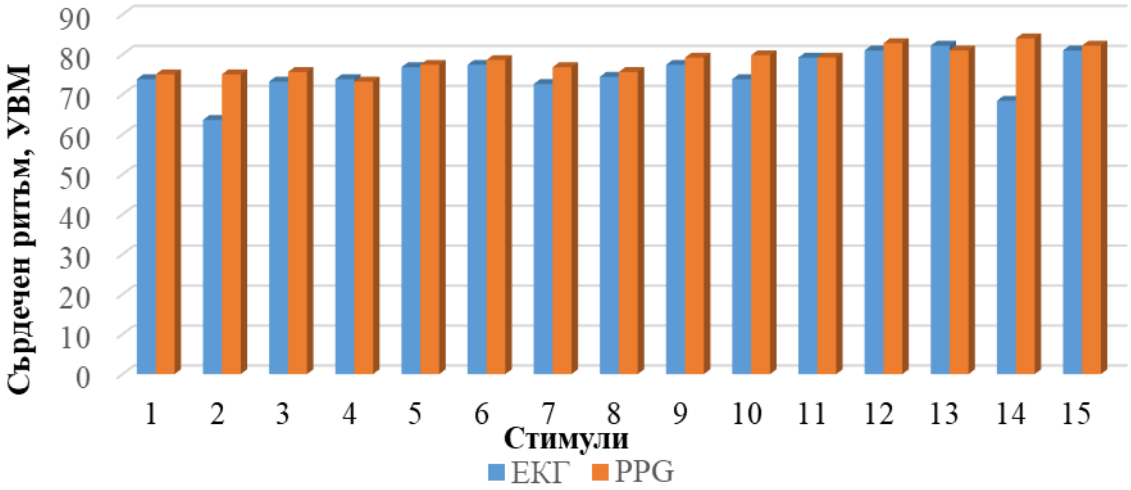
Към настоящия момент в изследването и създаването на базата данни участват четирима доброволци. Участниците са представители на различни възрастови, социални и полови групи. Всички от тях се подлагат на един неутрален стимул за установяване на нормалните за индивида стойности на биосигналите и 15 аудио/визуални стимула, пораждащи емоционално въздействие, следователно и динамика в параметрите на наблюдаваните сигнали.

3.1. Сравнение на описатели, извлечени от Електрокардиограма и Фотоплетизмограма

Основен параметър за сравнение на измененията в сърдечната честота е пулсът, измерен в удари за минута (УВМ). Описателят за сърдечен ритъм се определя по уравнение (1). На фигура 4 е показано сравнението за един от участниците.

$$BPM = \frac{N}{L}.60s, \text{ където} \quad (1)$$

BPM е среден сърдечен ритъм измерен в УВМ, N е общ брой R пикове и L е продължителността на записа в секунди.



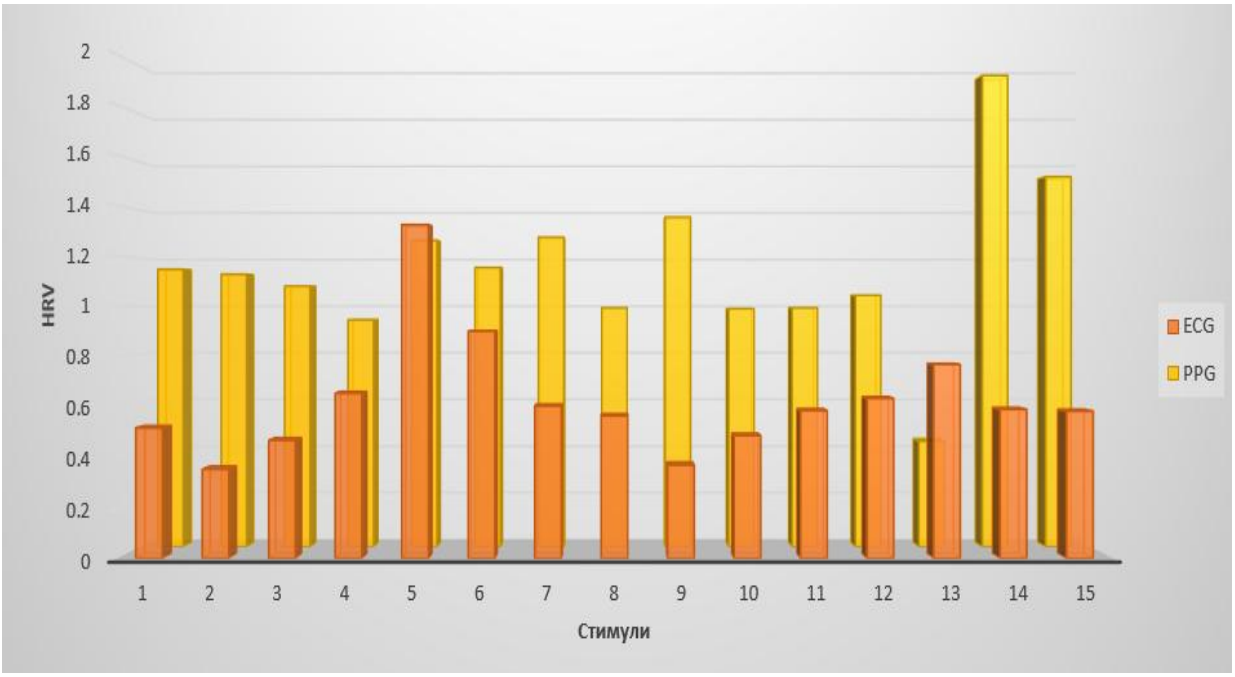
Фиг. 4. Сравнение на сърдечен ритъм при ЕКГ и PPG

Друг основен параметър, описващ сърдечната дейност и връзката ѝ със стреса, е изменчивостта на сърдечния ритъм (ИСР/Heart Rate Variability – HRV). Тя се определя като отношение, посочено в математически израз 2.

$$HRV = \frac{LF}{HF} \ , \tag{2}$$

където HRV е вариантността на сърдечната дейност, LF – спектър в лента 0.04 – 0.15 Hz, а HF – спектър в лента 0.15 – 0.4Hz.

На фигура 5 е извършено сравнение на ИСР, извлечено от ЕКГ и ФПГ.



Фиг. 5. Изменчивст на сърдечния ритъм (HRV) за един от участниците

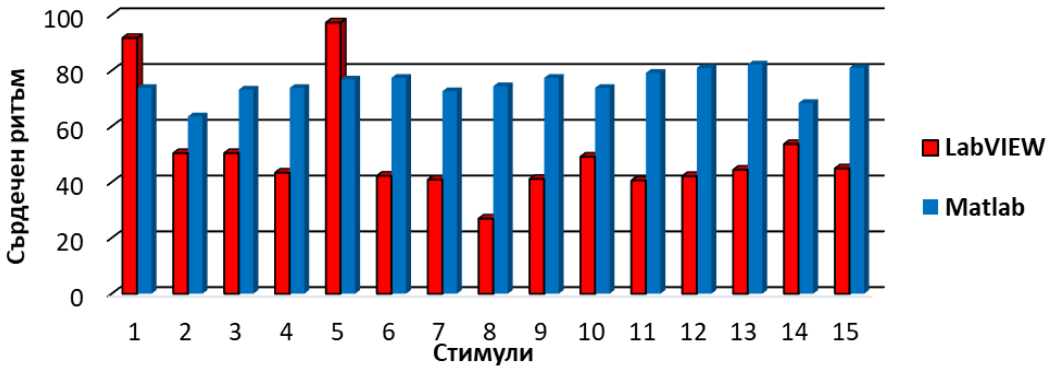
На фигура 5 се наблюдава сравнително по-висока стойност за ИСР при ФПГ спрямо ЕКГ. Аналогични сравнения се правят и за останалите описатели, като в таблица 3 е показана средната им разлика.

Таблица 3. Средна разлика за описателите от ЕКГ и PPG

Описател	Средна разлика (%)
Сърдечен ритъм	0.09
HRV	0.43
Спектър в VLF лентата	0.01
Спектър в LF лентата	0.21
Спектър в HF лентата	0.01
Спектър в LF лентата (н.е.)	0.25
Спектър в HF лентата (н.е.)	0.15
Спектър в VLF лентата (%)	0.09
Спектър в LF лентата (%)	0.11
Спектър в HF лентата (%)	0.20
Средна продължителност на RR интервалите	0.06
Средна продължителност на NN интервалите	0.09
Най-дълъг NN интервал	0.23
Най-къс NN интервал	0.25
pNN50	2.02
SDNN	1.08
RMSSD	1.62
SD	0.01

3.2. Сравнение на ЕКГ описатели, извлечени от Matlab и LabView

Сравнителният анализ между Matlab и софтуерно решение, разработено на по-ранен етап с LabView, показва интересни резултати. На фигура 6 е илюстрирано сравнение на стойностите на сърдечен ритъм за различни стимули, изчислени чрез двете програмни среди.



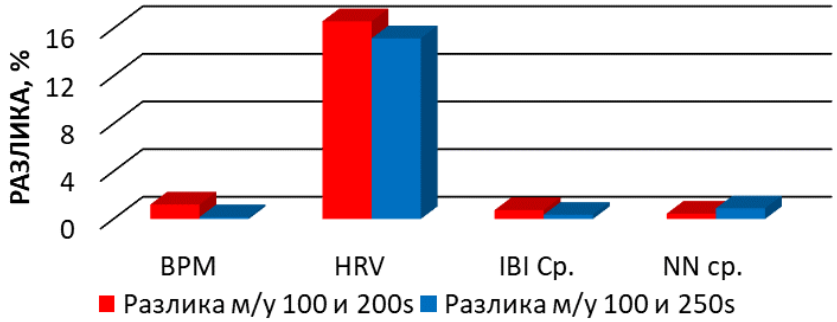
Фиг. 6. Сърдечен ритъм при анализ с LabView и Matlab

От фигура 6 става ясно, че има съществено разминаване между резултати, получени чрез двата алгоритъма за отчитане на RR интервалите. За определени стимули разликите в стойности на пулса са два пъти. Аналогична е тенденцията при останалите описатели, поради тясната им свързаност със сърдечния ритъм.

3.3. Изследване продължителността на записа при ЕКГ и ФПГ

На фигура 7 се вижда, че и при четирите изследвани описателя е налична разлика, като най-съществената тя е при ИСР. Разликите между продължителност 200s и 250s е малка, което показва, че дължина от 200s е по-оптимална продължителност за едно

изследване. Намаляване на продължителността на един стимул с 50s води до скъсяване на времето за изследване. От там следва намаляване влиянието на умората и се осигурява по-голяма тежест на стимулите.



Фиг. 7. Разлика в четири освнони описатели при записи с различна продължителност

4. Заключение

За целите на изследването е изградена база данни от биомедицински сигнали. Експерименталните резултати сочат за възможна взаимозаменяемост на ЕКГ и ФПГ. Сравнителният анализ на описатели от Matlab и LabView показва по-висока достоверност на описателите, извлечени от Matlab. Дължината на записа се явява значителен фактор за определянето на изменчивостта на сърдечния ритъм (Heart Rate Variability), като 200s се явява предпочитан времеви диапазон за бъдещи изследвания.

Резултатите, макар и оптимистични, не са категорични. Необходимо е провеждането на разширено изследване, при по-голяма база от данни, за да се изкаже категорично твърдение за получените, в хода на изследването, резултати.

Литература

[1]. Subramanian R., J. Wache, M. Abadi, R.Vieriu, S. Winkler, N. Sebe. ASCERTAIN:Emotion and Personality Recognition using Commercial Sensors, IEEE Trans. of Affective Computing, August 2016

[2]. Koelstra S., C. Muhl, M. Soleymani, J. Lee, A. Yazdan, T. Ebrahimi, T. Pun, A. Nijholt, I. Patras. DEAP: A Dataabse for Emotion Analysis Using Physiological Responses, IEEE Trans. of Affective Computing, vol.3, no.1, 2012

[3]. Soleymani M., J. Lichtenauer, T. Pun, M. Pantic. A multimodal database for affect recognition and implicit tagging, IEEE Trans. of Affective Computing, vol. 3, 2012

[4]. Abadi M., R. Subramanian, S. Kia, P. Avesani, I. Patras, N. Sebe. DECAF: MEG-based Multimodal Database for Decoding Affective Physiological Responses, IEEE Trans. of Affective Computing, vol.6, no.3, 2015

[5]. GSR – Pocket Guide, iMotions, 2016

[6]. Soleymani M., F. Dixon, T. Pun, G. Chanel. Toolbox for Emotional features extraction from physiological signals (TEAP), Front. ICT, 13 February 2017

[7]. Shaffer F., J. Ginsberg, An Overview of Heart Rate Variability Metrics and Norms, Front Public Health, 2017

За контакти:
инж. Калин Б. Калинков
катедра „Комуникационна техника и технологии”
Технически университет - Варна
E-mail: kalin.kalinkov@mail.bg

РЕКУРСИВНО ОЦЕНЯВАНЕ НА РЕАЛНИТЕ ПАРАМЕТРИ НА ПОСТОЯННО ТОКОВ ДВИГАТЕЛ

Иван В. Григоров, Наско Р. Атанасов

Резюме: Адаптивното управление обхваща набор от методи, които осигуряват систематичен подход за автоматично управление в реално време. Рекурсивните методи за оценяване на параметри трябва да удовлетворяват изискванията към алгоритмите за идентификация в реално време. Това се обуславя от факта, че корекцията на модела след постъпване на данните от новото наблюдение, както и изработването на ново управляващо въздействие, трябва да станат за един такт на дискретизация. В тази статия е предложено прилижение на рекурсивните методи за оценяване на параметри в адаптивна система за управление на постоянно токов двигател посредством самонастройващ се регулатор с минимална дисперсия .
Ключови думи: метод на най-малките квадрати, метод на инструменталната променлива, рекурсивни методи за оценяване на параметри, постоянно токов двигател.

Recursive estimation of the real DC motor parameters

Ivan V. Grigorov, Nasko R. Atanasov

Abstract: The adaptive control covers a set of methods that provide a systematic approach for automatic control in real-time. Recursive methods for parameter estimation have to meet the requirements for identification algorithms in real time. This is determined from the fact that the adjustment of the model after the submission of new data from monitoring, and the development of new control action should be made in a single cycle of discretization. In this article is proposed an application of recursive methods for parameter estimation in adaptive control of DC motor with minimal variance (MV).
Keywords: adaptive system, instrumental variable method, least squares method, recursive methods for parameter estimation, DC motor,

1. Увод

Задължително условие за контрол на една система е разбирането на динамичните характеристики на обекта [3]. Рекурсивните методи за оценяване се използват при идентификация в реално време. По време на работата на обекта, заедно с управлението, може да продължи уточняването на модела на базата на получаваните нови данни. Това е от особено важно значение, когато характеристиките на обекта се изменят във времето. Тези промени се отчитат в изготвянето на управляващото въздействие от така наречените адаптивни системи, често използвани в предаването и обработката на сигнали.

Рекурсивните методи са независим инструмент за контрол в реално време [1].

2. Рекурсивни версии на метода на най-малките квадрати за оценяване на параметри

2.1. Рекурсивен претеглен метод на най-малките квадрати

За да се получат рекурсивни оценки по метода на претеглените най-малките квадрати се използва

$$\hat{\theta}_{N+1} = \hat{\theta}_N + \frac{C_N f_{N+1}}{\frac{1}{w_{N+1}} - f_{N+1}^T C_N f_{N+1}} \left(y_{N+1} - f_{N+1}^T \hat{\theta}_N \right) \tag{1}$$

Според (1) старата оценка $\hat{\theta}_N$ се коригира пропорционално на разликата $y_{N+1} - \hat{y}_{N+1}$ с векторен коефициент на пропорционалност

$$\Gamma_{N+1} = \frac{C_N f_{N+1}}{\frac{1}{w_{N+1}} - f_{N+1}^T C_N f_{N+1}} \tag{2}$$

Величината $\hat{y}_{N+1} = f_{N+1}^T \hat{\theta}_N$ е предсказаната стойност за y_{N+1}, изчислена въз основа на коефициентите от предишната итерация $\hat{\theta}_N$.

Използване на остатъци вместо грешки на предсказване

Във формула (2) участва грешката от предсказване

$$e_{N+1} = y_{N+1} - \hat{y}_{N+1}\left(\hat{\theta}_N\right) = y_{N+1} - f_{N+1}^T \hat{\theta}_N \tag{3}$$

За повишаване на точността на предсказване вместо e_{N+1} се използва остатъка

$$r_{N+1} = y_{N+1} - f_{N+1}^T \hat{\theta}_{N+1} \tag{4}$$

От (2) и (3) следва

$$r_{N+1} - e_{N+1} = -f_{N+1}^T (\hat{\theta}_{N+1} - \hat{\theta}_N) \tag{5}$$

Ако от (2) се определи разликата $(\hat{\theta}_{N+1} - \hat{\theta}_N)$ и се замести в (5), се получава

$$r_{N+1} = \frac{\frac{1}{w_{N+1}} e_{N+1}}{\frac{1}{w_{N+1}} - f_{N+1}^T C_N f_{N+1}} \tag{6}$$

Стартиране на изчислителната процедура

За получаване на оценки по метода на претеглените най-малки квадрати са необходими $\hat{\theta}_N$ и C_N от предишни изчисления. Събират се $N \geq k$ наблюдения и по нерекурсивния метод на претеглените най-малки квадрати се намират начални оценки $\hat{\theta}_N$ и C_N. След това изчисленията продължават по (1) и (7)

$$C_{N+1} = C_N - \frac{C_N f_{N+1} f_{N+1}^T C_{N+1}}{\frac{1}{w_{N+1}} - f_{N+1}^T C_N f_{N+1}} \tag{7}$$

Алгоритъмът на рекурсивния претеглен метод на най-малките квадрати лежи в основата на редица други рекурсивни процедури. Той е следният:

- Събират се $N \geq k$ наблюдения и по нерекурсивния метод на претеглените най-малки квадрати се намират начални оценки $\hat{\theta}_N$ и C_N;
- Новите оценки се изчисляват по формула (1);
- Преизчислява се матрицата C_{N+1} по зависимост (7), за да се подготви за следващата итерация;
- С новите оценки $\hat{\theta}_{N+1}$ и C_{N+1} започва следващата итерация от т. 2 на алгоритъма.

2.2. Рекурсивен обикновен метод на най-малките квадрати

Рекурсивният метод на най-малките квадрати може да се разглежда като частен случай на РПМНМК при $W = I$. Това означава всички тегла да се приемат равни на единица, което е възможно, само ако $\rho = 1$ Поради това рекурсивният метод на най-малките квадрати (РМНК) може да се получи, като в процедурата от т.1.1. се положи навсякъде $\rho = 1$ [1,3].

3. Рекурсивен метод на инструменталната променлива

Процедурата за получаването на оценки по рекурсивния метод на инструменталната променлива е подобна на тази по рекурсивния метод на най-малките квадрати.

Видът на инструменталния вектор v_{N+1} се избира от (8) или (8')

$$v_i^T == \left[-\hat{y}_{i-1} \;\; -\hat{y}_{i-2} \ldots -\hat{y}_{i-n} \;\; u_{i-1} \;\; u_{i-2} \ldots u_{i-m}\right] \tag{8}$$

$$v_i^T = \left[u_{i-1} \;\; u_{i-2} \ldots u_{i-m}\right] \tag{8'}$$

Алгоритъмът за получаване на оценки по рекурсивния метод на инструменталната променлива е следният:

- Оценяването започва по рекурсивния претеглен МНК. Това повишава устойчивостта на началните оценки. След натрупване на определен брой $N \geq k$ наблюдения се преминава към оценяване по рекурсивния метод на инструменталната променлива. Обикновено се избира N да е равно на 3 до 5 пъти броя на коефициентите.
- Оценките на коефициентите на $N+1$-вата итерация се изчисляват по формулата

$$\hat{\theta}_{N+1} = \hat{\theta}_N + \frac{C_N v_{N+1}}{\rho + f_{N+1}^T C_N v_{N+1}} \left(y_{N+1} - f_{N+1}^T \hat{\theta}_N\right) \tag{9}$$

- Следващата итерация се подготвя, като се преизчисли C_{N+1} по формулата

$$C_{N+1} = \frac{1}{\rho}\left(C_N - \frac{C_N v_{N+1} f_{N+1}^T C_N}{\rho + f_{N+1}^T C_N v_{N+1}}\right) \tag{10}$$

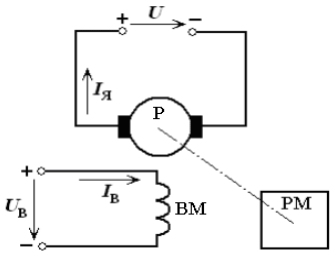
- Изчисленията започват от т. 2 с новите оценки. Ако не е необходимо претегляне на наблюденията, в (9) и (10) се полага $\rho = 1$.[1]

4. Система за управление на скоростта на постоянно токов двигател

Приложението на постоянно токовите двигатели е широко по отношение на автоматика, роботика и различни системи за управление. Подходящи са там, където е необходимо прецизно регулиране на момент и скорост, при подържане на момент при ниска и нулева скорост.

4.1. Постоянно токов двигател

Принципно устройство на двигатели за постоянен ток е показано на фигура 1, където РМ е работна машина, ВМ е възбудителна намотка, а Р е ротор.

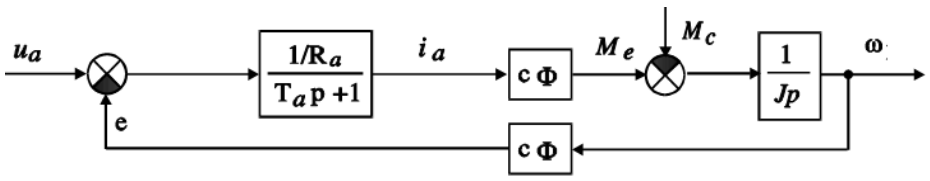


Фиг. 1. Принципно устройство на ПТД

Динамичните свойства на постоянно токовия двигател са описани с системата (11):

$$\begin{cases} u_a = R_a(T_a p + 1)i_a + c\Phi\omega \\ M_e = c\Phi i_a \\ M_e - M_c = Jp\omega \end{cases} \quad (11)$$

представена чрез структурната схема (фигура 2):

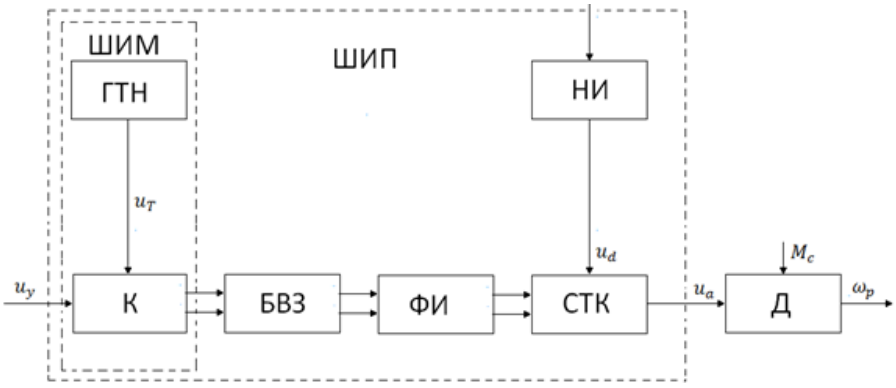


Фиг. 2. Структурна схема на ПТД

4.2. Управление на скоростта на постоянно токов двигател посредством широчинно-импулсен преобразувател

Съвременните системи за управление на постоянно токови двигатели са с широчинни импулсни преобразователи, тъй като се характеризират с по-добри енергетични показатели и по-малки пулсации на тока и скоростта, което от своя страна води до намаляване на загубите и разширяване на диапазона на регулиране.

На блоковата схема на фигура 4 е представена принципна блок схема на ШИП-ПТД , където: ГНТ- генератор на трионообразно напрежение, К – компаратор, БВЗ – блок за времезадръжка, ФИ – формировател на импулсите, НИ – неуправляем изправител, СТК – силов транзисторен комутатор, Д – постоянно токов двигател.

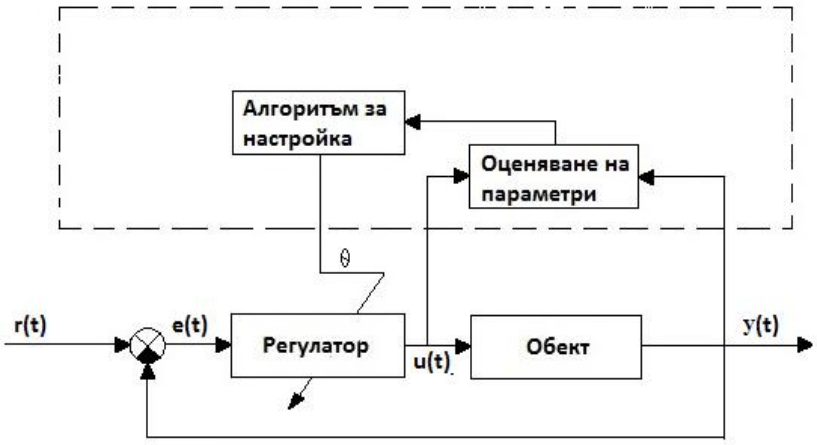


Фиг.4. Принципна блок схема на ШИП-ПТД

4.3. Адаптивна система със самонастройващ се регулатор

Самонастройващият се регулатор използва комбинация от процес за рекурсивна идентификация на базата на избран процес на модела и синтез на регулатора на базата на

оценените параметри на управлявания процес. Блок схема на самонастройващ се регулатор е показана на фигура 5. Самонастройващият се регулатор е един от методите за управление, при който е постигнат значителен напредък през последните години. Има три основни елемента (фигура 5). Първият представя класическа система със обратна връзка. Вторият осъществява идентификация на процеса. Третият реализира алгоритъма за настойки на параметрите на регулатора на базата на оценените параметри на процеса.



Фиг. 5. Блок схема на самонастройващ се регулатор

Тази блокова схема може да се използва както в стационарни системи с неизвестни параметри, така и в системи с променливи параметри. За целите на това изследване се използва директен самонастройващ се регулатор с минимална вариация (MV).

4.4. Синтез на адаптивна система за управление посредством самонастройващ се регулатор с минимална дисперсия

Комбинацията от метод на оценяване и закон за регулиране с минимална дисперсия води до следния алгоритъм.

4.4.1. Алгоритъм-1. Явен самонастройващ се регулатор с минимална дисперсия

- На базата на информация за входа и изхода на обекта към k-тия момент, чрез един от описаните методи за оценка на параметри, се определят оценките на параметрите $\hat{a}_i(k), \hat{b}_i(k), \hat{c}_i(k)$.
- Съгласно принципа на несъмнената еквивалентност, оценките се приемат за действителните стойности на параметрите. Определят се полиномите $E\left(z^{-1}\right)$ и $F\left(z^{-1}\right)$ чрез решаването на Диофантовото уравнение.

$$C\left(z^{-1}\right) = A\left(z^{-1}\right)E\left(z^{-1}\right) + z^{-d}F\left(z^{-1}\right) \tag{12}$$

- 3. Определят се коефициентите на полинома $G\left(z^{-1}\right) = E\left(z^{-1}\right)B\left(z^{-1}\right)$.
- 4. По формула (13)

$$\begin{aligned} u(k) = -\frac{1}{g_0}[\, & f_0 y(k) + f_1 y(k-1) + \ldots + f_{n-1} y(k-n+1) + \\ & + g_1 u(k-1) + g_2 u(k-2) + \ldots + g_{m+d-1} u(k-m-d+1)] \end{aligned} \tag{13}$$

се определя управлението $u(k)$, след което k се увеличава и алгоритъмът се повтаря от т.1.

4.4.2. Алгоритъм-2. Неявен самонастройващ се регулатор с минимална дисперсия

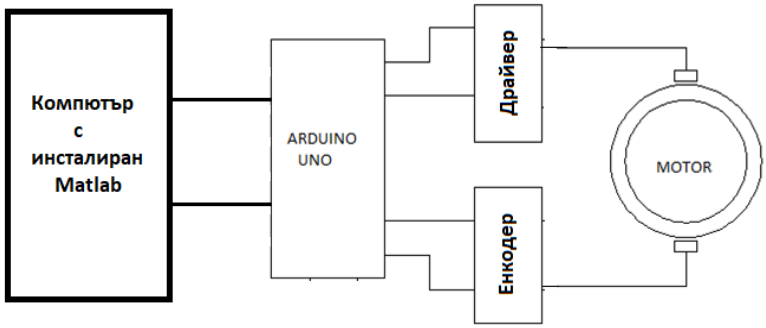
- На базата на информация за входа и изхода на обекта към k-тия момент, чрез RLS, се определят оценките на $f_i(k)$ и $g_i(k)$ на полиномите $F\left(z^{-1}\right)$ и $G\left(z^{-1}\right)$.

- Получените оценки се заместват в (13) и се определя управлението $u(k)$.

- След получаване на нова информация k се увеличава и алгоритъмът се повтаря от т.1.

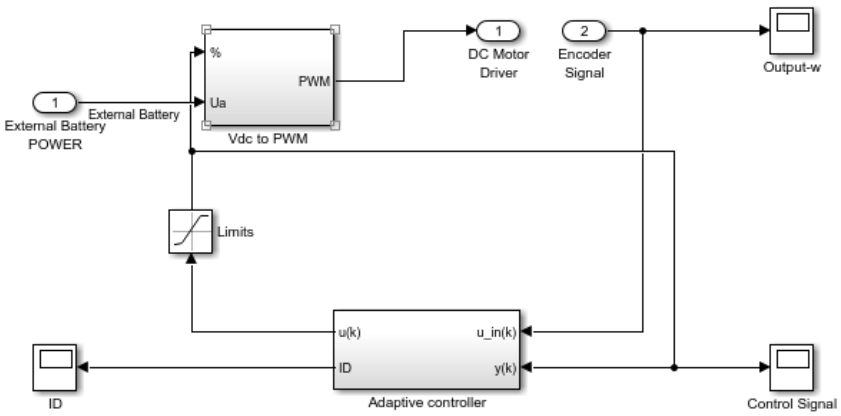
5. Експериментални изследвания и резултати

За целта на изследванията за приложението на рекурсивни методи за оценяване на реалните параметри в адаптивна система за управление постоянно токов двигател е разработена следната блок схема (фигура 6), като в нея е добавена микроконтролерната развойна платка Arduino Uno, с чиято помощ ще се извършва комуникацията между рекурсивните методи за оценяване на праметри, разработени в Matlab/Simulink и ПТД с НВ.

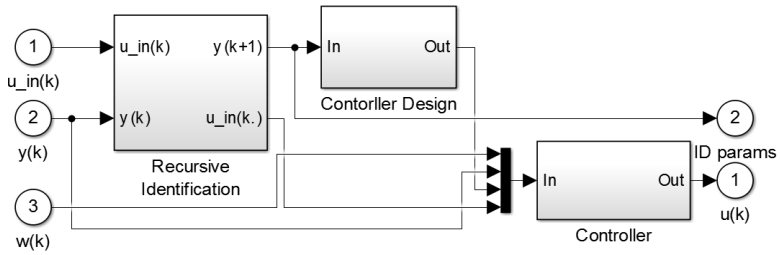


Фиг. 6. Блок схема на постановката

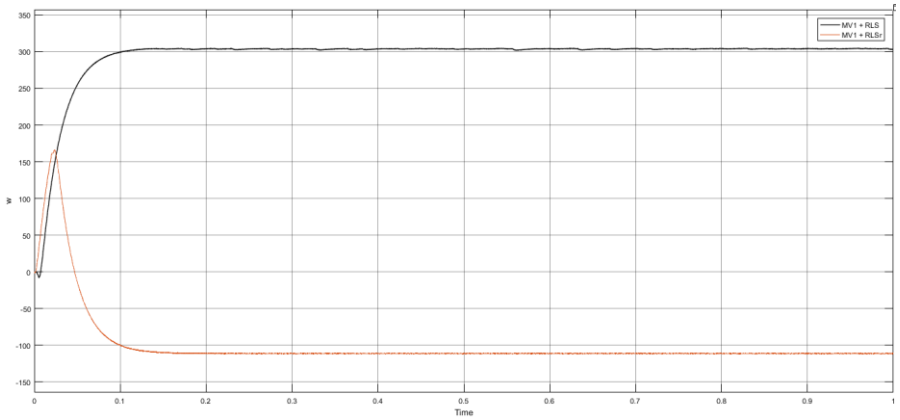
Изследванията са направени с използването на System Identification Toolbox в Matlab\Simulink и Arduino Support Package за Matlab\Simulink. За тази цел са направени блокове за адаптивно управление на скоростта на ПТД посредством явен самонастройващ се регулатор с минимална дисперсия (MV) и блокове за рекурсивно оценяване на параметри в реално време както следва: рекурсивен метод най-малките квадрати (RLS), рекурсивни най-малките квадрати с отчитане на остатъците вместо грешката на оценяване (RLSr).



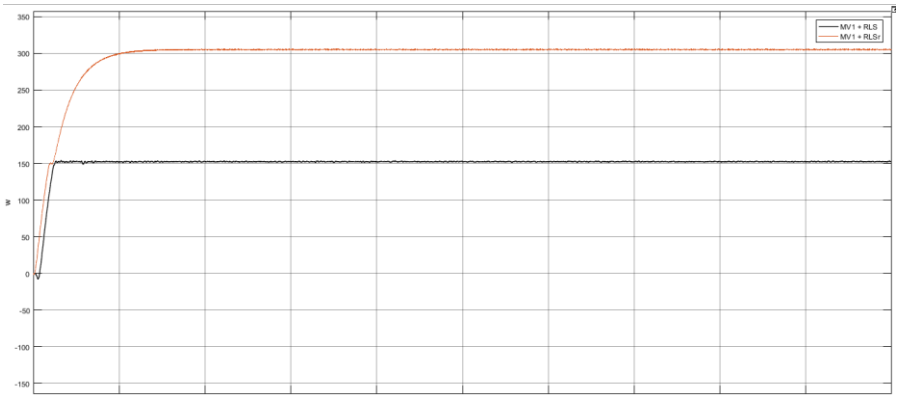
Фиг. 7. Блок схема в Simulink за връзка с Arduino Uno Rev.3 за управление на ПТД с НВ



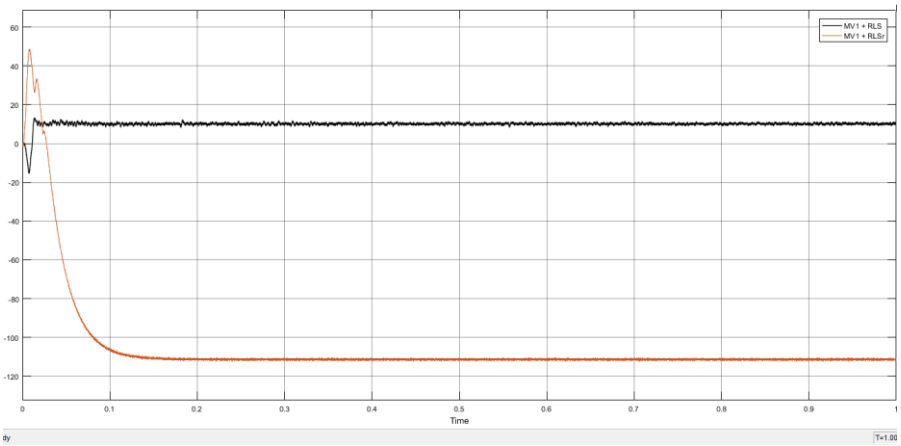
Фиг. 8. Блок схема в Simulink на система със самонастройващ се регулатор с минимална дисперсия представяща подсистемата "Adaptive controller"



Фиг. 9. Скорост на ПТД с MV +RLS и RLSr за $\omega = 304,7 rad / s$



Фиг. 10. Скорост на ПТД с MV + RLS и RLSr за $\omega = 152,35 rad / s$



Фиг. 11. Скорост на ПТД с MV + RLS и RLSr за $\omega = 10,1567 rad / s$

На фигура 9, фигура 10 и фигура 11 са представени скоростта на ПТД при система за адаптивно оценяване и управление на реалните параметри на ПТД посредством явен самонастройващ се регулатор с минимална дисперсия, като са използвани по-горе описаните методи RLS и RLSr при зададена скорост $\omega = 304,7 rad/s$, $\omega = 152,35 rad/s$ и $\omega = 10,1567 rad/s$.

6. Анализ и изводи

Проведените експерименти доказват възможността за приложение на рекурсивни методи за оценяване нареалните параметри на ПТД в адаптивна система за управление. Те дават възможност за оценяване параметрите на двигателя в реално време и адекватно регулиране на неговата скорост. Необходими са по-нататъшни изследвания за настройка за работата на схемата и на комуникацията между Matlab/Simulink и Arduino Uno, заради наличието на забавяне при изчисленията и изработването на управляващ сигнал при рекурсивни най-малките квадрати с отчитане на остатъците вместо грешката на оценяване (RLSr), идващо от ограниченията на развойната платка. Описаните методи за оценка на параметрите могат да бъдат модифицирани за по-добра производителност и качество на процесите в други адаптивни системи за управление в реално време. При избиране на различни тегла могат да се получат различни робастни оценки, които могат да бъдат нечувствителни на шум. Това ще бъде обект на по-нататъшно проучване.

БЛАГОДАРНОСТИ

Научните изследвания, резултатите от които са представени в настоящата публикация, са извършени по проект НП-7 в рамките на присъщата на ТУ-Варна научно изследователска дейност, финансирана целево от държавния бюджет за 2017 година.

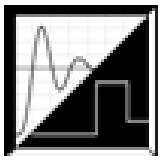
Литература

[1]. Agustin O., Oscar L, Francisco Q.: Identification of DC motor with parametric methods and artificiql neural networks,2012
[2]. Andonov A., Hubenova Z, Robust methods for control under undetermined criteria,2008
[3]. Allan Aasbjerg Nielsen, Least Squares Adjustment: Linear and Nonlinear Weighted Regression Analysis, 2013
[4]. Bobál V, Chalupa P, Kubalčík M, Dostál P.,Identification and Self-tuning Control of Time-delay Systems, 2012
[5]. Joshua D. Angrist and Alan B. Krueger:Instrumental Variables and the Search for Identification,2001
[6]. Kama O., Mahanijah K., Nasirah M. and Norhayati H. : System Identification of Discrete Model for DC Motor Positioning
[7]. Krneta R., Anti´c S., Stojanovi´c D. : Recursive Least Squares Method in Parameters Identification of DC Motors Models, 2005
[8]. Landau, I.D., Lozano R., M'Saad H,. Kariirii A.: Adaptive Control Algorithms, Analysis and Applications, 2011
[9]. Hassan, L.H., Unknown input observers design for a class of nonlinear time-delay systems, 2011
[10]. Naira Hovakimyan, Chengyu Cao: Adaptive Control Theory Guaranteed Robustness with Fast Adaptation,*2010*
[11]. Mohamed M. Hassan, Amer A. Aly, Asmaa F. Rashwan, Different identification methods with application to a DC motor

[12]. Adjustin PWM Frequencies,
http://playground.arduino.cc/Main/TimerPWMCheatsheet,2015
[13]. Arduino Software, Version 1.6.5, https://www.arduino.cc/en/Main/Software
[14]. Arduino\ UNO, https://www.arduino.cc/en/Main/arduinoBoardUno, 2015.

За контакти:
докторант, инж. Иван В. Григоров
катедра „Автоматизация на производството”
Технически университет-Варна
E-mail: ivan_grigorov@tu-varna.bg

доц. д-р Наско Р. Атанасов
катедра „Автоматизация на производството”
Технически университет-Варна
E-mail: nratanasov@tu-varna.bg



VLC КОНТРОЛЕР ЗА НИСКОСКОРОСТНА КОМУНИКАЦИЯ

Жейно Ив. Жейнов

Резюме: Разгледана е структурата на система за предаване на информация чрез използване на осветлението. Обсъжда се ниско скоростен модул за комуникация с използване на видимата светлина /Visible Light Communication/. Управлението на оптичните приемо-предаватели е реализирано чрез микрокомпютър. Целта на работата е създаване на образователен прототип.
Ключови думи: Предаване на информация чрез осветлението, ниско скоростна комуникация, микрокомпютър, управление на LED.

VLC CONTROLLER FOR LOW-SPEED COMMUNICATION

Zhejno I. Zhejnov

Abstract: The structure of a system for transmitting information using the lighting is explored. A low-speed communication module with Visible Light Communication is discussed. The control of the optical transceivers is realized by a microcomputer. The aim of the work is to create an educational prototype.
Keywords: Information transmission via lighting, low speed communication, microcomputer, LED control.

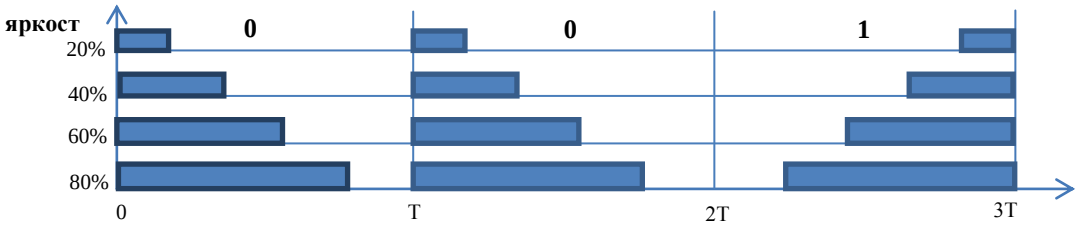
1. Увод

Безжичното предаване на данни по технологията Visible Light Communication позволява да се отстрани интерференцията със съществуващото радиоелектронно оборудване за предаване на данни с радиовълни и микровълни в някои помещения като болници и самолети. Използват се предимствата на широко разпространените бели светодиоди (LED) - източници на светлина с голяма яркост, бързо превключване, висока ефективност, дълъг експлоатационен живот. Освен това във видимата и инфрачервената област на оптичния диапазон много от използваните честотни ленти не се регулират.

2. Изложение

За да се използва светлината както за осветление, така и за предаване на съобщения, е необходимо да се модулира светлинен източник чрез модем, който е свързан към аналогов или цифров широколентов източник на данни. Използването на светлинното оборудване за разработка на технологията VLC се осъществява от голяма група университети, корпорации и глобални организации. Японската промишлена асоциация JEITA създава през 2008 г. стандарта „Visible Light ID System“ в сътрудничество с консорциума VLC (VLCC) и Асоциацията за инфрачервени данни (IrDA). В края на 2011 г. IEEE създаде проект-стандарта IEEE 802.15.7 за VLC. Той включва в себе си физическото ниво (PHY), интерфейса Air Interface и нивото за достъп до средата (MAC). Стандартът 802.15.7 използва код Манчестър с един и същи период на положителните и отрицателните импулси. Това удвоява пропускната способност при OOK модулацията. На по-високите скорости се използва RLL кодиране с по-висока спектрална ефективност. Изменението на яркостта става чрез разширяването на импулсите с OOK модулация и така се мени постояннотоковата съставна. Предаването на данните се осъществява чрез променлива позиционно-импулсна модулация /Variable pulse position modulation/ - VPPM. Данните се кодират чрез промяна на позицията на импулса в периода на кодиране. Дължината на периода, който съдържа импулс, трябва да бъде достатъчна за безпогрешната детекция на неговата позиция. Логическата 0 се кодира

като положителен импулс в началото на периода, а после следва отрицателен импулс. Логическата 1 се кодира като отрицателен импулс в началото на периода, а после следва положителен импулс – фигура 1.



Фиг. 1. Променлива позиционно-импулсна модулация и управление на яркостта

Стандартът описва физическото ниво на мрежата в приложения с ниска скорост - 12-256 kbit/s. В случая OOK и VPPM модулация за права корекция на грешките /Forward error correction/ се използва конволюционно кодиране (convolutional encoding). Единичните грешки се поправят чрез конволюционен декодер. Ползват се също кодове на Рид-Соломон. Осветлението в офиса с използването на светодиоди се стандартизира от Международната организация по стандартизация ISO и в съответствие с този набор от стандарти трябва да бъде от 300 до 1500 lx. За да покрива тези стандарти е необходимо да се създаде определена конструкция на осветлението, свързана с размерите на стаята и с използваните светодиодни източници. Поради бързото превключване на тока през светодиодите е необходимо да се използват бели светодиоди, които се монтират на тавана на помещението в правоъгълна матрица на равни разстояния един от друг. Те са свързани електрически последователно и се захранват с ниското захранващо напрежение на захранването през транзисторен превключвател. Това понижава изискванията към превключването. Тъй като приносът в интензивността на осветеността E_{hor} от всеки светодиод в хоризонталната равнина, перпендикулярна на оста на светодиода се определя от израза [2]:

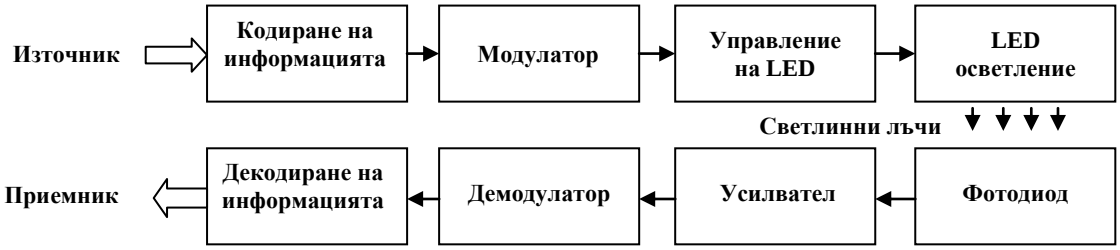
$$E_{hor} = I(0)cos^m(\phi)/D_d^2\cos(\psi)\,, \tag{1}$$

където $I(0)$ е централната интензивност на LED осветлението, ϕ е ъгъла на излъчването, ψ е ъгълът на падане, D_d е разстоянието до осветяваната повърхност. Във формулата (1) се предполага, че интензивността на излъчването на светодиода се определя по Ламбертиановски закон - зависи от ъгъла на излъчване ϕ и реда на Ламбертиана m, който определя половината от ъгъла на максималната интензивност на излъчване $\Phi_{1/2}$, където:

$$m = \ln 2/\ln(\cos\Phi_{1/2})\,. \tag{2}$$

Когато m=1, половината от ъгъла на максималната интензивност на излъчването е 60^0, както е например на светодиодите на фирма Cree.

Примерната блок-схема на система за предаване на информация с използването на осветлението е показана на фигура 2. Източник на цифрова информация е някакво устройство, ПК, компютърна мрежа. Информацията се кодира и постъпва на модулатор, който преобразува информацията, постъпваща от източника така, че да се променят съответните параметри на излъчената светлина. Управлението на излъчвателя се осъществява от мощни драйверни схеми, управляващи тока през светодиодите на осветлението. Фотодиод приема модулираната светлина от осветлението и я преобразува в напрежение. Усиленият сигнал се демодулира и декодира, а после постъпва на приемника на информацията.



Фиг. 2. Система за предаване на информация чрез осветлението

Кодирането и декодирането на информацията, а също така съгласуването на интерфейсите обикновено се реализира от специализирани устройства. Когато комуникацията е нискоскоростна, напълно оправдано е използването на евтини микрокомпютри за кодер/декодер и за съгласуване на интерфейсите на източника и получателя. Тук може да се ползват модулите Arduino [3], снабдени например с модул Ethernet Shield за свързване на мрежова карта и USB интерфейс.

За Arduino има достъпна развойна среда за програмирае на базата на и ОС Windows XP/7. Средата е безплатна и свободно се развива и усъвършенствува. В Интернет са предоставени базови примери, много библиотеки и свободен код. Програмата за управление на микрокомпютъра се пише подобно на програма на език С. Компилира се и после се зарежда във Flash паметта на модула за управление.

3. Заключение

Предложена е структура на система за предаване на информация с използване на осветлението. Тя може да се реализира с достъпни елементи, които се продават в търговската мрежа. Системата е подходяща за експерименти и изучаване на особеностите и протоколите при комуникация чрез видима светлина.

Литература

[1]. Boker, A., V. Eklind, D. Hansson, P. Holgersson, J. Nolkrantz, A. Severinson. An implementation of a visible light communication system based on LEDs: Bachelor Thesis / Gothenburg, Sweden: Chalmers University of Technology, 2015, с.20-25.

[2]. Komine, T., M. Nakagava. Fundamental Analysis for Visible-Light Communicatiom System using LED Lights. IEEE Transactions on consumer Electronics, Vol.50, №1, 2004, с. 1-8.

[3]. Виктор Петин. Проекты с использованием контроллера Arduino. Санкт Петербург, „БХВ Петербург”, 2016, ISBN 978-5-9775-3550-2.

За контакти:
Гл.ас. д-р Жейно Ив. Жейнов
катедра „Компютърни науки и технологии”
Технически университет - Варна
E-mail: zh_viv@tu-varna.bg

ИЗСЛЕДВАНЕ НА ФИЗИЧЕСКИ ПРОТОТИП ЗА LI-FI КОМУНИКАЦИЯ

Диян Ж. Динев

Резюме: Този доклад показва условията, при които е приложим разработен физически прототип за предаване на информация чрез Li-Fi технология. Докладът представя получените резултати при изследването на физическия прототип – разстояние за предаване без грешки, ъгли на осветеност, влияние на околната среда, преносна среда въздух-вода.
Ключови думи: Li-Fi, Интернет за нещата, Безжични комуникации

Research of physical model for li-fi communication

Diyan Zh. Dinev

Abstract: This report shows the conditions under which a developed physical model for transmitting information via Li-Fi technology is applicable. The report shows the results obtained in the study of the physical model - such as error-free transmission distance, angles of illumination, environmental impact, air-to-water transmission environment.
Keywords: Li-Fi, IoT, Wireless Communications

1. Увод

Li-Fi (Light Fidelity) е бърза и оптична версия на Wi-Fi, която зависи от Visible Light Communication (VLC). VLC е средство за предаване на данни, което използва спектъра на видимата светлина между 400 THz (780 nm) и 800 THz (375 nm) като оптичен преносител за предаване на информация. Той използва бързи импулси на светлината, за да прехвърля лесно информацията. Основните части на тази комуникационна система са: LED, който се използва като източник на информацията, и фотодиод, който служи като елемент за получаване. LED може да се включва и изключва, за да се създадат цифрови низове от нули и единици. Информацията може да бъде кодирана, за да се направи нов поток от данни, който се различава по диференциалната скорост на светодиода, да се направи по-чиста чрез модулиране на светодиодната светлина с индикация за данните. Сегашната светодиодна технология предлага ефективност, сигурност и висока скорост на управление на устройства. Тези източници излъчват безопасна за човешкото око светлина и не увреждат околната среда. Затова светодиодите са подходящи не само за осветление, но и за предаватели при комуникация чрез видима светлина. [1].

До момента са създадени повече от 30[2] работещи прототипи за предаване на информация чрез LED. Сравнено с Wi-Fi, Li-Fi може да бъде считано за по-добро, понеже Wi-Fi има няколко ограничения, като това че използва 2.4 – 5GHZ радио честоти за пренос на интернет и честотоната лента е ограничена до 50-100Mbps.

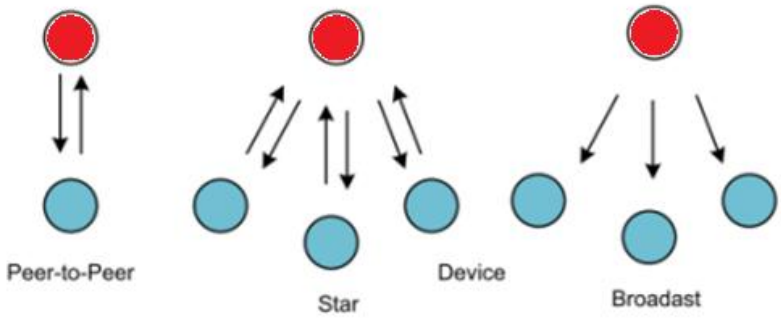
Li-Fi има следните предимства спрямо останалите безжични комуникации [3]:

- Първото и основно предимство на Li-Fi е, че може да бъде използвано в посредствена близост до чувствителни на електромагнитни смущения райони – като болници, атомни електроцентрали, самолети, без да предизвиква електромагнитни смущения;
- Технологията използва честотната лента на видимата светлина, която е в пъти по-голяма от тази на радиочестотите, което позволява по-високи скорости на предаване на информацията – достига до 10Gbps;

- Осигурява по-голяма защита и поверителност при предаването на информацията, понеже видимата светлина непреминава през стените и по този начин има защитен достъп;
- Също така може да се използва и за подводни изследвания за разлика от радиочестотните безжични комуникации.

2. Топологии

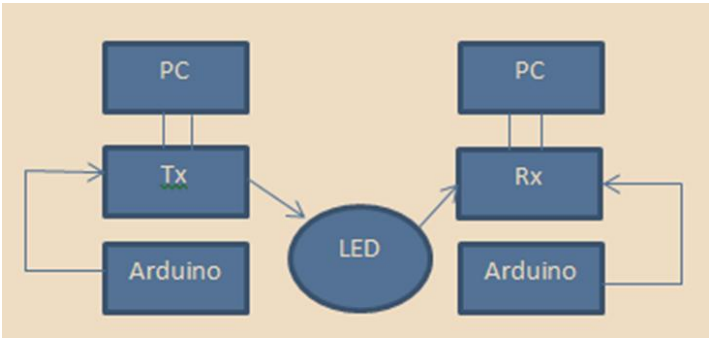
Топологиите поддържани от MAC нивото са точка-до-точка(peer-to-peer), broadcast извезда(star)които са показани на Fig. 1[4]. Топологията тип звезда се осъществява, чез употребата на централизирано устройство, което да координира работата.



Фиг. 1. Поддържани топологии

В peer-2-peer топология две устройства комуникират и едното от двете осигурява поддръжката на комуникацията - то става координатор (Terminal). В топологията звезда всички устройства имат двупосочна комуникация с координатора. Третата топологията е broadcast излъчване, при която потребителите само получават данни от предавателя. В рамките на тези три типа топологии се използват три типа предаване на информацията. Първият включва данните, изпратени от устройство на координатора, вторият изпраща данните от координатора на устройството и третият - данните се предават между устройствата в peer-to-peer връзката.

3. Опитна постановка



Фиг. 2. Блок-схема на физическия прототип

На фигура 2 е представена блок-схема на физическия прототип за Li-Fi безжична комуникация между два компютъра – един за предаване на данните (Tx)и един за приемането им (Rx), като за целта се използва LED светодиод, приемна и предавателна част и Arduino Uno за програмирането им.

Фиг. 3. Опитна постановка

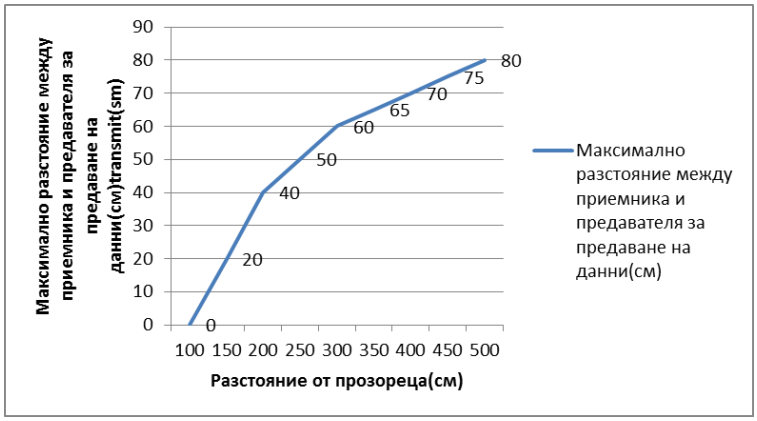
На фигура 3 е показан физическият прототип, чрез който са направени изследванията за разстояние, на което може да се осъществява комуникация, ъгли на осветеност, влияние на околната среда, преносна среда въздух-вода. Самият прототип, използван за експериментите, е описан в [5]. За извършването на експериментите с преносна среда въздух-вода е използван аквариум с размери 55см/26см/36см(50л.).

4. Резултати

Експериментите са направени в стая с южно изложение на дневна светлина между 13 и 17 часа следобед, при наличието на директна слънчева светлина, влизаща през прозорците.

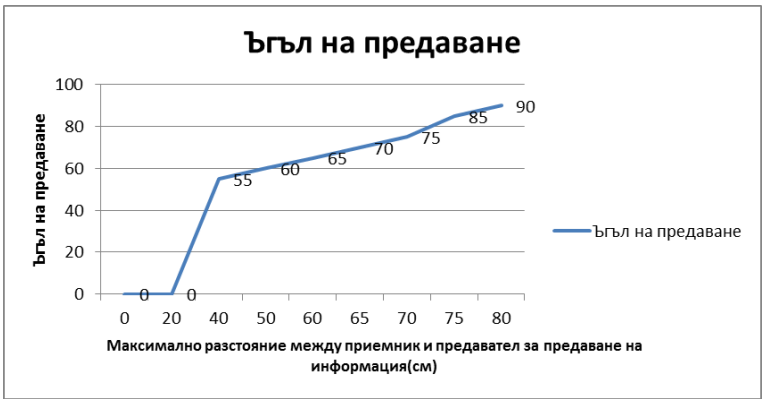
Първата част от направените експерименти се отнасят за това как светлината влияе на разстоянието на предаване на информацията. Всички експерименти за разстоянието между приемника и предавателя са направени при ъгъл на осветеност 90 градуса (фигура 4).

От фигура 4 може да бъде направено заключението, че пряката слънчева светлина води до 100% загуба на предаваната информация към приемника дори и той да бъде на 1 см от предавателното устройство. На 2 метра от прозореца, където няма пряко наличие на светлина, може да бъде изпратена информация при 20см разстояние между приемника и предавателя. С отдалечаването от прозореца се увеличава и почти правопрорционално разстоянието, на което може да се предаде информация между Tx и Rx. При достигане на 3 метра успешно да се достигне до 60 см разстояние. С отдалечаване от прозореца все повече и повече се увеличава и разстоянието. Може да бъде направен извод, че след достигане на 3 метра и последвало увеличаване на разстоянието от прозореца с по 50см, се наблюдава увеличаване с по 5см и на дистанцията, на която може да се предава информация. Максималната стойност от 80см е достигната при нощни условия, без наличието на светлина.



Фиг. 4. Влияние на слънчевата светлина на разстоянието на предаване

Втората част от екпериментите е фокусиране над ъгъла на осветеност в напълно тъмна стая (фигура 5).



Фиг. 5. Влияние на ъгъла на осветеност на разстоянието за предаване

От фигура 5 може да бъде направено заключението, че под ъгъл на осветеност по-малък от40 градуса не може да се наблюдава приемане на информацията. С увеличаването му успешно се постигат все по-големи и по-големи разстояния, като при достигане на прав ъгъл (90 градуса) се достигат максималните стойности от 80см.

Третата част от експериментите е насочена към това как влияят различните стандартни дебелини стъклени повърхости върху разстоянието, на което успешно се предава информация чрез изследвания прототип. Максималните получени стойности са показани на Таблица 1.

Таблица 1. Разстояние между приемник и предавател при наличието на прозрачно бяло стъкло между тях.

Милиметри „прозрачно бяло стъкло“	**Разстояние между приемника и предавателя.**
2мм	80
3мм	77
4мм	75
5мм	72
6мм	69
8мм	62
10мм	55
12мм	40

За последната част от експериментите е включен аквариум с размери 55см/26см/36см. Направени са експерименти за това на какво разстояние успява да работи устройството при наличието на преносна среда въздух-вода. Направени са експерименти със сладка вода, солена вода (10% солен разтвор и 20% солен разтвор). Резултатите са показани в Таблица 2.

Таблица 2. Преносна среда въздух - вода

Разстояние	Сладка вода	Солен разтвор 10%	Солен разтвор 20%
25	Да	Да	Да
50	Да	Да	Не
60	Да	Не	Не
75	Не	Не	Не

Поглеждайки Таблица 2, се достига до заключението, че показаният прототип успява да работи при наличието на вода като преносна среда, независимо каква е тя - дали сладка или солен разтвор. Колкото е по-голямо наличието на сол във водата, толкова повече намалява разстоянието, на което успешно се предава информация между двете устройства.

5. Заключение

В този доклад са показани изследвания върху възможностите на реализиран прототип за Li-Fi комуникация. Изследваният прототип е работоспособен при изследваните по-горе параметри – разстояние, ъгъл на осветеност, преносна среда въздух-вода и наличие на стъклени повърхости между устройствата.

Литература

[1]. How LiFi works, 2017, http://www.lifi.com/pdfs/techbriefhowlifiworks.pdf, последно посетен: 08. 08, 2018г.

[2]. LiFi: From Illumination to Communication, 2016, https://www.greyb.com/li-fi-vlc-patent-landscape-study, последно посетен 08.08.2018г

[3]. Advantages and disadvantages of Li-Fi. August 22, 2018.

[4]. https://www.profolus.com/topics/advantages-and-disadvantages-of-li-fi/, последно посетен 10.08.2018г.

[5]. Khan, L. U. Visible light communication: Applications, architecture, standardization and research challenges. //Digital Communications and Networks, 2017, Issue 2, Volume 3, pp. 78–88

[6]. Dinev D. Model for Research of Li-Fi Communication //Proceedings of the 1st International Conference "Applied Computer Technologies" ACT 2018, Ohrid, Macedonia, 21-23 June 2018, pp: 131 – 134. ISBN: 978-608-66225-0-3

За контакти:
Инж. Диян Ж. Динев
катедра „Софтуерни и интернет технологии"
Технически университет - Варна
E-mail: diyandinev@tu-varna.bg

ЕДИН НАЧИН ЗА НАСТРОЙКА НА ХИБРИДЕН ПИ РЕГУЛАТОР ПОСРЕДСТВОМ MATLAB, ПО ЖЕЛАНИ ПАРАМЕТРИ НА ПРЕХОДНИЯ ПРОЦЕС

Веско Хр. Узунов

Резюме: Разглежда се специализиран блок на MATLAB/SIMULIMK, служещ за настройка на зададени от потребителя параметри на регулатора с цел получаване на преходен процес със зададено качество. В конкретния случай той се използва за настройка на желания преходен процес на хибриден ПИ регулатор с конвенционална П съставна и размита И съставна, управляващ температурата на конкретен обект. Целта е да се изследва качеството на настройката върху реален обект и да се направят съответни изводи за практическата ѝ приложимост.
Ключови думи: Преходен процес, ПИ регулатор, размит регулатор, пропорционална съставна, интегрална съставна, пререгулиране, време на преходния процес, колебателност.

ONE WAY TO SET THE HYBRID PI CONTROLLER THROUGH MATLAB, AT REQUIRED PARAMETERS OF THE TRANSITION PROCESS

Vesko Hr. Uzunov

Abstract: A specialized MATLAB / SIMULIMK block is considered to set user-defined parameters of the controller to obtain a transition process of the specified quality. In this case, it is used to set the desired transient process of a hybrid PI regulator, with conventional P composite and fuzzy I composite, controlling the temperature of a particular object. The aim is to examine the quality of the setup on a real object and to draw relevant conclusions about its practical feasibility.
Keywords: Transitional Process, PI Regulator, Fuzzy Regulator, Proportional Component, Integral Component, Overshoot, Settling Time, Volatility

1. Въведение

За да се използва даден регулатор с даден обект, има две основни стъпки. Първата стъпка се състои в избора на вида на регулатора, който се очаква, че ще удовлетворява желаните от потребителя параметри на регулиране. Тази стъпка е сравнително добре проучена, тъй като се знаят възможностите на различните видове класически регулатори и съществуват много методи за техния избор според обекта за управление. Това означава да познаваме добре обекта и да имаме негово точно математическо описание. Без втората стъпка обаче, можем да не получим желания резултат, въпреки правилния избор на регулатора. Тази втора стъпка е настройката на параметрите на регулатора [3, 5]. Те зависят от конкретните стойности на параметрите на управлявания обект, но също така и от това какво желаем да получим. За тази настройка също са разработени много методи, както аналитични така и практически, но те са основно за класическите (П, ПИ, ПИД и др.) регулатори. Освен това са разработени основно методи, които оптимизират един показател, по-рядко два и много рядко повече. При оптимизиране на повече от един показател често се оказва, че те си противодействат. Например критериите за бързодействие и минимално пререгулиране са противоречиви (увеличаването на бързодействието увеличава пререгулирането). Тогава се търси компромисно решение. Още по-тежко става положението, когато имаме размити регулатори [2], тъй като при тях са твърде много и параметрите на

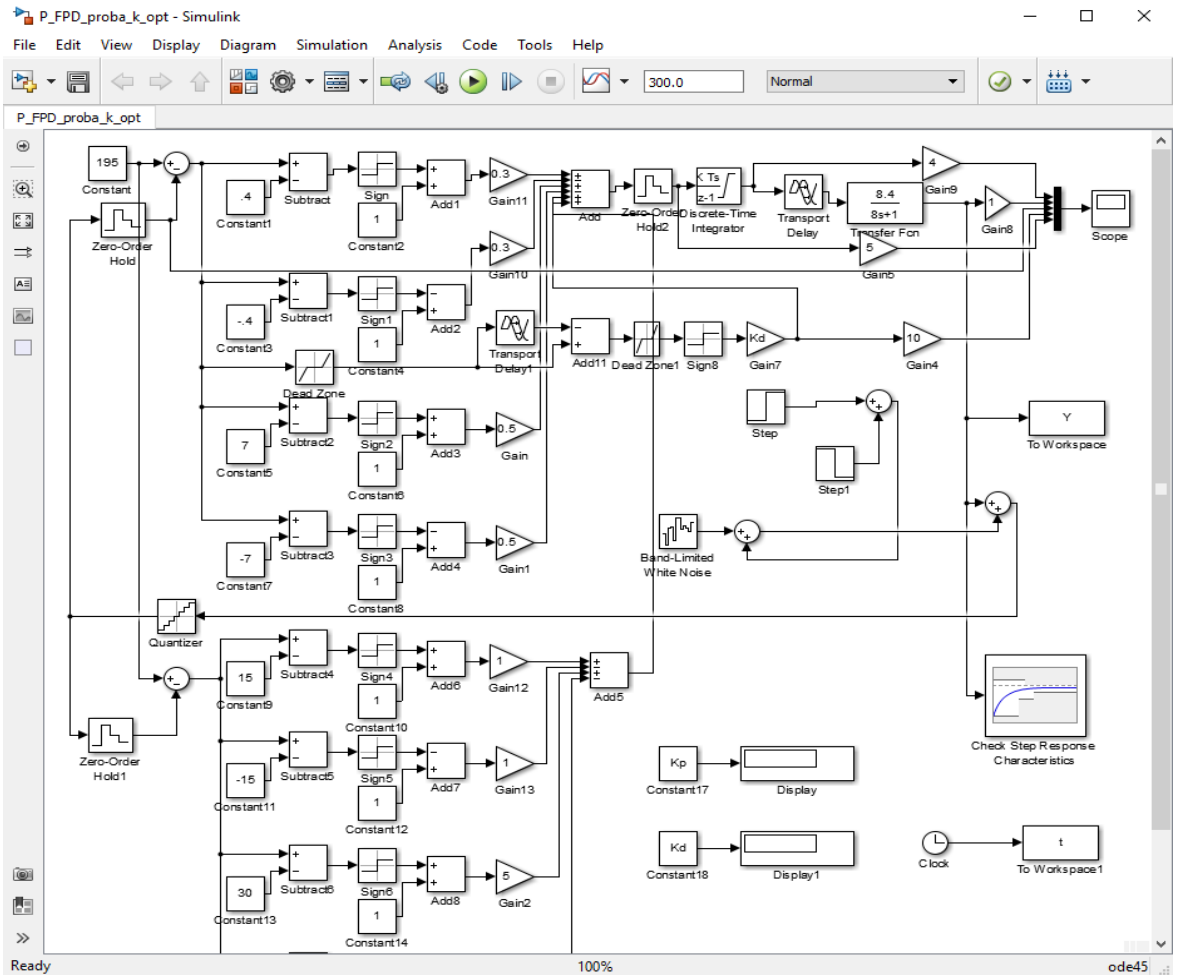
регулатора, които могат да се настройват и не са така добре разработени методи за тяхната настройка, поради голямото разнообразие на реализации.

В случая е разгледана една конкретна реализация на ПИ хибриден регулатор, с класическа П и хибридна И съставна, реализирана с контролер от нисък клас LOGO! на фирма Siemens. Извършена е настройка за сравнително бързодействащ температурен обект, като целта е постигане на определени параметри на преходния процес.

2. Изложение

За оптимизиране на преходния процес може да се използват и възможностите на MATLAB/Simulink [1]. За целта обаче, трябва да имаме сравнително точен модел на цялата система обект-регулатор. Разработката на модела на конкретния регулатор и негова разновидност за обект с интегриращи свойства са разгледани в други статии [4, 6].

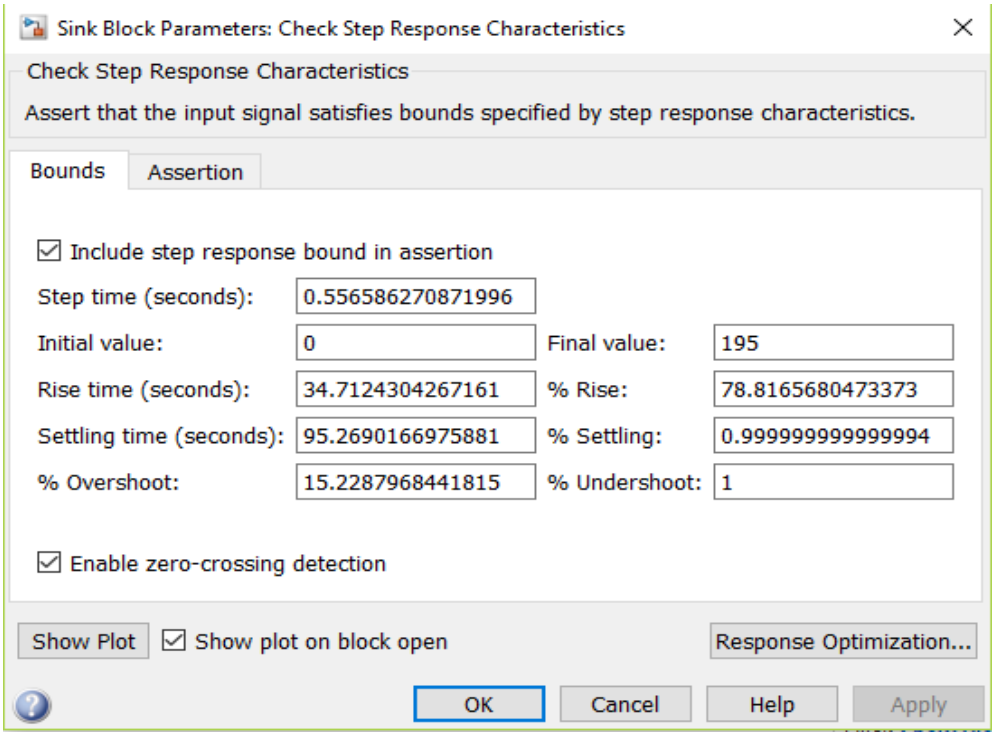
Това може да стане, като в модела на системата се въведе блокчето Check Step Response Characteristics, което избираме от подменюто Simulink Design Optimization/Signal Constraints. На фигура 1 е даден моделът на симулираната система с добавени необходимите блокове за стартиране на оптимизацията.



Фиг. 1. Модел, използван за оптимизация на параметрите на регулатора

Освен блока за конструиране на характеристиките на преходния процес при степенчато входно въздействие (Check Step Response Characteristics) са добавени и променливите Kp и Kd като блокчета Constant, а също и блокове Display към всяко едно от тях. Те показват текущите стойности на константите. Самите стойности на константите се задават в

работното пространство на MATLAB (Workspace) и се променят на всяка стъпка от оптимизацията до достигане на желания резултат.



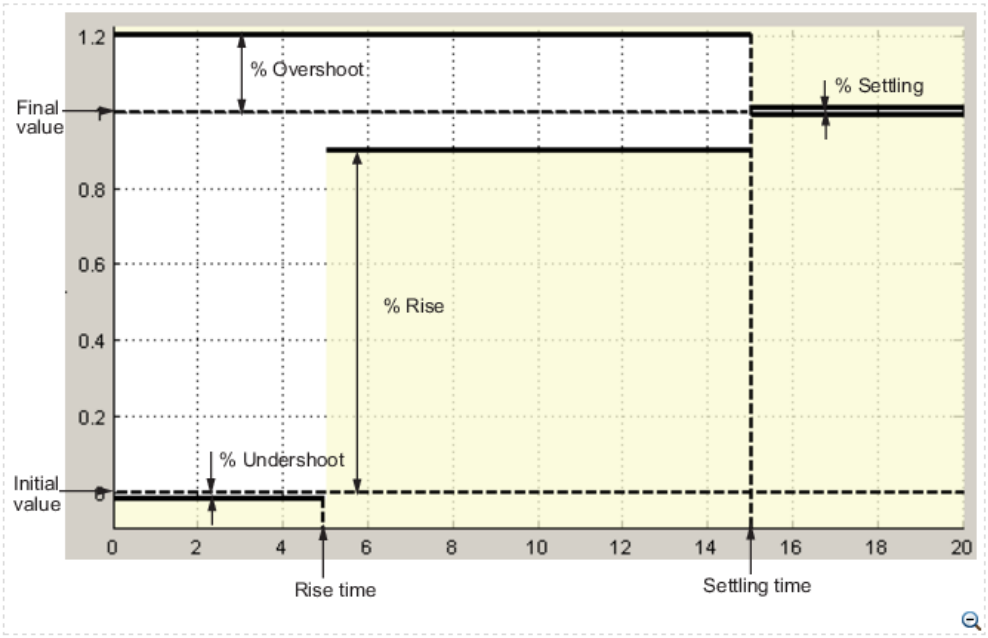
Фиг. 2. Прозорец за задаване параметрите на оптимизация.

Желаните параметри на преходния процес се въвеждат в диалогов прозорец за настройка на блока Check Step Response Characteristics. На фигура 2 е показан самият прозорец с произволни примерни стойности.

На фигура 3 в графичен вид се вижда кой параметър променя съответната величина на преходния процес. Задават се следните параметри, като в скоби са дадени стойностите им по подразбиране:

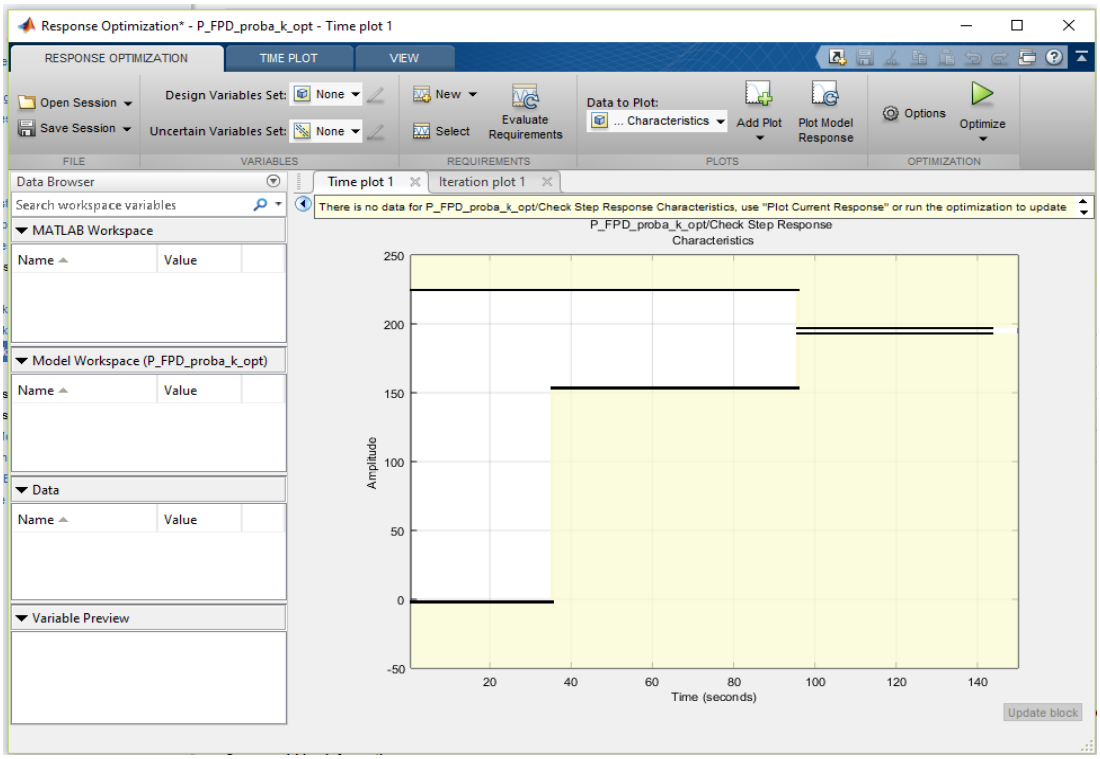
- Step time – времето на началото на стъпковото въздействие в секунди (0),
- Initial value – начална стойност на величината (0),
- Final value – крайна стойност на величината (1),
- Rise time – време, в секунди, необходимо на сигнала да нарасне до процента от крайната стойност, зададена в параметъра %Rise (5),
- %Rise – процентът от финалната стойност на величината, използван в Rise time (80),
- Settling time – времето, в секунди, необходимо на сигнала да влезе в специфични граници около крайната стойност, зададени в параметъра %Settling (7),
- %Settling – процент от крайната стойност след влизането, в който преходът се счита за завършен и се отчита времето на преходния процес (1),
- %Overshoot – пререгулиране, показващо процента на максималното първо отклонение спрямо крайната стойност (10),
- %Undershoot - сигнал, показващ допустимата стойност на сигнала под първоначалната стойност в проценти спрямо крайната стойност (1).

За графиката на фигура 3 са зададени: Step time – (0); Initial value – (0); Final value – (1); Rise time – (5); %Rise – (90); Settling time – (15); %Settling – (1); %Overshoot – (20); %Undershoot - (1).

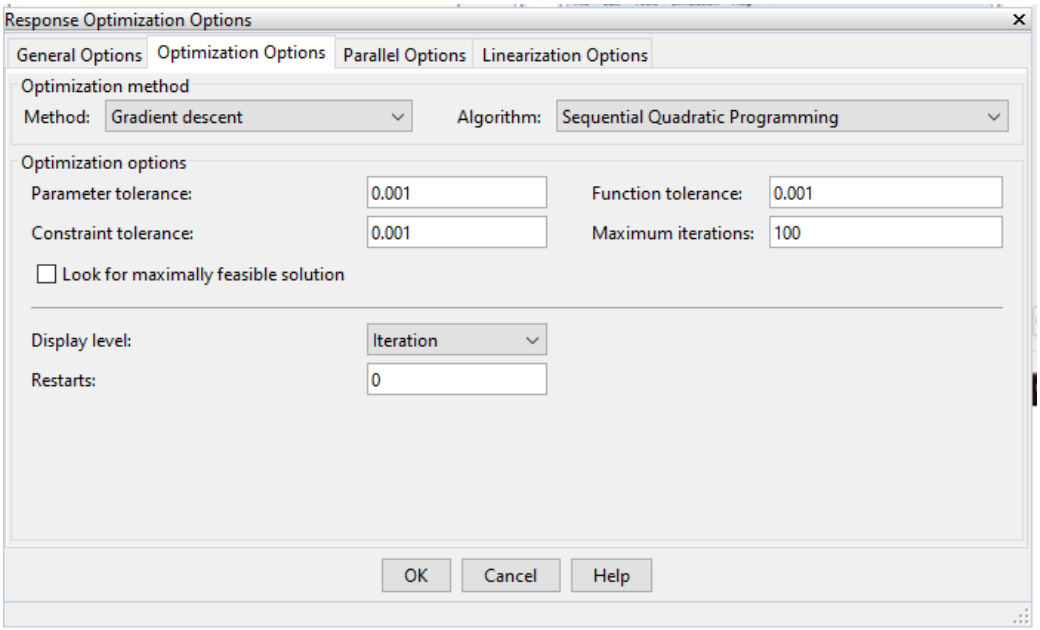


Фиг. 3. Параметри на оптимизацията

След като се зададат параметрите на преходния процес, се отваря прозорецът за оптимизация. В него (фигура 4) от Design Variables Set се избират параметрите на модела, които ще се оптимизират, като се задават имената на променливите и границите, в които е допустимо да се изменят. В случая това са Kp и Ki, а границите са от 0.001 до плюс безкрайност. Могат да се настроят и опциите на оптимизацията, като метод и алгоритъм на оптимизация, толеранси на параметрите, и максимален брой на итерациите (фигура 5). Като в примера са избрани метод „gradient descent" и алгоритъм „sequential quadratic programming".

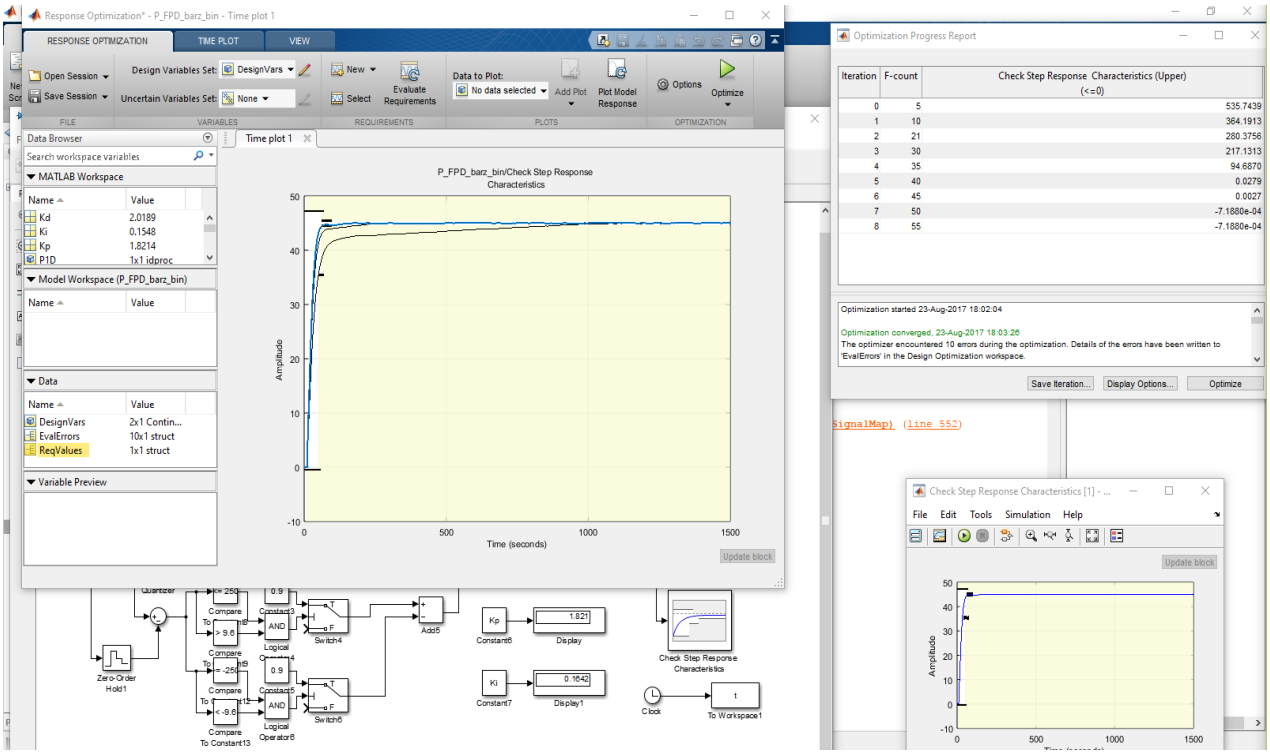


Фиг. 4. Прозорец за задаване желаните параметри на оптимизацията



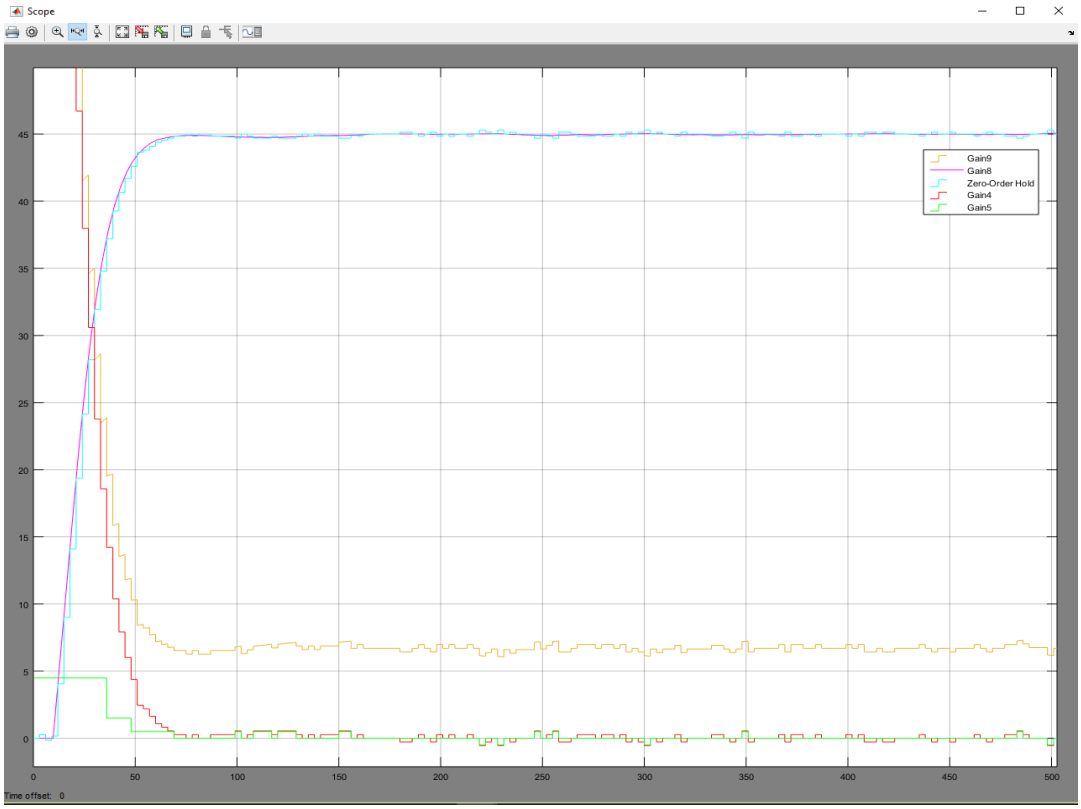
Фиг. 5. Прозорец за настройка на оптимизацията

След задаване на ограниченията за оптимизация и осем итерации, се стига до много бърз процес, при това без колебания (синята линия на фигура 6), който отговаря на нашите изисквания. Виждат се и другите итерации (в черно), които не отговарят на зададените ограничения.



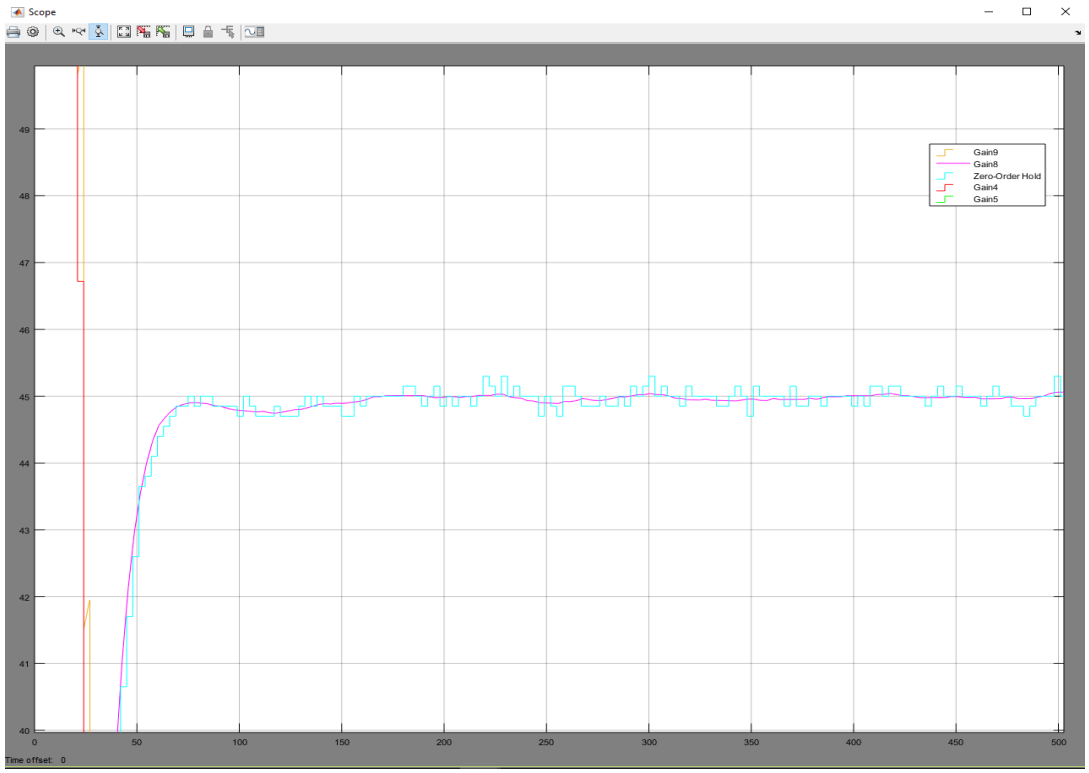
Фиг. 6. Екранът с прозорците на оптимизацията

На фигура 7 са дадени изходите на отделните компоненти на регулатора (в червено са стойностите на прпорционалната съставна, в зелено стойностите на размитата съставна преди интегрирането им и в жълто е сумарното управление на регулатора), както и изхода на обекта във виолетово и измерения изход от PLC с насложен в него шум (светло син).



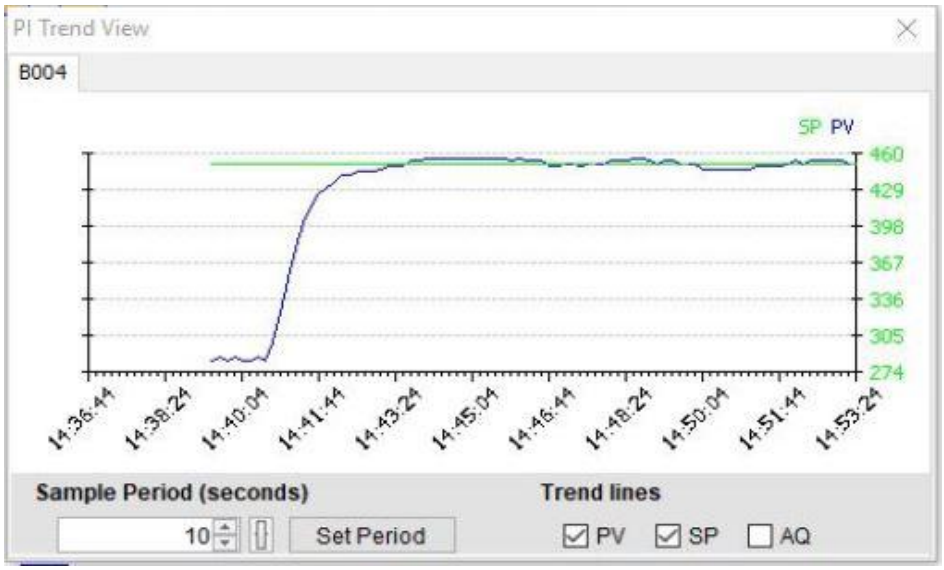
Фиг. 7. Графика на симулирания преходен процес

На фигура 8, на която са дадени в уголемен мащаб само стойностите около установената, се вижда че времето за достигане на заданието е под 70 секунди, а отклоненията след това са под 0.3 градуса (колкото е и точността на измерване).



Фиг. 8. Уголемена графика на симулирания преходен процес

След преизчисляване на реалните коефициенти, те са заложени в реалната установка за проверка на резултата. Резултатът на фигура 9 е записан в онлайн режим на продукта за програмиране на LOGO! контролера.



Фиг. 9. Графика на реалния преходен процес

3. Заключение

Въпреки симулираните случайни смущения в модела, които надали съвпадат с реалните, се вижда, че основните характеристики на симулирания и реално получения процес съвпадат с голяма точност.

Направените симулации и реални експерименти потвърждават работоспособността на използвания метод при оптимизиране на два параметъра. В случая е много важно да имаме точен модел на регулатора и обекта, за да разчитаме на коректно получени коефициенти.

Литература

[1]. MathWorks, Simulink - Users Guide R2014a, MathWorks, Inc, 2014.

[2]. Yordanova S., Robust Performance Design of Single Input Fuzzy System for Control of Industrial Plants with Time Delay, J. Transactions of the Institute of Measurement and Control, том 31, № 5, pp. 381-399, 2009.

[3]. Zigler J.; Nichols N., Optimum Settings for Automatic Controllers, в Trans. ASME, Vol 64, 1942.

[4]. Uzunov V., One Way of Tuning a Fuzzy PD Controller Implemented with PLC, *Information Technologies and Control*, *14*(4), 27-32, 2016.

[5]. Стоянов С. К., МЕТОДИ И АЛГОРИТМИ ЗА ОПТИМИЗАЦИЯ, София: Държавно издателство "Техника", 1990.

[6]. Узунов В., Моделиране и изследване на PLC-базиран размит регулатор - ГОДИШНИК НА ТЕХНИЧЕСКИ УНИВЕРСИТЕТ–ВАРНА, том 1, стр. 97-102, 2014 г

За контакти:
ас. д-р Веско Хр. Узунов
катедра „Автоматизация на производството”
Технически университет - Варна
E-mail: vuzunov@tu-varna.bg

ИЗСЛЕДВАНЕ ВЛИЯНИЕТО НА ПАРАМЕТРИТЕ НА ХИБРИДЕН ПИ РЕГУЛАТОР ВЪРХУ РАБОТАТА МУ

Веско Хр. Узунов

Резюме: Разглежда се хибриден ПИ регулатор, с конвенционална П съставна и размита И съставна, управляващ температурата на конкретен обект (разгледани в друга статия [5]). Изследва се влиянието на различни параметри на регулатора върху качеството на преходния процес. Целта е да се изследват възможностите за настройка на реален регулатор, управляващ реален обект и да се направят съответни изводи за практическата им приложимост.
Ключови думи: ПИ регулатор, размит регулатор, пропорционална съставна, интегрална съставна, пререгулиране, време на преходния процес, колебателност, интервали на размиване.

STUDY THE INFLUENCE OF PARAMETERS HIBRID PI CONTROLER ON ITS WORK

Vesko Hr. Uzunov

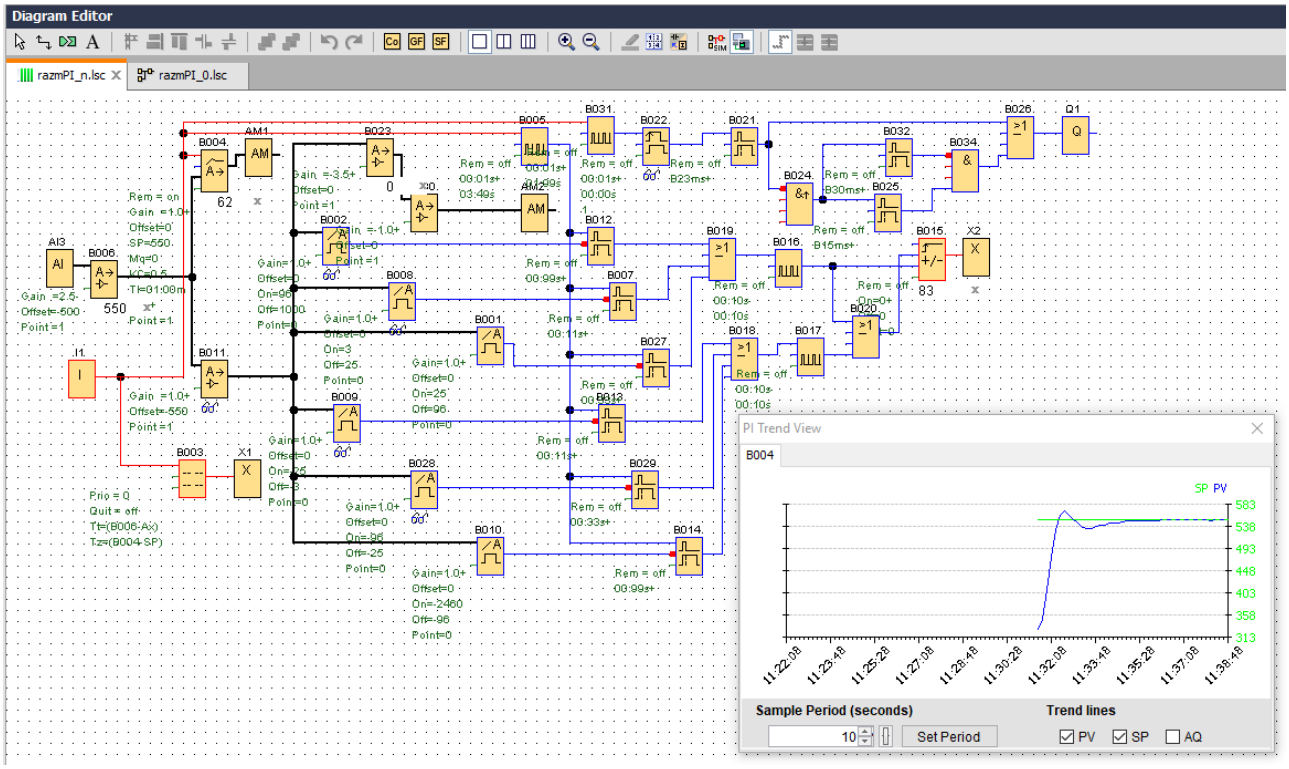
Abstract: Consider the hybrid PI controller, with conventional P component and fuzzy I constituent, controlling the temperature of a specific object (reviewed in another article [5]). The influence of various parameters of the regulator on the quality of the transition response is studied. The aim is to investigate the possibilities of setting up a real regulator controlling a real object and to draw conclusions about their practical applicability.
Keywords: PI controler, Fuzzy controler, Proportional Component, Integral Component, Overshoot, Settling Time, Volatility, Fuzzyfications intervals.

1. Въведение

За разлика от класическите регулатори, които имат 1÷3 основни параметъра за настройка, при размитите регулатори могат да се настройват много повече параметри. От една страна това дава възможност за по-добра настройка на регулатора спрямо конкретния обект и желаното от потребителя качество на преходния процес. От друга страна пък, поради голямото разнообразие на реализация и с нарастване на броя входове и изходи на регулатора, параметрите за настройка нарастват толкова много, че затрудняват самата настройка [1, 2, 3, 4, 6]. Освен това параметрите, които могат да се настройват, са специфични за конкретната програмна реализация, която освен това им налага и специфични за конкретното PLC ограничения. Поради това е изследвана само една конкретна програмна реализация на хибриден PI регулатор с конкретен температурен обект.

2. Изложение

Изследвана е структурата на хибриден PI регулатор дадена на фигура 1 със седем интервала на размиване на входната величина и независими тактови генератори на времето за интегриране (сумиране) и такта на ШИМ-а, използван за управление на обекта. Тази реализация дава най-добри възможности за настройка на регулатора, което е разгледано в предходна публикация [5].



Фиг. 1. ПИ хибриден със седем интервала на размиване на входната величина

Както вече е обяснено в [5], пропорционалната съставна на регулатора е изпълнена по класически способ и представлява грешката, получена на изхода на блок B011, умножена по коефициента на пропорционалност (Кп), който се явява коефициент на усилване на блок B023. Останалите параметри за настройка на регулатора имат следните константни стойности: Ти=3.5s (такт на интегриране); Тш=2 s (такт на ШИМ-а); Кф=1 (съгласуващ коефициент преди размиването). Таблицата на размиване не е променяна при провеждане на всички експерименти, защото се разчита на това, че тя се базира на експертни знания и е достатъчно оптимизирана. Освен това би се увеличил броят на настройваемите параметри, което би усложнило настройката на регулатора.

Направени са няколко експеримента, като е променян само коефициентът на пропорционалната част на регулатора. Графиките на получените преходни процеси са дадени на фигура 2, а параметрите на преходните процеси - в таблица 1.

Таблица 1. Параметри на преходния процес при промяна на Кп

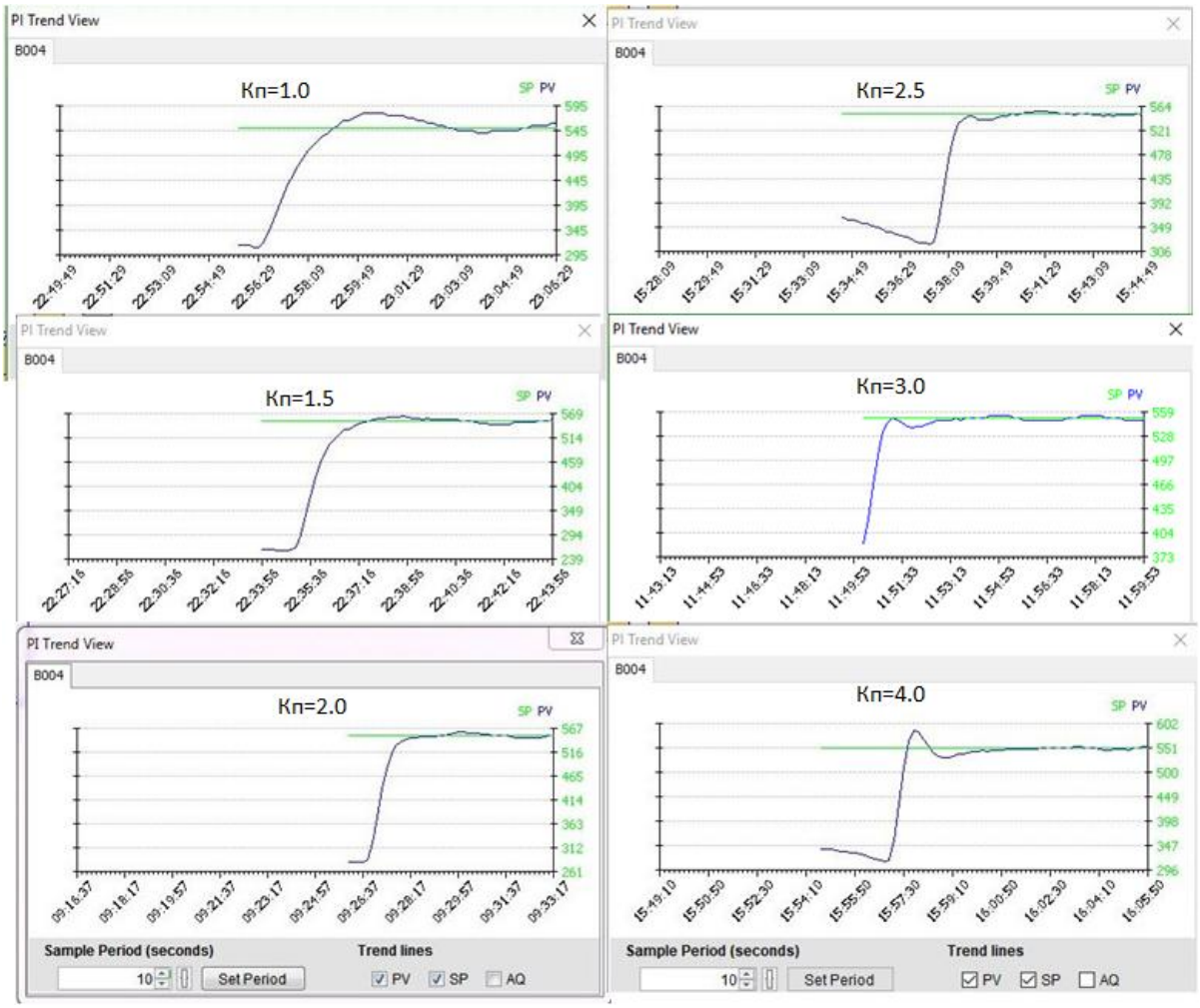
Кп	**1**	**1.5**	**2**	**2.5**	**3**	**4**
Тн(2%)[s]	140	130	80	70	60	40
Тп(2%)[s]	470	450	80	130	140	180
Пр[%]	6.3	2.7	1.8	0	0	7.2

В таблицата е дадено изменението на следните параметри на преходния процес:

- Тн – време за нарастване или за достигане на 2% преди заданието;
- Тп – време на преходния процес или време за влизане на колебанията в 2% граница около заданието;
- Пр – пререгулиране в %.

Вижда се, че от Кп еднозначно зависи времето за нарастване (Тн), като то намалява с повишаване на коефициента. Времето на преходния процес (Тп) и пререгулирането (Пр) обаче, имат оптимуми (минимуми) в този диапазон. Затова, разглеждайки и самите графики на фигура 2, можем да кажем, че най-бързият процес е този при Кп=2, който е с най-малко

време на преходния процес и практически без пререгулиране. При по-малки стойности процесът е колебателен, благодарение на засилване влиянието на интегралната съставна, а при по-големи стойности процесът отново става колебателен заради големия коефициент на пропорционалност. Това води и в двата случая до увеличаване на времето на преходния процес и пререгулирането.



Фиг. 2. Графики на преходните процеси при промяна на Кп

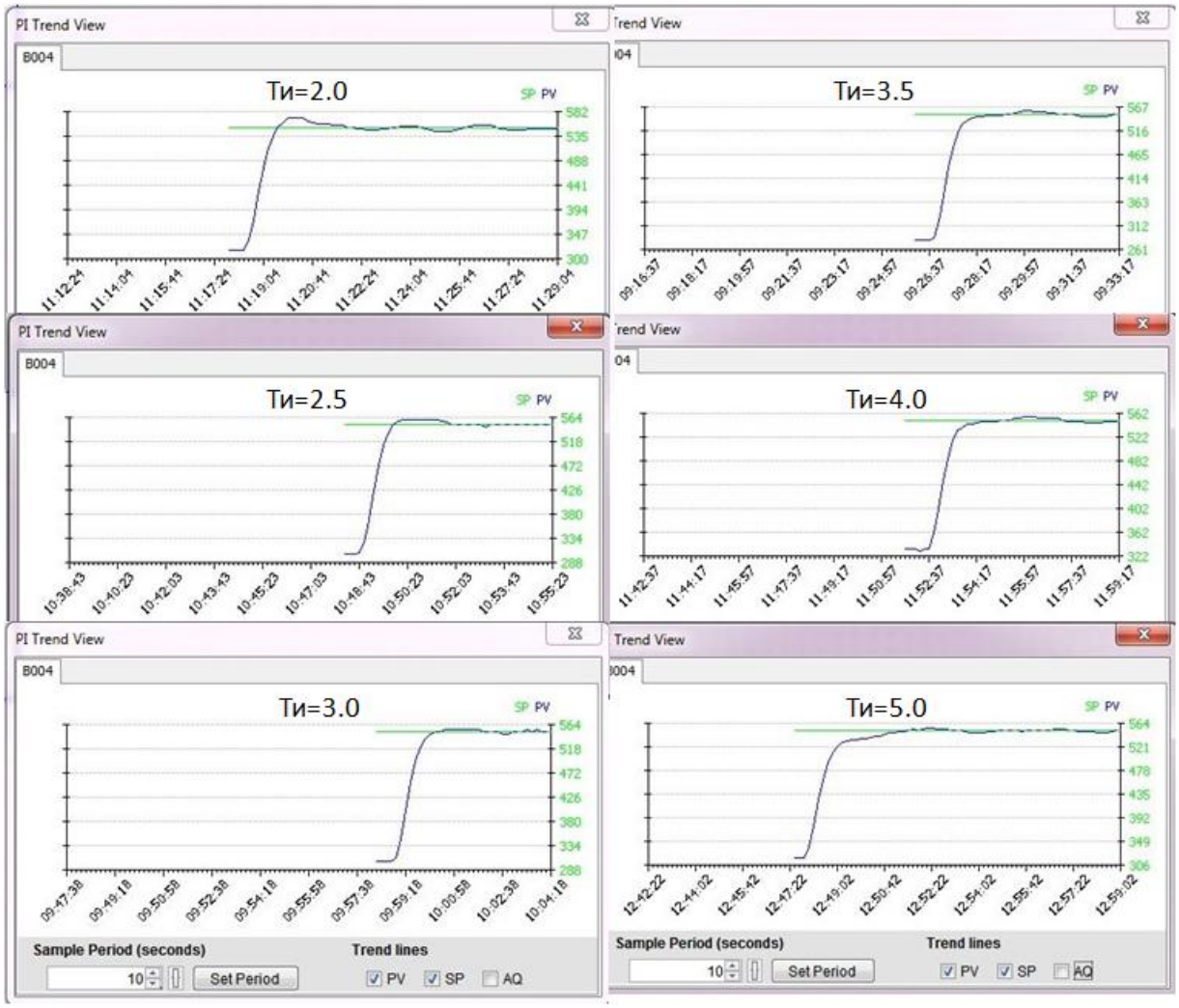
На фигура 3 и в таблица 2 се виждат графиките и стойностите на параметрите на преходния процес при изменение на такта на интегриране (Ти), при константни останали параметри на регулатора, а именно: Кп=2; Тш=2s; Кф=1.

Таблица 2. Параметри на преходния процес при промяна на Ти

Ти	**2**	**2.5**	**3**	**3.5**	**4**	**5**
Тн(2%)[s]	60	70	75	80	90	160
Тп(2%)[s]	160	70	75	80	90	160
Пр[%]	4.5	1.8	0.9	1.8	1.8	0.9

Вижда се, че от Ти еднозначно зависи времето за нарастване (Тн), като то расте с повишаване на такта на интегриране. Може да се каже, че зависимостта е обратна за перегулирането (Пр). Времето на преходния процес обаче, има оптимум (минимум) в този диапазон. Затова, разглеждайки таблица 2 и самите графики на фигура 3, можем да кажем, че най-бързият процес е този при Ти=2.5 (почти същите параметри имаме и при Ти=3.0), който е с най-малко време на преходния процес и практически без пререгулиране. При по-малки

стойности процесът е колебателен, благодарение на засилване влиянието на интегралната съставна, а при по-големи стойности процесът достига до заданието със забавяне, поради бавното нарастване на същата съставна. Това води и в двата случая до увеличаване на времето на преходния процес.



Фиг. 3. Графики на преходните процеси при промяна на Ти

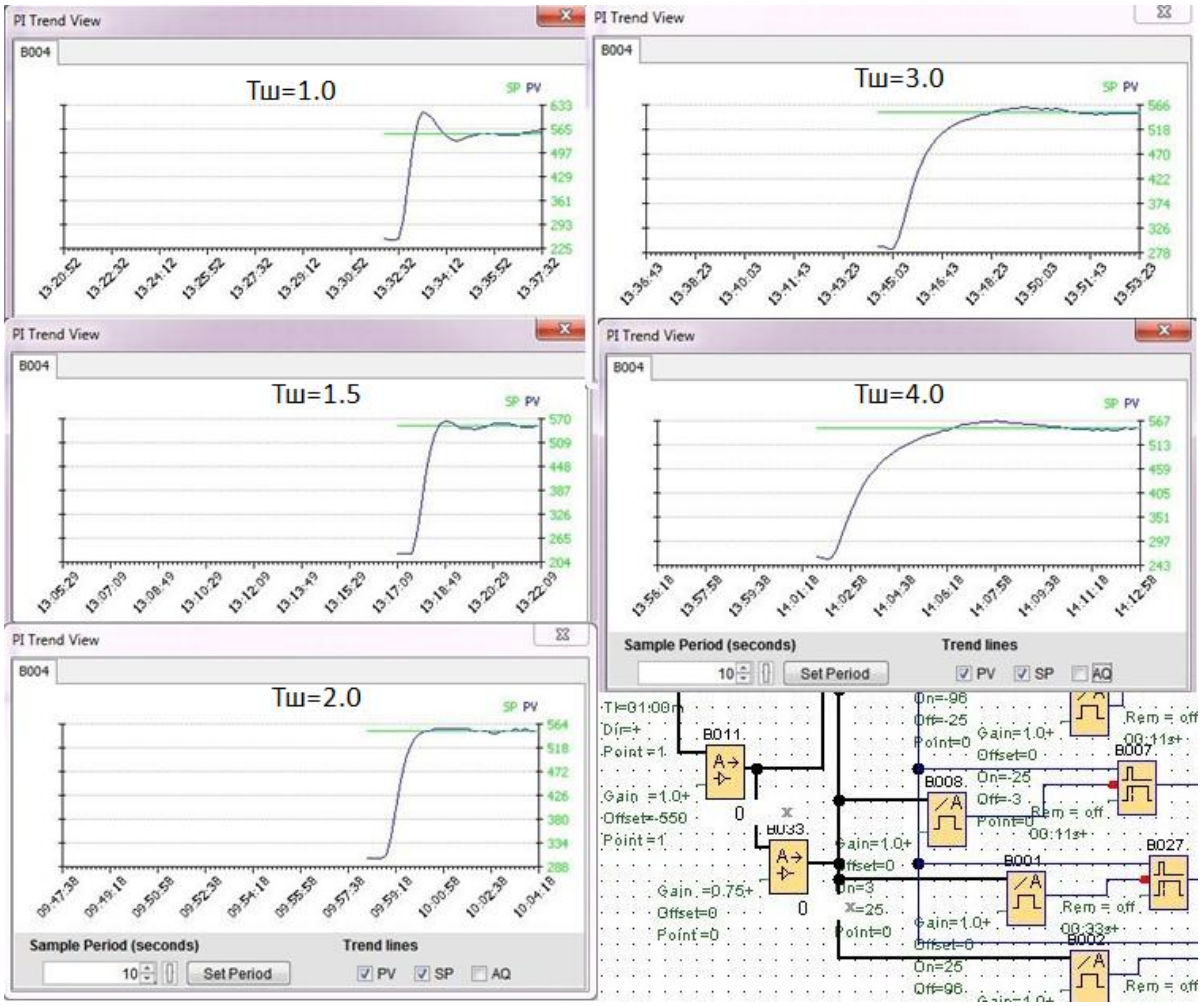
На фигура 4 и в таблица 3 се виждат графиките и стойностите на параметрите на преходния процес при изменение на такта на ШИМ-а (Тш), при константни останали параметри на регулатора, а именно: Кп=2; Ти=3s; Кф=1.

Таблица 3. Параметри на преходния процес при промяна на Тш

Тш	**1**	**1.5**	**2**	**3**	**4**	
Тн(2%)[s]	40	50	70	170	240	
Тп(2%)[s]	150	140	70	280	430	
Пр[%]	10.9	2.7	1.3	2.7	3.6	

Вижда се, че от Тш еднозначно зависи времето за нарастване (Тн), като то намалява с намаляване на такта на ШИМ-а. Времето на преходния процес (Тп) и пререгулирането (Пр) обаче, имат оптимуми (минимуми) в този диапазон. Затова, разглеждайки таблица 3 и самите графики на фигура 4, можем да кажем, че най-бързият процес е този при Ти=2, който е с най-малко време на преходния процес и практически без пререгулиране. При по-малки стойности процесът става колебателен, а при по-големи стойности процесът става твърде бавно

нарастващ. Това води и в двата случая до увеличаване на времето на преходния процес и пререгулирането. На практика Кш=1/Тш (коефициент на запълване на ШИМ) влияе като коригиращ коефициент на вече полученото управление, без да променя съотношението на двете съставни на регулатора.



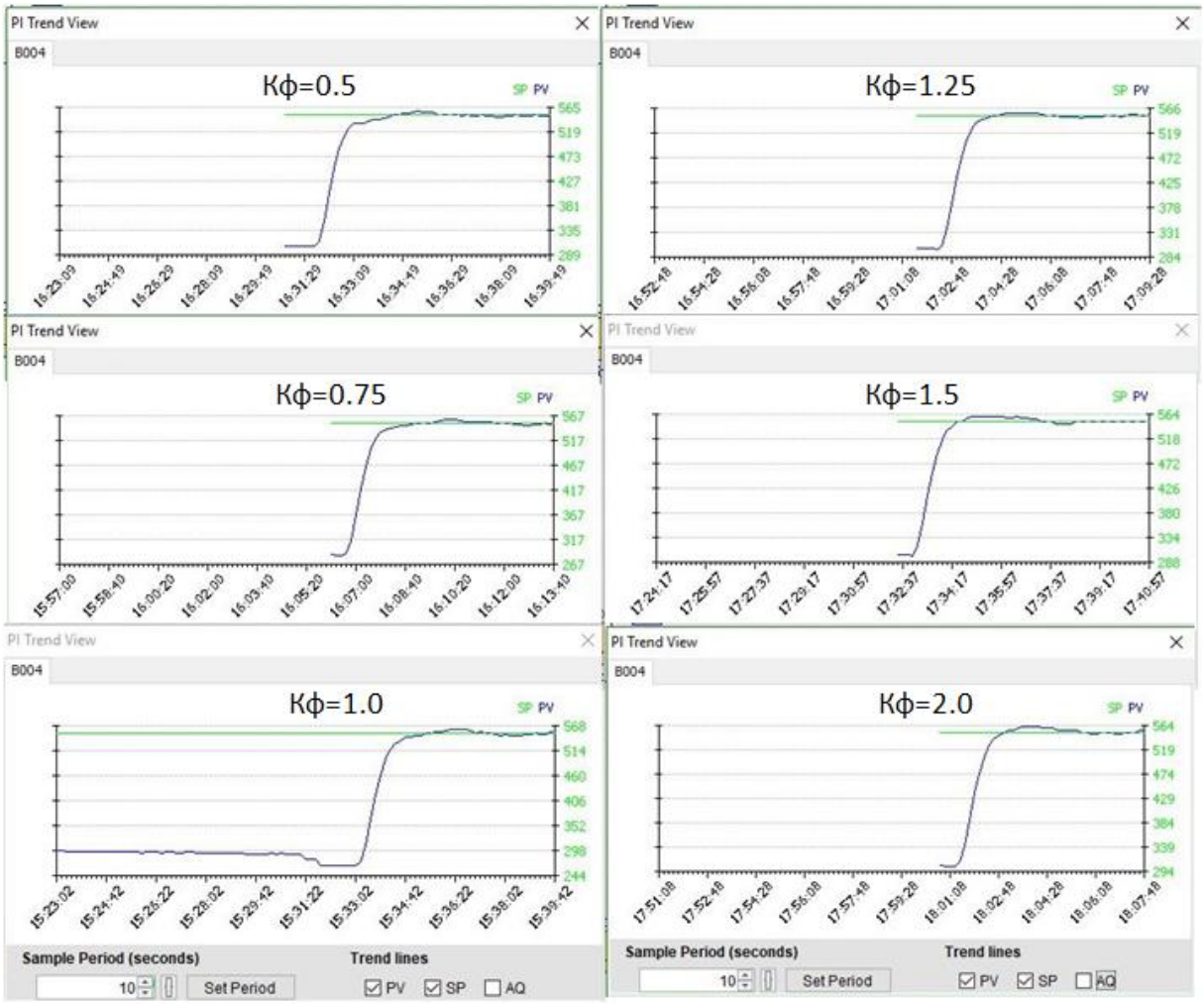
Фиг. 4. Графики на преходните процеси при промяна на Тш

На фигура 5 и в таблица 4 се виждат графиките и стойностите на параметрите на преходния процес при изменение на коригиращия коефициент на грешката преди размиване (Кф), при константни останали параметри на регулатора, а именно: Кп=2; Ти=3s; Тш=2s. За да се променя по-лесно този коефициент, е въведен блок B033, коефициентът на усилване на който представлява Кф. Може и без този нов блок, но това би означавало да променяме едновременно коефициентите на шестте компаратора след него, а така променяме само един коефициент, което е по-лесно. Този коефициент на практика свива или разтегля интервалите на размиване.

Таблица 4. Параметри на преходния процес при промяна на Кф

Кф	**0.5**	**0.75**	**1**	**1.25**	**1.5**	**2**
Тн(2%)[s]	140	130	120	100	80	90
Тп(2%)[s]	140	130	120	100	230	210
Пр[%]	1.8	1.8	1.8	0.9	2.7	3.2

Вижда се, че от Кф еднозначно зависи времето за нарастване (Тн), като то намалява с увеличаване на коефициента. Пререгулирането (Пр) почти не се променя при Кф=0.5 до 1.25. Времето на преходния процес (Тп) обаче има оптимум (минимум) в този диапазон. Затова, разглеждайки таблица 4 и самите графики на фигура 5, можем да кажем, че най-бързият процес е този при Кф=1.25, който е с най-малко време на преходния процес и практически без пререгулиране. На практика при Кф=0.5 интервалите са най-разтеглени. Например интервалът 0.3-2.5 градуса става 0.6-5.0 реални градуси. Това води до по-ранно влизане на грешката в по-вътрешните интервали, кадето коефициентът на интегриране намалява, което пък довежда до по-бавното нарастване. От друга страна се увеличава зоната на нечуствителност на регулатора, което води до по-голяма колебателност в установен режим. При свиване на интервалите (увеличаване на коефициента) се повишава бързодействието, но и пререгулирането.



Фиг. 5. Графики на преходните процеси при промяна на Кф

3. Заключение

В заключение можем да кажем, че коефициентът на пропорционалност Кп най-съществено влияе на бързодействието на преходния процес. Като с повишаването му расте бързодействието, но от друга страна расте и пререгулирането. Това е характерно и за класическите ПИ регулатори. Лошото при конкретната цифрова реализация е, че с увеличаване на коефициента расте и стъпката на дискретизация по ниво на изходния

управляващ сигнал, което при много високи коефициенти води до некоректно управление. Това може да се компенсира в известна степен от интегриращата съставна.

Тактът на интегриране Ти оказва най-съществено влияние на колебателността на процеса (пререгулирането). С неговото намаляване пада и пререгулирането. Това обаче намалява бързодействието, но значително по-бавно.

Тактът на ШИМ-а (Тш) също влияе съществено на времето за нарастване (Тн), като зависимостта е право пропорционална, но пък е почти в обратно пропорционална зависимост с времето на преходния процес. Затова би следвало да се използва за съгласуване на управлението с конкретния обект, като се търси оптимумът му, но не и за настройка на преходния процес.

Съгласуващият коефициент преди размиването (Кф) влияе най-съществено и еднопосочно на времето за нарастване. Времето на преходния процес и пререгулирането имат минимум. Вижда се, че за конкретния обект той не е при коефициент 1 а при 1.25, което показва, че диапазонът на интервалите на размиване за конкретния обект трябва да се свият малко, за да се получат най-добри резултати (минимум на времето на преходния процес и минимум на пререгулирането). Този коефициент може да се използва за съгласуване на конкретния измерван сигнал от обекта с интервалите за размиване, като се търси оптимумът.

В крайна сметка може да се каже, че за настройката на параметрите на преходния процес е най-добре да се използват първите два параметъра на регулатора, но за да се настрои той оптимално спрямо канкретния обект, трябва да се оптимизират и вторите два параметъра.

Литература

[1]. Kandel A., G. Langholz, Fuzzy Hardware. Architectures and Applications, Boston: Kluwer, 1998.

[2]. Petrov Michail, Ivan Ganchev, Albena Taneva, Fuzzy PID control of nonlinear plants, in *Intelligent Systems - First International IEEE Symposium*, 2002.

[3]. Kiam Heong Ang, G. Chong,, "PID Control System Analysis, Design, and Technology," *IEEE Transactions on Control Systems Technology,* vol. 13, no. 4, pp. 559 - 576, 13(4):559 - 576 · August 2005 2005.

[4]. Gotwald S., W. Pedrycz, "Problems of the design of fuzzy controllers," in *Approximate Reasoning in Expert Systems,* Amsterdam, Nort-Holland, 1985, pp. 393-405.

[5]. Uzunov V., "DEVELOPMENTOF A PLC-BASED HYBRID PI CONTROLLER," in *"Applied Computer Technologies" ACT 2018*, Ohrid, 2018..

[6]. Yordanova S., "Robust Performance Design of Single Input Fuzzy System for Control of Industrial Plants with Time Delay," *J. Transactions of the Institute of Measurement and Control,* vol. 31, no. 5, pp. 381-399, 2009.

За контакти:
ас. д-р Веско Хр. Узунов
катедра „Автоматизация на производството"
Технически университет - Варна
E-mail: vuzunov@tu-varna.bg

СЕКЦИЯ 4
СТУДЕНТИ И ДОКТОРАНТИ

Пета научна конференция с международно участие
"Компютърни науки и технологии"
28 - 29 септември 2018г.
Варна, България

Fifth International Scientific Conference
Computer Sciences and Engineering
28 - 29 September, 2018
Varna, Bulgaria

SECTION 4
STUDENTS AND PhD STUDENTS

САМООБУЧАВАЩИ АЛГОРИТМИ, ИЗПОЛЗВАНИ В МЕДИЦИНАТА

Теодора Х. Христева, Мария П. Маринова

Резюме: Една от основните задачи в биоинформатиката е класифицирането и предвиждането на биологични данни. Поради ускореното нарастване на обема на тези данни е важно процесите да бъдат автоматизирани. Алгоритмите за машинно самообучение се справят успешно при решаването на класификационни проблеми. Затова използването на алгоритми като Support Vector Machines (SVMs) е изключително често при анализирането на генетичните данни.
Ключови думи: Машинно самообучение, алгоритми, Support Vector Machine, DeepBind

Self-learning Algorithms Used In Medicine

Teodora H. Hristeva, Maria P. Marinova

Abstract: One of the main tasks in bioinformatics is to classify and predict biological data. Due to the accelerated increase in the volume of these data, it is important that processes are automated. Machine learning algorithms are used successfully in solving classification problems. Therefore, using algorithms such as Support Vector Machines (SVMs) is very common in analyzing genetic data.
Keywords: Machine learning, algorithms, Support Vector Machine, DeepBind

1. Увод

Развитието на технологиите и приложението им в различни области на медицината води до натрупване на огромно количество биологични данни. Експоненциалните темпове, с които растат генетичните бази данни, са следствие от автоматизирането на редица биологични експерименти и това довежда до необходимостта от прилагане на различни софтуерни решения, които да се справят с тези проблеми бързо и ефикасно. Биоинформатиката като дял от медицината разчита изключително на различни алгоритми, които да обработват събраната информация. Това, от своя страна, води до натрупване на още данни. Рационалното им използване с цел да се синтезират нови знания е от изключителна важност за учените и инженерите, работещи в тази област, както и в областта на програмирането и информатиката.

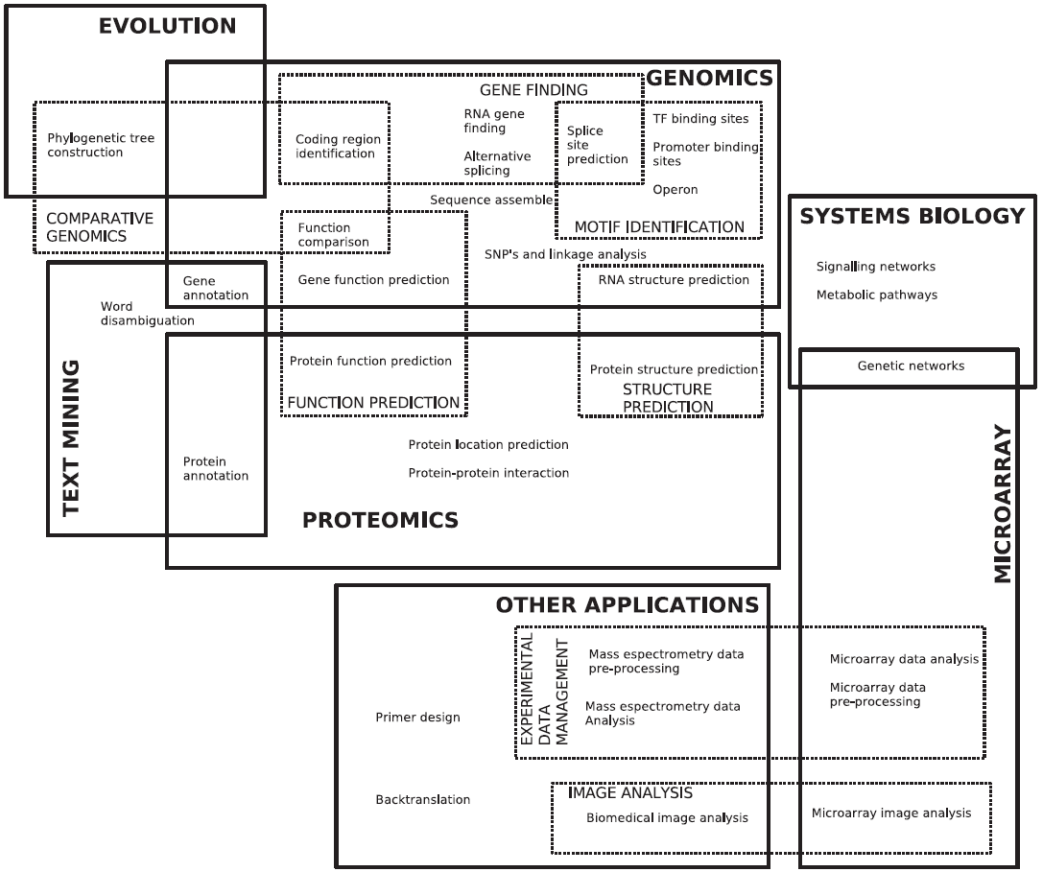
Машинното самообучение (МС) е дял от компютърните науки, който дава възможност на компютрите да учат, без да бъдат програмирани за това. МС изследва проучването и изграждането на интелигентни софтуерни системи, чрез които става възможно решаването на проблеми, обикновено считани в компетенцията на хората, с изчислителни средства.

Алгоритмите за МС се справят успешно в ситуации, при които се изисква откриване на различни модели на поведение, взаимовръзки и зависимости между различни елементи в дадено множество от данни. С тяхна помощ се генерира код, който позволява разпознаването на тези модели в нови данни.

Биоинформатиката е интердисциплинарна научна област, която се занимава с изследването на данни от различни сфери на биологията със средствата на информатиката. Тя силно разчита на разработване на алгоритми и софтуер, които да могат да бъдат използвани както за обработка на постъпващата информация, така и за извършване на числови експерименти върху нея.

Съществуват няколко дяла в биоинформатиката, където машинното самообучение намира своето приложение в извличането на информация. Фигура 1 показва схематично основните биологични проблеми, върху които се прилагат различни изчислителни методи. Приложението на МС е класифицирано в шест различни области – геномика (Genomics), протеомика (Proteomics), микромасиви (Microarrays), системна биология (System biology), еволюлия (Evolution) и текст майнинг (Text mining). В категорията „Other Applications" са изнесени всички останали проблеми.

Геномиката е един от най–важните дялове в биоинформатиката. Броят на достъпните поредици и бази в GenBank от създаването ѝ през 1982 досега се увеличава експонециално. Тези данни се нуждаят от обработване, за да бъде извлечена полезна информация от тях. Като първа стъпка в този процес е извличането на местоположението и структурата на гените от поредиците на даден геном. Освен това се извършва идентификация на регулиращи елементи, както и на некодиращи РНК гени. Информацията, получена от поредиците, може да се използва за предвиждане на генни функции и вторични РНК структури.



Фиг. 1. Области на приложение на МС в биоинформатиката [1]

Гените кодират информация за това как се синтезират определени протеини. Протеините играят изключително важна роля в биологичните процеси на организмите и триизмерната им структура е ключов фактор в изучаването на тяхната функционалност. В областта на протеомиката основното приложение на машинното самообучение е в предиждането на протеиновите структури. Протеините са изключително сложни макромолекули, изградени от хиляди атоми и връзки, поради което броят на възможните структури е огромен. Това прави предвиждането им сложен комбинаторен проблем, при който различни отпимизиращи техники са задължителни. В протеомиката, както и в геномиката, машинното самообучение се прилага при предвиждането на протеиновите функции.

Един от най–използваните представители на надзираваното машинно самообучение е т. нар. Support Vector Machine алгоритъм (SVM) или в буквален превод – машина на поддържащите вектори. Може да се използва както за класификация и регресия, така и за откриване на отклонения (outlier detection), което го прави изключително гъвкав. Създаден е да работи със сложни, но малки до средни по размер обеми от данни. Ефективен е дори при много голям брой свойства. Като всеки друг алгоритъм от своя тип, целта на SVM е да създаде модел въз основа на обучаващо множество, който да е в състояние да определи класа, с възможно най–голяма точност, на елементите от тестовото множество.

За да може SVM да класифицира елементите на дадено множесто, е необходимо да се намери решение на следната задача: При зададен вход X от вида:

$$X \subseteq \mathbb{R}^d, \text{ където } d \in \mathbb{N}; \text{ изход } Y = \{-1, +1\}$$

и обучаващо множество: $S = \{(x_1, y_1), (x_2, y_2) \dots (x_n, y_n)\} \in (X \cdot Y)$,

да се намери функция, която да напасне всеки елемент X в Y.

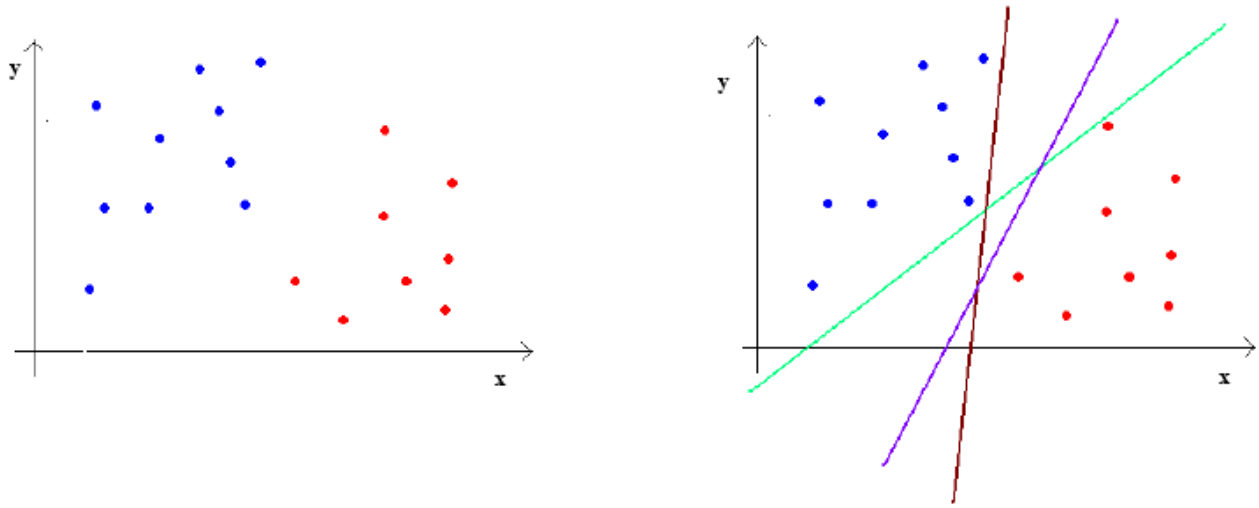
В контекста на класификацията, това означава да се намери такова правило, което при зададен елемент $x \in X$, да може да се определи към кой клас принадлежи (-1 или +1). Можем да построим линеен бинарен класификатор от вида:

$f: x \subseteq \mathbb{R}^N \to R$, от което следва правилото:

$f(x) > 0$, то x = -1, иначе x = +1. Следователно може да бъде изведена следната линейна функция:

$$f(x) = (w.x) + b; \text{ където } w \in R^d \quad (1)$$

Визуализирането на (1) в геометричното пространство представлява хиперравнина от всички възможни равнини, които разделят коректно входните данни в S множеството, както е показано на фигура 2.



Фиг. 2. Построяване на хиперравнина

2. Изложение

Националният център за биотехнологична информация на САЩ (на английски: National Center for Biotechnological Information, NCBI) е централен институт за обработка и съхранение на данни за молекулярна биология.

NCBI предоставя информация за бази данни на ДНК и РНК, бази данни за статии научна литература и таксономична информация, обезпечава търсене на данни за конкретни биологични видове, а също така съдържа различни програми по биоинформатика.

Основни задачи на NCBI са:

- Създаване на аавтоматизирани системи за анализ и съхранение на данни по молекулярна биология, биомедицина и генетика;

- Компютърна обработка на данни, получени при изследвания на биологически активни молекули и вещества;

- Съдействие за широко използване на базите данни и програмно обезпечаване и подпомагане на изследователи в областта на биотехнологията и медицински работници;

- Координиране събирането на биотехнологична информация по целия свят.

Протеините, които се свързват към ДНК и РНК играят централна роля в регулацията на гените, включително и процесите на транскрипция и сплайсинг. Протеин – кодиращите части в поредиците най–често се откриват с матрици на претеглените позиции (Position weight matrices (PWM)), които лесно се интерпретират и могат да бъдат приложени върху цели геноми за откриване на потенциални свързващи сектори.

Съществуват различни начини на добиване на информация за гените и техните функции. Например, при изследване с микромасиви данните, които се получават, представляват коефициент на съответствие за всяка проба, която е изследвана. Различните техники на изследване целят специфични проблеми и водят до получаване на разнородни данни.

DeepBind е софтуер, който използва конволюционна невронна мрежа, за да открива нови модели (шаблони), чиито места в поредиците са неизвестни – задача, за чието решение са необходими огромни количества от данни, ако бъде използвана традиционна невронна мрежа.

DeepBind може да използва както данни от микромасиви, така и от ДНК поредици. Може да бъде обучаван едновременно с милиони поредици чрез паралелна имплементация върху графични процесори. Успява да толерира умерена стойност на шум в данните, както и неетикирани данни.

Резултатът от обучението представлява числова оценка на свързване между зададения модел (ген) и изследваната поредица. Поредиците могат да бъдат с различна дължина от 14 до 101 нуклеотида, като оценките могат да бъдат дробни числа или дискретни стойности.

След приключване на оценяването, данните се конвертират до CSV формат, с цел по – удобна работа.

Транскрипционните фактори (Transcription Factors) – ТФ са протеини, които позволяват на определени гени да бъдат „включени“ или „изключени“. Те се наричат *активатори,* когато усилват транскрипцията на гените и *репресори*, когато я забавят. Участъците от ДНК, където са разположени ТФ, се наричат съответно *подобрители* и *заглушители*. ТФ позволяват на клетките да извършват „логически“ операции и да комбинират различни източници на информация, за да „решат“ да изразят даден ген или не. Клетката *изразява* определен ген, когато произвежда протеина, кодиран в него. За да се осъществи това, е необходимо да се премине през строго определен алгоритъм от стъпки, първата от които е *транскрипцията.*

Транскрипция се нарича процесът на копиране на ДНК ген в РНК молекула. За да бъде един ген транскриптиран е необходимо специален ензим, наречен *РНК полимераза,* да се прикрепи към ДНК на гена. Мястото, към което се закрепя ензимът, се нарича промотър. Промотърът инициира началото на транскрипцията.

За разлика от бактериите, при хората и останалите еукариоти РНК полимеразата може да се прикрепи към промотър единствено с помощта на друг протеин - транскрипционен фактор. Този факт обославя изключително важното значение на този вид протеини. Те играят съществена роля при активирането и репресирането на гени, което от своя страна води до редица верижни реакции в случай на липса на определен ТФ, например.

TCF7L2 или TCF4 е протеин, действащ като транскрипционен фактор, който при хората е кодиран в TCF7L2 гена. ТФ, който се кодира, указва влияние върху няколко

биологични пътища. Еднонуклеотидният полиморфизъм – ЕНП, който се наблюдава в TCF7L2 гена, до днешна дата е генетичният маркер с най–голямо значение при диагностицирането на диабет тип 2. Известно е, че ЕНП в този ген водят до висок риск от развиване на гестационен диабет и множество други болести.

TCF7L2 е ТФ, който влияе върху транскрипцията на няколко гена, затова и упражнява влияние върху различни функции в клетката. Свързва се с друг протеин – бетакатенин, с който правят двустранен ТФ, указващ влияние върху предаването на сигнали в клетката. Ролята на гена в глюкозния метаболизъм е изразена в много тъкани, изграждащи вътрешните органи, мозък и скелетните мускули. Въпреки това, TCF7L2 не регулира пряко процеса на разграждане на въглехидратите в бета клетките, но регулира усвояването на глюкозата в тъканите на панкреаса и черния дроб.

Генът, кодиращ TCF7L2 протеин, се намира в хромозома 10q25.2-q25.3. Състои се от 16 екзона, като 5 от тях са алтернативни. Съдържа 619 аминокиселини и молекулярната му маса е 67919 Da.

Целта на задачата е да реализира алгоритъма Support Vector Machine върху данни, получени от NCBI и обработени предварително с DeepBind софтуера. Моделът, обучен в експерименталната част, да може да разпознава кодиращите области от ДНК, където се намира TCF7L2 генът и следователно да класифицира дадена поредица като потенциално кодираща TCF7L2 протеин. За целта оценките от DeepBind ще помогнат за определянето на кодиращите области.

Входният формат на данните е CSV файл, съдържащ поредици, изтеглени от базата данни на NCBI и числовите стойности на оценките от DeepBind. Файлът има следната структура, показана на фигура 3.

	A	B
1	Sequence	TCF7L2
2	ATTAAGGCAGTGTGTTCCTCTCGCCCTGTCAATAATCTCCGCTCCCAGACTA	-0.4176
3	CGATCCCCCTTTTTCTATCTGTCAATCAGCGCCGCCTTTGAACTGAAAAGCT	3.273916
4	CACTCAAATCCGAGCGGCACGAGCACCTCCTGTATCTTCGGCTTCCCCCCC	1.470267
5	ACTTCTTGTTGTTGTTGGTGTTTTTTTTTTTTTTACCCCCCTTTTTTATTTATTAT	0.242704
6	ATCGGATCCTTGGGAACGAGAGAAAAAAGAAACCCAAACTCACGCGTGC	0.231449
7	CCTCCCCTCCTCCCTCTTTTCCCCTCCCCAGGAGAAAAAGACCCCCAAGCAC	-0.31313
8	CTCGTCTTTTTCTTGCAATATTTTTTGGGGGGGCAAAACTTTTTGGGGGTGAT	-0.982

Фиг. 3. Суров вид на оценените данни

Алгоритъмът за машинно самообучение SVM работи с числови данни. Това налага кодирането на базите в поредиците до числово представяне. Съществуват различни методи за кодиране на нуклеотидите. За целите на задачата е избран метод, при който се използват електрон – йонните потенциали на взаимодействие *(от англ. Electron – ion Interaction Pseudopotential EIIP)* на базите. Методът се състои в заменяне на символния израз на нуклеотида с неговия потенциал. Това води до значително намаляване на изчислителното натоварване с до 75% при всички изследвани случаи. Резултатът от това конвертиране е 70–мерен масив за всяка поредица. Това би било проблем в случай на използване на друг алгоритъм. SVM обаче, е пригоден именно за такива случаи.

Класификацията със SVM е алгоритъм за надзиравано МС, което означава, че освен входните данни е необходимо да бъде и подаден и очакваният резултат. Оценките от DeepBind са с различни стойности и не могат да бъдат използвани като етикети. Освен това, SVM очаква етикетите да бъдат представени в два класа: 1 и -1. Това налага резултатите да бъдат предварително клъстеризирани в две части. Етикетите са получени след сравнението на всяка стойност с предварително зададена „*проба*“. В конкретния случай тя е със стойност

0,99, което означава, че всички оценки, по–високи от нея, ще бъдат причислени към +1 класа, а всички под нея - към -1.

SVM зависи силно от стойностите на входните данни. Ако те са в голям диапазон или съдържат разнородни стойности, това би довело до редица грешки и неправилно обучение на модела. За да бъдат постигнати максимални резултати, е необходимо стойностите да варират около 0, както и средната им стойност също. Подходящият вид на данните се постига чрез няколко трансформации, след които средната стойност е $1{,}3755113e^{-6}$

Обучението на модела се изразява в намиране на такива хипер – параметри, при които моделът има най–висока оценка. Хипер – параметрите не могат да бъдат генерирани автоматично от алгоритъма, а трябва да бъдат подадени като аргументи.

3. Резултати

В настоящата задача са изследвани и класифицирани данните, получени от DeepBind. Обект на експерименталната част е генът TCF7L2, който се намира в хромозома 10 от човешкия геном.

Хромозома 10 е една от 23-те двойки хромозоми при хората, както и 10-тата най-голяма. Съдържа около 133 милиона нуклеотидни двойки и представлява около 4.5% от цялата ДНК в клетките. Доказано е, че мутации в гена на TCF7L2 могат да повлият за образуването на ракови образования в дебелото черво, диабет от тип 2 и други.

Целта на изложения задача е да се обучи модел, който да може да определя по зададена ДНК поредица с дължина 70 бази, дали в нея се намират кодиращи области на TCF7L2 гена. За целта от базата данни на NCBI се изтеглят подравнените поредици, кодиращи определения ген. Поредиците се оценяват с помощта на DeepBind. DeepBind връща оценка в интервала (-1; +1), която оценка означава вероятността в проценти, която съществува в тази част да се кодира съответният ген. Използвайки тези оценки, се създават етикети за всяка поредица, като оценките над 0,99 се етикират с +1, а всички останали - с -1. Поредиците се кодират по EIIP метода, след което се нормализират и с тях се обучава модел. Най-подходящият модел се избира след сравнението на два вида крос валидация. Данните се прочитат от DeepBind.

Функцията describe() показва основна статистическа информация за числовите данни в заредения DataFrame. В конкретния случай са показани средна стойност, стандартно отместване, минимална и максимална стойност на оценките от DeepBind. Числовите данни срещу 25%, 50% и 75% показват вътрешното съотношение между числата в масива.

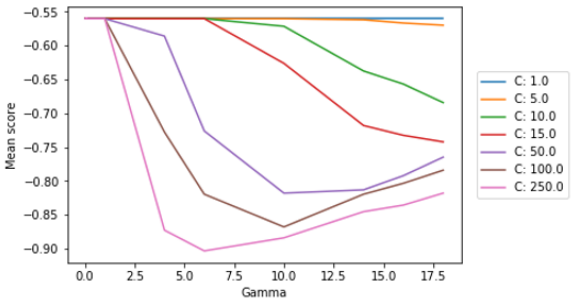
SVM зависи силно от стойностите на входните данни. Ако те са в голям диапазон или съдържат разнородни стойности, това би довело до редица грешки и неправилно обучение на модела. За да бъдат постигнати максимални резултати, е необходимо стойностите да варират около 0, както и средната им стойност също. Подходящият вид на данните се постига чрез няколко трансформации.

Обучението на модела се изразява в намиране на такива хипер – параметри, при които моделът има най–висока оценка. Хипер – параметрите не могат да бъдат генерирани автоматично от алгоритъма, а трябва да бъдат подадени като аргументи. Sklearn библиотеката разполага с автоматизирани техники за търсене на хипер – параметри, наречени крос – валидация. В изследването са използвани двете най–широко използвани – GridSearchCV и RandomSearchCV.

GridSearchCV обучава модел за всяка възможна комбинация от подадени параметри. В експерименталната част са изпозлвани две мрежи – с линейно ядро и RBF ядро. Оценките от всички модели се сравняват и най–добрият (с най–висока точност) се запазва.

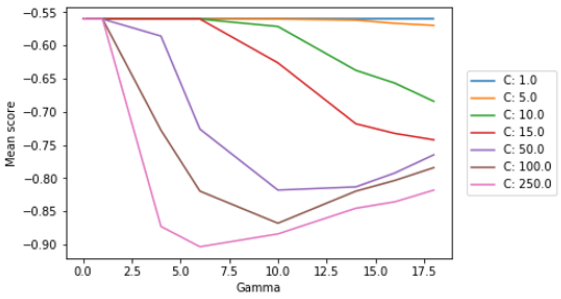
RandomSearchCV подбира опрделен брой кандидати от зададен диапазон на параметрите с определена дистрибуция, като начинът на селектиране на модел е аналогичен с този на GridSearchCV.

В края на експеримента двата най–добри модела се прилагат върху цялото тренировъчно множество. По–добрият бива избран и тестван още веднъж върху тестовото множество.




```
grid_search.best_params_

{'C': 1.0, 'gamma': 0.1}

mean: -0.56039, std: 0.00324, params: {'C': 1.0, 'gamma': 0.025},
mean: -0.56039, std: 0.00324, params: {'C': 1.0, 'gamma': 1.0},
mean: -0.56039, std: 0.00324, params: {'C': 1.0, 'gamma': 4.0},
mean: -0.56039, std: 0.00324, params: {'C': 1.0, 'gamma': 6.0},
mean: -0.56039, std: 0.00324, params: {'C': 1.0, 'gamma': 10.0},
mean: -0.56039, std: 0.00324, params: {'C': 1.0, 'gamma': 14.0},
mean: -0.56039, std: 0.00324, params: {'C': 1.0, 'gamma': 16.0},
mean: -0.56039, std: 0.00324, params: {'C': 1.0, 'gamma': 18.0},
mean: -0.56039, std: 0.00324, params: {'C': 5.0, 'gamma': 0.1},
mean: -0.56039, std: 0.00324, params: {'C': 5.0, 'gamma': 0.025},
mean: -0.56039, std: 0.00324, params: {'C': 5.0, 'gamma': 1.0},
mean: -0.56039, std: 0.00324, params: {'C': 5.0, 'gamma': 4.0},
mean: -0.56039, std: 0.00324, params: {'C': 5.0, 'gamma': 6.0},

Fitting 5 folds for each of 63 candidates, totalling 315 fits
[Parallel(n_jobs=2)]: Done  37 tasks       | elapsed:   14.4s
[Parallel(n_jobs=2)]: Done 158 tasks       | elapsed:  1.1min
[Parallel(n_jobs=2)]: Done 315 out of 315 | elapsed:  2.7min finished

GridSearchCV(cv=5, error_score='raise',
       estimator=SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,
  decision_function_shape='ovr', degree=3, gamma='auto', kernel='rbf',
  max_iter=-1, probability=False, random_state=None, shrinking=True,
  tol=0.001, verbose=False),
       fit_params=None, iid=True, n_jobs=2,
       param_grid={'C': [1.0, 5.0, 10.0, 15.0, 50.0, 100.0, 250.0], 'gamma': [0.1, 0.025, 1.0, 4.0, 6.0, 10.0, 14.0, 16.0, 18.0]},
       pre_dispatch='2*n_jobs', refit=True, return_train_score='warn',
       scoring='neg_mean_squared_error', verbose=2)
```

Фиг. 4. Крос – валидация с мрежово търсене (GridSearchCV)




```
grid_search.best_params_

{'C': 1.0, 'gamma': 0.1}

mean: -0.56039, std: 0.00324, params: {'C': 1.0, 'gamma': 0.025},
mean: -0.56039, std: 0.00324, params: {'C': 1.0, 'gamma': 1.0},
mean: -0.56039, std: 0.00324, params: {'C': 1.0, 'gamma': 4.0},
mean: -0.56039, std: 0.00324, params: {'C': 1.0, 'gamma': 6.0},
mean: -0.56039, std: 0.00324, params: {'C': 1.0, 'gamma': 10.0},
mean: -0.56039, std: 0.00324, params: {'C': 1.0, 'gamma': 14.0},
mean: -0.56039, std: 0.00324, params: {'C': 1.0, 'gamma': 16.0},
mean: -0.56039, std: 0.00324, params: {'C': 1.0, 'gamma': 18.0},
mean: -0.56039, std: 0.00324, params: {'C': 5.0, 'gamma': 0.1},
mean: -0.56039, std: 0.00324, params: {'C': 5.0, 'gamma': 0.025},
mean: -0.56039, std: 0.00324, params: {'C': 5.0, 'gamma': 1.0},
mean: -0.56039, std: 0.00324, params: {'C': 5.0, 'gamma': 4.0},
mean: -0.56039, std: 0.00324, params: {'C': 5.0, 'gamma': 6.0},

Fitting 5 folds for each of 63 candidates, totalling 315 fits
[Parallel(n_jobs=2)]: Done  37 tasks       | elapsed:   14.4s
[Parallel(n_jobs=2)]: Done 158 tasks       | elapsed:  1.1min
[Parallel(n_jobs=2)]: Done 315 out of 315 | elapsed:  2.7min finished

GridSearchCV(cv=5, error_score='raise',
       estimator=SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,
  decision_function_shape='ovr', degree=3, gamma='auto', kernel='rbf',
  max_iter=-1, probability=False, random_state=None, shrinking=True,
  tol=0.001, verbose=False),
       fit_params=None, iid=True, n_jobs=2,
       param_grid={'C': [1.0, 5.0, 10.0, 15.0, 50.0, 100.0, 250.0], 'gamma': [0.1, 0.025, 1.0, 4.0, 6.0, 10.0, 14.0, 16.0, 18.0]},
       pre_dispatch='2*n_jobs', refit=True, return_train_score='warn',
       scoring='neg_mean_squared_error', verbose=2)
```

Фиг. 5. Крос – валидация със случайно търсене (RandomizedSearchCV)

На фигура 4 и фигура 5 са показани резултатите от GridSearchCV и съответно RandomSearchCV. Моделът постига добри резултати в двата случая, което доказва, че алгоритъмът SVM се справя отлично с този вид данни. Най-добрият резултат за GridSearchCV е 0.962657, а за RandomSearchCV е 0.966183.

4. Заключение

Машинното самообучение е дял от компютърните науки, при който, чрез прилагането на различни математически похвати, става възможно добиването на нови знания от събрана вече информация. Една от основните задачи в биоинформатиката е класифицирането и предвиждането на биологически данни. С ускореното увеличаване на техния обем е важно тези процеси да бъдат автоматизирани. Алгоритмите за машинно самообучение се справят успешно при решаването на проблеми, за които често няма идеално решение или изисква твърде много усилия и поддръжка. Затова използването на алгоритми като Support Vector Machines (SVMs) е изключително често при анализирането на генетичните данни.

Support Vector Machines е алгоритъм за машинно самообучение, който се справя еднакво добре с проблеми, свързани с класификация и с регресия. В биоинформатиката най–често приложение намира при класифициране на данни, добити от ДНК разбиване на поредици (секвениране) с цел диагностициране на различни заболявания, откриване на причината за възникването им, определяне на генни функции и редица други проблеми. Популярността му се дължи основно на факта, че, за разлика от други алгоритми, SVMs успява да минимизира вероятността за грешка, което е и основна задача при класификацията.

Литература

[1]. Comparitive Analysis Of Classification Algorithm In Multiple Categories Of Bioinformatics, D.CHANDRA VARMA, D.DHARMAIAH M.TECH, International Journal of Engineering Research & Technology (IJERT) Vol. 1 Issue 7, September – 2012.

[2]. Kwan, Hon Keung and Arniker, Swarna Bai. Numerical Representation of DNA Sequences. Windsor, Ontario, Canada: University of Windsor, 2009. 978-1-4244-3355-1.

[3]. Machine Learning in Bioinformatics. Larranga, Pedro, Calvo, Borja and Santana, Roberto. 1, s.l.: Briefings in Bioinformatics, 2005, Vol. 7.

[4]. Furey, Terrance, Duffy, Nigel and Cristianini, Nello. Support Vector Machine Classification and Validation of Cancer Tissue Samples Using Microarray Expression Data. s.l. : Bioinformatics, 2000.

[5]. Guyon, I., et al. Gene Selection for Cancer Classification using Support Vector Machines. Machine Learning. 2002.

[6]. Intelligent Systems in Molecular Biology. Yang, C., et al.

[7]. Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning. Alipanahi, Babak, и др. 8, неизв. : Nature Biotechnology, August 2015 г., Том 33.

[8]. Linkage of type 2 diabeter mellitus and genetic location on chromosome 10q in Mexican Americans. Duggirala, R, et al. s.l. : American Journal of Human Genetecis, 1999

[9]. A Coding measure scheme employing electron-ion interaction pseudopotential (EIIP). Nair, AS and Sreenadhan, SP. 6, s.l.: Bioinformation, 2006, Vol. 1.

[10]. DeepBind, http://tools.genes.toronto.edu/deepbind/

За контакти:
маг. инж. Теодора Х. Христева
катедра „Компютърни системи и технологии”
Технически университет - София, филиал Пловдив
E-mail: hristeva@tu-plovdiv.bg

МЕТОДИ И СРЕДСТВА ЗА СИМУЛАЦИЯ НА SDN-NFV МРЕЖИ

Димитър Н. Тодоров

Резюме: Поради несъвместимостта на платформите, както и от необходимостта за добавяне на нови мрежови функции, се налага често закупуване на ново оборудване от различни доставчици. Виртуализацията на мрежови функции (NFV) има за цел да реши тези проблеми чрез прехвърляне на мрежови функции, които са специфични за производителя и частните хардуерни устройства към софтуер, който се хоства на Common-Off-The-Shelf (COTS) системи. Софтуерно дефинираните мрежи (SDN) са нова мрежова парадигма. Основната им функция е разделянето на мрежовото ниво за управление от данновото ниво. В настоящия доклад са разгледани инструменти за симулация на SDN и NFV мрежи. Също така са разгледани принципите на работа на мрежовите симулатори, техните архитектури и начините, по които могат да се модифицират, с цел промяна или добавяне на нови функционалности.
Ключови думи: Virtualization, SDN, NFV, Networks, Mininet, fs-SDN, CloudSimNFV

Methods and tools for simulating SDN-NFV networks

Dimitar N. Todorov

Abstract: Due to the incompability of platforms and the need to add new network functions, it is often necessary to purchase new equipment for different vendors. Network Function Virtualization (NFV) aims to solve these problems by transferring network-specific features, that are specific to the manufacturer and private hardware to software, which is hosted on Common-Off-The-Shelf (COTS) systems. Software-Defined Networks (SDN) are a new network paradigm. Their main function is to divide the network layer from data layer. This paper discusses simulation tools for SDN and NFV networks. It also contains the principles of network simulators, their architectures, and how they can be extended to modify or add new functionalities.
Keywords: Virtualization, SDN, NFV, Networks, Mininet, fs-SDN, CloudSimNFV

1. Увод

Частното хардуерно мрежово оборудване е навсякъде в бизнеса, домовете и данновите центрове. Всеки производител изследва и използва максималните възможности на платформите си, за да отговори на изискванията за производителност, надеждност и наличност, изисквани от различните типове потребители. Този подход води до несъвместимост между различните производители на технологии [1].

Поради несъвместимостта на платформите, както и от необходимостта за добавяне на нови мрежови функции (напр. защитна стена, балансиране на натоварването и др.) се налага често закупуване на ново оборудване от различни доставчици. Всеки елемент от оборудването е отговорен за дял от трафика, който изисква специфични стратегии за управление. Трудностите при управлението и конфигурирането на такива хетерогенни среди са типични. Изискването да се позволи такава гъвкавост при конфигурирането и внедряването на нови мрежови функции трябва да бъде постигнато чрез нови бизнес и инженерни модели. Сложността на текущата мрежова среда води до високи оперативни (OPEX) и капиталови (CAPEX) разходи [2].

Виртуализацията на мрежови функции (NFV) има за цел да реши тези проблеми чрез прехвърляне на мрежови функции, които са специфични за производителя и частните

хардуерни устройства към софтуер, който се хоства на Common-Off-The-Shelf (COTS) системи (т. е. системи със стандартна обработка, памет и компоненти за съхранение). Тези системи обикновено осигуряват мрежовите услуги във виртуални машини (напр. виртуални мрежови функции – NFV), като всяка осъществява различни операции (напр. защитна стена, проверка на пакети, маршрутизиране и др.) [3]. NFV има потенциал да позволи намаляване на разходите и увеличаване на скоростта на разширяване на мрежата. Също така, NFV има потенциал да увеличи гъвкавостта на мрежата за бързо предоставяне на услуги, което е много трудно да се постигне посредством традиционните методи.

Софтуерно дефинираните мрежи (SDN) са нова мрежова парадигма. Основната им функция е разделянето на мрежовото ниво за управление от данновото ниво. Сравнено с текущите мрежи, IP слоят интегрира двете нива вертикално в мрежовите устройства [4]. В нивото за управление на SDN, представено от софтуер (SDN контролер), служи за вземане на решения за това как да се борави с трафика на съответната мрежа по отношение на мрежовите политики и правила. SDN контролерът може да работи на COST системи, отделени от устройствата за препращане. Данновото ниво, разположено като мрежови устройства, отговаря за предаването на данни съгласно набор от правила. SDN контролерът позволява създаването и управлението на такива правила посредством интерфейс за приложно програмиране (API) в интерфейса Northbound.

В настоящия доклад са разгледани инструменти за симулация на SDN и NFV мрежи. Също така са разгледани принципите на работа на мрежовите симулатори, техните архитектури и начините, по които могат да се модифицират с цел промяна или добавяне на нови функционалности. Като заключение е направен обзор на направеното проучване.

2. Виртуализация на мрежови функции (NFV)

NFV предоставя мрежови услуги във виртуални машини (VM), работещи в облачни инфраструктури, където всяка виртуална машина изпълнява различни мрежови операции (напр. защитна стена, откриване на пробиви, интензивно пакетиране, балансиране на натоварването и др.) [5]. Някои от предимствата от внедряването на мрежови услуги като виртуални функции са [6]:

- Гъвкавост при разпределението на мрежовите функции в хардуера за общо предназначение;
- Бързо изпълнение и внедряване на нови мрежови услуги;
- Поддръжка на множество версии на услуги;
- Намаляване на CAPEX разходите, чрез ефективно управление на потреблението на енергия;
- Автоматизиране на оперативните процеси, като по този начин се подобрява ефективността и се намаляват OPEX разходите.

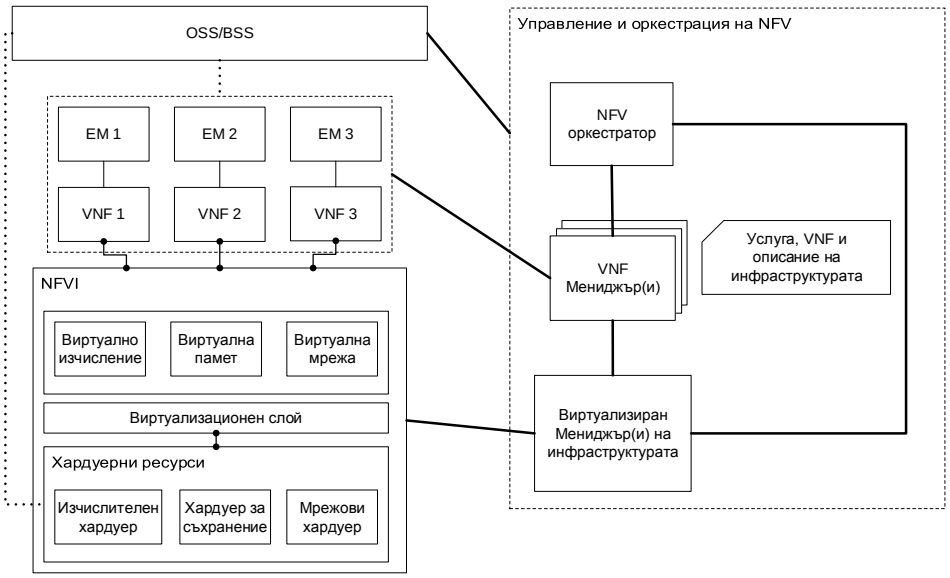
На фигура 1 е представена базова схема на NFV, която се състои от три основни функционални блока [1]:

- Виртуални мрежови функции (VNFs): VNF е виртуализацията на определена мрежова функция, която трябва да работи независимо от останалите. Тя може да работи на една или повече виртуални машини. Една VNF може да бъде разделена на няколко подфункции, наречени VNF компоненти (VNFCs). Системите за управление на елементите (EMS) могат да се използват за наблюдение на VNF;
- NFV инфраструктура (NFVI): NFVI е съвкупност от целия хардуер и софтуер, необходим за внедряване, работа и наблюдение на VNF. За тази цел NFVI има виртуализационен слой, необходим за извличане на хардуерните ресурси (обработка, съхранение и мрежова свързаност). Тя осигурява независимостта на VNF софтуера

от физическите ресурси. Виртуализиращият слой, обикновено се състои от сървър (напр. Xen, KVM, VMware и др.) и хипервайзори от мрежата (напр. VXLANs, NVGRE, OpenFlow и др.). NFVI Point of Resence (NFVI-PoP) определя мястото за внедряване на мрежовите функции, като една или много VNFs;

- Управление и оркестрация на NFV (MANO): MANO се състои от три компонента
  - Виртуализиран мениджър на инфраструктурата (VIM), който управлява и контролира взаимодействието на VNF с физическите ресурси (напр. разпределение, преразпределение и запис);
  - VNF мениджър (VNFM), който отговаря за управлението на жизнения цикъл на VNF (напр. инициализация, спиране и прекратяване);
  - NFV оркестратор (NFVO), който отговаря за осъществяването на мрежови услуги на NFVI. Освен това, той извършва операции по наблюдение на NFVI, като начин за събиране на информация за операциите и за управление на изпълнението.

Друг компонент, който се разглежда като част от NFV, е системата за поддръжка на операциите и системата за поддръжка на бизнеса (OSS/BSS). Този елемент опростява системите за управление на наследяването и подпомага MANO при изпълнението на мрежовите политики (автоматично или ръчно).



Фиг. 1. NFV архитектура

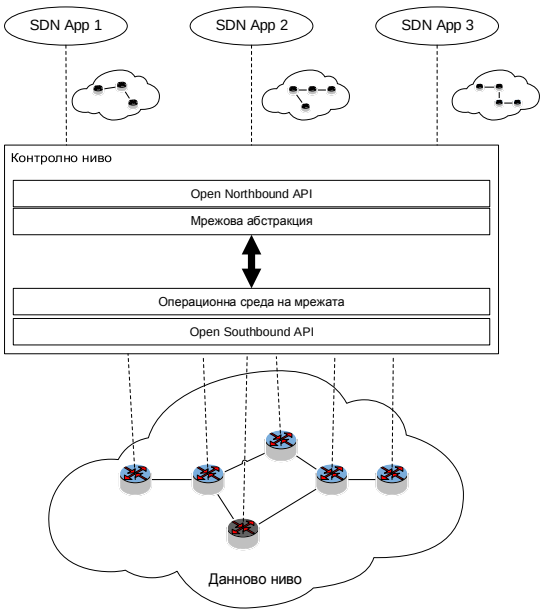
3. Софтуерно дефинирани мрежи (SDN)

SDN е нова мрежова парадигма, която е проектирана да избегне трудността при разработването и тестването на нови решения и протоколи в производствени среди, където основните кодиращи устройства в бизнес суичовете и маршрутизаторите са частни и затворени [7].

Основната характеристика на SDN парадигмата е отделянето на контролното и данновото ниво. Това има ясни предимства, при които програмируемостта на мрежата се постига, чрез централизиране на контролното ниво, съвпадащо с наличие на отворени API-та. По този начин се улеснява процесът на създаване и внедряване на нови мрежови приложения. SDN осигурява опростяване и гъвкавост при прилагането на мрежовата политика, като улеснява конфигурацията и управлението на мрежата [8].

Контролното ниво, което е представено от софтуер (наречен SDN контролер), отговаря за вземането на решения, за това как да се справи с мрежовия трафик. SDN контролерът може да работи на COST платформи, които са отделени от мрежовото оборудване. Платформата за данни, представлявана от мрежовите устройства, отговаря за предаването на трафика, в съответствие с набор от правила. Тези правила се създават и управляват от SDN контролера. SDN контролерът има глобален поглед към топологията на мрежата и има директен контрол върху елементите на данновото ниво, чрез свързващия протокол OpenFlow [9].

Фигура 2 илюстрира трите SDN архитектурни слоя и API-тата, които са отговорни за взаимодействието между тях. Приложният програмен интерфейс (SDN Northbound API) е отговорен за осигуряването на поддръжка за комуникация между приложния слой и слоя на контролното ниво. Той, също така, включва и поддръжка за SDN приложенията, като управление на трафика, рутиране, защитна стена, качество на услугата и други. Southbound API отговаря за комуникацията между контролерите на SDN и суичовете [1].



Фиг. 2. SDN архитектура

4. Mininet

Mininet е система, която позволява бързо създаване на големи мрежови прототипи на един компютър. С негова помощ се създават мащабируеми софтуерно дефинирани мрежи, използващи леки механизми за визуализация. Тези функции, позволяват на Mininet да създава, взаимодейства, персонализира и споделя прототипите бързо [10].

Някои от характеристиките, които водят до създаването на Mininet, са:

- Гъвкавост – нови топологии и нови функции могат да бъдат зададени в софтуера, като се използват програмни езици и общи операционни системи;
- Приложимост – коректно създадени прототипи, трябва да могат да се използват в реални мрежи, без никакви промени в кода;
- Интерактивност – управлението и изпълнението на симулирани мрежи, трябва да се осъществява в реално време, както се случва в реалните мрежи;

- Мащабируемост – прототипната среда, трябва да се мащабира до големи мрежи със стотици и хиляди суичове само на един компютър;
- Реалистичност – прототипното поведение, трябва да съответства на поведението в реално време с висока степен на точност, така че приложенията и протоколите, трябва да могат да се използват без никаква промяна на кода;
- Споделяемост – създадените прототипи трябва лесно да се споделят, така че те да могат да се стартират и да променят експериментите.

Mininet може да създава SDN елементи, да ги персонализира, да ги споделя с други мрежи и да осъществява взаимодействия. Тези елементи включват хостове, превключватели, контролери и връзки. Хостът в Mininet е прост процес със собствена мрежова среда, работеща върху операционната система. Всеки един хост осигурява процеси с индивидуален мрежови интерфейс, портове, адреси и маршрутизиращи таблици (като ARP за IP).

OpenFlow суичовете, създадени от Mininet, осигуряват една и съща семантика за предаване на пакети, която се поддържа от хардуерния суич. Суичовете на ниво потребител и ядро са налични. При симулация, контролерите могат да работят на реална или симулирана мрежа, докато машината, на която се изпълняват суичовете, имат връзка с контролера. При необходимост, Mininet създава стандартен контролер в локалната симулационна среда, както и виртуални връзки между елементите, чрез техните виртуални интерфейси.

5. fs-SDN

fs е Python базиран инструмент за генериране на записи в мрежовия поток и интерфейсни броячи на SNMP [11]. Въпреки че не е проектирана за целите на симулиране на мрежова дейност, са използвани техники за симулиране на дискретни събития за синтезиране на мрежови измервания. Възможностите за генериране на TCP трафик, базирани на Harpoon [12] (който на свой ред се базира на SURGE [13]), са вградени. fs допълнително използва съществуващите TCP пропускателни модели за симулиране на инвидуални TCP потоци. Той също така включва способността да генерира широк спектър от симулирани условия на трафика, които могат лесно да бъдат конфигурирани, чрез декларативна спецификация, основана на езика Graphviz DOT или в JSON формат.

Най-критичната ключова мрежова абстракция, върху която работи, не е данновият пакет, а по-високо ниво, което е наречено flowlet. flowlet се отнася за обема на потока, излъчван за определен период от време. Чрез повишаване на нивото на абстракция и по този начин на обекта, около който се обръщат повечето събития на симулатора, се постига много по-висока скорост и ефективност, от съществуващите симулатори на ниво пакет, подобно на ns-2 [14] и ns-3 [15].

Системен дизайн на fs

Основните цели за дизайн на fs-sdn са:

- Генериране на точни измервания, които са подобни на тези, които биха били събрани в реална или емулирана мрежа;
- Мащабиране до големи мрежи;
- Осигуряване на API, което да позволява лесно пренасяне на приложенията за контролера към други платформи, като POX [16], NOX [17], и Floodlight [18].

Генерирането на точни измервания е от важно значение. fs-sdn може да генерира подобен на SNMP байтов/пакетен/потоков брояч на конфигурируеми интервали, както и записи за потока. Оригиналната версия на fs [19] дава точни измервания, сравнени с тези, които са генерирани в реална мрежова настройка и подобна симулирана настройка.

Едно централно решение, което засяга колко добре може да се мащабира fs-sdn е, че не се опитва да моделира взаимодействията на ниво пакет, а по-скоро работи на по-високо абстракционно ниво, наречено flowlet. В текущата версия на fs-sdn могат да се мащабират добре стотици суичове и хостове в зависимост от конфигурацията на трафика [20,21].

Компонентите на контролера, разработени за POX, могат да се използват директно в fs-sdn. По-специално, кодът на контролера, разработен за fs-sdn, може да се прехвърли без промяна за използване в POX. Приложните програмни интерфейси (API), които са налични в POX, са подобни на тези в NOX и други платформи. В резултат на използването на съществуващите POI API-та, fs-sdn позволява на разработчиците да правят прототипи, тестове и отстраняване на грешки в симулирано приложение, след което директно да изпълнят същия код върху други среди [22].

6. CloudSimNFV архитектура

CloudSimNFV [23] е базиран на платформата CloudSim, като се възползва от имплементацията на центровете за данни на CloudSim. С нейна помощ могат да се разширят специфични NFV сценарии. Динамичните натоварвания могат да бъдат симулирани чрез зареждане на файлове с натоварвания. Чрез промяна на програмните кодове могат да се осигурят физическа и виртуална конфигурация на топологията.

Зареждането на сценария за симулация се извършва посредством текстови файлове. Всеки тип на работното натоварване включва четири файла с работно натоварване, които се използват за симулиране на работното натоварване на процесора, паметта, съхранението и натоварването на честотната лента поотделно. Тези файлове за работно натоварване могат да образуват различни режими на натоварване като работно натоварване в компютърен режим (изисква се най-много CPU ресурс) и работно натоварване в режим мрежа (най-много е необходим капацитет).

VM услугите отговарят за управлението на VM, като изчисляват изпълнението на приложенията и времето за предаване на пакети между VM. Също така осигуряват създаването на cloudlet и актуализирането на съответната виртуална машина.

Слоят за предоставяне на ресурси се състои от четири модула:

- Модул за предоставяне на CPU - осигурява CPU ресурс за VM;
- Модул за предоставяне на памет - осигурява памет за VM;
- Модул за предоставяне на пространство - осигурява дисково пространство за VM;
- Модул за предоставяне на мрежа - осигурява мрежови ресурс за VM.

Предоставянето на ресурси зависи от работното натоварване, което крайният потребител е избрал да използва.

Слоят “Разпределение на ресурсите" съдържа модули, които разпределят ресурсите в долния слой на архитектурата, в съответствие с политиката за миграция на VM и политиката за избор на VM, определени от потребителите на симулатора.

Последният слой на CloudSimNFV е облачният слой. Този слой съдържа центровете за данни, хост, VM и cloudlet, за да се симулират различните компоненти на реалните центрове за данни. В този слой са направени някои разширения на cloudlet като NFVlet, за да могат да бъдат симулирани NFV задачи. Конфигурацията на физическата топология се изпълнява, за да се представи връзката между компонентите.

7. Заключение

В настоящия доклад са разгледани методи и средства за симулация на SDN и NFV мрежи, както и техники за изследване на персонализирани функционалности. Разгледани са

някои от симулаторите (Mininet, fs-SND и CloudSimNFV), чрез които могат да бъдат симулирани и изследвани SDN и NFV мрежите.

След направените проучвания, бяха установени основните изисквания към съществуващите симулатори, както и необходимостта за миграция от тестова към реална среда. Също така бяха разгледани градивните компоненти на проучените симулатори, както и методите за модификация на съществуващите компоненти, чрез които могат да бъдат направени изследвания върху алгоритмите за маршрутизация и алгоритмите за балансирано натоварване на мрежата.

Като насоки за бъдеща работа се предвиждат изграждане на персонализирани функционалности за SDN и NFV, които ще бъдат симулирани във виртуална среда с помощта на проучените симулатори. Новите персонализирани функционалности ще бъдат сравнени с вече съществуващите решения.

Литература

[1] Michel S. Bonfim, Kelvin L. Dias, and Stenio F. L. Fernandes. 0000. Integrated NFV/SDN Architectures: A Systematic Literature Review. ACM Comput.

[2] ETSI. 2012. Network Functions Virtualisation - An Introduction, Bene_ts, Enablers, Challenges and Call for Action. White Paper (Out. 2012).

[3] ETSI. 2015. Network Functions Virtualisation (NFV) - Network Operator Perspectives on Industry Progress. White Paper (Jan. 2015).

[4] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner. 2008. OpenFlow: Enabling Innovation in Campus Networks. SIGCOMM Comput. Commun. Rev. 38, 2 (March 2008), 69–74.

[5] ETSI. 2015. Network Functions Virtualisation (NFV) - Network Operator Perspectives on Industry Progress., White Paper (Jan. 2015).

[6] ETSI. 2012. Network Functions Virtualisation - An Introduction, Benefits, Enablers, Challenges and Call for Action., White Paper (Out. 2012).

[7] Nick McKeown, Tom Anderson, Hari Balakrishnan, Guru Parulkar, Larry Peterson, Jennifer Rexford, Scott Shenker, and Jonathan Turner. 2008. OpenFlow: Enabling Innovation in Campus Networks. SIGCOMM Comput. Commun. Rev. 38, 2 (March 2008), 69–74.

[8] D. Kreutz, F. M. V. Ramos, P. E. Veríssimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig. 2015. Software-Defined Networking: A Comprehensive Survey. Proc. IEEE 103, 1 (Jan 2015), 14–76. DOI.

[9] ONF. 2015. OpenFlow Switch Specification - Version 1.3.5. (Mar. 2015).

[10] Mininet. (2013, Mar). An Instant Virtual Network on your Laptop (or other PC). [Online]. Available: http://mininet.org/

[11] J. Sommers, R. Bowden, B. Eriksson, P. Barford, M. Roughan, and N. Duffield. Efficient network-wide flow record generation. In Proceedings of INFOCOM ’11, pages 2363–2371, April 2011.

[12] J. Sommers and P. Barford. Self-Configuring Network Traffic Generation. In Proceedings of ACM Internet Measurement Conference, Taormina, Italy, October 2004.

[13] P. Barford and M. Crovella. Generating Representative Web Workloads for Network and Server Performance Evaluation. In Proceedings of ACM SIGMETRICS, Madison, WI, June 1998.

[14] N. Foster, A. Guha, M. Reitblatt, A. Story, M. Freedman, N. Katta, C. Monsanto, J. Reich, J. Rexford, C. Schlesinger, Story A., and D. Walker. Languages for software-defined networks. Communications Magazine, IEEE, 51(2):128–134, 2013.

[15] A. Greenberg, J.R. Hamilton, N. Jain, S. Kandula, C. Kim, P. Lahiri, D.A. Maltz, P. Patel, and S. Sengupta. VL2: a scalable and flexible data center network. In Proceedings of ACM SIGCOMM ’09, August 2009.

[16] POX, Python-based OpenFlow Controller. http://www.noxrepo.org/pox/about-pox/
[17] N. Gude, T. Koponen, J. Pettit, B. Pfaff, M. Casado, N. McKeown, and S. Shenker. NOX: towards an operating system for networks. ACM SIGCOMM Computer Communication Review, 38(3):105–110, 2008.
[18] Floodlight, Java-based OpenFlow Controller. http://floodlight.openflowhub.org/
[19] J. Sommers, R. Bowden, B. Eriksson, P. Barford, M. Roughan, and N. Duffield. Efficient network-wide flow record generation. In Proceedings of INFOCOM ’11, pages 2363–2371, April 2011.
[20] A. Curtis, J. Mogul, J. Tourrilhes, P. Yalagandula, P. Sharma, and S. Banerjee. DevoFlow: scaling flow management for high-performance networks. In Proceedings of ACM SIGCOMM ’11, 2011.
[21] B. Stephens, A. Cox, W. Felter, C. Dixon, and J. Carter. PAST: scalable ethernet for data centers. In Proceedings of ACM CoNeXT ’12, pages49–60, 2012.
[22] M. Gupta, J. Sommers, P. Barford, Fast, Accurate Simulation for SDN Prototyping. ACM 978-1-4503-2178-5/13/08, 2013.
[23] W. Yang, M. Xu, G. Li, W. Tian, CloudSimNFV: Modeling and Simulation of Energy-Efficient NFV in Cloud Data Centers, 2015

За контакти:

инж. Димитър Н. Тодоров
катедра „Компютърни науки и технологии”
Технически университет - Варна
E-mail: dntodorov@hotmail.com

CONSISTENCY OF APPLICATION OF THE METHODS OF RISK ANALYSIS IN ACCORDANCE WITH THE LEVEL OF KNOWLEDGE OF THE PROCESS

Kiril Kirov, Dobrina Ralcheva

Abstract: The author's research about the possibilities and the restrictions on the management of environmental objects based on standard methods of risk analysis is presented in this article. Emphasis is on the impact of the level of knowledge of the processes on the opportunities for adequate risk management and main approaches for their solution. Classification of methods is proposed in accordance with the level of knowledge of the analysis process and its interaction with the environment.
Keywords: Ecology, Knowledge, Risk analysis, Sustainable development.

1. Introduction

Today's economy and industrial development poses new challenges in front of us with unknown results and impact on the ecological system. In contemporary practice sustainable methods put on risk management, as a basis for avoiding potential problems and the possibility of accumulation of objective information, allowing us to predict the possibility of undesirable events. Within the framework of this decade the standards for risk management entered practically in all spheres of public life [4].

Today's economy requires extreme fast readjustment of industrial and social structures to a new technology. The needs and the criteria determining the adequacy of personnel and used technologies are also changing dynamically in time. Changing technology requires an adequate assessment of the existing system for environmental risks and achieving preventive management [5].

The application and effectiveness of the methods of analysis, monitoring and risk management to a large extent depends on the level of knowledge of the problem, as well as the accumulation of statistical information and the possibilities for its use of adequate forecasting [4].

2. Levels of knowledge of the risks to the process and environment

If we compare the modern theory and practice of risk analysis, a significant discrepancy between the approaches used in classification and management, could be found. In the theoretical approach has seen deepening concerns over separation and specificity in the models and approaches of management depending on the branch of industry, while modern standard models is achieved appropriate unification for the identification, analysis, discretion and risk impact[5].

Relatively rarely discuss basic aspect of the application of the methods of analysis of the risks and the adequacy of their use, the level of knowledge of the analysis process. The level of knowledge defines the ability to use certain methods, as well as to collect (organize) information. Knowledge of the process it is possible to be grouped according to the sequence of the stages and building knowledge about the process. They may be presented in the following order [5]:

- Expert stage – risk analysis is created and developed on the basis of a hypothesis about the nature of the process and/or by analogy with another process, which is assumed to be similar (model) of the analysis. In this case, risk management is realized on the basis of the experience of the participating experts; their intuition and depends on the reliability of the raised hypothesis;

- Statistical phase – risk analysis builds on the basis of accumulated information and hypothesis the essence of the analysis process. In this case, risk management is realized on the basis of hypotheses about the models defining the essence of the ongoing processes and depends on the reliability of the hypotheses;
- Theoretical stage – risk analysis builds on the basis of scientific models or laws governing the behavior of managed processes. In this case, risk management is realized on the basis of adequate theoretical knowledge about the process, his uncertainty depends on the ability of responsible person for analyze to appreciate the complex interaction of environmental and process their knowledge of this process.

If we try to imagine the sequence corresponding to the adequacy of the knowledge of the risks to the process and environment, it follows the above stages, initially to apply them to the process as an isolated phenomenon and in effect for the process and its interaction with the environment, such as a complex phenomenon.

3. **Classification of methods, in accordance with the level of knowledge of the processes**

The classification of methods is implemented in accordance with БДС EN ISO 31000 and accepted for standard methods of risk management. Risk management process includes the following main stages:establishing of context; risk identification; risk analysis; risk evaluation and risk treatment.

Establishment of circumstances sets basic parameters for risk management and control criteria.This stage of risk management is performed determine the internal and external factors that have an impact on the Organization, the circumstances of the individual management risks to be assessed, as well as the classification criteria for the risk[3].

Determination of external circumstances include: understanding the environment in which the organisation operates at national, international, regional and local level, determines the cultural, political, economic, financial factors associated with the competitive environment[2].

The establishment of the internal circumstances includes determining the capabilities of the organization associated with the resources, information for decision-making, internal stakeholders, the objectives and strategies, policies, standards and the responsibilities of the Organization. The risk assessment includes a wide range of reasons and consequences, including identification, analysis and estimation of risk. Identification is the process of detecting, recognizing and recording of risk. The process of identification involves determining the causes and sources of risk that may have a material impact on the purpose and nature of the impacts. The methods used at the stage of the stages of identification of risk may be defined as applicable and recommended or inapplicable[3].

In **Table 1** are shown the tools and methods applied and recommended at risk identification[1],[2],[3].

Table 1. Risk Identification

Type of Risk Assessment Technique	Applicable	Level of Knowledge
Brainstorming	NA	E
Structured or semi-structured interviews	NA	E
Delphi	NA	E
Check-lists	NA	S
Primary hazard analysis	NA	E
Hazard and operability studies (HAZOP)	SA	S
Hazard Analysis and Critical Control Points (HACCP)	NA	E

Environmental risk assessment	SA	SC
Structure « What if? » (SWIFT)	SA	E
Scenario analysis	SA	E
Business impact analysis	A	E
Root cause analysis	SA	E
Failure mode effect analysis	SA	S
Fault tree analysis	A	E
Event tree analysis	A	E; S
Cause and consequence analysis	A	S
Cause-and-effect analysis	NA	E
Layer protection analysis (LOPA)	A	E; S
Decision tree	A	E; S
Human reliability analysis	SA	S
Bow tie analysis	A	E
Reliability centered maintenance	SA	S
Sneak circuit analysis	NA	SC
Markov analysis	NA	S
Monte Carlo simulation	NA	S
Bayesian statistics and Bayes Nets	NA	S
FN curves	SA	SC
Risk indices	SA	SC
Consequence/probability matrix	SA	S
Cost/benefit analysis	A	S
Multi-criteria decision analysis (MCDA)	A	SC
	SA –Strongly applicable	E – Expert Level
	A - Applicable	S - Statistical Level
	NA – Not Applicable	SC - Scientific Level

The risk analysis provides input elements for risk assessment. The risk analysis involves examining the causes and sources of risks, their consequences and the probability that these consequences may occur. The factors that influence the probability and consequences have to be defined. Assessment of the totality of the potential consequences are related to the likelihood of an event or circumstance, to measure the level of risk [2].

The methods used in the risk analysis may be qualitative or quantitative, semiquantitative. The qualitative evaluation of the likelihood and consequences, determined the level of risk through terms such as high, medium and low and can effect and combines a chance to assess the level of risk in relation to the quality criteria. Semiquantative methods using digital scales for the assessment of consequences and likelihood and combine them in order to determine the level of risk using appropriate formulas[3].

Quantitative analysis gives estimates of the actual values for the consequences their probabilities and gives the values for the level of risk in specific units, certain other circumstances are clarified. In **Table 2** the methods used for risk assessment are grouped [1],[2],[3].

Table 2. Risk Analysis

Type of Risk Assessment Technique	Applicable	Level of Knowledge
Brainstorming	SA	E
Structured or semi-structured interviews	SA	E
Delphi	SA	E
Check-lists	SA	S
Primary hazard analysis	A	E
Hazard and operability studies (HAZOP)	SA	S
Hazard Analysis and Critical Control Points (HACCP)	SA	E
Environmental risk assessment	SA	SC
Structure « What if? » (SWIFT)	SA	E
Scenario analysis	SA	E
Business impact analysis	A	E
Root cause analysis	NA	E
Failure mode effect analysis	SA	S
Fault tree analysis	A	E
Event tree analysis	A	E; S
Cause and consequence analysis	A	S
Cause-and-effect analysis	SA	E
Layer protection analysis (LOPA)	A	E; S
Decision tree	NA	E; S
Human reliability analysis	SA	S
Bow tie analysis	NA	E
Reliability centered maintenance	SA	S
Sneak circuit analysis	A	SC
Markov analysis	A	S
Monte Carlo simulation	NA	S
Bayesian statistics and Bayes Nets	NA	S
FN curves	A	SC
Risk indices	A	SC
Consequence/probability matrix	SA	S
Cost/benefit analysis	A	S
Multi-criteria decision analysis (MCDA)	A	SC
	SA –Strongly applicable	E – Expert Level
	A - Applicable	S - Statistical Level
	NA – Not Applicable	SC - Scientific Level

The assessment of risk as an element of the risk assessment process involves a comparison of the assessed levels of risk criteria laid down when the circumstances are established based on the understanding of the risk, which is obtained during a risk analysis, to make decisions about future

actions. The solutions are related to risk needs impact; priorities for impact; whether a particular activity to be undertaken, and the paths to be followed [3].

In **Table 3** are presented the methods used for risk assessment at the stage of assessing the risk [1], [2], [3].

Table 3. Risk Analysis

Type of Risk Assessment Technique	Applicable	Level of Knowledge
Brainstorming	NA	E
Structured or semi-structured interviews	NA	E
Delphi	NA	E
Check-lists	NA	S
Primary hazard analysis	NA	E
Hazard and operability studies (HAZOP)	A	S
Hazard Analysis and Critical Control Points (HACCP)	SA	E
Environmental risk assessment	SA	SC
Structure « What if? » (SWIFT)	SA	E
Scenario analysis	A	E
Business impact analysis	A	E
Root cause analysis	SA	E
Failure mode effect analysis	SA	S
Fault tree analysis	A	E
Event tree analysis	NA	E; S
Cause and consequence analysis	A	S
Cause-and-effect analysis	NA	E
Layer protection analysis (LOPA)	NA	E; S
Decision tree	A	E; S
Human reliability analysis	A	S
Bow tie analysis	A	E
Reliability centered maintenance	SA	S
Sneak circuit analysis	NA	SC
Markov analysis	NA	S
Monte Carlo simulation	SA	S
Bayesian statistics and Bayes Nets	SA	S
FN curves	SA	SC
Risk indices	SA	SC
Consequence/probability matrix	SA	S
Cost/benefit analysis	A	S
Multi-criteria decision analysis (MCDA)	A	SC
	SA –Strongly applicable	E – Expert Level
	A - Applicable	S - Statistical Level
	NA – Not Applicable	SC - Scientific Level

4. Conclusions

1. Unification of methods of risk management through the practice of modern standardization does not offer a selection and linking the level of knowledge of the process with the applicability of the methods.

2. The introduction of the classification of the level of knowledge would help to determine the appropriate methods of analysis for the application in accordance with the level of knowledge of managed processes and objects.

3. Development of methods for the analysis of risk needs to be focused on their ability to accumulate and use experience, information and knowledge acquired in the process of studying the managed processes and objects.

4. Methods of risk management are necessary to be an adequate in their application for each phase of the life cycle and the level of knowledge of the process and/or object. Such an approach would allow us accumulation of information and knowledge, as well as tracking of the process of learning the object of management.

References

[1]. ISO Guide 73, Risk management- Vocabulary, (2009).
[2]. ISO/IEC 31010, Risk management- principles and guidelines, (2011)
[3]. ISO/IEC 31010, Risk management- Risk assessment techniques, (2011)
[4]. Pasman, H., Risk Analysis and Control for Industrial Processes - Gas, Oil and Chemicals A System Perspective for Assessing and Avoiding Low-Probability, ELSELVER, (2015)
[5]. Cox, L., Risk Analysis of Complex and Uncertain Systems, Springer, (2009)

For contacts:
Assoc. Prof. KirilKirov, Phd
Department of Manufacturing Technologies and Machine Tools
Technical University of Varna, Bulgaria
E-mail: kirov@tu-varna.bg

Eng. DobrinaRalcheva
Departament of Ecology and Environmental protection
Technical University of Varna, Bulgaria
E-mail: d.ralcheva@tu-varna.bg

Bulgaria Communications Chapter

Оценка на риска в системи за управление на сигурността на информацията – същност и насоки

Петко Г. Генчев, Милена Н. Карова

Резюме: В съвременния свят все повече нараства значението на сигурността на информацията. Това е причина много организации да изграждат системи за управление на сигурността на информацията. В основата на тези системи е заложено управление на риска. В тази статия са разгледани основните изисквания на стандартите на ISO в тази сфера. Основно внимание е отделено на изискванията към дейностите по оценка на риска за сигурността на информацията. В материала са описани някои проблеми, свързани с тези дейности и описани в цитирани статии по теми за сигурността на информацията. Направени са изводи, свързани с преодоляване на тези проблеми и повишаване на ефективността на оценката на риска за информационната сигурност.
Ключови думи: Управление на риска за сигурността на информацията, оценка на риска за сигурността на информацията, проблеми при оценка на риска за информационната сигурност.

Essence and problems in assessing the risk of information in information security management systems

Petko Genchev, Milena Karova

Abstract: In today's world, the importance of information security is growing. This is why many organizations are building information security management systems. The basis of these systems is risk management. This article discusses the core requirements of ISO standards in this field. Major focus is placed on the requirements for risk assessment activities for information security. The article describes some issues related to these activities and described in quoted articles on information security topics. Conclusions have been made to address these issues and to increase the effectiveness of the risk assessment for information security.
Keywords: information security risk management, information security risk assessment, problems in risk assessment for information security.

1. Увод

В съвременния свят все повече организации разчитат на информационни технологии, за да им помогнат да постигнат своите бизнес цели като по-бърз отговор на услугата или по-добро качество. Ето защо сигурността на информацията е от първостепенно значение за организациите. Необходим е систематичен подход за управление на риска по отношение на сигурността на информацията, който да помогне да се определят изискванията за сигурност на информацията и да се създаде ефективна система за управление. Рискът за информационна сигурност често се изразява в комбинация от последствията от събитие за информационна сигурност и вероятността да възникне това събитие. Предметът на оценката на риска се нарича информационен актив. Трябва да се отбележи, че информационният актив се състои от повече от хардуер и софтуер и обхваща хора, информация, сервиз и др. Процесът на оценката на риска, както и свързаните с него техники, предлагат аналитичен и структуриран подход към състоянието на сигурността на организацията [6].

Целта на тази статия е описание на същността на система за управление на сигурността на информацията (СУСИ), на дейностите по управление на риска за сигурността на информацията (УРСИ) и дейностите по оценка на риска, като основен елемент на УРСИ. В светлината на изисквания на стандартите на ISO в тази сфера са описани някои от основните

трудности при релизация на дейностите по оценка на риска за сигурността на информацията и са направени изводи за преодоляване им .

Статията е структурирана в 7 точки, като настоящият увод е първа точка, във втора са разгледани изискванията за управление на риска в светлината на стандарта за изграждане на **СУСИ** – ISO/IEC 27001 [1], в трета точка е описана структурата на процеса по **управление на риска** за информационната сигурност според препоръките, дадени в стандарта ISO/IEC 27005 [2]; в четвърта точка е обърнато особено внимание на дейностите по **оценка на риска** за сигурността на информацията, като основен елемент на УРСИ; в пета точка е направен опит за дефиниране на някои **трудности** при релизация на дейностите по оценка на риска за сигурността на информацията; в шеста точка, като заключение са направени изводи за бъдещи изследвания в сферата на оценка на риска за сигурността на информацията. В последната точка са дадени някои определения на термини, които са използвани в текста.

2. СУСИ – същност, стандарти, изисквания за управлението на риска

За организациите, за които информацията е особенно важен актив, е необходимо да изградят и внедрят Система за Управление на Информационната Сигурност (СУСИ), която представлява онази част от общата система за управление, основана на подхода за риска, свързан с организацията, за изграждане, внедряване, функциониране, наблюдение, преглед, поддържане и подобряване на сигурността на информацията. Системата за управление включва организационната структура, политиката, дейностите по планирането, отговорностите, практиките, процедурите, процесите и ресурсите.

Насоки за изграждане на СУСИ са дефинирани в стандарт ISO/IEC 27001 [1]. Това са общи препоръки за управление, които са приложими за всякакъв характер на дейност и структура на организацията. Като пример може да се даде препоръката за организация на управлението във вид на цикличност между 4 етапа: Планиране (Plan), Изпълнение (Do), Проверка (Check), Действие (Act). Този цикъл (цикъл на Дейминг) редува планиране, реализация на планираното, проверка между планирано и реализирано и действия за отстраняване на несъответствията, установени при проверката.

Стандартът ISO/IEC 27001 дефинира ясни организационни правила и описва най-важните действия, които трябва да предприеме организацията в областта на информационната сигурност. В основата на изграждането на СУСИ стандартът поставя управлението на риска за сигурността на информацията (СИ), като във връзка с това се дефинират изисквания за:

–Определяне на вътрешните и външни условия **(контекст)**, които влияят на постигането на целите на организацията;

–Планиране на действия по **отчитане на рискове** и оценка на ефективността на тези действия;

–Дефиниране и прилагане на процес на **оценяване на риска** за сигурността на информацията, в който: да се определят критерии за приемане и оценяване на риска; да се гарантира сравнимост на резултатите от оценяването на риска; да се идентифицират рисковете и собствениците на риска (имащи отговорност и пълномощия да упавляват риска); да се анализират рисковете за СИ и да се оценят последствията и вероятността да се случат и да се определят нивата на риска; да се сравнят резултатите от анализа с критериите за риска и да се определят приоритетите на рисковете с оглед на въздействие;

–**Въздействие върху риска** за СИ (за намаляване на нивото на риска);

–Извършване на оценка на риска за СИ през **планирани интервали от време** или при възникване на съществени промени;

–Преглед **от висшето ръководство** на резултатите от преценката на риска и състоянието на плана за въздействие върху риска.

3. Управление на риска за сигурността на информацията – същност, стандарти

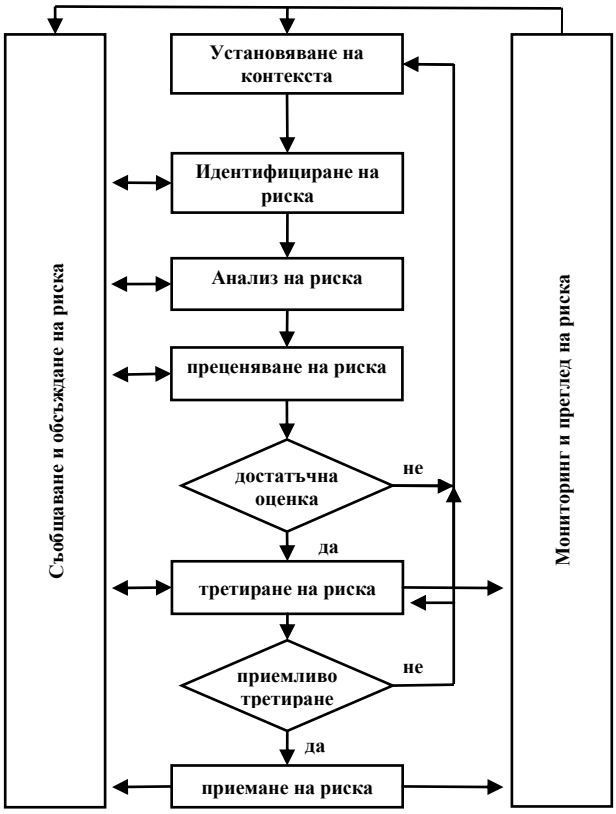
Стандартът ISO/IEC 27001 дава препоръки за създаване на система за управление на сигурността на информацията и се използва за сертифициране на организацията по отношение на информационната сигурност.

Стандартът ISO/IEC 27005 [2] е помощен стандарт, в групата стандарти ISO/IEC 27XXX, който е свързан с управление на риска за сигурността на информацията. Този стандарт не се използва за сертифициране на организации, а е предназначен да подпомогне удовлетворителното внедряване на сигурността на информацията, основано на подхода за управление на риска. Този стандарт разглежда общите препоръки за управление на риска, които са описани в стандарти ISO 31000 [3] и ISO 31010 [4], в светлината на сигурността на информацията.

Управлението на риска за сигурността на информацията трябва да бъде непрекъснат процес, който установява външния и вътрешния контекст, оценява рисковете и третира рисковете, използвайки план за третиране на риска.

На фигура 1 е показан процесът на управление на риска за сигурността на информацията според стандарта ISO/IEC 27005 [2]. В тази блокова схема:

- **Установяване на контекста** се извършва от управителните органи на организацията, съвместно със специалисти по сигурност на информацията и е свързано със стратегията, политиката и структурата на организацията, както и с целите и правилата за взаимодействие с външни организации. Крайната цел на тези дейности е определяне на политика по информационната сигурност, определяне на методология за оценяване на риска за информационната сигурност и дефиниране на нивата за оценяване на риска и определящите го параметри (уязвимост, заплаха, вероятност, въздействие ....), т.е. определя се рамката на оценка и управление на риска за сигурността на информацията;
- **Идентифициране на риска** е процес на намиране, разпознаване и описване на рискове;
- **Анализ на риска** е процес за разбиране естеството на риска и за определяне на нивото на риска;
- **Преценяване на риска** е процес на сравняване на резултатите от анализа на риска с критериите за риск (които са определени в етапа на установяване на контекста), за да се определи дали рискът и/или неговата значимост са приемливи или допустими (поносими). Преценяването на риска подпомага вземането на решение, сварзано с въздействието върху риска;
- **Третиране (въздействие) на риска** е процес за изменение на риска чрез: ***модификация на риска*** (чрез въвеждане, премахване или промяна на механизми за контрол), ***поемане на риска*** (когато нивото на риска е приемливо), ***избягване на риска*** (отказ от използването на рисковия актив) и ***споделяне на риска*** (застраховка или подизпълнител);
- **Приемане на риска** е информирано решение да се поеме определен риск;
- **Съобщаване и обсъждане на риска** е непрекъснат и повтарящ се процес, внедрен от дадена организация, който има за цел да се предостави, разпредели или получи информация и да се осъществи диалог със заинтересованите страни, свързани с управлението на риска;
- **Мониторинг и преглед на риска** включва наблюдение и преглед на рисковете и техните фактори (стойност на активите, въздействие, заплахи, уязвимости, вероятност от поява), за да се идентифицират всякакви промени в контекста на организацията на ранен етап и да има актуален поглед върху цялостната картина на риска. Изисква се също процесът по управление на риска за сигурността на информацията да бъде непрекъснато наблюдаван, подлаган на преглед и подобряван, когато е необходимо и подходящо.



Фиг. 1. Илюстриране на процеса на управление на риска за сигурността на информацията в стандарта ISO/IEC 27005 [2]

4. Оценяване на риска

Рискът е комбинация от последствията, които биха могли да бъдат резултат от случването на нежелано събитие и вероятността от възникване на самото събитие.

Оценяване на риска е цялостният процес за идентифициране на риска, анализ на риска и преценяване на риска. Оценяването на риска често се извършва на две (или повече) итерации. Първо се прави оценяване на високо ниво за идентифициране на потенциалните рискове от високо ниво, които дават основание за по-нататъшно оценяване. Следващата итерация може да включва по-нататъшно задълбочено разглеждане на потенциалните рискове от високо ниво, открити при началната итерация. Там, където това не предоставя достатъчно информация за оценяване на риска, се извършват допълнителни подробни анализи - обикновено на части от общия обхват и по възможност чрез използване на друг метод.

В приложение към стандарта ISO/IEC 27005 [2] са обсъдени подробно подходите за оценяване на риска за сигурността на информацията.

4.1 Идентифициране на риска:

4.1.1 Идентифициране на активите

За всеки актив трябва да бъде идентифициран собственик, за да се осигури отговорност и отчетност за актива. Собственикът на актива е отговорен за неговото производство, разработка, експлоатация, използване и защита и често е най-подходящото лице за определене на стойността на актива за организацията.

4.1.2 Идентифициране на заплахите

Заплахата има потенциал да навреди на активи като информация, процеси и системи, както и на организациите. Заплахите могат да имат природен или човешки произход и могат да бъдат инцидентни или преднамерени. Трябва да бъдат идентифицирани източниците както на инцидентните, така и на преднамерените заплахи. В приложение към стандарта ISO/IEC 27005 [2] е дадена изчерпателна информация за видовете заплахи.

4.1.3 Идентифициране на съществуващите механизми за контрол

Трябва да се идентифицират съществуващите механизми за контрол, за да се избегне ненужна работа или разходи, например при дублиране на механизми за контрол, и да се установи, че тези механизми за контрол дават резултат.

4.1.4 Идентифициране на уязвимостите

Необходимо е да се идентифицират уязвимостите, които могат да се използват от заплахите, за да предизвикат вреди на активите или на организацията.

Уязвимости могат да се идентифицират в следните области:

- Организация;
- Процеси и процедури;
- Управленски практики;
- Персонал;
- Физическа сред;а
- Конфигурация на информационната система;
- Хардуер, софтуер или комуникационни устроиства;
- Зависимост от външни страни.

Примери за уязвимости и методи за оценяване на уязвимости са дадени в приложение към стандарта ISO/IEC 27005 [2].

4.1.5 Идентифициране на последствията

Необходимо е да бъдат идентифицирани последствията за активите, до които може да доведе загубата на конфиденциалност, интегритет и наличност. Последствие може да бъде загуба на ефикасност, неблагоприятни условия на работа, преустановяване на дейност, загуба на репутация, щета и т.н.

Организациите трябва да идентифицират оперативните последствия от сценариите за инциденти от гледна точка на (без да се ограничава до):

- Време за разследване и възстановяване;
- Загуба на (работно) време;
- Пропуснати ползи;
- Здраве и безопасност;
- Финансови разходи за специфичните умения за възстановяване на щетите;
- Общоприета представа за репутация и добра воля.

4.2 Анализ на риска

4.2.1 Методологии за анализ на риска

Анализът на риска може да бъде разработен в различни степени на детайлност в зависимост от критичността на актива, степента на познаване на уязвимостите и предишни инциденти, свързани с организацията. Методологията за анализ на риска може да бъде качествена или количествена, или комбинация от двете, в зависимост от обстоятелствата. На практика често първоначално се използва качествен анализ за получаване на обща представа за нивото на риска и за разкриване на значителните рискове. По-късно може да се появи необходимост от предприемане на по-специфичен или количествен анализ на значителните рискове, защото обикновено е по-лесно и по-евтино да се направи качествен, отколкото количествен анализ.

4.2.2 Оценяване на последствията

Трябва да бъде оценено въздействието върху бизнеса на организацията, което може да бъде резултат от възможни или деиствителни инциденти със сигурността на информацията, вземайки предвид последствията от пробив в сигурността на информацията като загуба на конфиденциалност, интегритет или наличност на активите.

Оценяването на последствията се определя, като се използват два измерителя:

- Стойността на подмяна на актива: разходи за почистване, за възстановяване и замяна на информацията (ако въобще е възможно) и
- Последствията за бизнеса от загуба или компрометиране на актива, като потенциално неблагоприятни последствия за бизнеса на организацията и/или законови или регулаторни последствия от разкриване, модифициране, липса, неналичност и/или разрушаване на информацията и други информационни активи.

4.2.3 Оценяване на вероятността за инцидент

След идентифициране на сценариите за инцидент е необходимо да се оцени вероятността всеки сценарий да се случи, като се използват качествени или количествени методи за анализ. При това трябва да се вземе предвид колко често се реализира заплахата и колко лесно уязвимостта може да бъде използвана, като се отчита следното:

- Опитът и приложимите статистики за вероятността за заплаха;
- За преднамерените източници на заплаха - мотивацията и капацитета;
- За случайните източници на заплаха - географски фактори, възможност за екстремни състояния на времето и фактори, които могат да повлияят на човешки грешки и на неизправност в съоръжението;
- Уязвимости, както отделни, така и с натрупване;
- Съществуващи механизми за контрол и доколко ефикасно те намаляват уязвимостите.

4.2.4 Установяване на нивото на риска

При анализа на риска се определят стойности на вероятността и на последствията от риска. Тези стойности могат да бъдат количествени или качествени. **Анализът на риска е основан на оценените последствия и вероятността.** В допълнение той може да вземе предвид финансовите ползи, интересите на заинтересованите страни и други променииви, които са подходящи за преценяването на риска. Прецененият риск е комбинация от вероятността за сценарий за инцидент и последствията от него.

Примери за различни методи или подходи за анализ на риска за сигурността на информацията могат да бъдат намерени в приложение към стандарта ISO/IEC 27005 [2].

За допълнителни указание относно методите, които могат да бъдат използвани при подробното оценяване на риска за сигурността на информацията, виж в IEC 31010 [4].

4.3 Преценяване на риска

Нивото на рисковете трябва да бъде сравнено с критериите за преценяване на риска и критериите за приемане на риска, дефинирани при установяването на контекста. В приложение към стандарта ISO/IEC 27005 [2] е даден пример за това.

По време на етапа на преценяване на риска в допълнение към преценените рискове трябва да бъдат взети предвид фактори като договорни, законови и регулаторни изисквания.

5. Проблеми при оценка на риска

- **Липсва информация за конкретни реализации:**

Има много малко литература за прилагането и практиката на УРСИ в организациите. Има две причини за това. Първо, организациите не са склонни да обсъждат практиката за сигурност на информацията, предвид чувствителния характер на темата. Второ,

изследванията, оценяващи ефективността на практиката, изискват изследователите да имат задълбочени познания както за системите за управление на риска, така и за информационните системи [5].

- **Кампанийност:**

Оценката на риска за информационната сигурност обикновено се извършва на равни интервали, но ударно и за кратко време. Няма точни правила за периодичността на оценката на риска и дали да се извършва на точни периоди или само при необходимост. Но има твърде много взаимосвързани неизвестни и известни променливи. Всички те се променят във времето, в зависимост от различни обстоятелства като промени в системата и в бизнеса, трудови спорове, социални и политически промени, неизвестни вреди и успехи на конкуренти, враждебност и нерационалност. Осигуряването на надеждна и постоянно функционираща информация на организацията изисква разбиране на съответните променливи и как промените в околната среда могат да ги засегнат [5].

- **Несигурност:**

Анализът на риска за сигурността на информацията е трудна задача и включва несигурност, която се счита за основен фактор, който влияе върху ефективността на оценката на риска. Върху процеса влияе несигурността, присъща на контекста. Някои съществуващи подходи за анализ на риска за СИ трудно се справят с несигурността [11].

- **Ефективност:**

Предприемането на оценка на риска е скъпа дейност, като се има предвид много големият брой информационни активи в повечето организации, сложният и променящ се характер на риска за сигурността, трудността при точно определяне на вероятността или въздействието. Ето защо очевиден въпрос би бил "колко ефективна е практиката на оценката на риска в организациите, като се имат предвид ограниченията на бюджета за типична информационна сигурност?" [5].

- **Повърхностност:**

Съществуват значителни доказателства, че значителни източници на риск не се вземат предвид по време на процеса на управление на риска. Примери за такива пропуски включват: нематериални активи като знанието; уязвимости, възникващи от сложните взаимоотношения между множество информационни активи; индикации за злонамерени заплахи като измама, шпионаж или саботаж; липса на систематично и непрекъснато обучение от инциденти, свързани с информационната сигурност; неспособност да се идентифицират модели на нападение или да се моделират сложни или постоянни сценарии за атака [5].

- **Грешен избор:**

Не всички потребители са в състояние да оценят проблемите със сигурността на информацията за даден актив и могат да изберат несъществуващ риск. Несъществуващите двойки заплаха-уязвимост могат да накарат организациите да отделят ненужно време и пари, за да предотвратят риска, който може да не се случи, което може да накара мениджъра да пренебрегне реалните слабости или да инвестира в неправилни мерки за сигурност [6].

- **Многовариантност:**

Въпреки че има подготвени списъци с предложения за заплахи и уязвимости, експертите по оценка на риска трябва да направят конкретен избор за даден актив между стотици комбинации [6].

- **Сложност:**

За потребителите е трудно извършването на процеса за оценка на риска поради нужда от външна експертна помощ, загуба на много време и средства и опасност от използване на готови препоръки и предложения поради несъобразяване с конкретните условия. Освен това,

поради увеличаване на видовете заплахи и драстичното нарастване на разнообразието от информационни системи, адаптирането на конкретен модел за сигурност към дадена организация става все по-сложно [10].

- **Много методи:**

Методите за анализ на риска са зависими от характера на актива – т.е. за организации с много и разнообразни активи се ползват много методи за оценка. Освен това не съществуват конструкции, които да подпомагат организациите при определянето на най-подходящия за употреба метод за анализ на риска [9].

- **Субективност:**

Тъй като процесът на оценка на риска е важен, има разработени различни модели за подобряване на методологията. Макар че тези методи помагат осигуряване на общ подход, те не предлагат ясни практически насоки и техният подбор се основава на експертно мнение и субективна преценка. Тази липса на обективни количествени практически насоки засяга приоритизирането и анализа на уязвимите активи [7].

- **Събиране и обработване на данни:**

При невъзможността да останат "вечно бдителни" много организации изпускат жизненоважни данни, свързани с риска. Важно е организациите да съхраняват данни за инциденти, свързани с сигурността на информацията, както и данни, подсказващи за проблеми. Събирането на голямо количество информация, необходима за добра оценка на риска, поставя въпроса за селекция и оценка на качеството на събраната информация [8].

6. Заключение

В днешно време все повече организации изграждат и сертифицират системи за управление на информационната сигурност. Един от тежките за решаване проблеми е управлението на риска за сигурността на информацята. В групата стандарти на ISO по сигурност на информацията са дадени подробни насоки и изисквания, както и много примери, но въпреки това управлението на риска и в частност оценката на риска за СИ остава елемент от СУСИ, който се изгражда с много усилия и средства, като резултатите са съмнителни. В много случаи документално системата е оформена добре и подлежи на сертифициране, но формалното спазване на правилата не води до необходимия резултат. В този материал са описани основните изисквания и дейности по реализация на УРСИ и са разгледани някои проблеми и пречки, които са открити в статии по темата. На базата на описаните проблеми могат да се направят следните **изводи и препоръки** за бъдещи мерки за повишаване на ефективността от УРСИ:

- Процесът ще спечели ефективност, ако се изведе от сферата на експертността и резултатите от него се доведат до вид по-лесен за ползване от ръководството на организацията;
- Огромният обем информация, която се ползва от процеса, трябва да бъде събирана, съхранявана и обработвана с подходящо внимание и пълнота;
- Автоматизиране на процеса по събиране на доказателства за извършени дейности по УРСИ и история на инциденти със СИ, ще доведе до пълнота, облекчение на труда и по-лесен анализ на информацията.

7. Речник на термините

Сигурност на информацията (СИ) означава запазване на поверителност, цялостност и наличност на информацията**;** могат също така да бъдат включени и други характеристики като автентичност, отчетност, неотхвърляне и надеждност.

Риск е ефект на несигурност по отношение на целите.

Контекст е среда, в която организацията се стреми да постигне своите цели.

Заплаха е потенциална причина за нежелан инцидент, който може да доведе до увреждане на дадена система или организация.

Уязвимост е слабост на даден актив или контрола, които могат да бъдат използвани от една или повече заплахи.

Механизмите за контрол на СИ включват всеки процес, политика, процедура, указание, практика или организационна структура, които изменят рисковете за сигурността на информацията и могат да бъдат административни, технически, управленски или юридически по своята същност.

Литература

[1]. BDS, БДС ISO/IEC 27001. Bulgaria, 2014, p. 30.

[2]. BDS, БДС ISO/IEC 27005:2012. Bulgaria, 2012, p. 80.

[3]. BDS, БДС ISO 31000:2011. Bulgaria, 2011, p. 32.

[4]. BDS, БДС EN 31010:2010. Bulgaria, 2012, p. 103.

[5]. Webb J., A. Ahmad, S. B. Maynard, G Shanks, A situation awareness model for information security risk management. //Computers & Security,Volume 44, July 2014, Pages 1-15.

[6]. Yu-Chih Wei, Wei-Chen Wu, Ya-Chi Chu, Performance evaluation of the recommendation mechanism of information security risk identification. // Neurocomputing, Volume 279, 1 March 2018, Pages 48-53.

[7]. Nadher Al-Safwani, Yousef Fazea, Huda Ibrahim, ISCP: In-depth model for selecting critical security controls. //Computers & Security, Volume 77, August 2018, Pages 565-577.

[8]. Palaniappan Shamala, Rabiah Ahmad, Ali Zolait, Muliati Sedek, Integrating information quality dimensions into information security risk management (ISRM), // Journal of Information Security and Applications, Volume 36, October 2017, Pages 1-10

[9]. Palaniappan Shamala, Rabiah Ahmad, Mariana Yusoff, A conceptual framework of info structure for information security risk assessment (ISRA), // Journal of Information Security and Applications, Volume 18, Issue 1, July 2013, Pages 45-52;

[10]. Suleyman Kondakci, Analysis of information security reliability: A tutorial, // Reliability Engineering & System Safety, Volume 133, January 2015, Pages 275-299

[11]. Ana Paula Henriques de Gusmão, Lúcio Camara e Silva, Maisa Mendonca Silva,Thiago Poleto, Ana Paula Cabral Seixas Costa, Information security risk analysis model using fuzzy decision theory, // International Journal of Information Management, Volume 36, Issue 1, February 2016, Pages 25-34

За контакти:

Инж. Петко Генчев

Катедра „Компютърни науки и технологии”

Технически университет – Варна

Email: p.genchev@tu-varna.bg

Генериране на изображения чрез визуализация на отличителните черти на конволюционни невронни мрежи

Константин С. Георгиев

Резюме: Тази разработка демонстрира различни методи за генериране на изображения според пространството от структури, научено от една конволюционна невронна мрежа. Това ни дава възможност за задълбочено изследване на отделните слоеве на мрежата и характеристиките в едно входно изображение, спрямо които тя реагира. За целта се прилага итеративно увеличаване на градиента на един избран слой, което разширява откритите белези и трансформира изображението. На анализ са подложени известните архитектури на GoogLeNet и VGG, спечелили призови места в състезанието за визуално разпознаване ImageNet 2014.
Ключови думи: градиент, генериране, характеристики, визуализация, конволюционна невронна мрежа, VGG16, Inception, DeepDream

Image generation through visualizing features of Convolutional Neural Networks

Konstantin S. Georgiev

Abstract: This paper demonstrates different methods for generating images in terms of the space of patterns learned by a Convolutional Neural Network. This gives us the opportunity for a deeper exploration of various layers in the network and the features of an input image, according to which it responds. For that purpose, the gradient of a selected layer is increased iteratively, which amplifies the recognised patterns and transforms the image. Subjects of analysis in this case are the widely popular GoogLeNet and VGG architectures, famous for winning first prizes in the large-scale visual recognition competition ImageNet 2014.
Keywords: gradient, generation, features, visualization, Convolutional Neural Network, VGG16, Inception, DeepDream

1. Цел и значение на разработката

Невронните мрежи предизвикаха забележителен прогрес в области като класификацията на изображения и гласовото разпознаване през последните години. За сметка на това обаче, тези мрежи стават все по-сложни, броят на нивата и слоевете им се разраства и става все по-трудно да се проследи какво точно се случва във всеки един слой. Една конволюционна мрежа извлича характеристиките на едно входно изображение на все по-високо ниво с нарастване на слоевете. Например, първият слой на мрежата може да разпознава по-прости форми като ъгли или върхове. Междинните слоеве, тогава, следва да извличат цялостни компоненти и обекти, а финалният слой комбинира всичко, научено от предишните, за да формира цялостна интерпретация за това, което се вижда в изображението [1].

Визуализацията ни помага да разберем дали мрежата е научила очакваните от нас белези, като ни дава възможност да видим нейната интерпретация за обекта, който е обучена да разпознава. Идеята е мрежата сама да вземе решение кои белези да увеличи и до каква степен, като ѝ подадем единствено входно изображение и име на слой, чрез който да извърши генерирането. При избор на слой от по-ниско ниво мрежата има тенденцията да продуцира щрихове и прости декоративни форми, докато при по-дълбоките слоеве се изрисуват и цели обекти [1].

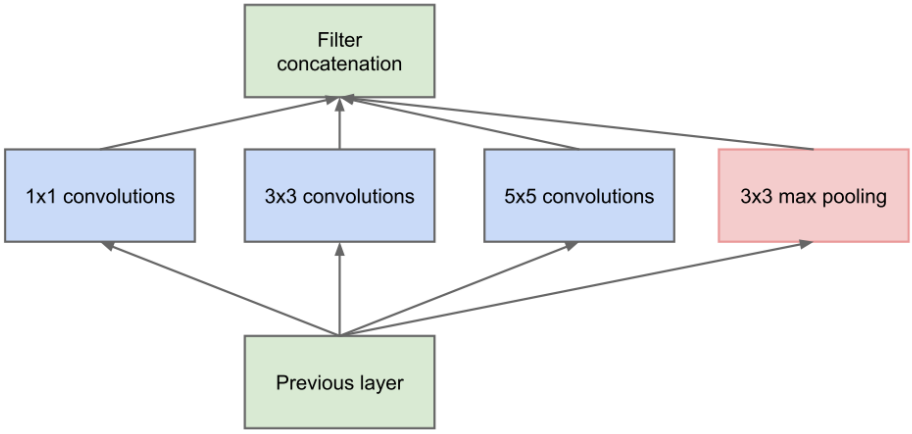
В тази разработка ще се разгледат няколко алгоритъма за визуализация на тези отличителни черти, които ще се приложат върху различни архитектури и слоеве и ще се посочат техните предимства и недостатъци от гледна точка на време за изпълнение и качество на продуцираните изображения.

2. Програмни средства

Разработката е създадена на езика Python – версия 3.6 и среда Jupyter Notebook – open-source web приложение, поддържащо голямо разнообразие от пакети за анализ и моделиране на данни. Също така е приложена и широко използваната библиотека Tensorflow (GPU версия 1.10.0), която внася значителни оптимизации в изчисленията (в този случай на градиенти), построявайки предварителен граф, съдържащ последователността от изчислителни операции. Това дава възможност за многоядрено изпълнение на изчисленията. Също така са използвани и други допълнителни пакети за математически и статистически анализ и обработка на изображения като Numpy, Scipy, PIL.Image и др.

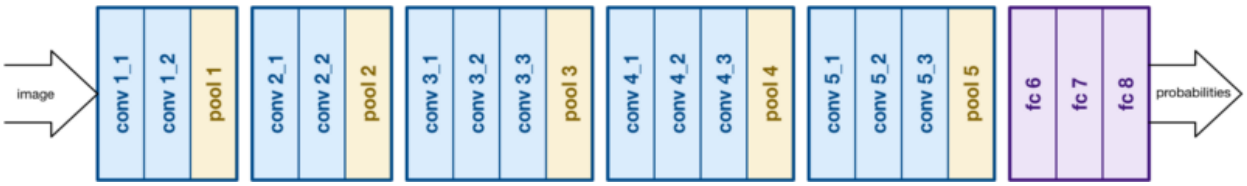
3. Избрани архитектури на невронни мрежи

Подложени на анализ, поради тяхната сложност и точност при разпознаване, са широко известните GoogLeNet и VGG, спечелили призови места в състезанието за визуално разпознаване ImageNet 2014. GoogLeNet, известна още като Inception, е носител на първото място в състезанието и се характеризира с изключително големия си брой конволюционни слоеве и нива на вложеност. Известна е с това, че използва различни комбинации от конволюционни филтри с различни размери, които се изравняват до линейни [4].



Фиг. 1. Комбинация от различни конволюционни филтри в Inception

VGG16 печели второ място в състезанието и е широко предпочитана заради своята проста линейна архитектура, състояща се само от 16 конволюционни слоя. Основният недостатък тук е в броя параметри за настройка, който надвишава 138 милиона.



Фиг. 2. Архитектура на VGG16

4. Имплементиране на алгоритмите за визуализация на отличителни черти

4.1. Формиране на частични градиенти

Обработката на изображения с много висока резолюция води до изчерпване на изчислителната памет на процесора. За да се избегне това, едно възможно решение е входното изображение да се раздели на множество по-малки части (*x'* и *y'*) и итеративно да се сумира градиентът за всяка отделна част. Също така е добре да се избере случайна част от изображението при всяка итерация, за да се избегнат видими линии и частици от изрязаното изображение. Като вход е необходимо да се подадат размер на частта и стартова позиция за координатите *x* и *y* – *xs* и *ys*. Като резултат от това могат да се изчислят и крайните позиции *xe* и *ye*.

$$\begin{aligned} &x', y' = 400 \\ &xs \in (-\frac{3}{4}x', -\frac{1}{4}x') \\ &ys \in (-\frac{3}{4}y', -\frac{1}{4}y') \\ &xe = xs + x' \\ &ye = ys + y' \end{aligned} \tag{1}$$

Тези стартови позиции осигуряват, че отделните части ще са поне една четвърт от цялото изображение. По-малки части биха внесли шум в стойностите на изчислените градиенти. След това остава само да се изчисли градиентът за текущата частица и да се премине към следващата, докато не се достигне максималната стойност на координатите – тези на оригинала. Това става, като в края на всяка итерация се приложи:

$$\begin{aligned} &xs = xe \\ &ys = ye \end{aligned} \tag{2}$$

4.2. Изчисляване на градиентите

Tensorflow поддържа автоматично изчисляване на градиентите за едно входно изображение относно избран модел и негов конволюционен слой, като прилага правилото за диференциране на сложни функции. Като вход единствено трябва да се подадат име на тензор (многомерен масив съдържащ част от оригиналното изображение) и средната стойност на тензора *t* (съдържащ стойностите на каналите на избраният конволюционен слой от мрежата) - t_m. За увеличаване размера на белезите в изображението може преди това стойностите на тензора *t* да се увеличат със степен на 2.

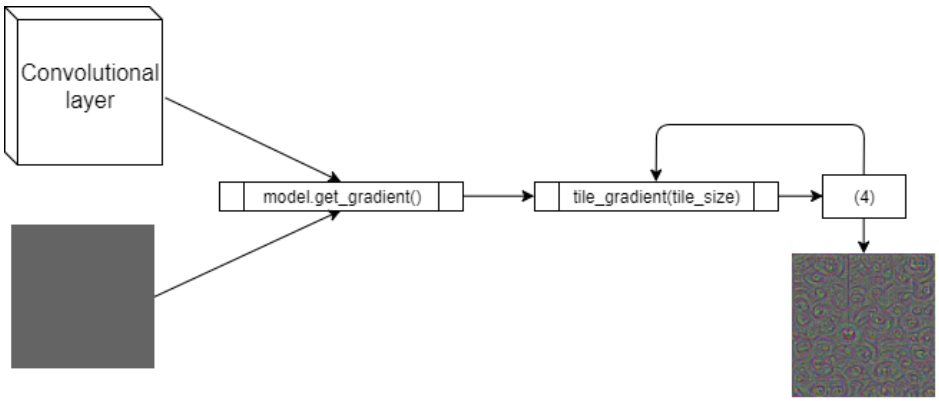
$$t_m = \frac{\sum_{i=1}^{n} t_i^{\,2}}{n} \tag{3}$$

4.3. Градиентно изкачване

Най-простият алгоритъм за генериране на отличителни черти се постига с метода на градиентно изкачване, приложен върху случайна част от изображението за определен брой итерации. Първата стъпка е определяне на формулата на изчисление на градиента за конкретен конволюционен слой с помощта на Tensorflow. Следва изчисляване на този градиент върху случайна част от изображението чрез (3). Полученият градиент се

нормализира за съвместимост с различни конволюционни мрежи и накрая се прибавя към оригиналното изображение *t*, умножено по някаква стъпка α [6].

$$\nabla t = \sigma_{\nabla} + \frac{1}{10^8}$$
$$t = \nabla t.\alpha \tag{4}$$

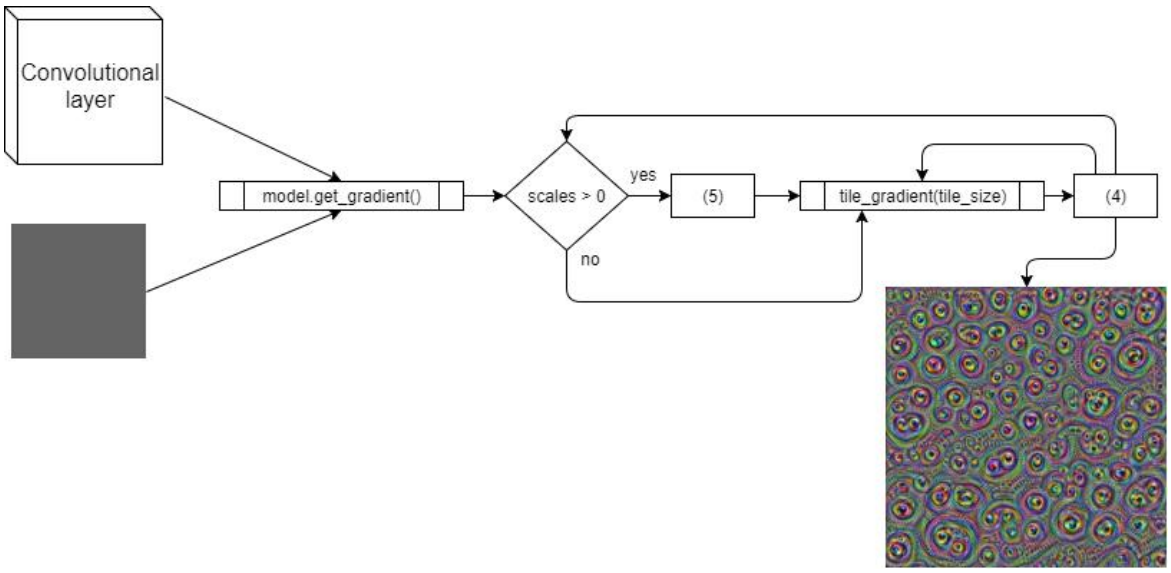


Фиг. 3. Градиентно изкачване

4.4. Градиентно изкачване с мултискалиране на изображението

За да се получи по-качествена резултатна картина за по-малки изображения, едно възможно решение е да се увеличат итеративно всички измерения на изображението *t*, като се умножат по някакъв зададен фактор *s* [6].

$$t_x = t_x.s$$
$$t_y = t_y.s$$
$$t_z = t_z.s \tag{5}$$

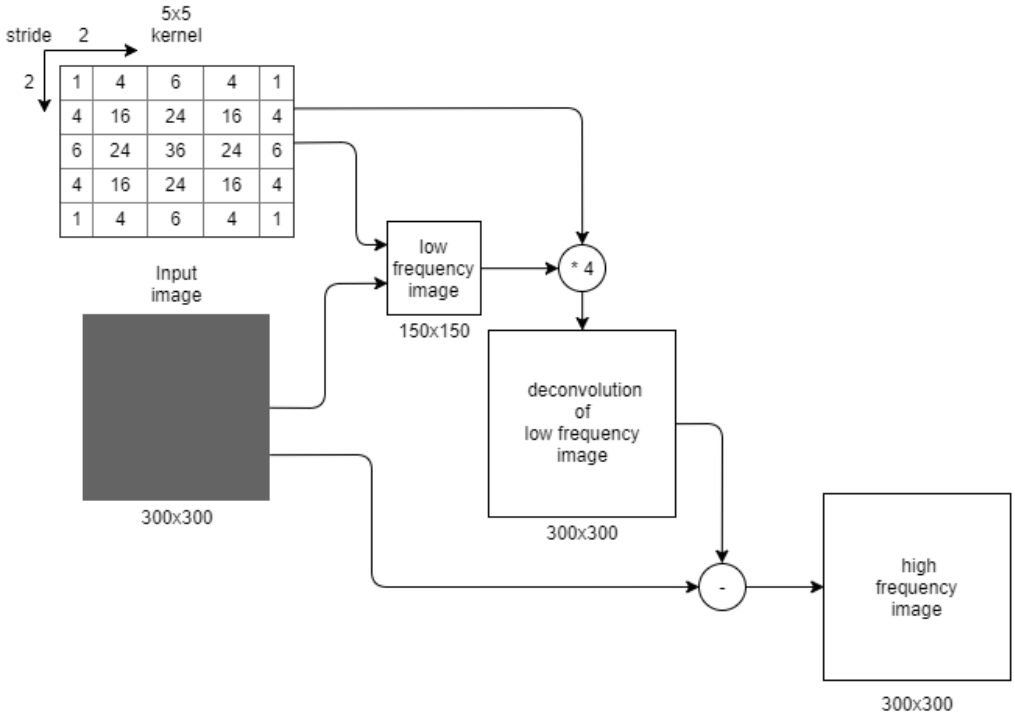


Фиг. 4. Градиентно изкачване с мултискалиране

Полученото качество е значително по-добро от обикновеното градиентно изкачване и вече ясно могат да се проследят белезите, които конволюционният слой наслагва върху изображението. Основният недостатък на алгоритъма е, че отново възниква опасността от изчерпване на RAM памет при изпълняване на алгоритъма с многократно скалиране.

4.5. Нормализация на градиентите чрез пирамидата на Лаплас

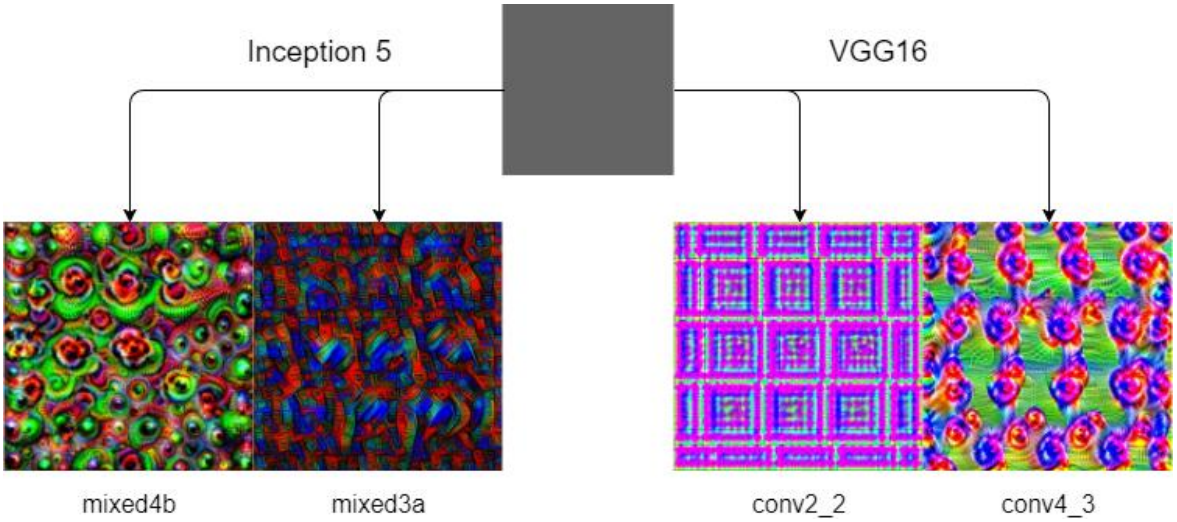
Полученото изображение при мултискалирането е добро, но има възможност то още да се детайлизира, като се изравнят неговите ниски и високи честоти. Пирамидалната декомпозиция на Лаплас позволява тези честоти да се разделят на два класа и селективно да се подсилят градиентите на ниските честоти [3]. Това изглажда финалното изображение и значително усилва интензитета на цветовете. За целта изображението итеративно се замъглява и се намалява неговата резолюция по някакъв зададен фактор за всяко ниво. Преди края на всяка итерация се запазва разликата между предишното и текущото изображение, за да може след това то да се реконструира обратно.



Фиг. 5. Процес на декомпозиция с пирамидата на Лаплас

Процесът включва три основни етапа: декомпозиция, нормализация на градиентите и реконструиране на изображението. Декомпозицията се осъществява с 5x5 конволюционно ядро, което се прилага със стъпка 2 и по двете оси на оригинала. Така се получава замъглено изображение с ниска резолюция. За да се получат и високочестотните сигнали, резултатното изображение се възстановява чрез деконволюция с фактор 4 и се изважда с оригинала. Процесът на реконструиране е огледален на този, с тази разлика, че крайният резултат е сборът между високочестотното и деконволюираното изображение [6].

Получените резултати за съответните модели са следните, показани на фигура 6. Виждаме че Inception 5 и VGG16 си приличат по това че разпознават прости форми като ъгли, извивки и ръбове в първите си конволюционни слоеве, а при крайните вече изрисуват и цели обекти(в този случай животни). По този начин можем да извлечем и полезна информация за етапа на тренировка на нашите модели. Например, слоят conv2_2 на VGG16 най-вероятно е приложим за разпознаване на ръбове и ъгли на обекти.

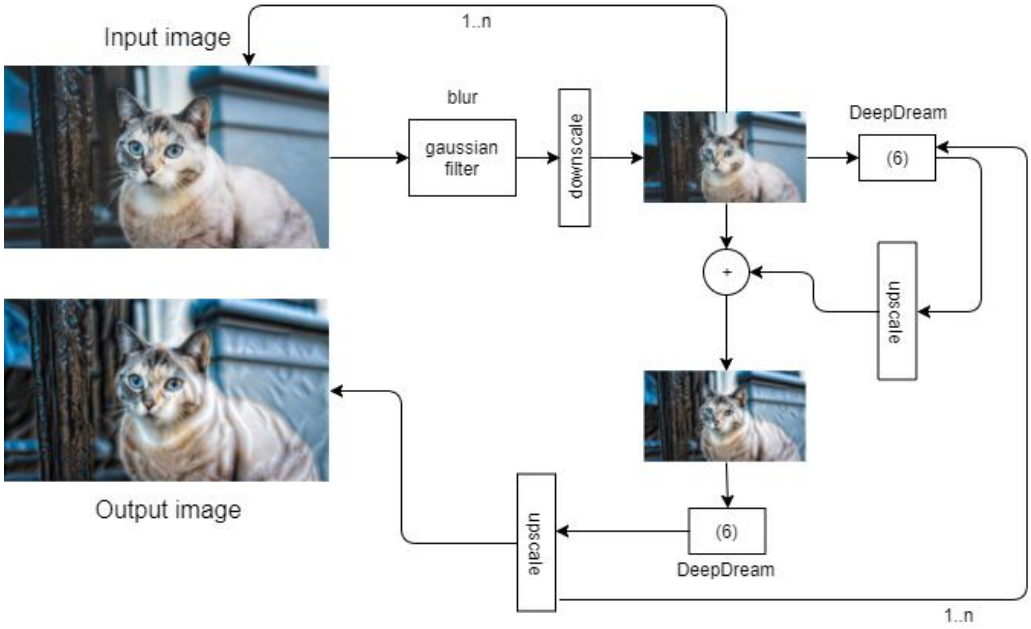


Фиг. 6. Генериране на изображения с пирамидата на Лаплас

4.6. DeepDream

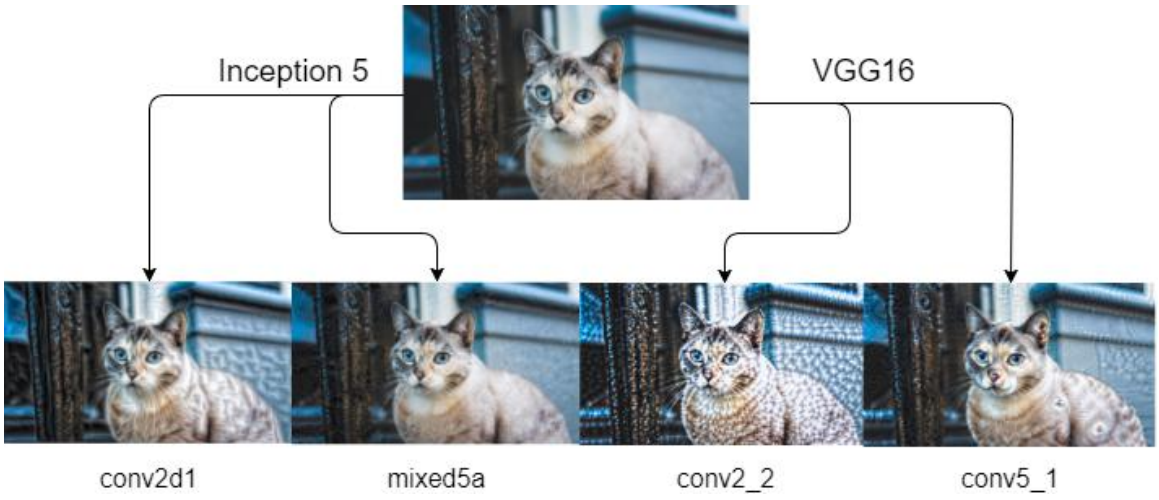
DeepDream алгоритъмът е създаден от Google и дава възможност за генериране на артистични изображения с различни стилове с изключително високо качество на картината. Тези стилове, разбира се, отново се базират на конволюционните слоеве на избрания модел. Идеята е подобна на алгоритъма на пирамидата на Лаплас, но освен това има и някои допълнителни стъпки на филтриране. Изображението тук се оптимизира рекурсивно, поради многократното намаляване и увеличаване на картината. Самата оптимизация се състои от прилагане на три Гаусови филтъра – g_1, g_2, g_3, приложени с различни стандартни отклонения. Тези филтри се прибавят към градиента на изображението, стъпката на градиента се нормализира и накрая резултатът се прибавя към оригинала. Стандартното отклонение на филтрите се променя според текущата итерация i, и общия брой итерации n, а x и y са съответно дължината и ширината на входното изображение [5].

$$
\begin{aligned}
\sigma_1 &= \frac{4.i}{n+0.5} \\
\sigma_2 &= 2.\sigma_1 \\
\sigma_3 &= \frac{\sigma_1}{2} \\
g_1(x_i, y_i) &= \frac{1}{2\pi\sigma_1}.e^{-\frac{x_i^2+y_i^2}{2\sigma_1^2}} \\
g_2(x_i, y_i) &= \frac{1}{2\pi\sigma_2}.e^{-\frac{x_i^2+y_i^2}{2\sigma_2^2}} \\
g_3(x_i, y_i) &= \frac{1}{2\pi\sigma_3}.e^{-\frac{x_i^2+y_i^2}{2\sigma_3^2}} \\
\nabla_t &= g_1 + g_2 + g_3 \\
\alpha &= \frac{\alpha}{(\sigma_\nabla + 10^{-8})} \\
t &= \nabla_t.\alpha
\end{aligned}
\qquad (6)
$$



Фиг. 7. Рекурсивна оптимизация с DeepDream

Ето и някои интересни резултати, получени от различни конволюционни слоеве на моделите:



Фиг. 8. DeepDream върху двата модела

От примера може да проследим как изглеждат белезите на избраните слоеве. Началните слоеве видимо указват по-голям ефект върху изображението, защото откриват много на брой прости форми. По-дълбоките слоеве, от друга страна, изграждат прозрачни обекти, които се сливат с обектите в картината.

5. Заключения и изводи

На базата на направения анализ може да се направи извода, че визуализацията на отличителни черти на конволюционни мрежи е полезно средство за проследяване какво точно се случва във всеки слой на мрежата и дали този слой е правилно обучен. От друга гледна точка тези алгоритми придават интересни ефекти на изображенията и е възможно те да се развият и да се използват за изкуствено рисуване на картини [2].

6. Благодарности

Специални благодарности към Магнус Е.Х. Педерсен, който е автор на алгоритъма за рекурсивна оптимизация с DeepDream, както и за неговите примери за работа с Tensorflow и построяване на конволюционни мрежи.

Литература

[1]. Mordvintsev, A., Olah, C., Tyka, M., Inceptionism: Going Deeper into Neural Networks, 2015
[2]. Spratt, M., E, Dream Formulations and Deep Neural Networks: Humanistic Themes in the Iconology of the Machine-Learned Image, 2017, с. 14-17
[3]. Prof. Jepson, A., Lecture notes on Image Pyramids, 2005
[4]. Szegedy, C., Wei, L., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan D., Vanhoucke V., Rabinovich A., Going Deeper with Convolutions, 2014
[5]. https://github.com/Hvass-Labs/TensorFlow-Tutorials
[6]. https://github.com/tensorflow/tensorflow/tree/master/tensorflow/examples/tutorials/deepdream

За контакти:
Константин С. Георгиев
катедра „Компютърни науки и технологии“
Технически университет – Варна
E-mail:dragonflarex@mail.bg

ESTIMATION OF PROBABILITY OF DEFAULT FROM INTERNAL DATA

Ventsislav Nikolov, Nikola Vasilev

Abstract: In recent years, finance institutions robustly need an instrument for risk management. From several committees on banking supervision require that institutions must have reliable rating scale for probability of default. The most important step is the transition towards Internal Ratings-Based (IRB) approach. This paper presents an approach to estimate implied probability of default (PD) and classify into desired credit scale. The calculation of PD is based on Newton's method and classification is done by competitive trained neural network.
Keywords: Credit rating, Internal rating-based approach, Newton's method, Classification, Self- organizing map.

1. Introduction

Probability of default (PD) is finance term, which describes possibility of bankruptcy in time horizon. In most cases, that horizon is one year. PD is widely used in a variety of credit analysis and risk management scenarios [1]. Results provided by PD are Loss given default (LGD), Expected loss and exposition associated with default. The loss value could be interpreted as fund, whose borrower does not desire or could not afford to pay the debt in time. The probability of default could be estimated by variety of methods, including discriminant function [2], logistic model, probability model, Panel analysis, Cox's model, decision trees and neural networks [4]. In this paper shown an approach based on heuristic algorithm to estimate an implied PD for each contract and neural network of class Self-organizing map (SOM) to clustering and classification of contracts in desired credit scale.

Main goal of subject is to comply with regulatory requirements specified in Regulation (EU) No 575/2013 of the European Parliament and of the Council of 26 June 2013 [3]. In summary, the module should provide automatic creation of credit scale and computation of PDs for credit contracts based on real cooperative and government data. This module should be portable into different software products.

The need of new methods for credit rating calculation emerges, because followed reasons:

• Reaction of credit agencies, sometimes are too slow, but the market is dynamic;

• The evaluation is done in relatively long periods. Results, obtained daily, may acquire such a significant benefit;

• Institutions could make an own rating scale to classify their borrowers. Thus, the credit rating is more precised and helps the decision making process.

2. Module architecture

Realization of the system must be independent to input data and separated of modules for easiest build and future support. Input data for algorithms is numeric data for expected loss, credit contract details and rating scale. The conceptual design of the architecture is shown in fig. 1. Mainly modules are three:

• Portfolio evaluator – output from this stage is pair of data – ID of contract and implied PD

• Clustering and Optimization – with data from previous stage, based on artificial intelligence, contracts are classified by desirable rating scale

• Aggregating the result, and returns the rating scale followed by probabilities of default.

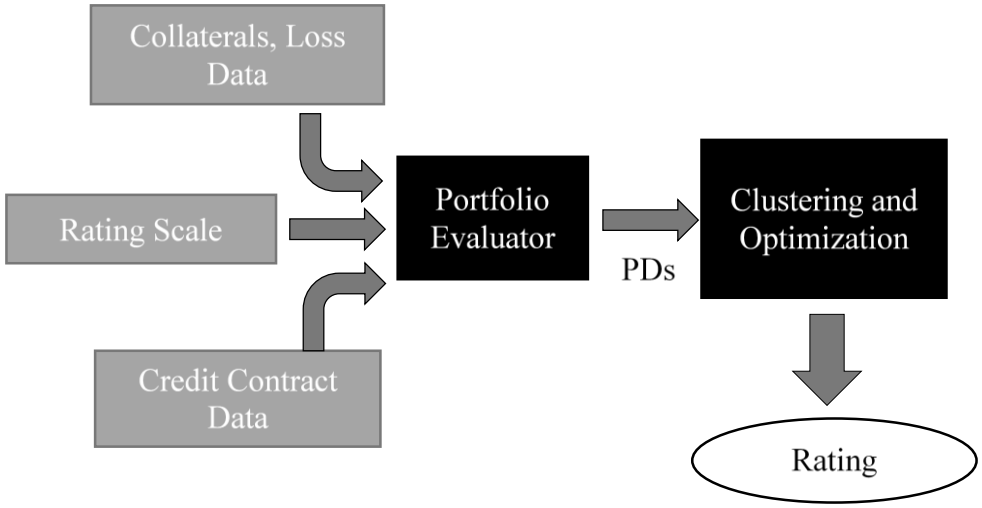


Fig. 1. Architecture of module

The first stage of mathematical modules is to choose rating scale. Different institutes work with different scale. This scale contains mainly lowercase or uppercase words. Some agencies classify their borrowers in groups of long or short term, cooperative or private. Next stage is to build portfolio model and to obtain PD for each contract. Then data is parsed for self-organizing map and then to be classified. Result from classification is to separate positions into different groups and computing the average PD.

3. Newton based optimizer for calculation of PD

The main idea of this algorithm is analysis type "what-if" [7]. End product is to obtain target amount, which is followed by constrains and conditions. Goal of this approach is to return the global minimum of given function, based on adaptive search method. This is accomplished via adaptive step. With every iteration, this step decrease. The optimizer works with general interface function, which condition is described by group of formulas and result is difference between target and estimated value. Pseudo code is shown in fig. 2.

The goal of this optimizer is to compute implied probability of default for given position – fig. 3. The search for global minimum is absolute value of difference between provisions and computed expected loss via formula. Formula for expected loss is shown in (1):

$$EL = \sum_{t=0}^{k} e^{\frac{rT}{360}} LGD_i MPD_i \tag{1}$$

where:

r is rate;

T is time in days;

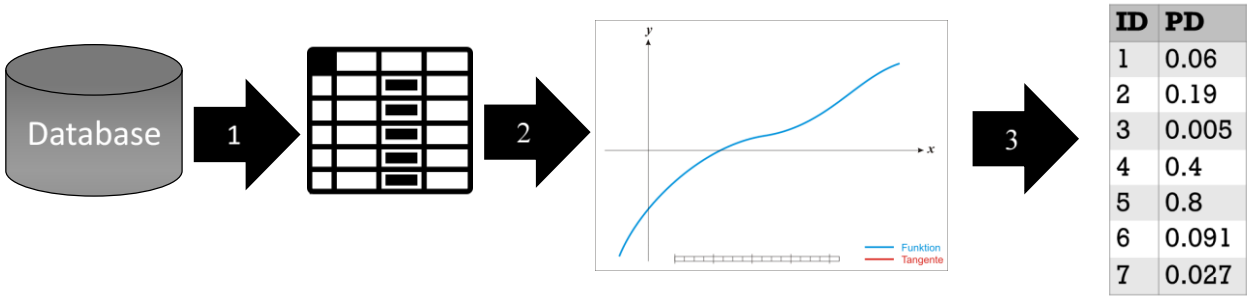
LGD_i is Loss given default for leg I;

MPD_i is Marginal PD, which is difference between current and previous interpolated PD;

The pseudo code of the algorithm is shown below.

```
searchMinimum () {
  for (num_search_iterations) {
    while (true) {
      double error = calculateError(currentValue);
      if (error is decreasing) {
        increase the step;
        currentValue = currentValue + step;
      } else {
            break;
      }
    }
    currentValue = currentValue - step;
    decrease the step;
    continue searching in the descending direction;
  }
  return currentValue;
}
```

Fig. 2. Pseudo code of Newton based optimizer



ID	PD
1	0.06
2	0.19
3	0.005
4	0.4
5	0.8
6	0.091
7	0.027

Fig. 3. Workflow of Newton based optimizer

4. Clustering with self-organizing map

This article uses the self-organizing map (SOM) [5] to obtain credit rating scale followed by PDs. Algorithm overview is shown on figure 4. This type of network is organized with connections, which every input node is completely connected with everyone in network. Nodes are representation of a single credit rating. Input data for this algorithm is pair of data, which contains ident for contract and implied PD. Also for the algorithm, the rating scale is necessary (1). Scale is key parameter to build lattice. Each node has specific mapped position (coordinates by X and Y) and has a vector of weights. In current solution lattice looks like vector, because size by Y is one. The main reason for choosing this approach for clustering in that, this network does not require predefined data.

After building the lattice, the next step is to fill the classifier with input data (2). The idea of SOM is make zones, which contains same or likely credit contracts. For example, in current case these are contracts with same PD. At the begining the contracts are randomized onto the grid. To be more precisely network should be trained (3). SOM evolves in epoch, with increasing the number of epochs, the solution will be more correct. This kind of approach have saturation point. The goal of training is to obtain grouped positions for each credit rating. After that is usage of learning rate, which is valueless variable and decreasing with time. Another key variable is radius of neighbourhood (h_t), which is also decrease. Size of node could be realized as range of each credit

scale. Classification is focused to that to find the smallest distance between input contract and distributed scale [6]. This distance can be mean as range factor for determining a contract to credit scale. In math is distance is known as Euclidean distance. In fact, of unavailability of decay factor formula looks as:

$$d = \sqrt{\sum_{i=0}^{k} (V_i - R_i)^2} \quad , \tag{2}$$

where:

V is vector that contains probabilities for each contract;
R is vector that contains random weights;
K is length of vector.

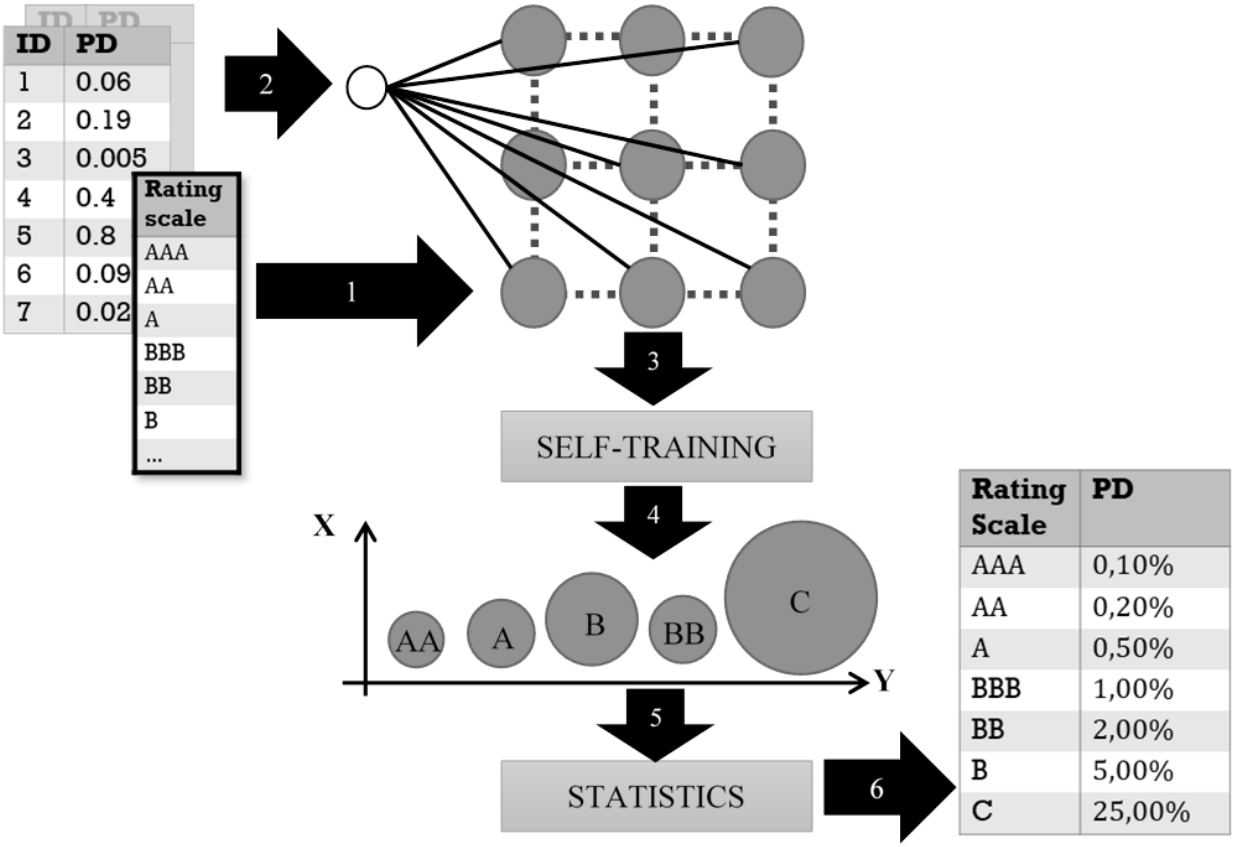


Fig. 4. Workflow of Clustering with self-organizing map

Correction to other nodes is accomplished with amount of influence:

$$\theta_t = e^{-\frac{d^2}{2h_t^2}} \tag{2}$$

After classification of input contracts, a result of the clustering stage is obtained, which contains IDs of contracts. In figure 4, the size of radius determines the size of each cluster. Main idea is similar PDs to be classified to one group.

5.1 Test of implied PD calculator

Tests are done by export of random position in spreadsheet software (e.g. Microsoft Excel). Thus position is evaluated by hand in the table and for computation of implied PD is used SOLVER. Result from algorithm and from excel model must be equivalent.

5.2. Performance test of implied PD calculator

In fig. 5 configuration of performance test for computation of implied PD is shown. It is noticeable that there is a linear trend of time increasing. There is neither large decrease nor increase jumps in the following test.

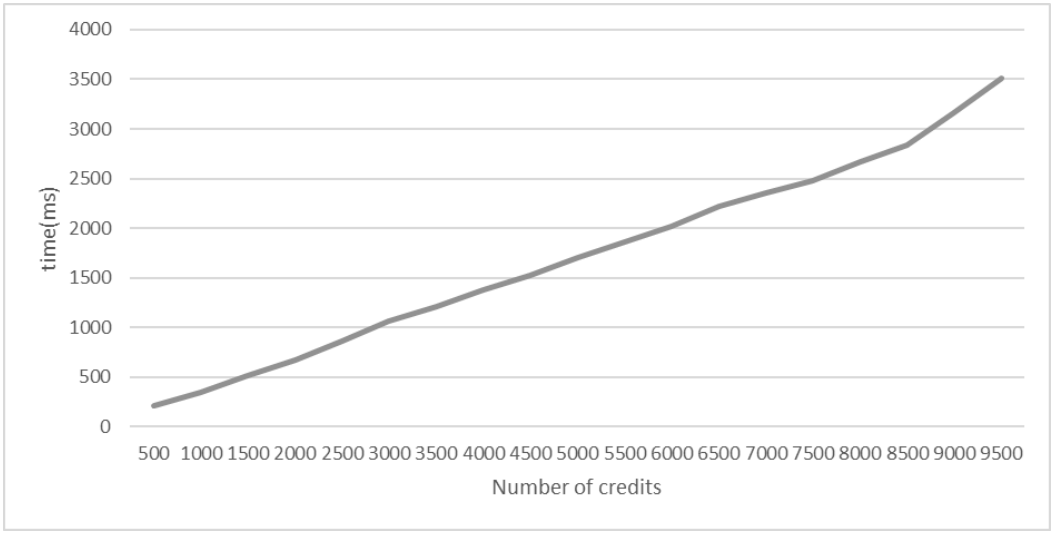


Fig. 5. Performance tests for computation of implied PD

5.3. Performance test clustering

From followed analysis can be noticed that with increasing of number of position and epoch, time is increased linearly. From table 1, it can be seen that the increase step is linear.

Table 1. Performance tests for clustering and building credit scale

Time (ms)		Epochs				
		250	500	1000	2000	5000
Position number	366	111	127	171	404	875
	693	103	159	318	774	1654
	1107	223	275	521	1061	2604
	2262	382	626	1146	2212	5444
	3417	567	926	1741	3634	8205

6. Conclusion and future work

The prototype of the module, which is retaliated in the described approach, is developed in Java. Module is integrated in software third party product. Targeted rating scala is used for building a transition matrix. Generated scale is column D in figure 6.

[%]	AAA	AA	A	BBB	BB	B	CCC	D
AAA	76,483617	6,628530	4,368130	4,189669	4,860872	1,718585	1,692698	0,057899
AA	6,628530	76,914201	3,020833	4,699269	3,129547	2,464819	2,630436	0,512364
A	4,368130	3,020833	70,906982	8,503522	7,146401	3,737581	1,042417	1,274134
BBB	4,189669	4,699269	8,503522	68,690014	6,961144	1,783850	2,624128	2,548405
BB	4,860872	3,129547	7,146401	6,961144	65,890968	5,297405	1,553352	5,160312
B	1,718585	2,464819	3,737581	1,783850	5,297405	63,842628	7,684537	13,470595
CCC	1,692698	2,630436	1,042417	2,624128	1,553352	7,684537	50,945947	31,826485

Fig. 6. Usage of generated credit scale.

The computational part is implemented as a JAR library, which can be used either directly or as a module in business layer logic of other software systems or as service operations. This module has complied requirements noted in EU regalement for credit institutes. In the process of development are used patterns for software design, software frameworks and libraries, which provide readability and easy support. In future development could have been developed multithreaded approach and optimization of mathematical calculation. Also could been built a special scheme for difference between private and cooperative contract.

References

[1]. CreditMetrics ® Monitor, Third Quarter 1999

a. Antonov, Securitization of Credit Portfolios, 7th European Credit Risk Conference June 2002

[2]. Regulation (EU) No 575/2013 of the European Parliament and of the Council of 26 June 2013

[3]. Franz Rothlauf, Design of Modern Heuristics: Principles and Application (Natural Computing Series), Springer 2011th Edition

[4]. Teuvo Kohonen, Self-Organizing Maps 3rd Edition, Springer

[5]. V. Nikolov, An Implied Rating Software System, 2017

[6]. Franz Rothlauf, Design of Modern Heuristics: Principles and Application (Natural Computing Series), Springer 2011th Edition

For contacts:
Assist. prof. Dr. Ventsislav Nikolov
Computer Science and Engineering Department
Technical University of Varna.
E-mail: v.nikolov@tu-varna.bg

Nikola Vasilev
Eurorisk Systems Ltd.
31, General Kiselov Str., 9002 Varna, Bulgaria
E-mail: nvasilev@eurorisksystems.com

ПОДХОД ЗА ИЗГРАЖДАНЕ НА КЛАС СОФТУЕРНИ СИСТЕМИ ЧРЕЗ КОМБИНАЦИЯ ОТ СОФТУЕРНИ ШАБЛОНИ

Димитричка Ж. Николаева, Виолета Т. Божикова

Резюме: Софтуерните шаблони (СШ) са признати като инструмент, който може да бъде полезен при решаването на повтарящи се проблеми, възникнали в реалната програмистка практика. Докладът коментира ползите от прилагане на комбинации от СШ, представя възможните комбинации от шаблони за проектиране, предлага класификация на комбинациите от СШ и представя подход за изграждане на клас софтуерни системи чрез комбинация от СШ. Докладът също така представя резултати от използването на подхода, а именно за реализация на информационна система (ИС) за целите на морския транспорт, както и автоматизирана оценка на разработения подход.
Ключови думи: Софтуерни шаблони, Комбинация от Софтуерни шаблони, Софуерно преизползване, Софтуерно инженерство

Approach to build a class of Software Systems through a Combination of Design Patterns

Dimitrichka Zh. Nikolaeva, Violeta T. Bozhikova

Abstract: Design Patterns (DPs) are recognized as a tool that could be useful in solving repetitive problems that have arisen in real programming practice. The report commented on the benefits of using Combinations of DPs, presented the possible combinations of DPs, offered a classification of DPs combinations, and presented an approach to build a class of software systems through a combination of DPs. The paper also presents real results from the use of the approach, namely the realization of an Information System (IS) for the purposes of Maritime Transport, as well as an Automated Assessment of Approach based on a combination of DPs.
Keywords: Design Patterns, Combination of Design Patterns, Software Reuse, Software Engineering

1. Увод

Повишените изисквания на бизнеса при създаване на нови приложения, използвани за различни бизнес цели, поставя разработчиците пред реални проблеми със скоростта и автоматизацията на процеса на разработване на тези приложения. Един от възможните начини за решаване на този проблем е използване на СШ. Това са решения на общи проблеми при програмирането. Използването им се извършва самостоятелно и в комбинация с други СШ [11]. Това помага да се използва повторно програмния код, което увеличава производителността на разработчиците на софтуер [1-18].

Комбинираното използване на СШ при изграждане на софтуер има следните предимства:

• Повторно използване на код и автоматизация, приложими за конкретна софтуерна разработка или нейното разширяване [13];

• Гъвкав, разширяем и модулен дизайн на приложенията [1, 2];

• Бързо и лесно изграждане на приложения, поради възможността за преизползване;

• Модуларност и по-добро повторно използване на кода. Последният става по-лесен за поддръжка, разходите са по-ниски и времето за пускане на пазара е по-кратко [14].

2. Комбинирано използване на СШ (КСШ)

КСШ е съвкупност от използването на множество шаблони заедно. Въпросите, които възникват по отношение на тяхното прилагане, са: Какви са възможните комбинации между софтуерните шаблони? Как се свързват софтуерните шаблони помежду си? Как се класифицират според връзките помежду им? Как се прилагат на практика в изграждане на софтуер? В Таблица 1 е представена извадка от анализ на литературни източници, в които се използват комбинации от СШ от групата на четиримата (GOF) [12].

Таблица 1.

№ статия/ СШ	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
1								X	X												X			
2			X																		X			
3								X													X		X	
6								X	X												X			
9		X	X		X						X					X		X		X				
10												X												X
15			X			X																		
16																		X	X					
16															X					X				
16									X											X				
16												X								X				
ПБКСШ	X				X					X				X										

Забележка: Abstract factory[1], Builder[2], Factory method[3], Prototype[4], Singleton[5], Adapter[6], Bridge[7], Composite[8], Decorator[9], Facade[10], Flyweight[11], Proxy[12], Chain of responsibility[13], Command[14], Interpreter[15], Iterator[16], Mediator[17], Memento[18], Observer[19], State[20], Strategy[21], Template method[22],Visitor[23], Others[24], **ПБКСШ** – Подход базиран на комбинация от Софтуерни шаблони

Последният ред в Таблица 1 показва комбинация от СШ, използвани в предложения от авторите подход за разработване на клас от системи, базиран на Комбинация от СШ, разгледани в раздел 3 от настоящия доклад.

Анализът на литературните източници [4, 5, 7, 9] позволява следните критерии за класификация на комбинациите от СШ:

- **Според релациите между СШ** [4] (Таблица 2):
  - A. Шаблон X използва Шаблон Y в решението си;
  - B. Шаблон X е подобен на Шаблон Y;
  - C. Шаблон X може да се комбинира с Шаблон Y.

- **Според сходството между СШ** [5,9]:
  - A. Factory Method, Abstract Factory, Builder;
  - B. Singleton, static class;
  - C. Abstract Factory, Singleton, Factory Method;
  - D. Prototype, Abstract Factory, Pool Object;
  - E. Miltition/ Registry of Singleton, Pool Singleton, Singleton and Pool Object;
  - F. Adapter, Façade.

- **В зависимост от припокриването между СШ** [7]:
  - А. Консервативна комбинация от СШ;
  - Б. Комбинация от припокриващи се СШ.

Консервативната комбинация поддържа разделяне на структурите на учавстващите СШ, така че те винаги да бъдат идентифицируеми в програмния код.

От друга страна, когато СШ се припокриват, елементите на тези шаблони се сливат в една структура. Не е възможно да се проследи кои елементи се припокриват поради частично или пълно сливане на елементите на СШ.

Таблица 2.

A	**x**	2	3	1	6	12	9	9	9	18	20	20	21	21	22
	y	1	2	8	11	11	3	16	18	19	3	19	3	10	5
B	**x**	1	1	6	7	2	16	23	4	4	5	22	14	-	-
	y	6	7	7	17	17	17	17	5	10	10	23	15	-	-
C	**x**	8	7	3	3	16	3	21	-	-	-	-	-	-	-
	y	13	3	16	18	18	10	23	-	-	-	-	-	-	-

Забележка: **A.** Шаблон X използва Шаблон Y в решението си;
B. Шаблон X е подобен на Шаблон Y;
C. Шаблон X може да бъде комбиниран с Шаблон Y.
Abstract factory[1], Flyweight[2], Composite[3], Proxy[4], Adapter[5], Prototype[6], Builder[7], Factory method[8], Interpreter[9], Decorator[10], Solitaire[11] , Observer[12], Template method[13], Glue[14], Mediator[15], Visitor[16], Strategy[17], Iterator[18], Memento[19], Command[20], Chain of responsibility[21], Bridge[22], State[23]

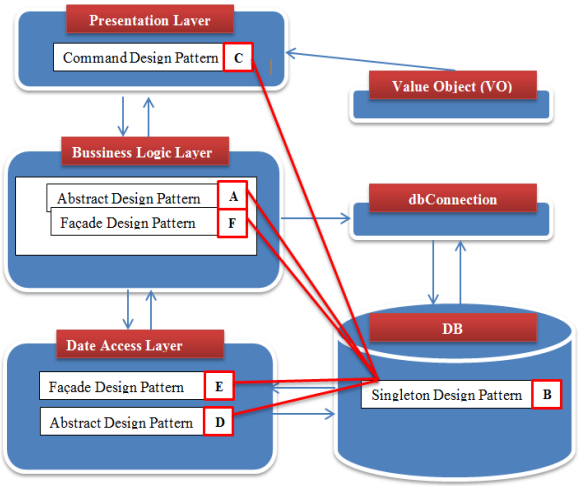
3. Подход за разработване на клас софтуерни системи, базиран на комбинация от СШ

Подходът е предназначен за ИС със следната обща функционалност:

- Достъп до Релационна база данни (РБД);
- Криптиране и декриптиране на данни;
- Изпращане на мейли;
- Работа с основни и спомагателни менюта;
- Въвеждане и модифициране на данни в РБД;
- Генериране на справки;

Предложеният подход (фиг.1) е комбинация от СШ от групата на GOF: Singleton, Abstract Factory, Façade и Command. Подходът включва 4 шаблона (таблица 1, последен ред ПБКСШ) за решаване на 6 типа задачи:

A. За работа с потребителски регистър, криптиране и декриптиране на данни - Abstract Factory Design Pattern;
B. За достъп до РБД - Singleton Design Pattern;
C. За работа с менюта - Command Design Pattern;
D. За въвеждане и модифициране на данни в РБД - Abstract Factory Design Pattern;
E. За генериране на отчети и изчисляване - Façade Design Pattern;
F. За изпращане на мейли - Façade Design Pattern



Фиг. 1. Общо описание на подход: софтуер, базиран на Комбинация от СШ

Следвайки този подход (фигура 1), бихме могли да разработим много други информационни системи.

4. Прилагане на предложения подход за създаване на ИС за целите на морския транспорт

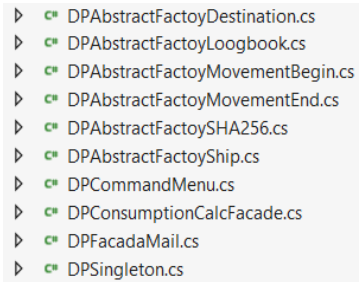
Използвайки вече разработения подход за създаване на софтуер, базиран на комбинация от СШ, е създадена ИС за целите на морския транспорт.

Техниките и инструментите, използвани за разработването на ИС за целите на морския транспорт, включват: РБД, реализирана чрез SQL Server, Visual Studio и C# език за програмиране за създаване на самото приложение.

ИС има два типа потребители – администратор и оператор. Операторът има няколко нива на достъп за преглед на съдържанието (пълни права), модифициране и редактиране на данни (ограничени права). Администраторът има пълни права за достъп до всеки модул от системата, както и възможност за изтриване на записи.

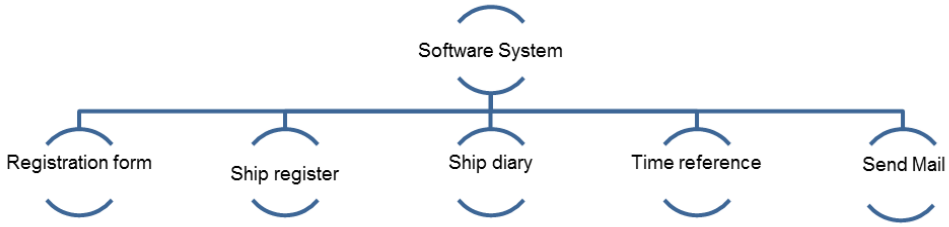
A. Фукционалност на ИС

Подходът, основаващ се на Комбинация от СШ, се прилага в разработената ИС. За тази цел е разработена комбинация от 10 СШ, базирана на СШ от групата на GOF (Singleton, Abstract Factory, Façade и Command):



Фиг. 2. Класове СШ

Основните секции на софтуерната система са представени на фигура 3.



Фиг. 3. Основни структурни елементи на Софтуерната система

• **Registration form** - Формулярът за регистрация дава възможност за: добавяне на нов потребител и промяна на забравена парола.

• **Ship register** - Корабният регистър позволява въвеждането на основни данни за: кораба, дестинация, начало и край на движението.

• **Ship diary** - Корабен дневник е модул, чрез който се изчислява потреблението на горива, масла и питейна вода.

• **Time reference** - Справки за период от време: за целите на приложението се генерират автоматично пет типа отчети. За целта първо се избира тип отчет и се задава период от време, след което се генерира отчет за:

- разход на горива, масла и вода (РГМВ) за 24 часов период (NOON);

- РГМВ от последния 24 часов период (т.е. от последния NOON) до пристигане на кораба в пристанището за маневра (End of Sea Passage);

- РГМВ за периода от отплаване на кораба до началото на плаване по море (Commence of Sea Passage);

- РГМВ за престоя на кораба в пристанището (CAST OF);

- РГМВ от началото на маневрата при пристигане на кораба в пристанище до заставане на кея (Finish with Engine).

Генерираният отчет е във формат .xmls. Отчетът NOON може да се види от таблица 3.

• **Send Mail** - Изпращане на мейли: меню, което предоставя ежедневни отчети в централния офис на корабната агенция.

Таблица 3. Справка NOON

m/v Tarifa						
		C/E NOON REPORT			**5/15/2018 12:00**	
M/E rpm	86.9	rpm	87.0	St. time	24.0	hrs
Brg. Dist	398.00	nm	Average	Eng. Dist.	435.8	nm
Slip	8.7%	P	8.6%	Av. Power		%
M/E speed	18.16	kts		M/E Power	10550.9	BHP
M/E Power	7868	kw	49.7%			
HSFO Consumption				**LSDO Consumption**		
M/E	40.4	m/t		M/E	0.00	m/t
D/G	4.60	m/t		D/G	0.00	m/t
Boiler	0.00	m/t		Boiler	0.00	m/t
Total	45.0	m/t		**Total**	0.00	m/t
LO Consumption		**Low BN**		**LSGO Consumption**		
M/E	20		Ltrs	M/E	0.00	m/t
M/E Cyl. Oil	170	0	Ltrs	D/G	0.00	m/t
D/G LO	12		Ltrs	Boiler	0.00	m/t
				Total	0.00	m/t
Fresh Water						
Produced	25	m/t				
Consumed	5	m/t				
		Low BN	**ROB**			
M/E LO	12935		ltrs.	HSFO	1705.3	m/t
M/E Sump TK	16600		ltrs.	LSDO	0.0	m/t
Cyl. Oil	16380	5630	ltrs.	LSGO	379.8	m/t
D/G LO	4627		ltrs.	TOTAL DO	379.8	m/t
FW	135	m/t				

Б. Клас диаграми и имплементиране

Таблица 4 представя както класова диаграма на всеки един от шестте типа задачи, използвани за внедряване на софтуер въз основа на комбинация от тези шаблони, така и тяхното изпълнение.

1. За работа с потребителски регистър криптиране и декриптиране на данни е разработен DP AbstractFactorySHA256, базиран на Abstract Factory Design Pattren (Шаблон **DPAFSH*** от Таблица 5).

2. За достъп до РБД е разработен шаблон DPSingleton, базиран на Singleton Design Pattrens (Шаблон **DPS*** от Таблица 5).

3. За работа с менюта е разработен шаблон DPCommandMenu, базиран на Command Design Pattrens (Шаблон **DPCM*** от Таблица 5).

4. За въвеждане и модифициране на данни в РБД е разработен шаблон DPAbstractFactoryShip, базиран на AbstractFactory Design Pattrens. (Шаблони **DPAFD, DPAFL, DPAFMB, DPAFME и DPAFS** от Таблица 5).

5. За генериране на отчети и изчисляване е разработен шаблон DPConsumptionCalcFacade, базиран на Facade Design Pattrens (Шаблон **DPCCF*** от Таблица 5).

6. За изпращане на мейли е разработен шаблон DPFacadaMail, базиран на Facade Design Pattrens (Шаблон **DPFM*** от Таблица 5).

Таблица 4 UML КЛАС ДИАГРАМИ И ИМПЛЕМЕНТИРАНЕ

UML CLASS DIAGRAMS

1.	2.	3.
		
4.	**5.**	**6.**
		

IMPLEMENTATION

1.

2.

3.

```
//LIST1
enum MenuTables{...}

interface IReciever{...}
abstract class Command{...}
class NOONCommand : Command{...}
class EOSPCommand : Command{...}
class COSPCommand : Command{...}
class CASTOFCommand : Command{...}
class FWECommand : Command{...}
class CommandCalcul : IReciever{...}
//LIST2
enum MenuReports{...}

interface IReciever2{...}
abstract class ACommand2{...}
class RegistrCommand : ACommand2{...}
class ShipCommand : ACommand2{...}
class DestCommand : ACommand2{...}
class GravCommand : ACommand2{...}
class MovBCommand : ACommand2{...}
class MovECommand : ACommand2{...}
class LogBCommand : ACommand2{...}
class ConsCommand : ACommand2{...}
class CommandCalcul2 : IReciever2{...}
```

4.

```
public interface IShip{...}
public interface ShipType{...}
public class ConcreteShipType : ShipType
{
    public IShip GetShip(bool b){...}
}
public class InsertShip : IShip
{
    int IDShip1 { get; set; }
    string IMONumber1 { get; set; }
    string VesselName1 { get; set; }
    double PitchImpeller1 { get; set; }
    int IShip.IDShip1 { get; set; }
    string IShip.IMONumber1 { get; set; }
    string IShip.VesselName1 { get; set; }
    double IShip.PitchImpeller1 { get; set; }
    void IShip.GetShip(){...}
}
public class UpdateShip : IShip
{
    int IDShip1 { get; set; }
    string IMONumber1 { get; set; }
    string VesselName1 { get; set; }
    double PitchImpeller1 { get; set; }
    int IShip.IDShip1 { get; set; }
    string IShip.IMONumber1 { get; set; }
    string IShip.VesselName1 { get; set; }
    double IShip.PitchImpeller1 { get; set; }
    void IShip.GetShip(){...}
}
```

5.

```
public class DPConsumptionCalcFacade
{
    ..........
    public double _TimeHRS1 { get; set; }
    public string MERevCntrBegin { get; set; }
    public string MERevCntrEnd { get; set; }
    public string ObsDist { get; set; }
    public class Consumption13 : DPConsumptionCalcFacade{...}
    public class Consumption13Facade : Consumption13{...}
    public class Consumption111: DPConsumptionCalcFacade{...}
    public class Consumption111Facade : Consumption111{...}
    public class Consumption112 : DPConsumptionCalcFacade{...}
    public class Consumption112Facade : Consumption112{...}
    AbstractDatabase db = new SqlServerDatabaseSingleton();
    public DPConsumptionCalcFacade InsertConsumption(){...}
    public DPConsumptionCalcFacade UpdateConsumption(){...}
}
```

6.

```
public class SmtpSettings
{
    public string Host1 { get; set; }
    public int Port1 { get; set; }
    public string Username1 { get; set; }
    public string Password1 { get; set; }
    public string From1 { get; set; }
    public string To1 { get; set; }
    public string Subject1 { get; set; }
    public string Body1 { get; set; }
    public SmtpSettings settings { get; set; }
    AbstractDatabase db = new SqlServerDatabaseSingleton();
    public class SmtpSettingsFacade : SmtpSettings
    {
        private readonly SmtpSettings smtpSettings;
        public SmtpSettingsFacade(){...}
        public void CreateSmtpSettingsFacade(){...}
        public SmtpSettings GetSmtpSettings(){...}
    }
}
```

5. Оценка на ПБКСШ

Оценката на всяка софтуерна разработка се извършва, като се измерват определени характеристики на обекти в кода ѝ. Резултатите носят информация за качеството на тези обекти, също така се правят прогнози за тенденции в процеса на разработване на софтуера, за които се твърди, че зависят от изследваните характеристики и могат да бъдат управлявани. [19] В подхода за изграждане на клас софтуерни системи чрез комбинация от СШ са използвани пет софтуерни метрики за оценка на кода: индекс на поддържаемост, цикломатична сложност, дълбочина на наследяване, свързване на класа и линии на кода. Оценката на тези пет показателя е направена преди и след прилагане на КСШ. Резултатите за двата случая са представени в Таблица 5.

Таблица 5. Софтуерни метрики

СШ	Код без СШ	Код със КСШ	индекс на поддър-жаемост		цикломатична сложност		дълбочина на наследяване		свързване на класа		линии на кода	
			без СШ	със СШ	без СШ	със СШ	без СШ	със СШ	без СШ	със СШ	без СШ	със СШ
1) DPAFD	Update() : void	UpdateDestination	61	81	1	10	-	1	9	11	11	25
	Insert() : void	InsertDestination	67	89	1	10	-	1	5	8	7	19
2) DPAFL	Update() : void	InsertLoogbook	55	89	1	62	-	1	11	9	14	83
	Insert() : void	UpdateLoogbook	62	89	1	62	-	1	7	12	7	88
3) DPAFMB	Update() : void	UpdateMovementBegin	61	81	1	10	-	1	9	11	11	25
	Insert() : void	InsertMovementBegin	67	89	1	10	-	1	5	8	7	19
4) DPAFME	Update() : void	UpdateMovementEnd	61	81	1	10	-	1	9	11	11	25
	Insert() : void	InsertMovementEnd	69	89	1	10	-	1	5	8	6	19
5) DPAFS	Update() : void	UpdateShip	60	88	1	18	-	1	9	11	11	33
	Insert() : void	InsertShip	65	89	1	18	-	1	5	8	7	28
6) DPAFSH*	CalculateHashedPassword(string) : string	ClientFactory SHA256	72	81	2	3	-	1	4	5	4	5
7) DPCM*	shipToolStripMenuItem_Click(object, EventArgs) : void	ShipCommand	85	85	1	2	-	2	3	2	2	4
	gravityToolStripMenuItem_Click(object, EventArgs) : void	GravCommand	85	85	1	2	-	2	3	2	2	4
	destinationToolStripMenuItem_Click(object, EventArgs) : void	DestCommand	85	85	1	2	-	2	3	2	2	4
	consumptionToolStripMenuItem_Click(object, EventArgs) : void	ConsCommand	85	85	1	2	-	2	3	2	2	4
8) DPCCF*	f_TotalConsHSFO_END() : void	f_TotalConsHSFO_END() : void	70	90	1	1	-	-	1	1	5	1
	f_TotalConsHSFO_ROB() : void	f_TotalConsHSFO_ROB() : void	70	90	1	1	-	-	1	1	5	1
	f_TotalConsLSDO_END() : void	f_TotalConsLSDO_END() : void	70	90	1	1	-	-	1	1	5	1
	f_TotalConsLSDO_ROB() : void	f_TotalConsLSDO_ROB() : void	70	90	1	1	-	-	1	1	5	1
	f_TotalConsLSGO_END() : void	f_TotalConsLSGO_END() : void	70	90	1	1	-	-	1	1	5	1
	f_TotalConsLSGO_ROB() : void	f_TotalConsLSGO_ROB() : void	70	90	1	1	-	-	1	1	5	1
9) DPFM*	smtp() : void	SmtpSettings.SmtpSettingsFacade	58	53	1	7	-	2	5	22	13	54
10) DPS*	sqlcon_() : void	AbstractDatabase	92	93	1	5	-	1	1	2	1	5

Изводите от прилагане на ПБКСШ в ИС за целите на Морския транспорт са: 1) Повишаване на експлоатационната характеристика на софтуера, а именно индекс на поддържаемостта. 2) Намаляване на времето за разработка на софтуера, поради възможността за преизползване на кода.

Не може да не се отбележи, че някои характеристики на софтуера като: цикломатична сложност, дълбочина на наследяване, свързване на класа и линии на кода се влошават, но те не касаят експлоатационните качества на софтуера.

6. Заключение и бъдещо развитие

В този доклад е представен подход за изграждане на клас софтуерни системи чрез комбинация от СШ. Използваните шаблони са от групата на GOF. СШ са полезен инструмент за решаване на повтарящи се проблеми в програмирането. Докладът обобщава ползите от прилагане на Комбинации от СШ при изграждане на софтуер, представя комбинации от Софтуерните шаблони за проектиране, класифицира комбинациите от СШ и предлага подход базиран на Комбинации от СШ. Предложения подход е приложен за реализиране на ИС за целите на морския транспорт. ПБКСШ е оценен чрез измерване на софтуерните метрики: индекс на поддържаемост, цикломатична сложност, дълбочина на наследяване, свързване на класа, линии на кода [17, 18], както и възможност за преизползване. Направените изводи са важни, тъй като показват подбряване на експлоатационните качества на софтуера. Нашето изследване продължава в посока на изследване на качествени показатели, които следят: управляемост, производителност и ефективност.

Литература

[1]. https://vinodnambiar.wordpress.com/2011/09/29/service-composition
[2]. http://blog.e-zest.com/design-pattern-combination-strategy-with-factory-method
[3]. http://www2.fiit.stuba.sk/~filkorn/docs/EEICT2004.pdf
[4]. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.37.8779&rep=rep1&type=pdf
[5]. http://andrey.moveax.ru/design-patterns/oop/
[6]. https://www.codeproject.com/Articles/13229/Implementing-Observer-Strategy-and-Decorator-Design
[7]. Design_Patterns_-_education_and_classification_cha.pdf
[8]. Erich Gamma et al, Design Patterns: Elements of Reusable Object-Oriented Software, ISBN: 0201633612, Addison-Wesley Publ. Co., January 15, 1995
[9].https://www.researchgate.net/publication/313578261_Evaluation_of_Software_Design_Pattern_on_Mobile_Application_Based_Service_Development_Related_to_the_Value_of_Maintainability_and_Modularity
[10]. https://www.cs.rug.nl/paris/papers/EPLOP11.pdf
[11]. http://serviceorientation.com/soaglossary/compound_design_pattern
[12].https://www.duo.uio.no/bitstream/handle/10852/61661/theepank_main_document.pdf?sequence=32&isAllowed=y, 2018
[13]. https://pdfs.semanticscholar.org/1833/f5927952dbeaad298306228c8e8c5cdc1c9c.pdf
[14]. http://www.righthandtech.com/controller-development.php
[15]. https://enterpriseprogrammer.com/2012/07/04/adapter-factory-design-pattern/
[16]. http://homepages.dcc.ufmg.br/~figueiredo/disciplinas/papers/jss2014cacho.pdf
[17].https://www.duo.uio.no/bitstream/handle/10852/60024/FinalThesis_Farooq.pdf?sequence=1&isAllowed=y, 2017
[18]. https://dailydotnettips.com/understand-the-complexity-and-maintainability-of-your-coding-using-code-metrics-in-visual-studio/
[19]. http://store1.data.bg/bobomir/NBU/INF520_ST/SoftwareEBook_.pdf

За контакти:
Димитричка Ж. Николаева
катедра „Софтуерни и интернет технологии”
Технически университет - Варна
E-mail: dima.nikolaeva@abv.bg
Доц. д-р Виолета Т. Божикова
катедра „Софтуерни и интернет технологии”
Технически университет - Варна
E-mail: vbojikova2000@yahoo.com

СИСТЕМА ЗА КРИПТИРАНЕ НА ФАЙЛОВЕ

Йоан С. Иванов

Резюме: Представената система дава възможност на потребителя за криптиране и декриптиране на (произволен) файл. В програмата са включени два алгоритъма - симетричен AES и асиметричен RSA 4096 (максимална големина на файла - 448 бита); ограничена е дължината на криптиращия ключ, което се обуславя и от асимтричността на алгоритъма; генериране на хеш на зададен файл – три метода – SHA-1,MD5,SHA-256. Възможност за генериране на случайни целочислени числа в зададен от потребителя интервал
Ключови думи: криптография, RSA, AES, хеш функция,SHA-1, MD5, SHA-256, случайни числа, асиметричен кр. алгоритъм, private key, public key, симетричен кр. алгоритъм

File encryption system

Yoan S. Ivanov

Abstract: This application was created as a part of a bigger project, regarding the Blockchain technology. The software can be used for file encryption/decryption, and also as a checksum generator. There is an option for generating pseudo random integer numbers in a given interval (specified by the user) with the help of the System.Security.Cryptography of the .NET Framework
Keywords: C#, .NET Framework, WinForms, library System.Security.Cryptography (in-built in C#) [4].

1. Увод (Въведение)

Това приложение е създадено като част от по-голям проект, замислено да изпълнява криптографската функция от една по-мащабна система, касаеща Bitcoin и криптовалутите и пренасянето им по мрежата.

Основни задачи, които решава приложението

Системата е реализирана като самостоятелно работеща програма, позволяваща на потребителя да криптира даден файл (като го защити с парола), след което същият да може да бъде разкриптиран отново чрез паролата въведена по-рано (AES 256 - Rijndael); софтуерът предоставя възможност на потребителя да генерира хеш сума на произволен файл (използва се главно за осигуряване сигурността на файла) и като генератор на случайни цели числа в зададен интервал.

По-важни понятия в криптографията

Криптографията днес се разглежда като подраздел на една по-голяма дисциплина – криптологията, която включва в себе си и криптоанализа, който се занимава с дешифрирането на криптирана информация **без** да е налице ключът, т.е. осъществяване на „криптографическа атака'' върху дадения шифър. Тя е сравнително нова дисциплина – развива се усилено през XX век, особено по време на Втората световна война, с навлизането на шифровъчните машини, по-известната от които Енигма, чийто код бива дешифриран от А. Тюринг

Модуларната (модулна) аритметика също намира широко приложение в криптографските (асиметрични) алгоритми, като разглежда числовите сравнения от вида $a \equiv b (\mathrm{mod}\, n)$ в множеството $\mathbf{Z}$ на целите числа. Съществуват главно два вида шифри:

Потокови шифри (stream ciphers) – симетричен ключ-шифър, чиито цифри, представени в чист вид, се комбинират с псевдослучаен шифърен числов поток (keystream).

Особеното за този вид шифри е, че цифрите се криптират една по една – всяка се сумира по модул 2 (операция XOR) със съответната ѝ случайна от цифровия поток и така се формира всяка цифра на потоковия шифър [5].

Блокови шифри (block ciphers - ECB, CBC, OFB, GCM) – използват детерминирани алгоритми, т.е. при един и същ подаден вход връщат един и същ резултат, преминавайки през една и съща поредица от състояния – поведението на алгоритъма е предсказуемо; алгоритъмът се прилага върху фиксирана група битове (block) чрез постоянна, неизменяща се трансформация, определяна от симетричния ключ. Блоковите шифри могат да се определят еднозначно със следната формула:

$$E_k(P) \coloneqq E(K,P) : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n \quad (1)$$

Приема се един ключ K с дължина k бита (key size) и символен низ от битове P с дължина n (block size); резултатът от функцията е изходният шифър C, състоящ се от n бита. P е текстът в чист вид (plain text). За всяко K функцията $E_k(P)$ е необратима, т.е. декриптирането чрез криптоанализ е практически невъзможно. Обратният процес (E^{-1}) се извършва със следната функция:

$$E_k^{-1}(C) \coloneqq D_k(C) = D(K,C) : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n \quad (2)$$

Подават се ключ K и криптираният шифър C, за да се върне разкриптирания текст P, така че за $\forall K : D_k(E_k(P)) = P$ [3].

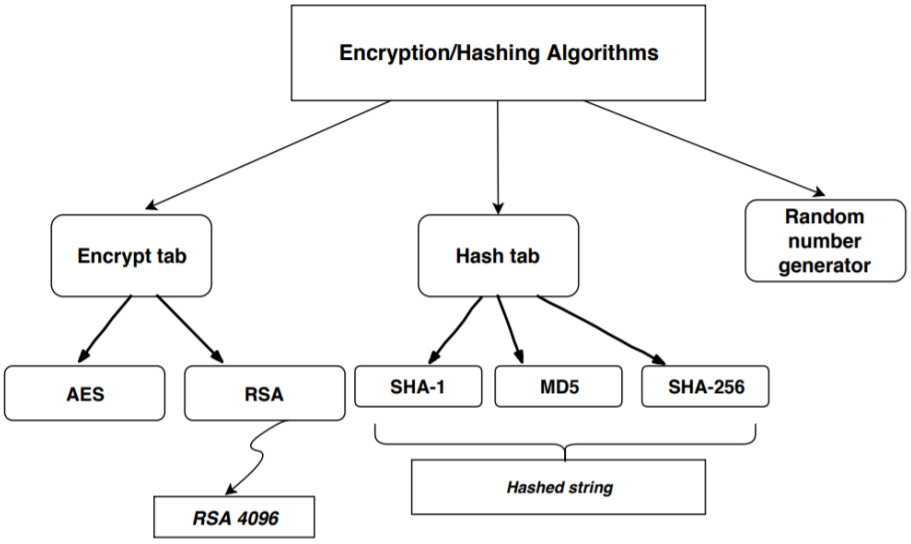
Криптографията, базирана на елиптични криви (ECC), също е много популярен метод, използван в Blockchain/Etherium, но не е quantum-safe!, т.е, възможно е дешифриране от суперкомпютър.

В приложението, както ще стане ясно и по-нататък, са имплементирани два алгоритъма – AES(Rjndael) и RSA [2]. Хеш функциите също намират много приложения, най-вече когато самостоятено се използват като начин да се провери целостта на даден файл посредством контролна хеш сума (checksum).

2. Интерфейс и програмна логика на приложението

2.1. Интерфейс на приложението

Диаграмата на организацията на интерфейса и логиката на системата е показана на фигура 1.

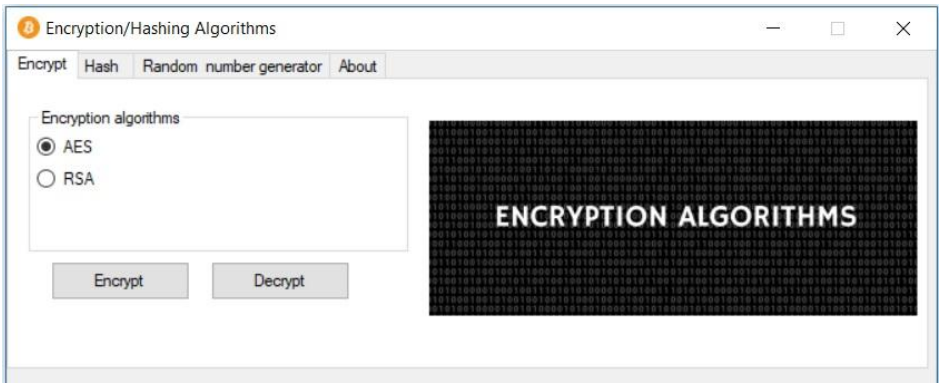


Фиг. 1. Диаграма на организацията на интерфейса и логиката на системата

2.2. Описание

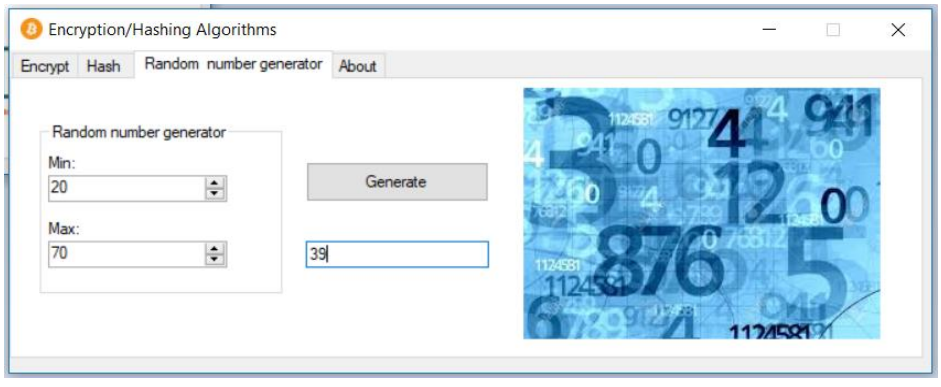
Интерфейсът е разработен с WinForms библиотеката на .NET Framework (C#) и със средата Visual Studio (2017) (фигура 2). Програмата се състои от три раздела:

- ***Encrypt*** (фигура 3) - разделът за криптиране/декриптиране на файл съдържа два радиобутона - съответно за AES и RSA криптиране. За RSA алгоритъма се отваря нов подпрозорец (подпрограма) от главната програма, който съдържа два бутона – съответно Encrypt и Decrypt.
- ***Hash*** – таб за генериране на хеш сума на файл – състои се от три радиобутона – SHA-256, MD5, RSA, както и от два бутона ***Open file*** и ***Hash***, с който се осъществява хеширането; резултатът се изписва в текстовото поле *Hashed string.*
- ***Random number generator*** (фигура 4) – състои се от две полета от тип *NumericUpDown* – в първото поле се въвежда долната граница на интервала (минимално число), а във второто – горната (максимално число); генерират се цели числа.
- ***About*** **–** раздел с информация за програмата.



Фиг. 2. Заглавна част на програмата

Фиг. 3. Hash таб



Фиг. 4. Random number generator

2.2. Програмна логика на приложението

Главният прозорец на системата е показан на фигура 1.

2.2.1. Имплементация на AES криптирането – Осъществява се посредством два метода:
`void FileEncrypt(string inputFile, string password)` – функция за криптиране на файла – приема като параметри пътя до файла, който ще се криптира, от текстов тип (string) и паролата, която ще се изисква по време на етапите на криптирането му.

2.2.2. Съдържание на основните части на тялото на метода `FileEncrypt(args…)`

- `byte[] salt = GenerateRandomSalt();` - генерира random сол;
- `byte[] passwordBytes = System.Text.Encoding.UTF8.GetBytes(password)` – конвертиране на въведената от потребителя парола в byte array;
- `System.Text.Encoding.UTF8.GetBytes(password)` – конвертиране на въведената от потребителя парола в byte array;
- ```
  RijndaelManaged AES = new RijndaelManaged();
      AES.KeySize = 256;
      AES.BlockSize = 128;
    AES.Padding = PaddingMode.PKCS7; -
  ```
  инициализация на алгоритъма (Rijndael – версия на алгоритъма AES, одобрен и приет като основен от Националния институт за стандарти и технологии в САЩ, NIST); инициализира се дължината на ключа в битове – 256 и padding – PKCS7;

- ```
  var key = new Rfc2898DeriveBytes(passwordBytes, salt, 50000);
    AES.Key = key.GetBytes(AES.KeySize / 8);
    AES.IV = key.GetBytes(AES.BlockSize / 8);
  ```
  `AES.Mode = CipherMode.CFB;` - този блок от код неколкократно хешира паролата,въведена от потребителя заедно със солта; IV – инициализационен вектор; CFB (Cipher Feedback). Инициализационният вектор се получава така (CFB Decryption – декриптиране на шифъра):
  $C_i = E_K(C_{i-1}) \oplus P_i$
  $P_i = E_K(C_{i-1}) \oplus C_i$
  $C_0 = \text{IV}$

- `void FileDecrypt(string inputFile, string outputFile, string password)` - метод за декриптиране на файла – приема като параметър 3 променливи от тип string

  - `string inputFile` – пътят до криптирания файл (във вида file.ext.aes);
  - `string outputFile` – пътят, където да се създаде декриптираният файл - използвана е функцията filename = file.Replace(".aes",".decr"), т.е. създава се в същата директория, където е криптираният файл, като се заменя само разширението в края на името;
  - `string password` – паролата за декриптиране на файла - трябва да съвпада с въведената парола при криптирането.

Параметрите при разкриптирането трябва да бъдат същите, които са използвани за криптирането на файла – като KeySize, BlockSize, Padding, Mode (блок шифъра) и пр.

Имплементация на RSA (4096) криптирането - могат да се криптират файлове с максимален размер 448 бита поради факта, че RSA е асиметричен алгоритъм и максималният брой битове в ключа, използван от този шифър, е 4096 (минимумът е 1024)

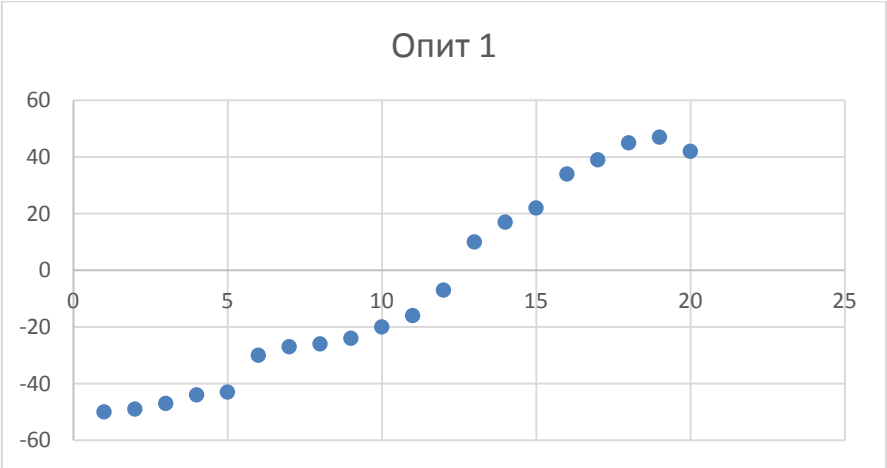
3. Изводи и тестване на приложението

Криптирането на файлове е от изключително значение при пренасянето им по мрежата, тъй като това гарантира тяхната конфидециалност; единият от потребителите криптира даден файл (произволен) чрез парола, след това този файл се изпраща по мрежата и бива получен от втория потребител. Чрез същата парола той разкриптира въпросния файл и, условно казано, „прочита'' съдържанието му.

Резултатите от опитите са показани в таблица 1 – всеки опит е обособен в отделна колона, като в нея е записан и поредният номер на генерираното случайно цяло число. Направени са три опита - всеки с по 20 случайни числа, както и графика (точкова диаграма) на тяхното разпределение.

Таблица 1. Данни от опити

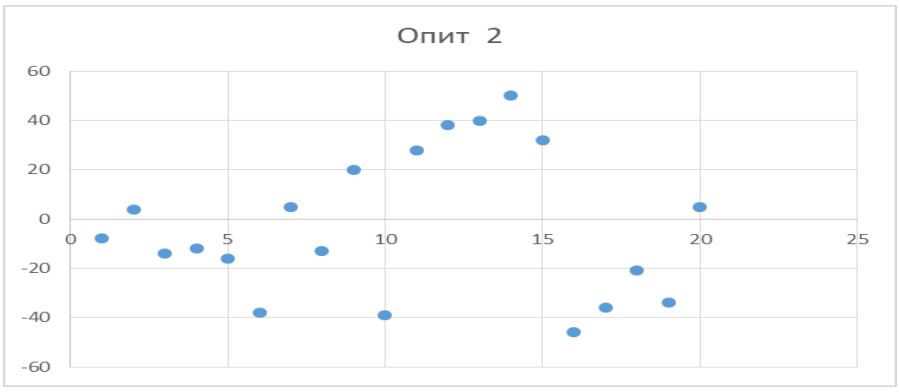
Опит 1		**Опит 2**		**Опит 3**	
№	Поредно генерирано число	№	Поредно генерирано число	№	Поредно генерирано число
1	-50	1	-8	1	13
2	-49	2	4	2	-7
3	-47	3	-14	3	9
4	-44	4	-12	4	36
5	-43	5	-16	5	-41
6	-30	6	-38	6	10
7	-27	7	5	7	-10
8	-26	8	-13	8	42
9	-24	9	20	9	11
10	-20	10	-39	10	-14
11	-16	11	28	11	22
12	-7	12	38	12	-45
13	10	13	40	13	-3
14	17	14	50	14	50
15	22	15	32	15	46
16	34	16	-46	16	-30
17	39	17	-36	17	-45
18	45	18	-21	18	-29
19	47	19	-34	19	12
20	42	20	5	20	-34



Фиг. 5. Разпределение на числата от опит 1

Ясно се вижда от графиката, че разпределението на числата при провеждане на опит 1 (фигура 5) е в нарастващ (възходящ) ред, т.е. получава се крива, която достига екстремална стойност в даден момент, а получената функция е строго растяща, с изключение на

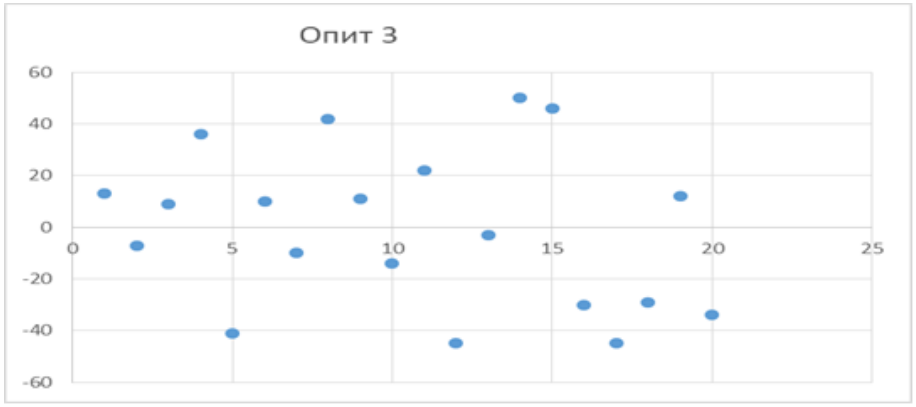
последната стойност (последното генерирано число), което е със стойност по-малка от екстремалната (47<42).



Фиг. 6. Разпределение на числата от опит 2

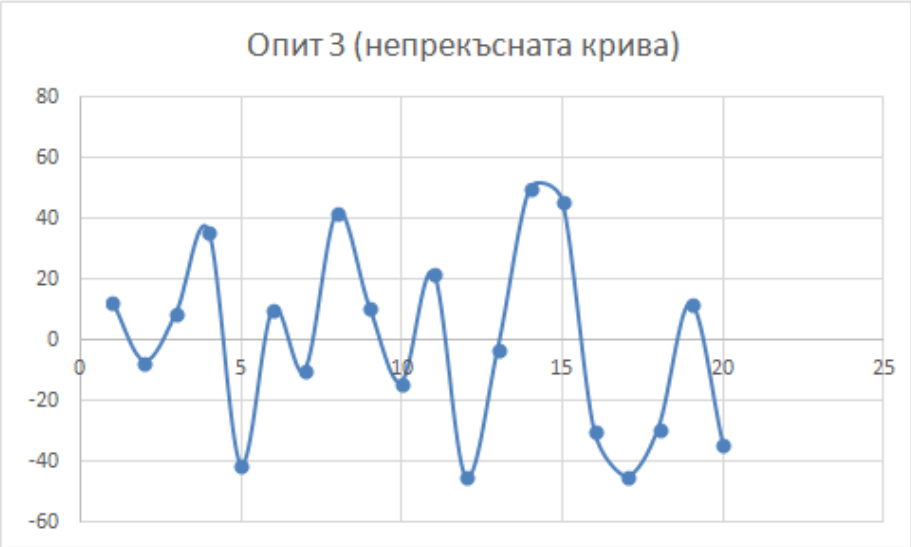
При опит 2 (фигура 6) се забелязва, че разпределението на точките е хаотично и трудно би могло да се опише единствена крива посредством съединяването им. Разликата в графиката на опит 1 и на опит 2 се дължи на факта, че генераторът продуцира числата с голяма разлика между абсолютните им стойности – интервалът, определен от двойка генерирани едно след друго числа, е с голяма дължина.

При опит 3 (фигура 7 и фигура 8) ситуацията е аналогична, както в опит 2 - тук обаче, е интересно да се отбележи, че разпределението на точките е сравнително симетрично спрямо абсцисната ос на координатната система. В случая я игнорираме и приемаме, че е зададена само оординатната ос, по продължението на която са нанесени стойностите на генерираните числа. Абсцисната ос не е значеща, но, ако си я представим като мислена линия, разделяща графиката на две половини, веднага ще можем да забележим симетричността, която се получава, когато я прекараме през началото на координатната система. Отново се наблюдава относително хаотично разпределение на точките, макар те да не са разположени по начин, предполагащ образуването на крива, описваща само строго растяща или намаляваща функция. Графиката може да бъде разглеждана и като крива, описваща функция, която има два абсолютни и множество локални екстремуми – тя расте и намалява последователно.



Фиг. 7. Разпределение на числата от опит 3

Важно е да се отбележи, че за реализирането на този генератор е използван вграденият в .NET Framework клас Random, който използва предсказуем алгоритъм за произвеждането на цели числа (PRNG) или още наричан deterministic random bit generator – алгоритъм за генериране на поредица от числа, чиито свойства приближават свойствата на една дадена поредица от цели числа. PRNG – генерираната поредица не се състои реално от случайни числа, защото е напълно определена от начална стойност (initial value), наречена seed, която може да съдържа реални random стойности.



Фиг. 8. Непрекъната крива на разпределението на числата от опит 3

Като идея за по-нататъшно развитие на проекта, е обмислено да се добави модул към програмата, който да осигурява автоматичното пренасяне на файловете между потребителите, като това може да стане или посредством въвеждане на email, до който трябва да се изпратят файла/файловете, или чрез друга форма за комуникация – най-често това може да е социална мрежа или skype.

Съществуват и различни варианти (методи) за пренасянето на информацията през Интернет, един от които е blockchain технологията, която доби голяма популярност покрай навлизането на криптовалутите (bitcoins). Авторът реши, че за в бъдеще ще може развие проекта в тази насока, като системата се разшири и се създаде една по-мащабна програма, предлагаща различни варианти за пренасянето на информация и даваща възможност за избор на множество файлове, които потребителят да селектира от своя компютър, след което да се осъществи предаване на данните, посредством предложената система.

Литература

[1]. https://en.wikipedia.org/wiki/Advanced_Encryption_Standard, Уикипедия (англоезична)
[2]. https://en.wikipedia.org/wiki/RSA_(cryptosystem)
[3]. https://en.wikipedia.org/wiki/Block_cipher
[4].https://docs. microsoft.com/en-us/dotnet/api/system.security.cryptography?view=netframework-4.7.2, Сайт на Micrsoft с официалната документация за .NET Framework
[5]. https://en.wikipedia.org/wiki/Stream_cipher

За контакти:
Йоан С. Иванов - студент
специалност „СИТ”
Технически университет-Варна
E-mail: yoanivanovit@gmail.com

ИЗИСКВАНИЯ ЗА ОФОРМЯНЕ НА СТАТИИТЕ ЗА СПИСАНИЕ "КОМПЮТЪРНИ НАУКИ И ТЕХНОЛОГИИ”

I. Статиите се представят разпечатани в два екземпляра (оригинал и копие) в размер до 6 страници, формат А4 на адрес: Технически университет – Варна, ФИТА, ул. „Студентска” 1, 9010 Варна, както и в електронен вид на имейл адреси: ned.nikolov@tu-varna.bg или yulka.petkova@tu-varna.bg.

II. Текстът на статията трябва да включва: УВОД (поставяне на задачата), ИЗЛОЖЕНИЕ (изпълнение на задачата), ЗАКЛЮЧЕНИЕ (получени резултати), БЛАГОДАРНОСТИ към сътрудниците, които не са съавтори на ръкописа (ако има такива), ЛИТЕРАТУРА и информация за контакти, включваща: научно звание и степен, име, организация, поделение (катедра), e-mail адрес.

III. Всички математически формули трябва да са написани ясно и четливо (препоръчва се използуване на Microsoft Equation).

IV. Текстът трябва да бъде въведен във файл във формат WinWord 2000/2003 с шрифт Times New Roman. Форматирането трябва да бъде както следва:

1. Размер на листа - А4, полета: ляво - 20мм, дясно - 20мм, горно - 15мм, долно - 35мм, Header 12.5мм, Footer 12.5мм (1.25см).
2. Заглавие на български език - размер на шрифта 16, удебелен, главни букви.
3. Един празен ред - размер на шрифта 14, нормален.
4. Имена на авторите - име, инициали на презиме, фамилия, без звания и научни степени - размер на шрифта 14, нормален.
5. Два празни реда - размер на шрифта 14, нормален.
6. Резюме и ключови думи на български език, до 8 реда - размер на шрифта 11, нормален.
7. Заглавие на английски език - размер на шрифта 12, удебелен.
8. Един празен ред - размер на шрифта 11, нормален.
9. Имена на авторите на английски език - размер на шрифта 11, нормален.
10. Един празен ред - размер на шрифта 11, нормален.
11. Резюме и ключови думи на английски език, до 8 реда - размер на шрифта 11, нормален.
12. Основните раздели на статията (Увод, Изложение, Заключение, Благодарности, Литература) се форматират в едноколонен текст както следва:
    a. Наименование на раздел или на подраздел - размер на шрифта 12, удебелен, центриран, един празен ред преди наименованието и един празен ред след него - размер на шрифта 12, нормален;
    b. Текст - размер на шрифта 12, нормален, отстъп на първи ред на параграф – 10 мм; разстояние от параграф до съседните (Before и After) за целия текст – 0.
    c. Цитиране на литературен източник - номер на източника от списъка в квадратни скоби;
    d. Текстът на формулите се позиционира в средата на реда. Номерация на формулите - дясно подравнена, в кръгли скоби.
    e. Фигури - центрирани, разположение спрямо текста: “Layout: In line with text”. Номер и наименование на фигурата - размер на шрифта 11, нормален, центриран. Отстояние от съседните параграфи – 6 pt.
    f. Литература – всеки литературен източник се представя с: номер в квадратни скоби и точка, списък на авторите (първият автор започва с фамилия, останалите – с име), заглавие, издателство, град, година на издаване, страници.
    g. За контакти: научно звание и степен, име, презиме (инициали), фамилия, организация, поделение (катедра), e-mail адрес, с шрифт 11, дясно подравнено.

Образец за форматиране можете да изтеглите от адрес http://cs.tu-varna.bg/ - Списание КНТ, Spisanie_Obrazec.zip.

Този брой от списанието се отпечатва със средства от научно-изследователски проект, финансиран целево от държавния бюджет.