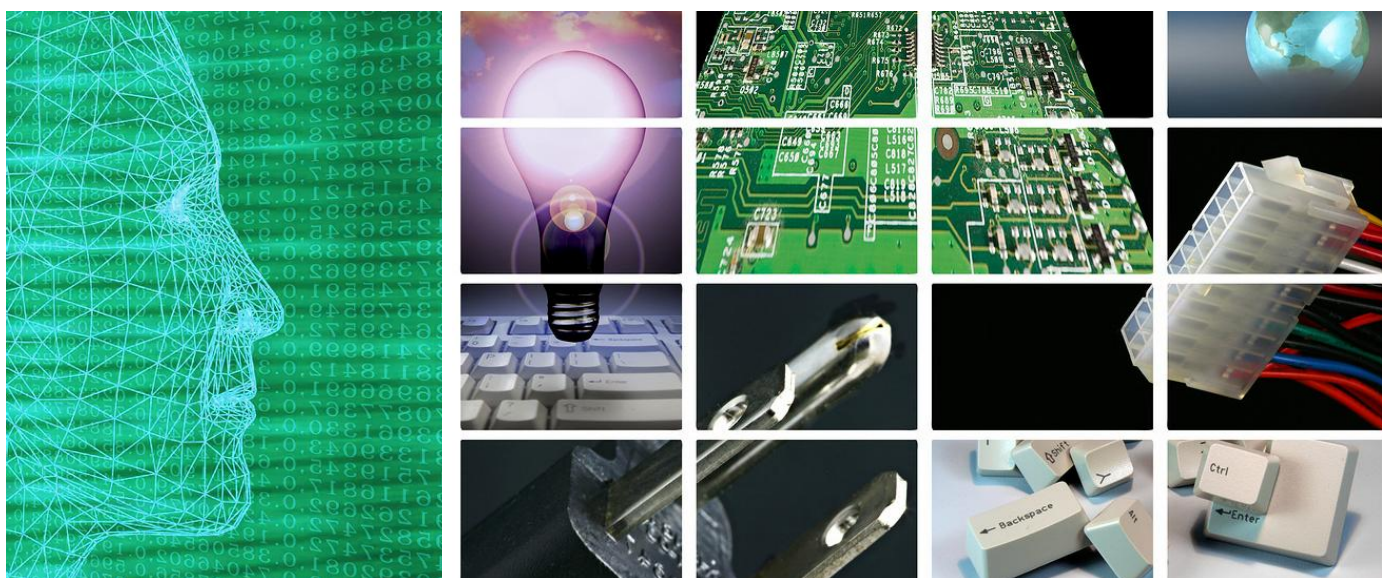




ТЕХНИЧЕСКИ УНИВЕРСИТЕТ - ВАРНА
TECHNICAL UNIVERSITY OF VARNA

Година XV, Брой 1/2017

КОМПЮТЪРНИ НАУКИ И ТЕХНОЛОГИИ



Bulgaria Communications Chapter

Faculty of Computing & Automation

COMPUTER SCIENCE AND TECHNOLOGIES

Year XV, Number 1/2017

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)
доц. д-р П. Антонов (Варна)
доц. д-р Р. Димова (Варна)
доц. д-р Х. Вълчанов (Варна)
доц. д-р В. Божикова (Варна)
доц. д-р Н. Николов (Варна)

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

Advisory Board:

Prof. L. Lichev, DSc (Ostrava)
Prof. M. Iliev, DSc (Ruse)
Prof. M. Lazarova, PhD (Sofia)
Assoc. Prof. P. Antonov, PhD (Varna)
Assoc. Prof. R. Dimova, Varna (Varna)
Assoc. Prof. H. Valchanov, PhD (Varna)
Assoc. Prof. V. Bozhikova, PhD (Varna)
Assoc. Prof. N. Nikolov, PhD (Varna)

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

СЪДЪРЖАНИЕ

CONTENTS

	КОМПЮТЪРНИ СИСТЕМИ И МРЕЖИ		COMPUTER SYSTEMS AND NETWORKS	
1.	АНАЛИЗ И СРАВНЕНИЕ НА МЕЖДУНАРОДНИТЕ СТАНДАРТИ ЗА ДЕЙТА ЦЕНТРОВЕ <i>Росен С. Радков</i>	8	ANALYSIS AND COMPARISON OF INTERNATIONAL STANDARDS FOR DATA CENTER INFRASTRUCTURE <i>Rosen S. Radkov</i>	1.
2.	МЕТОДИКА ЗА ОСИГУРЯВАНЕ НА ВИСОКА НАДЕЖДНОСТ НА ДЕЙТА ЦЕНТЪР <i>Росен С. Радков</i>	17	HIGH RELIABILITY DATA CENTER METHODOLOGY <i>Rosen S. Radkov</i>	2.
3.	БАЛАНСИРАНО РАЗПРЕДЕЛЕНИЕ НА ТРАФИКА В ИНТЕГРИРАНА ETHERNET СРЕДА (Част първа) <i>Деян Х. Стоев, Явор З. Дечев</i>	24	BALANCED TRAFFIC DISTRIBUTION IN AN INTEGRATED ETHERNET ENVIRONMENT (Part I) <i>Dean H. Stoeff, Yavor Z. Dechev</i>	3.
4.	БАЛАНСИРАНО РАЗПРЕДЕЛЕНИЕ НА ТРАФИКА В ИНТЕГРИРАНА ETHERNET СРЕДА (Част втора) <i>Деян Х. Стоев, Явор З. Дечев</i>	31	BALANCED TRAFFIC DISTRIBUTION IN AN INTEGRATED ETHERNET ENVIRONMENT (Part II) <i>Dean H. Stoeff, Yavor Z. Dechev</i>	4.
5.	ЕДИН ВАРИАНТ ЗА ИНИЦИАЛИЗАЦИЯ НА МАРШРУТИЗАТОР ЗА MPP КОМПЮТРИ С DLH МРЕЖОВА ТОПОЛОГИЯ <i>Милен Г. Ангелов</i>	40	A VARIANT FOR INITIALIZATION OF A ROUTER FOR MPP COMPUTERS WITH DLH NETWORK TOPOLOGY <i>Milen G. Angelov</i>	5.
6.	ИЗСЛЕДВАНЕ НА 4G КЛЕТЪЧНИ БЕЗЖИЧНИ МРЕЖИ <i>Айдън М. Хъкъ</i>	46	STUDY OF 4G CELLULAR WIRELESS NETWORKS <i>Aydan M. Haka</i>	6.
7.	ПРОУЧВАНЕ НА ИНТЕРФЕЙСИ ЗА УПРАВЛЕНИЕ НА НОСИМИ УСТРОЙСТВА <i>Юри Н. Димитров</i>	51	RESEARCH OF COMPUTER- HUMAN INTERFACES FOR WEARABLE DEVICES <i>Yuri N. Dimitrov</i>	7.

8.	ТЕХНОЛОГИЯ Li-Fi - ПЕРСПЕКТИВИ <i>Диян Ж. Динев</i>	59	Li-Fi TECHNOLOGY – TRENDS <i>Diyan Zh. Dinev</i>	8.
9.	КОМУНИКАЦИОНЕН МОБИЛЕН МОДУЛ ЗА СЪБИРАНЕ НА МЕТЕОРОЛОГИЧНИ ДАННИ <i>Юлиян Г. Димитров, Тодорка Н. Георгиева</i>	64	WIRELESS WEATHER STATION <i>Yuliyana G. Dimitrov, Todorka N. Georgieva</i>	9.
10.	СИСТЕМИ ЗА ОТКРИВАНЕ И ПРОТИВОДЕЙСТВИЕ НА БЕЗПИЛОТНИ ЛЕТАТЕЛНИ СИСТЕМИ ОТ ТИПОВЕ „МИНИ“ И „МИКРО“ <i>Васил Р. Люцканов</i>	70	COUNTERACTION SYSTEMS OF UNMANNED AIRCRAFT VEHICLES <i>Vasil R. Lyutskanov</i>	10.

	СОФТУЕРНИ И ИНТЕРНЕТ ТЕХНОЛОГИИ		SOFTWARE AND INTERNET TECHNOLOGIES	
1.	СИСТЕМА И МЕТОД ЗА ДИНАМИЧНО МАПВАНЕ НА ВЕРСИИ НА ИНФОРМАЦИЯ В СИСТЕМА ЗА СЪЗДАВАНЕ НА УЕБСАЙТОВЕ <i>Димитър З. Димитров, Елена В. Рачева</i>	77	A SYSTEM AND A METHOD FOR DYNAMIC DATA VERSION MAPPING IN A WEBSITE BUILDING SYSTEM <i>Dimitar Z. Dimitrov, Elena V. Racheva</i>	1.
2.	ОПТИМИЗИРАН ВАРИАНТ НА АЛГОРИТЪМА “IMAGE QUILTING” ЗА СИНТЕЗ НА ТЕКСТУРНИ ИЗОБРАЖЕНИЯ С ПРИЛОЖЕНИЕ В СИСТЕМИТЕ С ВИРТУАЛНА РЕАЛНОСТ <i>Даниела Д. Илиева</i>	83	OPTIMIZED VARIANT OF IMAGE QUILTING ALGORITHM FOR SYNTHESIS OF TEXTILE IMAGES WITH IMPLEMENTATION IN VIRTUAL REALITY SYSTEMS <i>Daniela D. Ilieva</i>	2.
3.	СИМУЛИРАНЕ НА ВОДНА ПОВЪРХНОСТ В СИСТЕМИ С ВИРТУАЛНА РЕАЛНОСТ <i>Даниела Д. Илиева</i>	90	WATER SURFACE SIMULATION IN SYSTEMS WITH VIRTUAL REALITY <i>Daniela D. Ilieva</i>	3.

4.	СРАВНИТЕЛЕН АНАЛИЗ МЕЖДУ НАИВНИЯ БЕЙСОВ КЛАСИФИКАТОР И МЕТОДА НА ОПОРНИТЕ ВЕКТОРИ ИЗПОЛЗВАЩ ОПТИМИЗАЦИЯ ПРИ КЛАСИФИКАЦИЯ НА БЪЛГАРСКИ ТЕКСТ В МАШИННОТО ОБУЧЕНИЕ <i>Нели Ан. Арабаджиева – Калчева, Недялко Н. Николов</i>	97	COMPARATIVE ANALYSIS OF THE NAIVE BAYES CLASSIFIER AND SEQUENTIAL MINIMAL OPTIMIZATION FOR CLASSIFYING TEXT IN BULGARIAN IN MACHINE LEARNING <i>Neli An. Arabadzieva – Kalcheva, Nedyalko N. Nikolov</i>	4.
5.	РАЗРАБОТВАНЕ НА ПРОГРАМЕН МОДЕЛ НА СИМУЛАТОР ЗА ОБУЧЕНИЕ НА РЪКОВОДИТЕЛИ НА ПОЛЕТИ <i>Милена Н. Карова, Анна Д. Николова, Друмка Н. Жекова</i>	106	DEVELOPMENT OF AIR TRAFFIC CONTROLLER TRAINING PROGRAMMING MODEL <i>Milena N. Karova, Anna D. Nikolova, Drumka N. Zhekova</i>	5.
6.	ДИНАМИЧНИ ВЪЗМОЖНОСТИ НА СОФТУЕРНА СИСТЕМА ЗА ПРОВЕРКА НА ЛИМИТНИ ОГРАНИЧЕНИЯ <i>Венцислав Николов</i>	114	DYNAMIC CAPABILITIES OF A COMPLIANCE CHECK SOFTWARE SYSTEM <i>Ventsislav Nikolov</i>	6.
7.	ПРЕГЛЕД НА СИСТЕМА ЗА АВТОМАТИЧНА ТЪРГОВИЯ <i>Венцислав Николов</i>	121	AN OVERVIEW OF AN AUTOMATED TRADING SOFTWARE SYSTEM <i>Ventsislav Nikolov</i>	7.

	АВТОМАТИЗИРАНИ СИСТЕМИ ЗА УПРАВЛЕНИЕ		AUTOMATED CONTROL SYSTEMS	
1.	ПРИЛОЖЕНИЕ НА РЕКУРСИВНИ МЕТОДИ ЗА ОЦЕНЯВАНЕ НА ПАРАМЕТРИ В СИСТЕМА ЗА АДАПТИВНО УПРАВЛЕНИЕ НА ПОСТОЯННО ТОКОВ ДВИГАТЕЛ ПОСРЕДСТВОМ САМОНАСТРОЙВАЩ СЕ РЕГУЛАТОР С МИНИМАЛНА ДИСПЕРСИЯ <i>Иван В. Григоров</i>	127	APPLICATION OF RECURSIVE METHODS FOR PARAMETER ESTIMATION IN ADAPTIVE CONTROL OF DC MOTOR WITH MINIMAL VARIANCE <i>Ivan V. Grigorov</i>	1.

2.	РАЗРАБОТВАНЕ НА МАТЛАБ ФУНКЦИЯ ЗА СЕЛЕКЦИЯ НА ПОДХОДЯЩ УЕЙВЛЕТ ЗА ФИЛТРАЦИЯ НА СИГНАЛИ <i>Ренета Д. Първанова, Марияна Г. Тодорова, Стефан Л. Стефанов</i>	136	DEVELOPMENT OF MATLAB FUNCTION FOR SELECTION OF SUITABLE WAVELET FOR SIGNAL FILTRATION <i>Reneta D. Parvanova, Mariyana G.Todorova, Stefan L. Stefanov</i>	2.
3.	РАЗРАБОТВАНЕ НА GUI В МАТЛАБ ЗА ИЗСЛЕДВАНЕ НА ДИНАМИЧНИТЕ И ЧЕСТОТНИ ХАРАКТЕРИСТИКИ НА ИДЕАЛНИ ЛИНЕЙНИ РЕГУЛАТОРИ <i>Мариела И. Александрова</i>	141	DEVELOPMENT OF MATLAB GUI FOR INVESTIGATION OF STEP AND FREQUENCY RESPONSES OF IDEAL LINEAR CONTROLLERS <i>Mariela I. Alexandrova</i>	3.
4.	РАЗРАБОТВАНЕ НА GUI В МАТЛАБ ЗА ИЗСЛЕДВАНЕ НА ДИНАМИЧНИТЕ ХАРАКТЕРИСТИКИ НА ПРОМИШЛЕНИ РЕГУЛАТОРИ <i>Мариела И. Александрова</i>	151	DEVELOPMENT OF MATLAB GUI FOR INVESTIGATION OF STEP RESPONSES OF REAL INDUSTRIAL CONTROLLERS <i>Mariela I. Alexandrova</i>	4.
5.	РАЗРАБОТВАНЕ НА МОДЕЛ НА АДАПТИВЕН НАБЛЮДАТЕЛ НА СЪСТОЯНИЕТО ЗА SISO ЛИНЕЙНИ СТАЦИОНАРНИ ДИСКРЕТНИ СИСТЕМИ В СРЕДАТА SIMULINK НА МАТЛАБ <i>Веселин Луков</i>	161	DEVELOPING A MODEL OF ADAPTIVE STATE OBSERVER FOR SISO LINEAR STATIONARY DISCRETE SYSTEMS IN MATLAB/SIMULINK ENVIRONMENT <i>Veselin Lukov</i>	5.

КОМПЮТЪРНИ СИСТЕМИ И МРЕЖИ



COMPUTER SYSTEMS AND NETWORKS

АНАЛИЗ И СРАВНЕНИЕ НА МЕЖДУНАРОДНИТЕ СТАНДАРТИ ЗА ДЕЙТА ЦЕНТРОВЕ

Росен С. Радков

Резюме: Дейта центровете са много важни за бизнеса. Те осигуряват услуги и приложения, които се използват както от фирмите така и от гражданите. При проектирането и изграждането на ДЦ трябва да се вземат под внимание не само изискванията на бизнеспроцесите, но и изискванията на международните и национални стандарти. Важно е специалистите, които проектират и изграждат ДЦ, да са запознати със съществуващите стандарти и добри практики. Тази статия анализира стандартите и определя тяхната приложимост при решаването на отделните задачи, решавани при проектиране и изграждане на ДЦ.

Ключови думи: достъпност, безотказност, дейта център, възстановяване на бизнеса, непрекъснатост на бизнеса, резервираност.

ANALYSIS AND COMPARISON OF INTERNATIONAL STANDARDS FOR DATA CENTER INFRASTRUCTURE

Rosen S. Radkov

Abstract: Data centers are essential for the business today. They provide services and applications that are used by both company and ordinary citizens. To build a DC infrastructure in order to meet the specifics of the business processes it is needed to consider both its peculiarities, international and local standards. It is important that the designers of DCs have knowledge of the ideas incorporated into the various standards and knowledge of best practices. This article analyzes the indicators used by various standards that define the requirements when designing a data center.

Keywords: Availability, reliability, data center, disaster recovery, business continuity, redundancy.

1. Въведение

Дейта центровете (ДЦ) са от съществено значение за фирмите и пазарите. Високата производителност и високата достъпност са основни изисквания към тях, независимо от размера им. Те трябва да са на разположение 24 часа в денонощието и 365 дни в годината. Само използването на висококачествено оборудване не е достатъчно, за да се гарантира безпроблемната им работа. Замяната на съществуващите компоненти с такива, които осигуряват по-висока производителност, също е неадекватно решение. Днес наличието на сигурен и надежден ДЦ, който работи ефективно, е конкурентно предимство за всяка компания. Експлоатацията на ДЦ трябва да отговаря на международните правни норми, изискванията за сигурност на информацията и да отговарят на специфичните изисквания на компанията.

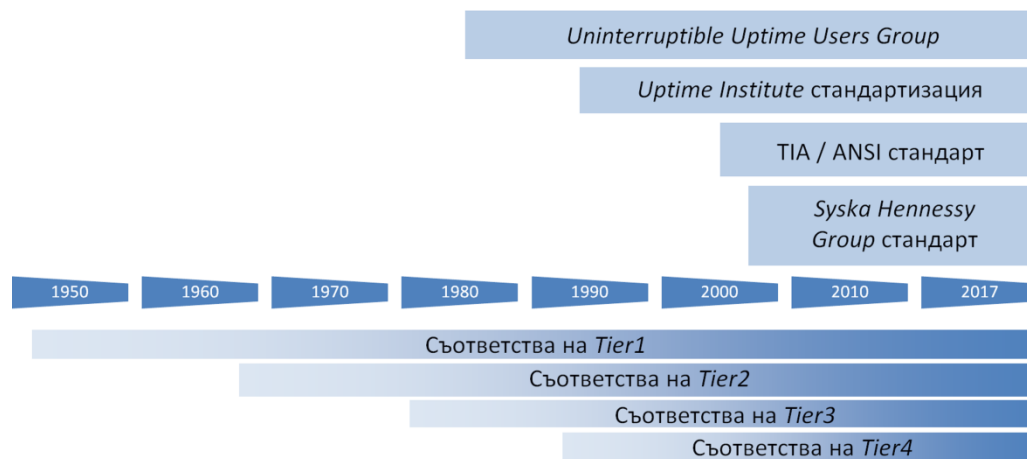
В настоящия момент съществуват множество стандарти за проектиране и изграждане на ДЦ или части от неговата инфраструктура. Задачата на тази статия е да определи кой от стандартите за центрове за данни трябва да се следва при проектиране и изграждане на ДЦ или част от неговата инфраструктура.

2. Стандарти за проектиране, изграждане и експлоатация на ДЦ

Анализът на литературата показва, че първият опит за обмен на знания относно критичната ИТ инфраструктура (ИТИС) е направен при създаването на Uninterruptible

Uptime Users Group (UUUG) през ноември 1989 г. Водещият принцип на UUUG, за да стане по-късно форум 7x24, е "да осигуряват форум за обмен на знания за тези, които проектират, изграждат, използват и поддържат критични корпоративни ИТИС". Няколко години по-късно, през 1993 г., фирмата The Uptime Institute Inc. създава сайта „Uptime Network“ с цел директно да адресира и разреши проблемите, отнасящи се до непрекъснатата работа на инфраструктурата на ДЦ (мощност, охлаждане и електрическа среда). След това те публикуват документа "Tier Classifications Define Performance Site Infrastructure White Paper", който става насока за бизнеса за проектиране и изграждане на ДЦ. През април 2005 г. Асоциацията на телекомуникационната индустрия (TIA) съвместно с Американския национален институт по стандартизация (ANSI) публикуват стандарта TIA-942. По-късно, още организации работят в областта на подготовката на стандарти, препоръки и най-добри практики за проектиране и експлоатация на ДЦ [1], [2].

Развитието на стандартите за ДЦ във времето е показано на фигура 1.



Фиг. 1. Развитие на стандартите за ДЦ [1]

Водещи стандарти за проектиране на ДЦ или част от системите в ДЦ в момента са:

2.1. Стандарт ANSI/TIA-942-A „Telecommunication Infrastructure Standard for Data Centers” [3], [4], публикуван през март 2014 г. с обем от 118 страници дефинира ДЦ като сграда или част от сграда, чието основно предназначение е да осигури зала за компютърна техника и необходимите съпътстващи помещения. Той си поставя за цел да дефинира изисквания и насоки за проектиране и изграждане на ДЦ по отношение на сградите, кабелните системи и мрежовите топологии. Предназначен е за тези, които се нуждаят от знания за цялостното проектиране на ДЦ, включително сградите на ДЦ, кабелната система и дизайна на мрежата. Стандартът дава възможност да се направи мултидисциплинарно проектиране, отчитайки особеностите на различните системи, които трябва да бъдат проектирани. Важно е да се отбележи, че насоките се отнасят и към възможностите за разширяване на ДЦ, за да се отговори на изискванията на новите бизнес процеси, които трябва да бъдат осигурени или на промените в бизнес процесите, които текущо се обслужват. Документът представя топологията на инфраструктурата за достъп и свързването на отделните елементи, влизащи в състава на ДЦ, използвайки знанията и вариантите, известни към този момент. Обръща се внимание и на разположението на елементите в помещенията от гледна точка на баланс между сигурност и управляемост, както и плътност на разположение на сървърните и комуникационните шкафове. Стандартът определя обща кабелна система за ДЦ и свързаните с него съоръжения, чиято основна функция е обвързана с информационните технологии. Има се предвид, че помещенията и кабелната система може да се използват както от една фирма, когато става дума за частен ДЦ, така и да бъдат предоставени на много фирми, когато става дума за ДЦ, предоставящ хостинг например. ДЦ се изгражда, за да

обслужва бизнес процесите в продължение на десетилетия. В този смисъл проектирането и начинът на инсталиране влияят на жизнения цикъл и устойчивостта на електронното оборудване. Те влияят и на околната среда, поради което стандартът насърчава проектантите да отчитат особеностите на местното законодателство и норми.

Стандартът определя две категории изисквания (критерии) [4]: задължителни (mandatory) и консултативни (advisory). Задължителните изисквания са обозначени с думата трябва (shall), а консултативните с думите: препоръчително (should), може (may), желателно (desirable). Задължителните изисквания се прилагат за защитата, производителността, администрирането и съвместимостта и указват минимално необходимите. Препоръчителните изисквания са насочени към разширяване на общата производителност на кабелната система. Проектантите на ДЦ трябва да имат предвид и всички други приложими ТИА стандарти, както и локални нормативни закони, правилници и наредби. В отделни раздели на стандарта са описани изискванията за нивата за комуникационната инфраструктура, архитектурата, електрозахранването и механичното (HVAC) оборудване. В допълнение F към стандарта, което е информативно и не се счита за част от него, се прави класификация на ДЦ според нивото (Rating) на устойчивост на неговата инфраструктура. За класификацията се има предвид нивото на резервираност на компонентите, възможността за едновременна поддръжка и експлоатация на ДЦ и устойчивостта на откази. Дефинираните четири нива се означават с: Rated 1, Rated 2, Rated 3 и Rated 4 (до 2014 г. са означавани с Tier1, Tier2, Tier3 и Tier4). Класификацията се отнася до архитектурата (A), механичните (M), електрическите (E) и комуникационните (T) части на ДЦ. Когато ДЦ има различно ниво за четирите си компоненти, за ниво на устойчивост на ДЦ се приема нивото на най-слабия компонент.

2.2. Стандарт EN50600-х

Стандартът EN50600-х е разработен от CENELEC, за да адаптира въпросите за проектирането, изграждането и поддръжката на ДЦ към Европейските условия. Огромният брой въпроси, които трябва да бъдат решени, е разделен в няколко части на стандарта: БДС EN 50600-1 - Общи концепции [5]; БДС EN 50600-2-1 - Конструкция на сгради [6]; БДС EN 50600-2-2 - Разпределение на захранването [7]; БДС EN 50600-2-3 - Контрол на околната среда [8]; БДС EN 50600-2-4 - Инфраструктура на телекомуникационно окабеляване [9]; БДС EN 50600-2-5 - Системи за сигурност [10]; БДС EN 50600-3-1- Информация за управлението и функционирането [11]; БДС EN 50600-4-1 - Общ преглед и общи изисквания за ключови показатели за резултатност [12]; БДС EN 50600-4-2 - Ефикасност на използване на енергията [13]; БДС EN 50600-4-3 - Коефициент на възобновяема енергия [14]. Подобно на другите стандарти е предназначен към широк кръг лица, свързани с управлението, консултирането, проектирането, доставката и инсталирането на оборудване в ДЦ. Детайлизира въпросите, които трябва да бъдат взети предвид при анализ на бизнес риска; използваната терминология; общите аспекти, отнасящи се до сградите и инфраструктурата; дефинираната класификационна схема; общите принципи за проектиране. Създателите изтъкват, че стремежът им е да създадат бизнес ориентиран, а не финансов подход [15].

2.3. Стандарт ISO/IEC 24764 „Generic cabling systems for data centres” от 2010 г. [16] – в рамките на 37 страници специфицира само кабелната система в ДЦ. Дефинирани са изискванията и са дадени препоръки как да бъде изградена кабелната система в ДЦ, за да се получи необходимата производителност.

2.4. Американският стандарт Uptime Institute Tier Standard (TS) от 2014 г. [17]–[20] с обем от 13 страници, който не е издание на стандартизираща организация, специфицира изисквания както за сградите, така и за инфраструктурата. Uptime Institute си поставя за цел да създаде обективна основа за сравнение на функционалността, капацитета и надеждността на топологията на даден ДЦ спрямо друг ДЦ или за сравнение на група от ДЦ. Uptime Institute, основан през 1993 г. с локация в Санта Фе (Ню Мексико), създава „Data Center Site

Infrastructure Tier Standard: Topology” (TS:T) [17] и „Data Center Site Infrastructure Tier Standard: Operational Sustainability” (TS:OS) [18] като за краткост, съвкупността от двата ще означим с TS, с цел да бъдат обективна база за сравняване на възможностите на различни топологии, както на група от сайтове.

TS:T описва функционалните изисквания към инженерните системи в ДЦ, които да позволят да се достигнат бизнес целите или да се изпълни стратегическата цел на организацията. Стандартът описва критериите за разделяне на инфраструктурата на ДЦ на четири нива, Tier I, Tier II, Tier III и Tier IV, според нивото на резервиране на активните компоненти и комуникационните трасета и възможността за едновременната експлоатация и поддръжка на ДЦ. Класификацията е създадена за систематично описание на инфраструктурата, необходима за работата на ДЦ. В нея не се описват характеристиките на конкретни системи и подсистеми.

Предназначението на TS:T е да предостави на проектантите на ДЦ, операторите в ДЦ и мениджърите (не технически) средства за обективно определяне на очакваната надеждност на различните топологични структури на ДЦ. Счита се, че работата на ДЦ зависи от успешната и съгласувана работа на електрическите, механичните и другите инженерни системи. Всяка една подсистема и система трябва да бъде изградена по такъв начин, че да отговаря на степента на отказоустойчивост, определена от съответното ниво (Tier). Най-важното, което собствениците и проектантите на ДЦ трябва да отчитат при вземането на решения, е как компромисите ще окажат влияние на работата на ИТ системата в процеса на експлоатация.

TS:OS [18] определя алгоритмите и рисковете за всяко едно от нивата, които влияят за достигането на целите на ДЦ или решаването на бизнес задачите в дългосрочен план. Предназначен е за собствениците, операторите и мениджърите на ДЦ. Той помага на собствениците на ДЦ да повишат ефективността на инвестициите си в ДЦ. Състои се от три части: управление и операции (Management & Operations); характеристики на сградите (Building Characteristics); място на ДЦ (Site location). Всяка една от тези три части се състои от няколко категории и компоненти, с които са свързани съответни алгоритми и рискове. Определените в стандарта алгоритми са класифицирани по приоритети и са представени в таблица, което дава възможност на собствениците да се ориентират лесно и да оценят кои от процесите са свързани с по-високи рискове, за да предприемат съответните действия. Действия трябва да бъдат предприети при проектирането, при прехода към експлоатация, както и при самата експлоатация на ДЦ. В таблиците са представени възможните алгоритми, а не конкретни изисквания, тъй като те могат да бъдат реализирани по различни начини. Важно е обаче, всички алгоритми да се изпълняват едновременно и да бъдат ефективни, за да може да се постигне съответното ниво на достъпност на ДЦ. Ефективността на TS:OS се определя чрез три принципа: проактивност, практическо използване и информираност. Един алгоритъм се счита за ефективен, когато са представени доказателства за използването и на трите принципа. Доказателство за „проактивност“ са добре документиранни процеси за всички съществуващи и предстоящи действия, както и за процедурите за непрекъснат контрол и обновяване. Вторият принцип „практическо използване“ служи за потвърждение, че процедурите и процесите се изпълняват винаги и по един и същ начин, независимо кой и кога ги изпълнява. „Информираност“ означава всички служители в организацията да притежават необходимите знания и умения за дейностите, които се налага или може да се наложи да изпълнят. За тази цел всеки служител трябва да знае: 1) че има процедура, по която се изпълнява съответната дейност, 2) къде да намери процедурата и 3) че има достъп до нея. TS:OS не включва мерки, свързани с: управление на сигурността, охраната, околната среда и персонала.

2.5. Стандарт ANSI/BICSI 002 “Data center design and implementation best practices” [3], [21]–[25], с последна версия от 2014 г. и обем от 534 страници – стандарт, създаден от

BICSI (Building Industry Consulting Service International, Inc.) с цел да предостави добри практики и методи на изпълнение, които да допълнят TIA, CENELEC, ISO/IEC и други публикувани стандарти при проектиране и експлоатация на ДЦ. Стандартът [21]–[28] представлява подробен справочник за терминологията на ДЦ и ръководство по отношение на добрите практики за проектирането на ДЦ. Предназначен е за инвеститорите и собствениците на ДЦ, операторите, консултантите и мениджърите на проекти, както и за инсталаторите на оборудването и инфраструктурата на ДЦ. В него подробно се разглеждат въпросите, свързани с избора на място на ДЦ, планирането и разпределението на площите в ДЦ, архитектурата на ДЦ и различните видове системи: електрически, механични, противопожарни, телекомуникационни, управляващи и системи за сигурност. Разгледани са и въпросите, свързани с въвеждането на ДЦ в експлоатация, неговата поддръжка и управление. В приложенията към стандарта има препоръки, отнасящи се до процеса на проектиране, оценката на надеждността и достъпността. Представени са процедури за класификация на ДЦ според нивото на съответните компоненти. В стандарта е представен подход за определяне на необходимия клас на достъпност на ДЦ, базиран на оперативните изисквания и необходимата оперативна достъпност. BICSI имат претенции, че техният стандарт превъзхожда останалите, защото: служи като ръководство и дава препоръки; осигурява добри практики, които в някои случаи са по-строги от изискванията на другите стандарти; засяга въпроси, които са или повърхностно разгледани, или въобще не са разгледани в стандартите за окабеляване. Основни раздели в стандарта са: планиране на помещенията и модулни ДЦ; избор на локация; проектиране на архитектурата; проектиране и изграждане на електрическите, механични системи, системите за сигурност, пожароизвестяване и пожарогасене, BMS; проектиране и изграждане на комуникациите и ИТИС; управлението и поддръжката на ДЦ; надеждността на ДЦ.

BICSI 002 дефинира пет класа ДЦ според нивото на достъпност (availability) [29]: клас 0 (Class-0); клас 1 (Class-1); клас 2 (Class-2); клас 3 (Class-3) и клас 4 (Class-4).

Въведени са префикси към класовете за различните типове системи: Class Fx – за сградите (електрическите и механични системи); Class Cx – за кабелните трасета; Class Nx – за мрежовата инфраструктура; Class Sx – за обработката на данни и сторидж системите; Class Ax – за приложенията.

Налице са и редица съпътстващи стандарти, като EN50173-5, NFRA Standard, ASHRAE Standard, TIA-568, ISO/IEC 11801 и други, които се отнасят до изграждането на отделни системи, които са неизменна част на ДЦ, като кабелните системи, електрическото захранване, климатизацията и т.н., както и стандарти на национални организации за изработване на стандарти, които са превели и/или адаптирали някой от описаните по-горе стандарти. Американските институти American National Standards Institute (ANSI) и Uptime Institute имат водещо значение за класификацията на ДЦ, а Uptime Institute е основополагащ при сертифицирането им.

3. Типове ИТИС, дефинирани в стандартите за ДЦ, според нивото на резервираност и отказоустойчивост

Класификациите на стандартите за типовете инфраструктура на ДЦ са изработени според степента на резервираност на компонентите и възможността за едновременна поддръжка и експлоатация на ДЦ. Те се отнасят за всички компоненти на ДЦ: климатизация, вентилация, ел. захранване, комуникации, мониторинг и т.н. Важно е да се отбележи, че се отчита устойчивостта на откази на всички оперативни и спомагателни процеси в ДЦ.

Основните характеристики на всяка една от инфраструктурите са:

Базовата инфраструктура за ДЦ (Tier I, Rated 1) е предназначена за малкия бизнес.

Основни характеристики: няма резервирани компоненти; няма резервирани трасета (distribution path) за информация и ел. захранване; единично ел. захранване; повредата на всеки един компонент влияе на работата на ДЦ; планираните дейности изискват повечето или всички компоненти на ДЦ да бъдат изключени; уязвим при изпълнение на планирани или непланирани събития; за извършване на годишна профилактика техниката трябва да бъде изключена; оперативни грешки на операторите или непредвидени откази са причина за спиране работата на ДЦ; нередовната профилактика увеличава риска от непланирани откази.

Инфраструктура на ДЦ с инсталирани резервни компоненти (Tier II, Rated 1), предназначена за средния бизнес, където бизнес процесите се ползват онлайн през работното време. Приемливи са интервали от време, през които ДЦ е недостъпен. Не е приемливо обаче, да има загуба на данни.

Основни изисквания: има резервирани компоненти, но те се включват в работа на мястото на дефектирани компоненти със същото предназначение; налице е само едно трасе за информация и ел. захранване; необходимо е да се поддържа гориво, осигуряващо работата на генератора в продължение на 12 часа; планираното извеждане на активен компонент от експлоатация не води до прекъсване на работата; извеждането от експлоатация на канал за разпределяне на информацията, за техническа профилактика или по друга причина, изисква спирането на работата на ДЦ; постоянно инсталираната изчислителна мощност е достатъчна за удовлетворяване на нуждите, така че извеждането от експлоатация на резервните компоненти да не нарушава работата на ДЦ или да увеличава натоварването на ИТ системата; уязвим при планирани или непланирани събития; разполага с единично захранване, но се изисква наличие на UPS и генератор; изисква се наличие на двоен под; за извършване на годишна профилактика техниката трябва да бъде изключена. При извършване на профилактика се увеличава риска за прекъсване на работата.

Следващите две архитектури на ДЦ удовлетворяват изискванията на фирми, обикновено свързани с електронна търговия, финансови услуги и други онлайн транзакции, чиито бизнес изискват достъпност 24x365.

Архитектура на ДЦ с резервна инфраструктура (Concurrently Maintainable – Tier III, Rated 3).

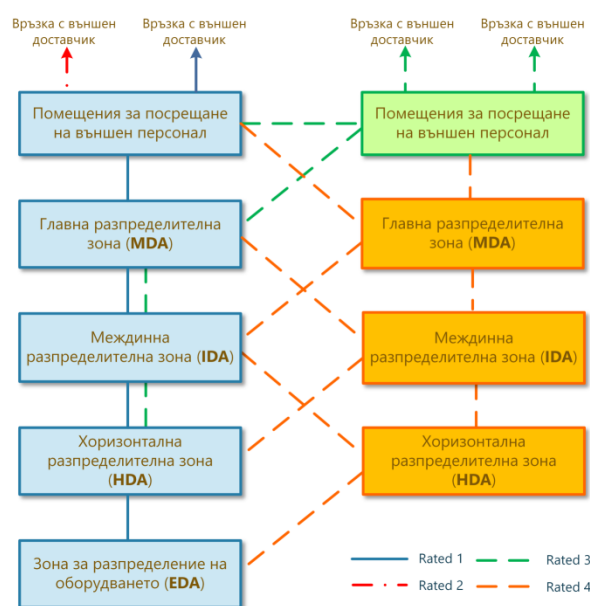
Основни изисквания: дублиране на всички компоненти и трасетата за информация и ел. захранване. За нормална работа на ДЦ е необходимо да се използва само едно от трасетата; всяко едно ИТ оборудване е с два захранващи блока и е включено по такъв начин, че да осигури съвместимост с топологията на инфраструктурата; необходимо е да се поддържа гориво, осигуряващо работата на генератора в продължение на 12 часа; резервните компоненти образуват резервна система. При дефектиране на компонент от работещата система, работата по обработката на данните се прехвърля към резервната система, т.е. само една от двете системи работи в даден момент от време; всеки един компонент може да бъде премахнат при планирана операция, без това да наруши работата на системата; постоянно инсталираната изчислителна мощност е достатъчна за удовлетворяване на нуждите, така че извеждането от експлоатация на резервните компоненти да не нарушава работата на ДЦ или да увеличава натоварването на ИТ системата; уязвима при непланирани дейности; всеки един от компонентите може да бъде изключен за планирана профилактика, без това да се отрази на натоварването на ИТ системата. По време на планираната профилактика работата се прехвърля към резервната система; при извършване на профилактика се увеличава риска за прекъсване на работата; грешки в операциите или случайни откази могат да причинят нарушаване на работата на ДЦ.

Отказоустойчива (Fault Tolerance) архитектура на ДЦ (Tier IV, Rated 3).

Основни изисквания: ДЦ е оборудван с няколко независими и физически изолирани системи, които са изградени чрез резервиране на оборудването и няколко независими

трасета; цялото ИТ оборудване е с двойно захранване и е инсталирано така, че да бъде съвместено с топологията на ДЦ; системите, които се резервират една друга, трябва да бъдат физически изолирани една от друга, за да не може едно събитие да повлияе на двете едновременно; задължително е осигуряването на непрекъснато охлаждане; необходимо е да се поддържа гориво, осигуряващо работата на генератора в продължение на 12 часа; инфраструктурата е устойчива на единични случайни откази; системата за управление на инфраструктурата реагира автоматично на отказ, така че работата на системите да продължи без прекъсване; всеки един компонент трябва да може планирано да бъде премахнат, без това да наруши работата на системата; когато компонент бъде изведен от експлоатация, това не трябва да се отразява на работоспособността и натоварването на ИТ системата; непланиран отказ не води до нарушаване на работата на системите; не е уязвим при извършване на планирани дейности; при извършване на профилактика рискът за прекъсване на работата може да се увеличи; задействането на системите за пожароизвестяване, пожарогасене или аварийното изключване на електрозахранването могат да доведат до нарушаване на работата на ДЦ.

Нагледна илюстрация за различията между нивата на ДЦ, отнесено към комуникационната инфраструктура, според ANSI/TIA-942-A е показана на фиг. 2[4].



Фиг. 2. Резервираност на комуникационните трасета

Основна характеристика за измерване на качеството на услугите, които осигурява ДЦ, е неговата достъпност [3], [24], [30], [31]. Във всички договори, които се сключват при наемане на услуги от ДЦ (Service level Agreement - SLA), като показател за нивото на услугите, фигурира число в проценти (например 99.9%), което дефинира достъпността на услугите на годишна база. Това число се свързва с нивото на ДЦ според класификацията, използвана от стандарта, по който е сертифициран ДЦ. В тези договори се описват и критериите, по които се оценява качеството на услугите, както и неустойките, които се дължат при неизпълнението на тези критерии. Независимо от това има специалисти, които твърдят, че достъпността не е правилната метрика. Според Милър [31], достъпността на ДЦ е подвеждаща метрика, защото в действителност е важно да се осигури работата на ИТИС на фирмата или на клиентите, а не само електрическото захранване на ДЦ. Основание за оспорването на класификациите по този критерий са разсъжденията върху това, че при едно и също сумарно време за недостъпност на ДЦ, времето за възстановяването (MTTR) му може да бъде различно. Например: а) има прекъсване на електрическото захранване на ДЦ за 5

минути; б) има прекъсване на електрическото захранване 10 пъти по 30 секунди, което сумарно пак е 5 минути. Еднакво ли ще бъде времето за възстановяване? [31]. Отговорът на този въпрос не е еднозначен, защото зависи от конкретната ИТ инфраструктура и принципа на работа на приложните ИС, но базирайки се на опита от практиката в повечето случаи времето за възстановяване във втория случай ще бъде много по-голямо. Това означава, че ако достъпността на ДЦ се гарантира на базата на горните таблици, има голяма вероятност да се направи грешка.

4. Заключение

Проектирането на ДЦ изисква добри знания в различни области на науката: комуникации, сървърна техника, електричество, климатизация и вентилация и т.н. Изискванията на стандартите и добрите практики са описани накратко по-горе, а резултатът от сравнението по различни критерии е дадено в Таблица 1, означението на инфраструктурите, според различните стандарти, е дадено в Таблица 2.

Таблица 1. Сравнение на стандартите за ДЦ

	ANSI/BICSI 002 (2014)	TIA-942-A (2014)	ISO24764 (2010)	EN-50600-X (2014)	Uptime TS (2014)
Избор на място	X	информативно	-	X	X
Разпределение на площите	X	X	-	X	X
Архитектура и структура	X	информативно	-	X	-
Електрическа система	X	информативно	-	X	X
Механични системи	X	информативно	-	X	-
Пожарна защита	X	Друг стандарт	-	X	-
Системи за сигурност	X	-	-	X	-
BMS	X	-	-	X	-
Комуникации	X	X	X	X	X
Информационни технологии	X	-	-	X	X
Въвеждане в експлоатация	X	-	-	X	-
Поддръжка	X	-	-	X	X
Персонал	-	-	-	-	X
Проектиране	информативно	-		X	-
Безотказност и достъпност	информативно	-		X	X

Таблица 2. Означение на инфраструктурите

	ANSI/BICSI 002	TIA-942-A	EN-50600-X	Uptime TS
Базова ИТИС	Class 0/1	Rated 1	Class 1	Tier I
ИТИС с резервиране	Class 2	Rated 2	Class 2	Tier II
ИТИС с едновременна поддръжка	Class 3	Rated 3	Class 3	Tier III
Отказоустойчива ИТИС	Class 4	Rated 4	Class 4	Tier IV

В резултат на анализа може да се направи извод, че само EN 50600-X и ANSI/BICSI 002 разглеждат всички аспекти на проектирането, внедряването и поддръжката. Единствено EN 50600-X определя минималните изисквания за проектиране, но ANSI/BICSI 002 предоставя най-изчерпателна информация, която може да се използва като препоръки и добри практики. ДЦ трябва да бъдат стандартизирани и сертифицирани. Това помага на заинтересованите да намерят правилния отговор на много въпроси, например: какъв тип ДЦ бизнес модел да се избере, колко пари да инвестира в ДЦ, как да избере подходящ доставчик на ДЦ и т.н.

Литература

- [1] R. Balodis and I. Opmane, "History of Data Centre Development," *Reflections Hist. Comput. Preserv. memories Shar. stories*, pp. 179–202, 2012.
- [2] J. Woods, "The evolution of the data center : Timeline from the Mainframe to the Cloud," *Silicon Angle*, 2014. [Online]. Available: <http://siliconangle.com/blog/2014/03/05/the-evolution-of-the-data-center-timeline-from-the-mainframe-to-the-cloud-tc0114/>.
- [3] P. Isaak, "BICSI 002 / TIA-942 / Uptime Which Data Center Standard Should I Follow."
- [4] ANSI/TIA, *TIA-942-A Telecommunications Infrastructure Standard for Data Centers*, no. March. USA, 2014, p. 120.
- [5] CENELEC, *БДС EN 50600-1:2012*. 2012, p. 29.
- [6] CENELEC, *БДС EN 50600-2-1:2014*. 2014, p. 33.
- [7] CENELEC, *БДС EN 50600-2-2:2014*. 2014, p. 38.
- [8] CENELEC, *БДС EN 50600-2-3:2014*. 2014, p. 29.
- [9] CENELEC, *БДС EN 50600-2-4:2015*. 2015, p. 38.
- [10] CENELEC, *БДС EN 50600-2-5:2016*. 2016, p. 40.
- [11] CENELEC, *БДС EN 50600-3-1:2016*. 2016, p. 49.
- [12] CENELEC, *БДС EN 50600-4-1:2017*. 2017, p. 17.
- [13] CENELEC, *БДС EN 50600-4-2:2016*. 2016, p. 3.
- [14] CENELEC, *БДС EN 50600-4-3:2017*. 2017, p. 21.
- [15] R. Cardigan, "European Data Centre Infrastructure Standards – EN 50600 series," 2014.
- [16] ISO/IEC, *ISO/IEC 24764*. 2010.
- [17] Uptime Institute, *Data Center Site Infrastructure Tier Standard : Topology*. 2014.
- [18] Uptime Institute, *Data Center Site Infrastructure Tier Standard : Operational Sustainability*. 2014.
- [19] P. Turner, J. H. Seader, V. Renaud, and K. G. Brill, "Tier Classifications Define Site Infrastructure Performance," 2008.
- [20] E. Maddison, "Tier Classification of Data Centres," 2014.
- [21] ANSI/BICSI, *ANSI/BICSI 002-2014*. 2014, p. 534.
- [22] B. Faber, "BICSI NxtGEN & Global Data Center Standards Update," 2016.
- [23] S. Banks, "BICSI Data Center Standard," 2010.
- [24] R. Ciordia, "ANSI / BICSI 002 , Data Center Design and Implementation Best Practices BICSI International Standards," 2014.
- [25] G. Humbert, "Data Center and IT infrastructure (BICSI)," *DATUM*, vol. 15, p. 28, 2015.
- [26] B. Camerino, "ANSI / BICSI 002-2014 Data Center Design and Implementation Best Practices," 2014.
- [27] J. Jew, "What BICSI , TIA , & ISO/IEC are doing with standards and why you should care."
- [28] J. Jew, "Data Center Practices – the Good , the Bad , & the Ugly," 2011.
- [29] ANSI/BICSI, *ANSI/Bicsi 002*. 2014.
- [30] Reichle & De-Massari AG, *R & M Data Center*. 2011.
- [31] R. Miller, "Class: New data center metric targets probability and risk," 2015. [Online]. Available: <http://datacenterfrontier.com/class-new-data-center-metric-targets-probability-of-failure/>.

За контакти:

ас. Росен С. Радков

Катедра „Софтуерни и Интернет Технологии“

Технически университет-Варна, България

E-mail: rossen.radkov@tu-varna.bg

МЕТОДИКА ЗА ОСИГУРЯВАНЕ НА ВИСОКА НАДЕЖДНОСТ НА ДЕЙТА ЦЕНТЪР

Росен С. Радков

Резюме: Дейта центровете осигуряват работата на бизнес процесите в организациите и фирмите. В резултат на тяхната работа гражданите и фирмите получават възможност да използват приложенията, използвани от съответен бизнес процес и услугите, които се генерират от него. Важно условие за предоставяне на качествени услуги е осигуряването на висока надеждност на дейта центъра. Тази статия анализира решенията за създаване на висока надеждност и предлага методика за нейното осигуряване.

Ключови думи: достъпност, безотказност, дейта център, възстановяване на бизнеса, непрекъснатост на бизнеса

HIGH RELIABILITY DATA CENTER METHODOLOGY

Rosen S. Radkov

Abstract: Data Centers provide business processes in organizations and companies. As a result citizens and companies are given the opportunity to use the applications used by the relevant business process and the services that are generated by it. An important condition for providing high-quality services is the high availability of the data center. This article analyzes the solutions for creating high reliability and offers a methodology for its provision.

Keywords: Availability, reliability, data center, disaster recovery, business continuity

1. Въведение

Дейта центровете (ДЦ) са важен компонент на ИТ инфраструктурата (ИТИС) на организациите и фирмите. Съвременните изисквания към тях са да бъдат на разположение 24 часа в денонощието и 365 дни в годината, както и да имат високи производителност и достъпност. Удовлетворяването на тези изисквания не се гарантира само от използването на висококачествено оборудване. Организациите и фирмите все повече осъзнават, че използването на сигурен и надежден ДЦ, който да работи ефективно, се превръща в предимство пред техните конкуренти. Високото качество на услугите, предоставяни от ДЦ, зависи не само от неговата надеждност, а и от правилната му експлоатация. Тя трябва да е подчинена на международните правни норми, изискванията за сигурност на информацията и на специфичните изисквания на компанията.

В настоящия момент съществуват множество стандарти и добри практики за проектиране и изграждане на ДЦ или части от неговата инфраструктура, но липсва методика, която да описва действията и дейностите, които е необходимо да се извършат, за да се осигури високо надежден дейта център (ВНДЦ). Задачата на тази статия е да дефинира методика за осигуряване на висока надеждност на дейта център.

2. Анализ на решенията за създаване на ВНДЦ

За удовлетворяване на изискванията за непрекъсваемост на бизнес процесите в организациите е необходимо да се изгради високонадеждна ИТИС, която да запазва своята устойчивост на функциониране в условията на среда с дестабилизиращи фактори (ДФ). Решенията, които се прилагат, са насочени към осигуряване на:

- непрекъснатост на функционирането на ИТИС, т.е. решения, осигуряващи устойчивост на бедствие (disaster tolerance);
- възстановяване на функционирането на ИТИС, т.е. решения, осигуряващи възстановяване на работата след бедствие (disaster recovery).

Осигуряването на висока надеждност на ИТИС се осъществява не само чрез изграждането ѝ с високонадеждни хардуерни и софтуерни средства, а е съвкупност от съвместното действие на множество компоненти:

- технически средства;
- инженерни решения;
- финансови средства;
- информация;
- персонал;
- инфраструктура и околна среда;
- управление.

Според добрите световни практики и международния стандарт ISO27001 [1], [2], осигуряването на висока надеждност на ИТИС и сигурност на информацията не са възможни без подкрепа от ръководството на съответната организация. Подкрепата се изразява не само в осигуряването на финансови средства за изграждането и поддръжката на ИТИС, но и в дефинирането, разпространението и утвърждаването на правилата за работа с ИТИС и информацията, както и в осигуряване на компетентност на персонала.

В зависимост от конкретните особености и изисквания на организацията, съставът на ИТИС може съществено да се различава. При организации, които имат множество обекти (сайтове), е възможно да бъде направено разпределение на изчислителните ресурси и/или организиране на резервни ИТИС от тип студен, топъл или горещ резерв, използвайки собствени или наети помещения. В случай че организацията е разположена на една площадка, то резервна ИТИС може да бъде реализирана чрез наемане на площ, оборудване или инфраструктура.

Проектирането на решение, което да отговаря на изискванията за ВНДЦ, е свързано с дефиниране на стойностите на следните два показателя: RTO (Recovery Time Objective) и RPO (Recovery Point Objective) [3]–[7]. С RTO се дефинира времето, необходимо за пълното възстановяване на работата на ДЦ след авария (бедствие, инцидент), отчитано от момента на настъпване на събитието. С RPO – интервалът от време преди бедствието, приет от бизнеса като допустим, за който ще бъдат загубени данните (фигура 1).



Фиг. 1. Графично представяне на RPO и RTO

Съгласно някои литературни източници [8]–[11], необходимото ниво за възстановяване на ИТИС може да бъде определено на базата на матрица на съответствието (Таблица 1) между необходимото ниво на устойчивост на бизнес процеса (БП) и приложенията, които го реализират. Нивото на устойчивост може да бъде определено с помощта на матрица на съответствие на необходимото ниво на възстановимост (Таблица 2).

Таблица 1. Матрица на съответствие на нивото на устойчивост на ДФ

	Приложение	Приложение	Приложение	Приложение
БП 1	Ниво 3	Ниво 3	Ниво 1	Ниво 1
БП 2	Ниво 1	Ниво 1	Ниво 1	Ниво 2
.....
БП m	Ниво 4	Ниво 2	Ниво 1	Ниво 2
Необходимо ниво	Ниво 4	Ниво 3	Ниво 1		Ниво 2

Таблица 2. Матрица на съответствие за определяне на необходимото ниво на възстановимост

		RTO			
		T1 < 1 ч.	T1 < 4 ч.	T1 < 48 ч.	T1 < 5 дни
RPO	P1 < 15 мин.		Пример: DNS		
	P1 < 4 ч.				
	P1 < 24 ч.				
	P1 > 24 ч.	Пример: AD			

В Таблица 2 съответните нива са кодирани цветово (1 - оранжев, 2 - син, 3 - лилав и 4 - зелен).

Прилагането на решения за ВНДЦ позволява не само да се намалят времената на неработоспособност и за възстановяване на ИТИС, но и да се намалят загубите на информация и финансовите щети.

Тази минимизация може да се постигне, ако предварително, т.е. преди да е настъпило събитието (инцидента, бедствието), са извършени следните действия:

- изграден е ВНДЦ;
- разработен е „План за възстановяване след бедствие“ (ПВБ);
- обучен е персоналът за изпълнение на ПВБ;
- регулярно се провеждат тренировки на сценариите, които са записани в ПВБ;
- анализирани са резултатите от проведените тренировки и са направени необходимите корекции в ПВБ.

Въвеждането на ВНДЦ може да бъде разгледано от гледна точка на сигурност на информацията (СИ), а също и от технико-икономическа (ТИ) страна. От гледна точка на СИ това означава да се:

- направи оценка на риска за влиянието на ДФ на околната среда и да се вземат мерки за неговото намаляване;

- намали вероятността от загуба на данни;
- защити информацията и критичните ресурси от погиване;
- осигури цялостност, конфиденциалност и наличност на информацията;
- осигури достъпността на ИТИС.

От ТИ гледна точка въвеждането на ВНДЦ означава да се:

- минимизират щетите върху техниката и помещенията от бедствие;
- минимизират финансовите загуби;
- минимизира загубата на репутация;
- реализира алтернативен начин за предоставяне на услугите към клиента.

Анализ на литературата показва, че най-разпространените технически решения включват следните елементи [12]–[25]:

- клъстери;
- апаратен излишък;
- резервни ДЦ;
- системи за съхранение на данни;
- системи за резервни копия на информацията;
- системи за репликация на информацията;
- виртуализация;
- решения, базирани на облачни технологии.

Основните им характеристики са представени в Таблица 3.

Таблица 3. Характеристики на решения прилагани във ВНДЦ

	Описание	Предимства	Недостатъци
Клъстери	Група от компютри, които са свързани помежду си и функционират като една единица за обработка на информация. Тя се осъществява на различни нива на системата. Колкото повече нива обхваща технологията, толкова по-надежден, мащабируем и управляем е клъстера.	Автоматично предаване на управлението на ресурсите при отказ на активен възел и връщане обратно след възстановяването му. Разпределяне на натоварването между възлите. Мащабируемост и приемлива цена.	Непредсказуем характер на мрежовия трафик и конфликти в мрежата, голямо време на очакване на съобщение. Продължителен период на възстановяване в случай на множествени откази.
Апаратен излишък	Включване на резервни компоненти/системи в структурата с цел повишаване на надеждността.	Намалява се или се изключва напълно прекъсване в работата на ИТИС.	Разходи за допълнително оборудване. Ниска ефективност. Дълъг период на възстановяване в случай на поразяване в по-голям мащаб.
Резервни ДЦ	РДЦ дават възможност за възстановяване на услугите при излизане от строя на основния център. РДЦ може да бъде изграден с оборудване, което е с по-малка изчислителна мощ. В РДЦ трябва да се намират актуални копия на информацията.	Пренебрежимо малко време за възстановяване на услугите предоставяни от ИТИС, сравнимо с времето за отстраняване на единичен отказ. Минимално време на недостъпност. Свързаност към Интернет в друга локация.	Големи разходи за изграждане и поддържане на РДЦ.
Системи за съхранение на данни	Масиви от дискове, лентови системи, които са включени към сървърните системи. Използват се различни технологии SAN, NAS, DAS и др.	Модулност. Възможност за лесно разширяване и достъп от разстояние. Надеждност на съхранението на данните. Въз-можности за репликация на ни-во дискови масиви или томовете.	Сравнително висока цена
Системи за резервиране на информацията (Backup)	Система, която осигурява автоматизирано създаване на архивни и резервни копия на информацията.	Надеждност на съхраняването на данните. Автоматизиране на процесите по създаване на копията на информацията. Намаляване на обема на необходимото дисково пространство, използвайки методи за дедубликация.	Възстановяването на данните може да отнеме значително време. Ако се ползва като услуга, трябва да се има предвид, че не всеки доставчик на услуга гарантира конфиденциалност на данните.
Системи за репликиране на	Системи, които осигуряват асинхронна или синхронна репликация на данните.	Универсалност - ако се прави на ниво сторидж, процесът е независим от операционните	Възможно е да има проблеми с производителността при репликиране на бази от данни.

информацията (Replication)		системи и базите данни.	Изисква свързаност между възлите с ниска латентност за репликиране на приложението.
Виртуализиране на сървъри	Технология, която дава възможност по-ефективно да се използва хардуера, като върху една физическа машина се инсталират множество виртуални. Необходимо е да се използва хипервайзор, който е независим или е част или добавка към ОС.	Висока ефективност. Лесно реконфигуриране на хардуерните ресурси. Удобно управление на виртуалната структура. Възможност за автоматизирано прехвърляне на работата на машините върху друг хост.	Необходимост от допълнителни лицензи и висока квалификация на персонала.
Облачни решения (Cloud solutions)	Технология, която дава възможност да се използват ресурси, които са наети. Предлагат се IaaS, PaaS, SaaS, NaaS, DRaaS, SECaaS и други	Няма капиталови, а само оперативни разходи за използването на услугите. Намалява се времето за внедряване на ДЦ. Лесно преконфигуриране на хардуера, както и преминаване към нови версии на системния и приложен софтуер.	Зависимост от доставчика на услуги. Не отпада необходимостта от ИТ персонал за администриране на системата. Трябва да се оцени риска при ползването на такъв вид услуга.

Внедряването в ИТИС на някое от посочените в Таблица 3 решения и конкретната реализация на системата зависи от изискванията на възложителя. Различно е и времето за проектиране, валидиране и внедряване на решението, както и неговата сложност и цена. Крайната цел е проектираното и внедрено решение за ВНДЦ да удовлетворява изискванията на възложителя. Не са рядко случаите, когато процесът е итеративен, защото се налага да се направи промяна в проекта за постигане на ценова ефективност на решението или необходимостта от обслужване на нови бизнес процеси.

3. Методика за осигуряване на висока надеждност

Решенията, които могат да се приложат за изграждане на ВНДЦ, включват съвкупност от действия, които са насочени към опазване на информацията, намаляване на времето на неработоспособност и бързо възстановяване на ИТИС и услугите, които тя предоставя. На базата и проведения анализ на литературните източници [26]–[32] и личния опит, натрупан в практиката, в настоящата статия се предлага методика, на базата на която се изгражда ИТИС с висока надеждност. Предложената методика включва следните действия:

- а) Определяне на активите на ИТИС: хардуер, софтуер, бази данни, човешки ресурси, ползвани услуги и други;
- б) Определяне на заплахите и тяхното въздействие върху работата на ИТИС, изготвяне на оценка на риска и определяне на методите и средствата за неговото третиране;
- в) Анализ на бизнес процесите и информационните потоци, които ги обезпечават. Определяне на изискванията на възложителя за тяхната непрекъсваемост и приемливата загуба на данни;
- г) Определяне на архитектурата на ИТИС;
- д) Избор на апаратните и софтуерни компоненти за реализация;
- е) Определяне на стратегията и начините за резервиране на данните. Разработка и утвърждаване на „План за резервиране на информацията“ (ПРИ);
- ж) Разработка и утвърждаване на ПВБ. В него се описват разработените сценарии за отработване на възможни критични събития и инциденти. Всеки от сценариите съдържа информация за последователността от действия, които се извършват при активирането му, с конкретна информация за:

- предназначение, обхват и цели;
- ситуациите, в които се активира;
- последователност на действията;
- максимална продължителност на всяко едно от действията;
- роли и отговорности на лицата или екипите за всяко едно от действията;
- начин за свързване с лицата или службите;
- начин на оповестяване за приключването на инцидента;
- периодите, през които планът трябва да бъде тестван.

Необходимо е създаването и поддържането на информация, доказваща реалното изпълнение на плана.

з) Разработка на концепция за планиране на капацитета;

и) Дефиниране, утвърждаване и оповестяване на „Политика за сигурност на информацията“;

к) Разработка на процедури за:

- документиране, отработване и оценка на инциденти;
- оценка на работата на ИТИС и вземане на решение за подобрения, ако се налагат такива.

4. Заключение

Постигането на висока надеждност на ДЦ е задача, която организациите и фирмите трябва да решат за да осигурят своята безпрепятствена работа и конкурентноспособност. Липсата на ясна методика затруднява бързото и оптимално решаване на задачата. В тази статия се предлага методика, която да помогне на организациите и фирмите да решат по оптимален начин тази задача. От нейното съдържание става ясно, че реализирането на решение за ВНДЦ се осъществява не само чрез проектиране и въвеждане на високонадеждна ИТИС и комплекс от планове и процедури, но и чрез обучение на персонала, тренировки и оценка на работата на системата.

Литература

- [1] BDS, БДС ISO/IEC 27001. Bulgaria, 2014, p. 30.
- [2] Р. Червенкова, “COBIT и управлението на ИТ - така ли е при вас”, СЮ България, 2012 .
- [3] J. Hauser, “What is RTO and RPO and why should I care? Part One”, 2015. [Online]. Available: <http://innovativeii.com/what-is-rto-and-rpo-and-why-should-i-care-part-one/>.
- [4] J. Hauser, “What is RTO and RPO and why should I care? Part Two”, 2015. [Online]. Available: <http://innovativeii.com/what-is-rto-and-rpo-and-why-should-i-care-part-two/>.
- [5] O. H. Alhazmi and Y. K. Malaiya, “Assessing disaster recovery alternatives: On-site, colocation or cloud”, Proc. - 23rd IEEE Int. Symp. Softw. Reliab. Eng. Work. ISSREW 2012, pp. 19–20, 2012.
- [6] M. Wiboonratr and K. Kosavisutte, “Optimal strategic decision for disaster recovery”, Int. J. Manag. Sci. Eng. Manag., vol. 4, no. 4, pp. 260–269, 2009.
- [7] O. H. Alhazmi and Y. K. Malaiya, “Are the classical disaster recovery tiers still applicable today?”, Proc. - IEEE 25th Int. Symp. Softw. Reliab. Eng. Work. ISSREW 2014, pp. 144–145, 2014.
- [8] A. N. Pavlov and B. V. Sokolov, “Structural analysis of disaster-tolerance information system”, SPIIRAS Proc., vol. 9181, no. 8, pp. 128–153, 2009.
- [9] R. M. Feio, “Disaster Recovery (DR) Tiers”, p. 5, 2012.
- [10] R. R. Schulman, “Disaster Recovery Issues and Solutions”, Hitachi Data Syst., no. March, 2002.

- [11] Cisco Inc., “Disaster Recovery : Best Practices”, 2008.
- [12] I. Koren and C. Mani Krishna, Fault Tollerant Systems. ELSEVIER, 2007.
- [13] D. J. Sorin, Fault Tolerant Computer Architecture, vol. 4, no. 1. Morgan&Claypool Publishers, 2009.
- [14] S. Snedaker, Business Continuity&Disaster Recovery for IT Professionals, vol. 15, no. 1. Burlington: Syngress Publishing, Inc., 2007.
- [15] B. Highleyman, B. Holenstein, and P. J. Holenstein, “Fault tolerance vs. High Availability,” no. October, pp. 9–10, 2005.
- [16] M. Richards, “The secret to Building Highly Available Systems”, p. 6, 2011.
- [17] J. Rath, “Data Center Strategies: Simplifying high-stakes, mission critical decisions in a complex industry”, no. July, 2011.
- [18] R. Govindan, I. Minei, M. Kallahalla, B. Koley, A. V. Google, and A. Vahdat, “Evolve or Die: High-Availability Design Principles Drawn from Googles Network Infrastructure”, Proc. 2016 Conf. ACM SIGCOMM 2016 Conf., pp. 58–72, 2016.
- [19] General Electric, “Five Essential Components for Highly Reliable Data Centers”, 2011.
- [20] K. Chen et al., “WaveCube: A scalable, fault-tolerant, high-performance optical data center architecture”, Proc. - IEEE INFOCOM, vol. 26, pp. 1903–1911, 2015.
- [21] CenturyLink, “Seven Secrets to High Availability in the Cloud,” pp. 1–8.
- [22] C. Engelmann, “Symmetric Active / Active High Availability for High-Performance Computing System Services”, School of Systems Engineering, 2008.
- [23] E. Vargas, “High availability fundamentals”, Sun Blueprints Ser., no. November, 2000.
- [24] J. M. Carlson and J. Doyle, “Highly optimized tolerance: a mechanism for power laws in designed systems”, Phys. Rev. E. Stat. Phys. Plasmas. Fluids. Relat. Interdiscip. Topics, vol. 60, no. 2 Pt A, pp. 1412–1427, 1999.
- [25] A. Valadarsky, M. Dinitz, and M. Schapira“, Xpander: Unveiling the Secrets of High-Performance Datacenters,” Proc. 14th ACM Work. Hot Top. Networks, p. 16:1--16:7, 2015.
- [26] N. R. Kern and V. Peltz, “Disaster Recovery Levels”, IBM Syst. Mag., no. November, pp. 1–5, 2003.
- [27] H. Estall, BCMS implementation and certification to ISO22301. 2012.
- [28] ISO, ISO 22301 Societal security — Business continuity management systems — Requirements, vol. 2012. 2012, p. 34.
- [29] EU, “Единни държавни изисквания към изграждането и сертифицирането на центрове за съхранение на особено чувствителна информация за нуждите на централна държавна администрация в съответствие с изискванията на БДС ISO/IEC 27001:2005”, 2011.
- [30] J.-C. Laprie, “Dependable Computing and Fault-Tolerance: Concepts and Terminology”, Proceedings of International Symposium on Fault-Tolerant Computing, vol. III. pp. 2–11, 1985.
- [31] Y. Liu, J. K. Muppala, and M. Veeraraghavan, “A survey of data center network architectures”, p. 22pp, 2014.
- [32] J. C. Laprie, “Dependable computing: Concepts, Limits, Challenges”, in Computer Software and Applications Conference, 2004. COMPSAC 2004. Proceedings of the 28th Annual International, 2004, p. 242 vol.1-vol.

За контакти:

ас. Росен С. Радков

Катедра „Софтуерни и Интернет Технологии“

Технически университет - Варна

E-mail: rossen.radkov@tu-varna.bg

БАЛАНСИРАНО РАЗПРЕДЕЛЕНИЕ НА ТРАФИКА В ИНТЕГРИРАНА ETHERNET СРЕДА (Част първа)

Деян Х. Стоев, Явор З. Дечев

Резюме: Използването на интегрирани мрежи за предаване на различните видове трафик налага да се изследва влиянието върху производителността на два или повече класа трафик, споделящи една и съща среда. В настоящата статия са разгледани основните алгоритми за балансиране на потоците трафик, направен е сравнителен анализ между DRR и DRR-AWC. Извършена е оценка на ефективността на DRR-AWC алгоритъма.

Ключови думи: балансиране на потоците трафик, DRR, DRR-AWC.

BALANCED TRAFFIC DISTRIBUTION IN AN INTEGRATED ETHERNET ENVIRONMENT (Part one)

Dean H. Stoeff, Yavor Z. Dechev

Abstract: The usage of integrated networks for transfer of different types of traffic creates necessity to explore the impact on the system performance, caused by two or more traffic classes sharing the same media environment. In this article the basic algorithms for balancing the traffic flows are discussed and is performed a comparative analysis between DRR and DRR-AWC algorithms. An evaluation of the efficiency of the DRR-AWC algorithm is calculated.

Keywords: balanced traffic distribution, DRR, DRR-AWC

1. Въведение

С развитието и все по-голямото разпространение на виртуализацията и облачните технологии през последните години в съвременните центрове за данни се използват различни мрежови архитектури. Основният мрежов трафик минава по Ethernet, а обменът от и към хранилището за данни минава през оптична среда - Fiber Chanel (FC).

Наличието на различни мрежови архитектури в един център за данни има един голям недостатък – високата цена за управление и поддръжка. Това налага тенденция да се използва една и съща архитектура за предаване на различните видове трафик. Ethernet е най-подходяща за избор среда, тъй като е по-евтина и по-разпространена от Fiber Chanel (FC). Голям неин недостатък е, че в условията на интензивен трафик възникват задръствания и тя може да започне да пропуска пакети. За да се предават различните видове трафик по Ethernet, е необходимо да се осигури работа без загуби.

Целта на разработката е сравнителен анализ на протоколите и алгоритмите за разпределение на видовете трафик в интегрирана Ethernet среда за пренос на данни за постигане на балансираното ѝ натоварване.

За постигането на тази цел са поставени следните задачи:

- Анализ на капацитета и латентността на различните приложения като iSCSI и Message Passing Interface (MPI) в интегрирана мрежа;
- Анализ въздействието на стандартния алгоритъм Deficit Round Robin (DRR) върху производителността и разпределението на ресурсите в интегрираната мрежа;
- Оценка на ефективността на модифицирания DRR алгоритъм върху комутаторите в центъра за обработка на данни чрез изчисляване на индекса JFI – Jain's Fairness Index.

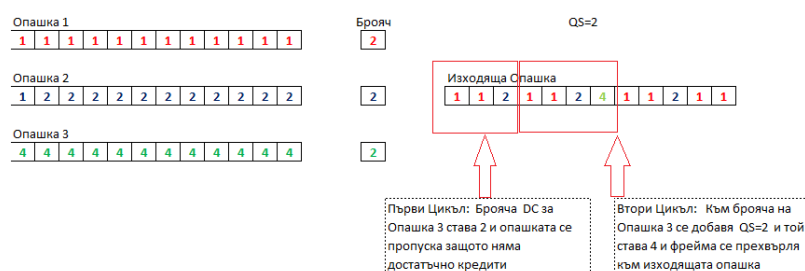
2. Справедливо (балансирано) разпределение на ресурсите

DCB технологията позволява на системния администратор да раздели целия трафик в мрежата на осем класа трафик. Един от проблемите, който се появява при разработката на DCB технологията, е, че поведението в един клас е неопределено. Докато производителността между осемте класа е добре дефинирана, то остава неразрешен проблемът с приоритизирането на трафика в рамките на един и същи клас. Повечето алгоритми за разпределение могат да се категоризират или като разпределение по потоци, или като разпределение по възли. Алгоритмите за разпределение по потоци осигуряват постигане на „равноправие“ между различните потоци трафик в рамките на мрежата, независимо от това къде са възникнали и независимо накъде са насочени. Този вид разпределение облагодетелства възли с множество потоци. Алгоритмите за разпределение по възли осигуряват „справедливо“ разпределение за всеки възел в рамките на мрежата. Този вид разпределение „ощетява“ възли с много потоци. Съвкупността от формални спецификации ни предоставя механизъм за определяне и кодиране на „равноправието“.

Алгоритъмът за претеглени „справедливи“ опашки Weighted Fair Queuing (WFQ) разделя данните, които се предават, в претеглени опашки и предават данни от всяка една опашка чрез „справедлив“ механизъм, често Simple Round-Robin Approach (прост алгоритъм с циклично управление). Най-простият алгоритъм разделя трафика на прости опашки с еднакви тегла и предава според Round-Robin принципа. По-сложните WFQ алгоритми могат да осигурят различни тегла на опашките и да предават от тях във вариации на RR или други способности. WFQ е много универсален алгоритъм и може да се използва в различни мрежови топологии.

Deficit Round-Robin алгоритъм (DRR)

WFQ е добре известен алгоритъм за разпределение, който поддържа справедливостта - равноправното разпределение на капацитета и ресурсите на мрежата между различните потоци от данни, осигурявайки добра производителност. Но той е „скъп“ за внедряване в хардуера и повечето комутатори използват DRR алгоритъма. DRR е сравним с WFQ по „справедливост“ и има по-ниска сложност. На фигура 1 е изобразен DRR алгоритъмът, като е представена примерна система с три входящи опашки отляво и една изходяща отдясно.



Фиг. 1. DRR Пример

Фреймовете се насочват към една от трите входящи опашки. Всяка входяща опашка е FIFO и съдържа фреймове, които се отнасят към отделни потоци. Алгоритъмът определя в какъв ред фреймовете се разполагат в изходящата опашка. От изходящата опашка фреймовете се предават по FIFO ред. Числата във всяка входяща опашка показват големината на фрейма в опашката. Алгоритъмът сканира всяка опашка, започвайки по ред 1, 2 и 3. При всяко сканиране 0 или повече фреймове от всяка входяща опашка се преместват към изходящата. Броят на фреймовете, преместени от всяка входяща опашка към изходящата, се определят от три променливи, а именно Quantum Size (QS) – размер на кванта в алгоритъма, Deficit counter (DC) - брояч и размер на фреймовете в опашката Frame Size (FS). Алгоритъмът има фиксиран размер на кванта, в този пример размерът на QS е 2.

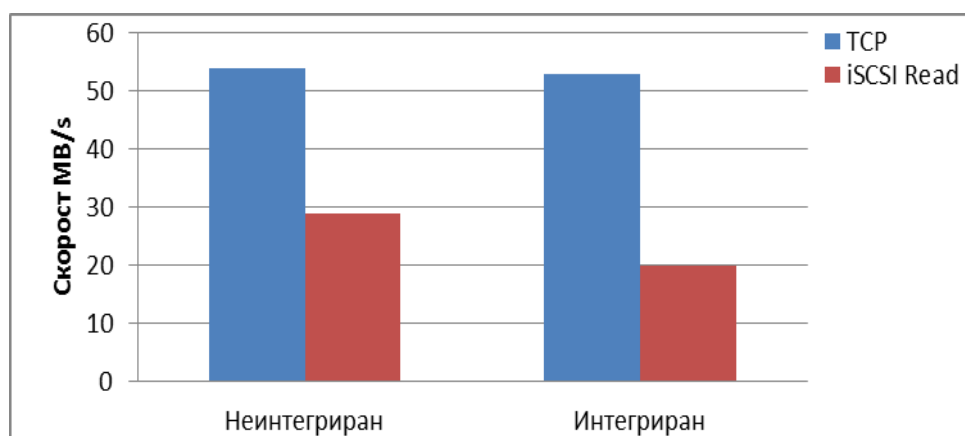
Броячите първоначално са установени в нула. Стойността на брояча определя броя на кредитите с натрупване през всеки цикъл. В началото на всеки цикъл размерът на кванта се добавя към брояча за всяка опашка. Броячът определя броя на кредитите за всяка входяща опашка заедно с неизползваните от предишния цикъл. След като броячът нарасне с размера на кванта в текущия цикъл, алгоритъмът започва да мести фреймове от входящите опашки към изходящата в последователен RR начин. Фреймът се премества от входящата в изходящата опашка, когато броячът е по-голям или равен на размера на фрейма в началото на опашката. След това съответният брояч се намалява с размера на преместените фреймове. В дадения пример очертаните в пунктир правоъгълници показват всеки цикъл от алгоритъма. При размер на кванта 2 се вижда, че при първия цикъл са преместени два фрейма с единичен размер от първата опашка и един двоен фрейм от втората опашка. Във втория цикъл брояч на опашка 3 запазва останалите кредити от първия цикъл и се прехвърля един четворен фрейм заедно с други единични и двойни фреймове. Този пример показва как по-малкият размер на кванта може да причини пропускане на фреймове, както в Опашка 3. За това се препоръчва размерът на кванта да се задава не по-малък от най-големия фрейм, който ще бъде прехвърлен.

3. Експериментална постановка

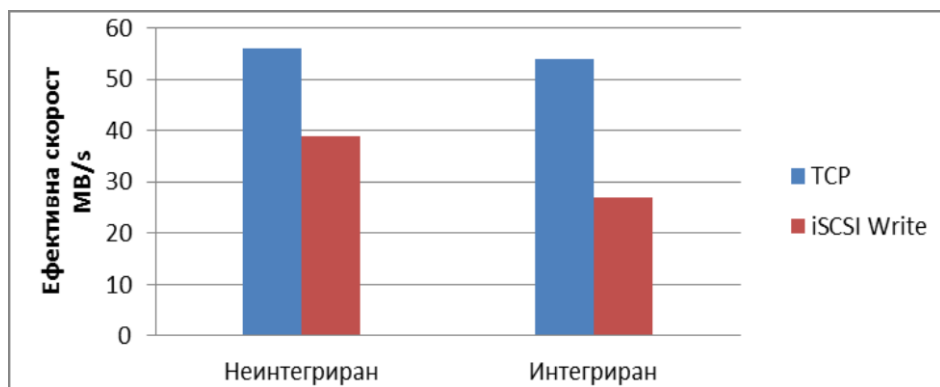
За да се определят предимствата и недостатъците в DCB протоколите, е важно да се разбере първо каква е производителността на различните потоци от трафик в интегрираната мрежа. Създаването на интегрирана мрежа дава възможност да се измери ефекта от загубите и на останалия трафик върху производителността на LAN и SAN трафика. Оценено е въздействието на задръстванията в интегрираната мрежа при нарастване на броя на различните класове трафик.

През първия етап на експериментите се използват SAN приемник и LAN1 приемник. На фигури 2, 3, 4 и 5 са визуализирани резултатите от експериментите. LAN трафикът в мрежата варира между TCP и UDP, докато SAN трафика варира между iSCSI READ и iSCSI WRITE трафик. Всяка графика изобразява пропускателната способност на един клас трафик, когато мрежата не е споделена и когато е споделена. От фигури 2 и 3 е видно, че TCP трафикът е с по-голям приоритет от SAN трафика.

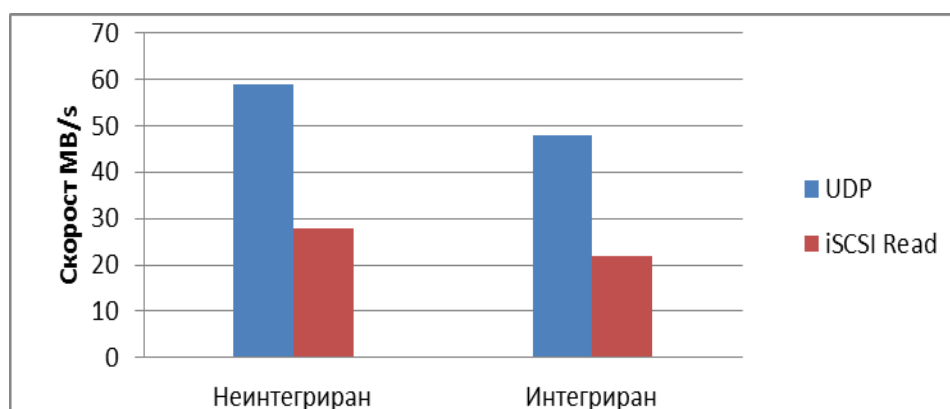
Докато пропускателната способност на SAN трафика и при четене, и при запис намалява в интегрираната мрежа, то пропускателната способност на TCP трафика остава непроменена.



Фиг. 2. Скорост на Интегрирани TCP и iSCSI Read Потоци



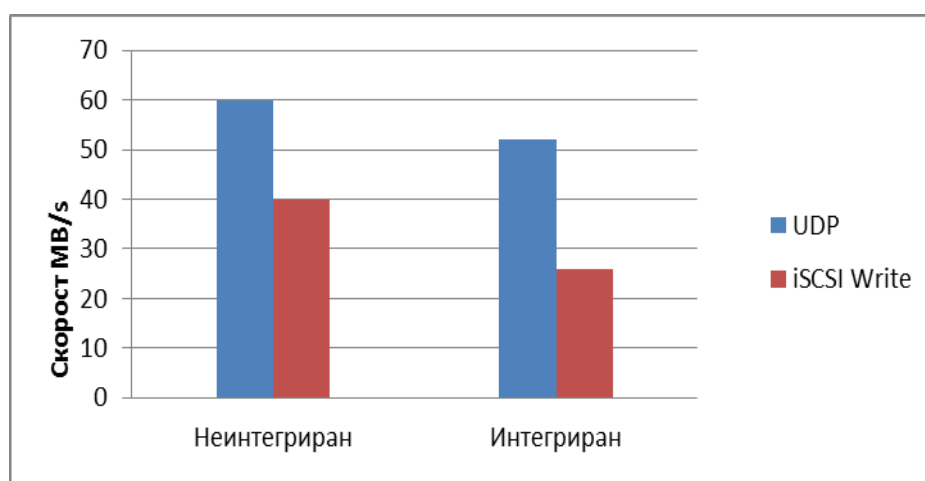
Фиг. 3. Скорост на Интегрирани TCP и iSCSI Write Потоци



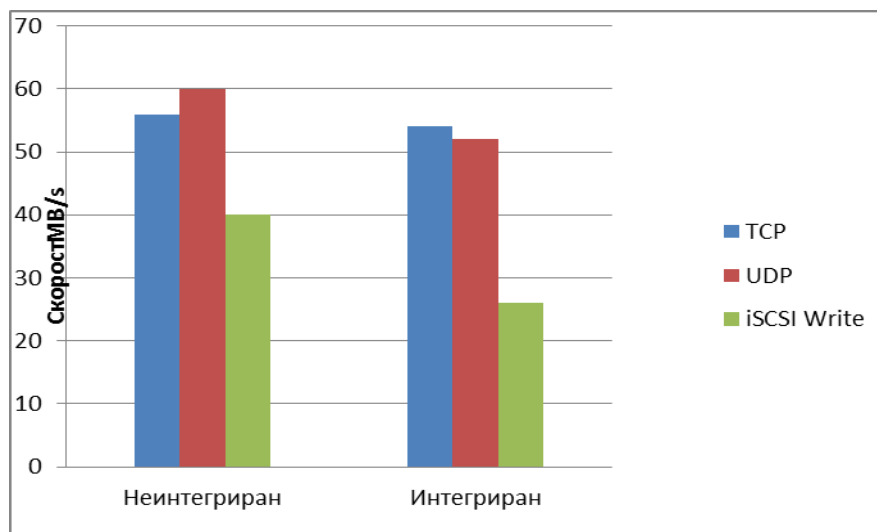
Фиг. 4. Скорост на Интегрирани UDP и iSCSI Read Потоци

Получените експериментални данни и построените графики доказват, че типът на трафика има пряк ефект върху производителността на класовете трафик в интегрираната мрежа.

Може да се приеме, че UDP и iSCSI Write трафиците са еднопосочни, докато TCP и iSCSI Read са двупосочни. Резултатите показват, че двупосочният трафик има по-висок приоритет от еднопосочния в интегрирана мрежа. Експериментите показват също, че подобните класове трафик се държат по подобен начин в интегрираната мрежа.

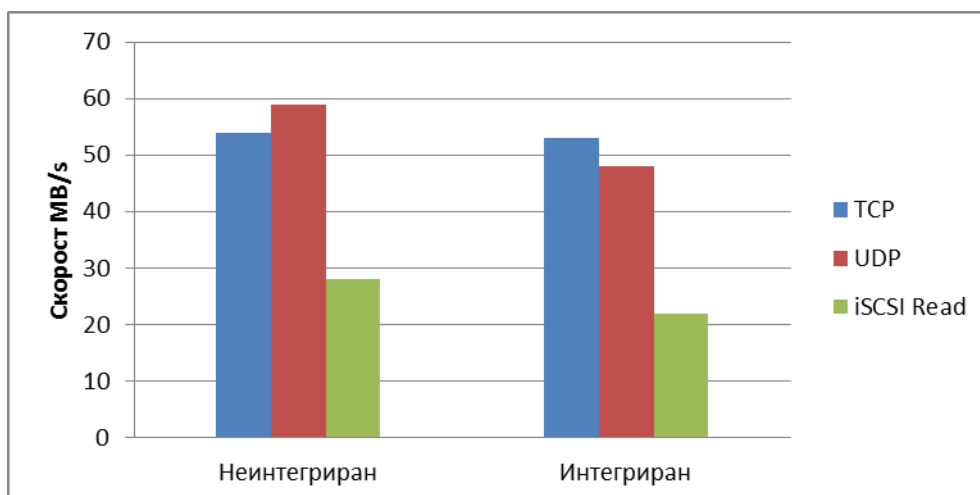


Фиг. 5. Скорост на Интегрирани UDP и iSCSI Write Потоци



Фиг. 6. Скорост на Интегрирани TCP, UDP и iSCSI Write Потоци

В този експеримент трафикът варира между TCP, UDP и SAN операции за четене и за запис. Фигури 6 и 7 визуализират ефекта от интегрираната мрежа върху пропускателната способност на трите класа трафик.



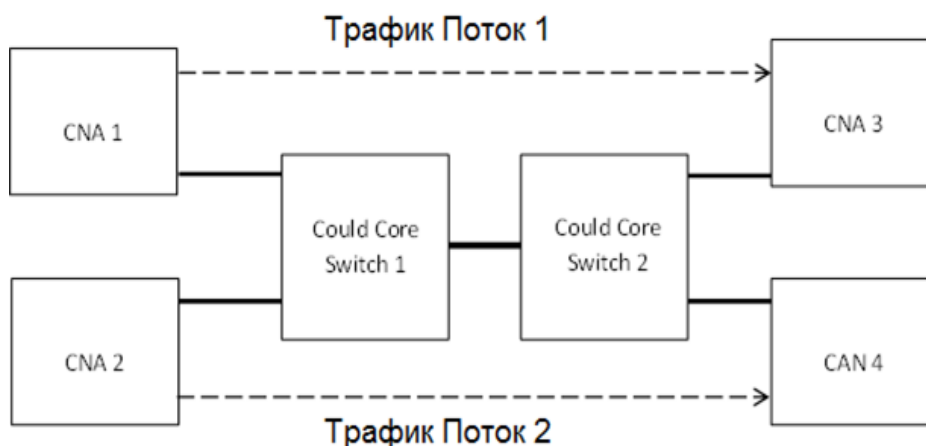
Фиг. 7. Скорост на Интегрирани TCP, UDP и iSCSI Read Потоци

Експериментът с три класа трафик потвърждава резултатите от експеримента с два класа трафик.

4. Анализ

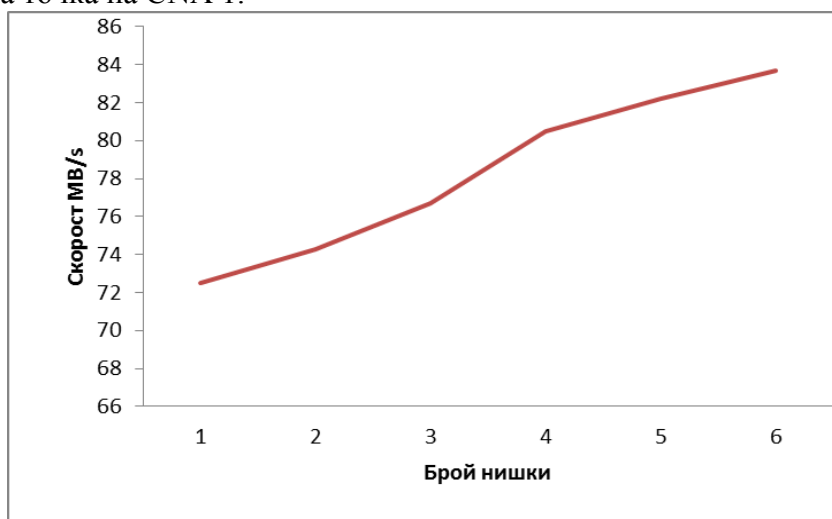
Резултатите показват, че скоростта на iSCSI Write потока намалява с 37%, когато споделя мрежата с потоци от TCP и UDP трафик.

Използването на хардуер с активирани DCB протоколи позволява да се анализира въздействието върху производителността на латентността и пропускателната способност на две най-често използвани приложения: съхранение на данни и междупроцесна комуникация. Експериментално е измерен ефектът от активиране и деактивиране на PFC протокола върху пропускателната способност на потока трафик. Експерименталната постановка е визуализирана на фигура 8.



Фиг. 8. Експериментална постановка

Първо е измерена базовата производителност при деактивирано PFC и стартиране на iperf с различни стойности на броя на нишките. При последващите експерименти всички потоци се подават към мрежата паралелно и е измерена производителността. Анализът на производителността в експериментите се базира на измерването на натоварването на всеки поток от гледна точка на CNA 1.



Фиг. 9. Скорост в зависимост от броя на нишките

През първия етап на експериментите, визуализиран на фигура 8, са използвани мрежови адаптери CNA1 и CNA3. На фигура 9 са показани резултатите от експериментите. PFC е деактивиран и се изпраща потребителски UDP трафик от CNA1 до CNA3 чрез iperf, а броят на нишките варира от една до шест и е измерена пропускателната способност.

Графиката показва, че с нарастване на броя на нишките нивото на производителност нараства линейно. Пропускателната способност е от порядъка от 70 MB/s до над 80 MB/s. При четири нишки пропускателната способност доближава границата от 1Gbit/s и при последващо нарастване на броя на нишките и при превключване на голям брой входно-изходни процеси води до стабилизиране на трафика на ниво от около 83 MB/s.

Литература

- [1] Hagen M., E. Varki, "Fiber channel arbitrated loop-performance analysis of loop tenancy overhead," in Networking, Architecture, and Storage, 2008. NAS'08. International Conference on, pp. 225-232, IEEE, 2008.

- [2] Feuser O., A. Wenzel, "On the effects of the IEEE 802.3x Flow control in full-duplex Ethernet lans," in Local Computer Networks, 1999. LCN'99. Conference on, pp. 160-161, IEEE, 1999.
- [3] Bestler C., L. Coene, "Applicability of Remote Direct Memory Access Protocol (RDMA) and Direct Data Placement (DDP)" tech. rep., RFC 5045, October,2007.
- [4] Pinkerton J., E. Delegates, "Direct Data Placement Protocol (DDP)/Remote Direct Memory Access Protocol (RDMA) Security", Self, 2007.
- [5] Recio R., P. Culley, B. Metzler, D. Garcia, J. Hilland, "A remote direct memory access protocol speciation", 2007.
- [6] Romanow A., J. Mogul, T. Talpey, S. Bailey, "Remote Direct Memory Access(RDMA) over IP Problem Statement," The Internet Society, 2005.
- [7] Bailey S., T. Talpey, "The architecture of direct data placement (ddp) and remote direct memory access (rdma) on internet protocols", Architecture, 2005.
- [8] Shah H., J. Pinkerton, R. Recio, P. Culley, "Direct data placement over reliable transports" tech. rep., IETF Internet-draft draft-ietf-rddp-ddp-04.(work in progress), 2005.
- [9] Jain R., D. Chiu, W. Hawe, A quantitative measure of fairness and discrimination for resource allocation in shared computer system. Eastern Research Laboratory, Digital Equipment Corporation, 1984.
- [10] Hagen M., "Fairness in a data center", University of New Hampshire Durham, NH, USA, 2012

За контакти:

Деян Стоев
 катедра „Информационни технологии”
 ВВМУ “Н. Й. Вапцаров”
 E-mail: dean@vizicomp.com

д-р Явор Дечев
 катедра „Информационни технологии”
 ВВМУ “Н. Й. Вапцаров”
 E-mail: dechco99@yahoo.com



БАЛАНСИРАНО РАЗПРЕДЕЛЕНИЕ НА ТРАФИКА В ИНТЕГРИРАНА ETHERNET СРЕДА (Част втора)

Деян Х. Стоев, Явор З. Дечев

Резюме: Използването на интегрирани мрежи за предаване на различните видове трафик налага да се изследва влиянието върху производителността на два или повече класа трафик, споделящи една и съща среда. В настоящата статия са разгледани основните алгоритми за балансиране на потоците трафик, направен е сравнителен анализ между DRR и DRR-AWC. Извършена е оценка на ефективността на DRR-AWC алгоритъма.

Ключови думи: балансиране на потоците трафик, DRR, DRR-AWC.

BALANCED TRAFFIC DISTRIBUTION IN AN INTEGRATED ETHERNET ENVIRONMENT (Part two)

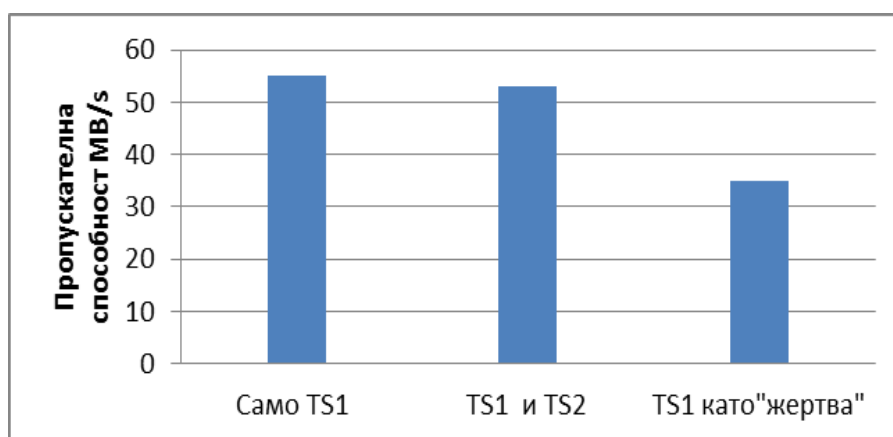
Dean H. Stoeff, Yavor Z. Dechev

Abstract: The usage of integrated networks for transfer of different types of traffic creates necessity to explore the impact on the system performance, caused by two or more traffic classes sharing the same media environment. In this article the basic algorithms for balancing the traffic flows are discussed and is performed a comparative analysis between DRR and DRR-AWC algorithms. An evaluation of the efficiency of the DRR-AWC algorithm is calculated.

Keywords: balanced traffic distribution, DRR, DRR-AWC

В този експеримент PFC е активиран на всички устройства в мрежата. Първо е измерен капацитетът между CNA1 и CNA3, стартирани самостоятелно, след това се стартира трафикът между CNA2 и CNA4. Точка на насищане (victim point), която е термин, даден от DCB работната група на IEEE, показва какво се случва с трафика между CNA1 и CNA3, когато трафикът между CNA2 и CNA4 е временно забавен. На фигура 10 е визуализиран ефектът от стартиране на нов трафик-поток върху първоначалния поток между CNA1 и CNA3.

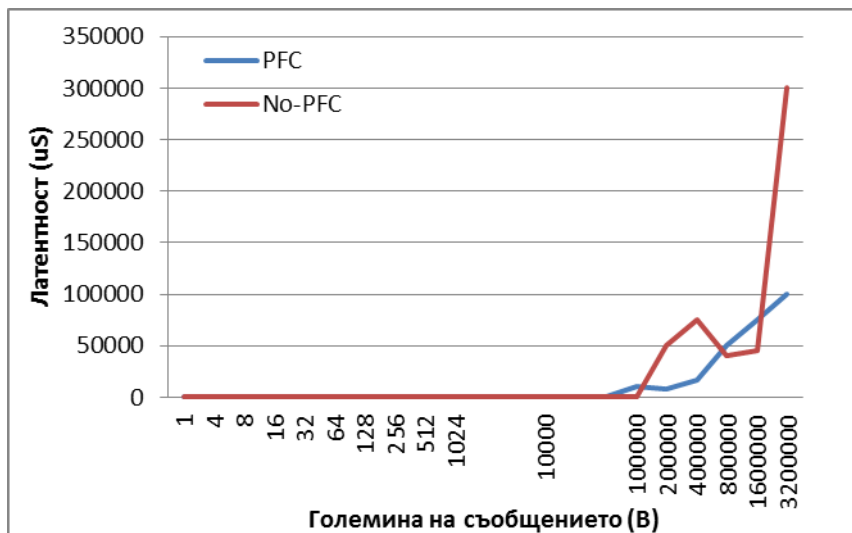
Измерването на пълния капацитет между две точки показва, че разпределението на използвания капацитет от потоците е минимално засегнат от стартирането на нов поток. Това показва, че една проста система с два потока може да бъде близка до това да е добре балансирана, използвайки само PFC.



Фиг. 10. Пропускателна способност при два потока спрямо CNA1

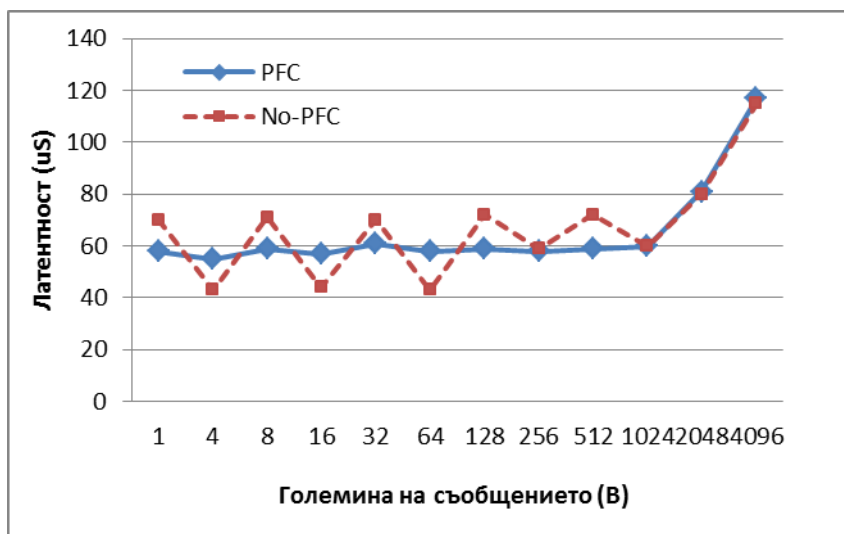
На всички системи е инсталиран Open MPI и се изпълнява Intel MPI Benchmarks. Латентността първо е измерена с деактивирано PFC, след това е измерена отново с активирано PFC. При деактивирано PFC, MPI разчита на TCP за контрол на потока.

Фигура 11 визуализира пълните резултати от тестовете. На фигура 12 са изобразени резултатите от същите тестове, но са използвани само резултати от съобщения с размер под 4096 байта, така че да може по-добре да се разгледат резултатите на съобщения с по-малък размер. На фигура 11 се вижда, че при съобщения с размер над 131 072 байта, латентността на мрежа с активирано PFC нараства пропорционално на размера на съобщението. Докато латентността на мрежа без PFC драстично нараства, дори и при малка промяна в големината на съобщението. Фигура 12 показва, че при мрежа с активирано PFC латентността е стабилна и равномерна с промяна на размера на съобщението, докато при изключено PFC е неравномерна и се колебае от размер до размер на съобщението.



Фиг. 11. MPI Exchange benchmark latency

Резултатите от MPI експеримента показват, че MPI може значително да се подобри чрез използване на PFC в мрежата. Двата резултата ясно демонстрират, че латентността е значително по-равномерна и стабилна, независимо от големината при наличието на PFC. Резултатите показват, че без PFC, MPI разчита единствено на TCP рамкиране (windowing) и предаване за контрол на потока. При възникване на TCP препредаване (TCP Retransmission),

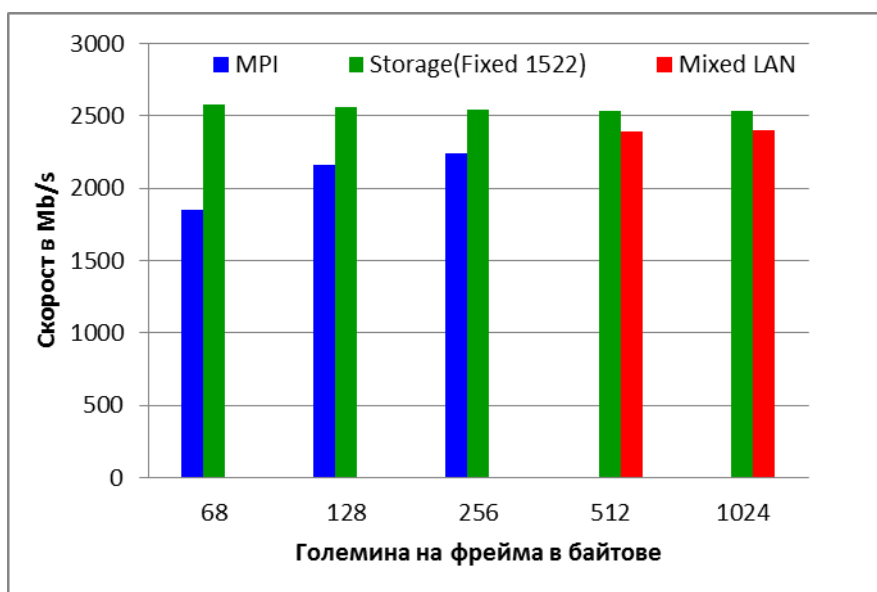


Фиг. 12. MPI Exchange benchmark latency (кратки съобщения)

латентността рязко се увеличава. При наличие на PFC в мрежата препредаването се избягва и латентността е много по-стабилна.

Комутаторите в съвременните центрове за данни прилагат DRR алгоритъма вместо традиционните претеглени опашки WFQ. DRR е лесен за внедряване и се доближава много до WFQ по отношение на „справедливостта. Алгоритмите за разпределение и теориите за опашките представляват балансиране между „справедливост“, производителност и простота на внедряването. Понастоящем много комутатори поддържат Weighted DRR, тъй като той е по-лесен за внедряване и осигурява голямо доближаване до нивото на справедливост в WFQ.

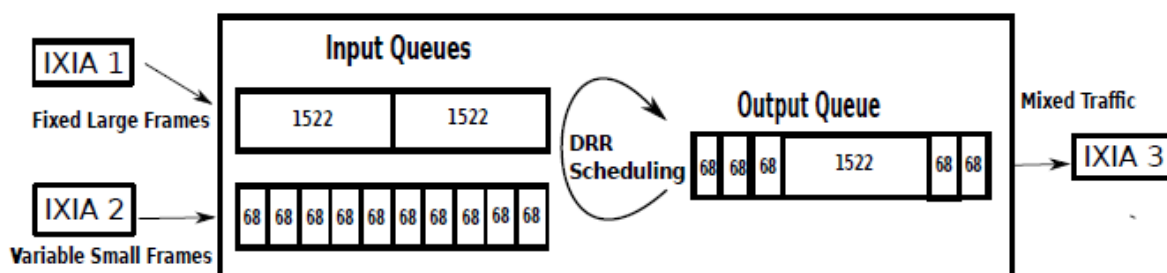
При голямо разнообразие от класове трафик е важно да се гарантира, че както потоците трафик от малки пакети, така и големите пакети ще използват мрежата „справедливо“. Фигура 13 изобразява фактическата производителност на фреймове с различни размери, интегрирани в един център за данни, получена при провеждането на експеримент. В най-крайния случай при 68 байта, най-малките MPI фреймове получават почти 25% по-малко от капацитета в сравнение с класа трафик за съхранение на данни.



Фиг. 13. Пропускателна способност на типичните класове трафик

За да бъде преодолян този недостатък, е приложена нова вариация на DRR алгоритъма, предложен от Хейгън [10].

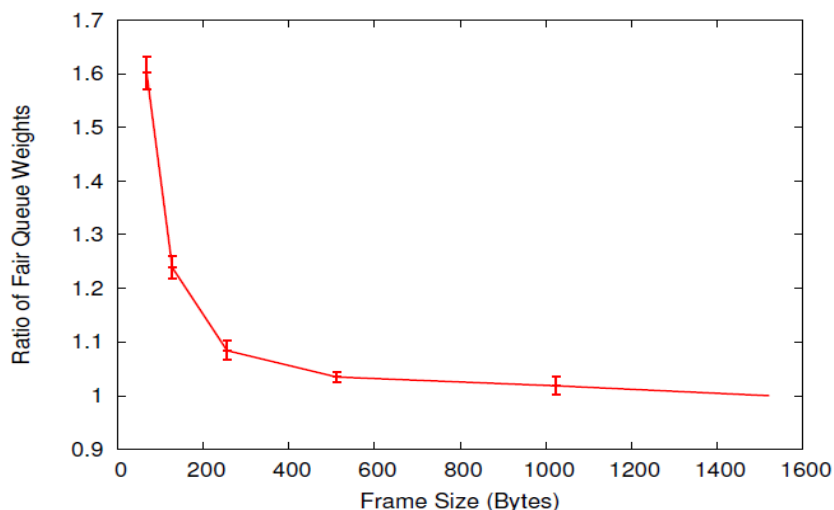
Експерименталната постановка е показана на фигура 14.



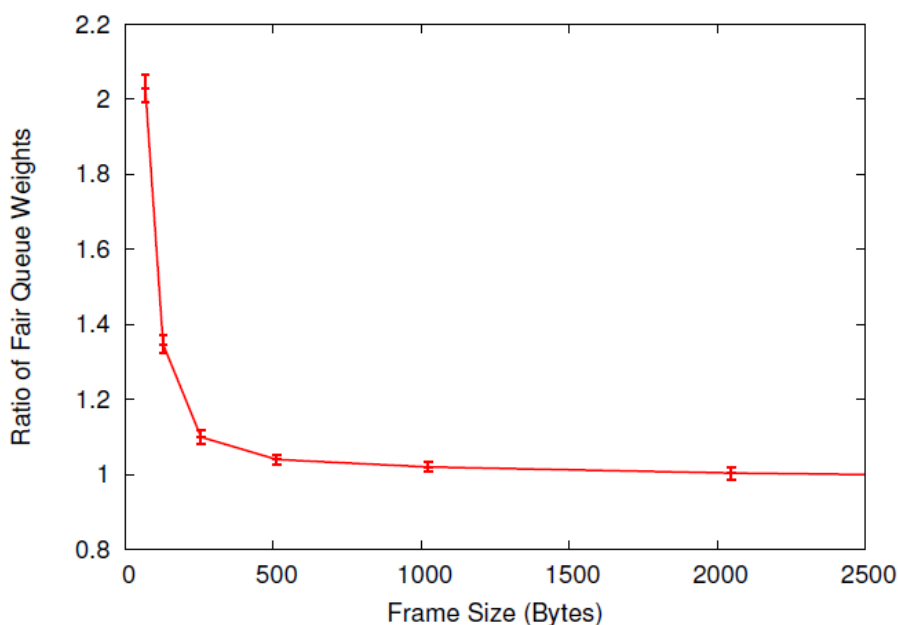
Фиг. 14. Експериментална постановка

За целите на експеримента са използвани 3 порта на тестовия хардуер. От Port1 към Port3 е генериран поток с големи фреймове (TS1) с размери на фрейма от 1522, 2500 и 5000 байта. Всяка от трите графики (фигура 15, 16 и 17) съответства на различния размер на фрейма в TS1. От Port2 към Port3 се генерира втори поток (TS2), който се състои от

фреймове с малки размери. При тази постановка и двата потока следва да се разпределят чрез DRR в порта на комутатора свързан към Port3 на тестовия хардуер.



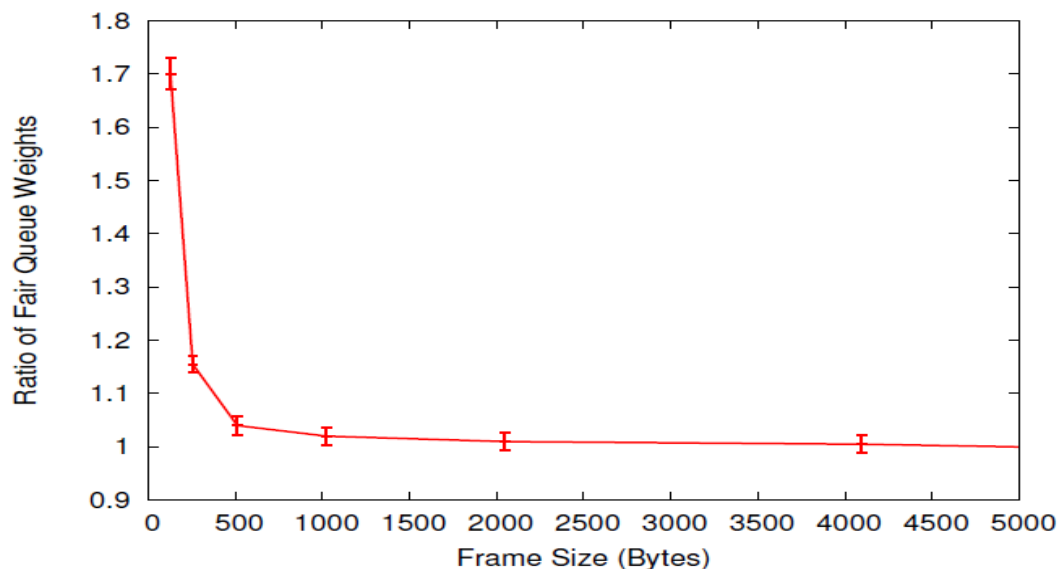
Фиг. 15. Тест 1 - Съотношение на теглата между TS1 и TS2 в опашката при WFQ и големина на фрейма 1522 байта



Фиг. 16. Тест 1 - Съотношение на теглата между TS1 и TS2 в опашката при WFQ и големина на фрейма 2500 байта

Целта на този експеримент е да се открият разпределенияте тегла на TS2, необходими да изравнят пропускателната способност на всеки поток трафик. Съотношението между теглото за TS2 към теглото на TS1 е изобразено на Y оста, а размерът на фрейма на TS2 е изобразен на X оста. Резултатите показват какво тегло е необходимо за всеки размер на фрейма, за да се постигне справедливо разпределение. На фигури 15, 16 и 17 се вижда как малките фреймове са много ошетени и са необходимо значително високи тегла, за да се изравнят с пропускателната способност на TS1.

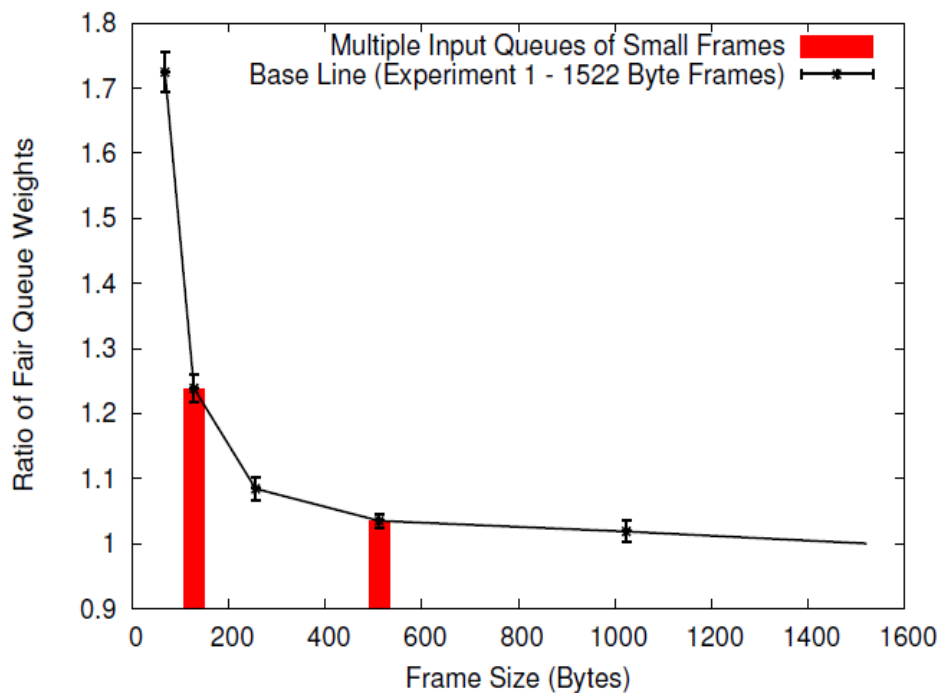
Когато TS2 достигне размер на фрейма между 200 и 500 байта, „ощетяването“ започва да намалява и бавно започва да се приближава към теглото на TS1. При по-малките фреймове теглото, необходимо за създаване на справедливо разпределение, е почти два пъти по-голямо от това на TS1.



Фиг. 17. Тест 1 - Съотношение на теглата между TS1 и TS2 в опашката при WFQ и големина на фрейма 5000 байта

Във втория етап на тестовете са изследвани множество входящи опашки от малки фреймове и към комутатора е свързан четвърти порт.

Вместо да се използва само една входяща опашка от малки фреймове и една входяща опашка от големи фреймове, са използвани две входящи опашки с различни размери (68, 128, 256, 512 и 1024 байта) заедно с една входяща опашка с големи фреймове. Резултатите от експеримента са визуализирани на фигури 18 и 19.

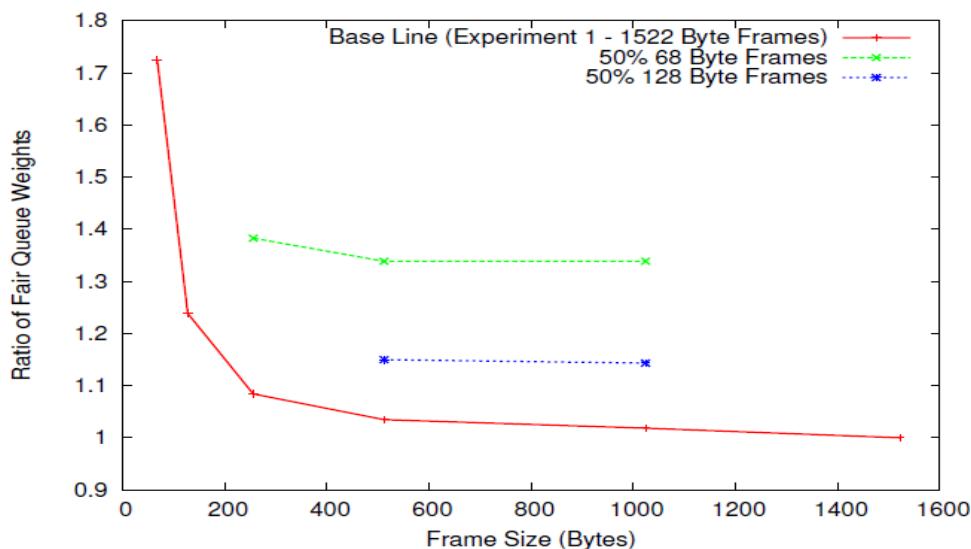


Фиг. 18. Тест 2 – Множество входящи опашки с различна големина на фрейма

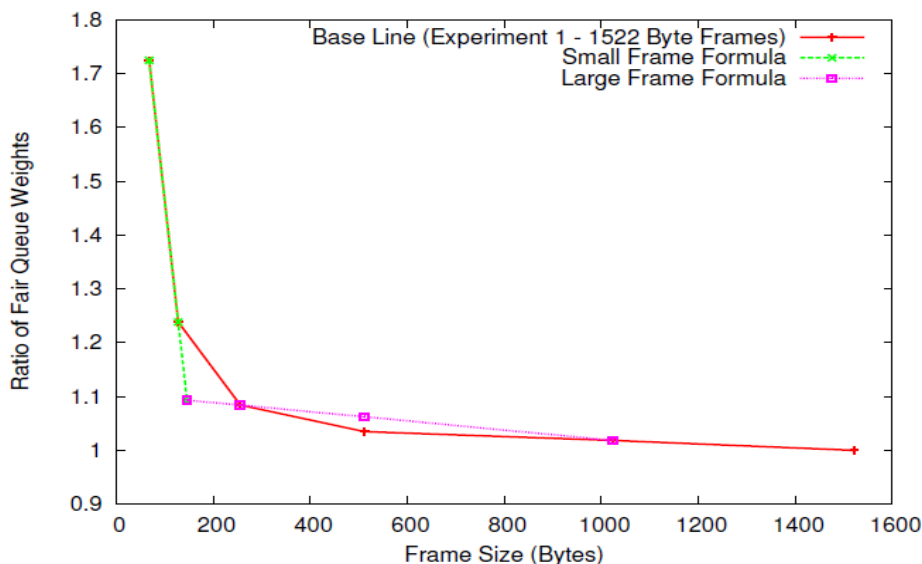
Независимо дали има една или множество опашки от малки фреймове, теглата, необходими за достигане на еквивалентна пропускателна способност с потока от големи фреймове, са еднакви.

Третият експеримент цели да установи как смесеният трафик в една опашка се отразява на резултатите. Създадена е една входяща опашка от големи фреймове, а за опашката с малки фреймове са избрани 50% от фреймовете да са с един размер (x байта) и 50% с друг размер (y байта). Променят се размерите на всеки от малките фреймове и резултатите са визуализират на фигура 19. На нея се вижда че смесения входящ трафик осреднява резултатите от предходните експерименти.

Резултатите от проведените тестове осигуряват достатъчно информация да се определи оптималното тегло за даден размер на фрейма. Предложеният алгоритъм се базира на статистическите броячи в комутаторите, които са сравнително универсални и следят броя на фреймовете с различни размери, които преминават, и ги групират по диапазони. Алгоритъмът взема тези статистики за всяка входяща опашка в комутатора и настройва DRR теглата на входящата опашка на база на дела на малките фреймове.



Фиг. 19. Тест 3 – Единична опашка и трафик със смесена големина на фрейма



Фиг. 20. Определяне на теглото за малките и големите фреймове с помощта на формулите (1) и (2)

Входящите опашки от по-малки фреймове получават тегла по-големи от тези на входящите опашки с големи фреймове. Това води до уеднаквяване на пропускателната способност на различните входящи опашки, като по този начин се гарантира

„справедливост“ за всички входящи опашки, независимо от средния размер на фрейма на всяка една от тях.

Формула (1) се отнася за потоци от трафик с фреймове по-малки от 146 байта, където X е среден размер на фрейма, а $f(X)$ е разпределеното тегло. Формула (2) се отнася за потоци трафик с фреймове по-големи от 146 байта. Тази големина от 146 байта е опорна точка, която осигурява най-малко отклонение от оригиналната крива и максимизира „справедливостта“, чрез използване на линейния метод. На фигура 20 са визуализирани резултатите от функциите за определяне на теглото за малки и големи фреймове по отношение на базовата линия от Тест1. Формулите са изведени, използвайки линеен метод, приложен върху резултатите от базовия експеримент.

$$f(x) = -0,008x + 2,276 \quad (1)$$

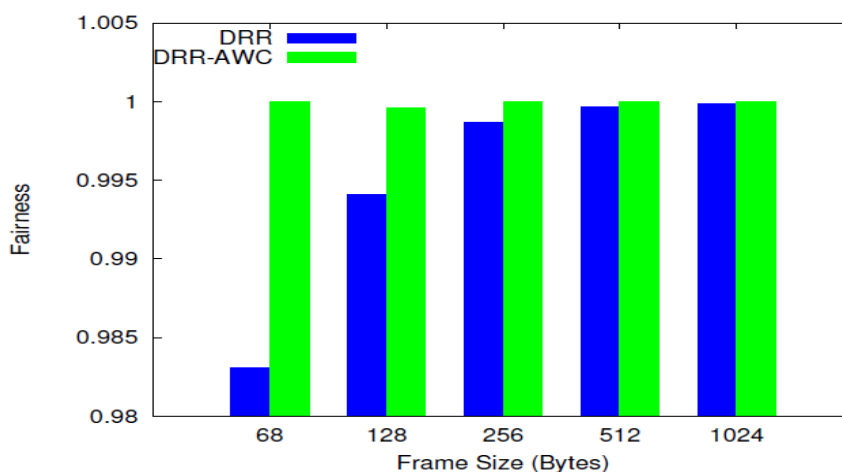
$$f(x) = -8,555 \cdot 10^{-5}x + 1,106 \quad (2)$$

Алгоритъмът започва с изчистване на статистиките за трафика, следва събиране на статистика за една минута и отново получаване на статистики. Това осигурява получаване на средна стойност за броя на фреймовете за всяка категория размер за една минута. След това алгоритъмът изчислява за всяка група отношение на теглото към общото тегло, базирано на броя на фреймове във всяка група и стойността на „справедливото“ тегло, получено чрез формули (1) и (2). Ако размерът е 146 байта или по-малък, се използва формула (1), в противен случай се използва формула (2). Накрая общото тегло се умножава по максималния размер на фрейм и теглото се задава за всяка входяща опашка в комутатора. По време на следващия цикъл на разпределение всяка входяща опашка се разпределя, базирана на новото тегло, което е изчислено.

За да се оцени ефектът от работата на DRR-AWC, се изпълняват предишните опити веднъж със стандартния DRR алгоритъм и втори път с модифицирания алгоритъм. За оценяване на подобрението на „справедливостта“ се използва индексът за справедливост на Джейн (Jain's Fairness Index) - формула (3). JFI изчислява стойността на „справедливост“ от едно върху N , където N е броят на потребителите до 1, където 1 е най-справедливо [9]:

$$J = \frac{(\sum_{i=1}^n x_i)^2}{n \sum_{i=1}^n x_i^2} \quad (3)$$

където x е пропускателната способност на дадена опашка, а n е броят на входящите опашки.



Фиг. 21. Сравнение на „справедливостта“ между DRR и DRR-AWC

На фигура 21 са визуализирани резултатите от този анализ. Стълбовете, означени с DRR, представляват JFI, когато комутаторът работи със стандартен DRR алгоритъм за разпределение. Стълбовете, означени с DRR-AWC, е изчислената справедливост, когато

комутаторът работи с модифицирания алгоритъм за разпределение. Може да се направи извод, че „справедливостта“ нараства значително в някои случаи от по-малко от 0,985 до над 0,999. JFI слага ударение върху недостига, така че стойностите не се различават много в целия обхват на индекса, а само от 0,985 до 0,999, което в реални условия представлява подобрене в пропускателната способност от почти 33%.

Заклучение

С използването на интегрираните мрежи анализът на производителността, различните класове трафик и протоколи, комбинирани в една интегрирана архитектура, става все по-необходим. Използвайки високоскоростна Ethernet мрежа в център за данни, това изследване разглежда влиянието върху производителността на два и на три класа трафик, споделящи една и съща среда.

При проведените експерименти в тази разработка се стигна до следните изводи:

- Влиянието на задръстването варира при различните класове трафик. Двупосочните класове трафик са по-малко засегнати от задръстването, отколкото еднопосочните класове. Двупосочните класове трафик използват мрежата по-ефективно, особено в случая на iSCSI Read, защото по-голямата част от трафика се връща по по-малко задръстената посока на мрежата.
- Без наличието на PFC, MPI системите разчитат на TCP рамкирането (windowing) и препредаване за контрол на потоците, които създават много неравномерни резултати в латентността при вариране на размера на съобщенията. При активиране на PFC общата латентност намалява и става много стабилна при всички размери на съобщенията.
- DRR-AWC не е много по-сложен от съществуващия алгоритъм за разпределение, но активно следи потоците трафик и настройва теглата на опашките, за да максимизира „справедливостта“. Чрез направените тестове и изчисления на JFI се доказва, че този алгоритъм е по-ефективен от DRR. Внедряването DRR-AWC в съвременните комутатори не би изисквало големи инвестиции и би довело до увеличение на производителността им при малки пакети с до 33%.

Литература

- [1] Hagen M., E. Varki, "Fiber channel arbitrated loop-performance analysis of loop tenancy overhead," in Networking, Architecture, and Storage, 2008. NAS'08. International Conference on, pp. 225-232, IEEE, 2008.
- [2] Feuser O., A. Wenzel, "On the effects of the IEEE 802.3x Flow control in full-duplex Ethernet lans," in Local Computer Networks, 1999. LCN'99. Conference on, pp. 160-161, IEEE, 1999.
- [3] Bestler C., L. Coene, "Applicability of Remote Direct Memory Access Protocol (RDMA) and Direct Data Placement (DDP)" tech. rep., RFC 5045, October, 2007.
- [4] Pinkerton J., E. Delegates, "Direct Data Placement Protocol (DDP)/Remote Direct Memory Access Protocol (RDMA) Security", Self, 2007.
- [5] Recio R., P. Culley, B. Metzler, D. Garcia, J. Hilland, "A remote direct memory access protocol speciation", 2007.
- [6] Romanow A., J. Mogul, T. Talpey, S. Bailey, "Remote Direct Memory Access(RDMA) over IP Problem Statement," The Internet Society, 2005.
- [7] Bailey S., T. Talpey, "The architecture of direct data placement (ddp) and remote direct memory access (rdma) on internet protocols", Architecture, 2005.

- [8] Shah H., J. Pinkerton, R. Recio, P. Culley, "Direct data placement over reliable transports" tech. rep., IETF Internet-draft draft-ietf-rddp-ddp-04.(work in progress), 2005.
- [9] Jain R., D. Chiu, W. Hawe, A quantitative measure of fairness and discrimination for resource allocation in shared computer system. Eastern Research Laboratory, Digital Equipment Corporation, 1984.
- [10] Hagen M., "Fairness in a data center", University of New Hampshire Durham, NH, USA, 2012

За контакти:

Деян Стоев
катедра „Информационни технологии”
ВВМУ “Н. Й. Вапцаров”
E-mail: dean@vizicomp.com

д-р Явор Дечев
катедра „Информационни технологии”
ВВМУ “Н. Й. Вапцаров”
E-mail: dechco99@yahoo.com



ЕДИН ВАРИАНТ ЗА ИНИЦИАЛИЗАЦИЯ НА МАРШРУТИЗАТОР ЗА MPP КОМПЮТРИ С DLH МРЕЖОВА ТОПОЛОГИЯ

Милен Г. Ангелов

Резюме: Изследванията на много автори са посветени на надеждността, отказоустойчивостта и възможността за реконфигуриране на комуникационните мрежи при паралелните компютри. Маршрутизаторите, които се явяват едни от основните функционални възли в мултипроцесорните системи, играят ключова роля за реализирането на тези качества. В тази връзка настоящата статия разглежда инициализацията в един маршрутизатор с Virtual Cut-Through управление на потока, предназначен за мултикомпютри с Double-Loop Hypercube мрежова топология. Предложен е вариант за изграждане на блок за инициализация и на неговия алгоритъм за работа.

Ключови думи: Маршрутизатор, Инициализация, Virtual Cut-Through(VCT), Flow Control Digit (флит)

A Variant for Initialization of a Router for MPP Computers with DLH Network Topology

Milen G. Angelov

Abstract: Much research has been dedicated to the reliability, fault tolerance and reconfigurability of parallel computer communication networks. Routers, some of the fundamental functional nodes in multiprocessor systems, play a critical role in realizing these goals. In this context, this paper focuses on flow initialization in a router with virtual cut-through control within a supercomputer network with a Double-Loop Hypercube topology. A method for designing such an initialization block is proposed along with its operating algorithm.

Keywords: Router, Initialization, Virtual Cut-Through (VCT), Flow Control Digit (flit)

1. Увод

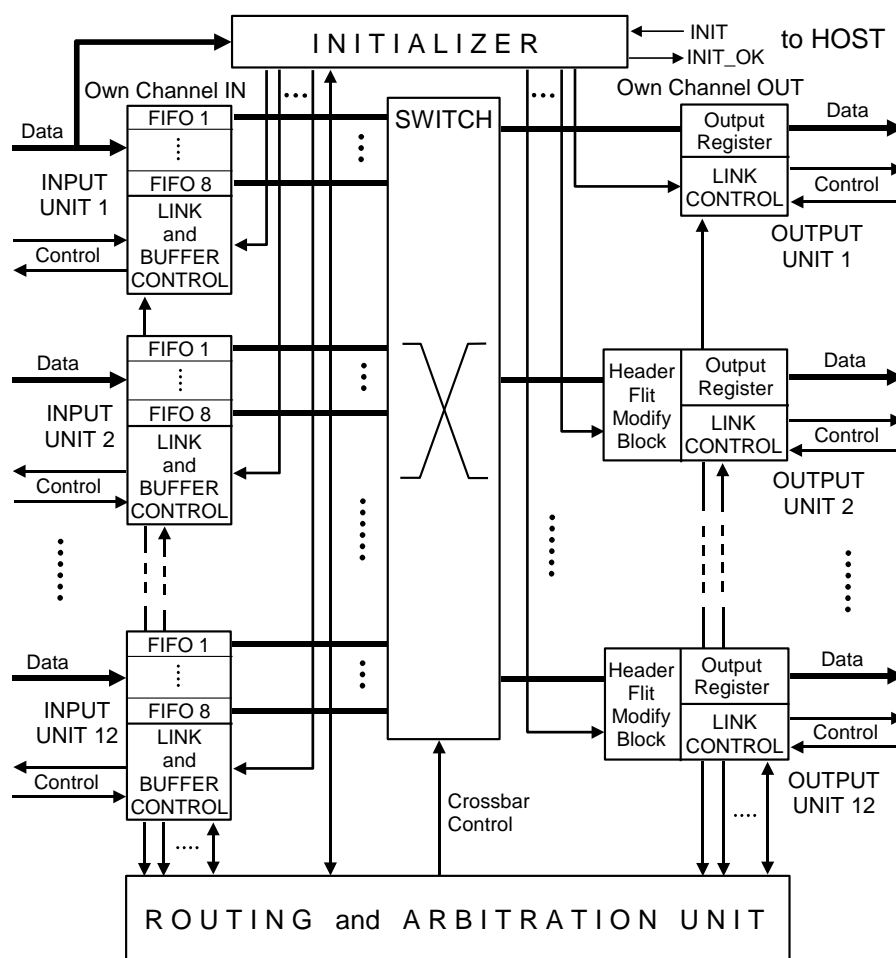
Инициализацията на маршрутизаторите, използвани при изграждането на паралелни компютри, е необходим и важен процес. Тя се реализира в определени моменти време от тяхната работа поради различни причини. Някои от тях са: първоначално стартиране на компютърната система, реконфигуриране на мрежовата топология, изключване на възли поради регистрирани проблеми, например дефектирали възли или линии за връзка и др. Процесът на инициализация е тясно свързан с архитектурата на маршрутизатора, с използваните блокове за формиране на данновите пътища, както и с тяхното управление. Поради тези зависимости за основа на предложения вариант за инициализация се използва структурата на маршрутизатор с Cut-Through управление на потока [3]. На тази база са описани отделните фази от инициализацията на този маршрутизатор, свързани с необходимата детайлизация относно регистри със специално предназначение, управляващи автомати и техните сигнали.

На фигура 1 е показана обобщената блокова схема на маршрутизатор за мултикомпютър със следните характеристики: 12 двупосочни канала с ширина 40 бита, входни буфери, всеки от които притежава пул с осем FIFO опашки [4], арбитраж при всеки от изходните канали, използване на минимална адаптивна маршрутизация, адаптирана към DLH мрежова топология [7]. Основните функционални блокове на схемата от фигура 1 са:

- **INITIALIZER:** Блок за инициализация на маршрутизатора. Блокът е свързан към хост процесора чрез данновите линии на входния канал към собствения възел;

- SWITCH: Crossbar комутатор. Логическа комбинационна схема, свързваща входните канали с изходите на маршрутизатора [2];

- INPUT UNIT: Един такъв блок обединява съвкупност от взаимосвързани модули, които реализират приемането на данни и управлението на един входен канал [1, 2]. Блокът включва: входен буфер с неговия пул от осем опашки, управляващите автомати на буфера и на всяка от опашките [8], логически схеми, входно/изходни буфери за сигнали и др.;



Фиг. 1. Обобщената блокова схема на маршрутизатор за мултикомпютър

- OUTPUT UNIT: Този блок също обединява съвкупност от взаимосвързани модули, които реализират предаването на данни и управлението за един изходен канал [1, 2]. В него са включени: изходен регистър, логически блок за модифициране на заглавния флит (с изключение на OUTPUT UNIT за собствения изходен канал), управляващ автомат на изходния канал и неговия арбитър [5], логически схеми, входно/изходни буфери за сигнали и др.;

- ROUTING and ARBITRATION UNIT: Блокът реализира разпределението на входовете и изходите след успешен арбитраж [5]. Включва входни и изходни арбитри;

- Сигнали: Връзките между отделните блокове са показани чрез два типа шини: даннови линии и управляващи линии. Поради сложността подробното описание на сигналите не е визуализирано. Самите сигнали са подробно описани на съответните места в [5, 6, 8]. Използваният начин за изобразяване е: групите даннови линии са означени с дебели линии, а групите управляващи линии - с по-тънки линии. Стрелките на линиите показват посоките на трансфер на данните и въздействието на управляващите линии. Конкретните сигнали за

свързване на изходен канал с входен буфер на два съседни маршрутизатора от една система, притежаващи описаната по-горе архитектура, са описани в [6].

Основната идея е инициализацията на един маршрутизатор да може да се извършва в произволни моменти от време, независимо от неговото текущо състояние и независимо от състоянието на маршрутизаторите в останалите възли.

2. Общо за блока за инициализация

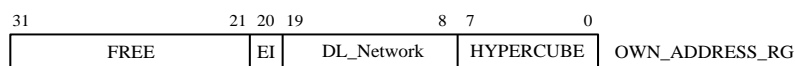
Блокът за инициализация е предназначен за конфигуриране поведението на маршрутизатора за конкретната мрежова топология и за следващи реконфигурации в следствие на последвали промени, свързани с условията по време на работа. Той дава възможност за запис и съхраняване на управляващи битове и полета, които въздействат върху работата на маршрутизатора. Първоначално всеки хост процесор на един възел обикновено получава конфигурационна информация през някакъв входно/изходен канал. Освен това, по време на работата на системата към хостовете може да се изпращат нови конфигурации с нормални пакети. Целта е чрез реконфигуриране да се запази цялостната стабилност в работата на системата, като се елиминират дефектирани възли и канали. Разбира се, системата трябва да е свързана в отказоустойчива мрежа.

Инициализацията на един маршрутизатор започва с установяване на активно високо ниво на сигнала INIT=HIGH, който се формира от хост процесора на възела (показан на фигура 1, фигура 3). Данните се предават през канала на собствения възел. Управляващите думи се записват в регистри за временно съхраняване, за да не се нарушат вече започнати трансфери на пакети. Всяка незапочнала маршрутизация на нормален пакет от момента на активиране на сигнала INIT се задържа. Пакетите, чийто трансфер е започнал, продължават да се предават към техните получатели. Хост процесорът задържа INIT=HIGH до края на инициализацията. Блокът за инициализация забранява всички входни и изходни буфери на маршрутизатора чрез сигнала PORT_DIS (показан на фигура 3), докато трае процесът на зареждане на управляващите думи от регистрите за временно съхраняване в съответните конфигурационни регистри. Краят на инициализацията се потвърждава към хост процесора чрез сигнала INIT_OK=HIGH (показан на фигура 1, фигура 3). Разрешават се входните и изходните буфери на маршрутизатора. Трансферът на пакети продължава с новите параметри.

3. Архитектура на блока за инициализация

Архитектурата на предложения блок за инициализация е показана на фигура 3(а). Той се състои от:

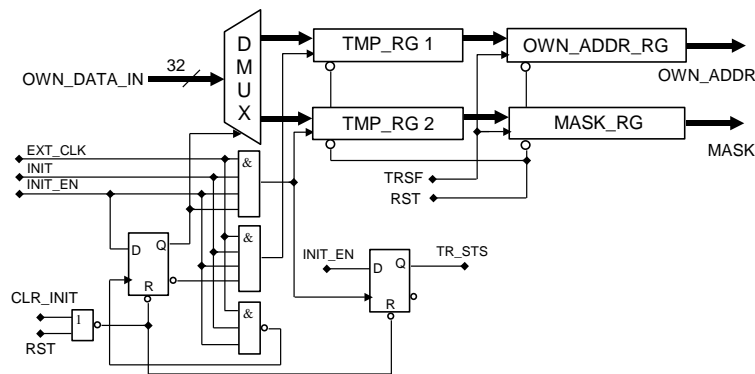
- OWN_ADDRESS_RG: 32-битов регистър, в който от $[D_{20}..D_0]$ се записва собствения 21-битов адрес OWN_ADR на възела от DLH мрежата. Форматът на този адрес е показан на фигура 2. Останалите 11 бита, от $[D_{31}..D_{21}]$ могат да се използват за бъдещи разширения. Подробно описание за отделните полета на адреса може да се намери в [4];



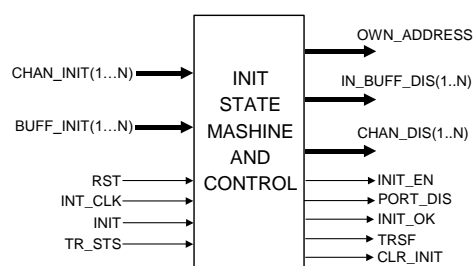
Фиг. 2. Формат на адреса

- TMP_RG_1 и TMP_RG_2: Група от два броя 32-битови регистри за временно съхраняване на информацията за инициализация;

- MASK_RG: 32 битов регистър, в който се записват битовете, разрешаващи работата на маршрутизатора. Регистърът е разделен на следните полета: INPUT_MASK – 12 бита, от [D₁₁..D₀], наречени IN_BUFF_DIS[n], n=1..12, чрез които индивидуално се разрешава или се забранява работата на всеки входен канал. Индивидуалните флагове IN_BUFF_DIS се подават към съответните управляващи автомати на входните буфери; OUTPUT_MASK – 12 бита, от [D₂₃..D₁₂] наречени CHAN_DIS[n], n=1..12, чрез които индивидуално се разрешава или се забранява работата на всеки изходен канал. Индивидуалните флагове CHAN_DIS се подават към съответните управляващи автомати на изходните канали; 8 бита, от [D₃₁..D₂₄] за бъдещи промени и разширения. Полетата INPUT_MASK и OUTPUT_MASK не са показани на фигура 3(а);
- DMUX: демултиплексор, чрез който регистрите TMP_RG_1 и TMP_RG_2 се комутират към входа за собствения възел на маршрутизатора;
- Логика и управление.



Фиг. 3 (а)



Фиг. 3 (б)

На фигура 3(б) е показан управляващият автомат на блока за инициализация. Предназначението на неговите входни и изходни сигнали е както следва:

RST: Входен асинхронен сигнал за начално установяване на маршрутизатора. Генерира се от хост процесора на възела, който обслужва маршрутизатора;

INT_CLK: Входен тактов сигнал за маршрутизатора, по който се синхронизира работата на всички функционални възли и на входните сигнали за управление на потока. Генерира се от локалния възел, който обслужва маршрутизатора;

INIT: Входен сигнал, генерира се от хост процесора на възела и служи за инициализация на маршрутизатора със стойности за регистри и флагове;

TR_STS: Вътрешен сигнал, показва край на записа в междинните регистри;

CHAN_INIT: Вътрешен сигнал за маршрутизатора, индициращ, че всички изходни канали са неактивни, където:

CHAN_INIT= \cap (CHAN_INIT[n]), n=1..12; Всеки от сигналите CHAN_INIT[n] се формира от управляващия автомат на съответния изходен канал, когато се намира в състояние S0 (Init);

BUFF_INIT: Вътрешен сигнал, индициращ, че всички входни буфери са неактивни.

BUFF_INIT= \cap (BUFF_INIT[n]), n=1..12; Всеки от сигналите BUFF_INIT[n] се формира от управляващия автомат на съответния входен буфери, когато се намира в състояние S0 (Init);

OWN_DATA_IN: Входни даннови сигнали [D₃₁..D₀] от собствения канал;

EXT_CLK: Входен сигнал, чрез който става запис на информация в регистрите на блока за инициализация на маршрутизатора. Генерира се от собствения изходен канал на възела;

OWN_ADDRESS: Сигналите са показани на фигура 3;

IN_BUFF_DIS[n]: Индивидуална забрана работата на всеки от входните буфери;

CHAN_DIS[n]: Индивидуална забрана работата на всеки от изходните канали;

INIT_EN: Разрешава запис в междинните регистри при инициализация;

INIT_OK: Потвърждава край на инициализация към хост процесора;

PORT_DIS: Забранява временно всички входни буфери и изходни канали, както и фазата на маршрутизация, докато трае процеса на инициализацията на регистрите и флагове;

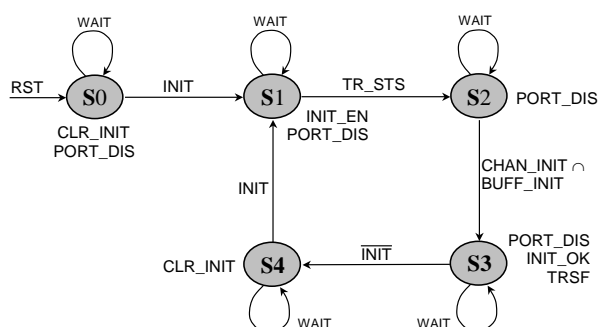
TRSF: Зарежда информацията от регистрите за временно съхраняване в конфигурационните регистри;

CLR_INIT: Нулира тригерите в блока за инициализация.

Трябва да се подчертае, че изходните сигнали, показани на схемата от Фиг.3(б), трябва да влияят върху работата на автоматите на всички входни буфери и на всички изходни канали на маршрутизатора. Описанията на тези автомати не са предмет на настоящата работа.

4. Управление на блока за инициализация

На фигура 4 е даден графът на състоянията на автомата, управляващ блока за инициализация. Състоянията, генерираните в тях сигнали и възможните преходи, са:



Фиг. 4. Граф на състоянията на автомата, управляващ блока за инициализация

S0: Начално състояние (Reset). Издават се сигналите **PORT_DIS** и **CLR_INIT**. В него се влиза след начално установяване чрез сигнала RST.

S0 → S1: Маршрутизаторът влиза в процес на инициализация след RESET;

S1: Първо състояние на инициализация (Init_1). Управляващите битове и полета се записват от хоста в регистрите за временно съхраняване в блока за инициализация. Генерират се сигналите **INIT_EN** и **PORT_DIS**.

S1 → S2: Маршрутизаторът е в процес на инициализация.

S2: Състояние на очакване (Wait_Chans_Transfer). Автоматът очаква да приключат всички започнати трансфери на пакети в маршрутизатора. Генерира сигнала **PORT_DIS**.

S2 → S3: Маршрутизаторът е в процес на инициализация, всички започнати трансфери на пакети са приключили.

S3: Второ състояние на инициализация (Init_2). Записва управляващите битове от регистрите за временно съхраняване в конфигурационните регистри. Издават се сигналите **PORT_DIS**, **INIT_OK** и **TRSF**;

S3 → S4: Маршрутизаторът е излязъл от процеса на инициализация;

S4: Неактивно състояние (Idle). Очаква се следваща инициализация. Генерира сигнала **CLR_INIT**.

S4 → S1: Маршрутизаторът влиза в процес на поредна инициализация.

5. Заключение

Представеният вариант за инициализация е приложим за маршрутизатори с определена архитектура, която е представена в т.1. Решението за инициализация на един маршрутизатор се взема от хост процесора на възела, който той обслужва. Блокът за инициализация е описан

достатъчно подробно и дава ясна представа за неговите функции и управление на ниво регистри и сигнали. Във връзка с разгледаната тематика са направени следните изводи:

- Чрез инициализация на маршрутизаторите се предоставя възможност за реконфигурация на системната комуникационна мрежа на един паралелен компютър във възможните граници на използваната мрежова топология.
- Основно предимство за работата на една система се явява възможността за изключване на дефектирани възли и линкове от системата.
- Инициализацията може да се извършва, без да се спира работата на системата като цяло. В маршрутизаторите не се губят пакетите, чийто трансфер е започнал преди момента на началото на инициализацията.
- При добавяне на възможност за оптимизация управлението на захранването за отделните модули (например индивидуално управление на входните и изходните звена) може да се намали консумираната от маршрутизатора енергия. Един от най-лесно реализуемите начини за това е да се минимизира консумираната енергия в маскираните поради някакви причини входни буфери и изходни канали. Методите за намаляване на консумираната мощност от маршрутизатора не са предмет на настоящата работа.

Литература

- [1]. Duato J., S. Yalamanchili, L.Ni. "Interconnection networks. An Engineering Approach", Morgan Kaufmann, San Francisco, CA, 2003. ISBN 9780585457451.
- [2]. Dally W., B. Towles. "Principles and Practices of Interconnection Networks", Morgan Kaufmann Press, San Francisco, 2004.
- [3]. Angelov M. "Routers for MPP Computers, Using Direct Communications Networks", John Atanasoff Society of Automatics and Informatics, International Conference AUTOMATICS AND INFORMATICS'2014 October 1-3, 2014, Sofia, Bulgaria, ISSN 1313-1869
- [4]. Angelov M., N. Ruskova. "Packet Transfer in DLH Networks", IOSR Journal of Computer Engineering (IOSR-JCE), e-ISSN: 2278-0661, p-ISSN: 2278-8727, Vol.19, Issue 5, Ver. II (Sep.-Oct. 2017), pp. 08-13
- [5]. Ангелов М. „Арбитраж на пакети в маршрутизатор за MPP компютри с DLH мрежова топология”, Трета научна конференция с международно участие "Компютърни науки и технологии", 25-26 Септември, 2015, Варна, България, ISSN 1312-3335, Бр.1/2015, стр. 38-45
- [6]. Ангелов М. „Един вариант за Cut-Trough управление на потока в маршрутизатор за MPP компютри”, Трета научна конференция с международно участие "Компютърни науки и технологии", 25-26 Септември, 2015, Варна, България, ISSN 1312-3335, Бр.1/2015, стр. 46-54
- [7]. Ангелов М. “Маршрутизация на пакети в MPP компютри с DLH мрежова топология”, Втора научна конференция с международно участие "Компютърни науки и технологии", 26-27 Септември 2014, Варна, България, ISSN 1312-3335, Бр.1/2014, стр. 64-69
- [8]. Ангелов М. “Структура и управление на буфер за входен канал на маршрутизатор за MPP компютри”, Втора научна конференция с международно участие "Компютърни науки и технологии", 26-27 Септември 2014, Варна, България, ISSN 1312-3335, Бр.1/2014, стр. 71-76

За контакти:

ас. Милен Г. Ангелов
катедра „Компютърни науки и технологии”
Технически университет-Варна
E-mail: angelovmg@tu-varna.bg

ИЗСЛЕДВАНЕ НА 4G КЛЕТЪЧНИ БЕЗЖИЧНИ МРЕЖИ

Айдън М. Хъкъ

Резюме: Тази статия разглежда тенденциите на развитие на клетъчните безжични мрежи. През последните години клетъчните мрежи по света се развиват в посока 5G, докато в България все още са актуални 4G технологиите. Докладът представя сравнителна характеристика за развитието на клетъчните мрежи в Европа, Северна Америка и България.

Ключови думи: 4G, 5G, клетъчни мрежи

STUDY OF 4G CELLULAR WIRELESS NETWORKS

Aydan M. Haka

Abstract: This paper consider examines the development trends of cellular networks. In recent years, cellular networks around the world have grown to 5G, while 4G technologies are still up to date in Bulgaria. The paper presents comparative features for the development of cellular networks in Europe, North America and Bulgaria.

Keywords: 4G, 5G, cellular networks

1. РАЗВИТИЕ НА КЛЕТЪЧНИТЕ БЕЗЖИЧНИ МРЕЖИ

В началото на 2012г. организацията ITU-R стартира проекта „IMT-2020 and beyond“ (International Mobile Telecommunication), която цели стартирането на 5G мрежите по света през 2020г. и след това. През 2013г. Европейската комисия подписа проекта 5G-PPP по програма „Horizon 2020“, който подкрепя развитието на 5G мрежите в Европа. През септември 2015г. ITU-R финализира визията си за 5G мобилните мрежи, която се изразява в използване на допълнителен спектър в подкрепа на бъдещето развитие на IMT.

Предимствата на клетъчните безжични мрежи спомагат за популяризирането им. Те започват да се развиват с високи темпове в Европа и Северна Америка. Съгласно публикувани прогнози [1, 2] до 2020г. броят на абонатите, използващи клетъчни безжични технологии, ще нарасне, като делът на LTE ще се утрои (фигура 2).



Фиг. 2. Прогнозен тренд за безжични мобилни технологии

Според [3, 4] до средата на 2017г. в Европа има 457 милиона уникални мобилни абонати, което представлява 84% уникална свързаност на населението, с почти 4% над

Северна Америка, където към юни 2017г. абонатите са над 290 милиона. Като се има предвид това ниво на развитие, в Европа наближава насищането на общия адресируем пазар, което ограничава пространството за бъдещ растеж на абонатите, като се очаква да бъдат добавени до 12 милиона потребители до 2020г. с годишен темп на растеж 0.7%. В Северна Америка за същия период се очаква абонатите да нараснат с 35 милиона. В Европа този растеж се очаква да бъде съсредоточен в по-големите мобилни пазари, където все още няма голямо насищане. Мобилните оператори тук инвестират сериозно на разгръщането на LTE мрежата, което довежда до 4G покритие, надхвърлящо 95% от населението в началото на 2017г., докато в Северна Америка 4G покритието на населението достига 99%.

Развитието на клетъчните мрежи не приключва с 4G технологиите. Паралелно с 4G се развива и 5G стандартът. Основната цел на 5G е осигуряване на всеобхватна свързаност на всякакъв вид устройства и приложения, които може да са полезни в мрежата. Освен това тези мрежи няма да се базират на една специфична технология за безжичен достъп [5].

Според [3] глобалното усвояване на 5G ще отнеме повече време от 4G, поради по-бавното внедряване на мрежата и несигурността в предложените цени в сравнение с LTE. Освен това все още няма глобално дефинирана спецификация за 5G. Според [6] възможностите на 5G мрежите трябва да надхвърлят тези на предишните поколения за мобилна комуникация. Тези възможности включват: мощен системен капацитет, много високи даннови скорости на предаване, много ниска латентност, ултрависока надеждност и наличност, много ниска цена на устройствата и подобрена енергийна ефективност, както е показано на Таблица 1.

Таблица 1. Изисквания към 5G мрежите

Скорости	над 10Gbps в домашна среда, около 100Mbps в градска и крайградска среда, от поне 10Mbps почти навсякъде, включително в слабопоселените селски райони
Латентност	от 1ms за поддържане на критични спрямо латентността приложения, като на практика латентността може да бъде по-голяма
Висока надеждност и достъпност	загубата на свързаност и отклонението от QoS трябва да бъдат изключително редки
Много ниски разходи за устройства и енергийна консумация	много ниска цена на 5G устройствата и много дълъг живот на батерията от няколко години
Честотен диапазон	в [7] според Европейската комисия за 5G комуникация в Европа ще се използва диапазона от 700MHz; според FCC диапазона от 600MHz може да се използва за 5G в САЩ

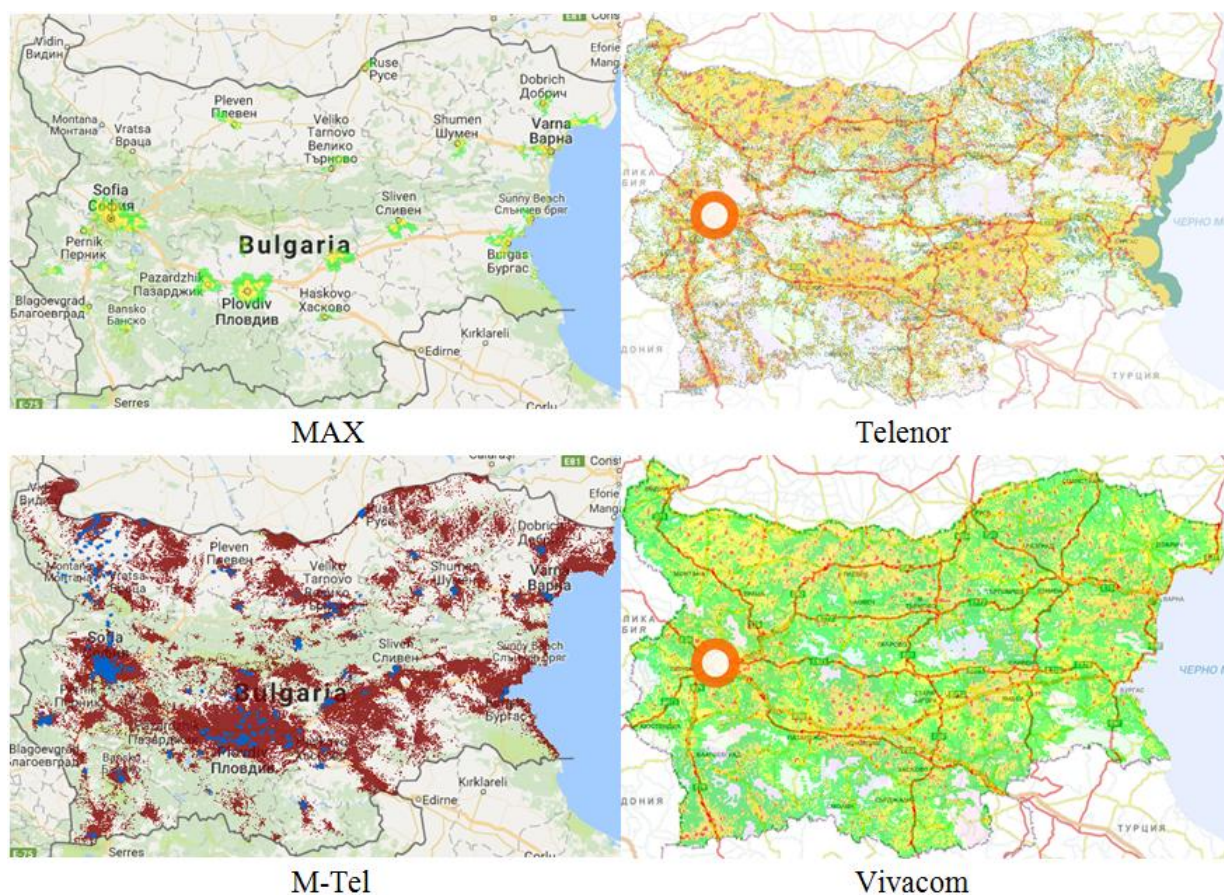
Според [8, 9] телеком операторите по света, а и в Европа, вече тестват мрежи от пето поколение, очаква се те да бъдат стартирани през 2020г. Една от причините за бавното развитие на 5G стандарта е, че няма изграден единен стандарт за технологията [9], като 3GPP работи върху създаването на първия 5G стандарт, който се очаква да излезе през 2018г. По време на стартирането на мрежите от пето поколение 4G мрежите ще бъдат актуални и широко използвани, тъй като според [3] до 2020г. 4G връзките ще завземат 3G, като ще се развиват и увеличават за бъдещо разрастване. Това става ясно от факта, че операторите съсредоточават инвестициите си върху подобряване на производителността на мрежата.

2. КЛЕТЪЧНИТЕ МРЕЖИ В БЪЛГАРИЯ

На 14.07.2016г. FCC одобрява честотен обхват за 5G от 28 GHz, 37GHz, 39GHz, а от 2017г. Европейската комисия освобождава за 5G честотния обхват от 700MHz. През юли

2017г. България подкрепи плана за 5G свързаност на Европа, според който ще се работи за осигуряване на радиочестотен спектър за най-новото поколение безжични мрежи [10, 11]. Постепенното навлизане на концепцията IoT и 5G мрежите кара телеком операторите да се стремят да подобряват своите мрежи. Това подобряване и обновяване в България обаче, става сравнително бавно, тъй като извършването на подобренията се диктуват от използването на Интернет услугите от клиентите. Според направената евростатистика в [12] България е на едно от последните места между държавите членки в Европейския съюз по отношение на активното използване на Интернет. Поради тази причина телеком операторите в страната все още поддържат 2G и 3G стандарта.

В България се развиват и двата основни 4G стандарта – WiMAX и LTE. През 2017г. най-известните телеком доставчици на мобилни услуга са Vivacom (БТК и Vivotel), Telenor (GloBul), M-Tel (Mobiltel) и Max (Max Telecom). През годините различни оператори като Nexcom България, Carrier BG, Max, Trans Telecom и M-Tel внедряват WiMAX мрежата като предполагат, че това е печелившата 4G технология и увеличават значително размерите на тези мрежи. От тях единствено Max поддържа за по-дълъг период WiMAX мрежата, като през 2015г. прехвърля всички свои WiMAX клиенти към LTE [13]. Телеком операторите постепенно се отказват от тази технология, тъй като световната тенденция е насочена към използването на LTE стандарта.



Фиг. 2. Интерактивни карти на разпространение на телеком операторите в България

MAX стартира първи 4G LTE мрежа в България през май 2014г., като реализира покритие на 60% от градското население. На 1 декември 2015г. Telenor стартира своята LTE мрежа в честотния спектър 1800MHz. Последните телеком оператори, които стартират своите LTE мрежи, са Vivacom и M-Tel през 2016г. В същата година LTE мрежата на Vivacom покрива 86.75% от населението [14]. M-Tel реализира покритие на 99.99% от населението и 99.51% от територията на страната [15]. През 2017г. LTE мрежата на Telenor

покрива 95% от магистралите, 95% от първокласните пътища и 77% от второкласните пътища в страната, като реализира покритие на 89.06% от населението [16].

От интерактивните карти [17] на разпространение на 4G мрежите на четирите оператора се вижда обаче, че разгърнатите мрежи обхващат основно големите градове и гъсто населените места, като покритият процент от населението е основно в градовете, където комуникационният сигнал е с най-високи скорости. До по-слабо населените селски райони и трудно достъпни места според картите или не е реализирано покритие, или наличното покритие е с ниски скорости на предаване.

Тенденцията на развитие на 4G мрежите в големите градове и гъсто населените места се диктува от по-голямата необходимост от мрежови услуги на тези места, поради многобройното население в тях в сравнение със селските райони [19], както и по-доброто заплащане за изисквана услуга.

В рамките на сравнително кратък период от време в България се развиват мрежи от четвърто поколение, но въпреки това, на фона на Европа и света, у нас 4G технологиите навлизат и се развиват бавно. От темпа на развитие на мобилните мрежи в страната може да се заключи, че по време на вече стартираните 5G мрежи в Европа и другите водещи световни пазари през 2020г., в България ще бъдат все още актуални 4G мрежите.

ЗАКЛЮЧЕНИЕ

Въпреки че тенденцията на развитие на клетъчните мрежи е насочена към 5G, паралелно се развива и разширява и 4G технологията. Според направената сравнителна характеристика между водещите лидери при клетъчните мрежи в света, необходимостта от осигуряване на високи скорости и непрекъсваемост на предоставяните услуги неизбежно ще доведе до мигриране към 5G технологиите по цял свят. Тази миграция обаче, няма да започне по-рано от 2020г. Според направената сравнителна характеристика за България това мигриране към 5G ще се забави още повече, тъй като текущите телекоми залагат основно на 4G LTE технологията, като концентрацията на тези услуги основно е в големите градове и гъсто населените места.

Литература

- [1]. Mobile Network Statistics for 2016, 2016, <https://www.thinksmallcell.com/Opinion/mobile-network-statistics-for-2016.html>
- [2]. The Mobile Economy 2017, <https://www.gsma.com/mobileeconomy/#>
- [3]. The Mobile Economy Europe 2017, <https://www.gsma.com/mobileeconomy/europe/>
- [4]. The Mobile Economy North America 2017, <https://www.gsma.com/mobileeconomy/northamerica/>
- [5]. 5G Radio Access, 2016, Ericsson, <https://www.ericsson.com/assets/local/publications/white-papers/wp-5g.pdf>
- [6]. 5G Radio Access, April 2016, <https://www.ericsson.com/assets/local/publications/white-papers/wp-5g.pdf>
- [7]. 5G Spectrum Public Policy Position, November 2016, <https://www.gsma.com/spectrum/wp-content/uploads/2016/06/GSMA-5G-Spectrum-PPP.pdf>
- [8]. Mobile Future Spectrum Frontiers, July 31, 2017, http://mobilefuture.org/wp-content/uploads/2017/08/Mobile-Future-Spectrum-Frontiers-Ex-Parte-Letter_7.31.17.pdf
- [9]. 5G PAN-EUROPEAN TRIALS ROADMAP VERSION 1.0, May 23, 2017, https://5g-ppp.eu/wp-content/uploads/2017/05/5GInfraPPP_TrialsWG_Roadmap_Version1.0.pdf
- [10]. EU flagship initiative, the 5G declaration signed by EU Ministers in Tallinn, July 18, 2017, <https://www.mkm.ee/en/news/eu-flagship-initiative-5g-declaration-signed-eu-ministers-tallinn>

- [11]. Министър Ивайло Московски: Ще работим за 5G свързаност, 18.07.2017г., <https://www.mtict.government.bg/bg/category/1/ministur-ivaylo-moskovski-shte-rabotim-za-5g-svurzanost>
- [12]. Digital economy and society statistics – enterprises, March 2017, http://ec.europa.eu/eurostat/statistics-explained/index.php/Digital_economy_and_society_statistics_-_enterprises
- [13]. Макс мигрира всички свои клиенти от WiMAX към 4G LTE, <http://maxtelecom.bg/bg/zanas/novini/maks-migrira-vsichki-svoi-klienti-ot-wi-max-kym-4g-lte-operatoryt-dobavi-2-novi-grada-s-4g-pokritie>
- [14]. Годишен интегриран доклад 2016, <https://www.vivacom.bg/web/catalog/sustainable/files/assets/basic-html/index.html#1>
- [15]. Защо да изберем М-Tel за партньор, <https://www.mtel.bg/why-mtel>
- [16]. 4G LTE мрежата, <http://www.telenor.bg/bg/4g-lte-network?ref=home>
- [17]. Защо да изберем М-Tel за партньор, <https://www.mtel.bg/why-mtel>
- [18]. Интерактивни карти на разпространение, <https://www.telenor.bg/bg/coverage-map>, <http://maxtelecom.bg/map>, <https://www.mtel.bg/fix-internet-pokritie>, <https://www.vivacom.bg/bg/residential/ceni-i-uslugi/mobilni-uslugi/pokritie/interaktivna-karta>
- [19]. Население по области, възраст, местоживее и пол, <http://www.nsi.bg/bg>

За контакти:

ас. Айдын М. Хъкъ
катедра „Компютърни науки и технологии“
Технически университет – Варна
e-mail: aydin.mehmed@tu-varna.bg



ПРОУЧВАНЕ НА ИНТЕРФЕЙСИ ЗА УПРАВЛЕНИЕ НА НОСИМИ УСТРОЙСТВА

Юри Н. Димитров

Резюме: Проучването разглежда и категоризира направените до момента проучвания в областта на интерфейсите за управление на носими устройства (в частност смарт часовници), които имат за цел да разрешат проблема с малкия размер на устройствата от една страна, и човешките фактори от друга, които затрудняват използването на стандартния към момента метод за управление - използване на чувствителен на допир дисплей. Проучването показва насоките в които могат да продължат проучванията в тази област.

Ключови думи: носими устройства, смарт часовници.

Research of Computer-Human Interfaces for Wearable Devices

Yuri N. Dimitrov

Abstract: This paper evaluates and classifies the latest researches on Computer-Human Interfaces for Wearable devices (particularly Smartwatches). These researches are meant to solve the small device factor against human's factors problem which effects the touch screen interfaces usability. The research shows also future works areas.

Keywords: wearable, smartwatch.

1. Увод

Носимите устройства и в частност смарт часовниците (към които от гледна точка на еднаквост на интерфейсите могат да бъдат причислени и т.нар. фитнес тракери) навлизат все по-масово в употреба. Техните малки размер водят до редица проблеми, свързани с управлението им - функциите, които изпълняват, са много повече в сравнение със стандартните часовници, а това налага изграждането на сложни интерфейси за управление от страна на потребителите. Тенденцията за механично пренасяне на методи за управление от устройства с по-голям дисплей и размери - смарт телефони и таблети, върху смарт часовниците допълнително затруднява потребителите в процеса на използването им [1], като много от тези интерфейси не отчитат понякога спецификата на устройствата - размер, форма [2], условия на средата, в която се използват [1]. Използването на малки чувствителни на допир дисплеи, управлявани с помощта на пръсти, често води до скриване от погледа на потребителя на бутоните и друга информация, визуализирана на дисплея.

В момента има носими устройства, чиито минимални размери не се определят вече от размера на електронните компоненти или дори и на батерията, а от размера на площта, необходима за осъществяване на интерфейс между човека и устройството. Докато всички компоненти на устройствата могат да бъдат постоянно намалявани като размери, то областите върху чувствителните на допир дисплеи, върху които трябва да се програмират екранни бутони, слайдъри и други елементи, както и физическите механични или чувствителни на допир бутони няма как да бъдат намалявани, поради непроменяемите характеристики на човешката ръка [3].

Целта на настоящето проучване е да се събере информация за направените до момента изследвания в областта на интерфейсите между потребител и смарт часовник и да се формулират насоки за последващи проучвания в тази област. Проучени са само интерфейси,

които решават описания по-горе основен проблем с размера на устройствата, и които не са намерили все още своето място в масовото произвеждане на смарт часовници. Масово използваните средства за управление на интерфейси на смарт часовници - механични или чувствителни на допир бутони, въртящи се коронки, чувствителните на допир дисплеи и гласови команди не са предмет на настоящето проучване. Проучени са основно проучвания, публикувани през последните четири години.

2. Интерфейси на носими устройства

Проучените интерфейси са категоризирани спрямо това върху коя част на носимите устройства се въздейства и с какво средство се извършва въздействието. Те не са категоризирани спрямо конкретни функционалности на устройствата, които трябва да се управляват.

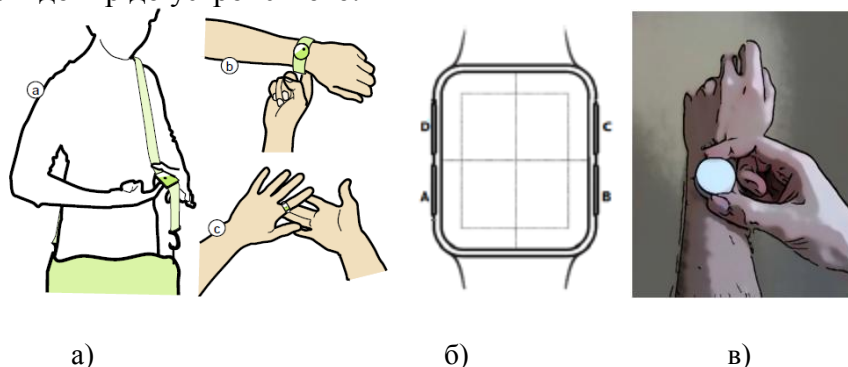
Интерфейси върху устройствата

Един от първите подходи за решаване на проблема с малкия размер на смарт часовниците и носимите устройства като цяло е проучване и разработване на концепции, базирани на използването на други части на устройствата като повърхности за взаимодействие, които са различни от чувствителен на допир дисплей и бутони.

Такъв пример е използването на задната (противоположната на дисплея) страна за управление на функциите на носимите устройства [4], показано на фигура 1а. Въпреки добрата прецизност и възможностите за визуална и тактилна обратна връзка след изпълнената операция, този подход не е удачен за управление на часовници, тъй като при стандартното им използване като ръчен часовник, прихванати с каишка около китката на потребители, до задната им страна няма достъп.

Друг подход при използването на други части на устройството за управление е използването на страничните му повърхности. За разлика от задната част, при часовниците тези повърхности са достъпни до потребителя и същите са били използвани и се използват за управление на стандартните ръчни електронни часовници [1]. Изследваните методи са най-различни, но най-приложимите са управление, чрез оказване на натиск върху чувствителни на натиск сензори върху страничните повърхности [5], показано на фигура 1б, или посредством чувствителни на допир сензори [6], показано на фигура 1в.

Един от основните проблеми, описани в проучванията на тези две технологии е наличието на вероятност тези интерфейси да бъдат активирани неволно, вследствие на оказан натиск или допир до устройството.

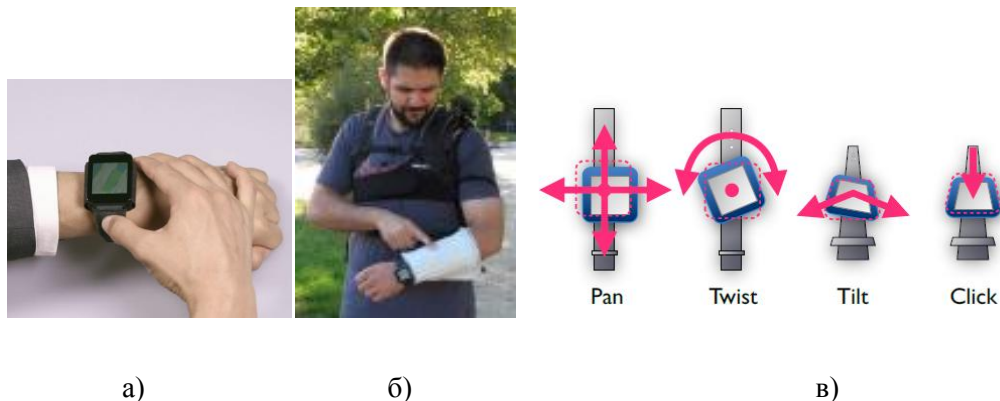


Фиг. 1. Интерфейси върху устройствата - а) задната страна [4]; б) от страни с натиск [5]; в) от страни с чувствителни на допир елементи [6]

Като разширяване на наличната повърхност за осъществяване на интерфейси между човек и смарт часовник, са проучвани и възможностите за използване на по-горе описаните технологии върху каишката [7], показано на фигура 2а, която е неделима принадлежност на

всеки ръчен часовник, както и върху други повърхности, до които потребителят има достъп, и в които е възможно интегрирането на интерфейси, чувствителни на допир или натиск - например ръкав на дреха [8], показано на фигура 2б.

Наличието на каишка при ръчните часовници, която е сравнително фиксирана върху китката на потребителя, позволява и използването на преместването на устройството спрямо каишката да бъде използвано за изграждането на интерфейси [9], показано на фигура 2в. Тази технология би могла да се разшири като се измерват приложените сили върху устройството в точките му на контакт с каишката.

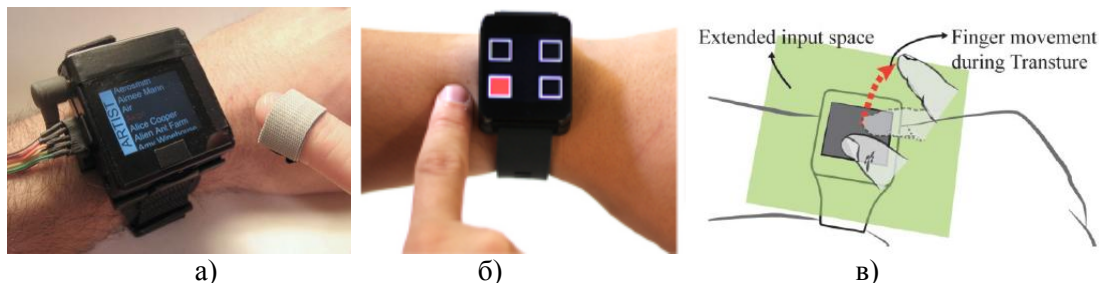


Фиг. 2. Интерфейси върху устройствата - а) върху каишка [7]; б) върху друг обект [8]; в) с преместване спрямо каишката [9]

Жестови интерфейси

Извършени са редица проучвания, които целят да се разработят интерфейси, които да позволят да се управляват носими устройства с помощта на ръцете на потребителя, но без да се докосва устройството. Жестовите интерфейси най-общо могат да се разделят на две категории - жестове със свободната ръка (ръката, на която не се намира часовника) и жестове с основната ръка (ръката, на която се намира часовника)

Жестовите със свободната ръка имат за цел да се извършват движения в близост до устройството, като технологично регистрирането на жестовите от страна на устройството може да става визуално, с помощта на камера, радиолокационно, акустично, на база промени в магнитното поле [10], показано на фигура 3а, електростатично [11], показано на фигура 3б. Жестовите могат да бъдат извършвани в различна отдалеченост спрямо устройството [12]. С цел по-прецизно извършване на жестовите, част от тях може да се извършват и върху опорни повърхности - например върху основната ръка, в близост до устройството [11]. Има проучвания и на жестове, които са продължение във въздуха на операция, започната върху чувствителния на допир дисплей на устройството [13], показано на фигура 3в.



Фиг. 3. Жестови интерфейси - а) магнитно поле [10]; б) електростатично [11]; в) с продължаване на движение, започнало върху дисплея [13]

Жестове с основната ръка са многократно и задълбочено проучвани, базират се на вградените в устройството акселометри, жirosкопи, компас и се използват масово в произвежданите в момента устройства, затова няма да бъдат предмет на това проучване. Само трябва да се спомене, че те имат едно основно предимство - може да се оперира само с една ръка, но и недостатъци, свързани с лесното предизвикване на нежелано активиране на интерфейса.

Биомеханични интерфейси

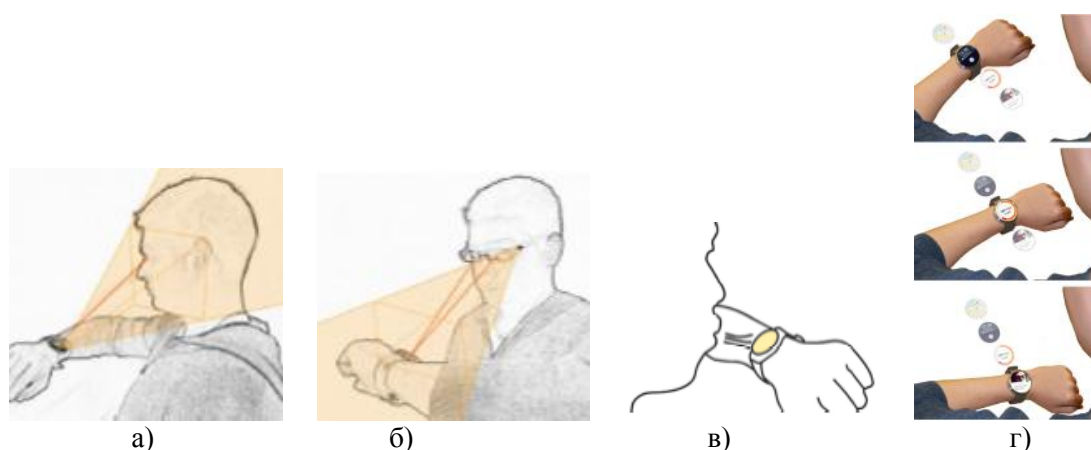
Това е група интерфейси, които използват части от човешкото тяло или негови функционални възможности за управление на носими устройства.

Интерфейсите с проследяване на погледа работят с камера, която регистрира позицията и движението на очите на потребителя, като на база местоположение на погледа и продължителност на задържане на погледа (втренчване) се изгражда интерфейс за избор на меню [14] [15], показано на фигура 4а. За да работят с приемлива скорост и висока надеждност, тези интерфейси трябва да се комбинират с друг интерфейс за потвърждаване на избора на селектирания с поглед елемент от менюто [14]. Възможно е да се регистрира погледа и на база разположение на смарт часовника спрямо друго очила, на които има монтирана камера [15], показано на фигура 4б.

Друг проучен подход е електромиографично измерване на мускулната активност и изграждане на интерфейси, базирани на подаваните сигнали [16]. Това е една от технологиите, които позволяват управлението на устройството само с една ръка - водещата. Проучванията показват обаче два пъти по-висока вероятност от грешка и три пъти по-голяма вероятност от непредизвикано сработване на интерфейса, в сравнение с работата с чувствителен на допир дисплей.

Blowwatch [17], показано на фигура 4в, е проучване и концепция за управление на носими устройства с помощта на въздушен поток, генериран от човек в процеса на издишване. Този интерфейс позволява управление на устройство без ръце, като има възможност за генериране на различни действия посредством различна честота и продължителност на издишванията. Недостатък е ниската му шумозащитеност.

Измерването на близост на устройството до части от човешкото тяло също може да се използва за управление [18], показано на фигура 4г.



Фиг. 4. Биомеханични интерфейси - а) поглед с камера на смарт часовника [15]; б) поглед с камера върху очила [15]; в) въздушен поток [17]; г) близост на устройството до тялото [18]

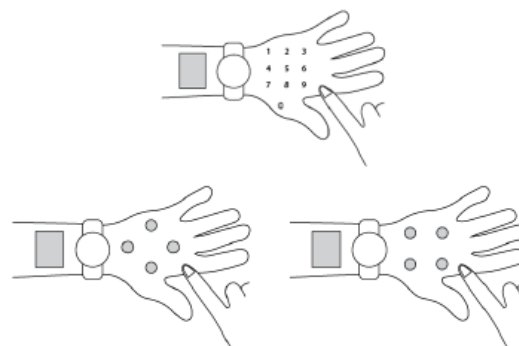
Прожекционни интерфейси

Това са интерфейси, които се базират на разширяване на повърхността за взаимодействие с устройството посредством прожектиране на виртуални бутони и/или

повърхности за виртуални чувствителни на допир дисплеи и съответно отчитане на взаимодействията на потребителя с тези виртуални средства за управление. Предвид разположението на смарт часовниците на ръката на потребителя, нормалната повърхност, върху която се проектират виртуалните бутони е ръката на потребителя [19], показано на фигура 5а. Тези интерфейси позволяват проектираните бутони, като функция, разположение и брой, да са и контекстно зависими от приложението или функциите на устройството, които се очаква да бъдат управлявани във всеки един отделен момент [20], показано на фигура 5б. Надеждността на тези интерфейси при нарастване на броя на виртуалните елементи, според направените изследвания [20], не е достатъчна за масовото им прилагане в момента.



а)



б)

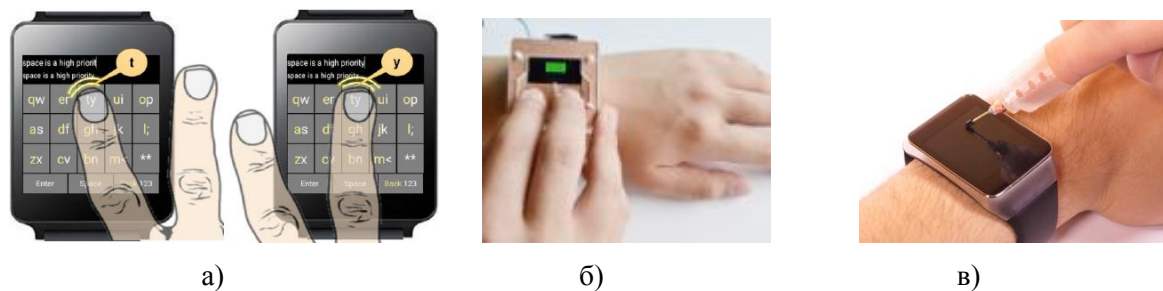
Фиг. 5. Проежекционни интерфейси - а) виртуални бутони [19]; б) контекстно зависим интерфейс [20]

Управление на менюта

Друга насока, в която се извършват редица проучвания, свързани с решаването на проблема с ограничените възможности за традиционни интерфейси за носими устройства, базирани на използване на чувствителен на допир дисплей и бутони, е управлението им да става с методи, различни от осъществяване на контакт с един пръст. Един от подходите е потребителя да използва различни пръсти за взаимодействие с устройството в едни и същи области, като се отчита с кой пръст е осъществен контакта, и на база това се изпълнява различна интерфейсна команда [21], показано на фигура 6а. Недостатъците на този метод са, че се изисква време и усилия от страна на потребителя да научи и привикне с интерфейса.

Друг проучен подход е изграждане на интерфейс, който отчита почуквания върху устройството, които потребителя извършва с различна честота и брой пръсти [22], показано на фигура 6б. Проучени са и методи, използващи допълнителни средства, различни от пръстите на потребителя, за взаимодействие с устройството посредством чувствителния на допир дисплей - например писалка [23], показано на фигура 6в. Недостатък на тази технология е необходимостта от съхранение и използване на допълнителните средства, при използване на устройството или конкретни негови функции.

Модифицирането на менютата на смарт часовниците с цел увеличаване на размера на бутоните и намаляването на броя натискания, както и използването на натиск с няколко пръста едновременно, също се проучва, като подход за решаване на проблемите, свързани с управлението на малки устройства [24]. Тези модификации също изискват от потребителите време и усилия, за да научат правилната последователност на използването им.



Фиг. 6. Управление на менюта - а) с различни пръсти [21]; б) с почукване [22]; в) с писалка [23]

Не на последно място трябва да се отбележи, че от критична важност освен задаването на команда от потребителя към устройството, е и обратната връзка, която той получава в потвърждение на това, че действието, което е изпълнил, е било получено от устройството [25].

3. Заключение

Въпреки непрекъснатите проучвания, повечето от които са проведени в рамките на последните четири години, все още не е установено и наложено в индустрията ефективно и ефикасно решение на проблема с малкия размер на чувствителните на допир дисплеи на смарт часовниците.

Основните проблеми, върху които трябва да се извършат проучвания, са

- функционалността на интерфейсите - доколко те са подходящи за управление на всички функции на устройствата;
- ергономичността - работата с интерфейсите не трябва да създава допълнително натоварване у потребителя;
- обратната връзка - тъй като не всички интерфейси могат да осигурят обратна връзка на база извършената операция, то трябва да се извършат проучвания в тази област;
- надеждността - висока успеваемост на изпълнение на операциите от първия опит и минимална вероятност от неволно активиране на интерфейса;
- социалната възприемчивост - потребителят трябва да се чувства комфортно при използване на интерфейса и да не предизвиква дискомфорт в обкръжаващите го хора - проучвания показват например, че различните по амплитуда жестове при жестовите интерфейси, показват различна степен на нежелание у някои от потребителите да ги използват на публични места [12];
- енергоемкостта - поради малкия размер и съответно капацитет на батериите на носими устройства, от критична важност е новите интерфейси да не консумират значително по-голямо количество електрическа енергия и съответно да съкращават значително цикъла на използване на устройствата.

На база извършеното проучване, интерфейсите, върху които е подходящо да се извършат допълнителни проучвания, са тези върху други повърхности на устройството, жестовите интерфейси, подобряване на използваните в момента интерфейси и най-вече - на комбинирането на различните технологии.

Литература

- [1] K. Lyons, "What can a dumb watch teach a smartwatch?," *Proc. 2015 ACM Int. Symp. Wearable Comput. - ISWC '15*, pp. 3–10, 2015.
- [2] D. L. Ashbrook, K. Lyons, and T. Starner, "An investigation into round touchscreen wristwatch interaction," *Proc. Int. Conf. Hum. Comput. Interact. with Mob. Devices Serv. - MobileHCI'08*, p. 311, 2008.

- [3] K. A. Siek, Y. Rogers, and K. H. Connelly, "Fat Finger Worries: How Older and Younger Users Physically Interact with PDAs," pp. 267–280, 2005.
- [4] P. Baudisch and G. Chu, "Back-of-device interaction allows creating very small touch devices," *Proc. 27th Int. Conf. Hum. factors Comput. Syst. - CHI 09*, no. c, p. 1923, 2009.
- [5] R. Darbar, P. K. Sen, and D. Samanta, "PressTact: Side Pressure-Based Input for Smartwatch Interaction," *Proc. 2016 CHI Conf. Ext. Abstr. Hum. Factors Comput. Syst. - CHI EA '16*, pp. 2431–2438, 2016.
- [6] I. Oakley and D. Lee, "Interaction on the Edge: Offset Sensing for Small Devices," *CHI '14 Proc. SIGCHI Conf. Hum. Factors Comput. Syst.*, pp. 169–178, 2014.
- [7] Y. Ahn, S. Hwang, H. Yoon, J. Gim, and J. Ryu, "BandSense: Pressure-sensitive Multi-touch Interaction on a Wristband," *Proc. 33rd Annu. ACM Conf. Ext. Abstr. Hum. Factors Comput. Syst. - CHI EA '15*, pp. 251–254, 2015.
- [8] S. Schneegass and A. Voit, "GestureSleeve: using touch sensitive fabrics for gestural input on the forearm for controlling smartwatches," *Proc. 2016 ACM Int. Symp. Wearable Comput. - ISWC '16*, pp. 108–115, 2016.
- [9] R. Xiao, G. Laput, and C. Harrison, "Expanding the input expressivity of smartwatches with mechanical pan, twist, tilt and click," *Proc. 32nd Annu. ACM Conf. Hum. factors Comput. Syst. - CHI '14*, pp. 193–196, 2014.
- [10] C. Harrison and S. E. Hudson, "Abracadabra: wireless, high-precision, and unpowered finger input for very small mobile devices," *Proc. 22nd Annu. ACM Symp. User interface Softw. Technol.*, pp. 121–124, 2009.
- [11] J. Zhou, Y. Zhang, G. Laput, and C. Harrison, "AuraSense," *Proc. 29th Annu. Symp. User Interface Softw. Technol. - UIST '16*, no. Figure 1, pp. 81–86, 2016.
- [12] S. S. Arefin Shimon, C. Lutton, Z. Xu, S. Morrison-Smith, C. Boucher, and J. Ruiz, "Exploring Non-touchscreen Gestures for Smartwatches," *Proc. 2016 CHI Conf. Hum. Factors Comput. Syst. - CHI '16*, pp. 3822–3833, 2016.
- [13] J. Han, S. Ahn, K. Park, and G. Lee, "Designing Touch Gestures Using the Space around the Smartwatch as Continuous Input Space," pp. 210–219, 2017.
- [14] A. Esteves, E. Velloso, A. Bulling, and H. Gellersen, "Orbits: Gaze Interaction for Smart Watches using Smooth Pursuit Eye Movements," *Proc. 28th Annu. ACM Symp. User Interface Softw. Technol. - UIST '15*, no. 1, pp. 457–466, 2015.
- [15] D. Akkil, J. Kangas, J. Rantala, P. Isokoski, O. Spakov, and R. Raisamo, "Glance Awareness and Gaze Interaction in Smartwatches," *Ext. Abstr. ACM CHI'15 Conf. Hum. Factors Comput. Syst.*, vol. 2, pp. 1271–1276, 2015.
- [16] F. Kerber, P. Lessel, and A. Krüger, "Same-side Hand Interactions with Arm-placed Devices Using EMG," *Chi 2015 Ea*, vol. 2, pp. 1367–1372, 2015.
- [17] W.-H. Chen, "Blowatch: Blowable and Hands-free Interaction for Smartwatches," *Ext. Abstr. CHI '15*, vol. 2, pp. 103–108, 2015.
- [18] F. Müller, S. Günther, N. Dezfili, M. Khalilbeigi, and M. Mühlhäuser, "ProxiWatch: Enhancing Smartwatch Interaction through Proximity-based Hand Input," *Proc. SIGCHI Int. Conf. Hum. Factors Comput. Syst. - CHI '16*, pp. 2617–2624, 2016.
- [19] G. Laput, R. Xiao, X. "Anthony" Chen, S. E. Hudson, and C. Harrison, "Skin buttons: cheap, small, low-powered and clickable fixed-icon laser projectors," *Proc. 27th Annu. ACM Symp. User interface Softw. Technol. - UIST '14*, pp. 389–394, 2014.
- [20] C. Zhang *et al.*, "TapSkin: Recognizing On-Skin Input for Smartwatches," *Proc. 2016 ACM Interact. Surfaces Spaces - ISS '16*, pp. 13–22, 2016.
- [21] A. Gupta and R. Balakrishnan, "DualKey: Miniature Screen Text Entry via Finger Identification," *Proc. 2016 CHI Conf. Hum. Factors Comput. Syst. - CHI '16*, no. Figure 1, pp. 59–70, 2016.
- [22] I. Oakley, D. Lee, M. R. Islam, and A. Esteves, "Beats: Tapping Gestures for Smart

Watches,” *Proc. 33rd Annu. ACM Conf. Hum. Factors Comput. Syst. - CHI '15*, pp. 1237–1246, 2015.

[23] H. Xia, T. Grossman, and G. Fitzmaurice, “NanoStylus: Enhancing Input on Ultra-Small Displays with a Finger-Mounted Stylus,” *Proc. 28th Annu. ACM Symp. User Interface Softw. Technol. - UIST '15*, pp. 447–456, 2015.

[24] B. Lafreniere, C. Gutwin, A. Cockburn, and T. Grossman, “Faster Command Selection on Touchscreen Watches,” *Proc. 2016 CHI Conf. Hum. Factors Comput. Syst. - CHI '16*, pp. 4663–4674, 2016.

[25] C. Yu, H. Wen, W. Xiong, X. Bi, and Y. Shi, “Investigating Effects of Post-Selection Feedback for Acquiring Ultra-Small Targets on Touchscreen,” *Proc. 2016 CHI Conf. Hum. Factors Comput. Syst. - CHI '16*, pp. 4699–4710, 2016.

За контакти:

докторант Юри Николов Димитров
катедра „Компютърни науки и технологии”

ТУ Варна

E-mail: yndimitrov@gmail.com



ТЕХНОЛОГИЯ Li-Fi - ПЕРСПЕКТИВИ

Диян Ж. Динев

Резюме: Този доклад разглежда перспективите за внедряването на Li-Fi технологията в нашето ежедневие. През последните години идеята за предаването на информация посредством видимата светлина набира все по-голяма популярност в научните среди. Докладът представя същността на технологията, сферата на приложение и предимства и недостатъци на технологията.

Ключови думи: Li-Fi, Стандарт 802.15.7, Видима светлина

Li-Fi technology–Trends

DiyanZh. Dinev

Abstract: This paper considers examines the implementation of Li-Fi in our daily routine. Nowadays the idea for data transmission through Visible Light has become increasingly popular in the scientific circles. The paper presents the technology, field of application and advantages and disadvantages of the technology.

Keywords: Li-Fi, 802.15.7, Light Fidelity

1. Предаване на данни с видима светлина. Стандарт 802.15.7

802.15.7 (2011) стандартът е първият IEEE стандарт, който включва в себе си информация за безжична оптична комуникация посредством видимата светлина. Той определя и физическия слой (PHY), и MAC слоя. Архитектурата се основава на определяне на няколко слоя и под-слоеве, за да опрости стандарта и да предложи услуги и логически връзки от по-ниските слоеве към по-високите. Вземат се под внимание много фактори, надграждащи характеристики и функционалности като затъмняване и видимост, цвят за функционалност и стабилизация. Стандартът е проектиран предимно за предаване на мултимедийни данни, но може да се ползва и за предаване на други услуги.

Като IEEE стандарт 802.15.7 е свободен. Той е създаден, за да запълни нарастващите обществени нужди за комуникация, защото днешните стандарти скоро няма да могат да ги покриват. Също има пряко влияние върху пазара, създавайки нови приложения и необходимия за тях хардуер или чрез разширяване и обновяване на вече съществуващи такива, като например LED пазара, което трябва да направи достъпна среда светодиодите за безжично предаване на данни. Чрез комуникациите посредством видимата светлина (Visible Light Communication – VLC) се вземат предпазни мерки от рисковете за здравето чрез ограничаване на емисии на максимална мощност. 802.15.7 е стандарт, който се поддържа технически възможно най-прост и също се основава на структура, която позволява използването на някои широкоизползвани и по-стари стандарти като IEEE Std 802.15.4-2006, ITU-T I.432.1, ANSI/INCITS 373, които са от основни за неговото прилагане.

През 2011 г. проф. Харалд Хаас предлага технологията за безпроводно предаване на информация на базата на видимата светлина [1], [2].

Идеята на проф. Хаас се състои в това, че видимата светлина може да бъде използвана за предаване на данни с много по-голям капацитет в сравнение с традиционните радио вълни.

Li-Fi, подобно на Wi-Fi, дава възможност на електронните устройства от рода на компютри, лаптопи, смартфони и др., безпроводно да се свързват с Интернет, а също така с принтери, телевизори, слушалки. С други думи Li-Fi е еквивалентна на Wi-Fi, но използва светлинни лъчи вместо радио вълни.

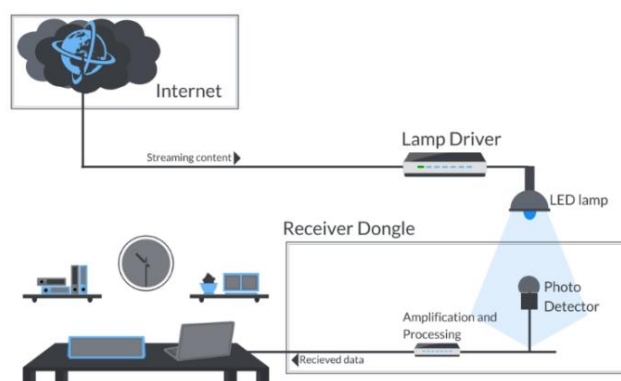
2. Същност на Li-Fi технологията

В основата на Li-Fi технологията са светодиодни лампи, които бързо изместват лампите с нагреваема жичка. Компаниите, производители на осветителни тела, преживяват истинска революция с развитието на светодиодите. С малката си консумация на енергия, по-дългия си експлоатационен период и възможността да предават едновременно информация с много високи скорости, светодиодите се явяват решение, което се изправя пред предизвикателството за намаляване на въглеродните емисии в световен мащаб.

В основата на технологията Li-Fi се използва възможността на светодиодите да се включват и изключват при много високи скорости и невъзможността на човешкото око да възприема бързите промени в интензивността на светлината. Невъоръженото човешко око не може да реагира на повече от 15 трептения на светлината в секунда.[2]

Бързото включено/изключено състояние на светодиода дава възможност да се предават данни в двоичен код: включено - логическа „1“, а изключено - логическа „0“. Концептуално този процес е подобен на кода на Морз.

На фигура 1 е илюстриран принципът на управление на светодиодна лампа. Постъпващата от Интернет информация се подава като управляващ сигнал (кодираща последователност) на драйвер, който включва и изключва светодиодната лампа с висока честота. Фотодетектор фиксира слабите изменения в интензивността, преобразува ги в електрически сигнали, декодира ги, след което възстановената информация се подава на компютър, мобилно устройство или друг клиент.



Фиг. 1. Еднопосочно предаване на информация чрез видим асвeтлина, излъчвана от светодиодна лампа.

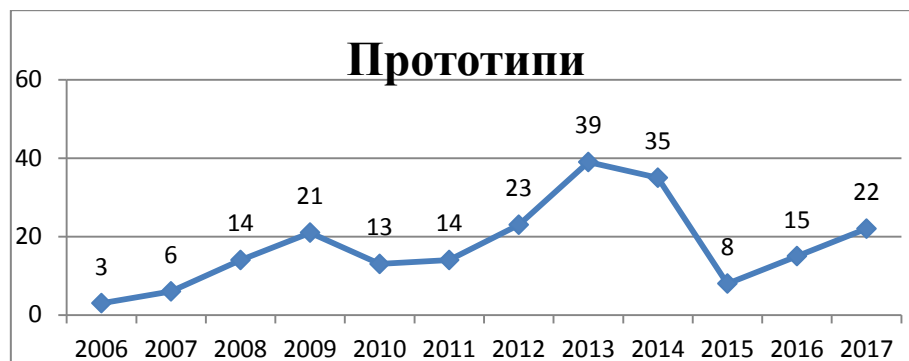
За осъществяване на двупосочна свързка в компютрите и другите устройства трябва да са вградени оптични предавател и приемник.

При радиокомуникациите са необходими антени и сложни приемници, докато Li-Fi е много по-проста и използва принципа на директна модулация.

3. Преди понятието Li-Fi и след него

Използването на видимата светлина като метод за предаване на информация е обект на изследване години преди официалното въвеждане на понятието LightFidelity (Li-Fi) от проф. Х. Хаас през 2011г. Редица световно известни фирми като Samsung, Panasonic, както и

редица университети като университета в Единбург и Бремен инвестират средства в изучаването и изследването на този тип предаване на информацията. Първите опити за внедряване на технологията до тогава, позната под името Visible Light Communication (VLC), започват през 2006г, като според [3] и [4] работните проекти наброяват едва три, като в следващите години интересът към нея нараства. До 2011г., преди официалното обявяване на понятието Li-Fi, проектите са наброявали 14. От тук нататък, след появяването на понятието Li-Fi, започва ежегодно нарастване на институциите и проектите, свързани с Li-Fi.



Фиг. 2. Годишен брой прототипи на база VLC и Li-Fi

Най-голямо дялово участие в изследването на технологията преди 2011г. прави компанията Samsung, инвестирайки големи суми, за да се опита да интегрира този метод на предаване на информацията в човешкото ежедневие – болнични заведения, военното дело, науката и др.

През октомври 2011 година няколко фирми от Германия, Норвегия, Израел и САЩ се обединяват в Li-Fi Консорциум, с цел развитието и довеждането до пазара на перспективната Li-Fi технология. В консорциума са уверени във възможността чрез Li-Fi технологията да се достигнат по-високи скорости от 10Gbps.

Паралелни изследвания в областта на Li-Fi оптична безпроводна свързка се провеждат в Обединеното Кралство, Южна Корея, Франция, Япония, Китай и др.

През 2012 година е основана фирмата pureLi-Fi, партньор на Единбургския университет (UoE). [8]

В началото на август 2013г. е обявено реализирането на първата в света високоскоростна, двупосочна Li-Fi система, разработена от Университета в Единбург (UoE), в изследователско сътрудничество с някои от водещите университети във Великобритания. Постигната е скорост 3 Gbps с единствен микросветодиод, разработен в Университета Стратклайд.[5]

На 20 март 2014 година pureLi-Fi анонсира, че в Центъра за проучване и разработване (Li-Fi R&D) на Единбургския университет са постигнати безпрецедентни скорости на предаване на данни чрез налични на пазара ('off-the-shelf') LED лампи, при намалена интензивност на светлината до нива, близки до изключено състояние. Това са случаите, когато не е необходимо осветление или при връзка на мобилно устройство с точка за достъп в и Интернет. С това фундаментално постижение се осигурява висока енергийна ефективност при предаване на данни чрез използване на светодиоди.

През март, по време на MWC 2014 (Световния конгрес по мобилни комуникации, Барселона) и CeBIT 2014, pureLiFi успешно демонстрира системата Li-1st, показан на фигура 3.[5]



Фиг. 3. Първият в света Li-Fi продукт на пазара - Li-1st

4. Предимства и недостатъци на Li-Fi

Li-Fi има следните предимства спрямо останалите безжични комуникации [6]:

- 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 не преминава 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.

Въпреки че не се характеризира с ниска цена, Li-Fi е по-подходящ за Point-to-Point системи. Потенциалната полза от такава технология може да бъде голяма, като се има предвид, че използва светлинни източници, които са навсякъде около нас. Въпреки това Li-Fi страда от доста недостатъци като [7]:

- Смущения от други светлинни източници като слънчевата светлина, електрически крушки, и т.н.
- Неспособност да предават информация през непрозрачни материали като стени и т.н.
- Високи разходи на електроенергия.
- Приемниците са неспособни да връщат отговори чрез светлина.

ЗАКЛЮЧЕНИЕ

Въпреки многобройните плюсове, които технологията ни предлага, се стига до заключението, че все още не е достатъчно развита за масово внедряване в нашето ежедневие поради нужда от допълнително оборудване, което на този етап не е евтино, както и горепосочените недостатъци като неспособност на приемниците да връщат отговор към предавателя и др. Преди да стане възможно масовото навлизане на технологията в нашето ежедневие, трябва да бъдат направени още много симулации и експерименти. Основните проблеми, върху които трябва да се извършат проучвания, са:

- Функционалност;
- Обратна връзка;
- Надеждност на предаването;

- Енергоемкост;
- Качество на услугите(QoS).

ЛИТЕРАТУРА

- [1] TED.http://www.ted.com/talks/harald_haas_wireless_data_from_every_light_bulb, 2011, Jul
- [2] Dinesh Khandal¹, Sakshi Jain², Li-Fi (Light Fidelity): The Future Technology in Wireless Communication, https://www.ripublication.com/irph/ijict_spl/ijictv4n16spl_11.pdf
- [3] Shelza Gupta, LiFi: From Illumination to Communication, <https://www.greyb.com/li-fi-vlc-patent-landscape-study/>
- [4] Shabaz Khan, Which companies were researching in lifi before Harald Haas?, <https://www.greyb.com/companies-research-lifi-before-harald-haas/>
- [5] Pure LiFi, <https://purelifi.com/>
- [6] V.J. RajaKumar, S. Bhuvaneshwari, A Study on Li-Fi-Internet at the Speed of Light, https://www.ijrcce.com/upload/2017/june/36_A_Study_on_Li-Fi-Internet_At_The_Speed_Of_Light.pdf
- [7] Advantages of LiFi/ Disadvantages of LiFi, <http://www.rfwireless-world.com/Terminology/Advantages-and-Disadvantages-of-LiFi.html>
- [8] Б. Йовчев, Р. Младенова, Видимата светлина като комуникационна среда – комуникационна среда – от осветление към предаване на данни, Общество и наука 9/2014 <http://www.bnci.eu/konfer/light-2014/26%20Li-Fi-doklad%202-2014.pdf>

За контакти:

ас. Диян Ж. Динев
 катедра „Софтуерни и Интернет технологии”
 Технически университет - Варна
 E-mail: diyandinev@tu-varna.bg



**СОФТУЕРНИ И ИНТЕРНЕТ
ТЕХНОЛОГИИ**

КОМУНИКАЦИОНЕН МОБИЛЕН МОДУЛ ЗА СЪБИРАНЕ НА МЕТЕОРОЛОГИЧНИ ДАННИ

Юлиян Г. Димитров, Тодорка Н. Георгиева

Резюме: В статията се представя проект на мобилен модул за събиране на метеорологични данни. Разработена е мобилна метеорологична станция за измерване на температура и влажност на въздуха. Основните акценти на работата са свързани с избора на цифров сензор за измерване на температура и влага, проектиране на модул с отворен код за извършване на научни изследвания, калибриране на сензора за постигне максимална точност, избор на технология за безжично предаване на събраната информация. Създаден е програмен код за визуализиране на получените данни на течно-кристален дисплей.

Ключови думи: безжична връзка, метеорологични данни, сензори

WIRELESS WEATHER STATION

Yuliyana G. Dimitrov, Todorka N. Georgieva

Abstract: The article presents a project for a mobile module for collecting meteorological data. A mobile meteorological station has been developed to measure temperature and humidity. The main focus of the work is the selection of a digital sensor for measuring temperature and humidity, designing an open source module to perform scientific research, calibrating the sensor to achieve maximum accuracy, choosing a technology for wireless transmission of the information gathered. Software for displaying received data on a liquid crystal display was created.

Keywords: amplitude modulation, LCD display, wireless mobile modul

1. Увод

Метеорологичните станции са снабдени с инструменти за извършване на наблюдения и измервания на атмосферните условия. Точността на всяко измерване е свързано с изготвянето на прогнози за времето и изследване на закономерностите в климата.

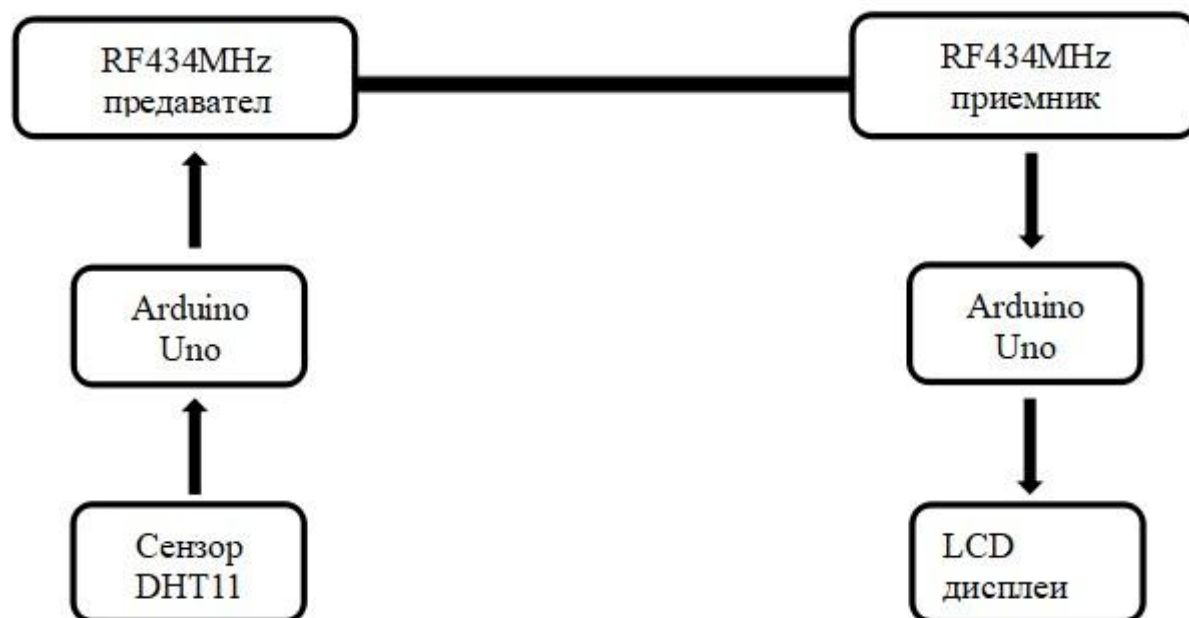
Задачите, които стоят пред предложението комуникационния мобилен модул за събиране на метеорологични данни, са:

- измерване на температура и влага с постигне на максимална точност;
- безжична комуникация с осигуряване на възможно максимално разстояние за предаване на данни;
- визуализиране на получената информация на течно-кристален дисплей.

2. Изложение

Проектиране и разработване на функционалните блокове

Мобилната метеорологична станция за измерване на температура и влага е реализирана на базата на платка тип Arduino Uno и цифров сензор DHT11 [1]. Получените данни от измерването се предават по 434MHz радио канал до приемник. Информацията се обработва и се извежда на течнокристалния дисплей [2]. При предаването на данни се използва ASK модулация (фигура 1).



Фиг. 1. Архитектура на безжична метеорологична станция

Системата е онлайн базирана, има лесен и удобен интерфейс. Тя дава възможност за извършване на тестове и експерименти с други подобни модули. Тестването на програмния код се извършва във виртуална среда чрез компютърна симулация на модула.

```

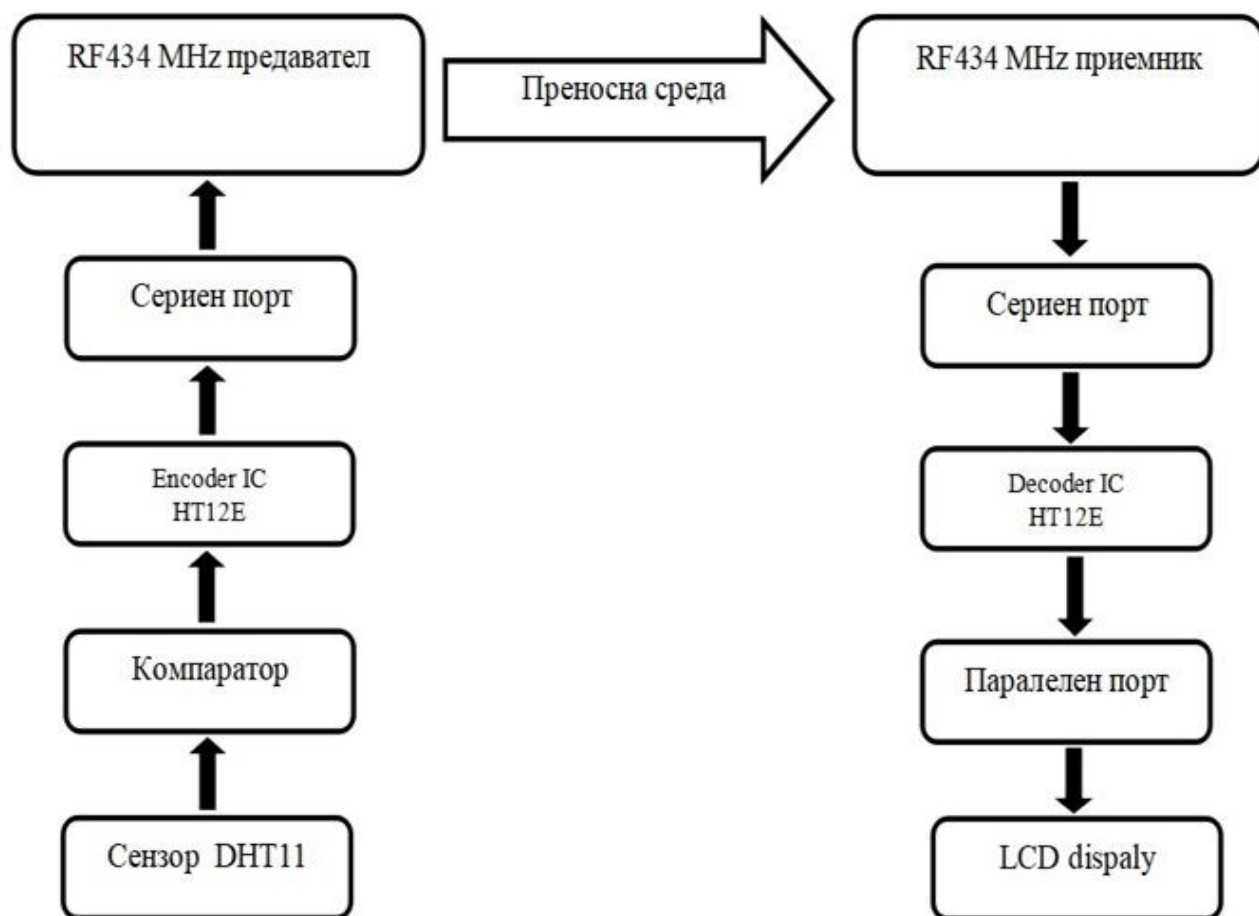
#include <stdio.h>
#include "DHT.h"
DHT* dht;
const int nSensors = 4;
int cycle;
void setup() {
  Serial.begin(9600);
  Serial.println("Serial initialized...");
  dht = (DHT*)malloc(nSensors*sizeof(DHT));
  for (int i=0; i<nSensors; i++) {
    dht[i] = DHT(i+2, DHT11);
    dht[i].begin();
  }
  cycle=0;
}
void loop() {
  char buffer[64];
  for (int i=0; i<nSensors; i++) {
    delay(2000./nSensors + 250);
  }
}
  
```

```

float h = dht[i].readHumidity();
float t = dht[i].readTemperature();
if (isnan(h) || isnan(t)) {
    Serial.println("No good boss.");
    continue;
}
sprintf(buffer,"%6d,%d,%2d,%2d",cycle,i,int(h),int(t));
Serial.println(buffer);
}
cycle++; }

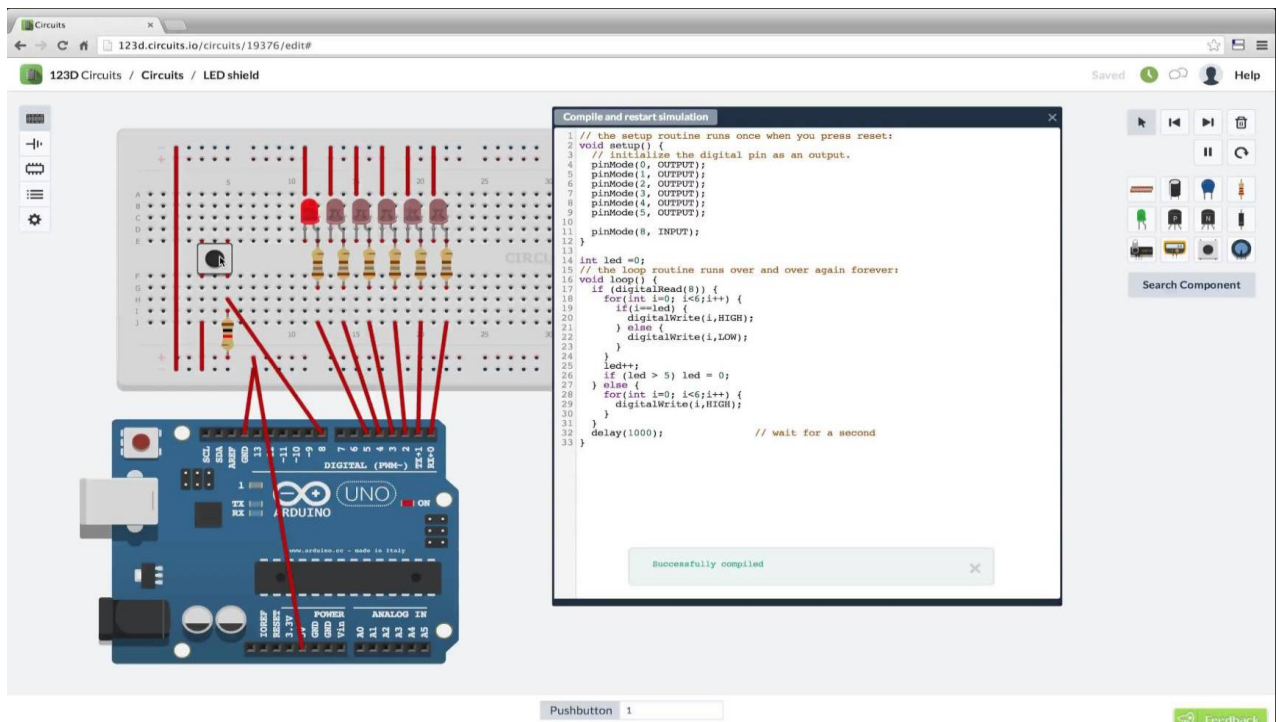
```

За хардуерната реализация е използвана Arduino Micro платка, базирана на Arduino Leonardo архитектура. Модулът събира и обработва данни от сензорите в реално време, като е постигната максимална точност. Използван е код, чрез който да се обработват данните от сензорите и да се извеждат в конзолния прозорец [3]. Събирането на данни е продължителен, непрекъснат цикъл със зададен период от време 12 часа (фигура 2).



Фиг. 3. Алгоритъм за събиране на данни

За да могат тестовете да се проведат в безопасна среда, е реализирана компютърна симулация, като за целта е използван специализиран софтуер, чрез който е създаден прототипът [4] (фигура 3).

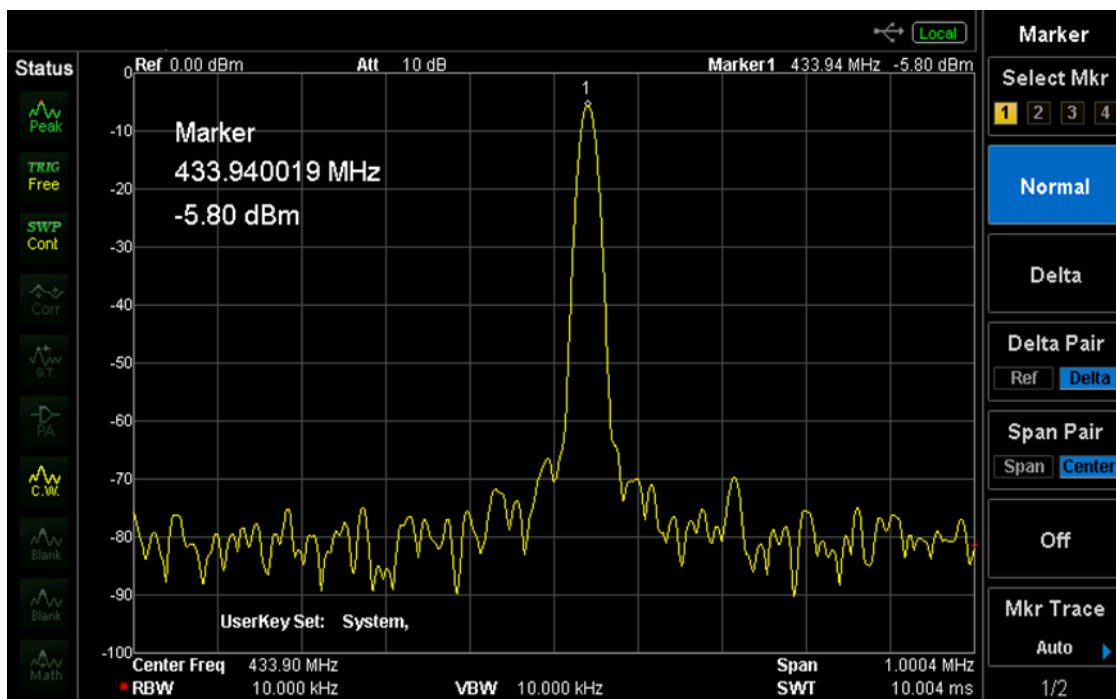


Фиг. 3. Онлайн среда за компютърна симулация

Проектът е реализиран с микроконтролери на SmartProjects. За софтуерната комуникация са осигурени групи от цифрови и аналогови компоненти за вход-изход (I/O), позволяващи свързване с други модули чрез сериен комуникационен интерфейс и USB [5], [6]. Програмирането на микроконтролерите в Ардуино платформата ползва интегрирана среда за разработка (IDE), основана на проекта Processing и поддържаща програмните езици C и C++.

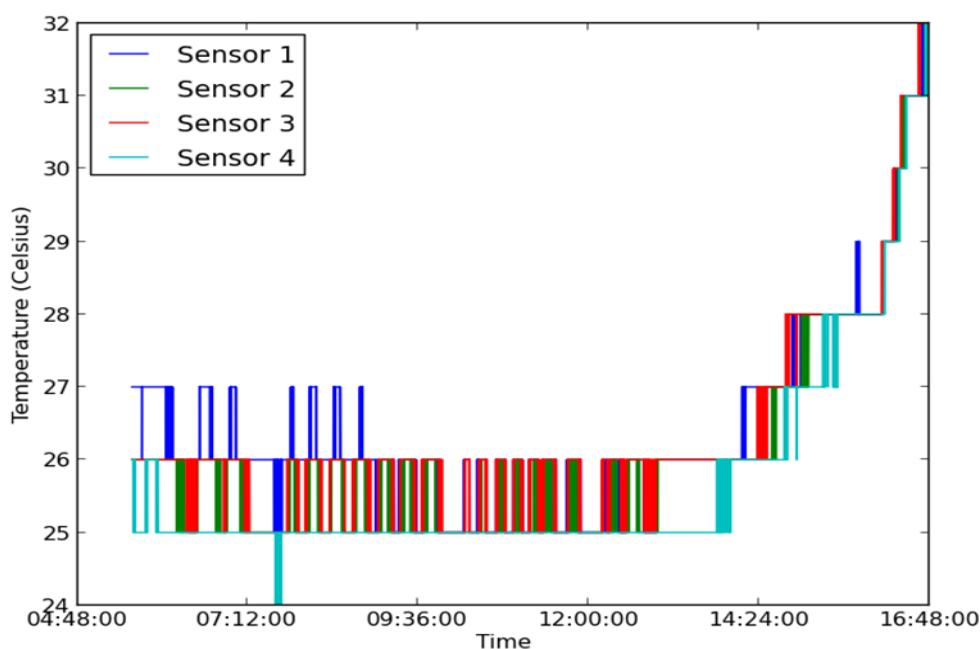
3. Резултати

Изследването е реализирано чрез спектрален анализатор на предавания сигнал. Данните между предавател и приемник се предават по безжичен път, като информацията се обновява на всеки 2s (фигура 4). По време на тестовите е постигнато разстояние от 60m, при което се гарантира качеството на предаваната информация. Получените данни са без закъснение и загуба на пакети. При отдалечаване на един от двата модула на разстояние над 60m се наблюдава прекъсване в комуникацията.



Фиг. 4. Спектър на предавания сигнал

При реализиране на експерименталните тестове е извършена корекция на алгоритъма за обработката на измерените данни. Това е направено с цел постигне на максимална точност на измерване. Сензорите се донастроят според климатичните условия, в които ще се използват. След всяка корекция се прилага визуален анализ, за да бъде установена разликата, като грешката от измерване вече е сведена до $\pm 2^\circ\text{C}$ (фигура 5).



Фиг. 5. Тестово измерване на температура

4. Заключение

В разработката е предложен мобилен модул за събиране на метеорологични данни в реално време. Мобилната метеорологична станция е с малки размери и е лесно преносима. Постигната е оптимална точност при измерване на параметрите. Проектирана е така, че да се използва лесно и удобно.

За да се увеличи разстоянието на предаване на получената информация, е предвидено в модула да се проектират антени с по-голям обхват на покритие. За подобрене на точността на измерване се предлага интегриране в мобилната системата на сензор DHT22, притежаващ по-добри технически параметри.

Комуникационният мобилен модул успешно измерва температура и влага, предава информацията по безжичен път, като получената информацията се визуализира на точно-кристален дисплей. Модулът е проектиран с учебна цел и е предназначен за научни и лабораторни цели.

Литература

- [1]. Bruce Eckel, Temperature and humidity module DHT11, Published 2016
- [2]. Matt Clary, Interfacing to an LCD Screen Using an Arduino, Published 2015
- [3]. Jeremy Blum, Exploring Arduino: Tools and Techniques for Engineering Wizardry, Article published, 2016
- [4]. Michael Margolis, Arduino Cookbook 2nd Edition, London 2013
- [5]. Тренков Й, Енциклопедия на електрониката. Принципи, системи и мрежи, том 4, София, 2014.
- [6]. Тренков Й, Енциклопедия на електрониката. Интегрални схеми и сензори, том 3, София, 2014.

За контакти:

Юлиан Димитров
катедра „Комуникационна техника и технологии“
ТУ-Варна
E-mail: yuliandimitrov.vn@gmail.com

Доц. д-р Тодорка Георгиева
катедра „Комуникационна техника и технологии“
ТУ-Варна
E-mail: tedi_ng@mail.bg



СИСТЕМИ ЗА ОТКРИВАНЕ И ПРОТИВОДЕЙСТВИЕ НА БЕЗПИЛОТНИ ЛЕТАТЕЛНИ СИСТЕМИ ОТ ТИПОВЕ „МИНИ“ И „МИКРО“

Васил Р. Люцканов

Резюме: В настоящата статия са разгледани някои модерни системи за противодействие на безпилотни летателни системи. Обърнато е внимание на използваните в тях технологии и възможностите им за интегриране с цел максимална ефективност, ниски разходи и безопасност за хората и околната среда. Анализирана е степента на ефективност на технологиите при различните групи платформи за БЛС от типове „Мини“ и „Микро“ съгласно класификацията на НАТО за БЛС.

Ключови думи: безпилотни летателни системи, БЛС

Counteraction systems of unmanned aircraft vehicles

Vasil R. Lyutskanov

Abstract: This article discusses some advanced systems to counteract unmanned aircraft systems. Attention is paid to the technologies used and their integration capabilities for maximum efficiency, low cost and people and environmentally friendly. An analysis has been made of the degree of effectiveness of the technologies in the different clusters of "Mini" and "Micro" UAV platforms according to the NATO classification for UAV.

Keywords: unmanned aircraft vehicles, counteraction-UAV

1. Увод

С навлизането на голяма част от държавите в света, в частност и на Република България, в информационната ера все по-голямо значение придобиват роботизираните системи за събиране, обработка и разпределение на информационните потоци в реални мащаби на времето. Особено през последните две десетилетия най-авангардните от тези системи са използващите третото измерение, като голяма част от тях са безпилотните летателни системи (БЛС). Тези системи, комбинирани със съответното оборудване, биха могли да решават широк кръг от мисии и задачи.

Появата на пазара на необходимите компоненти (летателни апарати, авионика, телекомуникация, мобилни приложения за управление), доведе до поредния технологичен скок в информационните технологии, а именно: масовизиране на използването на безпилотните летателни апарати (БЛА), снабдени с необходимата сканираща или измерваща апаратура за набиране на разнообразна информация за пространствено обособени обекти и явления [1]. От друга страна масовото използване на БЛС заражда нуждата от нормативна рамка относно безопасното им използване, без нарушаване на лично пространство, както и използването им за противоправни и престъпни действия.

Към момента няма действаща категорична нормативна уредба, с която да бъде съобразено ползването на БЛС, което провокира развитието на системите за противодействие на БЛС, някои от които ще бъдат разгледани в тази статия.

2. Системи и технологии, използвани за противодействие на БЛС

Системата DroneWatcher на американския производител DeTest е доста гъвкава и приспособима система за намиране, проследяване, предупреждение и забрана на БЛС, включваща както радарни (Harrier DSR), така и технологии за анализ на сигналите (DroneWatcher RFI DroneWatcher APP) [5]. Всяка една от тези технологии може да са използва както индивидуално, така и в комбинация, за да осигури ниво на сигурност специфично за всеки потребител. Приложението DroneWatcher APP, което е достъпно в Google Play, предоставя евтина, но сравнително ефективна технология за защита на личната неприкосновеност и сигурността на малки предприятия. При нужда от покриване на голяма площ е възможно разгръщане на няколко устройства с активиран Drone Watcher APP и използващи допълнителната услуга на DeTest, DroneWatcherWEB, чрез която може да се осигури голяма зона контролирана и забранена за БЛС, което е подходящо за летища, затвори, големи спортни мероприятия на открито и други. Недостатък на посоченото приложение, е че регистрира единствено БЛС, които работят с WiFi.

За по-високи изисквания към сигурността към DroneWatcher APP могат да бъдат интегрирани и Harrier DSR (Drone Surveillance Radar) и DroneWatcher RF (Radio Frequency), с което да се достигне 100 % сигурност на въздушното пространство, като APP и RF технологиите откриват, проследяват и идентифицират радиоуправляеми БЛС, а технологията Harrier DSR намира и проследява БЛС без радио управление или автоматично програмирани за изпълнение на определени мисии.

Harrier DSR представлява усъвършенстван, военен, електронен, доплеров радар, проектиран за откриване на цели с малки размери, който надеждно открива БЛС в радиус от 2 и повече мили. За разлика от посочените дотук технологии, Harrier DSR засича БЛС, които са без радиоуправление или са програмирани да следват задаен маршрут по GPS. Електронният радар на DeTest, предлага по-висока производителност, по-дълъг полезен живот и по-ниски разходи за поддръжка спрямо конвенционалните радарни системи, базирани на магнетрон. Harrier също така включва усъвършенствана SQL система за данни, която осигурява класифициране на целевите групи в реално време и отхвърляне на фалшиви цели като птици и смущения. Системата може да взаимодейства и с видео, акустични и други технологии на други производители и включва консолидирана WEB услуга за показване на ситуацията в реално време, за конкретни сайтове и региони.

Harrier, може да се интегрира с технологиите за контрол на въздушното пространство, произведени не само от DeTest, но и от трети страни. Едно такова решение предлагат от компанията Prime Consulting & Technologies при тяхната система Anti Drone (www.anti-drone.eu), където има възможност Harrier DSR да бъде комбиниран с радара за периметрово наблюдение Spotter RF, който чрез извършване на спектрален анализ открива както БЛС, така и местонахождението на оператора [6].

Друга технология, която може да бъде интегрирана в AntiDrone, са акустичните сензори, които са в състояние да открият БЛС, дори ако са невидими за радари, или нямат радиочестотни връзки.

Акустичните сензори се използват от силите за сигурност, за да се идентифицира неразрешеното проникване на БЛС в защитени зони. Сензорната мрежа осигурява сигнали в реално време, както и събиране на цифрови доказателства. Сензорите за откриване на БЛС работят съгласно следната схема: Слушане-Анализ-Идентификация-Сигнали.

Слушане: Датчиците за откриване на БЛС разпознават уникални звукови сигнали от различни типове платформи. С други думи, акустичните сензори слушат околните дейности в реално време и автоматично вземат звукови проби, в случай че усещат активността на БЛС наблизо.

Анализ: Сензорите за откриване на БЛС сравняват звуковите проби с акустичните сигнали от вградената база от данни. Ако се намери съвпадение, акустичната система за откриване на апарати записва идентифициращата информация и автоматично издава предупреждение.

Идентификация: Системата за акустично откриване на БЛС има богата база от данни за акустични подписи, които позволяват да се разграничат БЛС от други шумови източници с високо ниво на точност и ниско ниво на фалшиви аларми.

Сигнали: Моментните предупреждения за регистрирани БЛС могат да се предават чрез електронна поща, SMS, съществуващи видео системи или системи за управление на инциденти, както и визуален интерфейс с отразяване в реално време на заобикалящите акустични дейности.

В AntiDrone системата могат да бъдат интегрирани технологии не само за откриване и проследяване, но и за неутрализиране на БЛС. Това са следните заглушители на сигнали:

RCIED JAMMER – устройството е базирано на съвременната технология на реактивно заглушаване и е специално конструирано за борба с дистанционно управляеми импровизирани взривни устройства (Remote Controlled Improvised Explosive Devices). Системите за заглушаване RCIED, напълно отговарят на съвременните заплахи и се характеризират с възможност за поддържане на работно състояние за дълъг период от време, както и лесна инсталация и демонтаж на място.

Заглушителите на RCIED имат разширени функции за програмиране, бързи настройки и лесна употреба, имат външни батерии за автономна работа и могат лесно да бъдат транспортирани.

Основното предимство на заглушителите RCIED е програмируемото генериране на сигнал, което осигурява ценна адаптация към различните вредни сигнали.

MERITIS JAMMER - Заглушителите на Meritis са специално разработени, за да бъдат използвани в системи за защита от БЛС. Те са в състояние автоматично да откриват радио сигнали с предварително определена честота в периметъра на сигурност.

Заглушителят може да бъде персонализиран и да бъдат дефинирани зони без БЛС. Когато нарушителят се доближи до критична инфраструктура или наруши поверителността на частните помещения, блокиращият блок се активира автоматично, за да заглуши откритите честоти на потенциална заплаха.

PHANTOM JAMMER – Компанията Phantom Technologies LTD е специализирана в производството на високотехнологични системи за заглушаване, използвайки най-модерните компоненти и технологии. Един от най-известните продукти на компанията е заглушителят за БЛС Eagle108. Този заглушител е специално създаден, за да неутрализира всякакви БЛС, които влизат в защитени зони.

Тази система за заглушаване на БЛС използва масив от насочени антени за сканиране на дрони. Когато системата открие заплаха, тя изпраща автоматична команда към блока за блокиране. Заглушителят блокира всички радиокомуникационни канали, включително видеовръзка и GPS сигнали.

Системата е в състояние точно да открие посоката на БЛС и да предаде конкретен лъч директно към нея. Когато това се случи, БЛС губи комуникация с оператора си, както и ориентация за местоположението си.

Системата има удобен за потребителя интерфейс, който показва откриващия азимут и състоянието на заглушаване. Благодарение на усъвършенстваните технологии за заглушаване на БЛС, използвани в системата, работният диапазон достига до 2 километра.

Други технологии, които могат да бъдат интегрирани в AntiDrone системата, са видео и термовизионно наблюдение и не на последно място - акустични устройства с голям обхват, чиято цел е да противодействат на оператора на БЛС, като го дезориентират и отвлекат

вниманието му от контрола на БЛС, като в същото време осигуряват на служителите по сигурността необходимото време за залавяне на нарушителя.

Една изцяло европейска система за противодействие на БЛС е проектът SafeShore, който е част от европейската програма за научни изследвания и иновации „Хоризонт 2020“. Основната цел на SafeShore е да покрие съществуващите пропуски в наблюдението на крайбрежните граници, да увеличи вътрешната сигурност, като предотврати трансграничната престъпност, като например трафика на хора и контрабандата на наркотици. Системата има възможност да бъде интегрирана във вече съществуващите системи за наблюдение на морските граници, с което да се създаде една непрекъсната линия на наблюдение на бреговата ивица [4].

Основната технология, която ще се използва за откриване на малки обекти, прелитащи на малка височина, е 3D LIDAR, който сканира небето и създава над защитената зона виртуален куполен щит. Тази технология използва импулсен лазер, с който облъчва целта и определя разстоянието до нея според измерения отразен импулс. Разликите във времето за връщане на импулса и дължината на отразената вълна дава възможност за издраждане на триизмерна възстановка на целта. За да се подобри откриването, SafeShore интегрира 3D LIDAR с пасивни акустични сензори, пасивно радиочестотно засичане и видео анализ. Плавателните средства и хората по брега ще бъдат откривани с 2D LIDAR, интегриран с видео анализ. Тези технологии могат да се разглеждат като бюджетни технологии, щадящи околната среда („зелени“). Очаква се, че комбинация от технологии като LIDAR, пасивни радио и акустични и видео анализи, ще станат задължителни за бъдещи системи за граничен контрол в чувствителни към околната среда области [2].

Мобилните платформи SafeShore ще бъдат свързани чрез софтуерната платформа Situation Awareness and Management System (SSMS), която ще предоставя изчерпателен набор от софтуерни инструменти за управление на целия цикъл на работа от команден и контролен център. Системата SSMS SafeShore ще подобри значително и ефективността на граничната и външната сигурност, освен това проектът ще насърчи и сътрудничеството между властите.

Проектът е на обща стойност над 5 милиона евро и се координира от Белгия. Българското участие в него е от фирма „ОПТИКС“ АД. Сред другите участници са: Кралската военна академия на Белгия, лондонският университет Queen Mary, Институтът по оптоелектроника в Румъния, Белгийската брегова охрана, румънската фирма UTI Group, израелската национална полиция, румънската гранична полиция и други. Срокът за изпълнение е 3 години, като крайна дата е определен 31 октомври 2018 година.

3. Приложение на технологиите за откриване и противодействие на БЛС от клас I категория „Мини“ и „Микро“.

Следва да разгледаме до каква степен са приложими споменатите до тук технологии за бързото и адекватно откриване на БЛС. За целта ще използваме класификацията на НАТО за безпилотни летателни системи и ще разгледаме клас I (до 150 кг.), където са включени категориите „Мини“ (2 до 20 кг.) и „Микро“ (< от 2 кг.), които са най-широко използваните. БЛС от тези класове, или части за изграждането им, могат да бъдат закупени директно от търговската мрежа [7].

Таблица 1: Класификация на НАТО за безпилотни летателни системи клас I

Клас	Категория	Височина на полета	Радиус на действие
Клас I (до 150 кг.)	Малки (> от 20 кг.)	до 1650 м.	до 50 км.
	Мини (от 2 до 20 кг.)	до 1000 м.	до 25 км.
	Микро (< от 2 кг.)	до 650 м.	до 5 км.

БЛС в тези класове могат да се разделят на следните 3 групи според вида и конструкцията на летателната платформа [7]:

- Мултироторни (с ротиращи крила) – широко разпространени, лесни за управление, използват 3 или повече ротиращи крила.
- Планери (с неподвижни крила) – произведени от материали, поглъщащи или слабо отразяващи радарния сигнал.
- Джетове (задвижвани с реактивна турбина) –оборудвани с малък реактивен двигател. Развиват скорост от 400 км/ч.



Планер



Мултиротор



Джет

Фиг. 1. Видове платформи за БЛС

В таблица 2 са описани способностите на различните технологии, използвани в системите за противодействие на БЛС, да откриват съответните летателни платформи от типове „Мини“ и „Микро“. Прави впечатление, че най-широко разпространените мултироторни БЛС са лесно откриваеми от всички разгледани по-горе системи.

Таблица 2. Способности на различните технологии, използвани в системите за противодействие на БЛС за откриване на летателни платформи от типове „Мини“ и „Микро“

	Мултиротор	Планер	Джет
Видеонаблюдение	●	●	●
Активен радар	●	●	●
LIDAR	●	●	●
Акустични сензори	●	●	●
Спектроанализатори (WiFi, управление комуникации и други.)	●	●	●
Откриване на магнитни полета	●	●	●

● Откриваем ● Трудно откриваем ● Неоткриваем

Не така стои въпросът с планерите, които са произведени от материали, които поглъщат радарния сигнал или имат съвсем слабо отражение. Тези платформи използват въздушните течения за придвижването си, но могат да бъдат оборудвани и с двигател, който има възможност да се спира за дълги периоди от време, което ги прави неоткриваеми за акустичните сензори.

От таблица 2 става ясно, че най-трудни за откриване са БЛС, задвижвани с реактивна турбина, като това се обуславя от високата скорост, която развиват. Единствената сигурна технология за откриването им са акустичните сензори, поради силния и специфичен звук, издаван от двигателя.

В таблицата се въжда също, че технологиите за откриване на магнитните полета, излъчвани от БЛС, няма да бъде ефективна, поради малките размери на платформите от разгледаните категории.

4. Заключение

Съпоставяйки разгледаните системи за противодействие на БЛС и използваните в тях технологии с характеристиките на различните летателни платформи, могат да бъдат направени следните заключения:

1. Системата DroneWatcher на производителя DeTect е приложима за най-широко разпространената на пазара група БЛС, но не може да гарантира висока ефективност при всички групи БЛС от категориите „Мини“ и „Микро“.

2. Интегрирането на технологии като акустични сензори и заглушители към системата AntiDrone, значително повишава ефективността ѝ при откриване на БЛС, но използването на активен доплеров радар не е особено успешно при откриване на малки бързодвижещи се цели и такива изработени от материали поглъщащи или слабо отразяващи радарния сигнал.

3. Технологията LIDAR, използвана в проекта SafeShore, се очертава като водеща в изграждането на бъдещите системи за противодействие на БЛС, а интегрирането ѝ с технологии като акустични сензори и термовизионни камери, значително ще повиши ефективността на системата при всички групи БЛС от категориите „Мини“ и „Микро“.

ЛИТЕРАТУРА:

- [1]. Мардиросян Г. Х., „Основи на дистанционните аерокосмически технологии“, издателство на НБУ 2015г., стр. 60-62
- [2]. Marian Buric and Geert de Cubber: „Counter Remotely Piloted Aircraft Systems“, MTA Review, Vol. 17, No. 1, June 2017, стр.16-18
- [3]. http://cordis.europa.eu/project/rcn/203302_en.html
- [4]. <http://safeshore.eu/> - Bringing maritime border security to new dimension
- [5]. <http://detect-inc.com/> - Intelligent radar & sensors
- [6]. <http://anti-drone.ru/> - Система обнаружения и нейтрализации беспилотных летательных аппаратов
- [7]. Gabriel C. Birch , John C. Griffin, „UAS Detection, Classification, and Neutralization“, Market Survey 2015 - Sandia National Laboratories, стр. 9-15

За контакти:

докторант. Васил Р. Люцканов
катедра „Компютърни науки и технологии“
Технически университет – Варна
E-mail: bacuto@mail.bg

СОФТУЕРНИ И ИНТЕРНЕТ ТЕХНОЛОГИИ



SOFTWARE AND INTERNET TECHNOLOGIES

A SYSTEM AND A METHOD FOR DYNAMIC DATA VERSION MAPPING IN A WEBSITE BUILDING SYSTEM

Dimitar Z. Dimitrov, Elena V. Racheva

Abstract: A system for a website building system, implemented on a server, the server having at least one processor and a memory, including a website data managing system to process data to the currently described system. A method, implemented by the described system, for dynamic data version mapping further referred to as “DVM”. The system allows the storage of multiple versions of a data structure of a component block, depending on the time of creation, to migrate and work to any version of the website building system, which in turn enables rapid development and short release processes.

Key words: Website building system, Multiwebsite system, Data mapping, Data versioning, Data migration, Big Data

СИСТЕМА И МЕТОД ЗА ДИНАМИЧНО МАПВАНЕ НА ВЕРСИИ НА ИНФОРМАЦИЯ В СИСТЕМА ЗА СЪЗДАВАНЕ НА УЕБСАЙТОВЕ

Димитър З. Димитров, Елена В. Рачева

Резюме: Предлага се система, представляваща част от по-голяма система за създаване на уебсайтове. Системата е имплементирана на сървър, имащ поне един процесор и оперативна памет и включващ система за управление на уебсайт данни, предназначени за описаната система. Предложен е метод за динамично мапване на версии на данни, по-долу наричани „DVM“, имплементиран в описаната система. Системата позволява съхранението на множество версии на структури от данни от компонентен блок, в зависимост от времето на създаването им, мигрирането и оперирането им към всяка софтуерна версия на система за управление на уебсайтове, което на свой ред позволява ускореното разработване и кратките процеси на издаване на нови версии.

Ключови думи: Система за създаване на уебсайтове, Многосайтова система, Мапване на данни, Версии на данни, Миграция на данни, Големи данни

1. Field of the Invention

The present invention relates to website building systems generally and to editing tools in particular.

2. Background of the Invention

Interactive website building systems are now commonly used to design dynamic websites. The definition “multiwebsite system” [1] refers to a website building system architecture which consists and controls a set of many websites. The DVM system is applicable to a multiwebsite system. A website building system can be a standalone system, or may be embedded in a larger editing system. The website building systems can be implemented to execute in an online, offline or partially online environments. The proposed system is applicable to all types of website building systems in this classification. It has most advantages to online implementations, such as SaaS (Software as a Service) delivery, allowing a rapid development of a system with fast delivery methods.

Websites created using a website building platform are typically created and edited by users and then accessed by end users.

A website building system typically allows the user to create and edit applications. A visually

designed application consists of pages, containers, and components. Pages are typically displayed separately and contain components. Components are typically arranged in a hierarchy of containers (single page and multipage) inside the page including atomic components. A multipage container may display multiple minipages [2].

Components typically have attributes (e.g. size, position, color, frame type, etc.) and possibly content (e.g. contained text in a text component).

Pages may use templates either general page templates or component templates. Specific cases for templates include the use of an application master page containing components replicated in all other regular pages, and the use of application header/footer (which repeat on all pages).

Components may be fixed (such as a given geometrical shape) or based on a content management system (CMS) (such as a component containing a picture selected from a database of available pictures). Such a database may be extended by the user through importing (in this example) additional pictures.

Components may also be based on user-specified content, (such as a text area into which the user enters text) or be based on external information (static or dynamic), such as a RSS feed displaying information from an external data source or the content of a given internet page.

Component appearance can be modified by resizing, rotation and similar operations. Their appearance and behavior can also be modified by various modifiers, such as blurring and animation.

A specific set of attributes for a given components (e.g. a button having a given shape, color, text font) is referred to as component configuration. It will be appreciated that the component configuration definition does not include its content (e.g. the buttons' text label), position and size [3].

3. Summary of the Invention

This paper describes a system which manages a live migration of object data structures, representing website components and a method for dynamic data mapping used by the described system. The system will allow rapid development of new component versions. The system works alongside other systems in the website builder system. Such closely dependent systems are a dependency manager system, which takes care of how component versions work together and a data manager system, usually a parent system of the DVM system, which controls the input and output data from and out of the DVM system. Often website builder systems evolve in size and has and its data management system should be planned as a big data system. The suggested system is designed to work most efficiently NoSql (Object/Document) [4, 5] type of database with an appropriate ODM (Object Document Mapping) System [6], but it also can be implemented on a relational database with an appropriate ORM (Object Relational Mapping) System [7]. ORM can also be blended with an ODM System [8, 9].

4. Requirements to the Data Management System of the Website Builder System

The data management is a part of a separate system in the website building system, but a requirement for high performance of the data management system is mandatory for the DVM system to operate, being designed to work in real time.

Big Data applications are typically characterized by the 5Vs [10, 11], i.e. volume, velocity, variety, value, and veracity. More databases capable of high data I/O speed are necessary (i.e. velocity); the possibility of storing variable types of data by means of nonfixed schemas (i.e. variety) is desirable; the data accuracy and correctness (i.e. veracity) must be guaranteed by means of data quality techniques; and the storage, analysis and mining of data should lead to new insights which can be turned into new value. NoSQL databases are key to supporting Big Data applications, since they enable handling large quantities (i.e. volume) of highly-variable (i.e. variety), user-generated contents while guaranteeing fault tolerance, availability (i.e. velocity) and scalability [12].

A database system, capable of storing whole data objects and being able to operate with them at high speed and volume is important for the suggested system to operate, but it is not mandatory if otherwise optimized for the above-mentioned performance qualities or in cases where the website builder system is smaller and do not require optimizations in the field of the big data.

Independently to the database selection an ODM or/and ORM layer is mandatory to be included in the database management system, which will allow migration between databases in experiments for optimization. An example of widely used open source project which includes both systems is the Doctrine Project [13, 14].

5. Related Work

The suggested system can be classified as a system using a so called lazy migration method. Several object mapping libraries exist with a native support to lazy migration. [15]. While a convenient short-term fix, it is rather easy to specify conflicting migration operations in object mapper class declarations, even for very basic changes such as adding, removing, or renaming a property [16]. Moreover, migrations that involve more than one entity at-a-time, or require upgrades from several releases back, increase the challenge. Other lazy loading, data mapping methods exist, such as Datalog-based model [17], which formalizes all structural variants of persisted entities in non-recursive Datalog with negation. In particular, it captures strongly consistent writes and reads via put and get calls [18].

The method implements data migration but do not explicitly manage a required schema migration between release versions. It does not define a possibility of backward migration in cases of software release restoration, also do not define the possibility of a software application to run numerous versions at the same time. With the suggested DVM system a solution to those issues are also provided. While the datalog based model is a good approach for stateless dynamic systems, where the data can be updated numerous times without the need of migration between updates, it is not optimal for implementation in a website builder system, where data is being represented visually before updates, and the put requests are most often preceded by at least one get request for the data, where the lazy migration happens.

For standard data migration there are numerous methods and techniques, which mostly apply for smaller datasets, for applications that can be usually put offline between releases. Such methods do not work in a rapidly developed online applications, managing big data.

Data migration is often defined as synchronizing database replica sets, moving data from one location to another between databases and other type of replications. In the current paper, we do not refer to this definition and refer to the definition of the migration of a version of a software component that works with an explicitly defined version of a schema and data set, which can be migrated between versions of the schema and data, dependent to the component's version.

6. Full Description of the Invention

The producers of website building systems have chosen different approaches to the version release process. The mainstream approach is to have long version release cycles. In most cases, we observe those cycles to be around 3 years. In those systems, some providers decide to put their systems offline while the migration applies. In bigger datasets, it can result in a downtime in hours and when migration is not successful it might result in lowering service quality and downtime for days.

When a producer of a website building system choses a rapid development and version release process, new versions can be released from several times per year to even several times per week. This approach requires a more sensitive approach for migration of data between the versions required.

A release can be reverted in case of general issues, in that case versions of the newly stored data should be able to be reverted to its previous versions.

A producer can decide for different users to use different versions of the website building system. Even to use different versions of modules between users. In those cases, different versions of the system can be deployed and work at the same time. Which in term means that the data should be available for all versions of the software at the same time.

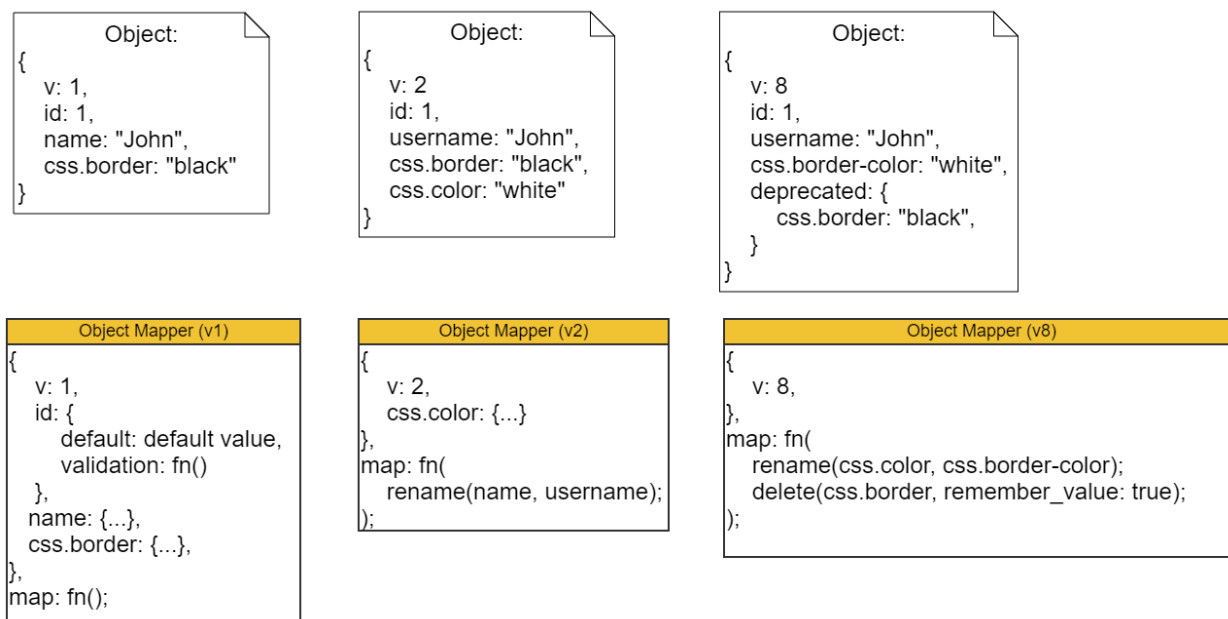


Fig. 1. Visual representation of versions of a sample data object and object mappers for each version.

The method requires the first version of the object mapper to include a full description of its properties. This is a requirement in future development of data management subsystems.

In different implementations, it can include a validation function, which can also be migrated through the object versions. Validation function is not required, but using it can benefit in developing a data validation system as a subsystem of a the DVM system, instead of designing it separately.

A requirement for the mapping function is for each of its methods to be reversible. This enables multiple versions of a software or a component to be running at the same time, also the ability to revert safely to older versions in case of general issues and disasters.

The definitions of the methods in the mapping function should be implementing to be reversible. An example of an implementation of the renaming function is:

```
function rename(field_name, new_field_name, revert_to_previous_version=false){
  revert_to_previous_version ?
    this.field_name = new_field_name :
    this.new_field_name = field_name;
}
```

Some mapping functions might require a storage of data, which is not in use in the latest versions, but might still be in use in the future version. The example with the delete method requires an additional deprecated data to be stored in the object itself just for the cases of reverting to older versions. Mappers can be designed to delete deprecated data for long unsupported versions. For example the property "css.border" was deprecated in version 8, but still accessible if version 7 is live to a set of users, when the component reaches version 30, and version 7 is no longer available to the users, the producer can decide to clear the deprecated data, with an appropriate mapping

function for version 30.

When using an Object Mapper, developers are required to create such a mapper for each new version of a component, which requires changes in the schema or the data itself.

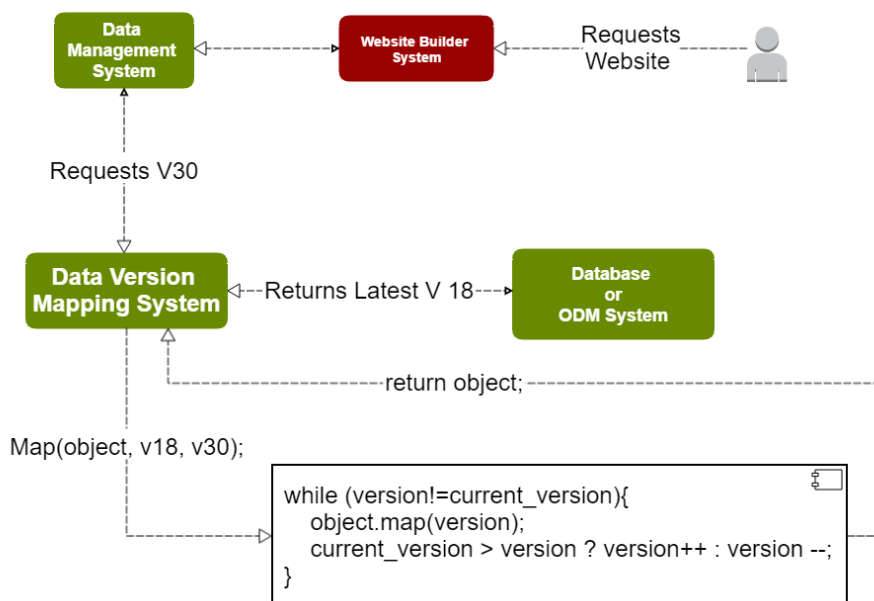


Fig. 2. General representation of the workflow of the Data Version Mapping System

In figure 2 a sample version migration function is shown as an example to a possible interface implementation. The graphic is simplified to show a version of a single component, for the workflow to be clearer. The data of a website object can consist of hundreds of data objects, including the data structures of containers like pages and components.

Each data object can have a separate version flow. For example, a component like a Text Widget can have a new version in the next release of a website builder system, while all other components can remain in the older versions if no data changes are required.

The example mapping function works in mapping the objects toward a newer or an older version if required.

The user requests a website with all its components, data management system requests each component through the DVM system, sending the requested version of the component's data. The DVM system in turn requests the version from the database, which can differ from the requested version. The DVM maps the object and applies all the mapping functions to migrate it to the requested version.

The DVM system can be called asynchronously even using multithreading if possible in the process of data retrieval to optimize the migration process. In this way, the hundreds of components, which a website includes can be migrated in parallel.

By using the DVM system, data which is not in use is never migrated, and in such a way expensive computing resources are saved.

Conclusions

The described method and system for dynamic data version mapping is another step towards the intensively growing requirements of sensitive online applications, working with big data. This paper in particular, suggests the system as a part of a website builder system in particular. Website builder systems may use hundreds of terabytes of data which is required to be online and live without any downtime. This is another reason for methods and systems, like the one described in this paper, to emerge.

Contemporary online applications tend to move to forms of processes with rapid deployment, especially with software distributed with the SaaS (Software as a Service) method. The online website building systems in particular, usually use this distribution method. While deployment interval in many vendors used to be between up to 3 years between versions, and migration processes in those cases can be done with the cost of downtime and disruption of service, more and more vendors choose rapid development processes which can result in shorter deployment intervals in some cases that being several times per week. These type of rapid development and deployment processes should be backed up with a rapid data migration methods, including the suggested method and system for dynamic data version mapping.

REFERENCES

- [1]. Dimitrov D. (2012), standardization based on multiwebsite system integration – International Science and Technology Conference, Dubai
- [2]. R. Ben-Aharon, N. Abrahami (2016), System and method for the creation and use of visually-diverse high-quality dynamic visual data structures, US 2016/0357527 A1 - United States Patent Application Publication
- [3]. A. Geva, O. Gartzman, G. Kaplan (2016), System and method for the generation of an adaptive user interface in a website building system, US 2016/0357527 A1 - United States Patent Application Publication
- [4]. P. J. Sadalage, M. Fowler (2012), NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence - Addison-Wesley Professional
- [5]. NoSQL-database.org - <http://nosql-database.org/>.
- [6]. Doctrine MongoDB Object Document Mapper (2010) - <http://www.doctrine-project.org/2010/05/13/doctrine-mongodb-object-document-mapper.html>
- [7]. A. Kurniawan (2012), Object-Relational Mapping (ORM): MongoDB, Mongoosejs and Node.js - PE Press
- [8]. Blending the ORM and MongoDB ODM - <http://docs.doctrine-project.org/projects/doctrine-mongodb-odm/en/latest/cookbook/blending-orm-and-mongodb-odm.html>
- [9]. Mapping Classes to the ORM and ODM - <http://docs.doctrine-project.org/projects/doctrine-mongodb-odm/en/latest/cookbook/mapping-classes-to-orm-and-odm.html>
- [10]. D. Laney (2001), "3D data management: Controlling data volume, velocity, and variety," - META Group
- [11]. Enterprise Architect, "The 5 V's of Big Data" <http://enterprisearchitects.com/the-5v-s-of-big-data/>
- [12]. M. Scavuzzo, D. Tamburri (2016) Providing Big Data Applications with Fault-Tolerant Data Migration Across Heterogeneous NoSQL Databases
- [13]. K. Dunglas (2014), Persistence in PHP with Doctrine ORM - Packt Publishing
- [14]. Doctrine Project - <http://www.doctrine-project.org/>
- [15]. U. Störl, T. Hauf, M. Klettke (2015), Schemaless NoSQL Data Stores – Object-NoSQL Mappers the Rescue? - In BTW
- [16]. S. Scherzinger, T. Cerqueus (2015), A Framework for Controlled Schema Evolution in NoSQL Application Development. In Proc. ICDE'15
- [17]. S. Scherzinger, U. Störl, M. Klettke (2015) A Datalog-Based Protocol for Lazy Data Migration in Agile NoSQL Application Development – ACM Digital Library
- [18]. S. Scherzinger, M. Klettke (2013), Managing Schema Evolution in NoSQL Data Stores. In Proc. DBPL'13

For contacts:

Dimitar Zdravkov Dimitrov
Computer Science and Engineering Department
Technical University of Varna
dimitrov@oxy.com

ОПТИМИЗИРАН ВАРИАНТ НА АЛГОРИТЪМА “IMAGE QUILTING” ЗА СИНТЕЗ НА ТЕКСТУРНИ ИЗОБРАЖЕНИЯ С ПРИЛОЖЕНИЕ В СИСТЕМИТЕ С ВИРТУАЛНА РЕАЛНОСТ

Даниела Д. Илиева

Резюме: Текстурният синтез се прилага широко в компютърната графика и обработката на изображения, като съществена причина за прилагането му е необходимостта от използване на изображения като покриващи повърхности за триизмерни обекти. Ето защо е необходимо разработването на оптимизирани варианти на актуални алгоритми за синтезиране на текстuri, разпределящи работата между максимално голям брой изпълнителни устройства. Тук се предлага вариант на алгоритъма IQ с използване на стандарта за програмиране върху хетерогенна цифрова платформа - OpenCL (*Open Computing Language*).

Ключови думи: Текстурен синтез, алгоритъм Image Quilting, Open CL, Open GL

OPTIMIZED VARIANT OF IMAGE QUILTING ALGORITHM FOR SYNTHESIS OF TEXTILE IMAGES WITH IMPLEMENTATION IN VIRTUAL REALITY SYSTEMS

Daniela D. Ilieva

Abstract: Texture synthesis is widely applied in computer graphics and image processing, and the essential reason for its implementation is the need to use images as covering surfaces for 3D objects. Therefore, it is necessary to develop optimized variants of current algorithms for texture synthesis that distribute the work between a maximum number of executive devices. Here is proposed variation of the IQ algorithm using OpenCL (*Open Computing Language*) programming standard on a heterogeneous digital platform.

Keywords: texture synthesis, algorithm “Image Quilting”, Open CL, Open GL

1. Увод

Текстурният синтез намира приложение в различни области на компютърните науки, но най-често той се прилага в сферата на компютърната графика и обработката на изображения. Съществена причина за прилагането му е необходимостта от използване на изображения като покриващи повърхности за триизмерни обекти. Този процес е по-известен като Texture Mapping или „текстуриране“.

Методът се поддържа от всички модерни приложно-програмни интерфейси (API) за реализиране на 2D и 3D графика (например: OpenGL и Direct3D). Текстурните изображения обикновено се получават от сканирани снимки и в повечето случаи те са с твърде малък размер, за да покрият цялата повърхност на обекта. В тази ситуация простото покриване ще доведе до неприемливи резултати като видими повторения и ясно изразени граници между основните блокове.

От друга страна са налице твърде мащабни изображения, които поради големия си брой правят едно графично приложение некомпактно, заемащо значително по обем цифрова памет.

Текстурният синтез решава тези проблеми чрез генериране на текстuri с желани размери и качество. Същността на процеса може да се формулира така: при дадена входна текстура *I_{in}* да се синтезира изходна текстура *I_{out}*, която да е достатъчно различна от дадената, но все пак да изглежда, че е генерирана от същия стохастичен процес.

Други приложения включват различни задачи за обработка на изображения, като например запълване на снимки или видео при цифрова компресия. Тези и още други причини карат синтезирането на текстури да се превърне в процес – по възможност по-оптимизиран и ефективен.

В много случаи по-бързото изпълнение на дадена задача се реализира чрез разпаралелването ѝ на множество по-малки. Ето защо е необходимо разработването на оптимизирани варианти на актуални алгоритми за синтезиране на текстури, разпределящи работата между максимално голям брой изпълнителни устройства. Този ефект се постига посредством прилагането на стандарта за програмиране върху хетерогенна цифрова платформа - OpenCL (*Open Computing Language*). Той превръща генерирането в процес, изпълняващ се за необходимото време и качество, а същинското текстуриране се извършва посредством взаимовръзката му с графичния стандарт OpenGL.

2. Текстурни изображения- определения, синтезиране

Текстурата представлява комплексно графично понятие. Чрез нея може да се опише голямото разнообразие от характеристики на една повърхност като терен, растения, минерали, кожа (фигура 2.1). Тъй като възпроизвеждането на визуалния реализъм на света около нас е основна цел на компютърна графика, текстурите са често използвано средство за генериране на синтетични изображения. Тези текстури могат да бъдат получени от различни източници, като например ръчно рисувани картини или сканирани снимки. Ръчно рисувани картини могат да бъдат естетически приятни на вид, но е трудно да се направят фото-реалистични. Повечето сканирани изображения обаче са с недостатъчен размер и това може да доведе до видими граници или повторения, ако те се използват директно за текстуриране.



Фиг. 2.1. Текстурни изображения

Няколко дефиниции за текстура:

1. Визуално и осезаемо качество на повърхността – грапавост.
2. Структура от преплетени конци, въжета или други подобни, които изграждат тъкани.
3. Характеристика, давана за физическа структура на обект – материал, размер, форма, разположение и пропорции на части.
4. Основно качество или характерна особеност на обект, същност.
5. В изобразително изкуство:
 - Характерно, визуално и осезаемо качеството на повърхността на едно произведение в резултат от начина, по който се използват материали.
 - Имитация на реално качество на представян обект.

Текстурите могат да бъдат аранжирани в широк диапазон: от стохастични до симетрични (фигура 2.2):

- Стохастични текстури – текстурните изображения изглеждат подобно на шум: цветни точки, които са разпределени случайно в изображението, като яркостта на избрания цвят се изменя от минимум до максимум. Повечето текстури изглеждат стохастични ако бъдат наблюдавани от разстояние.

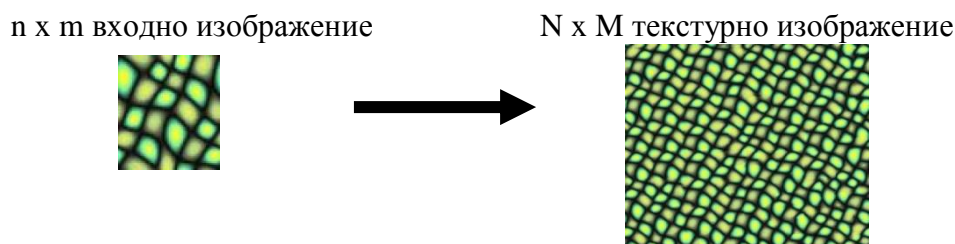
- Симетрични текстури – при този тип текстури се наблюдава периодично повторение на даден елемент. Пример за такъв тип текстура е тухлената стена.



Фиг. 2.2. Видове текстурни изображения

3. Процес на синтез на текстурни изображения

Синтезирането на текстурни изображения е процес на алгоритмично конструиране на цифрови изображения с големи размери на базата на структурното съдържание на изображение с малки размери (фигура 3.1). Този процес е обект на разглеждане от компютърната графика и се използва в множество приложения като обработка на цифрови изображения, 3D компютърна графика и продукции на филми.



Фиг. 3.1. Процес на текстурен синтез

Основната цел на процеса на синтезиране на текстурни изображения е създаване на изображение, което да изпълнява следните изисквания:

- Изходните размери на изображението да бъдат определени от потребителя.
- Полученото изображение трябва да прилича максимално на изходното изображение.
- Полученото изображение не трябва да притежава видими повторения или дефекти.
- Полученото изображение не трябва да бъде повтарящо се, т.е. да не притежава идентични структурни елементи.

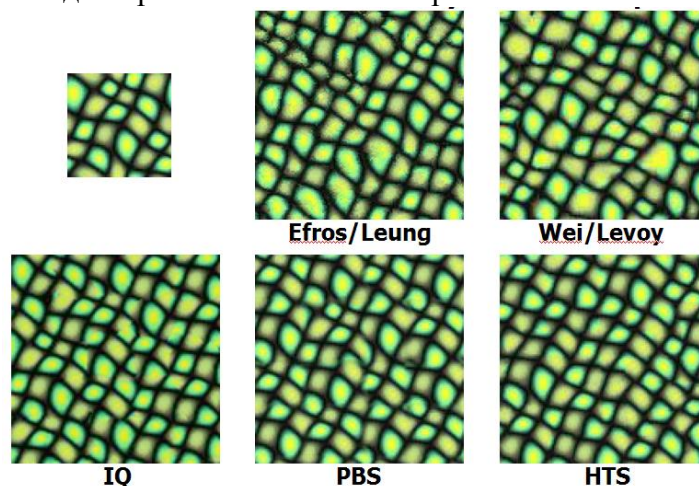
Основни методи за синтезиране на текстурни изображения :

- Облицоване (Tiling)
- Стохастичен текстурен синтез (Stochastic texture synthesis)
- Текстурен синтез с определена цел (Single purpose structured texture synthesis)
- Хаотична мозайка (Chaos mosaic)
- Текстурен синтез на базата на пиксели (Pixel-Based Texture Synthesis)
- Текстурен синтез на базата на парчета (Patch-Based Texture Synthesis)
- Химично базиран текстурен синтез (Chemistry based)

По отношение на качеството на полученото текстурно изображение в зависимост от използвания метод за премахване на припокриването са анализирани следните основни методи:

- метод на Efros/Leung (EL)
- метод на Wei/Levoy (WL)
- метод на Ashikhmin (AS)
- Patch-based Sampling (PBS)
- Image Quilting (IQ)
- Hybrid Texture Synthesis (HTS)

Резултатите от сравнителния анализ на полученото текстурно изображение са показани на фигура 3.2, като може да се различи наличие на проблемни зони.

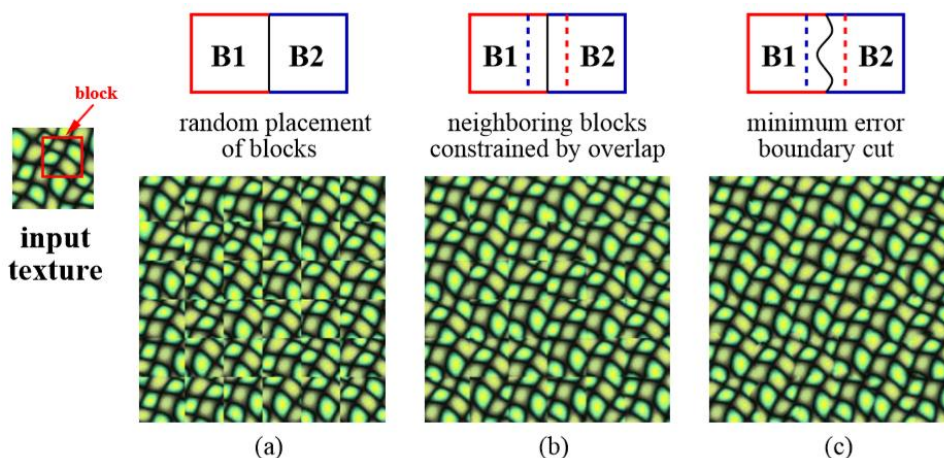


Фиг. 3.2. Сравнителен анализ на качеството на текстурата

От сравнителния анализ ясно личи, че алгоритъм ImageQuilting предоставя добри възможности за премахване на припокриването, като генерира текстура с минимални видими дефекти. Основната причина за неговия избор е най-бързото време за изпълнение.

4. Модифициран алгоритъм за синтезиране на текстурно изображение, основан на метод IQ

Идеята на алгоритъма е следната: Комбиниращ се произволното поставяне на блокове при Chaos Mosaic (хаотична мозайка) с пространствени ограничения на Efros & Leung. (фигура 4.1)



Фиг. 4.1. Основни стъпки в IQ алгоритъма

4.1. Общо описание

Алгоритъмът спада към групата на Patch-Based Texture Synthesis методите (синтезиране на база на парчета). Нека да дефинираме единицата за синтез B_i като блок с размери, определени от потребителя. Този блок принадлежи на множеството от всички възможни блокове, които могат да се формират от входното изображение S_B . За да синтезираме новата текстура на първа стъпка, нека просто я покрием с блокове, взети на случаен принцип от S_B . Резултатът вече изглежда разумно и за някои видове текстури се изпълнява не по-лошо от много предишни сложни алгоритми. Все пак, резултатът не е удовлетворяващ и колкото да се изглаждат краищата, за най-структурирани текстури е съвсем очевидно, че блоковете не съвпадат.

Като следваща стъпка добавяме припокриването на блокове като част от поставянето им върху новото изображение. Сега вместо да взимаме блок на случаен принцип, нека да потърсим от S_B за такъв блок, който по някакъв принцип съвпада със своите съседи от района на припокриване. Този подход значително подобрява структурата на получената текстура, обаче ръбовете между блоковете все още са доста забележими. Още веднъж изглаждането в краищата би намалил този проблем, но тук се прилага нов подход.

Разглеждаме грешката в региона на припокриване между блокове в него. Формираме крива на минималните грешки и установяваме тя да бъде границата на новия блок.

Целта е да намерим кривата между два припокриващи се блока, формирана от точки, където двете парчета си съответстват най-добре (когато грешката от припокриване е ниска). Това може лесно да се направи с динамично програмиране (алгоритъма на Dijkstra също може да се използва).

Кривата се изчислява по следния начин. Ако B_1 и B_2 са два блока, които се припокриват по вертикала с региони на припокриване B_1^{ov} и B_2^{ov} , тогава грешката на повърхността се дефинира като:

$$e = (B_1^{ov} - B_2^{ov})^2.$$

За да се намери оптималната крива за тази повърхност се изчислява натрупаната грешка E за всички пътища:

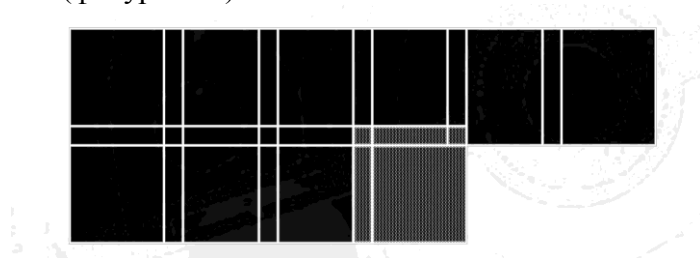
$$E_{i,j} = \sum (e_{i,j} + \min(E_{i-1,j-1}, E_{i-1,j}, E_{i-1,j+1}))$$

Минималната стойност на E означава край на оценяването. Аналогични методи могат да се прилагат по отношение на хоризонталните припокривания. Когато е налице както вертикално, така и хоризонтално, минималните пътеки се срещат в средата и общият минимален е избран за рязане.

4.2. Алгоритъм

1. Избират се размери на работния блок и препокривания регион, както и стойност за толеранс на грешката.

2. Формирането на изходната текстура се извършва на стъпки - блок по блок, като се отчита и препокриването (фигура 4.2.).



Фиг. 4.2. Процес на формиране на текстурата

3. На всяка стъпка се претърсва входната текстура за набор от блокове, които отговарят на ограничения за припокриване (според избрания толеранс на грешка). Избира се произволно един от тях.

4. Изчислява се грешката между последно избрания блок и старите блокове в припокриващия регион. Намира се кривата на минималните грешки в региона на припокриване и се взема за границата на новия блок.

5. Новосформираният се блок се прилага (“закърпва”) към текстурата.

Размерът на блока е единственият параметър, контролиран от потребителя и зависи от свойствата на входната текстура; блокът трябва да бъде достатъчно голям, за да улови съответните структури в текстурата, но и достатъчно малък, така че взаимовръзката между тези структури да е основана на алгоритъм.

Във всички от нашите експерименти ширината на препокриването (от едната страна) е 1/6 от размера на блока. Грешката е изчислена като се използва норма L2 за стойностите на пикселите. Толерантността на грешката е създадена да бъде в рамките на 0,1 пъти от най-добрия съвпадащ блок.

4.3. Резултати

Резултатите от процеса на синтез за широк спектър от входни текстури са показани на фигура 4.3. Докато алгоритъмът е особено ефективен за полу-структурирани текстури (които винаги са били най-трудни за статистическия текстурен синтез), изпълнението е удовлетворяващо и за стохастичните текстури. Двата най-типични проблема са прекомерното повторение (например изображение на плодове), и несъответстващи или изкривени граници (например на изображението с ядки). И двете са най-вече в резултат на липса на разнообразие в структурата на входната текстура.



Фиг. 4.3. Резултати от IQ, входни размери 70X70, изходни размери 140X140

Алгоритъмът е не само лесен за прилагане, но също и доста бърз: на неоптимизиран MATLAB код, използван за генериране на тези резултати (фигура 4.3) изпълнението завършва в рамките на 15 секунди. Докато направени тестове с алгоритъма, имплементиран върху C#, дават следните резултати (таблица 1):

Таблица 1. Резултати от извършените тестове

Входни размери	Изходни размери	Размер на блока	Брой канди- дат блокове	Зона на пре- покриване	Време
256x256	256x256	16x16	16	4	6 сек.
256x256	512x512	16x16	16	4	23 сек.
256x256	1024x1024	16x16	16	4	2мин.16сек.
256x256	2048x2048	16x16	16	4	21мин.20сек.

Тъй като формиращото се парче е винаги с еднакви размери, става много лесно да се оптимизира процеса на търсене, без да се прави компромис с качеството на изходното изображение.

Тези качества на описания алгоритъм допринесоха за това той да бъде избран като основа, която да бъде оптимизирана и след това вложена в конвейера на едно триизмерно приложение.

Литература

- [1]. D. Ebert, K. Musgrave, D. Peachey, K. Perlin, S. Worley "Texturing and modeling. A procedural approach. Third edition. Morgan Kaufman Publishers. 2003. ISBN: 1-55860-848-6
- [2]. Lin, Z.; Zhu, Q. & Gold, Digital Terrain Modeling: Principles and Methodology, CRC Press, C. (2005)
- [3]. L. Wei and M. Levoy. Fast texture synthesis using tree-structured vector quantization. In K. Akeley, editor, Siggraph 2000, Computer Graphics Proceedings, pages 479-488. ACM Press / ACM SIGGRAPH / Addison Wesley Longman, 2000.
- [4]. M. Ashikhmin, Synthesizing natural textures. In The proceedings of 2001 ACM Symposium on Interactive 3D Graphics, pages 217-226, 2001.
- [5]. L. Liang, C. Liu, and H. Y. Shum. Real-time texture synthesis by patch-based sampling. Technical Report MSR-TR-2001-40, Microsoft Research, 2001.
- [6]. A. A. Efros and W. T. Freeman. Image quilting for texture synthesis and transfer. In Eugene Fiume, editor, SIGGRAPH 2001, Computer Graphics Proceedings, pages 341-346. ACM Press / ACM SIGGRAPH, 2001.
- [7]. D. Zhou and G. L. Gimel'farb. Texel-based texture analysis and synthesis. In Proc. Image and Vision Computing New Zealand (IVCNZ 2004), Akaroa, New Zealand, pages 215-220, November 2004.
- [8]. Miedema J (2010) Synthetic Environments in Design Processes. PhD thesis, University of Twente
- [9]. Wang, R. & Wang, X. (2009). Experimental Investigation of Co-Presence Factors in a Mixed Reality-Mediated Collaborative Design System, Proceedings of the 6th International Conference on Cooperative Design, Visualization, and Engineering, pp. 333-340, Luxembourg, Springer, Berlin

За контакти:

докторант Даниела Д. Илиева
Технически университет – Варна
Катедра „Компютърни науки и технологии”
E-mail: ilievadaniela@mail.bg

СИМУЛИРАНЕ НА ВОДНА ПОВЪРХНОСТ В СИСТЕМИ С ВИРТУАЛНА РЕАЛНОСТ

Даниела Д. Илиева

Резюме: Средства за симулиране и изобразяване на вода и друг вид течности са една от основните потребности в съвременната индустрия, свързана с изграждането или използването на визуални ефекти. Те се използват в киното, рекламната индустрия, видео игри и други. Водата, генерирана с помощта на компютър, може да бъде изградена с високо ниво на реализъм както за спокойни води, така и за води с голямо количество пяна, пръски и вълни. В тази статия се разглежда алгоритъм за създаване на вълни в океан чрез вълни на Герстнер и наслагване на шум на Перлин. Полученият резултат е със задоволително качество и може да се използва успешно в компютърните системи с виртуална реалност.

Ключови думи: Вълни на Герстнер, шум на Перлин, виртуална реалност

WATER SURFACE SIMULATION IN SYSTEMS WITH VIRTUAL REALITY

Daniela D. Ilieva

Abstract: Tools for simulating and displaying water and other liquids are one of the major necessities in the modern industry related to the construction or use of visual effects. They are used in the cinema, advertising industry, video games and more. Computer generated water can be built with a high level of realism, both for serene water and for water with a large amount of foam, splashes and waves. This article explores an algorithm for creating waves in the ocean by waves of Gerstner and the overlay of Perlin's noise. The result is of satisfactory quality and can be successfully used in computer systems that recreate virtual reality.

Keywords: waves of Gerstner, Perlin's noise, virtual reality

1. Увод

В океанографската литература съществуват множество различни методи, описващи поведението на водата. Тук се спираме на някои от тези методи. За визуализиране на движението на водната повърхност се използва алгоритъм, разработен от Герстнер. За разнообразяване на повърхността на водата чрез добавяне на множество малки вълни, се използва алгоритъма на Кен Перлин за синтезиране на т.нар. шум на Перлин. Засветяването на водната повърхност се извършва с помощта на модифициран модел на Блин и Фонг. За създаване на пяна по фронтовете на вълните се използват методите за текстуриране на OpenGL и Cg.

Основните аспекти, свързани с компютърната симулация на океан, са:

- движение на водната повърхност, включващо различни нива на интензитет, амплитуда, скорост, както и дължина на вълните.
- отражение на околната среда в различни степени.
- визуализиране на пяна по върховете на вълните, които са с достатъчно висока амплитуда и стръмност.
- вариации в параметрите на водата, в зависимост от атмосферните условия.
- пречупване на падащата светлина, предизвикано от движението на водната повърхност.

2. Алгоритъм на Герстнер за създаване на вълни на океан

Този метод е един от първите приблизително точни модели, описващи движението на течностите. Физическият модел се базира на движението на самостоятелни точки от дадена повърхност. Когато вълна премине през някои от тези точки, те извършват кръгово движение. Нека дадена точка е описана от $x_0 = (x_0, z_0)$, а височината ѝ е $y_0 = 0$, при неподвижна повърхност. Когато през нея премине вълна, тя ще бъде отместена във време t с:

$$\begin{aligned}x &= x_0 - \left(\frac{\kappa}{k}\right) \sin(K \cdot x_0 - \omega t) \\y &= A \cos(K \cdot x_0 - \omega t)\end{aligned}\quad (1)$$

където K е вектор, описващ хоризонталното движение на вълната. Неговата дължина е κ и е:

$$k = \frac{2\pi}{L}$$

където L - дължина, ω - честота на вълната.

Използваме уравнение (1) във вида (2): [2]

$$\begin{aligned}x &= x_0 + \sum_{i=0}^n (Q_i A_i D_i \cdot x \cos(\omega_i D_i \cdot \vec{xz} - \varphi t)) \\z &= z_0 + \sum_{i=0}^n (Q_i A_i D_i \cdot y \cos(\omega_i D_i \cdot \vec{xz} - \varphi t)) \\y &= \sum_{i=0}^n (A_i \cdot \vec{xz} - \varphi t)\end{aligned}\quad (2)$$

Тук позицията на дадена точка от повърхността се описва с вектора \vec{xz} и има следните параметри:

Q_i – Стръмност на вълната;

A_i – Амплитуда на вълната;

D_i – Посока на движение на вълната, описана от вектора \vec{xz} ;

ω – Честота на вълната;

φ – Фаза на вълната;

t – Време;

Параметърът Q_i може да варира в границите $[0, 1]$ и определя наклона на вълната. Стойност $Q_i = 0$ предизвиква нормална синусоидна вълна, при $Q_i = 1$, вълната има заострен връх. Ако $Q_i > 1$, то хребетът на вълната ще се превърти и ще предизвика нереалистичен вид. За да варира между възможно най-изгладена вълна и възможно най-заострена вълна, то Q_i може да се представи като: $Q_i = \frac{Q}{\omega_i A_i n}$, където n е броя на вълните.

Честотата ω се определя от уравнението: $\omega = \frac{2\pi}{L}$.

Фазата на вълната зависи от скоростта ѝ и нейната честота: $\varphi = S\omega$.

Скоростта на вълната се влияе от нейната дължина, дълбочина и напрежението на повърхността. При по-плитки води, дъното има забавящ ефект върху скоростта на вълната. За дълбоки води обаче, влиянието на дълбочината може да се пренебрегне [3]. От това следва, че:

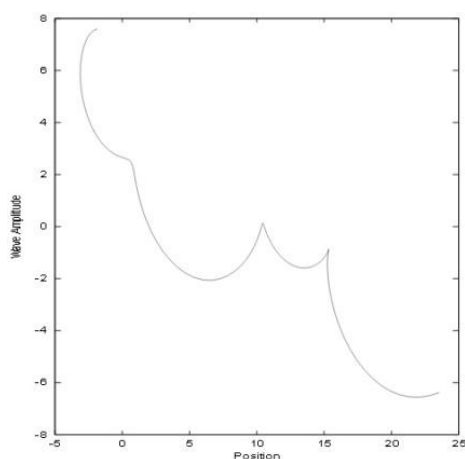
$$S = \sqrt{g \frac{2\pi}{L}}\quad (3)$$

където g е гравитационната константа 9.81 m/s^2 .

За да се получи по-интересно движение на повърхността, се сумират n на брой вълни със съответстващи им параметри. Тъй като при добавянето на всяка следваща вълна се

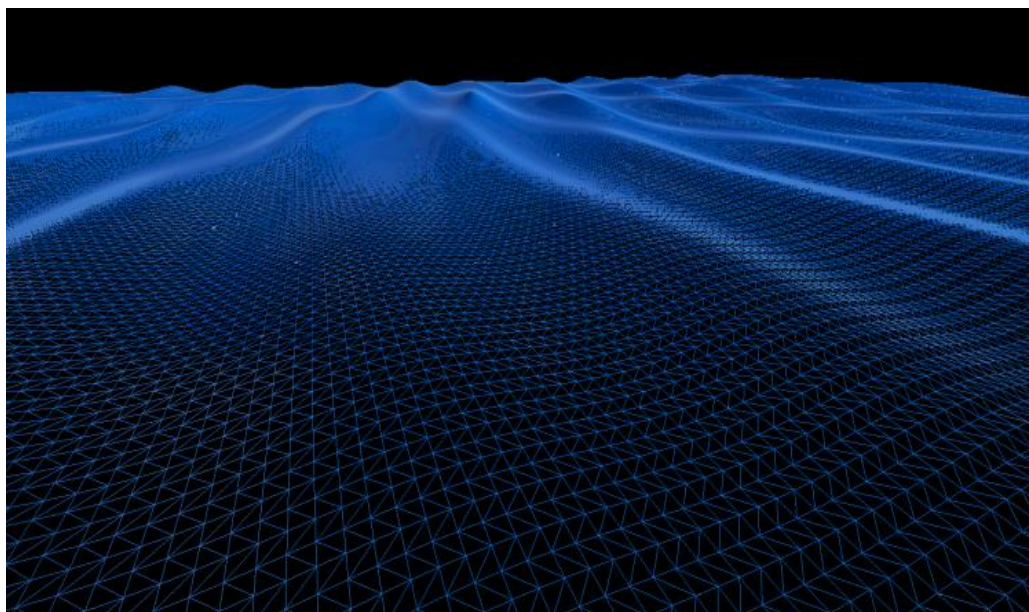
налага нейните отмествания да се изчисляват наново, то от броя на вълните зависи производителността.

Пример за визуализацията на 3 вълни с определени параметри е даден на **Error! Reference source not found.1.**



Фиг. 1. Профил на 3-режимна вълна на Герстнер

При сумирането на 4 вълни на Герстнер се получава следния резултат: (фигура 2)



Фиг. 2. Сумиране на 4 вълни на Герстнер

Резултатът наподобява повърхност на океан, но липсва детайл както по самите вълни, така и по местата, където все още не е минала вълна (спокойните места). Това е един от основните недостатъци на алгоритъма на Герстнер, когато се използват малко на брой вълни.

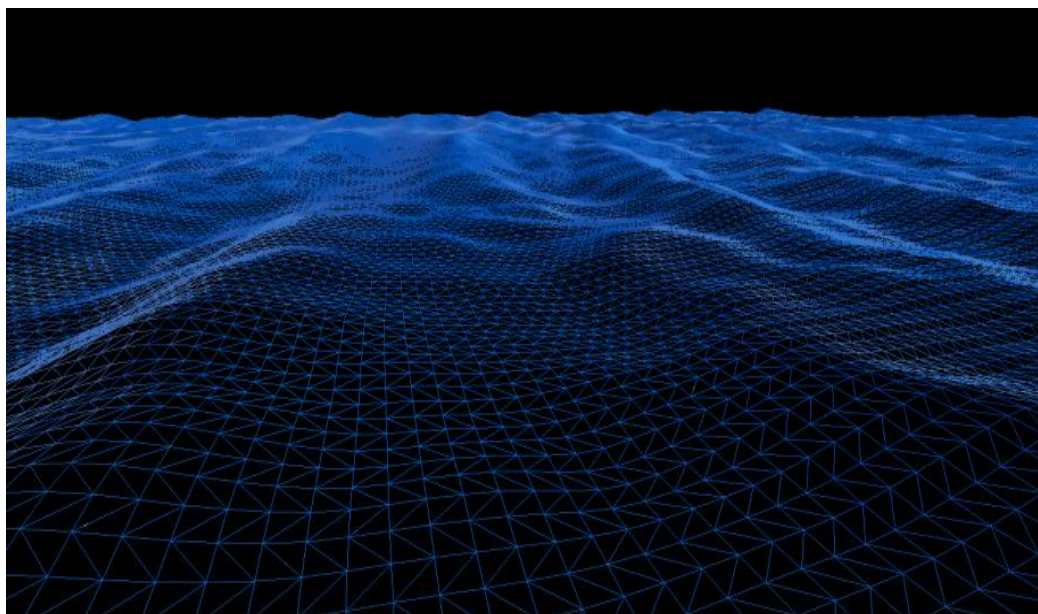
За да се увеличи детайлността на водната повърхност може да се използва шум. Идеята е да се синтезира достатъчно произволна поредица от изменения на даден параметър. На базата на хоризонтална позиция на точка от повърхността, описана от точките x_0 и z_0 , и времето се изчислява отместване по вертикалата.

3. Шум на Перлин

Шумът на Перлин е компютърно генериран, визуален ефект, който най-често се използва за симулиране на природни явления като облаци, дим, вода и др. Идеята е да се

синтезира последователност от стойности в интервала $[-1, 1]$. Тези стойности изглеждат произволни, но всъщност всяка следваща зависи от околните. Така преминаването от една точка към следващата става плавно. Последователността от тези точки, може да бъде създадена за определена резолюция и след това да бъде многократно повтаряна, без да се получават т. нар. артефакти при достигане на края и започване от началото на последователността.

Тук използваме тримерен шум, като третото измерение е времето. Така се изчислява стойност за шум, която се използва за анимиране на височината на вълните. По този начин освен големите синусоидни вълни на Герстнер, повърхността се насища с повече на брой по-малки вълни. Тъй като основната цел на алгоритъма на Перлин е да осигури плавно преминаване от една точка към друга, то той се явява подходящ за анимирането на малки, динамични вълни (фигура 3).



Фиг. 3. Вълни на Герстнер и шум на Перлин

Недостатък на този алгоритъм е, че при сравнително малка резолюция се забелязва повтораемост. Резолюция от 256 елемента се оказва достатъчна, за да не се забелязва тази повтораемост и да се постигне добра производителност на алгоритъма.

4. Недостатъци при използването на алгоритъма на Герстнер и алгоритъма на Перлин за генерирането на вълни

Основен недостатък при вълните на Герстнер е, че те са синусоидни вълни. Ако броят на използваните вълни не е достатъчно голям, то получения резултат няма да бъде реалистичен. Големият брой вълни обаче, може значително да намали производителността.

Друг недостатък на шума на Перлин е това, че в близост до наблюдателя липсва детайл.

Изчисленията на новата позиция на всяка една точка от графичния модел се извършва на ниво връх. Поради тази причина, шумът на Перлин се явява значително тежък за изчисление.

Един от начините за оптимизация е да се разпределят изчисленията на отделните елементи между процесора и видео картата. Това, обаче, е пряко свързано с конкретната цел на приложението и ресурсите, необходими за неговата безотказна работа.

5. Изграждане на примерна сцена с използване Cg на и OpenGL

Езиците за програмиране, използвани за изграждане на сцената са C/C++ и Cg. Като интерфейс за комуникация между графичната карта и процесора се използва OpenGL.

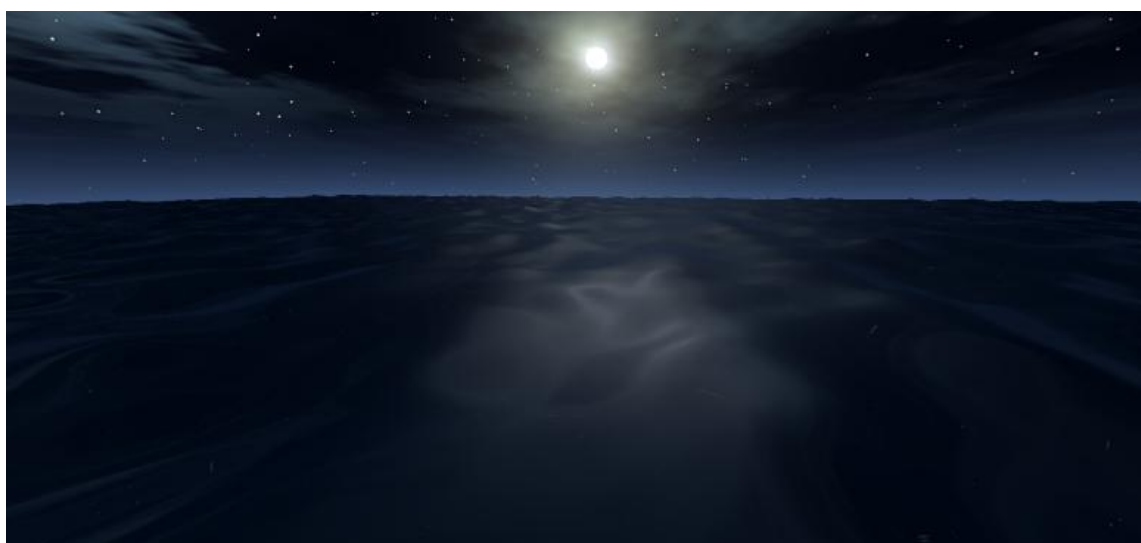
За да изглежда по-реалистично движението на вълните в сцената, те трябва да се влияят от някои фактори, но в същото време да се различават една от друга. Фактор, който влияе на големината на вълните, тяхната скорост и стръмност, е вятърът, описан с посока и сила. Към посоката на вълните е добавен произволен елемент в определени граници, чрез който всяка следваща вълна се завърта под произволен ъгъл спрямо вектора, описващ посоката на вятъра:

$$\begin{aligned} \text{rotFactor} &= \text{angleDeviation} * \frac{\pi}{180} \\ r_x &= \sin(\text{rotFactor}) \\ r_y &= \cos(\text{rotFactor}) \\ \text{waveDir}_x &= \text{windDir}_x * r_x + \text{windDir}_y * (-r_y) \\ \text{waveDir}_y &= \text{windDir}_x * (-r_y) + \text{windDir}_y * r_x \end{aligned} \quad (4)$$

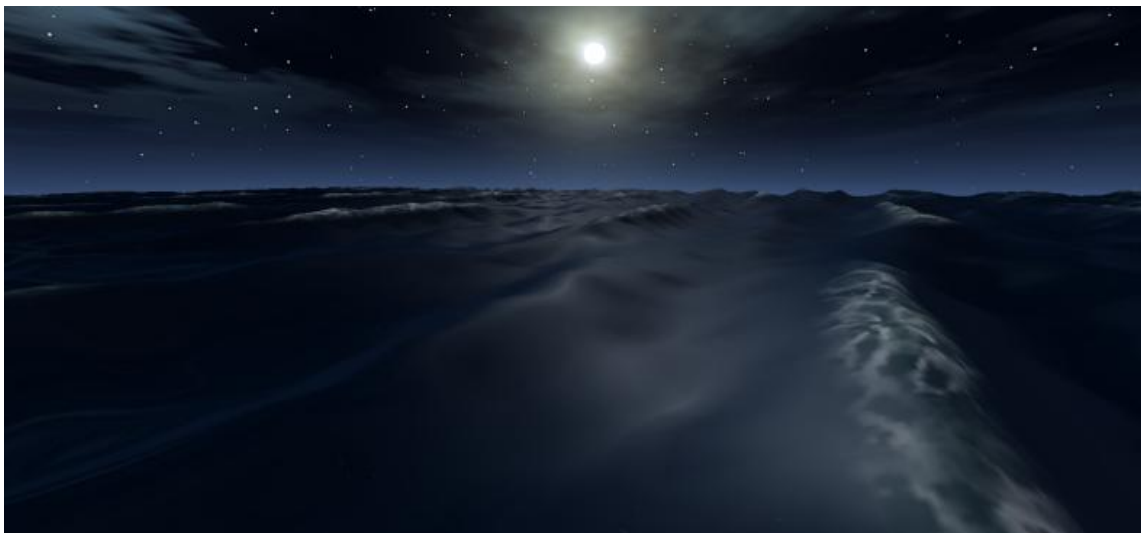
Променянето на параметрите на вълните, които вече се движат по повърхността на океана е значително нереалистичен подход, който може да доведе до появата на артефакти. Тук се използват 4 вълни и за всяка една от тях има определен живот. Той представлява интервала от появата на вълната, до нейното разливане. Наличието на такъв период позволява при разливане на вълната да се създаде нова с различни параметри. Точно по този начин се постига значително реалистичен ефект при движението на вълните по повърхността, като това позволява да не се забелязва повторемостта в алгоритъма.

Тук използваме една композиция от шумове на Перлин – „Брауново движение” (турболенция). Идеята е да намери сумата от шумове на Перлин. Всеки шум се явява отделен слой, като всеки слой се мащабира на базата на амплитудата. Подадените координати, като входни параметри на функцията за шум, се увеличават на базата на честотата. Честотата и амплитудата нарастват с предварително зададени коефициенти, при всеки следващ слой. Колкото по-голям е броя на слоевете, толкова повече са детайлите, изградени на базата на шума, тоест броя на слоевете увеличава броя на малките вълни по повърхността.

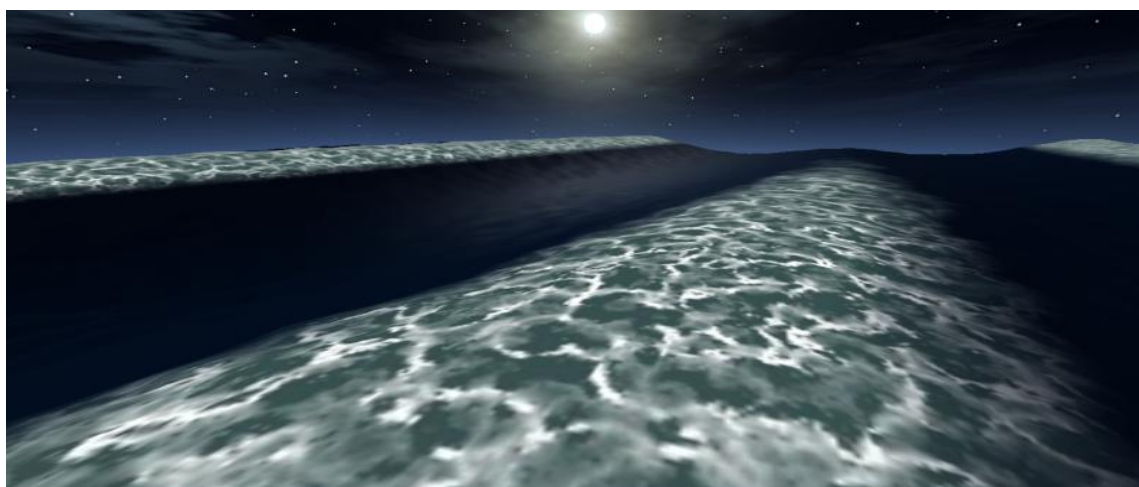
Получени резултати при различни стойности на параметрите L_{min} , L_{max} , $windSpeed$, $reflectivity$, $ampGain$ фигури 4а,б,в,г.



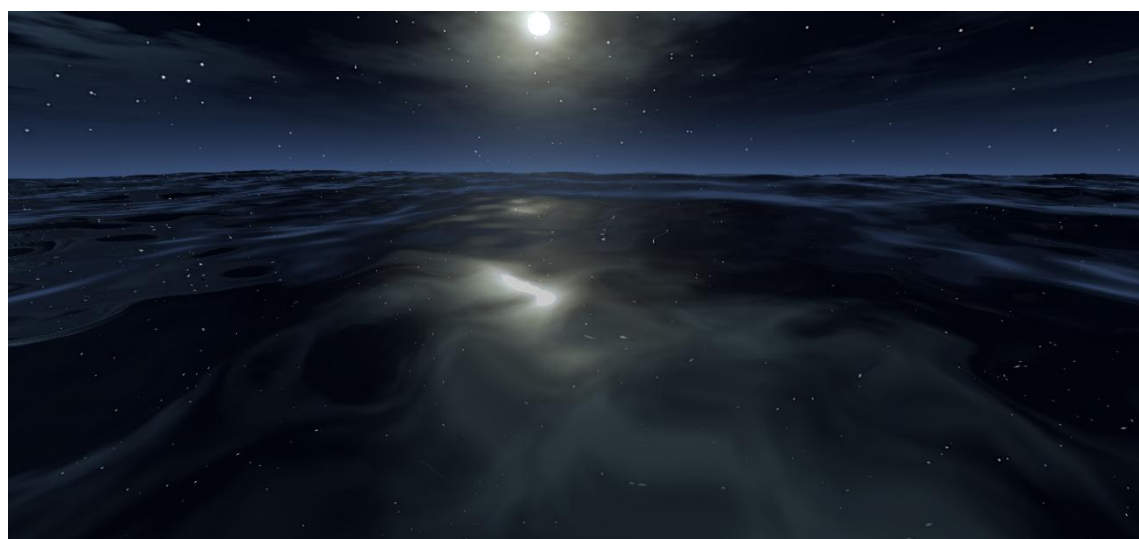
Фиг. 4а. $L_{min}=5$ $L_{max}=10$ $windSpeed=0,1$, $reflectivity=0,2$, $ampGain=0$



Фиг. 4б. $L_{\min}=,15$ $L_{\max}=,30$ windSpeed=3, reflectivity=0.13, ampGain=0.3



Фиг. 4в. $L_{\min}=,50$ $L_{\max}=,100$ windSpeed=25, reflectivity=0.05, ampGain=2



Фиг. 4г. $L_{\min}=,10$ $L_{\max}=,20$ windSpeed=0, reflectivity=1, ampGain=0

6. Изводи

Алгоритъмът на Герстнер е лесен за изчисление, което го прави подходящ за симулиране на вода в компютърни игри. Недостатък на този алгоритъм е не голямото разнообразие в получените резултати, особено когато се използват малко на брой вълни.

Алгоритъмът на Перлин е един от най-често използваните алгоритми за синтезиране на шум в компютърните игри. Той намира приложение при решаването на редица проблеми, свързани с реалния свят. В игрите най-често се използва за динамично генериране на „височинна карта“ (heightmap) на терени и различни видове повърхности.

При тези модели, височината на вълната се явява произволна стойност, получена от функция на хоризонталната позиция и времето: $h(x, t)$. Друга особеност на статистическите модели е способността им да разделят височината на вълната на синусоидни и косинусоидни вълни. За тази декомпозиция се използват бързи трансформации по Фурие, но те са обект на последващо разглеждане.

Литература

- [1]. NVIDIA Corporation, „The Cg Tutorial,“ 2007. http://http.developer.nvidia.com/CgTutorial/cg_tutorial_frontmatter.html.
- [2]. NVIDIA Corporation, „GPU Gems,“ 2004. http://http.developer.nvidia.com/GPUGems/gpugems_ch01.html.
- [3]. J. Tessendorf, „Simulating Ocean Water,“ 2001. <http://graphics.ucsd.edu/courses/rendering/2005/jdewall/tessendorf.pdf>.
- [4]. NVIDIA Corporation, „Ocean Surface Simulation,“. https://developer.nvidia.com/sites/default/files/akamai/gamedev/files/sdk/11/OceanCS_Slides.pdf.
- [5]. L. Vandevenue, „Texture Generation using Random Noise,“ 2004. http://lodev.org/cgtutor/random_noise.html.
- [6]. H. Homb, P. Johansen и K. S. Pedersen, „Water simulation with Cg,“ 2006. <http://tihlde.org/~pedjoha/prosjekter/Cg/waterreport.pdf>

За контакти:

докторант Даниела Д. Илиева
Технически университет – Варна
Катедра „Компютърни науки и технологии“
E-mail: ilievadaniela@mail.bg



**СОФТУЕРНИ И ИНТЕРНЕТ
ТЕХНОЛОГИИ**

СРАВНИТЕЛЕН АНАЛИЗ МЕЖДУ НАИВНИЯ БЕЙСОВ КЛАСИФИКАТОР И МЕТОДА НА ОПОРНИТЕ ВЕКТОРИ ИЗПОЛЗВАЩ ОПТИМИЗАЦИЯ ПРИ КЛАСИФИКАЦИЯ НА БЪЛГАРСКИ ТЕКСТ В МАШИННОТО ОБУЧЕНИЕ

Нели Ан. Арабаджиева – Калчева, Недялко Н. Николов

Резюме: Целта на публикацията е изследване и сравнителен анализ на алгоритмите: Наивен Бейсов класификатор (NB) и метод на опорните вектори с оптимизация (SMO) за класификация на български текст в машинното обучение. Резултатите са изведени аналитично и графично и показват, че при малък обем данни, двата алгоритъма дават точност по-висока от 90%. Когато текстът съдържа сходни думи, SMO е с по-високи стойности, докато при съществена разлика в терминологията, почти със същата преднина от 7.1% е NB. При разлика в броя на документите в пъти, NB дава по-голяма точност, докато в пъти разлика в броя на думи, SMO е с по-високи стойности. При класификация на текст при голям обем от данни точността и на двата алгоритъма е $\approx 55\%$, с малка преднина на SMO от $\approx 1.8\%$.

Ключови думи: Наивен Бейсов класификатор, метод на опорните вектори, SVM, SMO, класификация на текст, машинно обучение.

COMPARATIVE ANALYSIS OF THE NAIVE BAYES CLASSIFIER AND SEQUENTIAL MINIMAL OPTIMIZATION FOR CLASSIFYING TEXT IN BULGARIAN IN MACHINE LEARNING

Neli An. Arabadzieva – Kalcheva, Nedyalko N. Nikolov

Abstract: The goal of the publication is to research and comparatively analyze the algorithms: Naive Bayes Classifier (NB) and Sequential Minimal Optimization (SMO) for classification of text in Bulgarian. The results have been displayed analytically and graphically and show that with a small volume of data both algorithms have an accuracy rate of more than 90%. When the text contains similar words, SMO is more accurate, while when the texts differ a great deal the two almost exactly swap their accuracy, and NB leads by 7.1%. When the number of documents of different author in an order of magnitude, NB is more accurate, while when the difference is the number of words, SMO gives higher results. When classifying large volumes of data, both algorithms have an accuracy of $\sim 55\%$, with SMO leading by $\sim 1.8\%$.

Keywords: Naive Bayes classifier, Support Vector Machines, SVM, SMO, text classification, machine learning

1. Увод

През последните десетилетия улесненият достъп до интернет увеличи експоненциално онлайн информацията. Различни хранилища на знания, библиотеки, съдържат огромни масиви от данни. Невъзможността да се получи актуална и полезна информация за конкретна тема превръща в безполезна голяма част от съществуващите ресурси. Използването на класификация позволява да се съкрати трудността при търсене на нужната информация, представена като електронни текстове.

Ще разгледаме формална постановка на задачата за класификация. Нека е дадено крайно множество от класове $C=\{c_1, c_2, \dots, c_k\}$ и крайно множество от документи $D=\{d_1, d_2, \dots, d_k\}$. Целевата функция $f: D \times C \rightarrow \{0,1\}$ за всяка двойка <документ, клас> е неизвестна. Необходимо е да се намери класификатор f' , т.е. функция, максимално близка към функцията f .

Основната задача на класификацията е приобщаването на даден документ към определен клас в съответствие с набор от признаци. Традиционно в качеството на признаци се използват честотата на срещане на думите.

2. Изложение

2.1. Наивен Бейсов класификатор

Един от класическите алгоритми в машинното обучение е Наивният Бейсов класификатор, който се базира на теоремата на Бейс за определяне на апостериорната вероятност за настъпване на дадено събитие. Приемайки „наивното” предположение за условна независимост между всяка двойка атрибути, Наивният Бейсов класификатор се справя ефективно с твърде голямото количество на атрибутите за описание на един пример, т.е. с т.нар. „проклятие на размерността”.

Теорема на Бейс [1]

$$P(y = c|x) = \frac{P(x|y=c)P(y=c)}{P(x)} \quad (1)$$

където: $P(y=c|x)$ е вероятността обект да принадлежи на клас c (апостериорна вероятност на класа)

$P(x|y=c)$ – вероятността обекта x да среща в средата на обекта на класа c

$P(y=c)$ – безусловна вероятност да се среща обект y в клас c (априорна вероятност на класа)

$P(x)$ – безусловна вероятност на обекта x

Целта на класификацията се състои в това, да се определи към какъв клас принадлежи обекта x . Следователно е необходимо да се намери вероятностен клас на обекта x , т.е. необходимо е от всички класове да се избере този, който дава максимална вероятност $P(y=c|x)$.

$$c_{opt} = \arg \max_{c \in C} P(x|y = c)P(y = c) \quad (2)$$

2.2. Метод на опорните вектори - SVM (Support Vector Mashine).

Методът на опорните вектори (SVM, Support Vector Machines) представя обучаващите примери като точки в n -мерно пространство. Примерите са проектират в пространството по такъв начин, че да бъдат линейно разделими. При работа с два класа се търси начин да се начертае линия, която да разделя данните от двата класа. Линията, която разделя данните, се нарича разделителна хипер равнина. Тази хипер равнина трябва да се избере по такъв начин, че да се намира възможно най-далеч от примерите и на двата класа.

Функцията $f(x)$ на линейната класификация е във вида: [2]

$$f(x) = w^T x + b \quad (3)$$

където: w^T е тегловен вектор, а b е отклонението

След като се определи, къде да бъде поставена разделителната линия, се калкулира допустимата границата:

$$label * (w^T x + b) \quad (4)$$

Целта е да се намерят стойностите на w^T и b , които ще определят класификатора. За да се направи това, е необходимо да се намерят точките с най-малко отклонение, който трябва да се максимизира. Това може да се запише по този начин:

$$\arg \max_{w,b} \left\{ \min_n \text{label} * (w^T + b) \cdot \frac{1}{\|w\|} \right\} \quad (5)$$

Решаването на този проблем директно е доста трудно, затова се налага оптимизиране на производението вътре във фигурните скоби. Такива оптимизационни задачи могат да се решат с множители на Лангранж. Функцията за оптимизация е следната:

$$\max_{\alpha} \left[\sum_{i=1}^m \alpha - \frac{1}{2} \sum_{i,j=1}^m \text{label}^{(i)} \cdot \text{label}^{(j)} \cdot a^i \cdot a^j \cdot \langle x^{(i)}, x^{(j)} \rangle \right] \quad (6)$$

При спазване на следните ограничения:

$$c \geq \alpha \geq 0 \quad (7)$$

$$\sum_{i=1}^m \alpha_i \cdot \text{label}^{(i)} = 0 \quad (8)$$

При нелинейно разделими данни основната идея е да се постигне линейно разделение чрез преминаване на данните в друго по-високо измеримо функционално пространство чрез трансформация с функция на входните нелинейни данни. Това се осъществява от така наречената функция на ядрото K , която се дефинира по следния начин:

$$K(x_i, x_j) = f(x_i) \cdot f(x_j) \quad (9)$$

Някои от най-използваните функции на ядрото са:

- Полиномиална функция на ядрото;
- Гаусова радиална базисна функция;
- Експоненциална радиална базова функция;
- Многопластов перцептрон;
- Би Сплайн функция;
- Добавени ядра.

Модификация на алгоритъма, използващ Метода на опорните вектори, е т.нар. SMO (Sequential Minimal Optimization), който на всяка стъпка, с цел оптимизация, избира два множителя на Лангранж. Този алгоритъм е по-бърз и има по-добри свойства за мащабиране от стандартния SVM алгоритъм. [3]

2.3. Оценка на точността на класификацията

В процеса на класификация на текст се използват обучителни и тестови данни. Обучителните данни „обучават“ тестовите данни. За оценяване на класификатора се прилага метода кръстосано валидиране (cross-validation). Процесът използва всеки път различна извадка и се повтаря десет пъти. При крос валидацията се изчисляват основно точност (Accuracy), прецизност (Precision), пълнота (Recall) и комбинираната мярка F-measure. Точността (Accuracy) е процентът на правилно класифицирани примери от цялото тестово множество. Прецизността (Precision) е мярка, която дава доколко е точен класификатора и дава процентът на правилно класифицираните примери от съответния клас. Recall показва пълнотата или чувствителността на класификатора, т.е. колко от всички примери от даден клас е успял да определи като принадлежащи на този клас. Мярката F-measure, която комбинира Precision и Recall е тяхното претеглено хармонично средно. Трудно се постигат добри стойности и за Precision и за Recall. При класификация може да се постигне висока степен на Recall или на Precision, но не и на двете. Затова изборът за целеви стойности е по-труден и не се очаква постигане на изключително добри стойности и за двете. [2]

2.4. Изследване, получени резултати и анализ

В изследването е използван програмният пакет WEKA, който е софтуер с отворен код, издаден под общ лиценз на GNU. Анализираните алгоритми са: Наивен бейсов класификатор и Метод на опорните вектори, използващ оптимизация SMO с полиномиална функция на ядрото.

В доклада са въведени следните популярни съкращения на използваните алгоритми в текста:

- Наивен бейсов класификатор – NB (Naive Bayes)
- Метод на опорните вектори използващ оптимизация – SMO (Sequential Minimal Optimization)

Първоначално са класифицирани двама български автори - Пейо Яворов и Димчо Дебелянов с по 21 стихотворения и приблизително равен брой думи – 2018 думи за първия автор и 2031 думи за втория автор. Двамата поети са живели и творили в приблизително еднакъв период – края на 19 и началото на 20 век.

Резултатите показват, че **SMO** 100% правилно е класифицирал авторите, а разликата с NB е 7.1429%. (таблица 1)

Таблица 1. Класификация чрез NB и SMO на 2 автора с по 21 стихотворения и с ≈ 2000 броя думи

Автори	Брой стихотворения	Брой думи	Брой правилно класифицирани стихотворения	
			NB	SMO
Пейо Яворов	21	2018	21	21
Димчо Дебелянов	21	2031	18	21
Точност - Процент правилно класифицирани стихотворения			92.857%	100 %

При почти двойно увеличаване броя на думите и стихотворенията (таблица 2), Наивния Бейсов класификатор отново е на втора позиция след SMO. Наблюдава се съществена разлика в резултата на SMO с предходното изследване – 6.25 %, докато тази стойност при NB е едва 0.3571%.

Таблица 2. Класификация чрез NB и SMO на два автора с \approx равен брой стихотворения и с ≈ 4000 броя думи

Автори	Брой стихотворения	Брой думи	Брой правилно класифицирани стихотворения	
			NB	SMO
Пейо Яворов	38	4024	36	35
Димчо Дебелянов	42	4017	38	40
Точност - Процент правилно класифицирани стихотворения			92.5 %	93.75 %

При повторно почти двойно увеличаване броя на думите (таблица 3), сумарно вече 16400 думи, NB показва по-добри резултати и е с 4.8267 % повече от SMO. Наивният Бейсов класификатор с повече от 8% намалява резултата си спрямо предходните две изследвания.

Таблица 3. Класификация чрез NB и SMO на два автора с \approx равен брой стихотворения и с ≈ 8000 броя думи

Автори	Брой стихотворения	Брой думи	Брой правилно класифицирани стихотворения	
			NB	SMO
Пейо Яворов	73	8023	55	60
Димчо Дебелянов	68	8017	55	52
Точност - Процент правилно класифицирани стихотворения			84.2593 %	79.432%

При двойно увеличаване на броя на авторите от два на четири (таблица 4), т.е. броя класове с приблизително равен брой думи (≈ 2000), с по-добри резултати е NB и е с преднина от 8.4338%.

Таблица 4. Класификация чрез NB и SMO на 4 автора с \approx равен брой стихотворения и с ≈ 2000 броя думи

Автори	Брой стихотворения	Брой думи	Брой правилно класифицирани стихотворения	
			NB	SMO
Пейо Яворов	21	2018	14	15
Димчо Дебелянов	21	2031	18	15
Христо Фотев	20	2022	18	18
Петко Славейков	21	2000	16	11
Точност - Процент правилно класифицирани стихотворения			79.5181%	71.0843%

При увеличаване на броя на авторите на осем (таблица 5), процентите на правилно класифицираните стихотворения на двата класификатора са равни.

Таблица 5. Класификация чрез NB и SMO на 8 автора с \approx равен брой стихотворения и с ≈ 2000 броя думи

Автори	Брой стихотворения	Брой думи	Брой правилно класифицирани стихотворения	
			NB	SMO
Пейо Яворов	21	2018	14	13
Димчо Дебелянов	21	2031	10	12
Христо Фотев	20	2022	15	17
Петко Славейков	21	2000	14	11
Пенчо Славейков	20	2020	8	9
Гео Милев	21	2000	7	8
Любен Каравелов	19	2020	12	13
Николай Лилиев	21	2000	19	16
Точност - Процент правилно класифицирани стихотворения			60 %	60 %

При класифициране на други 2 български автора Петко Славейков и Николай Лилиев с по 21 стихотворения и с по 2000 брой думи, живели и творили в различен период : първия - средата и края на 19 век, а втория - началото и средата на 20 век се получават коренно различни резултати (таблица 6). Наивният Бейсов класификатор не е разпознал само едно

стихотворение, докато SMO е сгрешил с три стихотворения. Точността на NB е с 7.1957% по-висока от тази на SMO.

Таблица 6. Класификация чрез NB и SMO на други 2 автора с по 21 стихотворения и с 2000 броя думи

Автори	Брой стихотворения	Брой думи	Брой правилно класифицирани стихотворения	
			NB	SMO
Петко Славейков	21	2000	20	20
Николай Лилиев	21	2000	21	18
Точност - Процент правилно класифицирани стихотворения			97.6719 %	90. 4762 %

В изследването (таблицы 1, 2, 3, 4, 5, 6) броят стихотворения и броят думи е приблизително равен при всеки един от авторите. В пъти по-малко на брой стихотворения и равен брой думи (таблица 7) резултата е в полза на NB. Например, при 5 на брой стихотворения на Пейо Яворов и 21 на Димчо Дебелянов (≈ 2000 думи), точността на NB е по-голяма спрямо SMO със 7.6923%. При тройно увеличаване на броя на стихотворения на първия автор на 15 и двойно увеличаване на броя стихотворения на втория автор на 42 (≈ 4000 думи), отново NB се справя по-добре, но с по-малка разлика от 3.5087%.

Таблица 7. Класификация чрез NB и SMO на 2 автора с \neq брой стихотворения и с \approx равен брой думи

Автори	Брой стихотворения	Брой думи	Брой правилно класифицирани стихотворения	
			NB	SMO
Пейо Яворов	5	2018	2	0
Димчо Дебелянов	21	2031	21	21
Точност - Процент правилно класифицирани стихотворения			88.4615%	80.7692%
Пейо Яворов	15	4020	6	4
Димчо Дебелянов	42	4017	42	42
Точност - Процент правилно класифицирани стихотворения			84.2105%	80.7018 %

Таблица 8. Класификация чрез NB и SMO на 2 автора с равен брой стихотворения и с \neq брой думи

Автори	Брой стихотворения	Брой думи	Брой правилно класифицирани стихотворения	
			NB	SMO
Пейо Яворов	20	5129	11	16
Димчо Дебелянов	20	1937	20	20
Точност - Процент правилно класифицирани стихотворения			77.5%	90 %
Пейо Яворов	40	9856	24	40
Димчо Дебелянов	40	4017	39	38
Точност - Процент правилно класифицирани стихотворения			84.2105%	97.5 %

В случая, когато броят на стихотворенията е равен, а има разлика в броя на думите в пъти (таблица 8), по-точен е SMO и то със значителна преднина от 12.5% при първото изследване и 13.2895% при второто. Двата класификатора имат по-висок процент на правилно класифицирани стихотворения при по-голям брой думи.

При увеличаване на броя на авторите на 20, стихотворенията на 591, думите на 103398 и не съблюдаване на равенство на брой стихотворения и думи за веки един от авторите и двата класификатора не дават добри резултати. Точността на Наивния Бейсов класификатор е 54.8223 % (фигура 1), а точността на метода на опорните вектори с оптимизация е 56.6836 % (фигура 2). Precision, Recall и F-measures и при двата класификатора са с приблизително равни стойности ≈ 0.55 . Параметърът Kappa Statistic, който показва степента на съгласуваност между предсказването и реалния клас и при двата класификатора е ≈ 0.5 (много по-ниска стойност от 1, съответстваща на идеалната съгласуваност).

Отношението между правилно и неправилно класифицираните обекти за генерирания модел на класификация чрез Наивния Бейсов класификатор е: ≈ 1.21 , а чрез метода на опорните вектори с оптимизация е ≈ 1.26 . Получената приблизителна стойност 1.21, съответно за SMO 1.26 показва, че броят на правилно класифицираните записи е приблизително 1.21 пъти, респективно 1.26, по-голям от броя на неправилно класифицираните записи. Тези резултати показват, че на всеки 1.21 за NB и съответно 1.26 за SMO, правилно класифицирани записи се получава една грешка.

Разликата между получените отношения Правилно/Неправилно класифицирани записи за двата класификатора е много малка и е 0.05.

```
Time taken to build model: 5.66 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances      324          54.8223 %
Incorrectly Classified Instances    267          45.1777 %
Kappa statistic                    0.5135
Mean absolute error                 0.0457
Root mean squared error             0.2045
Relative absolute error             48.9061 %
Root relative squared error         94.6387 %
Total Number of Instances          591

=== Detailed Accuracy By Class ===
```

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0,476	0,026	0,400	0,476	0,435	0,414	0,958	0,434	Botev
	0,739	0,036	0,729	0,739	0,734	0,698	0,939	0,765	PYavorov
	0,500	0,065	0,357	0,500	0,417	0,373	0,842	0,429	DDebelianov
	0,118	0,002	0,667	0,118	0,200	0,273	0,902	0,398	DChintalov
	0,640	0,076	0,438	0,640	0,520	0,477	0,877	0,523	GMilev
	0,605	0,000	1,000	0,605	0,754	0,766	0,960	0,865	HFotev
	0,600	0,002	0,938	0,600	0,732	0,742	0,992	0,887	LKaravelov
	0,300	0,054	0,286	0,300	0,293	0,240	0,802	0,324	PenchoSlavejkov
	0,000	0,000	0,000	0,000	0,000	0,000	0,987	0,779	GRakovski
	0,523	0,027	0,605	0,523	0,561	0,530	0,896	0,594	HSmirnenski
	0,533	0,000	1,000	0,533	0,696	0,726	0,997	0,941	KMiladinov
	0,659	0,059	0,475	0,659	0,552	0,518	0,874	0,603	KVelichkov
	0,561	0,049	0,460	0,561	0,505	0,467	0,892	0,472	PetkoSlavejkov
	0,300	0,000	1,000	0,300	0,462	0,544	0,953	0,408	RGenzifov
	0,757	0,023	0,683	0,757	0,718	0,699	0,929	0,716	SStambolov
	0,000	0,000	0,000	0,000	0,000	0,000	0,819	0,229	SMihajlovski
	0,000	0,000	0,000	0,000	0,000	0,000	0,851	0,385	VasilPopovich
	0,500	0,035	0,394	0,500	0,441	0,415	0,959	0,578	MaraBelcheva
	0,765	0,031	0,605	0,765	0,675	0,658	0,954	0,764	NLiliev
	0,333	0,000	1,000	0,333	0,500	0,574	0,944	0,722	SirakSkitnik
Weighted Avg.	0,548	0,034	0,576	0,548	0,539	0,520	0,910	0,607	

Фиг. 1. Weka класификация чрез NB на 20 автора с \neq брой стихотворения и с \neq брой думи

Time taken to build model: 4.03 seconds

=== Stratified cross-validation ===

=== Summary ===

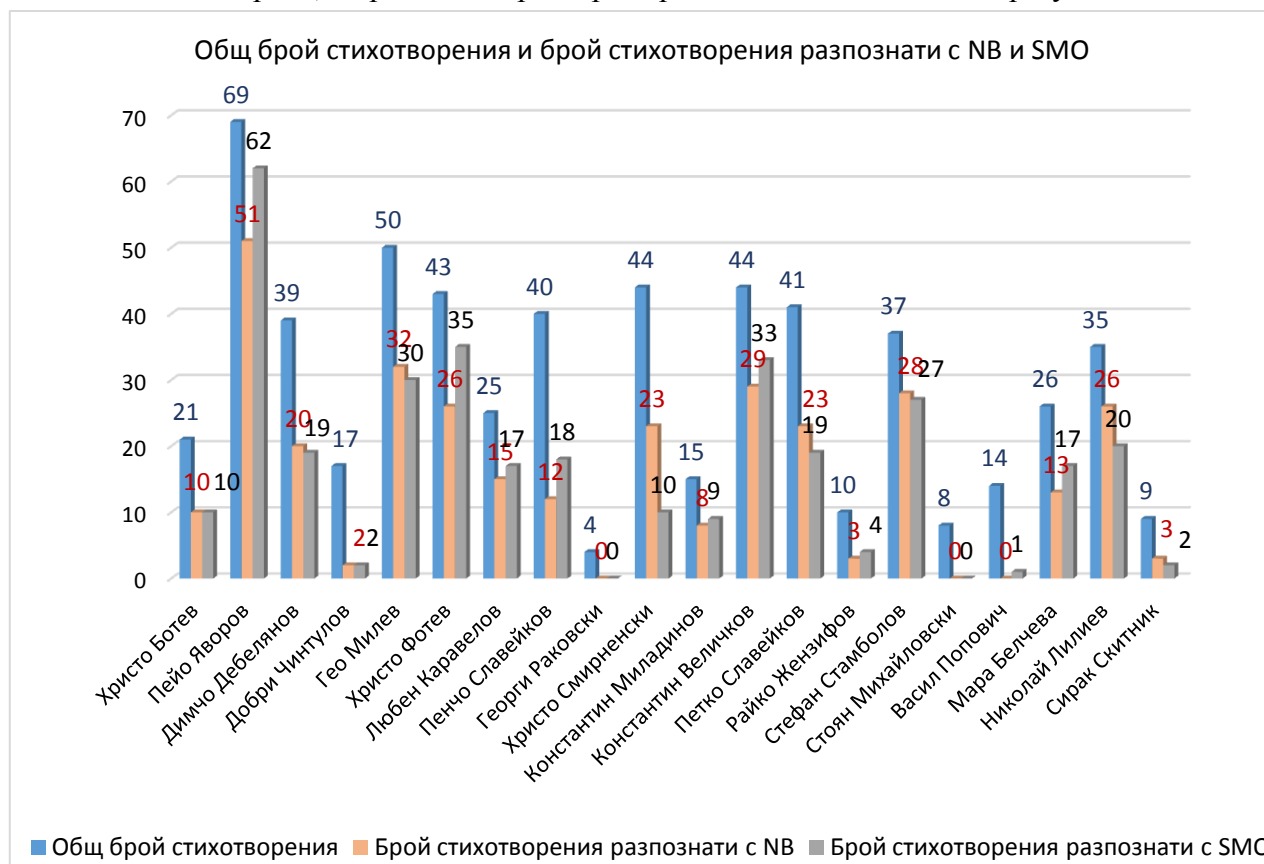
Correctly Classified Instances	335	56.6836 %
Incorrectly Classified Instances	256	43.3164 %
Kappa statistic	0.5315	
Mean absolute error	0.0911	
Root mean squared error	0.211	
Relative absolute error	97.4809 %	
Root relative squared error	97.6314 %	
Total Number of Instances	591	

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0,476	0,016	0,526	0,476	0,500	0,483	0,930	0,388	Botev
	0,899	0,102	0,539	0,899	0,674	0,647	0,921	0,524	Pyavorov
	0,475	0,042	0,452	0,475	0,463	0,424	0,830	0,325	DDebelianov
	0,118	0,002	0,667	0,118	0,200	0,273	0,790	0,275	DChintalov
	0,600	0,085	0,395	0,600	0,476	0,428	0,841	0,313	GMilev
	0,814	0,018	0,778	0,814	0,795	0,779	0,955	0,708	HFotev
	0,680	0,002	0,944	0,680	0,791	0,794	0,967	0,754	LKaravelov
	0,450	0,047	0,409	0,450	0,429	0,385	0,827	0,274	PenchoSlavejkov
	0,000	0,000	0,000	0,000	0,000	0,000	0,499	0,007	GRakovski
	0,227	0,013	0,588	0,227	0,328	0,337	0,808	0,295	HSmirnenski
	0,600	0,000	1,000	0,600	0,750	0,771	0,972	0,893	KMiladinov
	0,750	0,049	0,550	0,750	0,635	0,609	0,929	0,476	KVelichkov
	0,463	0,029	0,543	0,463	0,500	0,468	0,887	0,363	PetkoSlavejkov
	0,400	0,000	1,000	0,400	0,571	0,629	0,852	0,466	RGenzifov
	0,730	0,023	0,675	0,730	0,701	0,681	0,931	0,577	SStambolov
	0,000	0,000	0,000	0,000	0,000	0,000	0,743	0,079	SMihajlovski
	0,071	0,000	1,000	0,071	0,133	0,264	0,778	0,195	VasilPopovich
	0,654	0,028	0,515	0,654	0,576	0,559	0,940	0,419	MaraBelcheva
	0,588	0,014	0,714	0,588	0,645	0,629	0,933	0,554	NLiliev
	0,222	0,000	1,000	0,222	0,364	0,469	0,928	0,622	SirakSkitnik
Weighted Avg.	0,567	0,037	0,599	0,567	0,544	0,532	0,886	0,448	

Фиг. 2. Weka класификация чрез SMO на 20 автора, всеки с \neq брой стихотворения и с \neq брой думи

Видно от фигура 3, двата класификатора разпознават различен брой стихотворения на всеки един от авторите, а при малък брой примери не дават положителен резултат.



Фиг. 3. Разпределение на общ брой стихотворения и брой стихотворения разпознати с NB и SMO по автори

3. Заключение

Алгоритъмът, използващ метода на опорните вектори с оптимизация с два Лангранж множителя и полиномиална функция на ядрото (SMO) при класифициране на текст с два класа, всеки с около 2000 думи и с 21 документа, дава разлика в точността повече от 7.14% от Наивния Бейсов класификатор (NB) както положителна, така и отрицателна в зависимост от различието в данните. Когато текста съдържа сходни думи, както е примера при двама автори живели и творили през един и същи период, SMO е с по-високи стойности, докато при съществена разлика в терминологията в текста, почти със същата преднина е NB. При почти двойно увеличаване на данните, SMO намалява точността си с 6.25%, докато тази разлика при NB е малко по-голяма от 0.35%. При повторно почти двойно увеличаване на данните, близо по 8000 думи от всеки клас, NB разпознава двата класа с точност малко по-голяма от 84% и се справя по-добре от SMO с преднина повече от 4.8%. При четири класа и почти равен брой документи и думи, Наивния Бейсов класификатор дава точност по-голяма от 79.5% и е с повече от 8.4% от SMO. При осем класа и приблизително равен брой документи и думи за всеки автор и двата класификатора дават точност 60%. При разлика в броя на документите в пъти, Наивният Бейсов класификатор дава по-голяма точност, докато в пъти разлика в броя на думи, SMO е с по-високи стойности. При голям брой данни – 20 класа, 591 документа, 103 398 думи и различен брой думи за всеки документ и клас, точността на SMO е 56.6836 % и е само с 1.8613% повече от Наивния Бейсов класификатор.

Проведените изследвания показват, че точността на предсказване на Наивния Бейсов класификатор и алгоритъма използващ метода на опорните вектори с оптимизация с два Лангранж множителя и полиномиална функция на ядрото, при два класа е над 90%, докато при класификация на текст при голям обем от данни не е достатъчно висока.

Литература:

- [1]. Уиткин Л.В. Машинное обучение (Machine Learning) Метрические методы классификации и регрессии, 2017, стр. 6-8
- [2]. Harrington P., Machine Learning in Action, 2012, стр. 105-106, стр. 144
- [3]. Platt J., Fast Training of Support Vector Machines using Sequential Minimal Optimization, 1998, стр.44

За контакти:

ас. Нели Ананиева Арабаджиева - Калчева
катедра „Софтуерни и интернет технологии”
Технически университет - Варна
E-mail: n_kalcheva@tu-varna.bg

доц. д-р Недялко Николаев Николов
катедра „Софтуерни и интернет технологии”
Технически университет - Варна
E-mail: ned.nikolov@tu-varna.acad.bg

DEVELOPMENT OF AIR TRAFFIC CONTROLLER TRAINING PROGRAMMING MODEL

Milena N. Karova, Anna D. Nikolova, Drumka N. Zhekova

Abstract: The article presents a programming model of simulator training for air traffic controllers. The development of the mathematical model includes approximation of arcs using cubic Bezier curves. The advantage is that the curve passes through both of the endpoints while the two intermediate points act as magnets which can easily control the shape of the curve. The model simulates the control of aircraft landing and aircraft take off, the change of trajectory, the communication with flight control center and more.

Keywords: Air Traffic Control, simulator, aircraft landing, aircraft take off, Bezier curves, aircraft trajectory, runway

РАЗРАБОТВАНЕ НА ПРОГРАМЕН МОДЕЛ НА СИМУЛАТОР ЗА ОБУЧЕНИЕ НА РЪКОВОДИТЕЛИ НА ПОЛЕТИ

Милена Н. Карова, Анна Д. Николова, Друмка Н. Жекова

Резюме: Статията представя програмен модел на симулатор за обучение на ръководители на полети. Разработката на математическия модел включва апроксимация на дъги с използване на кубични криви на Безие. Предимството е, че кривата преминава през двете крайни точки, докато двете междинни точки действат като магнити, които лесно могат да контролират формата на кривата. Моделът симулира контрола на излитането и кацането на самолета, промяната на траекторията, комуникацията с центъра за управление на полети и др.

Ключови думи: контрол на въздушното движение, симулатор, кацане на самолет, излитане на самолет, криви на Безие, траектория на самолета, писта за излитане и кацане.

1. Introduction

Air Traffic Control (ATC) or flight controller training is a complicated domain in which many specialists should communicate with each other, and manage an air traffic situation in a collaborative way in order to guarantee safe and efficient air traffic. It is necessary to train a large number of professionals working in the air traffic control area.

The objective of this research is to conceive, design, and implement an ATC system that can be easily used at schools, on courses or simply manage an air traffic situation.

The responsibilities of a flight controller are [2]:

- Communicate with the pilots, coordinate air traffic between control centers.
- Allow departure and landing, give implementation instructions and any other information for pilots.
- Recommend traffic route changes of planes flying in fog, storm or in urgent and other extraordinary situations.
- Transfer responsibility for aircraft taking off to air transport control center.
- Communicate with the plane crew and notify the aircraft center for problems in the field.
- Fill a daily report and store messages received by airplanes.
- Data analysis such as weather, fuel requirements and maps in order to organize air flights and routes.
- Announce reportedly flight information such as altitude flight, estimated time of arrival and route to the aircraft control center.

- Control the aircraft movement at the aerodrome.

There are a large number of ATC simulators providing a realistic simulation of operational environments to train air traffic controllers in managing aircraft operations and handling emergency situations. The advanced simulators facilitate realistic training in the full-spectrum of ATC operations – from approach and departure phases to the enroute phase of flight, as well as ground movement at the airport.

Examples of famous simulator systems are:

- ATC-SIM – web simulator (choosing the level of difficulty)
- SKY-HIGH – web simulator (professional skills are required)
- VATSIM – desktop application (pilot and air traffic control). This is one of the best ATC software.
- Mathematical Modeling of Aircraft Trajectory includes two constructors: one for usual trajectory change and one for landing. It uses Bezier curves: parametric curves given by controlling points. Bezier curves, and more specifically cubic Bezier curves, are a standard tool in many computer-aided design applications such as PhotoShop, InDesign, GIMP, Scribus, etc. The cubic Bezier curves controlling polygon consists of 4 points: start and end point and two intermediate ones. The advantage is that the curve passes through both of the endpoints while the two intermediate points act as magnets which can easily control the shape of the curve. Moreover, every given curve can be represented as a finite number of such four-point Bezier curves, connected to each other and its smoothness around the connection point can easily be controlled[4].

The presented Air Traffic Control simulator allows the following process:

- Choose and change the aircraft direction
- Choose and change the aircraft altitude
- Choose flight departure or landing
- Create aircraft activity reports

2. Mathematical Modeling of Aircraft Trajectories

Bezier curves are polynomial parametric curves. In the 1960s the French mathematicians Paul de Casteljaeu (Citroen) and Pierre Bezier (Renault), independent of each other, came to the idea of using parametric curves given by controlling polygons of points. Similar ideas existed as early as the twenties in the field of classical differential geometry, but those ideas never found an application at the time. Nowadays, the cubic Bezier curves are a standard tool in numerous Computer Aided Design (CAD) applications such as PhotoShop, InDesign, GIMP, Scribus, etc.

To define a Bezier curve of n -th order, one needs $n+1$ control points P_0, P_1, \dots, P_n and Bernstein's polynomials. On successive connection, these points form the controlling polygon of the curve. Bernstein's polynomials $B_{n,i}(u)$ are polynomials of a single real variable and have the form:

$$B_{n,i}(u) = \frac{n!}{i!(n-i)!} u^i (1-u)^{n-i}, \quad i = 0, 1, \dots, n \quad (1)$$

Then the Bezier curve $C(u)$ of n -th order, determined by the controlling points P_0, P_1, \dots, P_n , has the form:

$$C(u) = \sum_{i=0}^n B_{n,i}(u) P_i, \quad (2)$$

where $u \in [0,1]$.

Most widely used are Bezier curves of 3-rd order, given in the form

$$C(u) = \sum_{i=0}^3 B_{3,i}(u) P_i = (1-u)^3 P_0 + 3u(1-u)^2 P_1 + 3u^2(1-u) P_2 + u^3 P_3. \quad (3)$$

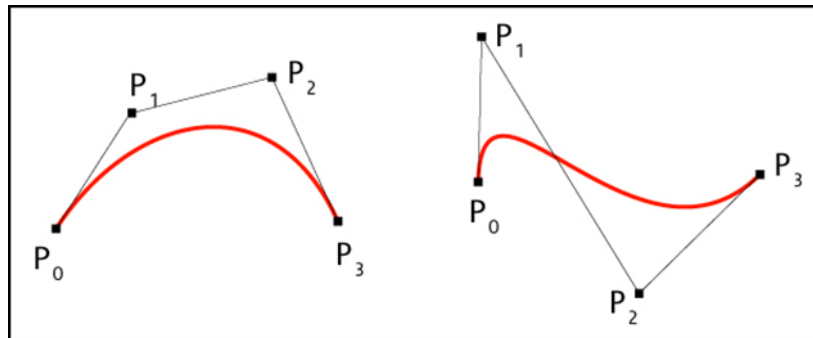


Fig.1. Cubic Bezier Curves

On Fig. 1 are given Bezier cubic curves, determined by 4 control points P_0, P_1, P_2, P_3 . The curves pass through the start and the end control points. These control points are found by setting the parameter to the limits of its domain $[0, 1]$, i.e. $C(0)=P_0$ and $C(1)=P_3$. Bezier curves are tangent to their first (P_0P_1) and last (P_2P_3) control lines. Every Bezier curve lies entirely in the convex hull of its control points.

Many practical problems require the approximation of circular arcs with Bezier curves. Let us define a circular arc with radius r and magnitude less than 90 degrees. The coordinate system is oriented as shown on Fig. 2, i.e. the positive direction of the x axis splits the arc in half.

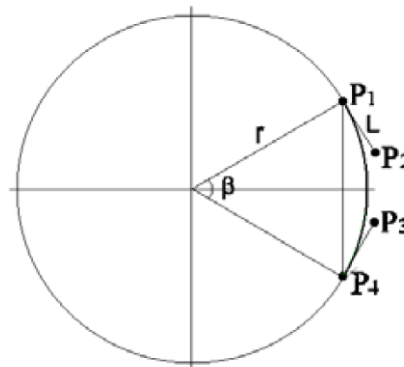


Fig.2. Approximation of a circular arc by cubic Bezier curve

The resulting Bezier curve connects P_1 and P_4 and its tangents at the ends are collinear with the vectors (P_1-P_2) and (P_4-P_3) . The variation of tangent magnitude L is within the domain $[0, r \tan \varphi]$, where φ is half the angular width of the arc segment [5], i.e. $\varphi = \frac{\beta}{2}$. Then the coordinates of the cubic Bezier control points P_2 and P_3 are [5]:

$$\begin{aligned} x_2 &= x_1 + k L \sin \varphi \\ y_2 &= y_1 - k L \cos \varphi \\ x_3 &= x_4 + k L \sin \varphi \\ y_3 &= y_4 + k L \cos \varphi \end{aligned} \quad (4)$$

where k is the so-called „magic number“, i.e. $k=0.5522847498$.

3. Architecture Design

The main project files which describes and declare different view screens, the main user class, the title of the project and login screen is App.java.

The controllers contain action functions, performed by the user.

- LoginController.java – controller, responsible for application initialization.
- MainController.java and FlightController.java - controllers responsible for the main panel (manage the flights). Controller functions are used for airport drawing, runs runaways, the boundary of the flight range, course changing etc.
- ReportController.java - controller for report writing.

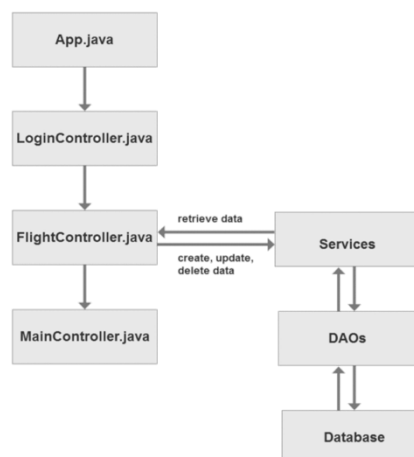


Fig. 3. Simulator system design

4. Graphical User Interface (GUI) Display

The user interface of the desktop application is built by FXML. The elements of each screen in the application are vector-based controls and graphics resource. The appearance of the application describes using a software called JavaFX Scene Builder. There is login panel with user name and password given in advance by the administrator.

There are several display options of the main panel during initialization: display area, dynamically drawn courses, runways (waypoints), flight zone (dotted red line), airport linseys and flight information [3]. All these functions are in MainController.java. Starting a timer to generate new flights and calling functions for drawing graphics is done in the constructor of FlightController.java. This controller also declares parameters and maximum altitude, fuel, altitude landing, etc.

There are different interface functions for:

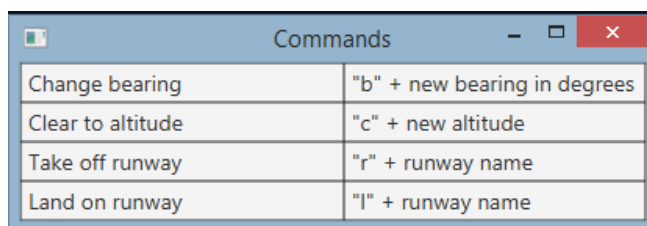
- 1) Changing flight labels (Fig. 4 Change Description())

<input type="text"/>		<input type="button" value="Transmit"/>
Flight	5QGJ	
Coordinates	660x375	
Bearing	270°	
Altitude	1700ft	
Origin	Varna Airport	
Destination	to Sofia Airport	
Fuel	8mins	
Plane	Boeing 767-336(ER) - 6000ft/min	

Fig. 4. Information fields

2) Introducing commands to control the flight with 4 modes (Fig.5, Fig. 6) (function `transmitButtonAction()` in `MainController.java`):

- "b" + new direction in degrees
- "c" + new altitude
- "r" + choice of runway
- "l" + choice of runway



Commands	
Change bearing	"b" + new bearing in degrees
Clear to altitude	"c" + new altitude
Take off runway	"r" + runway name
Land on runway	"l" + runway name

Fig.5. Commands list

- 3) Changing flight altitude, taking in mind that each type of aircraft has different parameters to change in the database. The timer which is active for each object will change the altitude every second. (Function `set FlightAltitude` Fig. 7)
- 4) Changing flight course –direction. the user must enter a valid direction in degrees. The functions in `mainController.java` calculate and draw a path in case of a changing course, taking the current and the new course, the angle of the bend and starting coordinates of the maneuver (1).
- 5) processing of flight takeoff - `takeOffRunway ()` function automatically rises to the minimum altitude and waits for instructions to be redirected to the correct runway
- 6) Flight landing – incoming flights should be directed to the runway for landing (function `LandOnRunway`). For landing process, the function needs landing altitude, the correct angle and the runway name.
- 7) Listing of reports – function `reportService.listReports ()`. List of reports of the activities of a dispatcher are taken from Reports table in the database. On Fig.8 is shown menu with some sample reports.

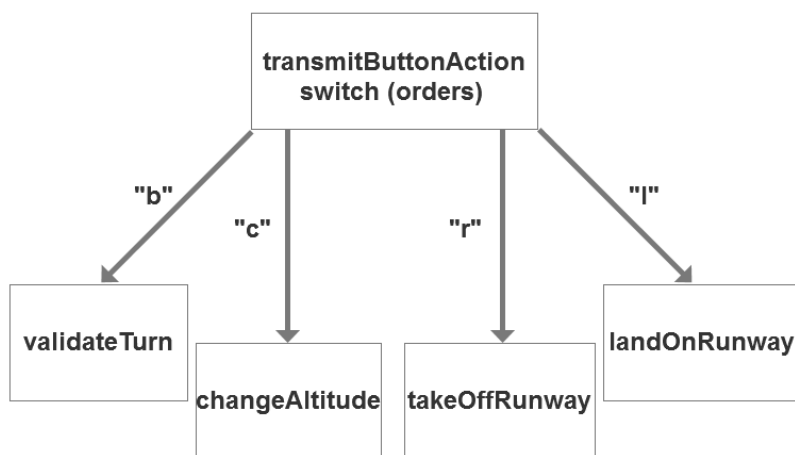


Fig.6. Command switch in `transmitButtonAction()`



Fig.7. Indicator of altitude change

Reports				
Positive reports : 3		Negative reports : 2		Crashes : 2
Flight	Origin	Destination	Altitude	Status
J7QI	Burgas Airport	Sofia Airport	5506ft	Did not reach destination
F7GT	Varna Airport	Sofia Airport	5401ft	Crashed without fuel
RMMI	Varna Airport	Sofia Airport	1700ft	Landed successfully
1Y45	Burgas Airport	Sofia Airport	4000ft	Crashed with flight #DPD3
OWQC	Sofia Airport	Burgas Airport	2700ft	Left our airspace successfully
JTP9	Burgas Airport	Sofia Airport	2966ft	Did not reach destination
ND9T	Sofia Airport	Burgas Airport	2700ft	Left our airspace successfully

Fig.8. Report List

5. Database Design

Data storage is designed on Oracle 11g XE database, the connection is performed by Hibernate ORM. For this purpose the hibernate.cfg.xml file must be configured with the correct parameters for the driver, address, port, username and password of the user.

There is a class that contacts with table from the database and adds annotations (the table name, data type and the relations between objects).

The DAO class is responsible for operations with the data service and runs the application (Fig 9).

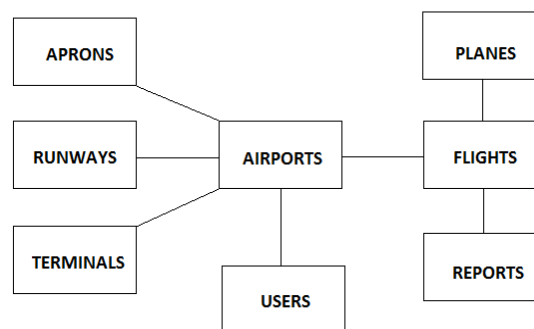


Fig.9. Database structure

The data in the tables Airports, Planes, Users, Terminals, Runways, Aprons are completed by the administrator through direct access to the database. In the future, additional admin panels could be built to facilitate the process. Flights and Reports Tables are completed recording reports during the simulation.

6. Modeling of Actual Aircraft Trajectories

In order to change the movement trajectory of a given flight, the user has to provide a valid direction in degrees. The input correctness is verified in validateTurn() in mainController.java. The actual calculation and the tracing of the path after a change in the direction is performed by the class TurnPath, located inside the customs directory. The class takes into account the current and the new direction, the angle of the turn and the start and end points of the trajectory.

In generateFlightPath(), the trajectory of the flight is changed by generating a new trajectory of type FlightPath. This class has two constructors: one for usual trajectory change and one for landing. The code for drawing the landing trajectory can be found on Fig.10. For this purpose, the

coordinates of the flight and of the landing track are used as start and end points. A standard formula for calculating the distance between two points is used for finding the value of transitionLength.

To generate a usual change of direction, we need to use the start X and Y and the degree of the new direction:

```
angle = Math.toRadians(newBearing - 90);
```

```
x = x + 2000 * Math.cos(angle);
```

```
y = y + 2000 * Math.sin(angle);
```

```
publicFlightPath(double startX, double startY, double endX, double endY,
PlaneElement oPlane) {
    LineTo lineTo = new LineTo(endX, endY);
    this.getElements().add(new MoveTo(startX, startY));
    this.getElements().add(lineTo);
    oPlane.transitionLength = Math.sqrt((startX-endX)*(startX-endX) +
```

Fig.10. Generation of landing trajectory

When the dispatcher requests a change in direction, first the argument correctness is verified by `validatTurn()` and then the method `changeBearing()` in `MainController.java` is invoked. The actual change is performed by `changeFlightBearing` (FIX THE SENTENCE!) The method takes into account the coordinates of the flight, the current direction and the new one in degrees. An algorithm for approximation of circular arcs by cubic Bezier curves is used. The algorithm calculates the endpoint and the two intermediate points by using (1), the original position coordinates and the requested turn angle. The latter are used for instantiation of an object of type `CubicCurvesTo`. The parameter r is the radius of the desired arc, a is the angular width of the arc segment and $k = 0.5522847498$ is the so called “magic number”. A 90-degree arc drawn using this algorithm is shown on Fig.11.

```
x1 = r * Math.cos(a/2);
```

```
y1 = r * Math.sin(a/2);
```

```
x4 = x1;
```

```
y4 = -y1
```

```
x2 = x1 + k * tan(a/2) * y1;
```

```
y2 = y1 + k * tan(a/2) * x1;
```

```
x3 = x2; y3 = -y2;
```

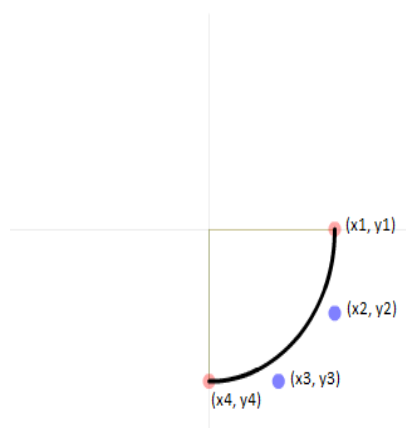


Fig.11. Approximation of arc by a cubic Bezier curve

After the turn is completed, the algorithm in FlightPath.java is used to draw the trajectory of the flight

7. Conclusions

This paper discusses the need and requirements for ATC computer simulations as automation training tool or school game. The complex Model of Actual Aircraft Trajectory is explained, using Bezier curves: parametric curves given by controlling points. The presented simulator could extend simulators capabilities: tower simulator, radar simulation, putting dynamical environment and weather modelling, piloting and other advanced ATM tools.

REFERENCES

- [1]. Erzberger H., "Design Principles and Algorithms for Automated Air Traffic Management," AGARD, Lecture Series No. 200 on Knowledge-Based, Functions in Aerospace Systems, Madrid, Paris and San Francisco, November 1995.
- [2]. Flight Planning and Management System, Software Requirements Specification, Version <1.0>, 2/20/2015
- [3]. Plaettner-Hochwarth, J, Zhao Y., A Modularized Approach For Comprehensive Air Traffic System, Simulation, American Institute of Aeronautics and Astronautics, 2000
- [4]. Yamaguchi F., Curves and Surfaces in Computer Aided Geometric Design. Springer Verlag, 1988.
- [5]. Riškus A., Approximation of a cubic Bezier curve by circular arcs and vice versa, Information technology and control, 2006, Vol.35, No.4

For contacts:

Assoc. Prof. Milena Karova, PhD
Technical University of Varna, Department of Computer Science and Engineering
mkarova@ieee.bg

Assist. Prof. Anna Nikolova
Technical University of Varna, Department of Mathematics
anna.nikolova@tu-varna.bg

Drumka Zhekova
PGI "Dr Ivan Bogorov", Varna
drumka.jekova@gmail.com

DYNAMIC CAPABILITIES OF A COMPLIANCE CHECK SOFTWARE SYSTEM

Ventsislav Nikolov

Abstract: In this paper automated logical dependencies support for a limit checking software system is considered. By providing and observing of dynamic limits the system based on production systems truth maintenance is able to work effectively on solving large scale problems. The system is appropriate for regulatory checks in different fields.

Keywords: Compliance check, Limits, Production system, Automated check.

ДИНАМИЧНИ ВЪЗМОЖНОСТИ НА СОФТУЕРНА СИСТЕМА ЗА ПРОВЕРКА НА ЛИМИТНИ ОГРАНИЧЕНИЯ

Венцислав Николов

Резюме: В статията се разглежда автоматизирана поддръжка на логически зависимости в софтуерна система за лимитни ограничения. Чрез осигуряване и наблюдение на динамични лимити, системата, базирана на продукционна система с поддръжка на истиност, може да работи с висока ефективност за решаване на значителни по сложност проблеми. Системата е подходяща за проверка на регулаторни изисквания в различни области.

Ключови думи: Проверка за съответствие, Лимити, Продукционна система, Автоматизирана проверка.

1. Introduction

In every corporation, institution or market participant there are some regulatory requirements that should be regularly checked for violation. These requirements can be on different levels: international, national, corporate, team or other specific level. Some of these regulations are obligatory while others are only recommending. Their check can be done either manual or automated or partly automated. The advantage of the automated check is that it can be done faster and the system doing that could be permanently executed so for example some automated trading systems can always observe for any violations [1] [2] [7] [12] [14]. A software system started on a server can be used as automated checking system observing some preliminary defined limits definitions. Such limits could be for example: "the amount of assets in a given investment must not exceed 15% of all items in the portfolio". The solution presented here is realized as compliance check software system based on a production system which is optimized to observe any rules having left-hand side as antecedent and right-hand side as consequent [9]. The developed system is generally intended for the financial institutions but it could easily be adapted for other domains. The system here is able to check automatically some defined beforehand rules which are based not only on the regulations but also on the hedge or insurance in order to prevent any significant losses in the investment. The assets are considered in the system as elements which can be grouped in sets of elements representing portfolios. Every element contains properties according to the instrument type of the asset. An example of an element is a financial contract and its properties could be: start date, end date, name, value, etc. The properties depend on the type of the contract and some of them can be changed in contrast with others that stay permanent during the contract time period. For example the values can be changed but the start and end dates stay the same.

Some similar software solutions provide graphical controls as user interface that is not always very flexible [13]. In our solution a specific language is defined for this purpose which allows convenient rules to be defined for internal data types and limits. In [10] such a system is described with three data types and the last one, that is the most important, shows the violated and satisfied limits.

2. Solution

The statements of the input language are translated into production system code for automatic check. Such a system works in a quite different way compared to the traditional object oriented languages. The production systems comprise productions which are considered as rules which can be executed in arbitrary order according to the pattern matching. Example of two statements in the compliance check language realized here is:

```
set set1 = subset(set0, (not (PV < 100.6)));  
double_result result4 = avg(set1, 1.0 + 0.2 * volume);
```

The first statement define a new set called set1 as subset of another existing set called set0 selecting only the elements of set0 for which the property PV is not smaller than the constant value 100.6.

The same example with exchanged statements will generate two rules but if there is not conflict the activation of the second rule in some cases could be before the first one. Yet if a fact matches both rules left-hand sides then according to the conflicts resolution strategy the rules fire. That is why a check is done in the parsing stage that all identifiers definitions exist before using them.

The translation from the compliance check language into the production system code is done by ANTLR [11] which automatically generates lexical analyzer and parser by analyzing a grammar provided beforehand. The translation is conducted by performing actions expressed by code fragments into the grammar which are executed when the parser recognizes an expression.

Tree representation

The entire structure of a program in the compliance check language can be represented as a tree providing an easy way for visualization and edit of the nodes. The tree is built in the parsing stage and always is available in the memory on demand. It can be provided to the user so that if a node is selected the information it contains can be used to modify the programming code. Once the expressions are loaded however, it is impossible to modify them while the system works. Modification is permitted only when the system is stopped.

An example of the translation and tree representation can be given for the following expression:

```
set set0 = positions;  
set set1 = subset(set0, (not (CLEAN_VALUE < 100.6)));  
set set2 = subset(set0, CLEAN_VALUE = 1.5e+2) and set1;
```

It creates a new set of elements as a subset of set0, which should already exist, combined with the elements of set1 that should exist too. The production system used here is Drools [3] and the expressions above are translated into the following rules:

```
rule "inline2" // subset with given filter from set 'set0'  
when  
    NamedItem(name == "inline0", $pos : value)
```

```

    eval((((getPropertyValue($pos,"CLEAN_VALUE" )) == 150.0) ))
  then
    insertLogical(new NamedItem("inline2", $pos));
end

rule "inline3" // 'inline2' and 'set1'
  when
    NamedItem(name == "inline2", $pos0 : value)
    NamedItem(name == "inline1", $pos1 : value)
    eval((getPropertyValue($pos0, "id")).equals((getPropertyValue($pos1, "id"))))
  then
    insertLogical(new NamedItem("inline3", $pos0));
end

// custom set 'set2' is 'inline3'

```

As it can be seen every custom name is internally represented by its system name generated by the internal name generator. This is done in order to avoid any problems with the variables names provided by the user. The translation process is automated by the parser and lexer. Taking into account the common production systems features the expressions can easily be translated into another language like well-known C based CLIPS [4] [5].

One of the main advantages of the production systems is that they are based on the RETE algorithm [8] which is optimized in respect to the rules execution. It is a pattern matching algorithm based on internally built nets which nodes correspond to the patterns in the left-hand side of the rules and there are facts associated to them which determine the patterns matching. In opposite, the native rule matching algorithms are very ineffective. One of the disadvantages of the RETE algorithm is that in some cases it may cause significant memory consumption.

The tree representation of the upper expression is shown in fig. 1. A tree in the user interface is shown in fig. 2.

```

|-set2 (1:65)(1:110)[set set2=subset(set0, CLEAN_VALUE=1.5e+2) and set1;]
/
/      ...
/      +
/      |-inline2 (1:76) (1:100) [subset(set0, CLEAN_VALUE = 1.5e+2)]
/      /      ...
/      /      +
/      /      |-set0 (1:1) (1:0) []
/      /      /      ...
/      /      /      |-set1 (1:32) (1:63) [subset(set0, (not (PV < 100.6)))]
/      /      /      ...
/      /      /      +
/      /      /      |-set0 (1:1) (1:0) []
/      /      /      ...

```

Fig. 1. Tree of syntactic elements

The tree in fig. 1 is internal programming structure which contains syntax elements in its nodes similarly to an abstract syntax tree. The coordinates of the first and last characters of the

syntax elements are shown as number of line and number of character in the line. These coordinates can be seen in braces in fig. 1. The text of the element is also available as information.

Selecting a given node in the user interface tree, shown in fig. 2, provides additional information as type of the node, its definition and other specific information.

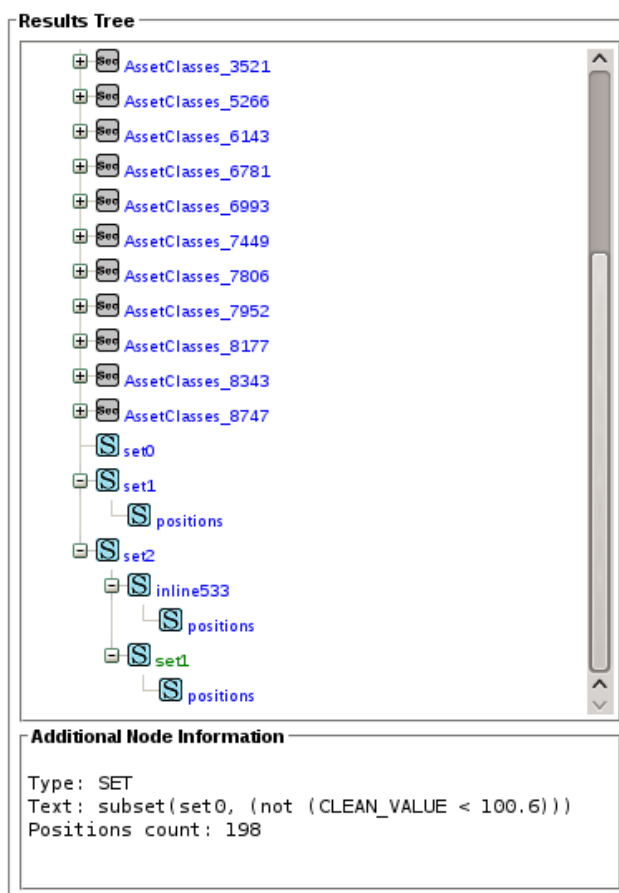


Fig. 2. Tree representation of the language elements

Software implementation

The main part of the parser relies on the objects structure realized using the “interpreter” design pattern [6]. Every element, either terminal or non-terminal, is implementation of an interface using a context of internal data and producing the resulting element of the transformed language. The context data contains data structures, internal names generators and other programming constructs needed for the translation process.

```
public interface DroolsElement {
    public String createElement(Context cxt);
}
```

Dynamic capabilities

It is important that the memory state depends on the time moment of the system working. The system usage can be separated to the following stages:

- 1) The script is loaded.
- 2) The elements of the general (default) set are inserted.

- 3) Additional elements are inserted or existing are removed or modified.
- 4) The resulting limits and other data are extracted from the memory.

The resulting limits extraction is just retrieving and analyzing the facts in the production system working memory. Once loaded the script stays in the system and every change in the facts set causes changes of the other dependent facts. The modification of an element in the production system is equivalent to its removal and insertion again with new data filled in its fields. Thus the logical dependencies are updated in real time and the system dynamically keeps the facts updated.

The dynamic capabilities are usable either in the case when there are too many values for a given property of an element or when it is not known which values can be assigned to it. For example, if the elements have a property called country and a set should be created for every country then an expression should be written for every possible country.

```
set set1 = subset(positions, country = 'Germany');
set set1 = subset(positions, country = 'Italy');
...
```

Instead of that only one expression can be written using the dynamic capabilities:

```
dynamicSet(set1, country, 'countries');
```

The named prefix ‘countries’ can be used after that to refer all sets together or selecting some of them:

```
limit myLimit = Forall('prefix1', distinct_by(country = 'all'), Sum('countries', volume * 0.1)
>= Sum('countries', PV * 0.2));"
```

Omitting the “distinct_by” construct causes all the corresponding sets to be compared. If a tree node, that is a set of elements, does not exist in a tree then the check is not performed and the limit is not violated.

A variety of additional operations are possible for the dynamic sets: union of subsets according to a given condition, counting the number of dynamic subsets, using of different mathematical functions, etc.

External functions

Taking into account that the system is realized as a software library it provides some interfaces which can be implemented outside the library in the embedding system and thus additional functions can be used to extend the existing ones. For example, in the expression below an external function called “add_year” is used which adds a year to a specified date:

```
set mySet1 = subset(positions, date = function("add_year", "10.01.2009", "1"));
```

The number of parameters of the “function” varies according to its purpose. Another example is a function returning the date for which the check is performed:

```
set mySet2 = subset(positions, date < function("eval_date"));
```

It is also possible to use a function as parameter to another function:

```
set mySet3 = subset(positions, date = function("add_year", function("eval_date"), "1"));
```

Thus the additional functions can be added without need to change the library which provides additional flexibility to the system.

Summary

The system is realized as a usable software module and it is practically used as part of other more complex systems. It has been tested for complex expressions and a huge number of elements in the memory. For example 180 lines of expressions in one of the tests were transformed into 340 production rules and 10000 elements were inserted as facts into the production system working memory. In the first inference these facts caused creation of new 110 436 facts overall that lasted 13 minutes and 19 seconds. After that an element is removed which causes removing of other 11 logically dependent facts and that operation lasts 129 milliseconds. Insertion of a new element lasts 1 ms and it does not caused automatic insertion of outer facts. Obtaining of information for the facts in the working memory however also takes some time, about 300 ms were needed. The logical inference and the limits results were completely correct. In fig. 3 the table of the limits can be seen in the user interface where the violated limits are shown with “No” in the column “satisfied”.

Limits					
Num	Limit Name	Satisfied	LHS	RHS	Remaining
1	limit_AC_1723_Bonds	No	-	-	-
2	limit_AC_1723_Bonds_down	No	21.21	25.00	15.17%
3	limit_AC_1723_Bonds_up	Yes	21.21	31.00	31.59%
4	limit_AC_1723_Equities	No	-	10.00	100.00%
5	limit_AC_1723_FXOptions	Yes	-	-	-
6	limit_AC_1723_FXOptions_down	Yes	51.19	51.00	0.36%
7	limit_AC_1723_FXOptions_up	Yes	51.19	52.00	1.57%
8	limit_AC_1723_Liquidities	Yes	-	-	-
9	limit_AC_1723_Liquidities_down	Yes	27.61	9.00	67.40%
10	limit_AC_1723_Liquidities_up	Yes	27.61	48.00	42.48%
11	limit_AC_1936_Bonds	Yes	-	-	-
12	limit_AC_1936_Liquidities	Yes	-	-	-
13	limit_AC_1986_AlternativeInvestments	Yes	-	-	-
14	limit_AC_1986_Bonds	No	-	-	-
15	limit_AC_1986_Equities	Yes	-	-	-
16	limit_AC_1986_Liquidities	Yes	-	-	-
17	limit_AC_1986_MoneyMarket	Yes	-	-	-
18	limit_AC_1986_RealEstates	Yes	-	-	-

Limit:

Fig. 3. The list of resulting limits

There are different kinds of limits. Some of them are comparison between two sides: left-hand side and right-hand side and in this case the calculated values of both sides are additionally shown. Some other limits however are only check of some condition for all elements in a given set or many sets. Such limits are shown without values for left-hand and right-hand side.

References

- [1]. Algorithmic trading. An overview. Instinet, 2005
- [2]. Baker, G., S. Tiwari. Algorithmic trading: perceptions and challenges. Edhec Risk Advisory, 2004.

- [3]. Bali, M. Drools JBoss Rules 5.0 Developer's Guide. Develop rules-based business logic using the Drools platform. Packt Publishing, 2009.
- [4]. CLIPS Reference Manual, Volume I, Basic Programming Guide, 2007.
- [5]. CLIPS Reference Manual, Volume II, Advanced Programming Guide, 2008.
- [6]. Gamma, E., et all. Design Patterns: elements of reusable object-oriented software. Addison Wesley, 1994.
- [7]. Kim, Kendall. Electronic and algorithmic trading technology. The complete guide. Elsevier, 2007.
- [8]. Ligeza, A. Logical Foundations for Rule-Based Systems. Springer Science & Business Media, 2006.
- [9]. Mayr, H. Database and Expert Systems Applications: Proceedings of the 12th International Conference, DEXA 2001 Munich, Germany, September 3 – 5, 2001.
- [10]. Nikolov, S., V. Nikolov, A. Antonov. A constraint-based approach for analysing financial market operations. Proceedings of the 14th International Conference on Computer Systems and Technologies, ACM New York, NY, USA, ISBN: 978-1-4503-2021-4, pp. 231 – 238.
- [11]. Parr, T. The Definitive ANTLR Reference: Building Domain-Specific Languages. Pragmatic Bookshelf, 2007.
- [12]. Rao, S. Algorithmic Trading: Pros and Cons. Tata Consultancy Services Limited, 2007.
- [13]. Strasberger, M. Risk Limit Systems and Capital Allocation in Financial Institutions. Banks and Bank Systems, Vol. 1, Iss. 4, 2006.
- [14]. Ward, S., Sherald, M. Successful Trading Using Artificial Intelligence.

For contacts:

Assist. Prof. Ventsislav Nikolov, PhD
 Computer Science and Engineering Department
 Technical University of Varna
 E-mail: v.nikolov@tu-varna.bg



AN OVERVIEW OF AN AUTOMATED TRADING SOFTWARE SYSTEM

Ventsislav Nikolov

Abstract: The paper illustrates the architecture and strategies used in an automated trading software system. Such a system can automatically buy, hold or sell shares at specific time moments according to the set strategy in order to obtain some profit. The architecture of the system is described together with the communication, decision making and strategy module.

Keywords: Automated trading, Decision making, Prediction, Financial information exchange.

ПРЕГЛЕД НА СИСТЕМА ЗА АВТОМАТИЧНА ТЪРГОВИЯ

Венцислав Николов

Резюме: В статията се представят архитектурата и стратегиите за действие в автоматизирана система за търговия във финансовата област. Такава система може автоматично да купува, задържа и продава акции в определени времеви моменти, в зависимост от зададената стратегия, с цел постигане на максимална печалба. Към модулната архитектура на системата са описани комуникацията, модулите за вземане на решения и стратегията за действие.

Ключови думи: Автоматична търговия, Вземане на решения, Прогнозиране, Обмен на финансова информация.

1. Introduction

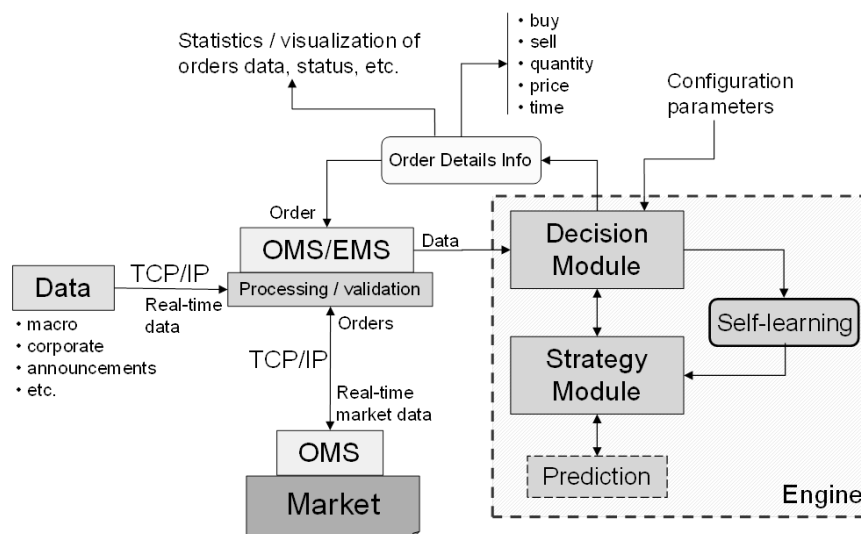
The automation of the human activities in different fields is increasing too fast today. It also tends to involve the decision making which naturally leads to using of intelligence computation software systems. Here an automation of a trading system is described as a software implementation which together with some modules can realize appropriate trading strategies and prediction of the market indicators. The system automatically generates orders to buy, sell or hold shares with given price and quantity at specific time moments according to the parameters and constraints of the chosen algorithm [2]. In order to do that information from the market and other sources should be obtained real-time. In this way no human intervention is needed for the system which relies on a software program which works infinitely. In addition it should allow monitoring of the statistics, current status and other additional information. Generally the technical design of such a system is not standardized [1] and the system should also be open for new options.

The technical details of the algorithmic trading system are shown in fig.1.

The communication module connects the trading system with the rest of the world. It receives information and sends orders about the trades. If the system uses some intelligent approaches of analysing additional information like corporate data, companies' announcements, etc. it should be able to receive that data from certain reliable and authentic data sources. There are some important requirements about the communication which are as follows.

- High speed connection. The connection of the system should be very fast because the difference between successful and unsuccessful trade may be a matter of milliseconds.
- Low latency. The data should be obtained straight from the source providing very low latency.

- **Capacity.** The system should allow transfer of a huge amount of incoming and outgoing messages. If the capacity of the communication does not withstand to the loading then it could lead to system algorithms frustration.
- **Reliability and recoverability.** This requirement concerns not only the communication infrastructure but also the ability of the software to recover after a communication crush.



OMS/EMS – Order/Execution Management System

Fig. 1. Description (title) of the figure

The most important components of the trading system are described below.

2. Communication

One commonly used approach for data exchange is by using of the Financial Information eXchange (FIX) protocol. It is an open and free XML protocol for real-time communication using trade-related messages allowing sending and receiving orders as well as messages processing and validation. The data exchange is managed by an engine called FIX Engine which controls the session and application layers of the TCP/IP protocol stack. It also manages the network connections, creates and parses the information messages and recovers the system when an error occurs.

3. Decision making module

At the core of the trading system engine works a decision making module. It is responsible for choosing and processing of assets, order type, price, quantity and time of order execution according to the configuration parameters. The orders are technically executed by the Order Management System (OMS) which receives all needed market and corporate data and sends the trading orders.

The module should be designed in a way allowing continuously improvement according to any additional client-specific requirements. Moreover it should be adaptable, self-learning and flexible withstanding to the huge range of possible market scenarios. Thus some soft computing techniques are appropriate to be applied to the decision making module.

4. Strategy module

The strategy module performs the operations required by the strategies chosen by the Decision Making Module. There are many known strategies that could be used. Some of the most commonly used are shown below.

- Volume Weighted Average Price (VWAP). Parameters of this strategy are: start time, end time, max participation rate, level of aggression. In a given moment if the price is lower than intra-day VWAP average the corresponding share should be bought, and when the current price is higher than the average price it should be sold – fig.2. Thus in short term there will not be losses because the profit range is kept by buying under the average and selling above the average.

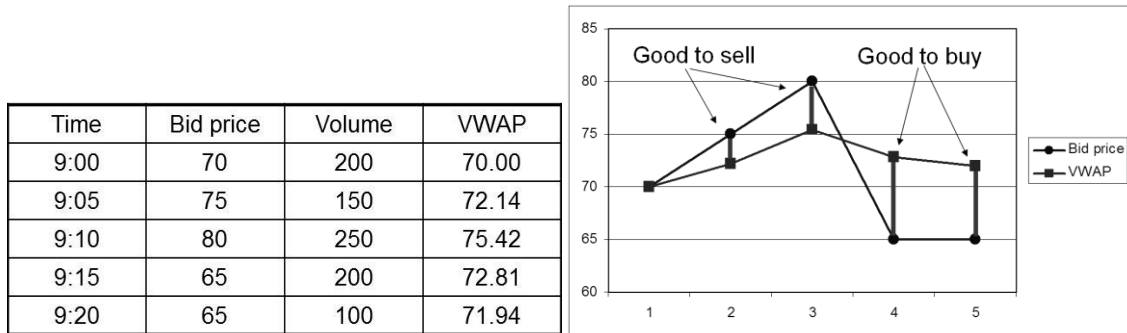


Fig.2. The VWAP approach used by the strategy module

- The VWAP is calculated in this strategy as shown in (1).

$$P_{\text{vwap}} = \frac{\sum_j P_j Q_j}{\sum_j Q_j} \quad (1)$$

where P_{vwap} is volume weighted average price

P_j – price of trade j

Q_j – quantity of trade j

j – individual trade that

VWAP is often used as a trading benchmark by investors that aim to be as passive as possible in their execution.

- If current price is below the VWAP benchmark up to the end of the chosen time horizon, the current bid price is determined as good to buying in, but bad for selling out;
- If current price is above the VWAP benchmark up to the end of the chosen time horizon, the current bid price is determined as good to selling out but bad for buying in.

Thus there will not be losses in short term as a profit range is kept by buying under the benchmark and selling above the benchmark (e.g. based on intra-day VWAP).

- Mean reversion could be used similarly to the VWAP: the average price is computed for a specified period and when the current price is less than the average, the stock is considered as attractive for purchasing with the expectation that the price will rise, and vice versa, if the current price is expected to fall and it is good for selling if it is above the average price.
- Time weighted average price (TWAP). Parameters of this strategy are: start time, end time, max participation rate, level of aggression. This strategy is used to execute orders over a specific time so to keep the price close to that one which reflects the true market price, i.e. the objective is to minimize the market impact.
- Implementation shortfall. Parameters here are: start time, end time, reference price, participation rate aggressiveness, participation rate tolerance, limit price. This strategy is based on the terms:
 - Paper portfolio: represents the ideal case where the transaction costs, commission, etc. do not happen.
 - Actual portfolio: reflects the real markets situation where commissions, etc. are taken into account.

- Implementation shortfall = Paper portfolio – Actual portfolio
that is also: theoretical portfolio – implemented portfolio
- Volume participation. Parameters of this strategy are: start time, end time, participation rate and maximum tolerance from participation rate. The objective is to participate in volume as it arises. For example if the participation rate is 10% and other market participants transacted 10000 shares then 1000 shares have to be executed.

Some other commonly used strategies are: Guerilla, Liquidity, Trend Following, Arbitrage, Mean Reversion, etc. These strategies have some specific parameters: start time, end time, max participation rate, level of aggression, participation rate tolerance and others.

The strategy module may refer to some specific computational modules [3, 4]. For example, as it is shown on fig.1, the prediction module is called from the strategy module. These specific mathematical computations are needed to realize the specific strategy.

The prediction could be performed for:

- Stock prices. The prediction is able to spot trend and patterns more quickly than any human could
- Volumes – in case VWAP is used as a benchmark. The daily volume is not known at the beginning of the day but could be predicted. Some techniques also include principal component analysis on the volumes beforehand.

The prediction generally includes data pre-processing, solving of matrix equations (in batch mode or iteratively) and data post-processing. Also because of the self-learning approach some new strategies may also be added and some of them need to use uni-variate or multi-variate prediction from the prediction sub-module in the engine.

When prediction is performed some indicators for the prediction quality should be calculated for preliminary separated test values. Usually additional confidence bounds are used as shown in grey colour around the predicted values in fig.3 where an example of historical values of a time series and its prediction for a time horizon of ten future values are shown. Usually the confidence levels are 95% or 97.5%.

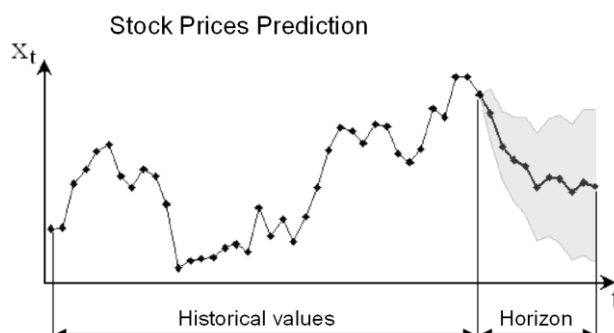


Fig.3. Example of time series and prediction for a given time horizon

Often only correct moving direction of the prediction is enough because if only 60-65% of the trades are successfully performed based on the predictions then it could be enough for profit to happen.

In Fig. 4 one of the most commonly used approach for time series prediction is shown using the sliding window approach.

There are different mathematical models which work using this approach. The Box-Jenkins approach, autoregressive methods as well as neural networks are amongst them. The work is separated in two stages: model identification and prediction. The first one is based on historical values analysis and the second uses the built model from the first stage in order to generate the predictions. The model identification is in fact an optimization problem and the prediction is filtering the input through the model.

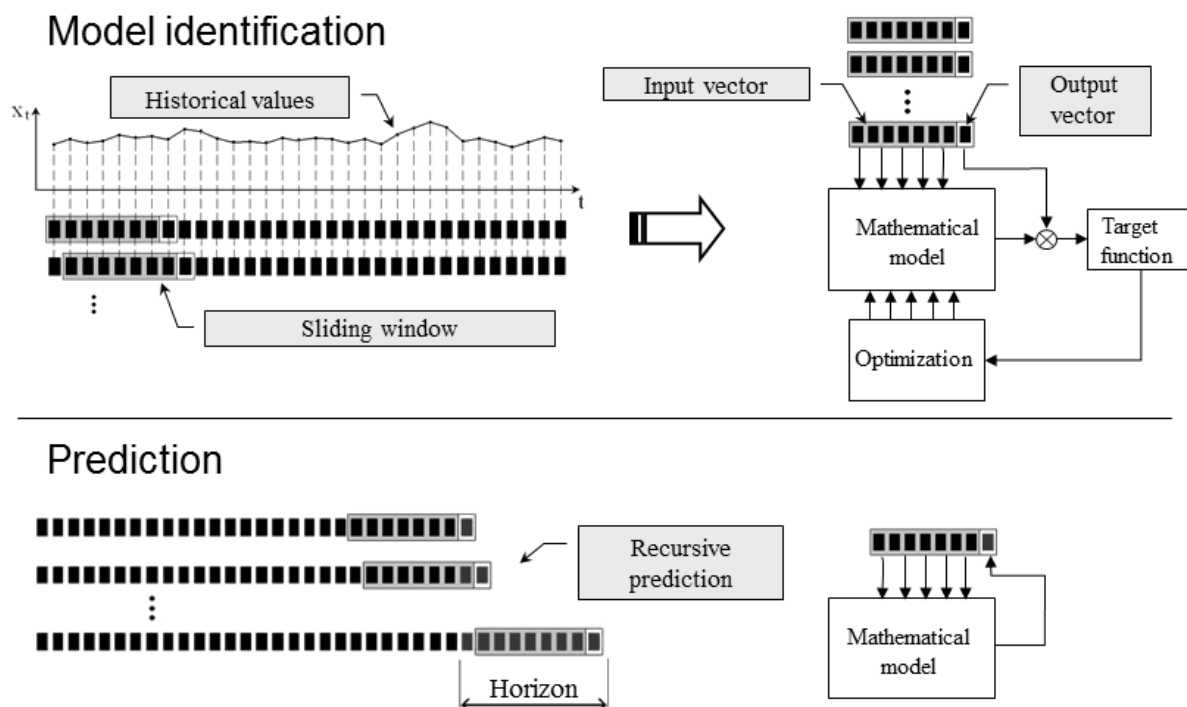


Fig.4. Fitting a mathematical model to historical values and prediction

Another approach for the prediction is performing of fundamental analysis in addition to the technical analysis.

5. Conclusions and future work

The implementation of the automated trading system is in progress and from scientific point of view the results of the trading simulations should be analyzed. They should be done without human intervention for long enough time periods. The results will generally depend on the strategies set in the system as well as on the properties and behaviour of the traded shares. The market situation may result in series of data with different variations and thus the different strategies may be appropriate or inappropriate in a given time moments.

References

- [1]. Kim Kendall, Electronic and algorithmic trading technology. The complete guide. Elsevier, 2007.
- [2]. Algorithmic trading. An overview. Instinet, 2005
- [3]. Baker G., S. Tiwari, Algorithmic trading: perceptions and challenges. Edhec Risk Advisory, 2004.
- [4]. Ward S., Sherald M., Successful Trading Using Artificial Intelligence. Ward Systems Group, 2006.
- [5]. Rao S., Algorithmic Trading: Pros and Cons. Tata Consultancy Services Limited, 2007.

For contacts:

Assist. Prof. Ventsislav Nikolov, PhD
Computer Science and Engineering Department
Technical University of Varna
E-mail: v.nikolov@tu-varna.bg

АВТОМАТИЗИРАНИ СИСТЕМИ ЗА УПРАВЛЕНИЕ



AUTOMATED CONTROL SYSTEMS

ПРИЛОЖЕНИЕ НА РЕКУРСИВНИ МЕТОДИ ЗА ОЦЕНЯВАНЕ НА ПАРАМЕТРИ В СИСТЕМА ЗА АДАПТИВНО УПРАВЛЕНИЕ НА ПОСТОЯННО ТОКОВ ДВИГАТЕЛ ПОСРЕДСТВОМ САМОНАСТРОЙВАЩ СЕ РЕГУЛАТОР С МИНИМАЛНА ДИСПЕРСИЯ

Иван В. Григоров

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 приложение 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.

APPLICATION OF RECURSIVE METHODS FOR PARAMETER ESTIMATION IN ADAPTIVE CONTROL OF DC MOTOR WITH MINIMAL VARIANCE

Ivan V. Grigorov

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

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

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

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

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

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

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

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

От (2) и (3) следва

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

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

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

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)

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

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

Рекурсивния метод на най-малките квадрати може да се разглежда като частен случай на РПМНК 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')

$$v_i^T = [-\hat{y}_{i-1} \quad -\hat{y}_{i-2} \dots -\hat{y}_{i-n} \quad u_{i-1} \quad u_{i-2} \dots u_{i-m}] \quad (8)$$

$$v_i^T = [u_{i-1} \quad u_{i-2} \dots u_{i-m}] \quad (8')$$

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

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

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

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

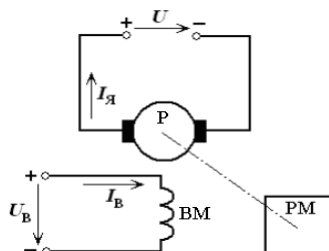
- Изчисленията започват от т.2 с новите оценки. Ако не е необходимо претегляне на наблюденията, в (9) и (10) се полага $\rho = 1$. [1]

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.

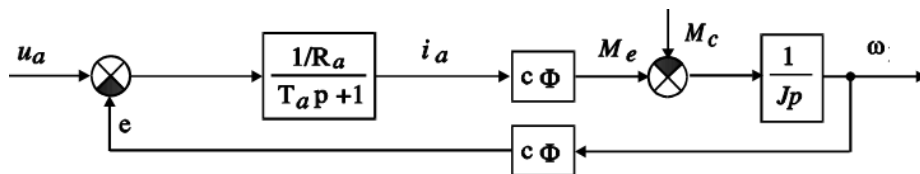


Фиг. 1. Принципно устройство на ПТД

Динамичните свойства на постоянно токовия и двигател са описани със 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).

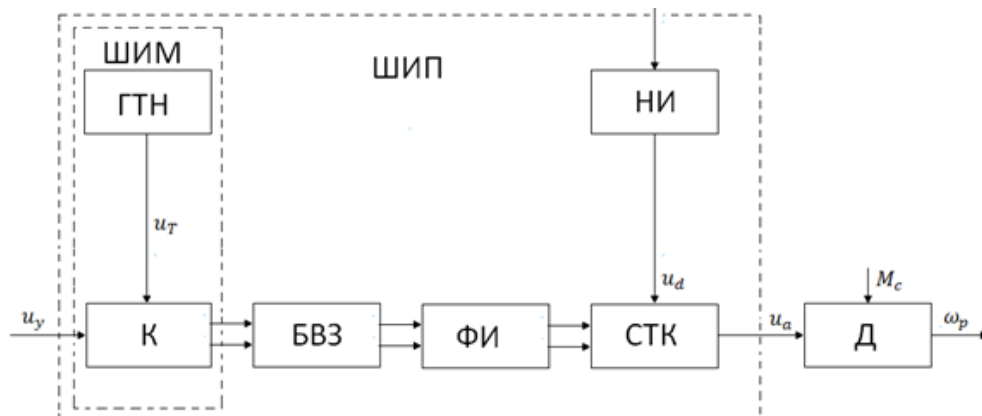


Фиг. 2. Структурна схема на ПТД

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.

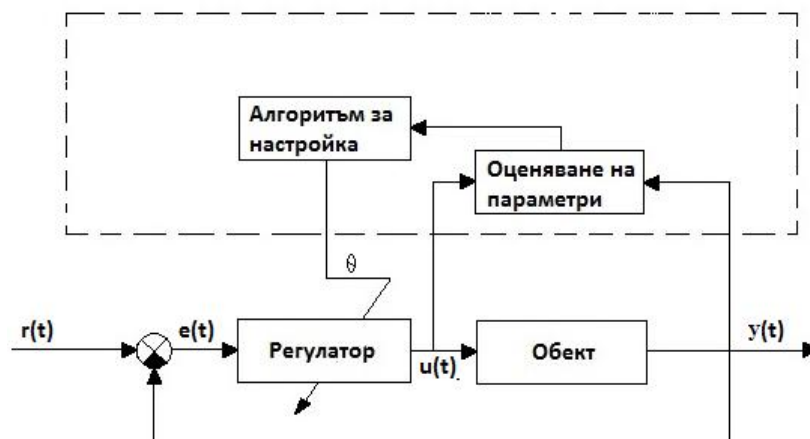
На блоковата схема на фигура 3 е представена принципна блок схема на ШИП-ПТД, където: ГНТ – генератор на трионообразно напрежение, К – компаратор, БВЗ – блок за времезадръжка, ФИ – формирова̀тел на импулсите, НИ – неуправляем изправител, СТК – силов транзисторен комутатор, Д – постоянно токов двигател.



Фиг. 3. Принцидна 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 4. Самонастройващият се регулатор е един от методите за управление при който е постигнат значителен напредък през последните години. Има три основни елемента (фигура 4). Първият представя класическа система със обратна връзка. Вторият осъществява идентификация на процеса. Третият реализира алгоритъма за настройка на параметрите на



Фиг. 4. Блок схема на самонастройващ се регулатор

регулатора на базата на оценените параметри на процеса. 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 $E(z^{-1})$ и $F(z^{-1})$ чрез решаването на Диофантовото уравнение.

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

- 3. Определят се коефициентите на полинома $G(z^{-1}) = E(z^{-1})B(z^{-1})$.

- 4. По формула (13)

$$u(k) = -\frac{1}{g_0} [f_0 y(k) + f_1 y(k-1) + \dots + f_{n-1} y(k-n+1) + g_1 u(k-1) + g_2 u(k-2) + \dots + g_{m+d-1} u(k-m-d+1)] \quad (13)$$

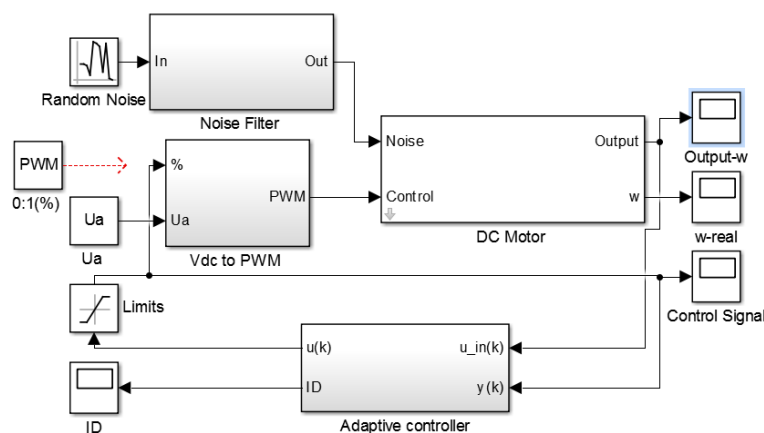
се определя управлението $u(k)$, след което k се увеличава и алгоритъма се повтаря от т.1.

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

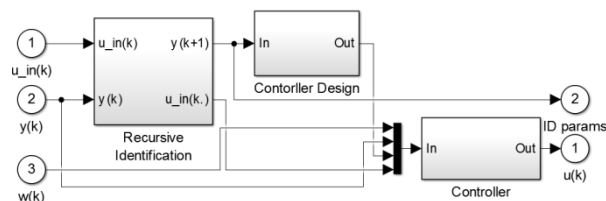
- На базата на информация за входа и изхода на обекта към k -тия момент, чрез RLS, се определят оценките на $f_i(k)$ и $g_i(k)$ на полиномите $F(z^{-1})$ и $G(z^{-1})$.
- Получените оценки се заместват в (13) и се определя управлението $u(k)$
- След получаване на нова информация k се увеличава и алгоритъма се повтаря от т.1.

5. Експериментални изследвания и резултати

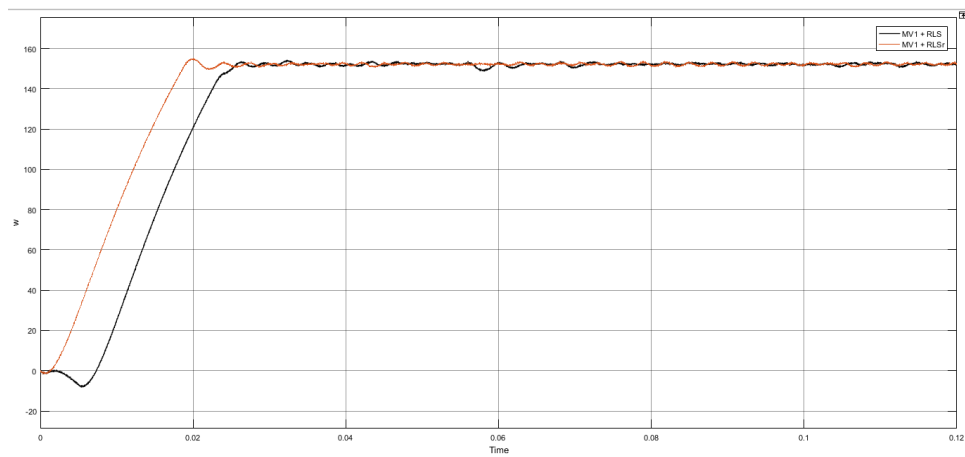
Изследванията тук се отнасят до работоспособността, възможностите и надеждността на описаните вече рекурсивни методи за оценяване на параметри в адаптивно управление на ПТД в реално време. За целта на изследването е генериран случаен входен сигнал, симулиращ зашумяване, с помощта на програмната среда MATLAB. Изследванията са направени с използването на System Identification Toolbox в Matlab\Simulink. За тази цел са направени блокове за рекурсивно оценяване на параметри в реално време както следва: явен самонастройващ се регулатор с минимална дисперсия (MV1), неявен самонастройващ се регулатор с минимална дисперсия (MV2); 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).



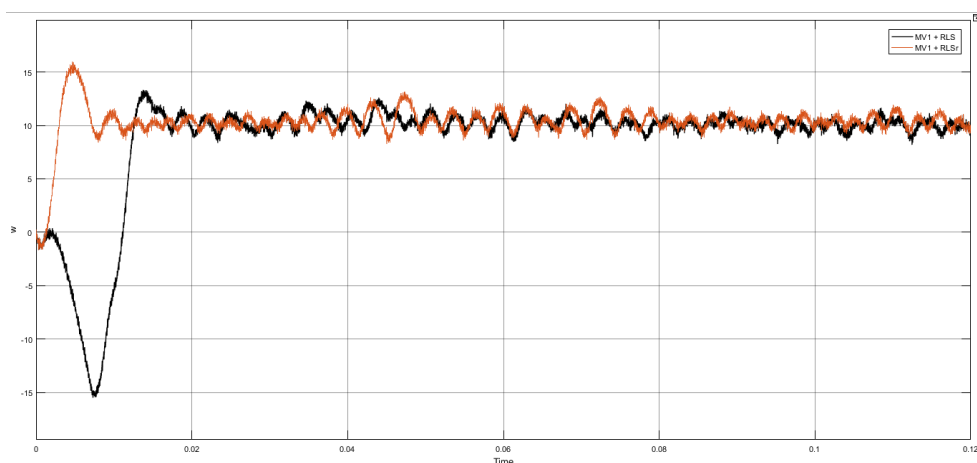
Фиг. 5. Блок схема в Simulink на система за управление на ПТД със самонастройващ се регулатор с минимална дисперсия



Фиг. 6. Блок схема в Simulink на система със самонастройващ се регулатор с минимална дисперсия представяща подсистемата “Adaptive controller”

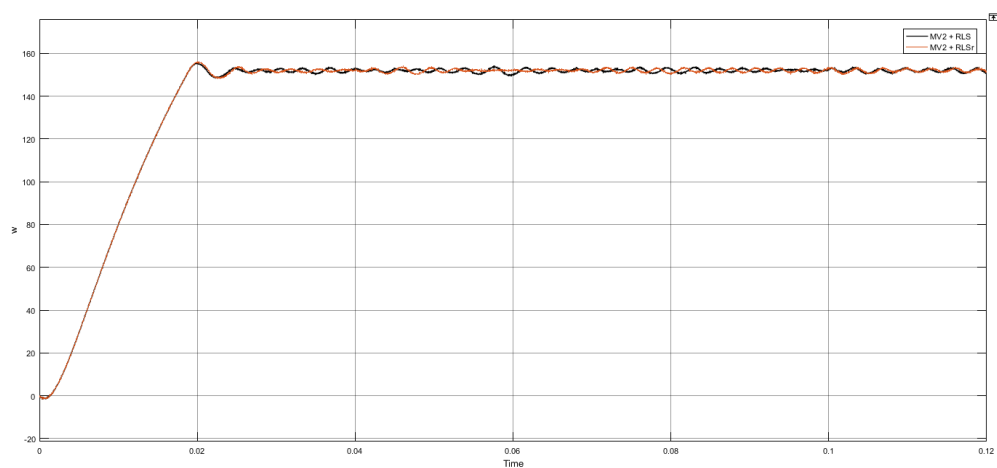


Фиг. 7. Скорост на ПТД с MV1 +RLS(b) и MV1 +RLSr(r) за $\omega = 152,35rad / s$

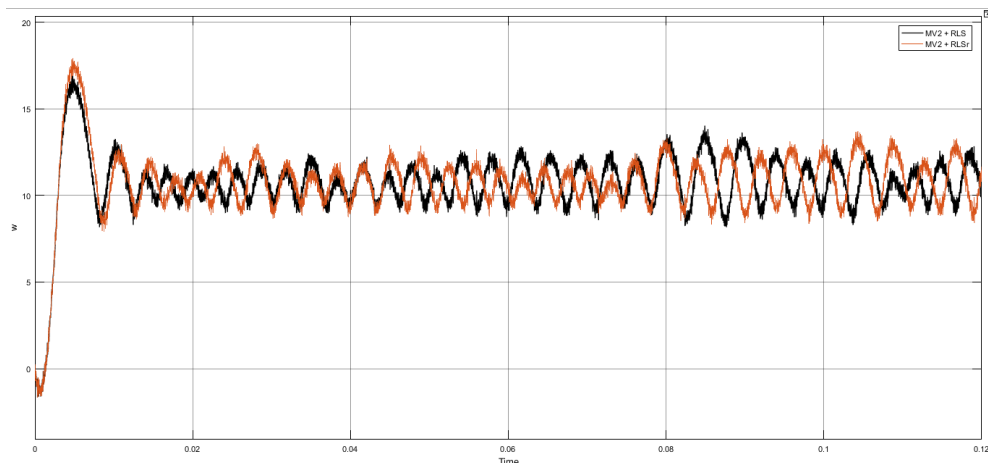


Фиг. 8. Скорост на ПТД с MV1 +RLS(b) и MV1 +RLSr(r) за $\omega = 10,1567rad / s$

На фигура 7 и фигура 8 са представени скоростта на ПТД при система за адаптивно управление на ПТД посредством явен самонастройващ се регулатор с минимална дисперсия, като са използвани по-горе описаните методи RLS и RLSr при зададена скорост $\omega = 152,35rad / s$, $\omega = 10,1567rad / s$.



Фиг. 9. Скорост на ПТД с MV2 +RLS(b) и MV2 +RLSr(r) за $\omega = 152,35rad / s$



Фиг. 10. Скорост на ПТД с MV2 +RLS(b) и MV2 +RLSr(r) за $\omega = 10,1567 \text{ rad} / \text{s}$

На фигура 9 и фигура 10 са представени скоростта на ПТД при система за адаптивно управление на ПТД посредством неявен самонастройващ се регулатор със минимална дисперсия като са използвани по-горе описаните методи RLS и RLSr при зададена скорост $\omega = 152,35 \text{ rad} / \text{s}$, $\omega = 10,1567 \text{ rad} / \text{s}$.

6. Анализ и изводи

Резултатите от симулациите потвърждават възможността за приложение на описаните рекурсивни методи за оценка на параметрите в адаптивно управление на ПТД. Както може да се види от резултатите при ниски скорости се съдържат най-големи разлики от зададената стойност. Както вече е известно, методът на рекурсивната инструментална променлива изисква втори шумов сигнал за изхода на процеса за по-добри оценки, но това влияе върху качеството на управлението. 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. Фактът, че във неявния алгоритъм е спестено решаването на Диофантовото уравнение, което води до по-голямо бързодействие от страна на изчислителната техника, прави алгоритъма по-приложим практически.

БЛАГОДАРНОСТИ

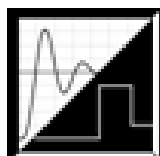
Научните изследвания, резултатите от които са представени в настоящата публикация, са извършени по проект ПД-1 в рамките на присъщата на ТУ-Варна научноизследователска дейност, финансирана целево от държавния бюджет за 2017 година.

Литература

- [1]. Agustin O., Oscar L, Francisco Q., Identification of DC motor with parametric methods and artificial 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., Kariiriii 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, and Asmaa F. Rashwan, Different identification methods with application to a DC motor

За контакти:
докторант Иван В. Григоров
катедра „Автоматизация на производството”
Технически университет - Варна
E-mail: mad_doc@abv.bg



DEVELOPMENT OF MATLAB FUNCTION FOR SELECTION OF SUITABLE WAVELET FOR SIGNAL FILTRATION

Reneta D. Parvanova, Mariyana G. Todorova, Stefan L. Stefanov

Abstract. In this paper a Matlab function for selection of suitable wavelet functions for filtration of signals is developed. The choice of the wavelet is performed based on a quantitative assessment of the effectiveness of filtration. Different signals and a big number families of wavelet functions to test the m-function are used.

Keywords: filtration, Matlab function, noisy signals, wavelet functions.

РАЗРАБОТВАНЕ НА МАТЛАБ ФУНКЦИЯ ЗА СЕЛЕКЦИЯ НА ПОДХОДЯЩ УЕЙВЛЕТ ЗА ФИЛТРАЦИЯ НА СИГНАЛИ

Ренета Д. Първанова, Марияна Г. Тодорова, Стефан Л. Стефанов

Резюме: В статията е разработена функция в Матлаб за избор на подходящи уейвлет функции за филтриране на сигнали. Изборът на уейвлет се извършва въз основа на количествена оценка на ефективността на филтрацията. За тестване на m-функцията са използвани различни сигнали и голям брой семейства уейвлет функции.

Ключови думи: филтрация, Матлаб функции, зашумени сигнали, уейвлет функции

1. Introduction

Wavelets are powerful tool for analyzing digital signals. They are simply mathematical functions exhibiting oscillatory behavior. A wavelet function, known as a "mother wavelet", gives rise to a family of wavelets that are translated and dilated versions of the original mother wavelet. Unlike sinusoidal functions, where the oscillation dominates the entire signal, wavelets show only localized oscillation [1], [4].

Wavelet analysis provides a flexible technique for signal processing in comparison to conventional Fourier analysis. Wavelets have advantages over traditional methods in analyzing physical situations where the signal contains discontinuities and sharp spikes [3].

Wavelets have an important application for removing noise from a variety of signals (denoising). Signals do not exist without noise, which may be negligible under certain conditions. However, there are many cases in which the noise corrupts the signals in a significant manner, and it must be removed from the data in order to proceed with further data analysis. There is a wide range of applications in which denoising is important - data mining, medical image analysis, radio astronomy, etc. Each application has its special requirements [2], [6], [7].

In this paper Matlab function for selection of suitable wavelet function for signal filtration is developed. Seven families of wavelets that have proven to be especially useful are included. There is a choice between Haar, Daubechies, Symlets, Coiflets, Biorthogonal, Reverse Biorthogonal and Discrete Meyer wavelets. The effectiveness of filtration is evaluated and the results are compared. The criteria used are the relative error E and the coefficient of noise reduction CNR.

$$E = \sqrt{\frac{\sum_{i=1}^n (ye_i - ym_i)^2}{\sum_{i=1}^n (ye_i)^2}} \quad (1)$$

$$CNR = \sqrt{\frac{\sum_{i=1}^N [y_n(t_i) - y(t_i)]^2}{\sum_{i=1}^N [Y(t_i) - y(t_i)]^2}} \quad (2)$$

where:

$y(t)$ - original signal;

$y_n(t)$ - deteriorated signal;

$Y(t)$ - filtered signal;

2. Development of m-function for selection of suitable wavelet for signal filtration

A m-function *Filtsig* for the purpose of filtering of deteriorated signals and selection of suitable wavelet is developed. The description of the function is given below.

Input argument: original (non-filtered) signal

Output arguments: filtered signal, relative error (E), coefficient of noise reduction (CNR).

Base steps of the algorithm:

- Step 1. Loading the signal which will be filtered.
- Step 2. Applying user defined level of noise to the signal.
- Step 3. Choosing a wavelet which will be used for filtering.
- Step 4. Choosing the wavelet level (if necessary).
- Step 5. Turning the wavelet name into a word.
- Step 6. Filtering the signal using the chosen wavelet.
- Step 7. Plotting the noised and filtered signals.
- Step 8. Calculating the relative error and noise reduction coefficient.

In Step 1 the desired signal can be entered by the user or an original signal can be generated.

In Step 2 if the user wants, the signal can be noise with white Gauss noise at a chosen noise/signal ratio. This is done by the user by entering the ratio in percentages. That choice is used in a formula which allows for achieving the desired result by using normally distributed pseudorandom numbers. For a visual comparison of the noised and original signals the two are plotted together.

In Step 3 a wavelet for filtering the noised signal is chosen. There is a choice between 7 wavelet families.

If necessary, in Step 4 a wavelet level is chosen.

In Step 5 the word which will combine the name of the wavelet and its level (if any) is shaped. The word is saved as a string in the variable *wavname*. If the wavelet has a level, the user enters it after which the number is converted to a string using the function *int2str* and then gets added to its place in the word.

After the variable *wavname* is shaped filtering the signal in Step 6 with the function *wden* starts. It has 6 input arguments – the signal which will be filtered, the threshold selection rule, soft or hard thresholding selection, the threshold rescaling, the level at which wavelet decomposition is performed and the name of the desired orthogonal wavelet.

In Step 7, after setting up parameters, the filtered and noised signals are plotted together for a graphic presentation of the filter quality.

After this in Step 8 the relative error of filtering E and the noise reduction coefficient CNR are calculated. For grading the results the following rule is in order – the error should be as low as possible while the noise reduction coefficient should be as high as possible.

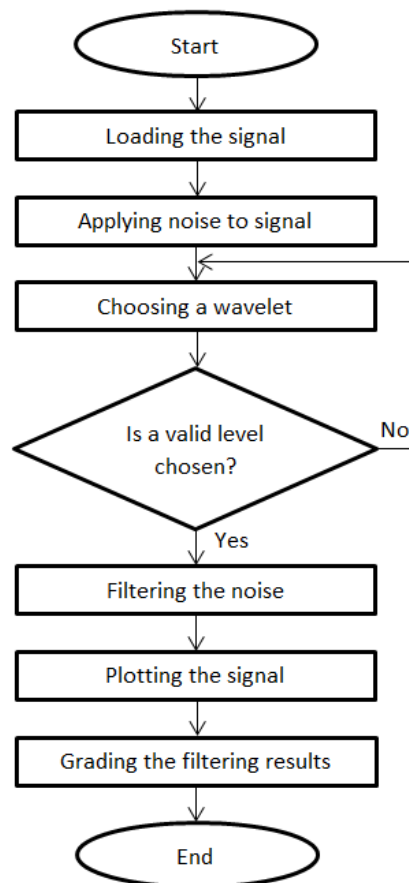


Fig. 12. Algorithm flowchart

3. Experimental research

Filtering of an oscillatory signal, defined by the coefficients of the numerator and denominator of the Laplace image, is done for testing the reliability of the developed m-function *Filtsig*. After that Gaussian white noise is applied to the signal at a 5% noise/signal ratio. Filtering is done with all of the seven types of wavelet families. Some of the results are shown below.

After conducting experiments with wavelets at a level of 4 the following results are achieved.

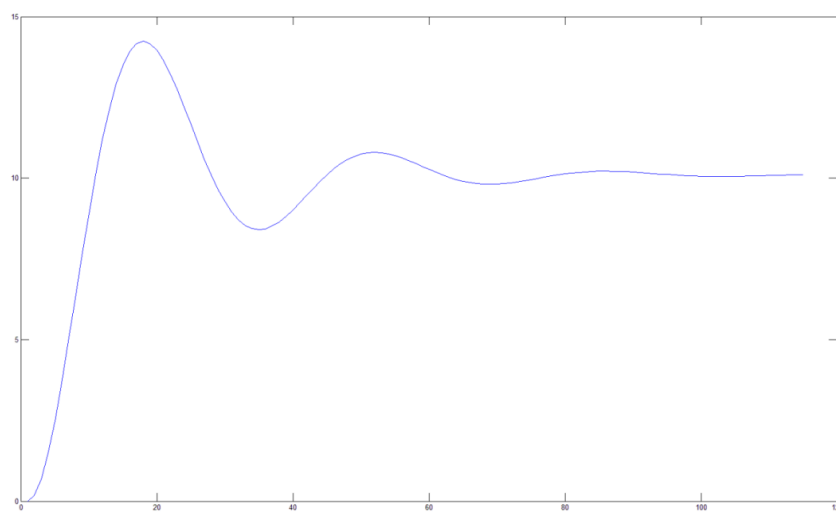


Fig. 13. Original signal

The best filtering is obtained by using Discrete Meyer wavelet. The smallest relative error $E=1.074\%$ and biggest coefficient of noise reduction $CNR=5.1536$ in this case are obtained. The deteriorated and filtered signals are shown on Fig. 3.

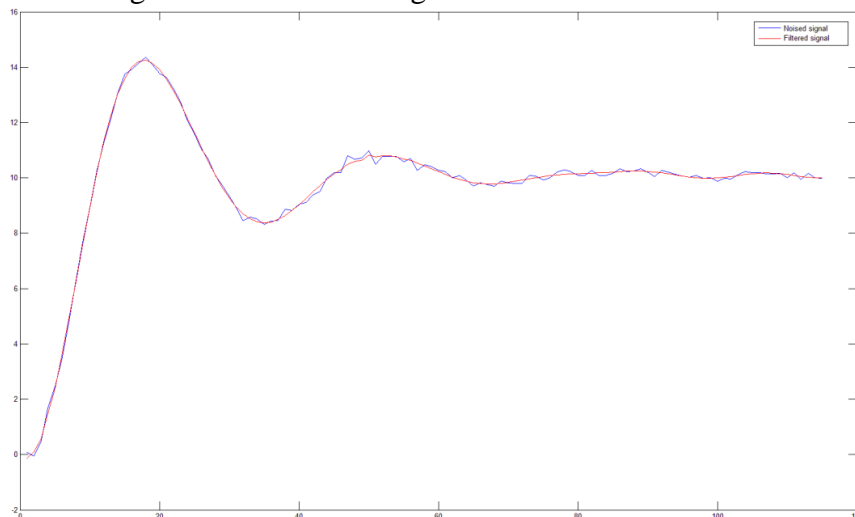


Fig. 14. Deteriorated and filtered signals (by Discrete Meyer wavelet)

The worst filtering is obtained by using Haar wavelet. The biggest relative error $E=2.0052\%$ and smallest coefficient of noise reduction $CNR=0.3636$ are obtained. Fig. 4 shows the deteriorated and filtered (by Haar wavelet) signals.

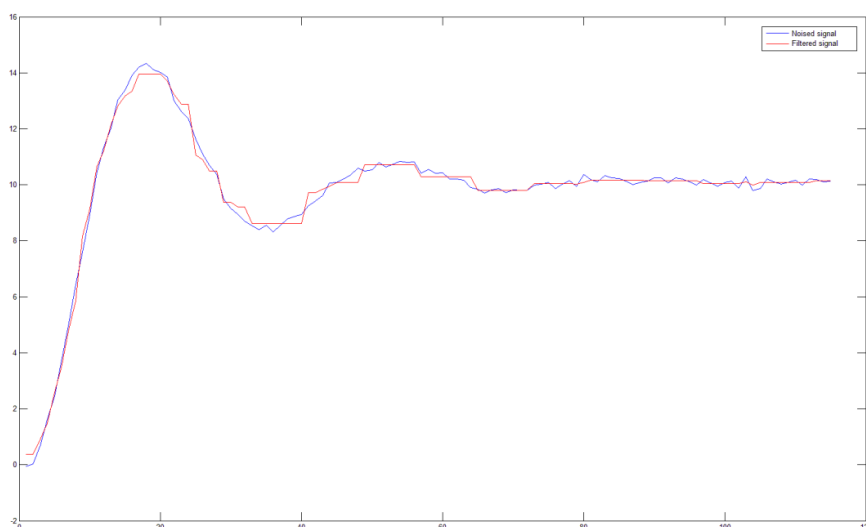


Fig. 15. Deteriorated and filtered signals (by Haar wavelet)

Conclusions

M-function for selection of suitable wavelet on the basis of quantitative assessment of the effectiveness of filtration is presented in this paper. The developed function is tested by filtering of an oscillatory signal, deteriorated by Gaussian white noise with 5% noise/signal ratio. Seven types of wavelet families are examined. The obtained results of the relative error and coefficient of noise reduction are compared and the suitable wavelet is selected.

REFERENCES

- [1] Chun-Lin, Liu: A Tutorial of the Wavelet Transform. (2010).

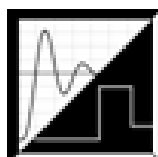
- [2] Cohen R.: Signal Denoising Using Wavelets. Project Report, Israel Institute of Technology (2012).
- [3] Graps A.: An introduction to wavelets. IEEE Computational Science & Engineering, vol. 2, No 2, pp. 50 – 61 (1995).
- [4] Leavey C. M., James M N, Summerscales J., Sutton R.: An introduction to wavelet transforms: a tutorial approach. Insight, vol 45, No 5, pp. 344 - 353 (2003).
- [5] Romeo A., Horellou C., Bergh J.: A wavelet add-on code for new-generation N-body simulations and data denoising (JOFILUREN). Mon Not R Astron Soc, vol. 354, issue 4, pp.1208-1222 (2004).
- [6] Todorova M., Parvanova R.: Filtration of noisy signals by wavelet functions. Computer science and technologies, vol. 2, ISSN 1312-3335, pp. 32-38, Varna (2016).
- [7] Todorova M., Ivanova R.: Research and signal analysis with different classes of wavelets. Signal processing, ICEST, pp. 157-161, Makedonia (2016).
- [8] Wavelet Toolbox, User's Guide, The Math Works, Inc. (2015).

For contacts:

Assoc. Prof. Mariana Georgieva Todorova, PhD
 Department of Automation
 Technical University of Varna
 E-mail: mgtodorova@tu-varna.bg

PhD Student Reneta Dancheva Parvanova
 Department of Automation
 Technical University of Varna
 E-mail: renetaivanova1@gmail.com

Stefan Lyubomirov Stefanov
 Student from Department of Automation
 Technical University of Varna
 slstefanov@abv.bg



РАЗРАБОТВАНЕ НА GUI В MATLAB ЗА ИЗСЛЕДВАНЕ НА ДИНАМИЧНИТЕ И ЧЕСТОТНИ ХАРАКТЕРИСТИКИ НА ИДЕАЛНИ ЛИНЕЙНИ РЕГУЛАТОРИ

Мариела И. Александрова

Резюме: Целта на настоящата работа е разработване на графичен потребителски интерфейс GUI (Graphical User Interface) в среда MATLAB за получаване и изследване на динамичните и честотните характеристики на идеални линейни регулатори. Графичният интерфейс ще позволява избор на закон за регулиране, като след въвеждане на стойности на настройките параметри ще се изчертават преходната функция и/или логаритмичните амплитудно-честотна (ЛАЧХ) и фазово-честотна характеристики (ЛФЧХ) на избрания регулатор.

Ключови думи: графичен потребителски интерфейс (GUI), преходна функция, ЛАЧХ, ЛФЧХ, закон за регулиране, идеални линейни регулатори

Development of MATLAB GUI for Investigation of Step and Frequency Responses of Ideal Linear Controllers

Mariela I. Alexandrova

Abstract: The aim of this paper is development of GUI (Graphical User Interface) in MATLAB for investigation of the step responses and logarithmic frequency responses (magnitude and phase) of ideal linear controllers. The developed GUI allows for choice of control strategy as well as setting of control strategy parameters in order the step response and the logarithmic frequency responses of control strategy chosen to be obtained.

Keywords: Graphical User Interface (GUI), step response, logarithmic frequency response, control strategy, ideal linear controller

1. Въведение

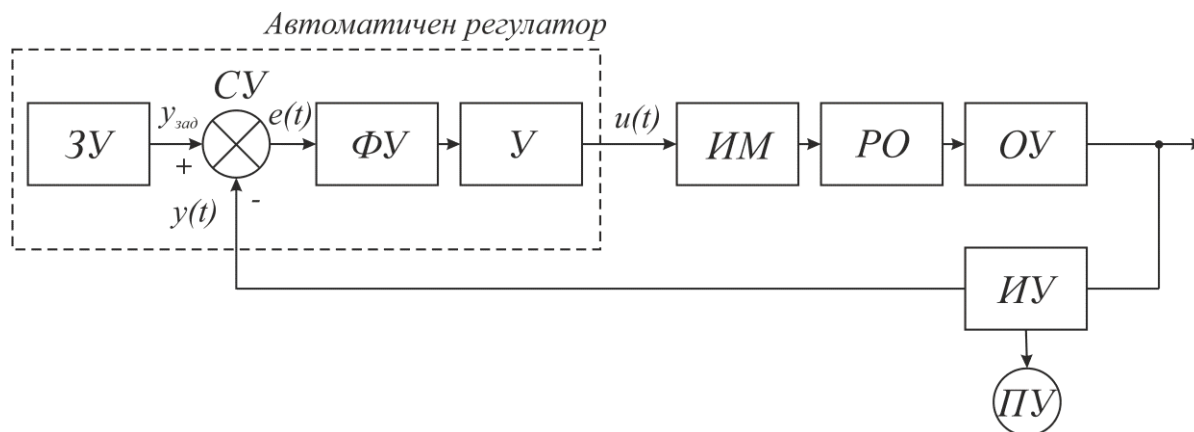
Автоматичният регулатор е основен градивен елемент на всяка автоматична система (виж фигура 1). Неговото функционално предназначение е да изработи управляващо въздействие, чрез което се регулира дадена технологична величина (температура, налягане, ниво и др.).

В зависимост от функцията, с която се описват законите за регулиране, регулаторите се разделят на линейни и нелинейни. Регулаторите, при които алгоритъмът на преобразуване на грешката за изработване на управляващо въздействие е съставен само от линейни операции (сумиране, умножение с постоянен коефициент, интегриране и диференциране) са линейни регулатори. Те са получили най-широко приложение в практиката, затова се приемат като основни регулатори и се наричат още стандартни. В общия случай, при изработването на управляващото въздействие от тези регулатори, грешката се обработва чрез умножение с коефициент, интегриране и диференциране. Затова в управляващото въздействие могат да съществуват три компоненти – пропорционална, интегрална и диференциална. В частните случаи някой от начините на обработка може да отсъства, а от там и съответната компонента. Линейните регулатори носят названия в съответствие с обработката, която извършват – пропорционален (П), интегрален (И), диференциален (Д), пропорционално-интегрален (ПИ), пропорционално – диференциален (ПД), пропорционално-интегрално-диференциален (ПИД).

В случаите, когато операциите сумиране, интегриране, диференциране и умножение с постоянен коефициент се извършват точно, регулаторите се приемат за идеални [1,2,3,4].

2. Идеални линейни регулатори

Обикновено регулаторите се изграждат според структурата, показана на фигура 1. При постоянно задаващо въздействие ($y_{зад}(t) = const$), уравнението, което отразява връзката между входната променлива на регулатора (грешката на регулиране $e(t)$) и изходната му променлива (управляващото въздействие $u(t)$, което се подава към входа на обекта), е строго специфично за различните закони за регулиране.



Фиг. 1. Функционална схема на система за автоматично управление

Уравнението $u(t) = f[e(t)] = f[y_{зад}(t) - y(t)]$, с което се описва функционалната зависимост на управляващото въздействие $u(t)$ от грешката $e(t)$ при едномерни системи и единична отрицателна обратна връзка (ООВ), се нарича закон за регулиране [1,4].

Познаването на динамичните и честотни характеристики на типовите линейни закони за регулиране е от съществено значение при техния избор с цел синтез на системите за автоматично регулиране на основните технологични параметри.

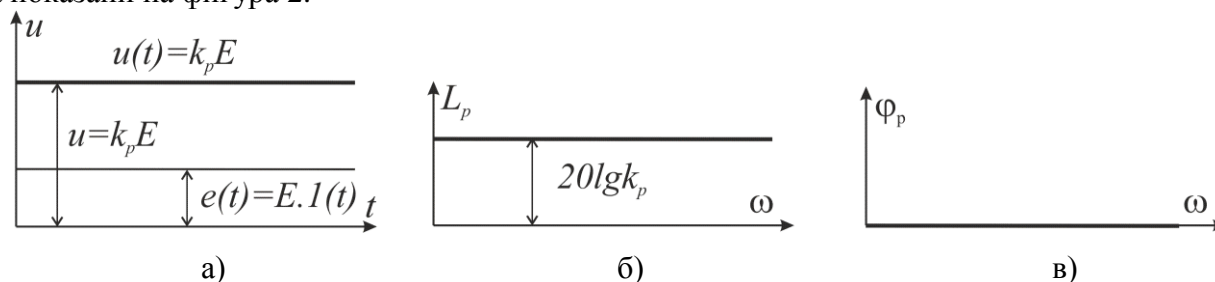
2.1. Пропорционален (П) закон за регулиране

Уравнението, което отразява връзката между входната и изходната променлива на пропорционалния регулатор (фигура 1), има вида:

$$u(t) = k_p[y_{зад} - y(t)] = k_p e(t), \quad (1)$$

където k_p се нарича коефициент на предаване (пропорционалност) на регулатора.

Следователно, при пропорционалния регулатор регулиращото въздействие е пропорционално на грешката на регулиране. Преходната функция на регулатора и логаритмичните му амлитудно-честотна (ЛАЧХ) и фазово-честотна (ЛФЧХ) характеристики са показани на фигура 2.



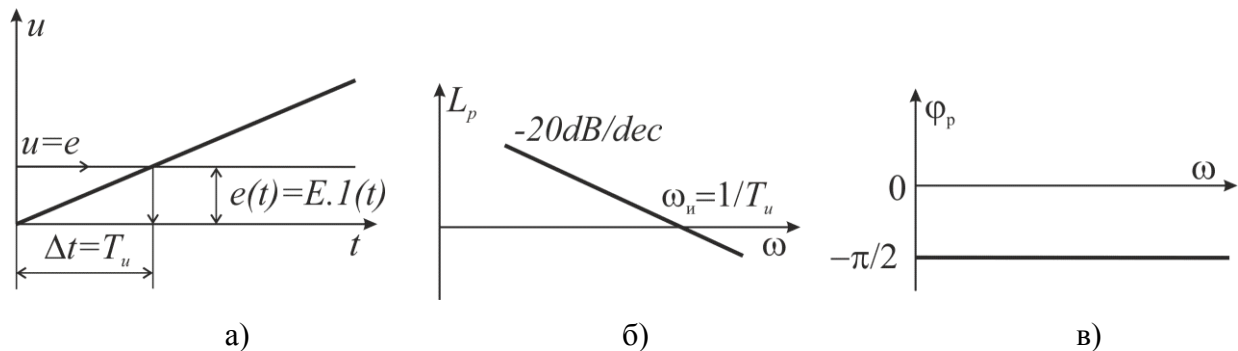
Фиг. 2. Преходна функция (а), ЛАЧХ (б) и ЛФЧХ (в) на П-регулатор

2.2. Интегрален (И) закон за регулиране

Регулиращото въздействие $u(t)$, което изработва И-регулатор, е пропорционално на интеграла от грешката на регулиране $e(t)$, т.е. във формирането на $u(t)$ в даден момент се отчита грешката от всички предходни моменти от стартирането на процеса. Диференциалното уравнение, с което се описва преобразуването на грешката на регулиране в управляващо въздействие е:

$$u(t) = \frac{1}{T_u} \int_0^t e(\tau) d\tau, \quad (2)$$

където T_u се нарича времеконстанта на интегриране. Последната има дименсия време и представлява времето, необходимо на регулатора да изработи на изхода си величина равна по стойност на входната. Преходната функция, ЛАЧХ и ЛФЧХ на И-регулатор са показани на фигура 3.



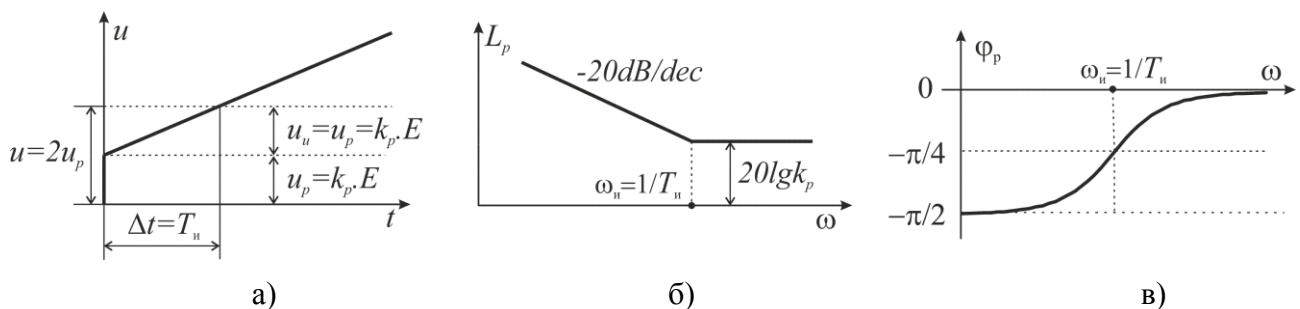
Фиг. 3. Преходна функция (а), ЛАЧХ (б) и ЛФЧХ (в) на И-регулатор

2.3. Пропорционално-интегрален (ПИ) закон за регулиране

Уравнението, с което се описва изходната величина на пропорционално-интегралния регулатор, е:

$$u(t) = k_p \left[e(t) + \frac{1}{T_u} \int_0^t e(\tau) d\tau \right], \quad (3)$$

където k_p е коефициент на предаване, а T_u - времеконстанта на интегриране и са основни параметри за настройка на регулатора. Регулиращото въздействие, изработвано от ПИ-регулатора, е сума от две съставки: пропорционална на грешката на регулиране и пропорционална на интеграла от тази грешка. Преходната му функция, ЛАЧХ и ЛФЧХ са показани на фигура 4.



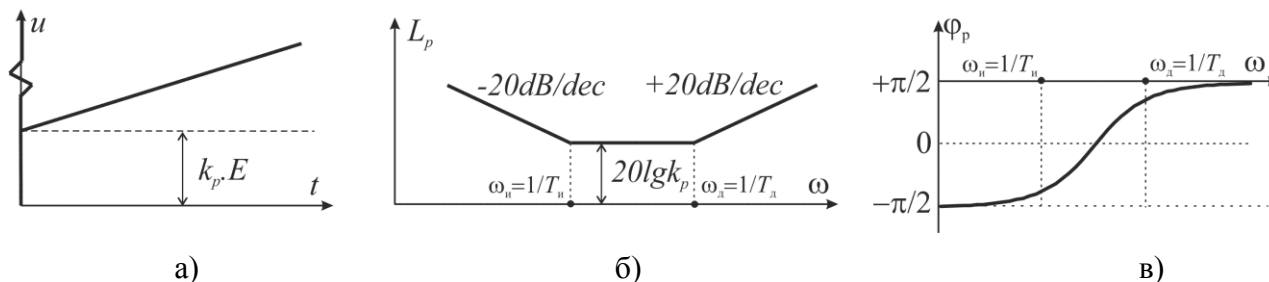
Фиг.4. Преходна функция (а), ЛАЧХ (б) и ЛФЧХ (в) на ПИ-регулатор

2.4. Пропорционално-интегрално-диференциален (ПИД) закон за регулиране

Уравнението, с което се описва зависимостта на изходния сигнал $u(t)$ на ПИД-регулатора от грешката на регулиране $e(t)$, е:

$$u(t) = k_p \left[e(t) + \frac{1}{T_u} \int_0^t e(t) dt + T_d \frac{de(t)}{dt} \right], \quad (4)$$

където k_p е коефициент на предаване, T_u е времеконстанта на интегриране, а T_d - времеконстанта на диференциране. Регулиращото въздействие, изработвано от ПИД-регулатора, е сума от три съставки: пропорционална на грешката на регулиране, на интеграла от тази грешка и на първата ѝ производна. Преходната му функция, ЛАЧХ и ЛФЧХ са показани на фигура 5.



Фиг. 5. Преходна функция (а), ЛАЧХ (б) и ЛФЧХ (в) на ПИД-регулатор

3. Постановка на задачата

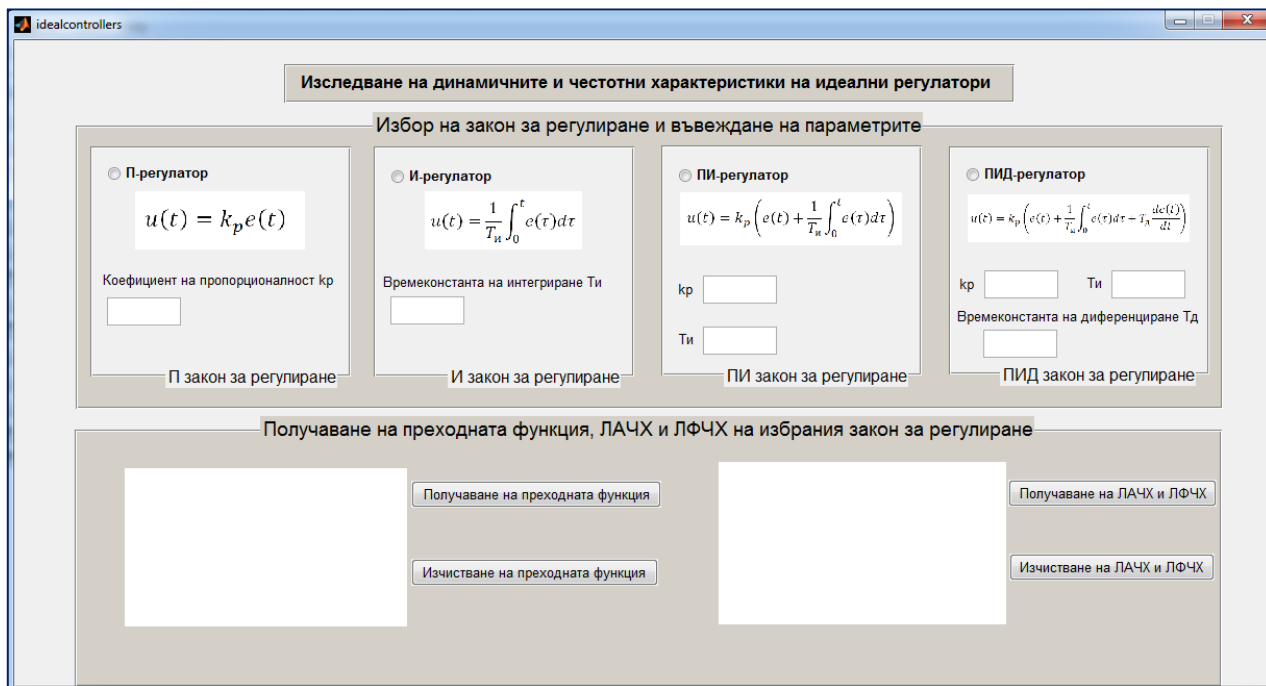
Получаването и изследването на динамичните и честотни характеристики на идеалните линейни регулатори може да се извърши с използване на програмен продукт MATLAB. Една от възможностите е създаване на m-функция, в която чрез функциите `step(.)` и `bode(.)` да се получават преходната функция и ЛАЧХ и ЛФЧХ.

Друга възможност е изграждането на графичен потребителски интерфейс (Graphical User Interface) в средата на MATLAB [5,6]. GUI на дадено приложение представлява един или няколко графични прозореца, върху които са разположени полета за въвеждане на данни, падащи менюта, координатни оси за извеждане на графики и др. В процеса на създаване на графичния интерфейс гъвкаво и лесно могат да се добавят нови елементи или да се модифицират вече добавени според нуждите на конкретната задача, да се тества тяхната работа и т.н.

В настоящата работа се цели разработване на GUI, чрез който да се осигури възможност за избор на закон за регулиране, за въвеждане на параметрите в избрания закон и получаване на преходната функция и/или логаритмичните честотни характеристики на избрания закон.

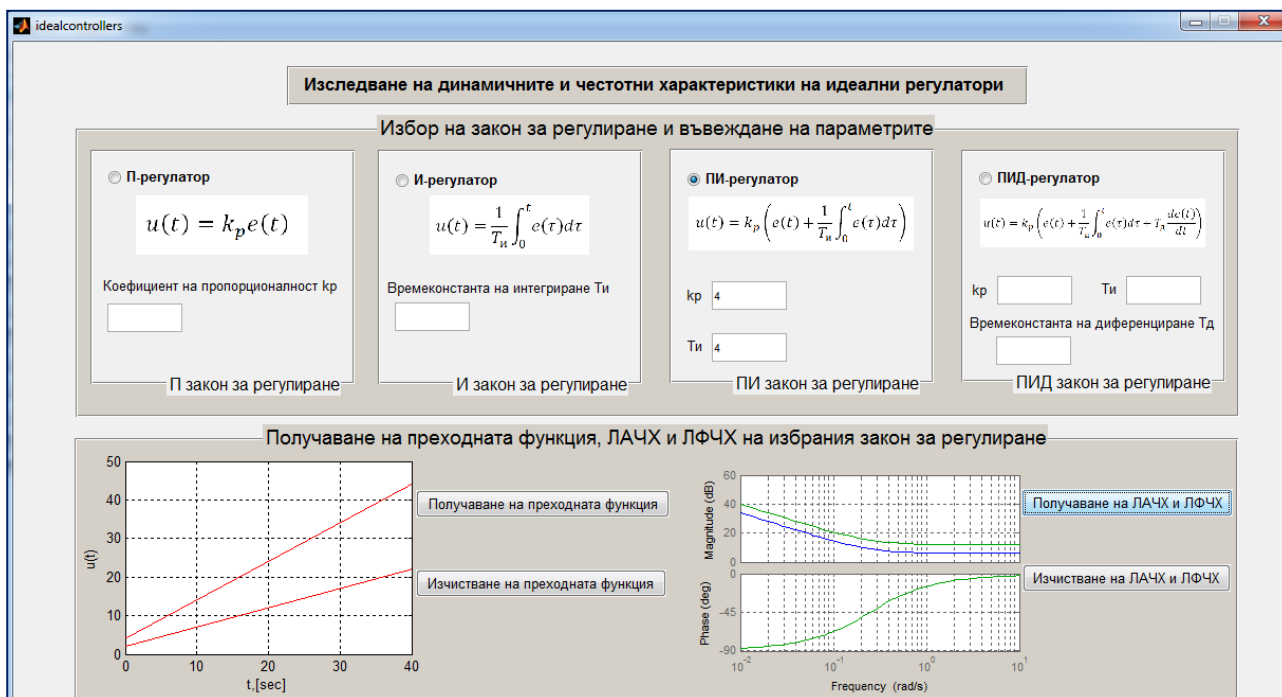
4. Разработване на GUI "Изследване на динамичните и честотни характеристики на идеални регулатори"

Графичният потребителски интерфейс е разработен с използване на `guide` в MATLAB [5,6] и се състои от два панела, както е показано на фигура 6. Първият, "Избор на закон за регулиране и въвеждане на параметрите" позволява избор на закон за регулиране и последващо въвеждане на параметрите му за настройка. Потребителят може да избира между П, И, ПИ и ПИД закон за регулиране.



Фиг. 6. GUI "Изследване на динамичните и честотни характеристики на идеални регулатори"

Във втория панел, "Получаване на преходната функция, ЛАЧХ и ЛФЧХ на избрания закон за регулиране", са разположени два графични прозореца, един за преходната функция и един за ЛАЧХ и ЛФЧХ. До всеки от графичните прозорци са разположени по два бутона - за получаване и за изчистване на вече изчертана характеристика.



Фиг.7. Получаване на преходната функция, ЛАЧХ и ЛФЧХ в GUI "Изследване на динамичните и честотни характеристики на идеални регулатори"

Преходната функция се изчертава с помощта на функцията $\text{step}(\cdot)$ и се визуализира след натискане на бутон "Получаване на преходната функция" (фигура 7). При промяна на параметър и повторно натискане на бутона новата преходна функция се изчертава в същата координатна система, като старата се запазва. Последното дава възможност да се оцени

влиянието на някой от параметрите в избрания закон за регулиране. Чрез бутон "Изчистване на преходната функция" се изчистват изчертаните досега преходни функции и потребителят може да започне изследване на влиянието на някой от останалите параметри или да премине към избор и изследване на друг закон за регулиране.

Честотните характеристики на избрания закон се изчертават с помощта на функция `bode(.)`. Аналогично на преходната функция, и тук е осигурена възможност за сравнение на характеристиките при промяна на някой от настроените параметри, като новите ЛАЧХ и ЛФЧХ се изчертават в същата координатна система за сравнение и анализ. Изчистването им също се осъществява с бутон.

Програмният код на разработения интерфейс е показан по-долу.

GUI idealcontrollers.m

```
%-----
%Графичен потребителски интерфейс за получаване и изследване на
% динамичните и честотните характеристики на идеални линейни регулатори
%-----

%Дефиниране на GUI: определяне на входните/изходните му променливи
function varargout = idealcontrollers(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name', mfilename, ...
    'gui_Singleton', gui_Singleton, ...
    'gui_OpeningFcn', @idealcontrollers_OpeningFcn, ...
    'gui_OutputFcn', @idealcontrollers_OutputFcn, ...
    'gui_LayoutFcn', [] , ...
    'gui_Callback', []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end
if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

%Отваряне на графичния интерфейс
function idealcontrollers_OpeningFcn(hObject, eventdata, handles, varargin)
    warning off all
    clc;
    i=1; handles.ws.i=i;
    axes(handles.axes1); I1 = imread('Equation_PID.png'); imshow(I1)
    axes(handles.axes2); I2 = imread('Equation_PI.png'); imshow(I2)
    axes(handles.axes3); I3=imread('Equation_I.png'); imshow(I3)
    axes(handles.axes4); I4 = imread('Equation_P.png'); imshow(I4)
    axes(handles.axes5); I5=imread('Blank_axes.jpg'); imshow(I5)
    axes(handles.axes6); I6=imread('Blank_axes.jpg'); imshow(I6)
    handles.output = hObject; guidata(hObject, handles);

% Изходите от тази функция се връщат към командната лента
function varargout = idealcontrollers_OutputFcn(hObject, eventdata, handles)
varargout{1} = handles.output;

% Избиране на закон за регулиране
% Изпълнява се при избор на ПИД закон за регулиране
function radiobutton1_Callback(hObject, eventdata, handles)
    if get(hObject,'Value')==1
        set(handles.radiobutton2,'Value',0);
        set(handles.radiobutton3,'Value',0);
```



```

        set(handles.radiobutton4,'Value',0);
    end
% Изпълнява се при избор на ПИ закон за регулиране
function radiobutton2_Callback(hObject, eventdata, handles)
    if get(hObject,'Value')==1
        set(handles.radiobutton1,'Value',0);
        set(handles.radiobutton3,'Value',0);
        set(handles.radiobutton4,'Value',0);
    end
% Изпълнява се при избор на И закон за регулиране
function radiobutton3_Callback(hObject, eventdata, handles)
    if get(hObject,'Value')==1
        set(handles.radiobutton1,'Value',0);
        set(handles.radiobutton2,'Value',0);
        set(handles.radiobutton4,'Value',0);
    end
% Изпълнява се при избор на П закон за регулиране
function radiobutton4_Callback(hObject, eventdata, handles)
    if get(hObject,'Value')==1
        set(handles.radiobutton1,'Value',0);
        set(handles.radiobutton2,'Value',0);
        set(handles.radiobutton3,'Value',0);
    end

% Създаване на полетата за въвеждане на настроечните параметри в закона за регулиране
% Създаване на поле edit1 за въвеждане на стойност на кр за ПИД закон
function edit1_Callback(hObject, eventdata, handles)
function edit1_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
% Създаване на поле edit2 за въвеждане на стойност на Ти за ПИД закон
function edit2_Callback(hObject, eventdata, handles)
function edit2_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
% Създаване на поле edit3 за въвеждане на стойност на Тд за ПИД закон
function edit3_Callback(hObject, eventdata, handles)
function edit3_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
% Създаване на поле edit4 за въвеждане на стойност на кр за ПИ закон
function edit4_Callback(hObject, eventdata, handles)
function edit4_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
% Създаване на поле edit5 за въвеждане на стойност на Ти за ПИ закон
function edit5_Callback(hObject, eventdata, handles)
function edit5_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
% Създаване на поле edit6 за въвеждане на стойност на Ти за И закон
function edit6_Callback(hObject, eventdata, handles)
function edit6_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

end
%Създаване на поле edit7 за въвеждане на стойност на kp за П закон
function edit7_Callback(hObject, eventdata, handles)
function edit7_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

% Получаване на преходната функция за избрания закон за регулиране
% Изпълнява се при натискане на бутон "Получаване на преходната функция"

```

```

function pushbutton1_Callback(hObject, eventdata, handles)
% Изпълнява се при избран П-регулатор
if get(handles.radiobutton4,'Value')==1
    edit7 = str2num(char(get(handles.edit7,'String')));
    kp=edit7; num=[kp];den=[1];
    axes(handles.axes5);SYS=tf(num,den);
    [Y,T]=step(SYS); plot(T,Y,'-r');grid on
    xlabel('t,[sec]');ylabel('u(t)'); axis([0 max(T) 0 kp+1]); hold on
% Изпълнява се при избран И-регулатор
elseif get(handles.radiobutton3,'Value')==1
    edit6 = str2num(char(get(handles.edit6,'String')));
    Ti=edit6; num=[1];den=[Ti 0];
    axes(handles.axes5);SYS=tf(num,den);
    [Y,T]=step(SYS); plot(T,Y,'-r');grid on
    xlabel('t,[sec]');ylabel('u(t)'); hold on
% Изпълнява се при избран ПИ-регулатор
elseif get(handles.radiobutton2,'Value')==1
    edit5 = str2num(char(get(handles.edit5,'String')));
    edit4 = str2num(char(get(handles.edit4,'String')));
    Ti=edit5;kp=edit4; num=[kp*Ti kp];den=[Ti 0];
    axes(handles.axes5);SYS=tf(num,den);
    [Y,T]=step(SYS); plot(T,Y,'-r');grid on
    xlabel('t,[sec]');ylabel('u(t)'); hold on
% Изпълнява се при избран ПИД-регулатор
elseif get(handles.radiobutton1,'Value')==1
    edit3 = str2num(char(get(handles.edit3,'String')));
    edit2 = str2num(char(get(handles.edit2,'String')));
    edit1 = str2num(char(get(handles.edit1,'String')));
    kp=edit1; Ti=edit2;Td=edit3; num=[kp*Ti*Td kp*Ti kp];den=[Ti*0.01 Ti 0];
    axes(handles.axes5);SYS=tf(num,den); time=4*Td;
    [Y,T]=step(SYS,time); plot(T,Y,'-r');grid on
    xlabel('t,[sec]');ylabel('u(t)'); axis([0 4*Td 0 5*kp]); hold on
end

```

```

% Изпълнява се при натискане на бутон "Изчистване на преходната функция"
function pushbutton2_Callback(hObject, eventdata, handles)
cla(handles.axes5)

```

```

% Получаване на честотните характеристики за избрания закон за регулиране
% Изпълнява се при натискане на бутон "Получаване на ЛАЧХ и ЛФЧХ"

```

```

function pushbutton3_Callback(hObject, eventdata, handles)
% Изпълнява се при избран П-регулатор
if get(handles.radiobutton4,'Value')==1
    edit7 = str2num(char(get(handles.edit7,'String')));
    kp=edit7; num=[kp];den=[1];
    axes(handles.axes6);SYS=tf(num,den);
    bode(SYS);grid on; hold on
% Изпълнява се при избран И-регулатор
elseif get(handles.radiobutton3,'Value')==1
    edit6 = str2num(char(get(handles.edit6,'String')));

```

```

Ti=edit6; num=[1];den=[Ti 0];
axes(handles.axes6);SYS=tf(num,den);
bode(SYS);grid on; hold on
% Изпълнява се при избран ПИ-регулатор
elseif get(handles radiobutton2,'Value')==1
    edit5 = str2num(char(get(handles.edit5,'String')));
    edit4 = str2num(char(get(handles.edit4,'String')));
    Ti=edit5;kp=edit4; num=[kp*Ti kp];den=[Ti 0];
    axes(handles.axes6);SYS=tf(num,den);
    bode(SYS);grid on; hold on
% Изпълнява се при избран ПИД-регулатор
elseif get(handles radiobutton1,'Value')==1
    edit3 = str2num(char(get(handles.edit3,'String')));
    edit2 = str2num(char(get(handles.edit2,'String')));
    edit1 = str2num(char(get(handles.edit1,'String')));
    Ti=edit1;kp=edit2;Td=edit3; num=[kp*Ti*Td kp*Ti kp];den=[Ti 0];
    axes(handles.axes6);SYS=tf(num,den);
    bode(SYS);grid on; hold on
end
% Изпълнява се при натискане на бутон "Изичстване на ЛАЧХ и ЛФЧХ"
function pushbutton4_Callback(hObject, eventdata, handles)
cla(handles.axes6)

```

5. Заключение

В настоящата работа е разработен графичен потребителски интерфейс в средата на MATLAB за получаване и изследване на преходните функции на идеални линейни закони за регулиране. Потребителят може да избира между няколко закона за регулиране - пропорционален (П), интегрален (И), пропорционално-интегрален (ПИ) и пропорционално-интегрално-диференциален (ПИД). След избора на закон и въвеждане на настроените параметри се изчертава преходната функция като при промяна на някой от параметрите може да се изследва неговото влияние.

Създаденият графичен потребителски интерфейс позволява получаване и изследване и на логаритмичните честотни характеристики (ЛАЧХ и ЛФЧХ) на избрания закон за регулиране. Аналогично на получаването и изследването на преходната функция, и тук е осигурена възможност да се изследва влиянието на някой от настроените параметри в закона върху неговите честотни характеристики.

Разработеният интерфейс би могъл лесно да бъде разширен с получаване и на други характеристики (например АФХ, АЧХ, ФЧХ) или с добавяне и на други закони за регулиране (например пропорционално-диференциален ПД), тъй като GUI позволява добавяне и модифициране на елементи както по време на създаването му, така и в последствие.

GUI "Изследване на динамичните и честотни характеристики на идеални регулатори" ще бъде използван за лабораторните упражнения по дисциплината "Технически средства за автоматизация - 2 част" от учебния план на специалности АИУКС и АРУКС на ТУ-Варна.

Литература

- [1]. Драготинов И., Г.Терзийски, Ж. Стойчев, К. Кутрянски, Технически средства за автоматизация, УХТ Пловдив, 2015.
- [2]. Костов К., Е. Николов, Технически средства за автоматизация, ТУ-София, 1988.
- [3]. Петров П., Я.Янев, Технически средства за автоматизация, ТУ-Варна, 2012.
- [4]. Стоянов В., Елементи на системи за автоматизация, Университетски издателски център на РУ "А. Кънчев", Русе, 2015.

- [5]. Тончев Й., "Matlab 7, част III, Преобразувания, изчисления, визуализация", Техника, София, 2009.
- [6]. <https://www.mathworks.com/discovery/matlab-gui.html>, последен достъп 2017.

За контакти:

гл. ас. д-р Мариела И. Александрова
катедра „Автоматизация на производството”
Технически университет-Варна
E-mail: m_alexandrova@tu-varna.bg



РАЗРАБОТВАНЕ НА GUI В MATLAB ЗА ИЗСЛЕДВАНЕ НА ДИНАМИЧНИТЕ ХАРАКТЕРИСТИКИ НА ПРОМИШЛЕНИ РЕГУЛАТОРИ

Мариела И. Александрова

Резюме: Целта на настоящата работа е разработване на графичен потребителски интерфейс GUI (Graphical User Interface) в среда MATLAB за получаване и изследване на динамичните характеристики на промишлени регулатори. Графичният интерфейс ще позволява избор на закон за регулиране и избор на структура за неговото реализиране, като след въвеждане на стойности на настройечните параметри ще се изчертава преходната функция на реалния регулатор.

Ключови думи: графичен потребителски интерфейс (GUI), преходна функция, закон за регулиране, реален промишлен регулатор

Development of MATLAB GUI for Investigation of Step Responses of Real Industrial Controllers

Mariela I. Alexandrova

Abstract: The aim of this paper is development of GUI (Graphical User Interface) in MATLAB for investigation of the step responses of industrial controllers. The developed GUI allows for choice of control strategy and controller structure as well as setting of control strategy parameters in order the step response of the structure chosen to be obtained.

Keywords: Graphical User Interface (GUI), step response, control strategy, real industrial controller

1. Въведение

Статичните и динамични характеристики на промишлените регулатори се различават от характеристиките на идеалните регулатори. Това трябва да се има предвид, тъй като оптималните настройки се изчисляват за идеален регулатор. Често това допускане води до влошаване на качеството на преходния процес или до поява на неустойчивост в системата. Познаването на особеностите на реалните регулатори позволява да се оцени в кои случаи изчисленията на базата на идеалния регулатор са достоверни. Разработването на графичен потребителски интерфейс GUI в среда MATLAB ще улесни получаването на динамичните характеристики на реални промишлени регулатори, както и изследването на влиянието на параметрите на регулаторите върху тези характеристики.

2. Структури за реализиране на промишлени регулатори

Предавателната функция на всеки реален регулатор може да се представи във вида:

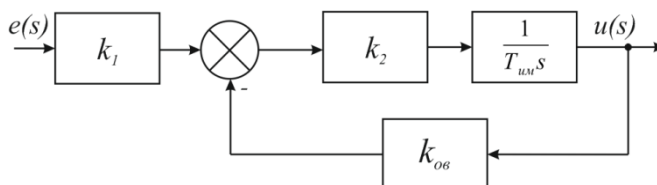
$$W_{pp}(s) = W_{up}(s) \cdot W_{\delta z}(s), \quad (1)$$

където с $W_{up}(s)$ и $W_{\delta z}(s)$ са означени съответно предавателните функции на идеалния регулатор, реализиращ точно даден закон за регулиране, и на баластното звено. Наличието на баластно звено води до изкривяване на динамичните характеристики на реалния регулатор спрямо тези на идеалния.

В литературата са известни различни структури за реализиране на основните типови линейни закони за регулиране [1,2,3,4]. В настоящата работа са разгледани някои структури за реализиране на пропорционален (П), пропорционално-интегрален (ПИ) и пропорционално-интегрално-диференциален (ПИД) закон за регулиране.

2.1. Структура, реализираща П закон за регулиране

На фигура 1 е показана структурата, по която се реализира промишлен пропорционален регулатор.



Фиг. 1. Структура за реализиране на П-регулатор

С k_1 е означен коефициентът на предаване на входния усилвател, а с k_2 - коефициентът на предаване на усилвателя на мощност за управление на изпълнителния механизъм (ИМ). С $T_{ум}$ е означена времеконстантата на ИМ, а с $k_{об}$ - коефициентът на предаване на обратната връзка.

Предавателната функция на регулатора, изграден по показаната структура, е:

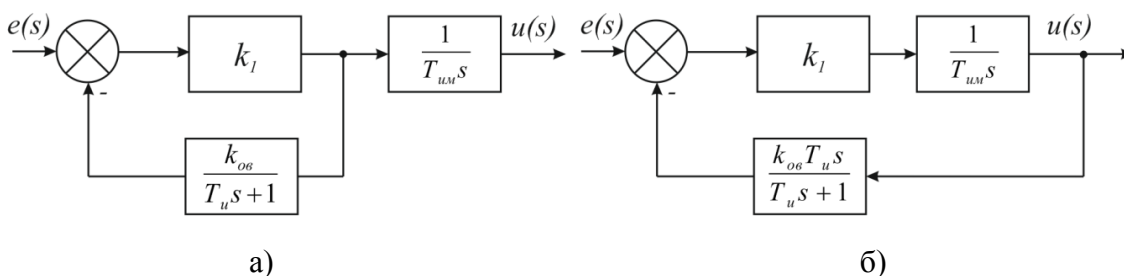
$$W_{pp}(s) = \frac{k_1}{k_{об}} \cdot \frac{1}{\frac{T_{ум}}{k_2 k_{об}} s + 1} = k_p \cdot W_{\bar{o}3}(s), \quad (2)$$

където $k_p = \frac{k_1}{k_{об}}$; $W_{\bar{o}3}(s) = \frac{1}{T_{\bar{o}3}s + 1}$; $T_{\bar{o}3} = \frac{T_{ум}}{k_2 k_{об}}$.

Следователно, реалният пропорционален регулатор може да се представи като последователно свързани идеален П-регулатор и баластно (паразитно) звено. Коефициентът на предаване на регулатора може да се настройва чрез промяна на коефициентите k_1 и $k_{об}$.

2.2. Структури, реализиращи ПИ закон за регулиране

На фигура 2 са показани две от най-често използваните структури за реализиране на реален ПИ закон за регулиране. Разликата между двете структури се състои в това дали обратната връзка (ОВ) обхваща (фигура 2б) или не обхваща (фигура 2а) изпълнителния механизъм.



Фиг. 2. Структури за реализиране на ПИ-регулатор

Предавателната функция на реален ПИ-регулатор, изграден по структурата от фигура 2а, е:

$$W_{pp}(s) = k_p \left(1 + \frac{1}{T_u s} \right) \cdot \frac{k_{\bar{o}3}}{(T_{\bar{o}3}s + 1)} = W_{up}(s) \cdot W_{\bar{o}3}(s), \quad (3)$$

където: $k_p = \frac{1}{k_{o6}}$; $k_{\bar{o}3} = \frac{k_1 T_u}{T_{um}(k_p + k_1)}$; $T_{\bar{o}3} = \frac{k_p T_u}{k_p + k_1}$.

За структурата, показана на фигура 2б, са в сила следните зависимости:

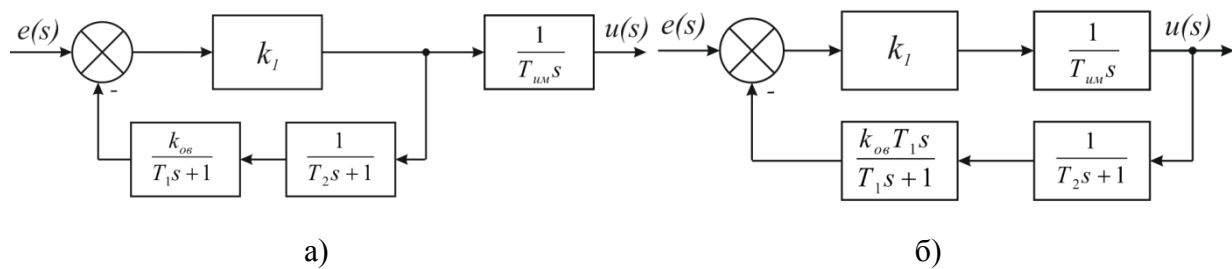
$$W_{pp}(s) = k_p \left(1 + \frac{1}{T_u s} \right) \cdot \frac{k_{\bar{o}3}}{(T_{\bar{o}3}s + 1)} = W_{up}(s) \cdot W_{\bar{o}3}(s), \quad)$$

където: $k_p = \frac{1}{k_{o6}}$; $k_{\bar{o}3} = \frac{k_1 T_u}{k_p T_{um} + k_1 T_u}$; $T_{\bar{o}3} = \frac{k_p T_u T_{um}}{k_p T_{um} + k_1 T_u}$.

Следователно, реалният ПИ-регулатор може да се представи като последователно съединение от идеален такъв и баластно апериодично звено от първи ред. Коефициентът на предаване на регулатора може да се настройва чрез промяна на коефициента в обратната връзка - k_{o6} .

2.3. Структури, реализиращи ПИД закон за регулиране

На фигура 3 са показани две от най-често използваните структури за реализиране на реален ПИД закон за регулиране. Аналогично на структурите за реализиране на ПИ закон за регулиране, и тук разликата се обуславя от това дали ОБ обхваща (фигура 3б) или не обхваща (фигура 3а) изпълнителния механизъм.



Фиг. 3. Структури за реализиране на ПИД-регулатор

Предавателната функция на реален ПИД-регулатор, изграден по структурата от фигура 3а, е:

$$W_{pp}(s) = k_p \left(1 + \frac{1}{T_u s} + T_{\bar{o}} s \right) \left(\frac{1/c}{\frac{a}{c}s^2 + \frac{b}{c}s + 1} \right) = W_{up}(s) \cdot W_{\bar{o}3}(s), \quad (5)$$

където: $k_p = \frac{1}{k_{o6}} \cdot \frac{T_1 + T_2}{T_{um}}$, $T_u = T_1 + T_2$, $T_{\bar{o}} = \frac{T_1 T_2}{T_1 + T_2}$, $a = \frac{T_1 T_2}{k_1 k_{o6}}$, $b = \frac{T_1 + T_2}{k_1 k_{o6}}$, $c = \frac{1}{k_1 k_{o6}} + 1$.

За структурата, показана на фигура 3б, са в сила следните зависимости:

$$W_{pp}(s) = k_p \left(1 + \frac{1}{T_u s} + T_{\bar{o}} s \right) \cdot \frac{k_1 k_{o6} T_{um}/c}{\frac{a}{c}s^2 + \frac{b}{c}s + 1} = W_{up}(s) \cdot W_{\bar{o}3}(s), \quad (6)$$

където: $k_p = \frac{1}{k_{o6}} \cdot \frac{T_1 + T_2}{T_{um}}$, $T_u = T_1 + T_2$, $T_{\bar{o}} = \frac{T_1 T_2}{T_1 + T_2}$, $a = T_1 T_2 T_{um}$, $b = (T_1 + T_2) T_{um}$, $c = T_{um} + T_1 k_1 k_{o6}$.

Следователно, реалният ПИД регулатор може да се представи като последователно съединение от идеален такъв и баластно колебателно звено.

Както се вижда от зависимости (2)-(6), основни параметри за настройка в показаните структури са: - коефициентът на предаване в правия канал - k_1 (за всички структури); k_{ov} - коефициентът на предаване в обратната връзка (за всички структури); $T_{им}$ - времеконстантата на изпълнителния механизъм (за всички структури); T_u - времеконстантата на интегриране (за структурите, реализиращи ПИ закон); T_1 и T_2 - времеконстантите в структурите, реализиращи ПИД закон.

3. Постановка на задачата

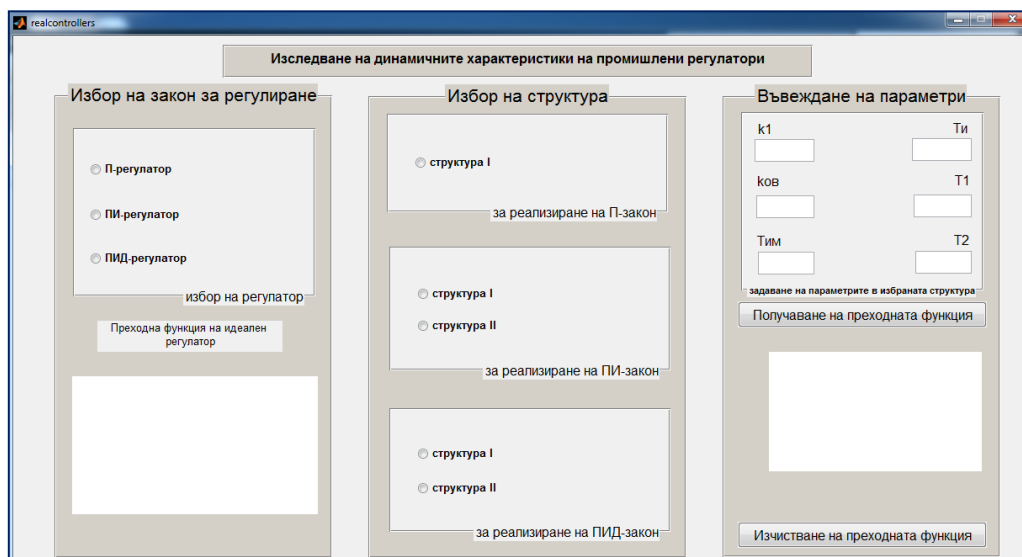
Получаването и изследването на преходните функции на различните структури е важно по отношение на познаването на качествата на динамичните характеристики на всяка една от тях с цел избор на конкретна структура за реализиране на закон за регулиране. То може да се осъществи с използване на програмен продукт MATLAB. Една от възможните реализации включва създаване на Simulink модел, в който да се изгради съответната структура и параметрите в закона за регулиране да се променят преди стартиране на симулацията.

Друга възможна реализация е изграждане на графичен потребителски интерфейс (GUI), чрез който да се осигури възможност да се избират закон за регулиране и структура за неговото реализиране, да се въвеждат настройките параметри в избраната структура и като последна стъпка да се изчертава преходната функция при направените настройки [5,6].

GUI на дадено приложение представлява един или няколко графични прозореца, върху които са разположени ленти с плъзгачи, полета за въвеждане на данни, менюта (включително падащи), координатни оси за извеждане на графики и др. В процеса на създаване на графичния интерфейс може да се добавят постепенно нови елементи, да се тества тяхната работа и т.н. Следователно, създаденият GUI ще позволява гъвкаво и лесно добавяне на нови структури, полета за въвеждане на параметри, както и допълнителни функции (за получаване например на честотните характеристики на избран закон за регулиране).

4. Разработване на GUI "Изследване на динамичните характеристики на промишлени регулатори"

Графичният потребителски интерфейс е разработен с използване на guide в MATLAB [5,6]. Състои се от три панела, както е показано на фигура 4.

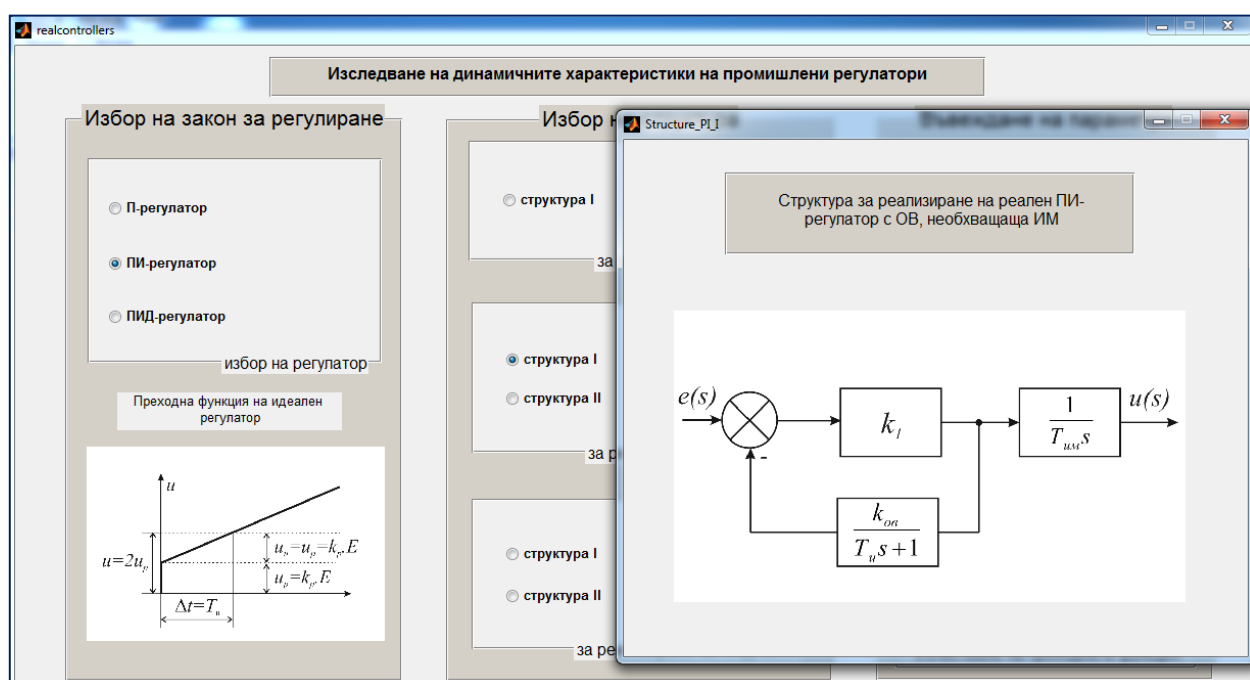


Фиг. 4. GUI "Изследване на динамичните характеристики на промишлени регулатори"

В първият панел, "Избор на закон за регулиране", се осъществява избор на закона за регулиране, за който ще се получават и изследват динамичните характеристики. Възможните закони за избор са: - пропорционален (П), пропорционално-интегрален (ПИ) и пропорционално-интегрален-диференциален (ПИД). При избор на някой от тях, в долната част на панела се визуализира преходната функция на идеалния регулатор.

Вторият панел, озаглавен "Избор на структура", осигурява възможност за избор на структура за реализиране на промишления регулатор. Заложени са по две структури за реализиране на ПИ-регулатор (фиг.2) и ПИД-регулатор (фигура 3), и една за реализиране на П-регулатор (фиг.1). При избор на структура съответната се отваря в отделен графичен прозорец, както е показано на фигура 5.

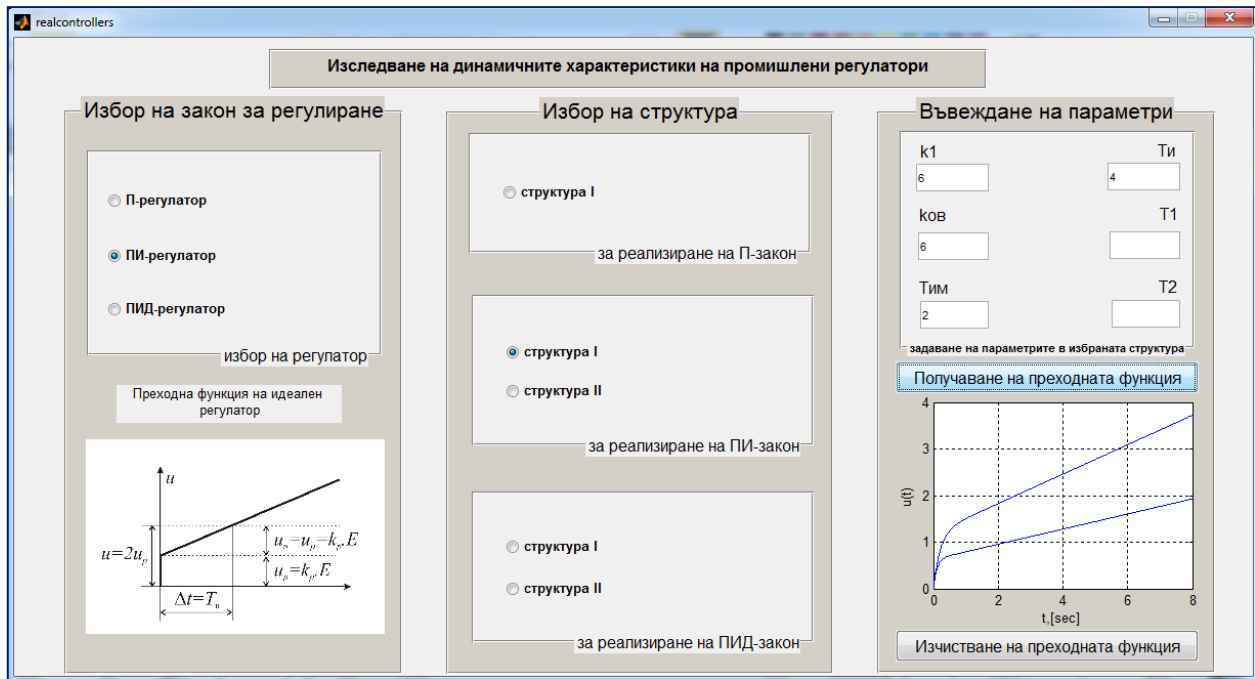
Третият панел, "Задаване на параметри", дава възможност на потребителя да въведе стойности на параметрите от избраната структура. Достъпни за въвеждане са параметрите: k_1 - коефициент на предаване в правия канал; $k_{ов}$ - коефициент на предаване на обратната връзка; $T_{им}$ - времеконстанта на изпълнителния механизъм; T_u - времеконстанта на интегриране; T_1 и T_2 - времеконстанти в структурите, реализиращи ПИД закон. Според избраната структура, в празните полета се въвеждат тези параметри, които участват в нея.



Фиг. 5. Избор на структура в GUI "Изследване на динамичните характеристики на промишлени регулатори"

Полетата на параметри, които не участват в избраната структура, се оставят празни. Дори в тях да се въведе параметър, който не участва в структурата, той не се използва в изчисленията за получаването на преходната функция. След въвеждането параметрите се използват за изчисляване на коефициентите в числителя и знаменателя на предавателната функция на избраната структура. Преходната функция се изчертава с помощта на функцията `step(.)` и се визуализира след натискане на бутон "Получаване на преходната функция" (фигура 6). При промяна на параметър и повторно натискане на бутона новата преходна функция се изчертава в същата координатна система, като старата се запазва. Последното дава възможност да се оцени влиянието на някой от параметрите в структурата за реализиране на избрания закон за регулиране. Под координатната система, в която се изчертават преходните функции, е разположен и бутон "Изчистване на преходната функция", който позволява да се изчистят изчертаните досега преходни функции и потребителят да

започне изследване на влиянието на някой от останалите параметри или да премине към избор и изследване на друга структура и/или друг закон за регулиране.



Фиг. 6. Получаване на преходната функция в GUI "Изследване на динамичните характеристики на промишлени регулатори"

Програмният код на разработения интерфейс е показан по-долу.

GUI idealcontrollers.m

```
%-----
%Графичен потребителски интерфейс за получаване и изследване на
%динамичните характеристики на промишлени регулатори
%-----

%Дефиниране на GUI: определяне на входните/изходните му променливи
function varargout = realcontrollers(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',    mfilename, ...
    'gui_Singleton', gui_Singleton, ...
    'gui_OpeningFcn', @realcontrollers_OpeningFcn, ...
    'gui_OutputFcn', @realcontrollers_OutputFcn, ...
    'gui_LayoutFcn', [] , ...
    'gui_Callback', []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end
if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

%Отваряне на графичния интерфейс
function realcontrollers_OpeningFcn(hObject, eventdata, handles, varargin)
    warning off all
    clc; i=1; handles.ws.i=i;
```

```

axes(handles.axes1); I1 = imread('Blank_axes.jpg'); imshow(I1)
axes(handles.axes3); I2=imread('Blank_axes.jpg'); imshow(I2)
handles.output = hObject; guidata(hObject, handles);
% Изходите от тази функция се връщат към командната лента
function varargout = realcontrollers_OutputFcn(hObject, eventdata, handles)
varargout{1} = handles.output;

% Изпълнява се при избор на П-регулатор
function radiobutton1_Callback(hObject, eventdata, handles)
if get(hObject,'Value')==1
    set(handles.radiobutton2,'Value',0); set(handles.radiobutton3,'Value',0)
    axes(handles.axes1) % избор на axes1, така че тя става текуща
    I1=imread('Step_Response_P.bmp'); imshow(I1)
    handles.output = hObject; guidata(hObject, handles);
else
    axes(handles.axes1)
    I1=imread('Blank_axes.jpg'); imshow(I1);
end

% Изпълнява се при избор на ПИ-регулатор
function radiobutton2_Callback(hObject, eventdata, handles)
if get(hObject,'Value')==1
    set(handles.radiobutton1,'Value',0); set(handles.radiobutton3,'Value',0)
    axes(handles.axes1); I1=imread('Step_Response_PI.bmp'); imshow(I1)
else
    axes(handles.axes1); I1=imread('Blank_axes.jpg'); imshow(I1)
end

% Изпълнява се при избор на ПИД-регулатор
function radiobutton3_Callback(hObject, eventdata, handles)
if get(hObject,'Value')==1
    set(handles.radiobutton1,'Value',0); set(handles.radiobutton2,'Value',0)
    axes(handles.axes1)
    I1=imread('Step_Response_PID.bmp'); imshow(I1)
end

% Изпълнява се при избор на структура за реализиране на П закон
function radiobutton6_Callback(hObject, eventdata, handles)
if get(hObject,'Value')==1
    set(handles.radiobutton5,'Value',0); set(handles.radiobutton11,'Value',0);
    set(handles.radiobutton12,'Value',0); set(handles.radiobutton13,'Value',0);
    Structure_P
end

% Изпълнява се при избор на структура I за реализиране на ПИ закон
function radiobutton11_Callback(hObject, eventdata, handles)
if get(hObject,'Value')==1
    set(handles.radiobutton5,'Value',0); set(handles.radiobutton6,'Value',0);
    set(handles.radiobutton12,'Value',0); set(handles.radiobutton13,'Value',0);
    Structure_PI_I
end

% Изпълнява се при избор на структура II за реализиране на ПИ закон
function radiobutton13_Callback(hObject, eventdata, handles)
if get(hObject,'Value')==1
    set(handles.radiobutton5,'Value',0); set(handles.radiobutton6,'Value',0);
    set(handles.radiobutton11,'Value',0); set(handles.radiobutton12,'Value',0);
    Structure_PI_II
end

% Изпълнява се при избор на структура I за реализиране на ПИД закон
function radiobutton12_Callback(hObject, eventdata, handles)
if get(hObject,'Value')==1
    set(handles.radiobutton5,'Value',0); set(handles.radiobutton6,'Value',0);
    Structure_PID_I
end

```

```

% Изпълнява се при избор на структура II за реализиране на ПИД закон
function radiobutton5_Callback(hObject, eventdata, handles)
    if get(hObject,'Value')==1
        set(handles.radiobutton6,'Value',0); set(handles.radiobutton11,'Value',0);
        set(handles.radiobutton12,'Value',0); set(handles.radiobutton13,'Value',0);
        Structure_PID_II
    end

% Създаване на полетата за въвеждане на настройките параметри в закона за регулиране
% Създаване на поле edit1 за въвеждане на стойност на kp
function edit1_Callback(hObject, eventdata, handles)
function edit1_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
function edit1_KeyPressFcn(hObject, eventdata, handles)
% Създаване на поле edit2 за въвеждане на стойност на kov
function edit2_Callback(hObject, eventdata, handles)
function edit2_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
% Създаване на поле edit3 за въвеждане на стойност на Тим
function edit3_Callback(hObject, eventdata, handles)
function edit3_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
% Създаване на поле edit5 за въвеждане на стойност на Ти
function edit5_Callback(hObject, eventdata, handles)
function edit5_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
% Създаване на поле edit6 за въвеждане на стойност на T1
function edit6_Callback(hObject, eventdata, handles)
function edit6_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
% Създаване на поле edit7 за въвеждане на стойност на T2
function text7_CreateFcn(hObject, eventdata, handles)
function edit7_Callback(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
% Създаване на обект axes2
function axes2_CreateFcn(hObject, eventdata, handles)

% Получаване на преходната функция
% Изпълнява се при натискане на бутон "Получаване на преходната функция"
function pushbutton2_Callback(hObject, eventdata, handles)
% Получаване на преходната функция при избран П-регулатор
if get(handles.radiobutton1,'Value')==1 && get(handles.radiobutton6,'Value')==1
    edit1 = str2num(char(get(handles.edit1,'String'))); edit2 = str2num(char(get(handles.edit2,'String')));
    edit3 = str2num(char(get(handles.edit3,'String')));
    k1=edit1;kov=edit2;Tim=edit3; num=[k1];den=[Tim/1,kov];
    axes(handles.axes3);SYS=tf(num,den);
    [Y,T]=step(SYS); plot(T,Y);grid on; xlabel('t,[sec]');ylabel('u(t)'); hold on

```

```

% Получаване на преходната функция при избран ПИ-регулатор и структура I
elseif get(handles.radiobutton2,'Value')==1 && get(handles.radiobutton11,'Value')==1
    edit1 = str2num(char(get(handles.edit1,'String'))); edit2 = str2num(char(get(handles.edit2,'String')));
    edit3 = str2num(char(get(handles.edit3,'String'))); edit5 = str2num(char(get(handles.edit5,'String')));
    k1=edit1;kov=edit2;Tim=edit3;Tint=edit5;kp=1/kov;
    Tbz=(kp*Tint)/(kp+k1);kbz=(k1*Tint)/(kp+k1);
    num=[kp*Tint*kbz, kp*kbz];den=[Tint*Tbz Tint 0];
    axes(handles.axes3);SYS=tf(num,den);time=2*Tint;
    [Y,T]=step(SYS,time); plot(T,Y);grid on; xlabel('t,[sec]');ylabel('u(t)'); hold on

% Получаване на преходната функция при избран ПИ-регулатор и структура II
elseif get(handles.radiobutton2,'Value')==1 && get(handles.radiobutton13,'Value')==1
    set(handles.pushbutton2,'Enable','on');
    edit1 = str2num(char(get(handles.edit1,'String'))); edit2 = str2num(char(get(handles.edit2,'String')));
    edit3 = str2num(char(get(handles.edit3,'String'))); edit5 = str2num(char(get(handles.edit5,'String')));
    k1=edit1;kov=edit2;Tim=edit3;Tint=edit5;kp=1/kov;
    Tbz=(kp*Tint*Tim)/(kp*Tim+k1*Tint);kbz=(k1*Tint)/(kp*Tim+k1*Tint);
    num=[kp*Tint*kbz kp*kbz];den=[Tint*Tbz Tint 0];
    axes(handles.axes3);SYS=tf(num,den);time=2*Tint;
    [Y,T]=step(SYS,time); plot(T,Y);grid on; xlabel('t,[sec]');ylabel('u(t)'); hold on

% Получаване на преходната функция при избран ПИД-регулатор и структура I
elseif get(handles.radiobutton3,'Value')==1 && get(handles.radiobutton12,'Value')==1
    set(handles.pushbutton2,'Enable','on');
    edit1 = str2num(char(get(handles.edit1,'String'))); edit2 = str2num(char(get(handles.edit2,'String')));
    edit3 = str2num(char(get(handles.edit3,'String'))); edit6 = str2num(char(get(handles.edit6,'String')));
    edit7 = str2num(char(get(handles.edit7,'String')));
    k1=edit1;kov=edit2;Tim=edit3;T1=edit6;T2=edit7;kp=(1/kov)*((T1+T2)/(Tim));
    Tint=T1+T2;Td=(T1*T2)/(T1+T2);a=(T1*T2)/(k1*kov);b=(T1+T2)/(k1*kov);c=(1/(k1*kov))+1;
    num=[kp*Tint*Td kp*Tint 1];den=[a*Tint b*Tint c*Tint 0];
    axes(handles.axes3);SYS=tf(num,den);
    [Y,T]=step(SYS); plot(T,Y);grid on; xlabel('t,[sec]');ylabel('u(t)'); hold on

% Получаване на преходната функция при избран ПИД-регулатор и структура II
elseif get(handles.radiobutton3,'Value')==1 && get(handles.radiobutton5,'Value')==1
    set(handles.pushbutton2,'Enable','on');
    edit1 = str2num(char(get(handles.edit1,'String'))); edit2 = str2num(char(get(handles.edit2,'String')));
    edit3 = str2num(char(get(handles.edit3,'String'))); edit6 = str2num(char(get(handles.edit6,'String')));
    edit7 = str2num(char(get(handles.edit7,'String')));
    k1=edit1;kov=edit2;Tim=edit3;T1=edit6;T2=edit7;kp=(1/kov)*((T1+T2)/(Tim));
    Tint=T1+T2;Td=(T1*T2)/(T1+T2);a=T1*T2*Tim;b=(T1+T2)*Tim;c=Tim+T1*k1*kov;
    num=[kp*k1*kov*Tint*Td*Tim kp*k1*kov*Tint*Tim kp*k1*kov*Tim];den=[a*Tint b*Tint c*Tint 0];
    axes(handles.axes3);SYS=tf(num,den);
    [Y,T]=step(SYS); plot(T,Y);grid on; xlabel('t,[sec]');ylabel('u(t)'); hold on
end

% Изпълнява се при натискане на бутон "Изчистване на преходната функция"
function pushbutton3_Callback(hObject, eventdata, handles)
cla(handles.axes3);

```

5. Заключение

В настоящата работа е създаден графичен потребителски интерфейс в средата на MATLAB, който позволява получаването на преходните функции на промишлени регулатори. Потребителят на GUI може да избира измежду три закона за регулиране: пропорционален (П), пропорционално-интегрален (ПИ) и пропорционално-интегрално-диференциален (ПИД). За реализиране на П-регулатор е заложена една структура, а за реализиране на ПИ- и ПИД-регулатори са заложиени по две структури. След избор на

структура и въвеждане на настройките параметри в нея се изчертава преходната функция, като при промяна на някой от параметрите може да се изследва и анализира неговото влияние.

Разработеният интерфейс лесно би могъл да бъде разширен или модифициран чрез добавяне на други закони за регулиране (например интегрален - И, пропорционално-диференциален - ПД) и/или други структури за вече включените в интерфейса закони.

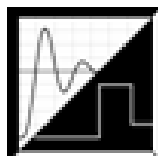
Създаденият GUI "Изследване на динамичните характеристики на промишлени регулатори" ще бъде използван за лабораторните упражнения по дисциплината "Технически средства за автоматизация - 2 част" от учебния план на специалности АИУКС и АРУКС на ТУ-Варна.

Литература

- [1]. Драготинов И., Г.Терзийски, Ж. Стойчев, К. Кутрянски, Технически средства за автоматизация, УХТ Пловдив, 2015.
- [2]. Костов К., Е. Николов, Технически средства за автоматизация, ТУ-София, 1988.
- [3]. Петров П., Я.Янев, Технически средства за автоматизация, ТУ-Варна, 2012.
- [4]. Стоянов В., Елементи на системи за автоматизация, Университетски издателски център на РУ "А. Кънчев", Русе, 2015.
- [5]. Тончев Й., "Matlab 7, част III, Преобразувания, изчисления, визуализация", Техника, София, 2009.
- [6]. <https://www.mathworks.com/discovery/matlab-gui.html>, последен достъп 2017.

За контакти:

гл. ас. д-р Мариела И. Александрова
катедра „Автоматизация на производството”
Технически университет - Варна
E-mail: m_alexandrova@tu-varna.bg



DEVELOPING A MODEL OF ADAPTIVE STATE OBSERVER FOR SISO LINEAR STATIONARY DISCRETE SYSTEMS IN MATLAB/SIMULINK ENVIRONMENT

Veselin Lukov

Abstract: The article presents the developing a model of adaptive state observer for single input-single output linear stationary discrete systems, based on non-recurrent algorithm for adaptive observation, using the least squares method. Examples, confirming the synthesized model workability will be presented to support the conclusions.

Keywords: control, adaptive, algorithm, observer, discrete, system

Разработване на модел на адаптивен наблюдател на състоянието за SISO линейни стационарни дискретни системи в средата SIMULINK на MATLAB

Веселин Луков

Резюме: Статията представя разработването на модел на адаптивен наблюдател на състоянието за линейни стационарни дискретни системи с един вход и един изход, базирани на нерекурентен алгоритъм за адаптивно наблюдение, използващ метода на най-малките квадрати. Примери, потвърждаващи работоспособността на синтезирания модел, ще бъдат представени в подкрепа на заключенията.

Ключови думи: управление, адаптивен, алгоритъм, наблюдател, дискретни, система

1. Introduction

In the modeling of control systems with implementation of state feedback, it is usually needed the state vector to be reconstructed by the measured values of the input and output variables of the plant [11].

The algorithm used for determination of the state vector is called state observer. The adaptive observation task is being solved by synthesis of adaptive observers with parameter estimators [2,4,5,6]. This is needed, because the matrices A and b or (depending on the canonical form chosen for state-space representation) are considered unknown.

During the observation process the parameters should be estimated and the unknown matrices determined, allowing reconstruction of the state vector.

For Takagi-Sugeno controller multi-model control synthesis it is wanted that the state vector is estimated. This publication will present a Simulink model of non-recurrent algorithm for adaptive observation of SISO linear stationary discrete systems, based on the least square method with examples, proving the model workability.

2. Problem definition

The SISO Linear stationary discrete system under investigation is presented in the state space as follows:

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{A}\mathbf{x}(k) + \mathbf{b}u(k), & \mathbf{x}(0) &= \mathbf{x}_0, \\ y(k) &= \mathbf{c}^T \mathbf{x}(k) + f(k), & k &= 0, 1, 2, \dots \end{aligned} \quad (1)$$

where:

$$\mathbf{A} = \begin{bmatrix} \mathbf{0} & \vdots & \mathbf{I}_{n-1} \\ \dots & \dots & \dots \\ & \mathbf{a}^T & \end{bmatrix}, \quad (2)$$

$$\mathbf{a} = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix}, \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}, \mathbf{c} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}. \quad (3)$$

The order of the system n is known, $\mathbf{x}(k) \in \mathbb{R}^n$ unknown vector of the current state of the system, $\mathbf{x}(0) \in \mathbb{R}^n$ is unknown initial state vector, $u(k) \in \mathbb{R}^1$ is scalar input, $y(k) \in \mathbb{R}^1$ is scalar output, \mathbf{a} and \mathbf{b} are unknown vector parameters.

The corresponding discrete transfer function to state space representation, given with (1) is:

$$W(z) = \frac{h_1 z^{n-1} + h_2 z^{n-2} + \dots + h_{n-1} z + h_n}{z^n - a_n z^{n-1} - \dots - a_2 z - a_1}. \quad (4)$$

The relation between the elements b_i of the vector \mathbf{b} , in accordance to the canonical phase-coordinate form chosen and the polynomial coefficients h_i in the numerator of the discrete transfer function (4) can be described with the following [7]:

$$\mathbf{T}\mathbf{b} = \mathbf{h}, \quad (5)$$

where:

$$\mathbf{h}^T = [h_1 \quad h_2 \quad \dots \quad h_n], \quad (6)$$

$$\mathbf{T} = \begin{bmatrix} 1 & 0 & \dots & 0 & 0 \\ -a_n & 1 & \dots & 0 & 0 \\ -a_{n-1} & -a_n & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ -a_2 & -a_3 & \dots & -a_n & 1 \end{bmatrix}. \quad (7)$$

The elements a_i of vector \mathbf{a} , according to the phase-coordinate canonical form chosen are the coefficients of the denominator of (4), in the opposite order with negative sign.

The task defined is aiming estimation of the unknown vector parameters \mathbf{a} and \mathbf{b} , initial state vector $\mathbf{x}(0)$ and current state vector $\mathbf{x}(k)$, $k=1, 2, \dots$

3. Structure and definition of adaptive observer algorithm

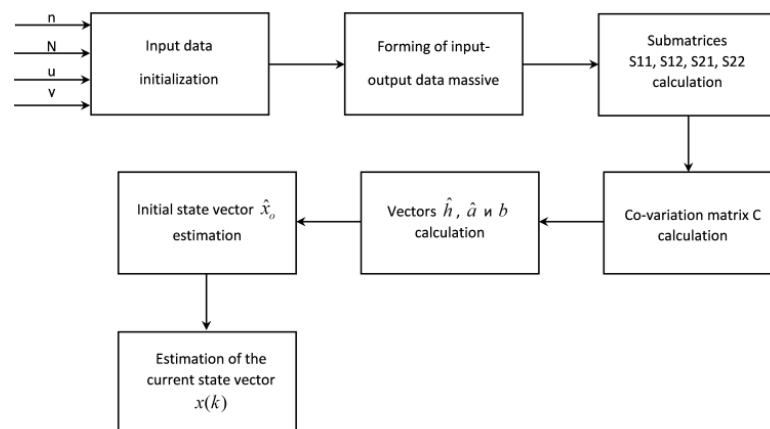


Fig. 1. Block-diagram of the algorithm

Block diagram of the algorithm developed is presented in Fig.1. During the initialization of input data processing are setup the order of the system n and the number of the input-output data $N \geq 3n$. The experiment will use $N=3n$. After the initialization stage follows the procedure for input-output data values forming (with input variable u and response y), which are renewed in every iteration.

Once the input-output data is loaded – respectively vector \mathbf{y} with $3n$ and vector \mathbf{u} with $3n-1$ values, the input-output data massive are formed:

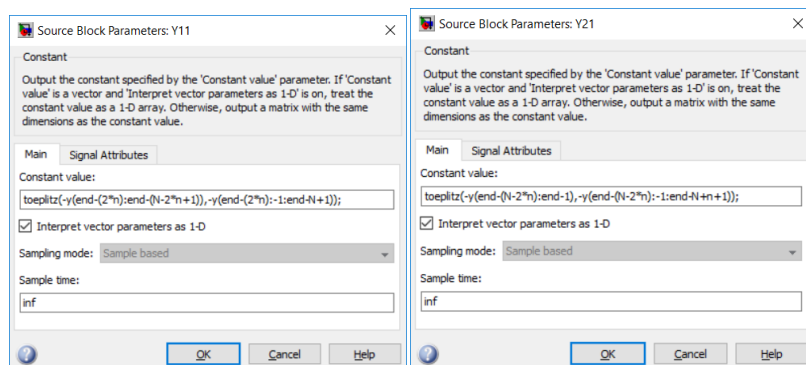
$$\mathbf{Y}_{11} = \begin{bmatrix} -y(\text{end}-2n) & -y(\text{end}-2n-1) & \cdots & -y(\text{end}-N+1) \\ -y(\text{end}-2n+1) & -y(\text{end}-2n) & \cdots & -y(\text{end}-N+2) \\ -y(\text{end}-2n+2) & -y(\text{end}-2n+1) & \cdots & -y(\text{end}-N+3) \\ \vdots & \vdots & \ddots & \vdots \\ -y(\text{end}-(N-2n+1)) & -y(\text{end}-(N-2n+2)) & \cdots & -y(\text{end}-N+n) \end{bmatrix}, \quad (8)$$

$$\mathbf{Y}_{21} = \begin{bmatrix} -y(\text{end}-(N-2n)) & -y(\text{end}-(N-2n)-1) & \cdots & -y(\text{end}-N+n+1) \\ -y(\text{end}-(N-2n)+1) & -y(\text{end}-(N-2n)) & \cdots & -y(\text{end}-N+n+2) \\ -y(\text{end}-(N-2n)+2) & -y(\text{end}-(N-2n)+1) & \cdots & -y(\text{end}-N+n+3) \\ \vdots & \vdots & \ddots & \vdots \\ -y(\text{end}-1) & -y(\text{end}-2) & \cdots & -y(\text{end}-n) \end{bmatrix}, \quad (9)$$

$$\mathbf{U}_{12} = \begin{bmatrix} u(\text{end}-(2n-1)) & u(\text{end}-2n) & \cdots & u(\text{end}-N+2) \\ u(\text{end}-(2n-2)) & u(\text{end}-(2n-1)) & \cdots & u(\text{end}-N+3) \\ u(\text{end}-(2n-3)) & u(\text{end}-(2n-2)) & \cdots & u(\text{end}-N+4) \\ \vdots & \vdots & \ddots & \vdots \\ u(\text{end}-n) & u(\text{end}-n-1) & \cdots & u(\text{end}-N+n+1) \end{bmatrix}, \quad (10)$$

$$\mathbf{U}_{22} = \begin{bmatrix} u(\text{end}-(n-1)) & u(\text{end}-n) & \cdots & u(\text{end}-N+n+2) \\ u(\text{end}-(n-2)) & u(\text{end}-(n-1)) & \cdots & u(\text{end}-N+n+3) \\ u(\text{end}-(n-3)) & u(\text{end}-(n-2)) & \cdots & u(\text{end}-N+n+4) \\ \vdots & \vdots & \ddots & \vdots \\ u(\text{end}) & u(\text{end}-1) & \cdots & u(\text{end}-N+2n+1) \end{bmatrix} \quad (11)$$

It is modeled in Simulink Matlab Environment with blocks Constant (Fig.2.), loaded with equations to generate Toeplitz matrix with input and output data values.



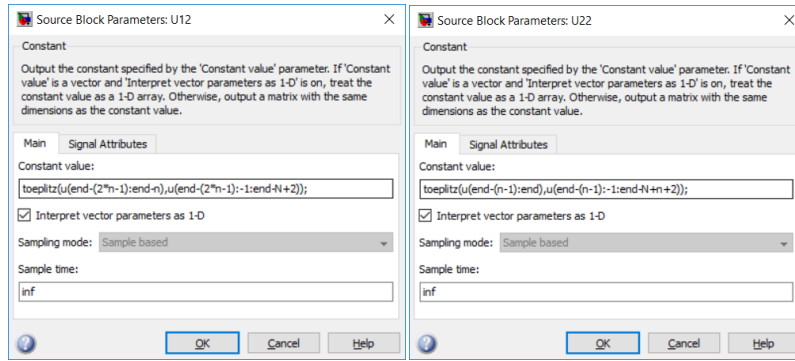


Fig. 2. Set-up of blocks “Constant” for input-output data massive forming

The calculation of the sub-matrices and the co-variation matrix \mathbf{C} is as follows:

$$\mathbf{C} = \begin{bmatrix} \mathbf{M}_1 + \mathbf{M}_1 \mathbf{S}_{12} \mathbf{M}_2 \mathbf{S}_{21} \mathbf{M}_1 & \vdots & -\mathbf{M}_1 \mathbf{S}_{12} \mathbf{M}_2 \\ \dots\dots\dots & \dots & \dots\dots\dots \\ -\mathbf{M}_2 \mathbf{S}_{21} \mathbf{M}_1 & \vdots & \mathbf{M}_2 \end{bmatrix}, \quad (12)$$

where:

$$\mathbf{M}_1 = \mathbf{S}_{11}^{-1}, \quad (13)$$

$$\mathbf{M}_2 = (\mathbf{S}_{22} - \mathbf{S}_{21} \mathbf{M}_1 \mathbf{S}_{12})^{-1}. \quad (14)$$

The calculations are performed in Embedded Matlab Function editor [1,3] and it is shown in Fig.3. In Fig.4. the Simulink structure is shown with the connections between Embedded Matlab Function blocks for the sub-matrices and the co-variation matrix \mathbf{C} calculation [7,8,9,10].

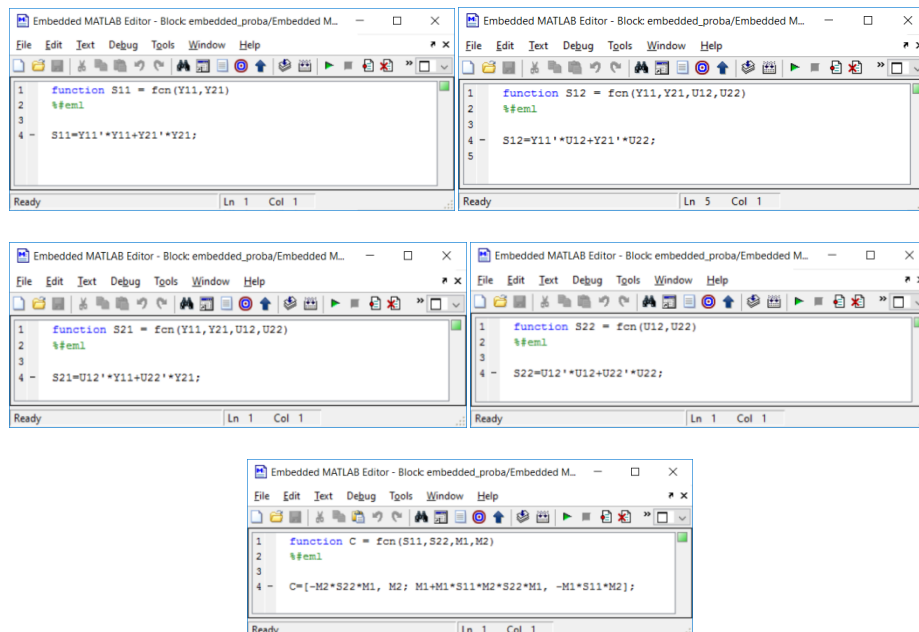


Fig. 3. Embedded MATLAB Function blocks for the sub-matrices and the co-variation matrix \mathbf{C} calculation

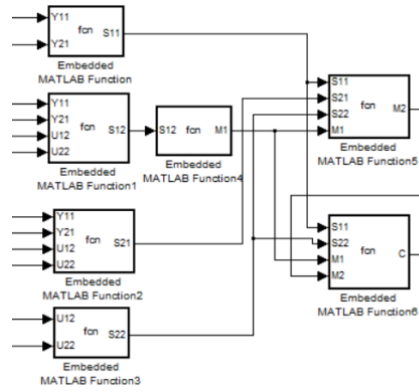


Fig. 4. Simulink structure with the connections between Embedded Matlab Function blocks for the sub-matrices and the co-variation matrix **C** calculation

The estimation of vectors **h** and **a** uses the following vector-matrix system of equations:

$$\hat{\mathbf{p}} = \mathbf{C} \begin{bmatrix} \mathbf{Y}_{11}^T \mathbf{y}_2 + \mathbf{Y}_{21}^T \mathbf{y}_3 \\ \mathbf{U}_{12}^T \mathbf{y}_2 + \mathbf{U}_{22}^T \mathbf{y}_3 \end{bmatrix}. \quad (15)$$

$$\hat{\mathbf{h}} = [\hat{h}_1 \quad \hat{h}_2 \quad \cdots \quad \hat{h}_n]^T = [\hat{p}_{n+1} \quad \hat{p}_{n+2} \quad \cdots \quad \hat{p}_{2n}]^T, \quad (16)$$

$$\hat{\mathbf{a}} = [\hat{a}_1 \quad \hat{a}_2 \quad \cdots \quad \hat{a}_n]^T = [-\hat{p}_n \quad -\hat{p}_{n-1} \quad \cdots \quad -\hat{p}_1]^T, \quad (17)$$

$$\hat{\mathbf{A}} = \begin{bmatrix} \mathbf{0} & \vdots & \mathbf{I}_{n-1} \\ \cdots & \cdots & \cdots \\ & \hat{\mathbf{a}}^T & \end{bmatrix} \quad (18)$$

It is modeled in Simulink through multiplexer/de-multiplexer system, shown in Fig.5.

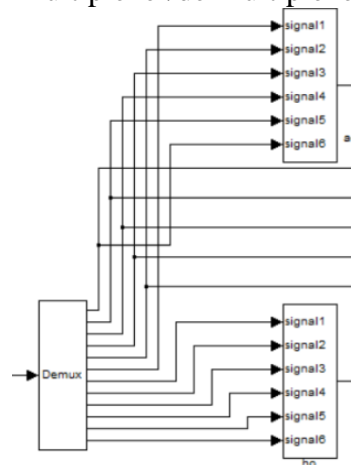


Fig. 5. Simulink structure for vectors **a** and **h** estimation

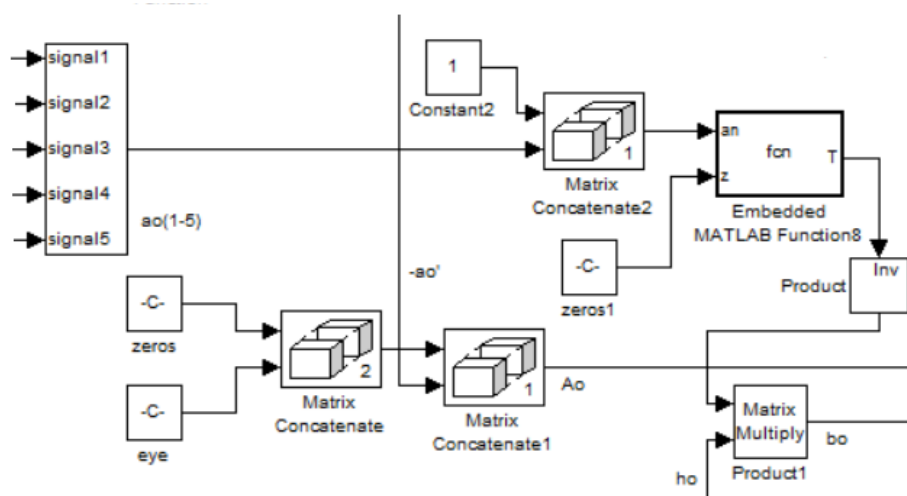
The estimation of vector **b** is according to the below equations system:

$$\mathbf{T}\hat{\mathbf{b}} = \hat{\mathbf{h}}, \quad (19)$$

where

$$\mathbf{T} = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 & 0 \\ -\hat{a}_n & 1 & 0 & \cdots & 0 & 0 \\ -\hat{a}_{n-1} & -\hat{a}_n & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ -\hat{a}_2 & -\hat{a}_3 & -\hat{a}_4 & \cdots & -\hat{a}_n & 1 \end{bmatrix} \quad (20)$$

In Simulink model the matrices **A** and **T**, as well as vector **b**, are interpreted with block for horizontal and vertical concatenation and block for matrix multiplication (Fig.6.). In Fig.7. is shown embedded Matlab Function block for matrix **T** calculation.



Embedded MATLAB Editor - Block: embedded_proba/Embedded M...

File Edit Text Debug Tools Window Help

function T = fcn(an,z)
 %#eml
 T=toeplitz(an,z);

Ready Ln 1 Col 1

For initial state vector $\hat{\mathbf{x}}_0$ estimation, first vector \mathbf{d} and matrix $\mathbf{D}\mathbf{b}$ are estimated in block Embedded MATLAB Function 9, shown in Fig.8.:

where:

$$\mathbf{D} = \begin{bmatrix} \mathbf{c}^T \\ \mathbf{c}^T \hat{\mathbf{A}} \\ \mathbf{c}^T \hat{\mathbf{A}}^2 \\ \vdots \\ \mathbf{c}^T \hat{\mathbf{A}}^{(N-1)} \end{bmatrix}_{(N \times n)}, \quad (22)$$

Matrix \mathbf{D} is formed using demultiplexer/multiplexer system, vertical concatenation block and block Embedded MATLAB Function 10, shown in Fig.9. The part $-\mathbf{D}\mathbf{u}^T$ from equation (16) is shown in Fig.10.

```

1 function [Db,d] = fcn(c,Ao,bo,N)
2 %#eml
3 Db=zeros(18,1);
4 d=zeros(18,6);
5 for j=1:N
6     Db(j,:)=c'*Ao^(j-1)*bo;
7     d(j,:)=c'*Ao^(j-1);
8 end

```

Fig. 8. Embedded MATLAB Function block for vector **d** and matrix **Db** calculation

```

1 function D = fcn(Db,zr)
2 %#eml
3
4 D=toeplitz(Db,zr);

```

Fig. 9. Embedded MATLAB Function block for matrix **D** calculation

```

1 function Dut = fcn(D,ut)
2 %#eml
3
4 Dut=D*ut';

```

Fig. 10. Embedded MATLAB Function block for $-\mathbf{D}\mathbf{u}^T$ calculation

Then using the needed block for algebraic operations is estimated the initial state vector $\hat{\mathbf{x}}_0$, shown in Fig.11.

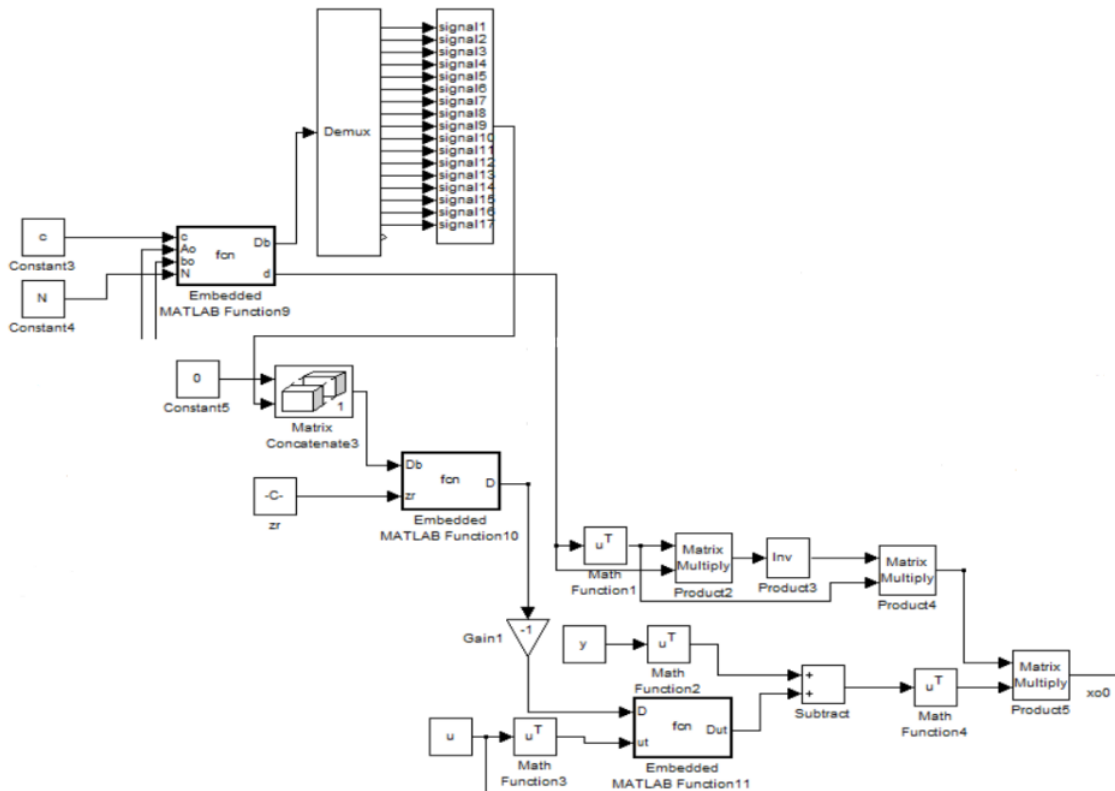


Fig. 11. Simulink structure for the initial state vector $\hat{\mathbf{x}}_0$ estimation

The current state vector $\mathbf{x}(k)$ is obtained as follows:

$$\hat{\mathbf{x}}(k+1) = \hat{\mathbf{F}}\hat{\mathbf{x}}(k) + \hat{\mathbf{b}}u(k) + \mathbf{g}y(k), \quad \hat{\mathbf{x}}(0) = \hat{\mathbf{x}}_0; \quad (24)$$

$$\hat{\mathbf{F}} = \hat{\mathbf{A}} - \mathbf{g}\mathbf{c}^T, \quad (25)$$

Vector \mathbf{g} can be derived using predefined pole placement procedure. It is recommended that the vector \mathbf{g} is synthesized in a manner that matrix $\hat{\mathbf{F}}$ will have zero eigenvalues or at least such smaller than the eigenvalues of matrix $\hat{\mathbf{A}}$.

The Simulink modeling structure used for calculation of vector \mathbf{g} , matrix $\hat{\mathbf{F}}$ and the current state vector $\mathbf{x}(k)$ is shown in Fig.12. In Fig.13. are shown Embedded MATLAB Function blocks for the last mentioned vectors and matrix calculation.

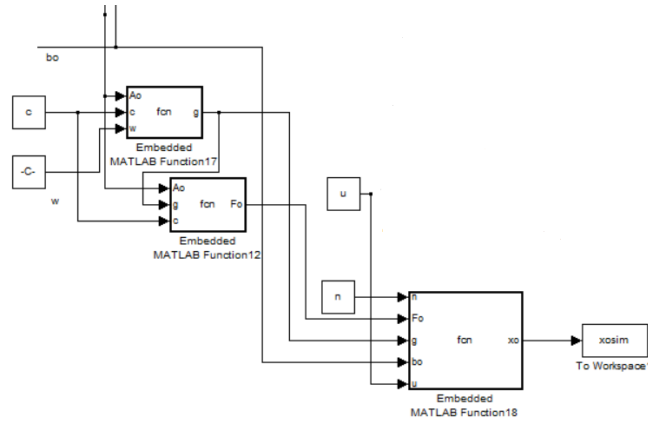


Fig. 12. Simulink structure for the vector \mathbf{g} and the matrix $\hat{\mathbf{F}}$ calculation and the current state vector $\mathbf{x}(k)$ estimation.

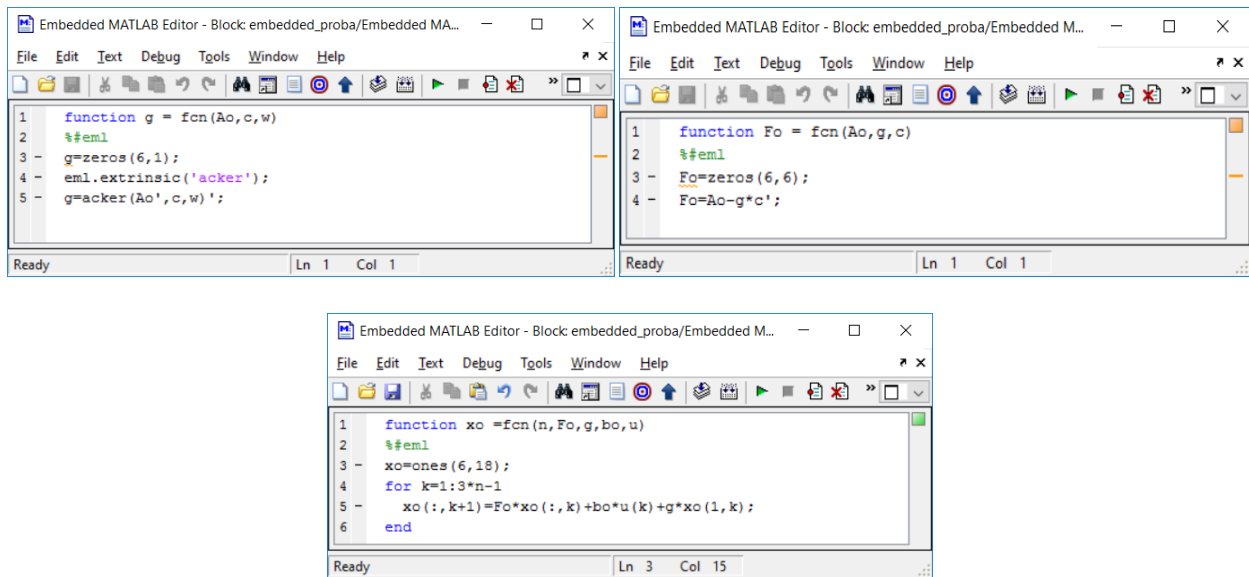


Fig. 13. Embedded MATLAB Function blocks for the vector \mathbf{g} and the matrix $\hat{\mathbf{F}}$ calculation and the current state vector $\mathbf{x}(k)$ estimation.

4. Adaptive observer modeling and performance check

The whole adaptive observer modeling structure in Simulink is shown in Fig.14.

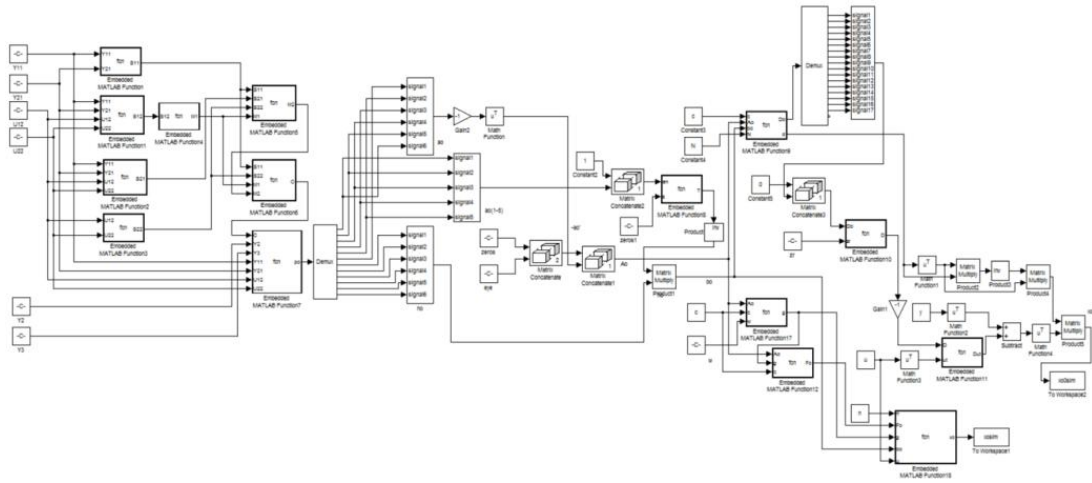


Fig. 14. Simulink structure of the adaptive observer

The model will be tested in offline regime, without noise applied to the output. For this purpose is set a file with input and output values to be loaded in the Workspace of Matlab. Once the simulation of the modeling structure is started the output automatically shows the estimations of vectors **a**, **b**, **x₀** and the current state vector estimation **x(k)**:

$$\hat{\mathbf{a}} = \begin{bmatrix} -1.4 \\ 0.7875 \\ -0.2275 \\ 0.0355 \\ -0.0028 \\ 0.0001 \end{bmatrix}, \quad \hat{\mathbf{b}} = \begin{bmatrix} 0.6 \\ 0.2 \\ 0.1 \\ 0.3 \\ 0.4 \\ 0.5 \end{bmatrix}, \quad \hat{\mathbf{x}}_0 = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix},$$

$$\hat{\mathbf{x}} = \begin{bmatrix} 1 & 7 & 9 & 10 & 1 & 13 & 19.8072 & 22.7641 & 23.33 & 21.1262 & 7.1163 & 27.2022 & 24.6040 & 31.9016 & 21.5634 & 12.5566 & -6.117 & 6.4499 \\ 1 & 3 & 4 & 7 & 7 & 13.8072 & 16.7641 & 17.33 & 15.1262 & 13.1163 & 21.2022 & 30.6040 & 25.9016 & 27.5634 & 18.5566 & -0.117 & 12.4499 & 14.1674 \\ 1 & 2 & 5 & 9 & 11.8072 & 14.7641 & 15.33 & 13.1262 & 11.1163 & 23.2022 & 28.6040 & 27.9016 & 25.5634 & 20.5566 & 1.8830 & 14.4499 & 16.1674 & 14.5803 \\ 1 & 4 & 8 & 12.8072 & 13.7641 & 14.33 & 12.1262 & 10.1163 & 22.2022 & 29.6040 & 26.9016 & 26.5634 & 19.5566 & 2.8830 & 15.4499 & 17.1674 & 15.5803 & 3.5101 \\ 1 & 5 & 9.8072 & 16.7641 & 11.33 & 9.1262 & 7.1163 & 19.2022 & 26.6040 & 29.9016 & 23.5634 & 22.5566 & -0.117 & 18.4499 & 20.1674 & 18.5803 & 6.5101 & 11.4006 \\ 1 & 5.8072 & 12.7641 & 15.33 & 5.1262 & 3.1163 & 15.2022 & 22.6040 & 25.9016 & 27.5634 & 18.5566 & 3.8830 & 14.4499 & 24.1674 & 22.5803 & 10.5101 & 15.4006 & -20.2264 \end{bmatrix}$$

The actual values of vectors **a** and **b** are preliminary known:

$$\mathbf{a} = \begin{bmatrix} -1.4 \\ 0.7875 \\ -0.2275 \\ 0.0355 \\ -0.0028 \\ 0.0001 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 0.6 \\ 0.2 \\ 0.1 \\ 0.3 \\ 0.4 \\ 0.5 \end{bmatrix}, \quad \mathbf{x}_0 = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix},$$

The current state vector value **x(k)**, is estimated:

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{b}u(k), \quad \mathbf{x}(0) = \mathbf{x}_0. \quad (26)$$

$$\mathbf{x} = \begin{bmatrix} 1 & 7 & 9 & 10 & 1 & 13 & 19.8072 & 22.7641 & 23.33 & 21.1262 & 7.1163 & 27.2022 & 24.6040 & 31.9016 & 21.5634 & 12.5566 & -6.117 & 6.4499 \\ 1 & 3 & 4 & 7 & 7 & 13.8072 & 16.7641 & 17.33 & 15.1262 & 13.1163 & 21.2022 & 30.6040 & 25.9016 & 27.5634 & 18.5566 & -0.117 & 12.4499 & 14.1674 \\ 1 & 2 & 5 & 9 & 11.8072 & 14.7641 & 15.33 & 13.1262 & 11.1163 & 23.2022 & 28.6040 & 27.9016 & 25.5634 & 20.5566 & 1.8830 & 14.4499 & 16.1674 & 14.5803 \\ 1 & 4 & 8 & 12.8072 & 13.7641 & 14.33 & 12.1262 & 10.1163 & 22.2022 & 29.6040 & 26.9016 & 26.5634 & 19.5566 & 2.8830 & 15.4499 & 17.1674 & 15.5803 & 3.5101 \\ 1 & 5 & 9.8072 & 16.7641 & 11.33 & 9.1262 & 7.1163 & 19.2022 & 26.6040 & 29.9016 & 23.5634 & 22.5566 & -0.117 & 18.4499 & 20.1674 & 18.5803 & 6.5101 & 11.4006 \\ 1 & 5.8072 & 12.7641 & 15.33 & 5.1262 & 3.1163 & 15.2022 & 22.6040 & 25.9016 & 27.5634 & 18.5566 & 3.8830 & 14.4499 & 24.1674 & 22.5803 & 10.5101 & 15.4006 & -20.2264 \end{bmatrix}$$

The estimation error in calculation of vector **a** is e_a , the estimation error e_b in vector **b** estimation and the error e_x , resulted in current state vector estimation are average-quadratic errors and are obtained using the following formulas:

$$e_a(k) = - \sqrt{\frac{\sum_{i=1}^n (a_i(k) - \hat{a}_i(k))^2}{\sum_{i=1}^n a_i(k)}}, \quad (27)$$

$$e_b(k) = - \sqrt{\frac{\sum_{i=1}^n (b_i(k) - \hat{b}_i(k))^2}{\sum_{i=1}^n b_i(k)}}, \quad (28)$$

$$e_x(k) = - \sqrt{\frac{\sum_{i=1}^n (x_i(k) - \hat{x}_i(k))^2}{\sum_{i=1}^n x_i(k)}}. \quad (29)$$

The values calculated are $e_a(k)=0$, $e_b(k)=0$ and $e_x(k)=0$, most probably due to the absence of disturbance in the output channel and as a result vector parameters of **a**, **b** and the state vectors are exact estimates.

Conclusion:

The modeled in Simulink adaptive observer algorithm is workable in offline mode, without noise applied to the output signal. For performance check of the observer under the presence of noise or disturbance influence rectangular matrices should be formed based on input-output data under the condition $N > 3n$.

References:

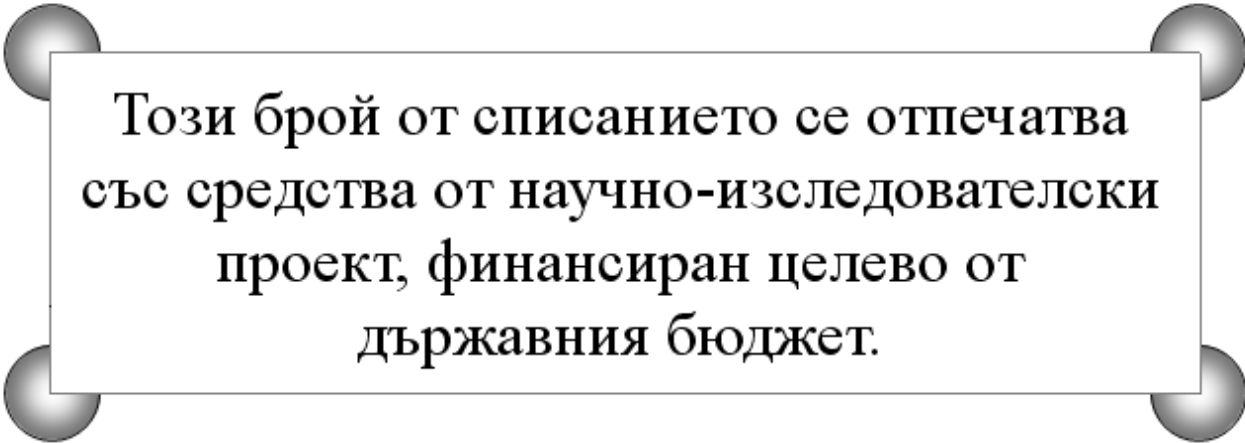
- [1]. Chapman S.J., Essentials of MATLAB™ Programming – 3rd edition, Cengage Learning, 2016.
- [2]. Darouach M, Existence and design of functional observers for linear systems, IEEE Trans. on Automatic Control, Vol. 45, p. 940-943, May. 2000.
- [3]. Embedded MATLAB™ User's Guide, The MathWorks, 2007.
- [4]. Lüders G., K.S. Narendra, An adaptive observer and identifier for a linear system, IEEE Trans. Autom. Contr., vol. AC-18, 496-499, 1973.
- [5]. Moreno J.A., Simultaneous observation of linear systems: a state-space interpretation, IEEE Trans. on Automatic Control, Vol. 50, p. 1047-1052, July 2005.
- [6]. Narendra K.S., P. Kudva, The discrete adaptive observer, Pros. IEEE Conf. Decision Contr. Phoenix. AZ., 301-306, 1974.
- [7]. Sotirov L., Theory of the automatic control discrete systems, TU-Varna, 2000.
- [8]. Tyukin I.Y., E. Steur, H. Nijmeijer, C. van Leeuwen, Adaptive observers and parameter estimation for a class of systems nonlinear in the parameters, Elsevier Automatica, 2013.
- [9]. Vuchkov I., Identification, Sofia, Yurapel PH, 1996.
- [10]. Zheng G., D. Boutat, Adaptive observer for simultaneous estimation of state and parameter for a class of nonlinear systems, IEEE 33rd Chinese Control Conference (CCC), 2014.
- [11]. Åström, K.J., R. Murray, Feedback systems. An Introduction for Scientists and Engineers, Karl Johan Åström and Richard M. Murray, 2007.

For contacts:

Veselin Lukov
Department of Automation
Technical University of Varna
E-mail: vesko_lk@abv.bg

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 използване 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 формират 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.