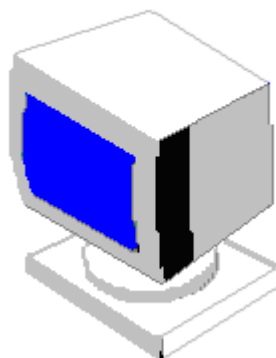


КОМПЮТЪРНИ НАУКИ И ТЕХНОЛОГИИ

година V

брой 2/2007

COMPUTER SCIENCE AND TECHNOLOGIES



**Bulgaria
Communications
Chapter**



Компютърни науки и технологии

Издание

на Факултета по изчислителна
техника и автоматизация
Технически университет - Варна

Редактор: доц. д-р О. Железов
Гл. редактор: доц. д-р П. Антонов

Редакционна колегия:

проф. дтн. Л. Сотиров
проф. дтн. С. Антошук
проф. дтн. К. Тенекеджиев
доц. д-р Н. Рускова
доц. д-р П. Петров
доц. д-р А. Антонов
доц. д-р Е. Маринов
доц. д-р В. Наумов

Печат: ТУ-Варна

За контакти:

Технически университет - Варна
ул. Студентска 1, 9010 Варна,
България
тел/факс: (052) 383 320
e-mail: peter.antonov@ieee.org
ognyanz@yahoo.com

ISSN 1312-3335

Computer Science AND Technologies

Publication

of Computing and Automation
Faculty
Technical University of Varna

Editor: Assoc. Prof. O. Zhelezov
Chief Editor: Assoc. Prof. P. Antonov

Advisory Board:

Prof. L. Sotirov, DSc
Prof. S. Antoshchuk, DSc
Prof. K. Tenekedzhiev, DSc
Assoc. Prof. N. Ruskova, PhD
Assoc. Prof. P. Petrov, PhD
Assoc. Prof. A. Antonov, PhD
Assoc. Prof. E. Marinov, PhD
Assoc. Prof. V. Naumov, PhD

Printing: ТУ-Varna

For contacts:

Technical University of Varna
1, Studentska Str., 9010 Varna,
Bulgaria
Tel/Fax: (+359) 52 383 320
e-mail: peter.antonov@ieee.org
ognyanz@yahoo.com

ISSN 1312-3335

СЪДЪРЖАНИЕ

CONTENTS

Информационни технологии**I Information Technologies**

- 1. Терминологични проблеми и комплексен подход към надеждността, безопасността и сигурността на компютърните и комуникационните системи и мрежи.**

Петър Ц. Антонов

3. *Peter T. Antonov*

- 2. Преглед на подходите и методите за проектиране на компютърни архитектури с ниска енергийна консумация**

Станислав Георгиев, Димитър Тянев, Веселин Йосифов

- 2. Review of Low Power Architectures Design Methodologies**

3 *Stanislav Georgiev, Dimitar Tyanev, Veselin Josiffov*

- 3. Метод за реализация на апаратни самоуправляващи се циклически структури - част II**

Димитър Тянев, Стамен Колев, Драгомир Янев

- 3. Method for Realization of Self-Controlling Loop Apparatus Structures □ Part II**

16 *Dimitar Tyanev, Stamen Kolev, Dragomir Yanev*

- 4. Предискретизация в реално време на цифров звуков сигнал чрез интерполация с ограничена честотна лента – част втора**

Станислав Б. Кирилов, Лъчезар Ил. Георгиев

- 4. Real-time digital audio signal resampling through bandlimited interpolation – part two**

31. *Stanislav B. Kirilov, Lâtchezar I. Georgiev*

- 5. Телевизионен координатор на автоматизирана система за управление на боен безпилотен летателен апарат**

Валентина Т. Цекова

- 5. Television Coordinator of Automated Control System of Unmanned Combat Air Vehicle**

30. *Valentina T. Tsekova*

- 6. Грешки в оценяването при построяване на едномерни немонотонни функции на полезност**

Наталия Д. Николова

- 6. Elicitation Errors In One-Dimensional Non-Monotonic Utility Functions**

42. *Natalia D. Nikolova*

СЪДЪРЖАНИЕ

CONTENTS

7 Проблеми при изграждане на електронен магазин		7. E-Store Building Problems	
<i>Димитър Й. Димитракиев, Иван С. Арменски</i>	48.	<i>Dimitar Y. Dimitrakiev, Ivan S. Armenski</i>	
9. Загуби при фотонно - кристални влакна и възможности за тяхното намаляване		9. Photonic Crystal Fibers Losses and Opportunities for their Decreasing	
<i>Йордан Урумов, Жейно Жейнов</i>	42.	<i>Jordan Urumov, Zhejno Zhejnov</i>	
10. Предискретизация в реално време на цифров звуков сигнал чрез интерполация с ограничена честотна лента – част първа		10. Real-Time Digital Audio Signal Resampling Through Bandlimited Interpolation – part one	
<i>Станислав Б. Кирилов, Лъчезар Ил. Георгиев</i>	48.	<i>Stanislav B. Kirilov, Lâtchezar I. Georgiev</i>	
11. Предискретизация в реално време на цифров звуков сигнал чрез интерполация с ограничена честотна лента – част втора		11. Real-Time Digital Audio Signal Resampling Through Bandlimited Interpolation – part two	
<i>Станислав Б. Кирилов, Лъчезар Ил. Георгиев</i>	54.	<i>Stanislav B. Kirilov, Lâtchezar I. Georgiev</i>	
12. Оценка параметрите повишаващи качеството на обслужване при wimax (QOS) технология		12. Examination of the Current Status and the Future Trends Associated with Distanse Learning in Technical Education	
<i>Илия Трендафилов Танчев</i>	58.	<i>Boyka Zh. Gradinarova, Mitko M. Mitev</i>	
13. Arctg-апроксимация на монотонно намаляващи едномерни функции на полезност		13. Arctg-Approximation Of Monotonically Decreasing One-Dimensional Utility Functions	
<i>Наталия Д. Николова</i>	67	<i>Natalia D. Nikolova</i>	
14. Изисквания за оформяне на статиите.	66	14. Guidelines for Preparing Manuscripts.	

ТЕРМИНОЛОГИЧНИ ПРОБЛЕМИ И КОМПЛЕКСЕН ПОДХОД КЪМ НАДЕЖДНОСТТА, БЕЗОПАСНОСТТА И СИГУРНОСТТА НА КОМПЮТЪРНИТЕ И КОМУНИКАЦИОННИТЕ СИСТЕМИ И МРЕЖИ

Петър Ц. Антонов

Резюме: Анализират се терминологични проблеми и концептуалната същност на основните термини в областта на компютърната и комуникационната надеждност, безопасност и сигурност. Обосновава се комплексен подход към тези проблеми и се предлага използването на комплексен показател *устойчивост (резистентност)*, включващ показателите за *обобщена надеждност, безопасност и сигурност*.

Terminological Problems and Complex Approach to the Dependability, Safety and Security of The Computer and Communications Systems and Networks

Peter Ts. Antonov

Abstract: This paper reviews terminological problems and conceptual core of the main terms in the area of the computer and communication dependability, safety and security. A complex approach to this problems and complex index (resistibility), including dependability, safety and security indexes are offered too.

Въведение

Както е известно, последните десетилетия се характеризират с интензивно развитие и приложение на компютърните и комуникационните системи и мрежи във всички сфери на живота. Едновременно с това, постоянно се повишават изискванията и се усъвършенстват методите и средствата за обеспечаване на необходимото ниво на показателите за компютърна и комуникационна *надеждност, безопасност и сигурност* [1,2,3 и др.]. Независимо от многото разработки и публикации обаче, в тази област все още съществуват нерешени таксономични и терминологични проблеми. Анализът на тези проблеми и комплексния подход към посочените важни показатели на компютърните и комуникационните системи и мрежи са предмет на настоящата работа, която се явява обобщение и по-нататъшно развитие на [4,5]. При това се предлага използването на нов термин – *обобщена надеждност*, обединяващ термините *надежд-*

ност, готовност и възстановимост (ремонтпригодност).

1. За термините *надеждност, безопасност и сигурност*

В [4,5] е направен подробен анализ на съдържанието и смисъла на използване на обсъжданите термини в разговорната реч и специализираната литература, като са анализирани редица тълковни речници на български, руски и английски езици, както и речници за превод между тези езици (разглеждането само на тези езици се обяснява с това, че от една страна това са основните езици, на които се ползва специализирана литература в България, а от друга страна, терминологията в областта на компютърните и комуникационните технологии на близките български и руски славянски езици се определя в значителна степен от англоезичната литература). Показано е, че в редица случаи, както в разговорната реч, така и в специализираната литература, термините *надеждност, безопасност и сигурност* се използват като

синоними и всеки един от тях често се пояснява с помощта на останалите. В българските тълковни речници [6,7 и др.] в общи линии са възприети следните определения:

надеждност - качество на надежден
надежден - който внушава доверие, на който или на когото може да се вярва, верен, сигурен
безопасност - липса на опасност, защитеност, сигурност
сигурност - качество на сигурен, увереност, положителност, отсъствие на опасност, безопасност
сигурен - който заслужава доверие, на когото може да се вярва

Подобна е ситуацията и в англо-руските тълковни речници [8,9 и др.], както и в руските такива [10 и др.]. Например:

reliability - the quality of being reliable
safety - the state of being safe
safe - secure, providing security or protection, reliable
security - freedom or protection from danger or worry, measures taken to guarantee the safety
secure - certain, guaranteed, safe, protected

Очевидно е, че отбелязаното по-горе различно тълкуване на думите и в трите естествени езика ще води до многозначност при превод на разглежданите термини от български език на английски, от български на руски, от английски език на руски език и обратно. Например [11,12,13]:

надеждност - trustworthiness, reliability, security
надежден - safe, reliable, dependable, trustworthy
безопасност - safety, security
сигурност - sureness, positiveness, certainty, certitude, security, safety
сигурен - secure, reliable, safe, dependable, positive, trustworthy
reliability - сигурност, надеждност, изправност, издръжливост
safety - безопасност, сигурност, запазеност, здравина
security - сигурност, безопасност, надеждност, увереност, защита, охрана, гаранция

Reliability се превежда от английски на руски език като надеждност, прочност, достоверност, *safety* – безопасност, сохранност, защита, охрана, *security* – безопасност, надеждност, увереност, охрана, защита [14].

Подобна, макар и в по-малка степен, синонимна обвързаност на разглежданите термини се наблюдава и в специализираната литература. За разлика от разговорната реч и художествената литература, където посочената синонимна обвързаност и взаимозаменяемост не само, че не са проблематични, но създават предпоставки за подобряване на качеството на изложението, то в специализираната литература това положение води до нееднозначност и трябва да бъде избягнато. За тази литература най-подходящото преводно съотношение на *надеждност*, *безопасност* и *сигурност* със съответните англоезични термини е следното [4,5]:

надеждност ↔ reliability
безопасност ↔ safety
сигурност ↔ security (1)

На български език понякога се използва и терминът *защита* (*protection*) и то най-вече като синоним на *сигурност* (*security*). В руската литература обаче, този термин (*защита*) се прилага много по-широко, при това и като синоним, както на термина *сигурност* (*security*), така и на *безопасност* (*safety*). Същото се отнася и за рускоезичния термин *безопасность*. Този факт се обяснява с липсата в езика на различни отделни думи за термините *сигурност* (*security*) и *безопасност* (*safety*), поради което те обикновено се превеждат и заместват на руски език с *безопасность* и по-рядко със *защита*.

Най-често използваният термин за *надеждност* в английската техническа литература все още е *reliability*, който според ISO се определя като способност на изделие или система да изпълнява функциите си за определен период от време, при съответни условия на експлоатация. Терминът *safety* се определя като способност за поддържане на поведение, което не води до нищо лошо, като в случая под "лошо" се разбират заплахи за човешкото здраве и права, за загуба на материални и духовни ценности, както и за нанасяне на екологични щети на околната среда. В този смисъл за най-подходящ съответстващ термин на български език, както бе вече отбелязано по-горе, може да се приеме термина *безопасност*. Понятиятната същност на термина *security* се изразява в способността за противодействие срещу несанкциониран (неразрешен, нелегален) достъп до данни и ресурси и в този смисъл най-подходящият български еквивалент на този термин е *сигурност*.

Следва да се отбележи обаче, че в някои специализирани области, различни от разглежданата, термините *безопасност* и *сигурност* се използват коректно, в представения по-горе смисъл. Като пример може да се посочи българската морска терминология в областта на корабоплаването, където проблемите на *безопасността* и *сигурността* придобиват през последните години все по-голяма значимост, поради увеличаващата се опасност

от екологични щети и зачестилите терористични нападения и отвлечения.

2. Комплексен подход към надеждността, безопасността и сигурността

Както е отбелязано в [4,5] *надеждността*, *безопасността* и *сигурността* могат да се разглеждат като комплексни характеристики на компютърните и комуникационните системи и мрежи, които се оценяват със съответни показатели, някои от които се отразяват едновременно на две или на всичките три характеристики. Това обуславя необходимостта от обхвато разглеждане и комплексен подход към тези проблеми. Трябва да се отбележи, че в литературата съществуват различни предложения за такова комплексно разглеждане, някои от които са противоречиви и взаимно-изключващи се. Например, в [1,4] като обобщен показател е предложен показателя за *комплексна сигурност* (*complex security*), обхващащ груповите показатели за *надеждност*, *безопасност* и *сигурност*. В [5] е направено развитие и за избягване на тавтологията вместо *комплексна сигурност* е предложено използването на нов термин - *устойчивост* (*издръжливост*). Този термин отговаря най-пълно на смисъла и съдържанието на обхвато (комплексен) показател, включващ груповите показатели за *надеждност*, *безопасност* и *сигурност*, които най-общо следва да се разглеждат като показатели за качество на системите и мрежите (в българските тълковни речници *устойчив* се пояснява като издръжлив на тежки условия, жизнеспособен, а *издръжлив* – като здрав, устойчив на природни условия и на трудности, което означава, че думите *устойчивост* и *издръжливост* могат да се разглеждат като синоними). Посочено е, че на английски език *устойчивост* е целесъобразно да се превежда като *resistance*, а на руски език – като *устойчивость*. На тази основа в българската специализирана литература е предложено да се използва и термина *резистентни системи*, по

отношение на компютърните и комуникационните системи и мрежи, които се явяват достатъчно надеждни, безопасни и сигурни [5].

В [15] *безопасността (safety)* и *сигурността (security)* се разглеждат като отделни надеждностни свойства на системите. В [16] *надеждността (reliability)* и *сигурността (security)* са представени като свойства на *безопасността (safety)*. В [7,18 и др.] се използва обобщения термин *dependability*, обхващащ *reliability (надеждност)*, *security (сигурност)* и *availability (готовност)*. В международния проект SAFERELNET (Safety and Reliability Net) по 5-та рамкова програма на Европейския съюз се разглеждат съвместно проблемите на *надеждността (reliability)* и *безопасността (safety)*. Такова разглеждане е направено и в [19]. В същото време в работния пакет WP7 (*Task 7.6. Dependability of Maintained Systems*) на цитирания по-горе проект SAFERELNET се отбелязва, че атрибути на *dependability* са: *reliability*, *availability*, *maintainability*, *safety* и *security*. В [20] *dependability* се представя като комплексен показател, отчитащ само *reliability*, *safety*, *security* и *availability*. В международната организация IFIP (International Federation for Information Processing) съществува работна група WG 10.4 on *Dependable Computing and Fault Tolerance*, която се занимава с проблемите на *dependability*, също включващи в себе си проблемите на *надеждността (reliability)*, *безопасността (safety)*, *сигурността (security)* и *готовността (availability)* на компютърните системи. В [21] основните задачи на *мрежовата сигурност (network security)* са *конфиденциалност (confidentiality)*, *цялостност (integrity)* и *готовност (availability)*. В материалите по 6-та рамкова програма на Европейската комисия проблемите на *dependability* и *security* се разграничават и разглеждат съвместно, като в същото време се включват в едно от основните направления за 2005–2006 г. в областта на приоритетните за тази програма “Технологии на информационното общество”

(Information Society Technologies) [22,23 и др.]. През 2004 г. авторитетната международна организация IEEE (Institute of Electrical and Electronics Engineers) започна издаването на научно списание IEEE Transactions on Dependable and Secure Computing, в което термините *dependable* и *secure* също се обособяват самостоятелно, но се разглеждат съвместно. В [24] *dependability* се представя като интегрален показател, включващ атрибутите: *availability (готовност)* – readiness for correct service, *reliability (надеждност)* – continue of correct service, *safety (безопасност)* – absence of catastrophic consequences of the user(s) and the environment, *integrity (цялостност)* – absence of improper system alterations и *maintainability (възстановимост, ремонтпригодност)* – ability to undergo modifications and repairs. В списанието Reliability Engineering & System Safety, както и в ESRA (European Safety and Reliability Association) проблемите на *reliability* и *safety* се разглеждат заедно, но се разграничават.

Тези и други подобни примери са доказателство за наличието на отбелязаните по-горе таксономични и терминологични проблеми и за отсъствието на общоприет комплексен подход към надеждността, безопасността и сигурността. На това основание в [5] е предложен и обоснован комплексния показател *устойчивост (resistance)*, включващ атрибутите *надеждност (reliability)*, *безопасност (safety)* и *сигурност (security)*. В подкрепа на това могат да се приведат множество примери, като провежданата през тази година вече 26-ма по ред международна конференция SAFECOMP 2007 - Computer Safety, Reliability and Security (тази конференция се провежда от 1979 г. по инициатива на EWICS TC7 - European Workshop on Industrial Computer Systems, Technical Committee 7 on Reliability, Safety, Security) и др. В случая, извеждането на англоезичния термин *reliability* като основен атрибут на предложениния комплексен показател *resistance (устойчивост)* се базира на множество източници

и факти, в това число и на съществуващата в структурата на IEEE организация IEEE Reliability Society, която обединява членове с професионални интереси в областта на надеждността на изделията и системите. Тази организация се занимава с проблемите на надеждността, качеството и ефективността на процеси, изделия, системи, хардуер и софтуер, провежда и спонсорира конференции и семинари, и издава научни списания като IEEE Transactions on Reliability и др.

Следва да се отбележи, обаче, че през последните няколко години в англо-езичната специализирана литература сериозно се наложиха термините *dependability* и *dependable*. Тези термини се налагат и в други страни, издания и конференции. Като примери могат да се посочат: провежданата в Полтава (Украйна) международна конференция Dependable Systems, Services & Technologies, организираната в Полша ежегодна международна конференция RELCOMEX (Reliability and Exploitation of Computer Systems) вече е преименована на DepCoS (Dependability of Computer Systems) и др.

Термините *dependability* и *dependable* най-често се превеждат на български език, съответно, като *надеждност* и *надежден, заслужаващ доверие*. Очевидно е, че при това се получава неприятната ситуация, когато два различни термина на английски език се превеждат по еднакъв начин на български език (същата ситуация е валидна и за руски език, където преводът е съответно, *надежность* и *надежный, заслуживающий доверия*). От приведените по-горе примери се вижда, че в английската специализирана литература терминът *dependability* е с по-широка, въпреки и не общоприета обхватност, в сравнение с термина *reliability*. При това положение, ако за *reliability* запазим наложилите се вече български термин *надеждност*, то за *dependability* следва да се използва друг преводен термин. Като се има в предвид по-широката обхватност на

dependability и тясната му връзка с *reliability*, то за такъв български термин би могъл да се използва термина **обобщена надеждност**. Ако това се възприеме, по-нататък следва да се уточни обхватността на *dependability*.

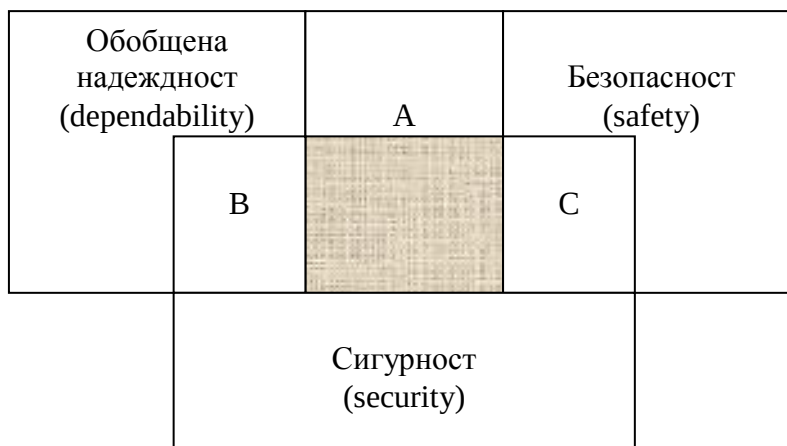
Независимо от съществуващите и отбелязани по-горе предложения, считаме, че много широката обхватност на *dependability* не е целесъобразна, тъй като проблемите на *безопасността* (*safety*) и на *сигурността* (*security*) си имат своя собствена специфика. Според нас е удачно *обобщената надеждност* (*dependability*) да включва само атрибутите *готовност* (*availability*), *надеждност* (*reliability*) и *възстановимост* (*maintainability*).

При това положение комплексният показател може също да бъде *устойчивост* (*resistance, resistibility*), който да обединява груповите показатели *обобщена надеждност* (*dependability*), *безопасност* (*safety*) и *сигурност* (*security*).

В този случай преводното съотношение (1) ще се представи като

обобщена надеждност ↔ dependability
безопасност ↔ safety
сигурност ↔ security
устойчивост ↔ resistibility

Това предложение за комплексен подход е представено образно на фиг. 1, където в рамките на *устойчивостта* са обособени частично-пресичащи се области на компонентните групови показатели *обобщена надеждност*, *безопасност* и *сигурност*. Областите А, В и С включват единични показатели, влияещи едновременно върху, съответно, *обобщената надеждност* и *безопасността*, *обобщената надеждност* и *сигурността* и *безопасността* и *сигурността*, а тъмната област съдържа такива единични показатели, които се отразяват едновременно на *обобщената надеждност*, *безопасността* и *сигурността* на компютърните и комуникационните системи и мрежи.



Фиг. 1. Обхват на *устойчивостта* (resistibility) на компютърните и комуникационните системи и мрежи

На базата на казаното по-горе комплексният показател *устойчивост* ще се определя от следните три групи показатели:

- показатели за обобщена надеждност;
- показатели за безопасност;
- показатели за сигурност.

Очевидно е, че нивото на *устойчивост* ще зависи от конкретните изисквания и ресурсни възможности на потребителите, както и от качеството на проектиране, изграждане и експлоатационно обслужване. Количествената оценка R на това ниво може да се определи от съотношението:

$$R = f(s_1, s_2, \dots, s_j, \dots), \quad (2)$$

където s_j в случая са количествените оценки на показателите за *устойчивост* - показатели за *обобщена надеждност*, *безопасност* и *сигурност* (под показател за *устойчивост* тук се разбира такава числова характеристика на системите, която е свързана с тяхната *устойчивост* чрез строго монотонна зависимост с положителен или отрицателен инградиент).

Такива системи могат да бъдат определени като *R-устойчиви* или *R-резистентни системи*. От своя страна, при това комплексно разглеждане на проблемите на обобщената надеждност, безо-

пасност и сигурност, съответните компютърни и комуникационни системи и мрежи е целесъобразно да се обозначават като *DSS-системи* (Dependable, safe, secure)-системи.

Съотношение (2) може да бъде представено и в следния вид:

$$R = F(K_{DEP}, K_{SAF}, K_{SEC}), \quad (3)$$

където $K_{DEP}, K_{SAF}, K_{SEC}$ - количествени оценки на груповите показатели, съответно, за *обобщена надеждност*, *безопасност* и *сигурност*.

На практика, поради очевидната разнотипност на показателите за *устойчивост*, извеждането на функционалните зависимости f и F в явен вид, на базата на математически модели с физическа интерпретация, е невъзможно. Това налага използването за оценка на R на някой от известните способности за претеглено усредняване (аритметичен, геометричен, хармоничен или квадратичен), които се явяват математически интерпретации на модела на справедливия компромис във векторните оптимизационни задачи.

В частност, при използване на най-простия за пресмятане аритметичен комплексен показател, съотношение (3) ще се запише по следния начин:

$$R = \frac{b_{DEP} K_{DEP} + b_{SAF} K_{SAF} + b_{SEC} K_{SEC}}{b_{DEP} + b_{SAF} + b_{SEC}}, \quad (4)$$

където $b_{DEP}, b_{SAF}, b_{SEC}$ - стойности на съответните тегловни коефициенти.

От своя страна груповият показател за обобщена надеждност K_{DEP} може да се определи аналогично, като

$$K_{DEP} = \frac{b_{AVA} K_{AVA} + b_{REL} K_{REL} + b_{MAI} K_{MAI}}{b_{AVA} + b_{REL} + b_{MAI}}, \quad (5)$$

където $K_{AVA}, K_{REL}, K_{MAI}$ - количествени оценки на груповите показатели за *готовност (availability)*, *надеждност (reliability)* и *възстановимост (maintainability)*, а $b_{AVA}, b_{REL}, b_{MAI}$ - тегловни коефициенти на тези показатели.

Готовността (availability) е свойство на системата да е работоспособна в произволен момент от времето, а *възстановимостта (ремонтпригодността)* – свойство за възстановяване на работоспособността чрез техническо обслужване и ремонт [2].

Показателите за безопасност зависят от конкретното изпълнение и приложение на компютърните и комуникационните системи и мрежи. Най-общо като основни аспекти на опасността за околната среда и човешкия живот могат да бъдат посочени [2]: природен (земетръси, наводнения, пожари и др.), социален (войни, тероризъм и пр.), медицински (болести, епидемии и др.), екологичен (замърсяване на въздуха, водата и пр.) и технически аспекти.

Показателите за сигурност по същество са показатели за нивото на защита срещу преднамерените (злоумишлени) въздействия в системите и мрежите. Обеспечаването на такава защита е сложна и широкообхватна задача, която се решава с помощта на комплекс от мерки с организационен и програмно-технически характер, които могат да бъдат обособени в следните основни групи [1]:

- организация на изграждането, управлението и контрола на системата за защита;
- физическа и техническа защита на съоръженията и ресурсите;
- защита на процесите на обмен на данни;
- автентикация на операторите в системите и мрежите;
- управление на достъпа до системните и мрежови ресурси

В заключение ще отбележим, че последните съотношения (4,5) могат да се използват, както за количествена комплексна оценка на *устойчивостта (резистентността)* на DSS-системи в процеса на проектиране и експлоатационно обслужване, така и за сравнителен анализ на различни компютърни и комуникационни системи и мрежи.

Литература

- [1].Антонов, П.Ц., С.В. Малчев. Криптография в компютърните комуникации. Варна, 2000, 315 с.
- [2].Христов, Х., В. Трифонов. Надеждност и сигурност на комуникациите. С., Нови знания, 2005, 255 с.
- [3].Семерджиев, Ц. Сигурност и защита на информацията, С., Класика и стил, 2007, 228 с.
- [4].Антонов, П.Ц. Терминологични проблеми и комплексен подход към сигурността на комуникационните системи. В: Компютърни системи и информационни технологии. Велико Търново, Фабер, 2004, с. 26-34.
- [5].Антонов, П.Ц. Комплексен подход към надеждността, безопасността и сигурността на компютърните и комуникационните системи и мрежи. //Компютърни науки и технологии, 2004, 1, с. 4-11.
- [6].Андрейчин, Л. и др. Български тълковен речник. С., Наука и изкуство, 1994, 1093 с.
- [7].Буров, С. и др. Съвременен тълковен речник на българския език с приложения (трето издание). В. Търново, Gaberoff - ЕООД, 2003, 1104 с.
- [8].LONGMAN Dictionary for Contemporary English. The Living Dictionary. Person Education Limited, 2005, 1949 p.

- [9].Reader's Digest Practical Dictionary. Build your Vocabulary, Improve your English. Reader's Digest Association Limited, 2007, 1088 p.
- [10].Лопатин, В.В., Л.Е. Лопатина. Русский толковый словарь. М., Изд-во "Русский язык", 1998, 832 с.
- [11].Чакалов, Г. Нов българско-английски речник: общ и политехнически. Т1 и Т2. С., ГО ООД, 1999, 1288 с.
- [12].Ранкова, М. и др. Английско-български речник. Том 1 и Том 2. С., Наука и изкуство, 1992.
- [13].Семерджиев, Г. и др. Английско-български политехнически речник. С., Техника, 2007, 1319 с.
- [14].Мюллер, В.К. Англо - русский словарь. М., Русский язык, 1988, 842 с.
- [15].Alexander, D. S. and others. Safety and Security of Programmable Network Infrastructures //IEEE Communications Magazine, October 1998, . pp. 84 – 86.
- [16].Guinier, D. Oriented-scenario dynamics in information systems safety //ACM Special Interest Group on Security, Audit and Control, 1994, Vol. 12, No. 3, pp. 6 – 11.
- [17].Arief, B., D. Besnard. Technical and Human Issues in Computer-Based Systems Security. – CS-TR 790, Department of Computer Science, University of Newcastle upon Tyne, 2003.
- [18].DIRC. Interdisciplinary Research Collaboration in Dependability [<http://www.dirc.org.uk/index.php>].
- [19].Александровская, Л., А. Афанасьев, А. Лисов. Современные методы обеспечения безотказности сложных технических систем. М., Логос, 2003, 206 с.
- [20].Avizienis, A. and others. Dependability. Basic concepts and terminology in English, French, German, Italian and Japanese. Springer-Verlag, 1992.
- [21].Cisco Systems. Fundamentals of Network Security. Cisco Networking Academy Program. Cisco Press, 2004, 870 p.
- [22].Towards a global dependability and security framework [<http://www.cordis.lu/ist/so/dependability-security/home.html>].
- [23].SIXTH FRAMEWORK PROGRAMME. Project no. 004547. SecureIST Advisory Board Recommendations for a Security and Dependability Research Framework. Issue 2.0, 15 January 2007, 50 p.
- [24].Avizienis, A. and others. Basic Concepts and Taxonomy of Dependable and Secure Computing //IEEE Transactions on Dependable and Secure Computing, Vol. 1, No 1, 2004, pp. 11-32.
- [25].Felici, M. Taxonomy of Evaluation and Dependability. School of Informatics, University of Edinburg [<http://www.informatik.unibonn.de/~gk/use/2003/papers/18500135.pdf>].

За контакти:

Доц. д-р инж. Петър Антонов
Катедра "Компютърни науки и технологии"
Технически университет - Варна
E-mail:peter.antonov@ieee.org

ПРЕГЛЕД НА ПОДХОДИТЕ И МЕТОДИТЕ ЗА ПРОЕКТИРАНЕ НА КОМПЮТЪРНИ АРХИТЕКТУРИ С НИСКА ЕНЕРГИЙНА КОНСУМАЦИЯ

Станислав Георгиев, Димитър Тянев, Веселин Йосифов

Резюме: Представено е извършеното проучване и направения анализ на методите, с помощта на които е възможно да се постигне намаляване на енергийната консумация в ново проектирани компютърни архитектури. Изявени и класифицирани са основните принципи, върху които се основават тези методи. Разгледана е тяхната същност и вариантите им, прилагани на различните архитектурни нива и степен на детайлизация на схемите. Посочени са стойностите на оценките, с които може да бъде намалена енергийната консумация, както за отделните методи, така и върху отделните архитектурни нива. Изказани са изводите от тяхното сравнение и са посочени софтуерните инструменти, с които може да се преследва енергийната оптимизация при схемно проектиране върху програмируеми интегрални схеми от вида CPLD и FPGA.

Review of Low Power Architectures Design Methodologies

Stanislav Georgiev, Dimitar Tyanev, Veselin Josifov

Abstract: This document presents research and analysis of methods through which it is possible to decrease (to reduce) the power consumption of newly designed computer architectures. The basic principles on which these methods are based are presented and classified. The methods are reviewed in detail, as well as their variations applicable at different architecture abstraction levels. The power consumption estimations are presented for the particular methods as well as the different architectural levels. Conclusions are made from the comparison of the different methods and automated tools through which it is possible to achieve energy consumption optimization when designing programmable chips such as FPGA or CPLD are listed.

1. Увод

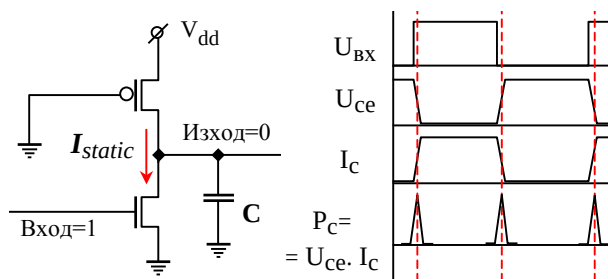
С напредъка в областта на минитюаризацията, числото на интегрираните в една интегрална схема транзистори непрестанно се увеличава. Това от своя страна води до намаляване размерите на електронните устройства, което от своя страна води до бума на преносимите електронни устройства – от телефони до компютри, през последното десетилетие. Едно от основните изисквания към преносимите устройства е ниското им тегло, което от своя страна поставя на преден план един въпрос, който преди това е бил в една или друга степен пренебрегван – това е въпросът с консумираната от устройствата енергия. Макар и да имат възможност за свързване към електрозахранващата мрежа, основният енергиен източник на мобилните устройства са

батериите. Изискването за ниска маса обаче означава, че не е разумно те да бъдат комплектовани с голяма батерия. Въпреки развитието на батерийната технология, сама по себе си тя не може да реши проблемът с консумацията. За това актуални за разработка стават така наречените компютърни архитектури с ниска енергийна консумация (*Low Power Architectures*).

2. Изложение

Преди да бъдат изложени подобренията, отнасящи се към понятието архитектура с ниска енергийна консумация, е редно първо да се обърне внимание на самия механизъм на енергийна консумация в CMOS интегралните схеми. Като цяло, енергийната консумация може да се раздели на две части – статична и дина-

мична. Статичната консумация е с по-малко влияние върху общата консумация и се определя от явления като ток на утечката. Динамичната консумация от своя страна се определя от “превключването” на транзисторите в интегралната схема. За по-добро обяснение е използвана схема на CMOS инвертор (фигура 1).



Фиг. 1 CMOS инвертор

Динамичната консумация има няколко компонента, но с най-голямо значение са преходните процеси при зареждане и разреждане на паразитните капацитети. На фигура 1 паразитният капацитет е изобразен чрез кондензаторът C . Представени са процесите в пълен период на входния сигнал от високо ниво до ниско ниво (нула) и отново до високо ниво. При превключване на входа от високо към ниско ниво, NMOS транзисторът се запуща, а PMOS транзисторът се отпушва. Кондензаторът C се зарежда до ниво U_{ce} , като при този процес се черпи енергия равна на $C \cdot U_{ce}^2$ [1, 2]. Половината от тази енергия се разсейва чрез PMOS транзистора а другата половина се натрупва в кондензатора. При промяна на входния сигнал от ниско към високо ниво, кондензаторът се разрежда през отпушеният NMOS транзистор. Така динамичната съставна на консумираната енергия може да се изрази със следната формула:

$$P_{dyn} = \left(\frac{1}{2} C U_{ce}^2 \right) \cdot \alpha \cdot f, \quad (1)$$

където f е означена тактовата честота а α е отбелязана активността на превключване на транзистор от крайния възел. Причината за наличието на тези две величини е следната. С f е отбелязана

честотата, с която постъпват входните за схемата данни. В синхронни схеми се приема, че тази честота е равна на тактовата честота. С α е отбелязана така наречената активност на данните [1]. Тя изразява очаквания брой превключвания на CMOS елементите, предизвикани от ново постъпилите данни в рамките на един период.

При разглеждане на възможните подходи за оптимизация на енергийната консумация е въведена представената подолу класификация, в която всеки метод е “прикачен” към дадено ниво на абстракция – например гейтово ниво, схемно ниво, архитектурно ниво. Но независимо за кое ниво на подобрения се говори, в основата си те следват три основни идеи, които са:

1. “Търгуване” на производителност срещу площ. Един от основните подходи за намаляване на енергийната консумация е мащабирането на захранващото или праго-вото напрежение в схемите. Като цяло обаче това от своя страна води до намаляване на производителността. Друг подход е въвеждането на суперскаларност в структурите, като по този начин дизайнът може да има същата прозводителност, но при по-ниско захранващо напрежение. В резултат на това обаче се увеличава заеманата от схемата площ в чипа.
2. Избягване на излишни разходи. Типичен пример за излишен разход е подаването на тактов сигнал към структурни модули, които не са ангажирани с получаването на резултат от текущата машинна команда. Основните подходи за избягване на този излишък са няколко. На първо място това е техниката *Clock Gating*, същността на която се изразява в насочване на тактовите последователности само към текущо работещите модули. На второ място е подходът към реализация на регулярни алгоритми, а на трето място е използването на специализиран (посветен) хардуер, вместо програмируем такъв.

3. Максимално прилагане на принципа на локалността. Обръщенията извън чипа по време на изчислителния процес са изключително скъпи от енергийна гледна точка. Това се дължи на големите паразитни капацитети на свързващите чипа шини с външните компоненти. Зле разпределен проект, в който данните са “разхвърляни” на много места, е също силно енергоемка ситуация. Целта на принципа на локалността е да се минимизират глобалните обръщения. Този подход е особено популярен при DSP архитектурите.

По-надолу ще бъдат представени изявените подходи в отделните абстрактни нива.

А) Технологично ниво

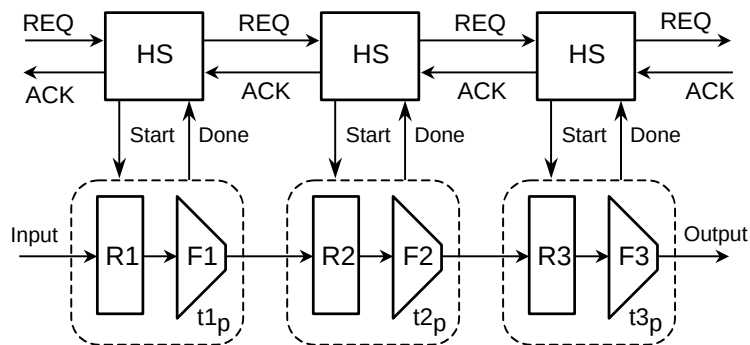
Най-ниското ниво в абстрактното разглеждане на архитектурата е технологичното. Тук става въпрос за технологията на изработка на чипа и на корпуса. Видът на корпуса до голяма степен влияе на консумираната енергия, тъй като генерира определена стойност на капацитета. Разликата в капацитетите вътре в чипа спрямо този в изводите е от три порядъка (от десетки фетмофаради вътре в чипа до 13-14 пикофарада в извод), както е посочено в [6]. Намалявайки този капацитет може да се постигне драстично намаляване на консумираната енергия. Пример за това е технологията MCM (*multi-chip modules*), използвана от фирмата от *Rockwell International* в процесори, предназначени за употреба в самолети. Изследвания [7] са установили, че чрез MCM технологията капацитетите на изводите на чипа може да бъде намален до порядъка на капацитетите вътре в чипа. Минимизирането на капацитета се обеснява с мащабиране на технологията. Например, при преминаване от 45 [nm] към 32 [nm] технология за съвремените процесори, на което сме свидетели, последва намаляване на енергийната консумация. Като пример за това може да се цитира изявлението на *MIPS Technologies* на международната конференция за *Solid State Circuits* през 1994 година, че с преминаване на 0,64 [μm]

технология от 0,8 [μm], се е намалила енергийната консумация на техния 64 битов RISC процесор с 25%. Но само по себе си мащабирането на технологията не може да реши проблема с консумацията. Съпротивлението на връзките в чипа зависи правопропорционално от дължината им и обратно пропорционално от ширината и дебелината им. Мащабирането на технологията означава, че и трите показателя ще намалеят с един и същи коефициент, което от своя страна означава че съпротивлението на връзките ще нарастне със стойност равна на коефициента, с който намаляват физическите им размери. Влиянието на връзките в чипа върху производителността и енергийната консумация е разгледано подробно в [26]. С продължителното мащабиране на технологията, влиянието на връзките в чипа върху консумираната енергия ще нараства, до момент в който понататъшното мащабиране няма да има влияние върху консумираната енергия или дори ще я увеличи.

Б) Схемно ниво

Следващото ниво е схемното. Тук трябва да се вземе предвид, че освен енергийна консумация, други фактори, на които трябва да се обърне внимание, са сложността и надеждността на дизайна. На това ниво вниманието е насочено към употребата на асинхронна логика, към оразмеряването на транзисторите и към стила на проектиране.

Фигура 2 илюстрира представата за асинхронна логика. В асинхронната логика липсва тактовият сигнал. Синхронизация се постига чрез така наречения *handshake* (HS, ръкостискане), което се осигурява от допълнителен хардуер. Поради тази причина, асинхронната логика често е пренебрегвана като опция, но имплементацията ѝ е полезна от гледна точка на енергийната консумация. Предимствата на асинхронната логика могат да се изразят с идеята за избягване на излишните разходи. Липсата на тактов сигнал означава че я няма и често сложната мрежа за неговото разпространя



Фиг. 2 Представа за асинхронна логика

ване. Също така използването на сигнали за начало и завършване на операцията (сигналите *start* и *done* на схемата) спомагат за избягването на гlichовете. На последно място фактът, че процесите започват при наличието на данни, означава, че асинхронната логика има вграден механизъм за “изключване” на неактивни модули. Но освен предимства, асинхронната логика си има и недостатъци. Допълнителният хардуер, който осъществява ръкостискането също така консумира енергия, и в зависимост от размера на логиката, която изгражда този хардуер, могат да бъдат загубени предимствата както и в размера на проекта така и като консумация. Логиката за ръкостискане няма пряко отношение към изчислителния процес, така че донякъде тя може да се сравни с логиката по разпространение на тактовия сигнал, въпреки че консумацията ѝ е по-малка поради по-ниската активност на сигналите за ръкостискане. Друг недостатък, който вече не е такава тежест, която е имал преди няколко години е, че по-малък брой развойни среди поддържат проектирането на асинхронна логика. На принципите на асинхронната логика се базира техниката *clock gating*. Пример за приложението на тази техника е даден в [23].

Независимо от метода за управление (синхронен или асинхронен), размерът на транзисторите е фактор при оптимизиране с цел постигане на ниско енергийна консумация. Тук проблемът се състои в намиране на баланс между производителност и цена, като под цена в случая се има предвид площта в чипа с ниска енергийна

консумация. Транзистори с по-големи размери могат да провеждат по-големи токове, което спомага за по-голяма производителност. Но също така транзистори с по-големи размери превключват по-голям ток на късо съединение. Те също така внасят и по-големи капацитивности в схемата, което от своя страна води до повишена консумация. В този случай е необходимо да се намери баланс между двете изисквания. Една добра стратегия е използването на транзистори с минимални възможни размери, като в най-важните части на схемата, така наречените *critical paths*, се използват транзистори с по-големи размери за постигане на необходимата производителност. Изследвания [8] показват, че при реализиране на суматори по тази стратегия, се наблюдава подобрене в консумация с до 36%.

Също така влияние върху консумацията има и стилът на проектиране, т.е. дали в проекта ще бъдат използвани специално проектирани логически клетки, стандартни логически клетки или масиви от гейтове. Най-добри от гледна точка на енергийната консумация са специално проектираните клетки, понеже при тях принципите на ниската енергийна консумация могат да бъдат приложени най-пълно и за всеки отделен елемент. Специално проектираните логически клетки съдържат само необходимите за конкретната задача логически елементи и по този начин не съдържат излишни такива. Излишни логически елементи обикновено остават, когато реализацията използва стандартни клетки. Голямата цена на

такива специални устройства, както и дълготното време, нужно за тяхното проектиране, правят този стил на проектиране скъп и приложим в редки случаи.

Стандартните клетки от друга страна, предлагат ниско време за проектиране и цена на проекта, но за тях може да се каже, че са анти-тезата на ниската енергийна консумация. Повечето стандартни клетки са оптимизирани за производителност, и често са с излишно големи размери, което води до увеличена енергийна консумация.

Масивите от гейтове се явяват като компромис между посочените по-горе стилове на проектиране. Макар и да нямат гъвкавостта на специализираните клетки, те предлагат възможност за оразмеряване на транзисторите и връзките между тях.

В) Гейтово ниво

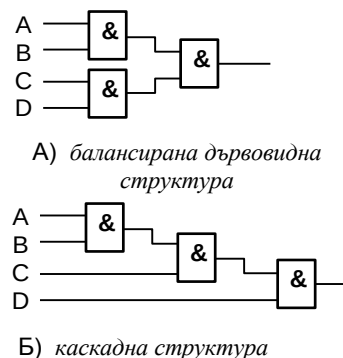
Техниките, приложими на следващото, гейтовото ниво, са базирани на трите основни принципа, които са изведени по-горе. На това ниво техниките, предлагащи оптимизация от гледна точка на енергийната консумация се изразяват в манипулиран мапинг на технологията, стремеж към намаляване на нивата на активност и борба с гlichовите.

Под мапинг се разбира процесът на преобразуване на двоичен файл в реални варианти за имплементацията на проекта в чип, като за всяка схема има няколко варианта за реализация. Например, три входа И-НЕ елемент може да бъде реализиран като един сложен CMOS гейт или като два каскадно свързани двувходови гейта. Всеки вариант предлага различни нива на активност и внася различни капацитети. Като цяло, сложните гейтове внасят по ниски паразитни капацитети, понеже повечето сигнали са “скрити” във вътрешните възли на гейта. Мачинг, целящ подобряване на енергината консумация, е мапинг, при който активността на превключване е намалена до минимум, а елементи с високо ниво на активност са “скрити” в сложни CMOS гейтове. По този начин сигнали с високо ниво на активност са мапирани във вътрешни възли с ниски

капацитивни товари, което намалява енергийната консумация. Ако обаче бъдат проектирани прекалено усложнени гейтове, същите ще имат прекалено голямо закъснение, което ще бъде заплатено с намалена производителност на схемата като цяло, т.е. за подобрената енергийна консумация е платено със загубената производителност. Разработени са няколко алгоритми за технологичен мапинг, които според [9] и [10] намаляват консумацията с от 12% до 21%.

Техниката за намаляване на нивото на активност се изразява в подходящо преподреждане на входовете. В този случай е желателно сигнали с висока активност да бъдат изместени назад в реда на постъпване на входовете, като по този начин по-малък брой гейтове ще бъдат засегнати (манипулирани) от тях.

Техниката за намаляване на гlichовите е свързана с намаляване на нежеланите (непроизводителните или още междинните) превключвания. Примерните схеми от фигура 3 илюстрират тази техника. Представени са два варианта на реализиране на една и съща логическа функция, като едната реализация представлява балансирана дървовидна структура, а втората – каскадна.



Фиг. 3 Намаляване на гlichовите

Ако се приеме, че входните променливи постъпват и за двете схеми по едно и също време и че закъсненията в елементите са еднакви, ще се установи, че при каскадната структура има по-голям брой последователни превключвания. На практика постъпване на входен сигнал на вход на който и да е от елементите,

предизвика превключване на изходите на всички елементи след него в каскадата. Докато в дървовидната структура всеки елемент се превключва само по веднъж при промяна на входовете му, поради балансираната му структура. Друг вариант за намаляване на глитчовете е като се намали сложността (дълбочината) на логиката, понеже броят на глитчовете е квадратично зависим от сложността на логиката [1]. Изследвания [11], показват, че чрез премахване на глитчовете са възможни подобрения в консумацията на схемата с до 20%.

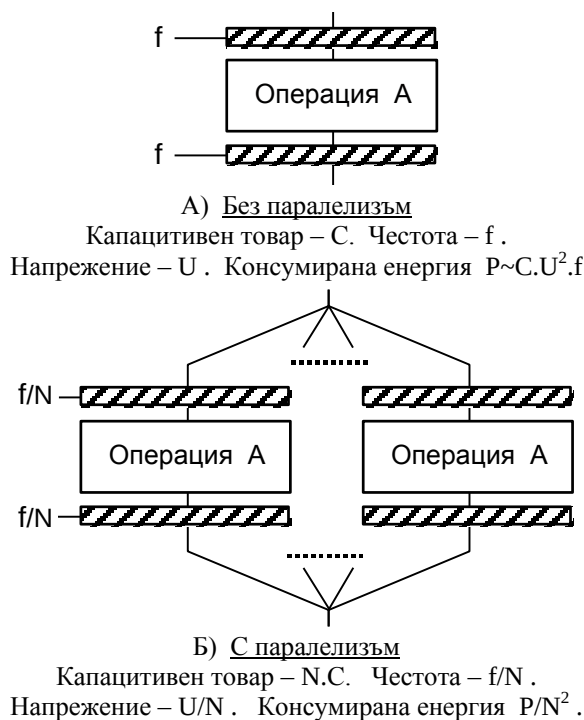
Г) Архитектурно и системно ниво

Разгледаните дотук нива се приемат като ниски. Архитектурно, системно и алгоритмово нива се приемат за високи. На тях ефектът от оптимизациите е значително по-висок. Докато оптимизациите на ниско ниво водят до двукратно подобрение, то при оптимизации на високо ниво подобрението може да е с няколко порядъка. Следващото ниво, което ще разгледаме, е архитектурно-системното ниво, като под архитектурно се имат предвид оптимизациите на RTL (*Register Transfer Level*) ниво на абстракция. Както и на ниските нива, техниките тук също се базират на трите основни принципа, дефинирани в началото на този документ.

На архитектурно ниво възможните техники са следните – паралелно изпълнение на изчислителните процеси, програмируемост, партишънинг, избор на формат на представяне на данните и адаптивни стратегии за захранване.

Паралелното изпълнение на изчислителните процеси (*concurrent processing*), е може би най-важната съвременна стратегия за намаляване на консумацията. Това е директна размяна на производителност за ниска консумация. Прилагат се добре известни техники като суперскаларност и конвейерна организация на изчислителния процес и придобитата по този начин производителност се “заменя” за ниска консумация чрез намаляне на захранващото напрежение.

Фигура 4 представя два изчислителни процеса: рисунка 4а илюстрира операционна структура без приложена суперскаларност, а рисунка 4б – с приложена суперскаларност за операция А.

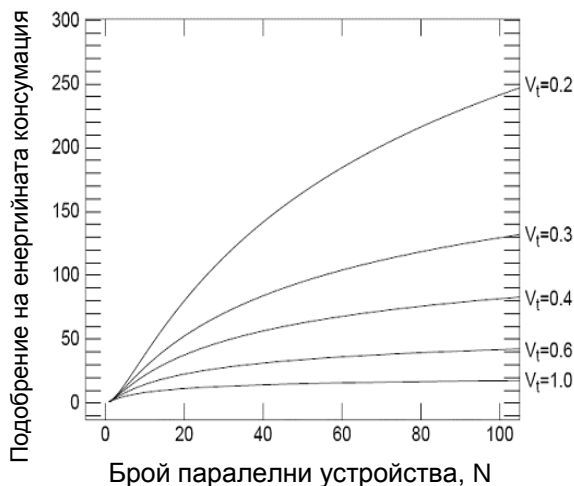


Фиг. 4 Суперскаларност

В първата рисунка има един регистър, който е тактуван с честота f . В схемата 4б хардуерът от схема 4а е дублициран N пъти. Поради това са налични N на брой процесори, като всеки от тях може да бъде тактуван с честота равна на f/N и производителността на схемата със суперскаларност ще бъде същата както тази на схемата без паралелизъм, при положение, че няма зависимости, които да пречат на паралелното изпълнение на операция А. Привлекателността на суперскаларната архитектура, от гледна точка на енергийната консумация, се състои в това, че всеки процесор е захранван с напрежение приблизително равно на V/N . Следва да се отчете, че поради наличието на N на брой процесори, капацитивният товар на схемата с паралелизъм е равен на произведението $C.N$. Ако се приложи зависимостта (1), то консумираната енергия ще се изрази както следва:

$$P_{par} = \frac{P}{N^2}, \quad (2)$$

където с P е означена енергийната консумация на схемата без супер скаларност. Формулата е валидна за идеалния случай. В този случай не се взема предвид, че алгоритмите рядко позволяват голяма степен на паралелизъм. Също така не се взема предвид и влиянието на праговото напрежение, но за разлика от влиянието на алгоритъма, то е положително. Влиянието на праговото напрежение е илюстрирано на фигура 5.



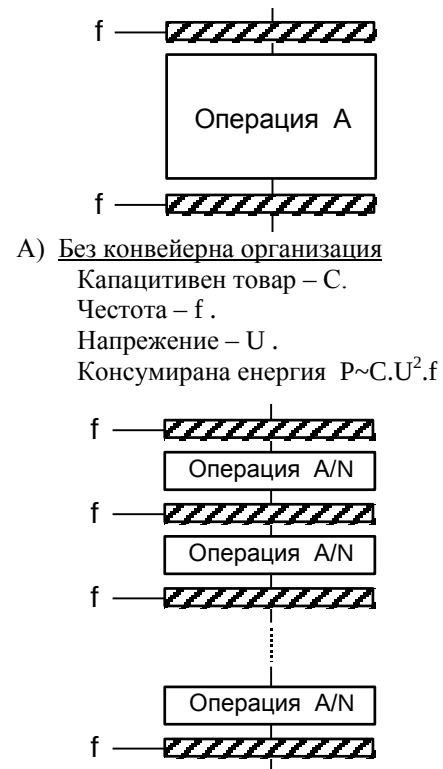
Фиг. 5 Влияние на праговото напрежение

Недостатъците на суперскаларността са следните. На първо място N -кратното дублиране на процесорната структура води до огромно нарастване на заеманата върху кристала площ. Второ, нараства допълнителният хардуер и връзките за реализиране на разпределението на сигналите по процесите, както и тези за събирането на резултатите. Това допринася за увеличаване на консумацията и е фактор за ограничаване максималния брой на процесорите.

Друг начин за реализация на паралелизъм е съвместеното изпълнение чрез реализиране на конвейерна организация. На фигура 6 е показано сравнение на схема със и без конвейерна организация.

Тук за разлика от суперскаларната архитектура няма дублиране на хардуера, а наместо това операцията се разбива на отделни части като за всяка част се посвещава част от хардуера. Работната

честота остава непроменена, както и капацитивният товар. Предимството в този случай идва от намалената работа по необходимите изчисления във всяка отделна степен на конвейера. Вместо цялата операция, за един цикъл се извършва $1/N$ -та част от нея. Това от своя страна позволява намаляне на захранващото напрежение до V/N . Прилагайки (1) излиза че консумираната енергия в този случай е същата както при прилагане на суперскаларност. Както при суперскаларността, конвейерната организация се характеризира с увеличена площ, понеже за всяка степен на конвейера се заделя регистър. Това от своя страна добавя капацитивен товар към мрежата за разпространение на тактовия сигнал. С увеличаване на дълбочината на конвейера този капацитивен товар нараства до степен, в която по нататъшно разширение на конвейера става безмислено. Но като цяло “преразходите” на конвейерната организация са по-малки от тези на суперскаларността, което го прави предпочитана алтернатива за оптимизации с цел намаляване на енергийната консумация.



Б) Конвейерна организация

Капацитивен товар – C .

Честота – f .

Напрежение – U/N .

Консумирана енергия $P \sim P/N^2$.

Фиг. 6 Конвейерна организация

Конвейерната организация има същите недостатъци както и суперскаларната архитектура – не всички алгоритми могат да бъдат оптимизирани за конвейерна организация на изчислителния процес. Освен това си има и уникални за себе си недостатъци с голямо значение за дълбоки конвейери. Такъв недостатък представлява извличането и декодирането на машинна команда $N(i)$, което става преди да бъдат завършени изчисленията заповядани от команда $N(i-1)$. Съществува вероятност изпълнението на i -тата команда да зависи от резултата на предишната $(i-1)$ -ва команда. Конвейерните зависимости и методите за тяхното разрешаване са подробно изяснени в [14]. Пример за приложението на този метод е даден в [23].

Целта на адаптивните стратегии за захранването е елиминирането на излишните разходи. Съществуват два основни варианта: селективно деактивиране и режим на сън. Селективно деактивиране е известно още под името *clock gating*. Селективното деактивиране се състои в спиране на тактовия импулс към модули, които нямат отношение към текущото изчисление. Стратегията изисква допълнителна логика, която да следи активността на отделните модули в системата. За целта обаче тя трябва да се прилага с малки изменения при динамична логика, където тактовият сигнал се използва за опресняване на информацията. В този случай е препоръчително вместо да се спира тактуването към този модул, същото да се забавя. Така информацията в динамичните възли ще бъде опреснявана периодично, но с по-ниска честота. Пример за селективно деактивиране е разгледан подробно в [28].

Режимът на сън представлява разширение на стратегията за селективно деактивиране. Разликата е в това, че вместо наблюдение на всеки отделен модул, активността се наблюдава в цялата система.

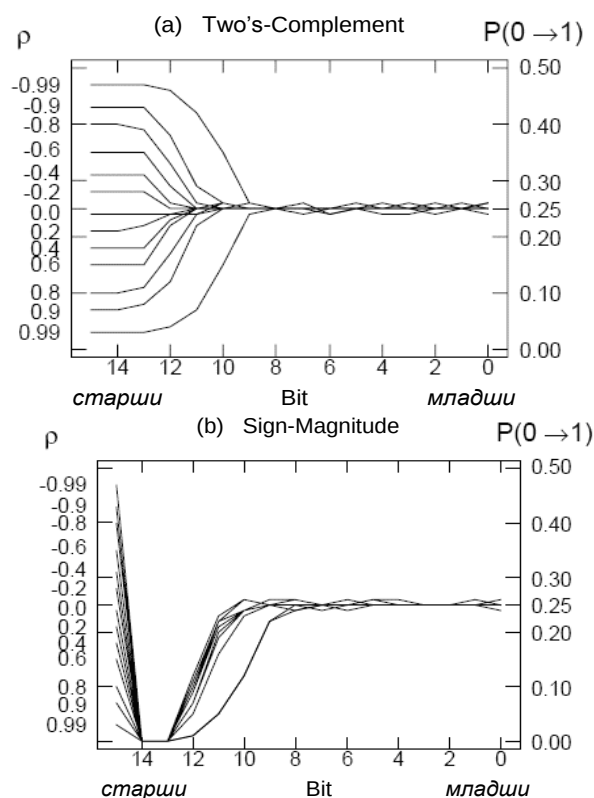
Ако системата стои “без работа” за определен срок от време, то тя или се изключва или навлиза в режим на сън. В режим на сън се наблюдават входовете на системата за необходимо активизиране, което ще “събуди” системата и ще я въведе в работен режим. Трябва да се отбележи, че превключването на системата от работен в спящ режим е свързано с преразход на енергия, така че е препоръчително да се установи точно продължителността на периода, в които системата няма да работи, за да премине в спящ режим. Тези техники са особено популярни при преносимите компютри, където намират широко приложение, и според данни на IBM и Intel спомагат за намаляване на консумацията с до 30%.

Така нареченият партишънинг представлява разпределение на ресурсите на изчислителната система. Това се прави с цел увеличаване на локалността и намаляване на глобалните обръщения. Същността на идеята се състои в това, че вместо един процесор, изпълняващ целия алгоритъм, изчислителната система може да се разпредели така, че всяка нейна част да се специализира за изпълнението на определена част от алгоритъма. Паметта също подлежи на подобно разпределение. От енергийна гледната точка е много по-ефективно към всеки процесор да има малка локална памет отколкото голяма по размер обща памет. Това разпределение се отнася и за управляващите автомати [5]. Вместо един краен автомат, който да генерира всички управляващи сигнали и да ги разпространява, при разпределен краен автомат само малка част от сигналите трябва да се генерират и разпределят глобално, в смисъл извън границите на чипа, което се отразява силно положително на енергийната консумация. Като част от темата за разпределен и централизиран изчислител процес може да се погледне и върху въпроса за програмируем и специализиран (наричан още посветен) хардуер. Специализираният хардуер е по-изгоден от гледна точка на енергийната консумация, понеже се избягват изшлините разходи на

програмируемия хардуер, който трябва да е способен да изпълнява много задачи, а не само една. От своя страна специализираният хардуер заема повече площ върху чипа, а също така той не притежава важното удобство на програмируемия хардуер – лесната промяна на дизайна като цяло. При програмируемият хардуер това изисква само промяна в микрокомандите, издавани от софтуера или фирмуера, докато при специализираният хардуер се налага на практика разработка на нова схема.

Последната стратегия на това ниво е изборът на формат за представяне на данните. Тук дизайнерът има голям избор – фиксирана запетая или плаваща, кодирано представяне или не, *sign-magnitude* или *two's-complement* и така нататък. Изборът на формат за представяне на данните е свързан с компромис между точност на представяне, сложност на дизайна, производителност и енергийна консумация. Ако започнем анализа с първия избор – фиксирана или плаваща запетая, то ще видим че фиксираната запетая има най-малки изисквания към хардуера и съответно с най-ниска консумация, но страда от ограничения на обхвата. При плаваща запетая тези ограничения не важат, но това е на цената на допълнителен хардуер, което се транслира в допълнителна площ и консумация. Също така трябва да се вземе предвид и дължината на разрядната мрежа. Често дизайнерите се презастраховат като използват по-голяма дължина от необходимата, като за тази сигурност се заплаща с производителност и консумация. За това е препоръчително този метод да се избягва и да се прави точен анализ на необходимата, а не на желаната точност на представяне за проекти с ниска консумация. Освен точността на представяне и дължината на думата, дизайнерът трябва да има в предвид и аритметичното представяне на данните. Например, машинните кодове прав, обратен (*sign-magnitude*) и допълнителен (*two's-complement*), както и техните модифицирани варианти. Допълнителният код е най-подходящ за аритметични изчисления, затова и се ползва най-

често. В допълнителен код старшите значещи цифри в изображението на числата съвпадат със знаковата цифра [14]. В резултат, в старшата част на разрядната мрежа тези битове могат да се приемат за излишна информация. Освен това те са източник на възможни и излишни превключвания, а от там и на излишна консумация при изчисления със смяна на знака. Правият код (*Sign-magnitude*) използва един бит за означаване на знака, така че от енергийна гледна точка тази форма на презентация на данните се счита за по-ефективна [13]. За съжаление, за аритметични изчисления в този формат се налагат допълнителни преобразувания, които елиминират ефекта на спестената консумация от по-малкия брой превключвания. Вероятността за активни превключвания по причина на възникнали преноси по дължината на разрядната мрежа е представена на фигура 7 [13].



Фиг. 7 Активност на превключване при *sign-magnitude* и *two's-complement*
Д) Алгоритмово ниво

Последното и най-високо ниво е алгоритмовото ниво. Алгоритмите имат както директно така и индиректно влияние върху енергийната консумация. Например

сложността и броят операции в един алгоритъм имат директно влияние, докато степента на паралелизъм, която определя колко ефективно алгоритъмът може да се приложи върху архитектура, оптимизирана за ниска енергийна консумация, има индиректно влияние. На това ниво се разглеждат качествата на алгоритмите и какви промени в тях могат да се направят за да бъдат приложими за архитектури с ниска енергийна консумация. Това разглеждане започва с качествата, които имат директно влияние върху енергийната консумация, като например сложността на алгоритъма. Сложността може да бъде измерена по няколко начина, като един прост начин за това е да приемем като оценка броя на командите. Това е полезна метрика понеже за изпълнението на всяка команда се консумира определено количество енергия. Съответно по-малък брой команди означава по-ниска консумация. Алгоритмите с голяма производителност често са с голям брой команди. Не само броят команди има влияние върху консумацията. Някои команди имат по-голяма консумация от други. Например, изпълнението на командите за умножение имат по-голяма консумация от изпълнението на тези за събиране. Също така изпълнението на команда за обръщение към паметта има голяма цена като енергийна консумация. Поради тази причина, броят на тези команди в алгоритъма, предназначен за работа върху ниска енергийна архитектура, трябва да е минимизиран. Докато сложността на алгоритъма влияе на консумацията в комуникационната среда, то регулярността влияе на управляващия автомат. Силно регулярен алгоритъм се нуждае от по-малко състояния на крайния автомат за да бъде той описан. Това означава по-просто устроен краен автомат. За програмируем процесор, регулярният алгоритъм означава по-малко команди за преход, а в случай на цикли – по-голям шанс за правилно предсказване на изхода от цикъла. Това води до намалена консумация. Влиянието на дължината на разрядната мрежа бе разгледано на архи-

тектурното ниво на абстракция. При определянето на дължината ѝ трябва да се вземе предвид точността на алгоритъма. Дори алгоритми, които изпълняват една и съща функция, могат да има различна точност, което води до различна дължина на разрядната мрежа на процесора.

Други характеристики на алгоритъма имат индиректно влияние върху консумираната енергия. Такива характеристика са степента на паралелизъм и модуларността. Паралелното изпълнение на изчислителните процеси позволява да се намали консумацията, като за това се заплаща с допълнителна производителност, евентуално спечелена от въвеждането на паралелността. Тази стратегия обаче е безполезна, ако алгоритъмът, използван за тази архитектура, не предлага достатъчно паралелизъм. Проблем тук се явяват последователните и рекурсивните команди, т.е. командите, чието изпълнение зависи от резултата на предишни команди. За да се преодолее този проблем се прилагат техники като разгъване. Същата техника може да се приложи и при циклите, предоставяйки допълнителна степен на паралелизъм [14]. Други техники, които могат да се приложат, са софтуерен конвейер, алгебрични трансформации и други. Модуларността на алгоритъма също е особено важна, ако архитектурата върху която трябва да бъде приложен алгоритъма използва разпределени изчислителни ресурси, памети и други. Особено полезни в този случай за графовете.

3. Енергиен анализ

От оптимизациите разгледани до тук само тези на архитектурно и алгоритмово ниво са в сферата на компетентност на компютърния потребител. При проектирането на компютърна система с понижена енергийна консумация, задачата на проектанта е да намери точния баланс между оптимизациите с цел подобряване на енергийната консумация и на производителността. До относително неотдавна не беше възможно да се разбере точната

консумация на дадена схема или чип без измервания на реален негов образец. Ситуацията започна да се променя с въвеждането на програмируемите чипове (CPLD, FPGA), чиято консумация и спецификации по охлаждането, както е отбелязано в [15], трябва да бъдат определени в ранен стадий на процеса по проектирането им. Често това става дори преди проектирането на логиката.

С цел извършване на енергиен анализ по време на процеса на проектиране, са създадени различни инструменти, някои от които са интегрирани в развойните среди или се предоставят допълнително от фирмите производители. Като пример ще споменем предлаганите инструменти за енергиен анализ в две развойни среди – ISE на фирмата Xilinx и Galaxy Design на фирмата Synopsys, без да се спираме подробно на тях. Фирмата Xilinx предлага за развойната среда ISE продуктите XPE (Xilinx Power Estimator) и XPA (Xilinx Power Analyzer), а фирмата Synopsys предлага за развойната среда Galaxy Design анализатора Prime Time PX и допълнението към него Prime Rail, а така също и автоматичната система за оптимизация на енергийната консумация Power Compiler.

4. Заключение

Това изследване представя извършения обзор и анализ на методи за оптимизация на енергийната консумация. Бе установено, че най-големи придобивки се получават от оптимизации на така наречените високи нива – RTL (архитектурно) и алгоритмово нива. Също така, бе изведен изводът, че поради голямата сложност на съвременната техника, все по-голяма става необходимостта от наличието на автоматизирани системи, подпомагащи на проектанта да извършва анализ и оптимизация на проектите. И докато подобни системи за анализ са разработени в голям брой варианти, то с малки изключения системите за оптимизация все още са насочени към оптимизации на ниски нива. С цел понижаване на времето за разработка на

устройства с понижена енергийна консумация е необходимо разработването на автоматични системи за оптимизация на високо ниво, както и запознаване на проектантите с тях.

5. Литература

- [1]. Paul E. Landman, “Low Power Architecture Design Methodologies”, University of California 1994.
- [2]. Ricardo E. Gonzalez, “Low Power Architecture Design”, Stanford University, 1997.
- [3]. Jeff Scott, Lea Hwang Lee, John Arends, Bill Moyer, “Designing the Low- Power M-core Architecture”, Proc. IEEE Power Driven Microarchitecture Workshop, 1998.
- [4]. Binu Mathew, Al Davis, Mike Parker, “A Low Power Architecture for Embedded Perception”, University of Utah, 2004.
- [5]. Shi-Yu Huang, Chien-Jyh Liu. “A Low-power architecture for extended Finite State Machines using input gating”, National Tsing-Hua University Taiwan, 2007.
- [6]. H. Bakoglu, “Circuits, Interconnections, and Packaging for VLSI”, Addison-Wesley, 1990
- [7]. D. Benson, Y. Bobra, B. McWilliams, “Silicon Multichip Modules”, Hot chips symposium III, Santa Ana CA, 1990.
- [8]. C. Nagendra, U. Mehta, R. Owens, and M. Irwin, “A Comparison of the Power-Delay Characteristics of CMOS Adders”, Proceedings of the 1994 International Workshop on Low Power Design, Napa Valley, CA, 1994.
- [9]. V. Tiwari, P. Ashar, and S. Mailk, “Technology Mapping for Low Power”, Proceedings of the 30th Design Automation Conference, Anaheim, CA, 1993.
- [10]. C.-Y. Tsui, M. Pedram, and A. Despain, “Technology Decomposition and Mapping Targeting Low Power Dissipation”, Proceedings of the 30th Design Automation Conference, Anaheim, CA, 1993.

- [11]. L. Benini, M. Favalli, and B. Ricco, "Analysis of Hazard Contributions to Power Dissipation in CMOS IC's", 1994 International Workshop on Low-Power Design, Napa Valley, CA, 1994.
- [12]. <http://www.intel.com/design/mobile/datasheets/318914.htm>
- [13]. A. Chandrakasan, R. Allmon, A. Stratakos, and R. W. Brodersen, "Design of Portable Systems", Proceedings of CICC '94, San Diego, 1994.
- [14]. Тянев Д. С., "Организация на компютъра", Том 2, ISBN 978-954-20-0413-4, Технически университет – Варна, 2008.
- [15]. Brian Philoflsky, "Seven steps to an accurate worst-case scenario power analysis using Xilinx Power Estimator", white paper, Xilinx, 2008.
- [16]. Xilinx, "Xilinx Power Estimator User Guide", June 2007.
- [17]. Derek Curd, "Power Consumption in 65 mm FPGA", February 2007.
- [18]. Xilinx Power Analyzer
- [19]. Xilinx Power Solutions
- [20]. Synopsys, "Synopsys PrimeTime PX Expanding the PrimeTime solution with power analysis", Synopsys Inc, 2007.
- [21]. Synopsys, "PrimeRail", Synopsys Inc, 2007.
- [22]. Synopsys, "Power Compiler, automatic power management within Galaxy Design platform". Synopsys Inc, 2007.
- [23]. Kyoung-Mook Lim, Seh-Woong Jeong, Yong-Chun Kim, Seung-Jae Jeong, Hong-Kyu Kim, Yang-Ho Kim, Bong-Young Chung, Hyung-Lae Roh, and H.S. Yang, "CalmRisc: A low power Microcontroller with efficient coprocessor Interface", Computer Design, 1999. (ICCD '99) International Conference on 10-13 Oct. 1999 Page(s):299 – 302, Samsung Semiconductor and KAIST.
- [24]. E.P. Ramo, J. Resano, D. Mozos and F. Catthoor, "Memory hierarchy for high-performance and energy aware re-configurable systems", IET Computer Digital Technology, Vol.1 No 5, 2007.
- [25]. Giuseppe Notarangelo, Marco Gibilaro, Gaetano Palumbo, Francesco Pappalardo, Agatino Pennisi, "Low power strategy for a TFT controller", Proceedings of the Euromicro Symposium for Digital System Design, 2002.
- [26]. Rajeev Balasubramonian, Naveen Muralimanohar, Karthik Ramani, Venkatanand Venkatachalapathy, "Microarchitectural wire management for performance and power in partitioned architectures", Proceedings of the 11th International Symposium on High-Performance Computer Architecture, 2005.
- [27]. David Garrett, Mircea Stan, "Low Power Architecture of the Soft-Output Viterbi Algorithm", University of Virginia – department of Electrical Engineering, 1998.
- [28]. Radu M. Secareanu, Ilon Hartin, "Low power architectures using localized non-volatile memory and selective power shut-down", ISCAS 2006.

За контакти:

инж. маг. Станислав Георгиев
E-mail: stanislav1983@yahoo.com

доц. д-р инж. Димитър Стоянов Тянев
катедра "Компютърни науки и технологии"
Технически университет – Варна
WEB: www.tyanev.com

професор д.т.н.инж. Веселин Йосифов
Department "Computer engineering", FHTW-
Berlin, Deutschland,
E-mail: vjossifo@fhtw-berlin.de

МЕТОД ЗА РЕАЛИЗАЦИЯ НА АПАРАТНИ САМОУПРАВЛЯВАЩИ СЕ ЦИКЛИЧЕСКИ СТРУКТУРИ - ЧАСТ II

Димитър Тянев, Стамен Колев, Драгомир Янев

Резюме: Изследването представя нов метод за проектиране на операционни (изчислителни) структури, реализиращи определени разновидности на циклически алгоритми от вида с предварително известен брой повторения. Съществената характеристика на получените операционни структури се изразява в това, че те функционират без управляващ краен автомат. Методът може да бъде прилаган при проектиране както на синхронно работещи, така и на асинхронно работещи операционни структури, т.е. това могат да бъдат тактувани или нетактувани структури. Основното достоинство на метода се състои в опростяване и хомогенизиране на апаратурата, реализираща изчислителния процес. Получените структури са високо производителни, тъй като могат да бъдат конвейерно организирани.

Method for Realization of Self-Controlling Loop Apparatus Structures – Part II

Dimitar Tyanev, Stamen Kolev, Dragomir Yanev

Abstract: The present research represents a novel method for operational structures design, which implement a versatile set of count-controlled loop algorithms. Main feature of these structures is that they operate without controlling automata. The method can be used for the design of synchronous and asynchronous structures. It simplifies the structures, gives us the ability to organize the structures in pipeline fashion, which results in better throughput.

1. Увод

Основната идея, за редукция на управляващите крайни автомати (като съставна част от всяко цифрово устройство), която авторите изследват, е изложена и обсъждана от тях в [1], а отделни нейни разновидности в [2] и в [3]. Основната цел при изследването на тази идея е да покаже, че известната декомпозиция на Глушков [4], на която може да бъде подложен всеки изчислителен автомат, е възможно да бъде редуцирана и опростена за сметка на неговата управляваща част. В резултат на това логическата структура на един автомат ще бъде сведена само до своята модифицирана операционна част ([1] фигура 3, както и [4] пункт 3.8.3), првърщайки я, според разбирането на авторите, в самоуправляваща се. Постигането на тази цел е възможно, ако бъдат разработени методи за проектиране на безавтоматно управление на

хардуерните реализации на основните алгоритмични структури, срещащи се в изчислителния процес. Към настоящият момент такива са предложени за реализация на линейни и разклонени алгоритмични структури [1], както и за някои видове циклически структури [2]. Поради своето многообразие последните продължават да бъдат обект и в настоящото изследване.

2. Изложение

Предложените в [2] решения за циклически алгоритмични структури от вида с предварително известен брой повторения използват в своите схеми декрементни броячи, които се превключват по задния фронт на входните импулси (-1). Тъй като в много синхронни елементи, а така също и в програмируемите логически клетки на CPLD и FPGA елементите, се реализират превключвания по предния фронт на тактуващите импулси, тук са предложени

съответните алтернативни решения. Това са логически структури, в които всички структурни възли се превключват по преден фронт.

Въпросът е актуален с това, че при използване на еднофазов тактов генератор за синхронизация на превключванията в логическите структури, маркирането на отделни фиксирани моменти в рамките на тактовия период е затруднено, а то е крайно необходимо в смисъла на основната цел на това изследване. Тук се има предвид това, че в самоуправляващите се структури се налага да се изграждат различни обратни връзки, които създават условия за възникване на състезаващи се превключвания. Последните не винаги е възможно да бъдат овладяни, така че синхронизацията изисква приложение на различни подходи, включващи използването на противосъстезателни елементи, комбинирано използване на схеми, превключващи се по различни фронтове, както и използването на няколко фазов тактов генератор.

Самоуправляваща се логическа структура за реализация на цикли с предварително известен брой повторения

Методът за реализация на самоуправляваща се циклическа структура с линейно тяло е изложен в [2]. Като решение на изявения проблем със синхронизацията и на възможното многообразие тук са представени синтезираните и изследвани две логически структури: за цикли с конструктивно предопределен брой повторения и за цикли с променлив брой на повторенията. На фигура 1 е представена логическа структура от първата разновидност. Структурата изпълнява действията в тялото на цикъла чрез апаратната им реализация в комбинационната схема ОКС. Входните данни за всяка поредна итерация се поддържат от регистъра фиксатор, който записва новите резултати по предния фронт на селектираните

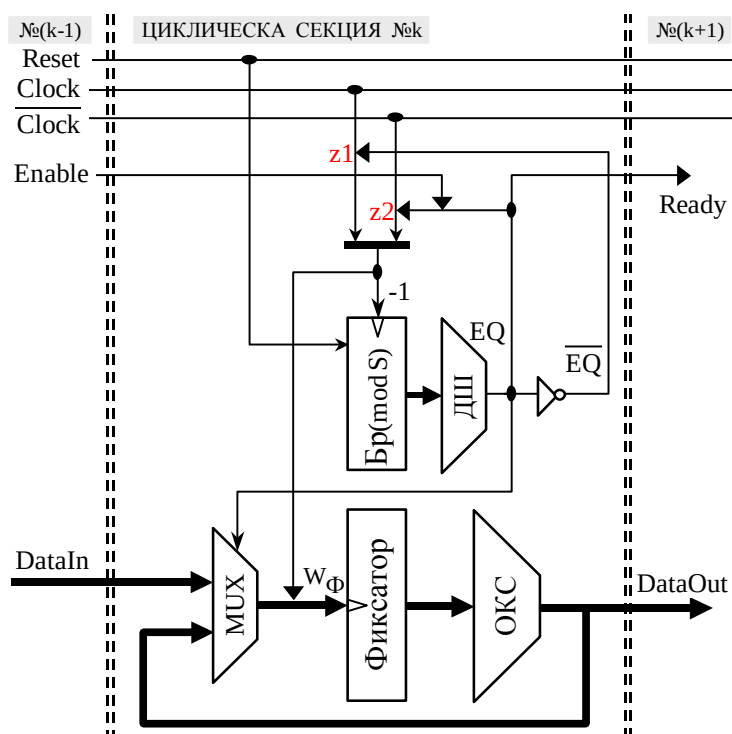
импулси W_{ϕ} . Условието за край на цикъла се реализира, модифицира и контролира от декрементния брояч и комбинационните схеми, свързани с него. В изходно състояние съдържанието на брояча Бр е нула, $EQ=1$ и входният мултиплексор е включен към шината *DataIn*. Въпреки че структурата се тактува непрекъснато, в нея не се извършват превключвания. Изчисленията започват с появата на разрешаващия сигнал *Enable*, както е показано на фигура 2. За правилно функциониране сигналът *Enable* следва да се появява по предния фронт на тактовите импулси и да има продължителност не повече от един период.

Както се вижда от примерната времедиаграма, за надеждна синхронизация на превключванията е използван и инверсния тактов сигнал. В изразения процес на превключване има два особени момента, които следва да бъдат пояснени.

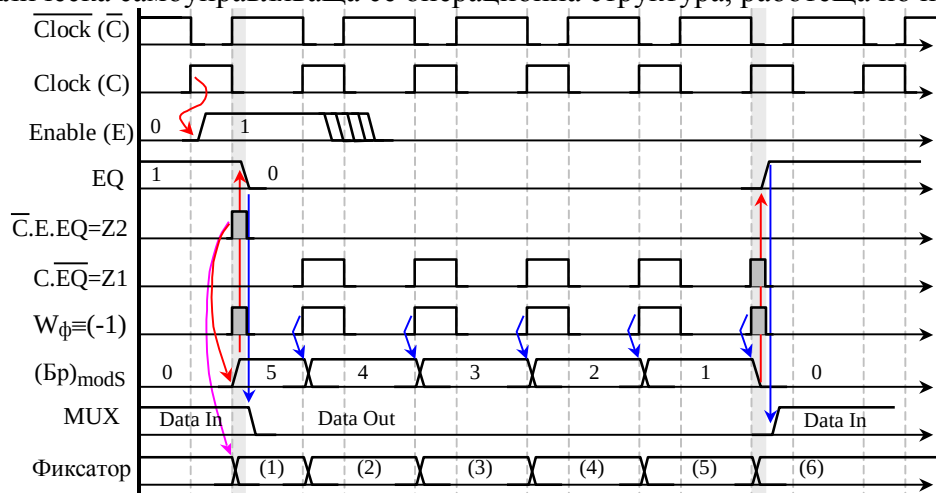
Първият се отнася до скъсената продължителност на първия и последния декрементни импулси (вижте абсцисата на сигнал (-1)). Тъй като в структурата от фигура 1 броячът се превключва по предния фронт на входните импулси, следствието - разпространяващият се по обратната връзка сигнал EQ (респективно и неговата инверсна стойност), затваря чрез нулевата си стойност със закъснение съответната конюнкция (вижте точка $z1$ и точка $z2$).

$$\begin{aligned} (-1) = z1 \cup z2 = & (\text{Clock} \cap \overline{EQ}) \cup \\ & \cup (\overline{\text{Clock}} \cap \text{Enable} \cap EQ) . \end{aligned} \quad (2.1.1)$$

В резултат на това, продължителността на тези скъсени импулси, е равна на времето за превключване на брояча, плюс времето за превключване на дешифратора, плюс времето за превключване на останалите логически елементи в обратната връзка. Това време е напълно достатъчно и въпреки че обратната връзка поражда това състезание, то не се самовъзпроизвежда, благодарение на използваните в брояча



Фиг. 1 Циклическа самоуправляваща се операционна структура, работеща по преден фронт



Фиг. 2 Времедиаграма за изпълнение на цикъл от 6 повторения

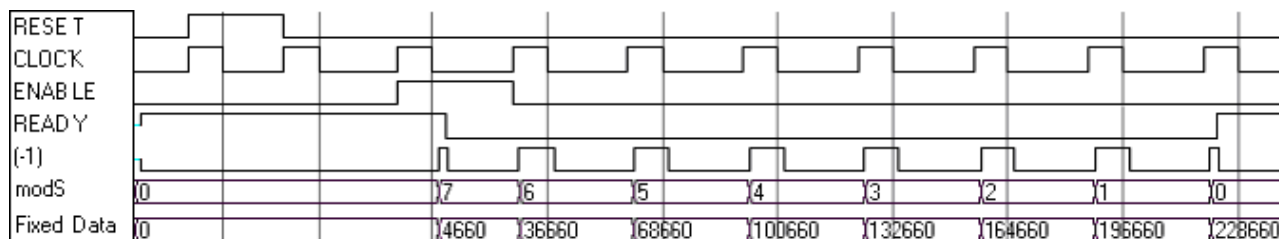
противосъстезателни тригери със структура *Edge*. Вторият особен момент се отнася до продължителността на инверсната фаза на тактовия сигнал. Тъй като тя определя престоя на началните входни данни във фиксатора (вижте абсцисата на фиксатора), то времето за първото операционно изчисление в схемата ОКС ще бъде по-малко от продължителността на периода. От тук следва, че за получаване на правилни резултати, трябва да се предяви изискване

както към тактовата честота, така и към коефициента на запълване на тактовия сигнал, който следва да е възможно най-малък. За всяко следващо изчисление (циклическо повторение), освен за първото, ще бъде отделено време, равно на продължителността на цял период.

Практическата проверка на синтезираната структура е изпълнена в средата ISE 10.1 на фирма *Xilinx* за интегрални схеми от серията *Spartan 3E*. На фигура 3 по-долу е представена

работата на генерирания модел. Изпълнен е цикъл с 8 повторения. За по-лесна проверка в операционната комбинационна схема ОКС е заложено итерационното изчисление на натрупваща се сума:

$$S_k = S_{k-1} + 32000, \text{ при } S_0 = 4660$$



Фиг. 3 Времедиаграма на Xilinx-модела

Самоуправляваща се логическа структура за реализация на цикли с променлив брой повторения

Тук на фигура 4 по-долу е представена синтезираната и изследвана логическа структура, съответстваща на циклическа алгоритмична структура от вида с предварително известен брой повторения, който обаче е променлив и е функция на предходни етапи на изчислителния процес. Циклическата организация с променлив брой повторения изисква използването на декрементен програмируем брояч, в който определената стойност S се записва в качеството си на негова начална стойност. Максимално възможната стойност е ограничена от модула за броене. Отчитайки споменатата програмируемост, логическата структура има отделен вход към брояча Бр. Освен това всички нейни логически възли с памет се превключват по преден фронт на синхро сигналите. Условието за край на цикъла се формира от дешифратора ДШ, който разпознава съдържание нула и формира сигнала EQ. Управлението на събитията, свързани със съдържанието на брояча, се постига чрез логическите схеми в неговите обратни връзки.

В изходно състояние съдържанието на брояча Бр е нула, $EQ=1$ и входният мултиплексор е включен към шината *DataIn*. Въпреки че структурата се тактува

Така върху последната абсциса *Fixed Data* от фигура 3 се виждат получаваните числени резултати: 4660, 36660, 68660 и т.н. .

непрекъснато, в нея не се извършват превключвания. Изчисленията започват с появата на разрешаващия сигнал *Enable*, както е показано на фигура 2. За правилно функциониране сигналът *Enable* следва да се появява по предния фронт на тактовите импулси и да има продължителност не повече от един период.

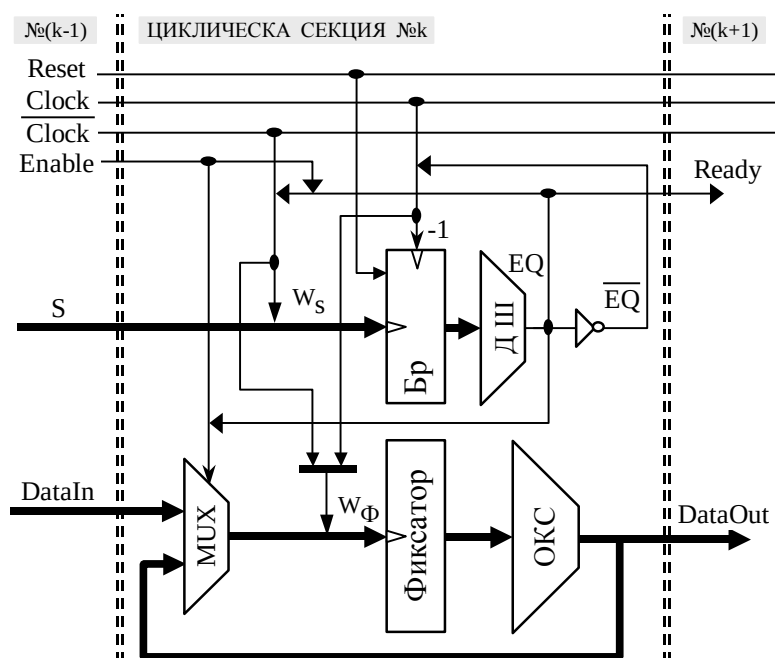
На фигура 5 е представена примерна времедиаграма за изпълнение на цикъл от 6 повторения. Както може да се види, условието за запис на нови данни и старт на изчисленията в структурата, се дава от сигналите W_S и W_Φ , които се формират според следната логика:

$$W_S = \overline{\text{Clock}} \cap (\text{Enable} \cap EQ) . \quad (2.2.1)$$

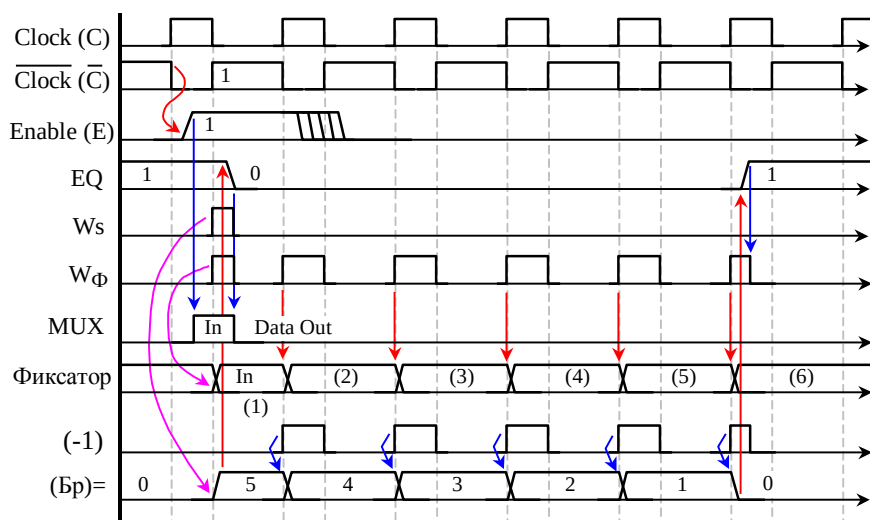
$$W_\Phi = W_S \cup (-1) = (\text{Clock} \cap \overline{EQ}) \cup (\text{Clock} \cap \text{Enable} \cap EQ) . \quad (2.2.2)$$

Използването на сигнала EQ в разглежданата структура е аналогично на изложеното за предишната. За тази структура цикълът с предусловие е възможен. В този случай, възможността за неизпълнение на действията от тялото на цикъла налагат операционната схема ОКС да бъде преместена в обратната връзка, а на изход *DataOut* да излиза съдържанието на регистъра-фиксатор. Така, ако записаната начална стойност в брояча се окаже равна на нула ($S=0$), то записаните

във фиксатора входни данни ще излязат без обработка на изход.



Фиг. 4 Програмируема циклическа самоуправяваща се операционна структура, работеща по преден фронт



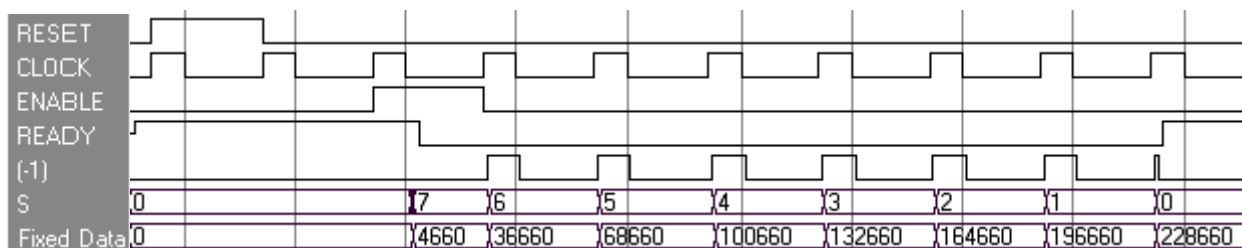
Фиг. 5 Времедиаграма на структура, работеща по преден фронт

Практическата проверка на синтезираната тук структура е изпълнена в средата ISE 10.1 на фирма *Xilinx* за интегрални схеми от серията *Spartan 3E*. На фигура 6 по-долу е представена работата на генерирания модел. Изпълнен е цикъл с 8 повторения при начална стойност $S=7$. Както в предишния модел, за по-лесна проверка, в

операционната комбинационна схема ОКС е заложено аналогичното итерационно изчисление на натрупващата се сума:

$$S_k = S_{k-1} + 32000 \quad , \quad \text{при } S_0 = 4660$$

Така върху последната абсциса от фигура 6 се виждат получаваните числени резултати.

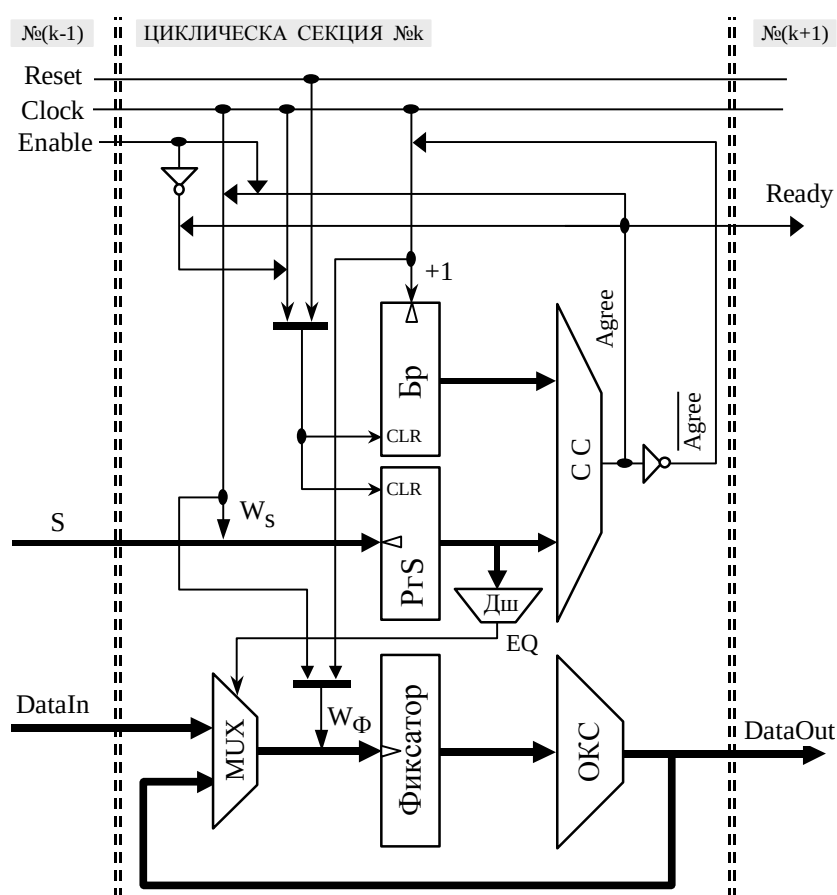


Фиг. 6 Времедиаграма на Xilinx-модела

Самоуправляваща се логическа структура за реализация на цикли с променлив брой повторения на базата на цифров компаратор

На фигура 7 е показан още един възможен вариант за структурата, реализираща цикли с предварително известен, но променлив брой повторения. Това е структура, в която е използван инкрементен брояч. Неговото изходно съдържание, както и това на регистъра на контролната стойност PrS, е винаги нула, което се постига чрез импулсите на

сигнала CLR (*Clear up*). След стартиране на процеса от сигнала *Enable*, входните данни *DataIn* се записват във фиксатора, контролната стойност *S* се записва в регистър PrS, а съдържанието на брояча започва да расте. Последното непрекъснато се сравнява с контролната стойност *S* от схемата за сравнение CC, която генерира в края на процеса сигнала за съвпадение *Agree*. Изчислителният процес протича под управлението вътрешни и външни сигнали, чиято логика е следната:



Фиг. 7 Циклическа операционна структура с променлив брой на повторенията

$$\text{CLR} = \text{Reset} \cup [C \cap (\text{Agree} \cap \overline{\text{Enable}})] \quad (2.3.1)$$

$$(+1) = C \cap \overline{\text{Agree}} \quad (2.3.2)$$

$$W_s = C \cap \text{Agree} \cap \text{Enable} \quad (2.3.3)$$

$$W_\phi = W_s \cup (+1) \quad (2.3.4)$$

Надеждността на превключванията и избягването на състезанията в структурата са осигурени чрез логически възли с различно превключване. Регистърът PrS и броячът Бр се превключват по заден фронт, а регистърът на фиксатора – по преден.

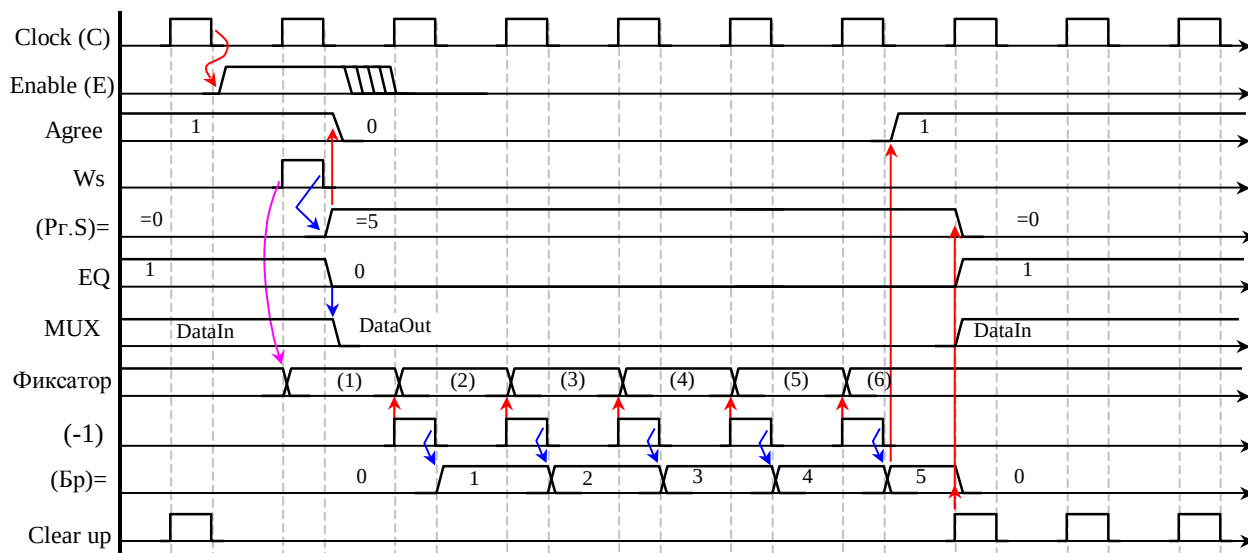
Процесът на функциониране на синтезираната структура е представен чрез времедиagramата от фигура 8.

3. Заключение

От сравнението на всички резултати, публикувани по-рано и получените тук, отнасящи се до циклическите структури, се налага следния извод: структури, които използват логически възли с памет, превключващи се по един и същ фронт на

управляващите сигнали, са склонни да генерират вътрешни за себе си състезания. В същото време, структурите, които комбинират превключващите фронтове не генерират състезания във вътрешността си. Възможните за някои структури състезания обаче не се самовъзпроизвеждат благодарение на използваните противосъстезателни тригерни елементи. В този смисъл за такива структури можем да говорим, че притежават частична асинхронност.

Представените тук резултати доказват възможността за апаратна реализация на самоуправляващи се изчислителни структури, свободни от автоматно управление. С това се засилва увереността на авторите, че апаратната реализация на произволни алгоритмични структури като самоуправляващи се, е възможна. Тук следва да се заяви, че все още съществуват алгоритмични структури, чието изследване в смисъла на основната идея е предстоящо.



Фиг. 8 Времедиagramа за изпълнение на цикъл с 6 повторения

Литература

- [1]. Josifov V., Kolev St., Tyanev D., Operational structures without controlling automata, International workshop on network and grid infrastructures, Institute

for parallel processing at the bulgarian academy of sciences department “distributed computing systems and networks”, Sofia, Bulgaria, 27-28 Sept. 2007, EC Project BIS21++ “Bulgarian IST centre of competence in 21 century”, <http://bis-21pp.acad.bg/events/events.htm> ;

<http://www.tyanev.com/home.php?lang=en&mid=23&mod=0&sub=4> .

- [2]. Тянев Д. С., Колев С. И., Йосифов В., Метод за реализация на апаратни самоуправляващи се циклически структури, Годишник на ТУ-Варна, Юбилеен сборник “45 години ТУ-Варна”, 2007, ISSN 1311-896X, стр. 130-135,

<http://www.tyanev.com/home.php?lang=bg&mid=17&mod=0&sub=4> .

- [3]. Kolev St., Jossifov V., Master thesis: “Design of asynchronous pipelines for floating point multiplication and division”, FHTW – Berlin, Deutschland, 2007.

- [4]. Тянев Д. С., Организация на компютъра, Том 1, ISBN 978-954-20-0412-7, Технически университет-Варна, 2008.

- [5]. Patterson D. A., Hennessy J. L., Computer Organization And Design, Morgan Kaufmann Publishers, ISBN 1-55860-604-1, 2005.

- [6]. Sutherland I. E., Micropipelines, <http://research.sun.com/vlsi/Publications/KPDisclosed/micropipelines/cmultipipelines.pdf>

- [7]. John F. Wakerly, Digital Design □ principles and practices, Fourth Edition, Prentice Hall, ISBN 0-13-186389-4, 2005.

- [8]. Булгаков С., Мещеряков В., Новоселов В., Шумилов Л., Проектирование

цифровых систем на комплектах микропрограммируемых БИС, Москва, “Радио и связь”, 1984.

- [9]. Тянев Д. С., Организация на компютъра – проектиране на логически структури, ISBN 954-20-0259-9, ТУ-Варна, 2004.

- [10]. WEB-адрес на фирма XILINX – <http://www.xilinx.com/>

- [11]. Yau-Hwang Kuo, Shaw-Pyng Lo, Partitioning and Scheduling of Asynchronous Pipelines, Institute of Information Engineering, National Cheng Kung University, Tainan, Taiwan., Proc. of CompEuro’92, The Netherlands, May, 1992.

За контакти:

доц. д-р инж. Димитър Тянев
катедра “Компютърни науки и технологии”,
Технически университет – Варна
WEB: www.tyanev.com.

инж. Стамен Колев
Фирма “TeamR3” ООД - Варна
sosage6@yahoo.com.

инж. Драгомир Янев,
Фирма “NAND” ЕООД – Варна,
dragomir_yanev@abv.bg.

ПРЕДИСКРЕТИЗАЦИЯ В РЕАЛНО ВРЕМЕ НА ЦИФРОВ ЗВУКОВ СИГНАЛ ЧРЕЗ ИНТЕРПОЛАЦИЯ С ОГРАНИЧЕНА ЧЕСТОТНА ЛЕНТА – ЧАСТ ВТОРА

Станислав Б. Кирилов, Лъчезар Ил. Георгиев

Резюме: В доклада се използва предлаганата модификация на алгоритъма за предискретизация чрез интерполация с ограничена честотна лента, който за разлика от линейната и кубичната интерполация дава високо качество на предискретизирания сигнал, но разчита на изчислителната мощ на съвременните микропроцесори и по-точно на техния FPU- и/или SIMD-блок. Тази реализация е подходяща за оптимизация с използване на SIMD-инструкции (3DNow!/SSE), което е осъществено на практика.

Real-time digital audio signal resampling through bandlimited interpolation – part two

Stanislav B. Kirilov, Latchezar I. Georgiev

Abstract: In this paper we use the proposed modification of the resampling algorithm with bandlimited interpolation, which unlike the linear and cubic interpolation gives a high quality of the resampled signal but relies on the computing power of the modern microprocessors, and more exactly on their FPU and/or SIMD block. This implementation is very suitable for optimisation using SIMD instructions (3DNow!/SSE), which has been realised in practice.

4. Практическа реализация

В част първа на статията [1] беше разгледана теорията на алгоритъма за предискретизация чрез интерполация с ограничена честотна лента.

Преди всичко да уточним, че обикновено входният и изходният сигнал са целочислени, но е възможно да бъдат и числа с плаваща запетая, а междинните изчисления се извършват задължително с числа с плаваща запетая и с по-висока точност от тази на изходния сигнал. Възможно е изчисленията да се извършват целочислено (с фиксирана запетая), но това би изисквало висока разрядност и множество операции за битови премествания и в общия случай би било по-бавно и доста по-трудно за реализация и изчисление.

За по-нататъшните изчисления ще са ни необходими следните константи, които не се променят през целия процес на предискретизация (*Заб.* Оттук нататък ще използваме запис, подобен на езика Си):

```
win_const = 2 * pi / M
win_sin = sin(win_const)
win_cos = cos(win_const)
win_cos_2 = 2 * win_cos
win_C = 0.5625
win1_const = 6 * pi / M
win1_sin = sin(win1_const)
win1_cos = cos(win1_const)
win1_cos_2 = 2 * win1_cos
win1_C = 0.0625
```

Нека променливата *speed_step* е равна на скоростта на възпроизвеждане. Ако *speed_step* = 1, то няма промяна на скоростта, т. е. тя е 100% от първоначалната.

Честотата на филтъра w_c изчисляваме така (за паралел във фигурни скоби {} са дадени зависимостите за идеалния филтър)

```
Wc=3.0/speed_step*(Fs_in/Fs_out)
{Wc=pi/speed_step*(Fs_in/Fs_out)}
Ако Wc > 3.14, тогава Wc=3.14
{Ако Wc > pi, тогава Wc=pi}
```

Ще ни трябва стойността на $2\cos(w_c)$:
 $\cos_Wc_2 = 2*\cos(Wc)$

Променливата *position* указва точната позиция във времето на точката от изходния сигнал, която искаме да изчислим.

Стойността на *position* се състои от цяла и дробна част, т. е. тя е от вида $n.d$. Стойността на точката от изходния сигнал, която търсим, се намира между отчети n и $n+1$ на входния сигнал и е на разстояние d от n -тия отчет. За удобство ще използваме $d < 1$, така че $position = n + d$, където n е цялата, а d е дробната част.

След изчисляването на текущия отчет от изходния сигнал новата стойност на *position* се определя така:

$position = position + speed_step$

Удобно е (но не е задължително) *position* и *speed_step* да са числа с фиксирана запетая, за да има лесен достъп до цялата и до дробната им част.

За удобство ще приемем n -тия (т.е. текущия) отчет от изходния сигнал за нулев (не стойността, а индексът му е 0 вместо n). Тогава $n-1$ ще има индекс -1 ,

$n+1$ ще има индекс 1 и т. н.

Възможни са 2 случая за стойността на *position* – да е цяло (т. е. $d=0$) или дробно число (т. е. $0 < d < 1$). Първият е частен случай и ако можехме да извършим идеална предискретизация, тогава стойността на изходния сигнал щеше да е точно равна на тази на входния и нямаше да се налага да правим никакви изчисления, но тъй като използваме филтър на честота, която в общия случай не е равна точно на $F_s/2$, то се налага изчислението и на тази стойност, защото формата на изходния сигнал няма да е точно същата като тази на входния.

1) Нека $d = 0$ (*position* е цяло число).

В променливата *A* ще натрупваме сумата от произведенията, необходима за конволюцията:

```
A = x[n] * Wc
s_1 = 0
s = sin(Wc)
w_1 = win_C
w1_1 = win1_C
w = win_C * win_cos
w1 = win1_C * win1_cos
i = 1
do {
    A += (x[n-i] + x[n+i]) * s * (w - w1 + 0.5) / i
    s_0 = s; s = s * cos_Wc_2 - s_1; s_1=s_0
    w_0 = w; w = w * win_cos_2 - w_1; w_1=w_0
    w1_0 = w1; w1 = w1 * win1_cos_2 - w1_1; w1_1=w1_0
} while (++i < M/2)
```

2) Нека $0 < d < 1$:

```
A = 0
x = -d
sl = sr_1 = sin(Wc*x)
sl_1 = sr = sin(Wc*(x+1))
wl = wr_1 = win_C * cos(win_const*x)           // Виж обяснението по-долу за
wl_1 = wr = win_C * cos(win_const*(x+1))         // оптимизация на тези изрази
wl1 = wr1_1 = win1_C * cos(win1_const*x)          //      "      "      "      "      "
wl1_1 = wr1 = win1_C * cos(win1_const*(x+1))      //      "      "      "      "      "
```

// left part:

```
i = 0
do {
    A += x[n-i] * sl * (wl - wl1 + 0.5) / x
    sl_0 = s; sl = sl * cos_Wc_2 - sl_1; sl_1=sl_0
    wl_0 = wl; wl = wl * win_cos_2 - wl_1; wl_1=wl_0
```

```

        wl1_0 = wl1; wl1 = wl1 * win1_cos_2 - wl1_1; wl1_1=wl1_0
        x -= 1
    } while (++i < M/2)

// right part:
x = 1 - d
i = 1
do {
    A += x[n+i] * sr * (wr - wr1 + 0.5) / x
    sr_0 = sr; sr = sr * cos_Wc_2 - sr_1; sr_1=sr_0
    wr_0 = wr; wr = wr * win_cos_2 - wr_1; wr_1=wr_0
    wr1_0 = wr1; wr1 = wr1 * win1_cos_2 - wr1_1; wr1_1=wr1_0
    x += 1
} while (++i <= M/2)

```

След изчисляването на който и да е от горните два случая (те могат да бъдат обединени в един и проверката на стойността на d да инициализира останалите променливи с различни стойности, но това не променя нищо принципно), остава да изчислим:

$$A = A * (1/\pi)$$

Вместо да делим A на π , може да го умножим с предварително изчислената константа $1/\pi$. Добре е да го умножаваме с число, малко по-малко от $1/\pi$, за да се избегнат „отрязванията“ на минималните и максималните стойности, които може да прехвърлят допустимите, когато резултатът трябва да се преобразува в цяло число. Причината за появата на такива екстремални стойности е т. нар. ефект на

Гибс, породен от филтрирането на сигнала и отстраняването на най-високочестотните му съставки.

Сега да разгледаме по-подробно дадените по-горе програмни фрагменти.

1) За $d = 0$:

Филтърът е симетрично разположен спрямо текущия (n -ти) отчет на входния сигнал. Търси се стойността на изходния сигнал точно в тази точка.

За централната точка на филтъра имаме $\sin(w_c \cdot x)/x$ при $x \rightarrow 0$, т. е. средният коефициент на филтъра (при $i=0$, $-M/2 \leq i \leq M/2$) е равен на w_c (допълнителното делене на π оставяме за края). И така:

$$A = x[n] * W_c \text{ /* Начална стойност по средата на филтъра */}$$

Означенията отгук нататък са следните:

s_1 – предишна стойност на $\sin$
 w – текуща стойност на първия $\cos$ в тегловната функция, умножена по $9/16$
 w_1 – предишна стойност на първия $\cos$ в тегловната функция, умножена по $9/16$
 $w1_1$ – предишна стойност на втория $\cos$ в тегловната функция, умножена по $1/16$
 $s_1 = 0$ // $\sin(0)=0$ - начална стойност (при $i=0$)
 $s = \sin(W_c)$ // $\sin(W_c)$ - текуща стойност (при $i=1$)
 $w_1 = \text{win}_C$ // начална стойност (при $i=0$)
 $w1_1 = \text{win1}_C$ // начална стойност (при $i=0$)
 $w = \text{win}_C * \text{win_cos}$ // текуща стойност (при $i=1$)
 $w1 = \text{win1}_C * \text{win1_cos}$ // текуща стойност (при $i=1$)

След като имаме подготвени началните стойности, можем да започнем цикъл, с който да изчислим останалите коефициенти, да ги умножим със съответните стойности от входния сигнал и да съберем получените произведения така, както го изискват правилата за конволюция.

$i = 1$ /* Вече имаме стойността при $i=0$, започваме от 1 */

Тъй като при $d=0$ стойностите на коефициентите, които са симетрично разположени спрямо средната стойност ($i=0$) са еднакви, ще изчисляваме коефициентите само веднъж и ще ги

използваме едновременно и за „лявата“ и за „дясната“ част на филтъра.

$A += (x[n-i] + x[n+i]) * s * (w - w1 + 0.5) / i$

Горният ред извършва едновременно умножението на стойностите на входния сигнал, разположени „вляво“ и „вдясно“ на филтъра, със синуса s от sinc-функцията и с тегловната функция $(w - w1 + 0.5)$, извършва делението от sinc-функцията и натрупва резултата в A . Същинската филтрация се извършва само в този ред.

Следните редове служат за рекурсивно получаване на следващите стойности на синуса от sinc-функцията и двата косинуса от тегловната функция (предварително умножени със съответните константи):

$s_0=s; s=s*\cos_Wc_2-s_1; s_1=s_0$

$w_0=w; w=w*\sin_2-w_1; w_1=w_0$

$w1_0=w1; w1=w1*\sin1_2-w1_1; w1_1=w1_0$

Вижда се, че за получаването на следващата стойност е необходимо само едно умножение с константа и едно изваждане. Новата стойност на всяка синусоида се получава от предишните две само чрез едно умножение и изваждане. Това е най-съществената оптимизация за скорост в алгоритъма.

$\text{while} (++i < M/2)$

Условието на този while би трябвало да се запише като $(++i \leq M/2)$, но тъй като тегловната функция има стойност 0 в крайните точки, това не е необходимо.

2) За $0 < d < 1$:

В този случай аналоговият филтър отново е симетричен спрямо централната си точка, т. е. тази, в която търсим

стойността на изходния сигнал, но тя се намира някъде между два съседни отчета (n -тия и $n+1$ -вия) на входния сигнал и коефициентите на цифровия филтър „вляво“ и „вдясно“ са различни. Затова имаме два отделни цикъла, изчисляващи съответно „лявата“ и „дясната“ част на филтъра (и извършващи умножението и събирането от филтрацията). И двата цикъла започват от центъра и се движат съответно „наляво“ и „надясно“ към краищата на филтъра.

Този начин на изчисление има предимството, че централните стойности на филтъра биват изчислявани по-точно, защото са в началото на цикъла. А те са по-важни, защото имат по-голяма абсолютна стойност (използват се за умножение със съответните стойности от входния сигнал). След това грешката от рекурсивното изчисление на следващите коефициенти на филтъра постепенно се натрупва към краищата на филтъра, но там стойностите на коефициентите постепенно спадат към нулата.

Започваме с изчисляване на началните стойности:

$A=0$ //Начална стойност на сумата

$x=-d$ //Начална точка "вляво"

Спрямо точката, която търсим, най-близките коефициенти „вляво“ и „вдясно“ са разположени на позиции $-d$ и $-d+1$ съответно по времевата ос (съответстват на отчети n и $n+1$ от входния сигнал). Означенията са следните:

sl – текуща стойност на $\sin$ за "левия" цикъл

sl_1 – предишна стойност на $\sin$ за "левия" цикъл

wl – текуща стойност на първия $\cos$ в тегловната функция, умножена по $9/16$, за „левия“ цикъл

wl_1 – предишна стойност на първия $\cos$ в тегловната функция, умножена по $9/16$, за „левия“ цикъл

$wl1$ – текуща стойност на втория $\cos$ в тегловната функция, умножена по $1/16$, за „левия“ цикъл

$wl1_1$ – предишна стойност на втория $\cos$ в тегловната функция, умножена по $1/16$, за „левия“ цикъл

sr – текуща стойност на $\sin$ за „десния“ цикъл

sl_1 – предишна стойност на $\sin$ за „десния“ цикъл

wl – текуща стойност на първия cos в тегловната функция, умножена по 9/16, за „десния“ цикъл
 wl_1 – предишна стойност на първия cos в тегловната функция, умножена по 9/16, за „десния“ цикъл
 w1 – текуща стойност на втория cos в тегловната функция, умножена по 1/16, за „десния“ цикъл
 w1_1 – предишна стойност на втория cos в тегловната функция, умножена по 1/16, за „десния“ цикъл

sl = sr_1 = sin(Wc*x)
 sl_1 = sr = sin(Wc*(x+1))

Началната стойност за sin „вляво“ използваме за предходна стойност на sin „вдясно“ и обратно.

За разлика от синусите, където грешката за числа с плаваща запетая с единична точност (24 бита мантиса, 8 бита порядък) е твърде голяма, можем да оптимизираме косинусите от тегловната функция

wl_1 = wr = win_C * cos(win_const * x + win_const)
 w1_1 = wr1 = win1_C * cos(win1_const * x + win1_const)
 или

wl_1 = wr = win_C*cos(win_const*x)*cos(win_const)+sin(win_const*x)*sin(win_const)
 w1_1 = wr1 = win1_C*cos(win_const1*x)*cos(win_const1)+sin(win_const1*x)*sin(win_const1

Тук sin(win_const), cos(win_const), sin(win1_const) и cos(win1_const) са изчислени преди цикъла (за предпочитане чрез fsincos); sin(win_const*x), cos(win_const*x), sin(win1_const*x) и cos(win1_const*x) се изчисляват чрез инструкцията fsincos.

Оттук нататък циклите за „лявата“ и „дясната“ половина на филтъра са аналогични на цикъла при $d=0$.

И накрая умножаваме A с $1/\pi$ или съответната подходяща константа, преобразуваме A в цяло число и го записваме като текуща стойност на изходния сигнал $y[m]$. Например за 16 бита:

```
if (A > 32767) A=32767
if (A < -32768) A=-32768
y[m] = (short) A
```

5. Точност на изчисленията. „Dithering“

Когато изходният сигнал е с точност 16 бита (CD-стандарт), тогава можем да използваме изчисления с числа с плаваща запетая с единична точност (24 бита мантиса, 8 бита порядък).

wl=wr_1=win_C*cos(win_const*x)
 wl_1=wr=win_C*cos(win_const*(x+1))
 w1=wr1_1=win1_C*cos(win1_const*x)
 w1_1=wr1=win1_C*cos(win1_const*(x+1))

по следния начин:

Ако изходният сигнал е с по-висока точност (24 или 32 бита), тогава трябва да използваме изчисления с двойна точност (48 бита мантиса и 16 бита порядък) или двойна увеличена точност (64 бита мантиса и 16 бита порядък).

При използване на изчисления с единична точност и изходен сигнал 16 бита, много рядко се получава разлика +1 или -1 спрямо изчисленията с двойна или двойна увеличена точност и ако се използва dither-сигнал, тия малки неточности се поглъщат напълно от него. Дори и да не се използва dither-сигнал, средното ниво на внесения от тези неточности шум е под -100 dB.

Първоначалната стойност на dither-сигнала изчисляваме като $dither=random()$, а следващите – като $dither=random()-dither$, където функцията $random()$ връща [псевдо]случайно число в интервала [0,1).

Текущата стойност от dither-сигнала се прибавя към текущата стойност на изходния сигнал точно преди преобразуването от подробно в цяло число.

6. SIMD-оптимизации

Всичко описано дотук се отнасяше за изчисления с помощта на x87 копроцесор, а алгоритмите бяха описани за моно-сигнал. Ясно е, че за стерео- или многоканален сигнал коефициентите на филтъра трябва да се изчисляват веднъж за всичките два или повече изходни отчета. Но самите умножения и събирания от конволюцията трябва да се направят два или повече пъти съответно. Това съвсем естествено навежда на мисълта за SIMD- (Single Instruction Multiple Data) оптимизация. С помощта на 3DNow!- или SSE2-инструкции могат да се изчисляват двата текущи отчета на стереосигнал едновременно, в първия случай с единична, а във втория – с двойна точност.

С помощта на SSE-инструкции могат да се изчисляват 4 отчета с единична точност едновременно, а при стерео-сигнал това могат да бъдат n -тите и $n+1$ -те отчета.

За съжаление няма SIMD-инструкции за изчисляване на $\sin$ и $\cos$. Затова можем да използваме формулите за апроксимация на тригонометричните функции от [2]:

$$\begin{aligned}\sin(a) &= (((((s1 * a^2 + s2) * a^2 + s3) * \\ &\quad a^2 + s4) * a^2 + s5) * a^2 + 1) * a \\ \cos(a) &= (((((c1 * a^2 + c2) * a^2 + c3) * \\ &\quad a^2 + c4) * a^2 + c5) * a^2 + 1 \\ &\quad \text{където:}\end{aligned}$$

$$\begin{aligned}s1 &= -0.0000000239 \quad c1 = -0.0000002605 \\ s2 &= +0.0000027526 \quad c2 = +0.0000247609 \\ s3 &= -0.000198409 \quad c3 = -0.0013888397 \\ s4 &= +0.0083333315 \quad c4 = +0.0416666418 \\ s5 &= -0.1666666664 \quad c5 = -0.4999999963\end{aligned}$$

Дадените константи са с по-висока точност от тази на числата с плаваща запетая с единична точност, но дори приблизителното им представяне в този формат е достатъчно за 16-битов изходен сигнал.

Важно предимство при SIMD е, че с помощта на тези формули и 3DNow!- или SSE2-инструкции могат да се изчисляват едновременно 2 синуса и 2 косинуса с единична точност или 1 синус и 1 косинус с двойна точност, а чрез SSE-инструкции – цели 4 тригонометрични операции (с единична точност) едновременно!

7. Изводи

Разгледаният алгоритъм е подходящ за SIMD-оптимизация. Новост в реализацията е апроксимацията на синуса и косинуса с формулите на Карлсън–Голдщайн вместо с бавни тригонометрични инструкции, с което се повишава бързодействието.

Литература

- [1] Станислав Б. Кирилов, Лъчезар Ил. Георгиев, „Предискретизация в реално време на цифров звуков сигнал чрез интерполация с ограничена честотна лента – част първа“, сп. „Компютърни науки и технологии“, кн. 1, Варна, 2008 г.
- [2] B. Carlson, M. Goldstein, “Rational approximation of functions,” report number LA-1943, Los Alamos Scientific Laboratory, Los Alamos, New Mexico, USA, 1955.

За контакти:

гл. ас. инж. Лъчезар Илиев Георгиев
кафедра „Компютърни науки и технологии“
Технически университет – Варна
E-mail: lucho@gawab.com

ТЕЛЕВИЗИОНЕН КООРДИНАТОР НА АВТОМАТИЗИРАНА СИСТЕМА ЗА УПРАВЛЕНИЕ НА БОЕН БЕЗПИЛОТЕН ЛЕТАТЕЛЕН АПАРАТ

Валентина Т. Цекова

Резюме: Излизайки от обстоятелството, че в Р. България се разработват и произвеждат безпилотни летателни апарати (БЛА), които могат да се използват за наблюдение и разузнаване, е предложен метод за тяхното усъвършенстване и превръщане в бойни. Разгледан е алгоритъмът на работа на телевизионен координатор на система за управление на такъв БЛА.

Television Coordinator of Automated Control System of Unmanned Combat Air Vehicle

Valentina T. Tsekova

Abstract: Coming from the facts that in Bulgaria are developed and produced unmanned air vehicle (UAV), which are used for observation and reconnaissance, is proposed a method for their elaboration and transformation into unmanned combat air vehicle (UCAV). In this paper is described the algorithm of the television coordinator's work of control system of the UCAV.

1. Увод

Постигнатите успехи в областта на самолетостроенето, авиационната електроника, бордовите сензори с различно предназначение, добиващи полезна информация и на комбинираните навигационни системи – спътникови, инерциални и радионавигационни, създадоха възможност за разработване и производство на различни видове безпилотни летателни апарати, които решават практически всички задачи, възлагани на пилотираната авиация [1,2], включвайки различни видове разузнаване до пряко ударно въздействие върху различни видове обекти и цели.

В Р. България работата по създаването на БЛА, които да се използват за бойната подготовка на войските, започва от 1970 г. и с известни периоди на прекъсване продължава и до сега. Понастоящем в страната ни съществуват няколко фирми, които разработват и произвеждат леки БЛА, предназначени основно за въздушно наблюдение и контрол на различни наземни и морски обекти по всяко време на денонощието във

видимия и инфрачервения диапазон и в реален мащаб на времето.

При модернизация този вид безпилотни летателни апарати могат да бъдат превърнати в бойни, които да поразяват целите чрез пряко попадение в тях. Това може да бъде осъществено по следния начин. Бордовата разузнавателна система (система за наблюдение) по команда от земята трябва да съвмести своята оптична ос с надлъжната ос на самия БЛА. В наземната командна станция, където се приема телевизионното изображение, трябва да се разположи апаратура (координатор), която да измерва отклонението на изображението на избраната за поразяване цел от центъра на екрана на мотитора, което е еквивалентно на отклонението на надлъжната ос на БЛА от направлението към целта. Въз основа на това отклонение ще се изработват команди, които така да изменят траекторията на полета на безпилотния самолет, че надлъжната ос на БЛА непрекъснато (приблизително) да съвпада с направлението към целта.

Целта на настоящата статия е да предложи именно такъв тип координатор.

2. Изложение

По принцип координаторът на автоматизираната система за управление на боен БЛА, поразяващ цел с пряко попадение в нея, не трябва да се отличава качествено от координаторите на други управляеми поразяващи средства. Той също трябва да определя координатите на подвижен обект (цел) в пространството и следователно неговата функционална схема може да бъде разглеждана като състояща се от две основни части: дискриминатор, който да определя ъгловото разсъгласуване на БЛА от равносигналното направление и филтър (изпълнителни органи), който да отстранява това разсъгласуване.

Тъй като нашата основна цел е създаване на система за управление на БЛА – телевизионен (ТВ) / инфрачервен (ИЧ) разузнавач, която да е способна по желание на оператора на самолета да го превърне в боен БЛА, който да може да поразии избраната цел, то е напълно естествено да използваме информацията, получавана от телевизионната разузнавателна система, за определяне на координатите на целта.

В телевизионните измерватели на координати принципно нов елемент е телевизионният чувствителен елемент (датчик). Поради това основно внимание ще бъде обърнато на телевизионния дискриминатор, като се счита, че начинът на отстраняване на измереното разсъгласуване не се отличава принципно от използваните дискриминатори в другите типове координатори.

Априори може да се приеме, че използването на телевизионни сензори е възможно само за оптически контрастни обекти (спрямо обкръжаващия фон). Това се отнася и за провеждане на самото разузнаване. Контрастността на обекта трябва да бъде такава, че да позволява увереното отделяне на сигнала на целта от сигналите на фона в комплексния телевизионен сигнал.

Евристически като се излиза от прин-

ципите, използвани в телевизията, могат да бъдат предложени голям брой функционални схеми на телевизионни датчици на координати и координатори [3]. Тук ще бъде разгледана една от тях, която е показана на фиг. 1.

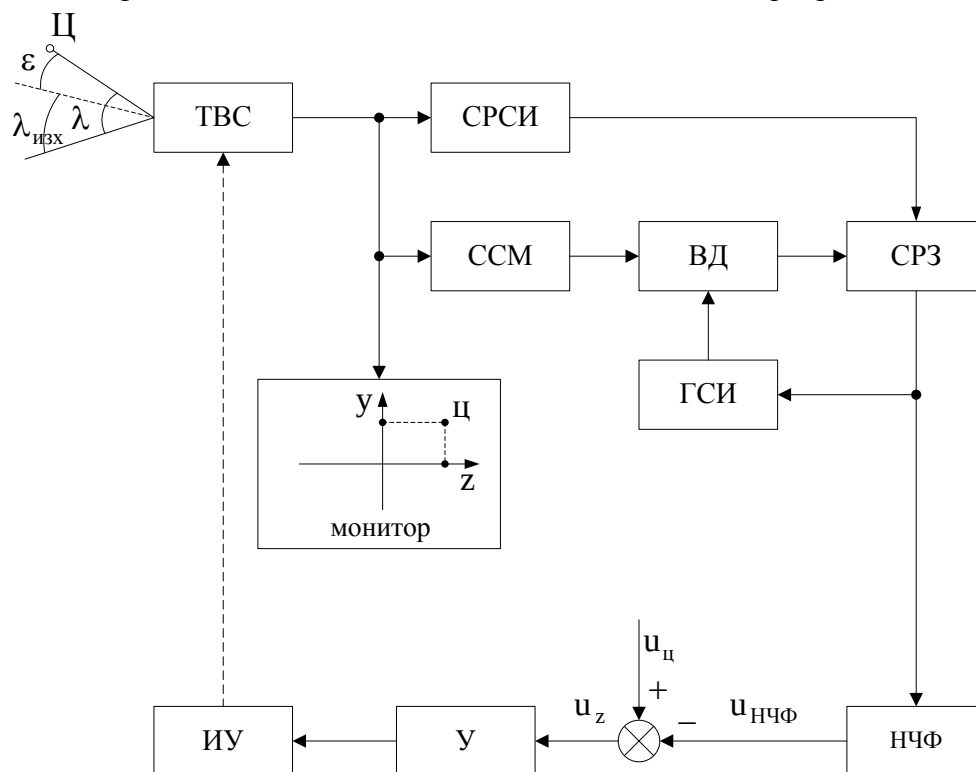
Функционалната схема на ТВ координатор е представена само за един канал (следене по оста Oz). Схемата за следене по другия канал се отличава само по това, че вместо редовите синхроимпулси се отделят кадровите синхроимпулси.

Телевизионното изображение на целта се проектира върху екрана на монитора, като за целите на изследването му ще го разглеждаме като обикновен телевизионен приемник. Това изображение има координати u и z спрямо центъра на екрана на монитора, който е приет за начало на измервателната координатна система. Тези координати се определят от маркера, с който операторът отбелязва (посочва) целта.

Тази система работи по следния начин. Комплексният телевизионен сигнал от телевизионната система (ТВС) постъпва в селектор на редови селектиращи импулси (СРСИ) и в селектор на сигнала от маркера (ССМ). След ССМ сигналът от маркера постъпва на входа на временен детектор (ВД), който може да бъде с един или два интегратора. Във временния детектор постъпват и стробиращите импулси от генератора на стробиращи импулси (ГСИ). Амплитудата и фазата на изходното напрежение на временния детектор са функция на площите на съвпадение на импулса на маркера от ССМ и стробиращите импулси от ГСИ. Това напрежение управлява ширината на импулса, изработван от схемата за регулиране на задръжката (СРЗ). Началото на импулса на СРЗ се задава от редовите импулси, а неговата продължителност е пропорционална на временното разстояние между редовия синхронизиращ импулс и импулса на маркера. Този импулс постъпва на входа на генератора на стробиращи импулси, където по задния фронт на изходния сигнал на ГСИ се образуват

стробиращи импулси, подавани към временния детектор. По този начин се

осъществява следенето за временното положение на маркера.



Фиг.1 Функционална схема на телевизионен координатор

Импулсът от схемата за регулиране на задръжката постъпва на входа на нискочестотен филтър (НЧФ). На неговия изход се получава напрежение, което е пропорционално на временното разстояние между редовите синхроимпулси и сигнала от маркера. Това напрежение може да се подава направо към усилвателя (У) на изпълнителното устройство (ИУ). Тогава началото на измервателната координатна система ще съвпада с началото на реда, в който се намира импулсът на маркера. За изследване на координатора е без значение къде се намира началото на измервателната координатна система, но за насочването на БЛА в най-общ случай е желателно динамичния диапазон на следене да бъде най-голям. Динамичният диапазон на следене от координатора е най-голям, когато началото на измервателната система съвпада с геометричния център на монитора - в случая със средата на реда.

При решаването на някои конкретни практически задачи е по-изгодно зоната на следене да бъде асиметрична, т.е. центърът на следене да може да се измества наляво или надясно от геометричния център. Това изместване може да бъде реализирано, ако напрежението от изхода на нискочестотния филтър $u_{нчф}$ се извади от някакво постоянно напрежение $u_{ц}$, което отговаря на следното условие $0 \leq u_{ц} \leq u_{нчфmax}$. За да съвпадне началото на координатната система с геометричния център, е необходимо да се изпълни равенството $u_{ц} = u_{нчфmax}/2$.

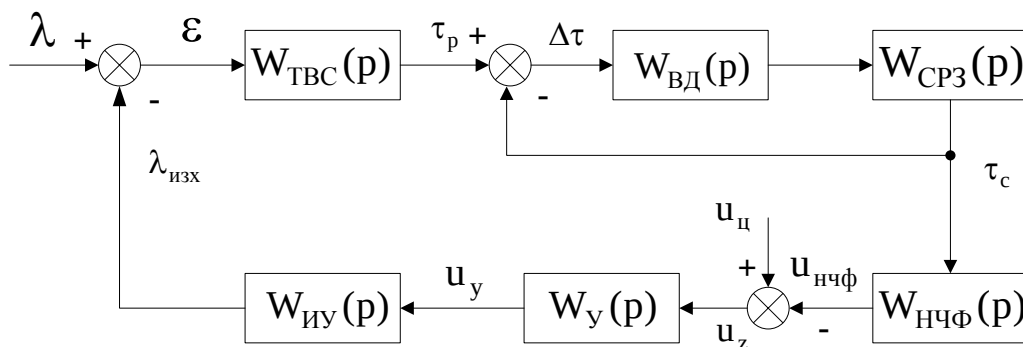
След изваждане на $u_{нчф}$ от напрежението $u_{ц}$, което определя началото на измервателната координатна система, се получава сигнал, чиято амплитуда и фаза зависят от отклонението на импулса на маркера от центъра на измервателната координатна система.

Това напрежение се усилва и постъпва в изпълнителното устройство, което завърта

ТВС, намираща се на БЛА, така че импулсът от маркера да съвпадне с началото на измервателната координатна система (постепенно да бъде преместен там от оператора в съответствие с преместването на изображението към центъра).

Разгледаният координатор е типичен представител на импулсните координатори и обработва информацията за определяне на координатите на целта в дискретни

моменти от времето. Както е известно, когато честотата на следване на информационните импулси е много по-голяма от лентата на пропускане на затворената система, то тя може да бъде линеаризирана и заменена с еквивалентна непрекъсната система. Това условие се изпълнява за разглеждания координатор, поради което структурната му схема може да бъде синтезирана като непрекъсната, както е показано на фиг. 2.



Фиг. 2 Структурна схема на телевизионен координатор

Предавателните функции на звената, образуващи координатора, могат да бъдат определени от следните съображения. ТВС може да бъде разглеждана като безинерционен преобразувател на ъгловото разсъгласуване ε във времеви интервали τ_p , вследствие на принципите, които са заложи в нея, т.е.

$$W_{TBC}(p) = k_{TBC} \cdot \quad (1)$$

Предавателната функция на временния детектор с един интегратор и нискочестотен филтър е:

$$W_{BD}(p) = \frac{k_{BD}}{p(T_{BD}p + 1)} \cdot \quad (2)$$

Схемата за регулиране на задръжката представлява практически безинерционно звено с предавателна функция:

$$W_{CP3}(p) = k_{CP3} \cdot \quad (3)$$

Нискочестотният филтър има предавателна функция:

$$W_{HCF}(p) = \frac{k_{HCF}}{T_{HCF}p + 1} \cdot \quad (4)$$

Изпълнителният двигател заедно с редуктора, независимо от неговата физическа природа, може да бъде описан с предавателна функция от типа:

$$W_{IU}(p) = \frac{k_{IU}}{p(T_{IU}p + 1)} \cdot \quad (5)$$

Обикновено усилвателят има достатъчно широка лента на пропускане и може да бъде разглеждан като безинерционно звено.

От формулите за предавателните функции от (1) до (5) се вижда, че координаторът в чист вид е структурно неустойчив. От практическа гледна точка корекцията, която осигурява структурна

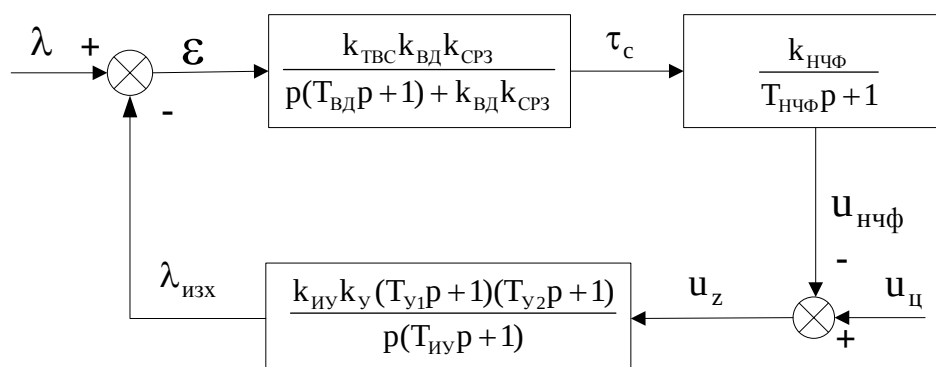
устойчивост с най-прости средства, може да бъде реализирана в усилвателя.

Поради това е желателно предавателната функция на усилвателя да бъде равна на:

$$W_y(p) = k_y(T_{y1}p + 1)(T_{y2}p + 1). \quad (6)$$

По този начин структурната схема на

телевизионния координатор на автоматизираната система за управление на БЛА от фиг. 2 може да бъде преобразувана във вида, показан на фиг. 3, като се използват предавателните функции от (1) до (6).



Фиг.3 Кorigирана структурна схема на телевизионен координатор

Заклучение

Предложената функционална схема на телевизионен координатор и алгоритъмът на работата му позволяват да се синтезира неговата структурна схема и да се определят предавателните функции на звената, които влизат в координатора. Тези схеми могат да бъдат използвани за анализ и синтез на телевизионни координатори със следящо стробиране на телевизионното изображение на целта.

Литература

- [1]. Janesick J., Klaasen K. and Elliott T. CCD charge collection efficiency and the photon

transfer technique" in Solid State Imaging Arrays, K. N. Prettyjohns and E. L. Dtrtniak, eds., Proc. SPIE 570, 7-19 (1985).

- [2]. UAV system, The Global Perspective 2006/2007. Paris: Blyenburg & Co.
- [3]. Ткаченко А.П., Кириллов В.И., Техника телевизионных измерений, Минск: Высшэйшая школа, 1976.

За контакти:

н.с. I ст. инж. Валентина Цекова
Институт за перспективни изследвания за
отбраната
БА „Г. С. Раковски”
E-mail: valsof20@mail.bg

АДАПТИВЕН НАБЛЮДАТЕЛ НА СЪСТОЯНИЕТО БАЗИРАН НА МЕТОДА НА НАЙ-МАЛКИТЕ КВАДРАТИ

Никола Н. Николов

Резюме: Представен е нерекурентен алгоритъм за адаптивно наблюдение на SISO линейни стационарни дискретни системи, разработен на базата на метода на най-малките квадрати. Адаптивният наблюдател на състоянието оценява параметрите, началният и текущият вектор на състоянието на дискретната система. Приложени са тестови изследвания със симулационни данни.

Adaptive State Observer based Least Squares

Nikola N. Nikolov

Abstract: This paper produces nonrecurrent adaptive observation algorithm for SISO linear stationary discret systems, based the least squares method. The adaptive state observer estimates the parameters, initial and the current state vector of the discret system. Test investigations with simulation data are applied.

Увод

При проектирането на системи за управление с обратна връзка по състоянието, много често се налага възстановяване на вектора на състоянието по измерванията на изходната величина и входното въздействие на управлявания обект.

Алгоритъмът чрез който се извършва определянето на вектора на състоянието се нарича *наблюдател на състоянието*.

Задачата за адаптивно наблюдение се състои в създаването на наблюдатели на състоянието, съдържащи в себе си оценители на параметри. При адаптивните наблюдатели, матриците **A** и **b** или **c** (в зависимост от избраната канонична форма за представяне в пространството на състоянията) се приемат за неизвестни. В процеса на наблюдение се извършва оценяване на параметрите, определят се неизвестните матрици и се изчислява вектора на състоянието.

Тук ще бъде представен един нерекурентен алгоритъм за адаптивно наблюдение на едномерни (SISO) линейни стационарни дискретни системи (ЛСДС), разработен на базата на метода на най-малките квадрати (МНМК).

Оценителят на параметрите, в адаптивния наблюдател, е изграден на базата на математическа процедура с ниска изчислителна сложност при обръщане на информационната матрица, която е представена в [4].

Постановка на задачата

Разглежда се едномерна ЛСДС описана в пространството на състоянията със следните уравнения

$$(1) \quad \begin{aligned} \mathbf{x}(k+1) &= \mathbf{A}\mathbf{x}(k) + \mathbf{b}u(k), \quad \mathbf{x}(0) = \mathbf{x}_0, \\ y(k) &= \mathbf{c}^T \mathbf{x}(k) + f(k), \quad k = 0, 1, 2, \dots \end{aligned}$$

където:

$$(2) \quad \mathbf{A} = \begin{bmatrix} \mathbf{0} & \vdots & \mathbf{I}_{n-1} \\ \dots & \dots & \dots \\ & \mathbf{a}^T & \end{bmatrix},$$

$$(3) \quad \mathbf{a} = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}, \quad \mathbf{c} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}.$$

Предварително е известен реда на системата n , $\mathbf{x}(k) \in \mathbb{R}^n$ е неизвестен вектор на текущото състояние, $\mathbf{x}(0) \in \mathbb{R}^n$ е неизвестен вектор на началното състояние, $u(k) \in \mathbb{R}^1$ е скаларен вход, $y(k) \in \mathbb{R}^1$ е скаларен изход, $f(k)$ е адитивен шум, **a** и **b** са неизвестни векторни параметри.

На описанието в пространството на състоянията, представено с (1), съответства дискретната предавателна функция

$$(4) \quad W(z) = \frac{h_1 z^{n-1} + h_2 z^{n-2} + \dots + h_{n-1} z + h_n}{z^n - a_n z^{n-1} - \dots - a_2 z - a_1}.$$

Съответствието между елементите b_i на вектора $\mathbf{b}$, при избраната фазовокоординатна канонична форма и коефициентите h_i на полинома в числителя на дискретната предавателна функция (4) се определя с израза [3]

$$(5) \quad \mathbf{T}\mathbf{b} = \mathbf{h},$$

където

$$\mathbf{h}^T = [h_1 \quad h_2 \quad \dots \quad h_n],$$

$$\mathbf{T} = \begin{bmatrix} 1 & 0 & \dots & 0 & 0 \\ -a_n & 1 & \dots & 0 & 0 \\ -a_{n-1} & -a_n & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ -a_2 & -a_3 & \dots & -a_n & 1 \end{bmatrix}.$$

Елементите a_i на вектора $\mathbf{a}$, съгласно избраната фазовокоординатна канонична форма, са коефициентите на полинома в знаменателя на дискретната предавателна функция (4), взети в обратен ред и с обратен знак.

Задачата е да се оценят неизвестните векторни параметри $\mathbf{a}$ и $\mathbf{b}$, началният вектор на състоянието $\mathbf{x}(0)$ и текущият вектор на състоянието $\mathbf{x}(k)$, $k=1, 2, \dots$

Алгоритъм за адаптивно наблюдение базиран на МНМК

Стъпка 1 Формират се масиви от входно-изходни данни [1,2]:

$$\mathbf{u}_1 = [u(0) \quad u(1) \quad \dots \quad u(N-2)],$$

$$\mathbf{y}_1 = [y(0) \quad y(1) \quad \dots \quad y(N-1)],$$

$$\mathbf{y}_2 = \left[y(n) \quad y(n+1) \quad \dots \quad y\left(\frac{N-n}{2} + n-1\right) \right]^T,$$

$$\mathbf{y}_3 = \left[y\left(\frac{N-n}{2} + n\right) \quad y\left(\frac{N-n}{2} + n+1\right) \quad \dots \quad y(N-1) \right]^T,$$

$$\mathbf{Y}_{11} = \begin{bmatrix} -y(n-1) & -y(n-2) & \dots & -y(0) \\ -y(n) & -y(n-1) & \dots & -y(1) \\ -y(n+1) & -y(n) & \dots & -y(2) \\ \vdots & \vdots & \ddots & \vdots \\ -y\left(\frac{N-n}{2} + n-2\right) & -y\left(\frac{N-n}{2} + n-3\right) & \dots & -y\left(\frac{N-n}{2} - 1\right) \end{bmatrix}$$

$$\mathbf{Y}_{21} = \begin{bmatrix} -y\left(\frac{N-n}{2} + n-1\right) & -y\left(\frac{N-n}{2} + n-2\right) & \dots & -y\left(\frac{N-n}{2}\right) \\ -y\left(\frac{N-n}{2} + n\right) & -y\left(\frac{N-n}{2} + n-1\right) & \dots & -y\left(\frac{N-n}{2} + 1\right) \\ -y\left(\frac{N-n}{2} + n+1\right) & -y\left(\frac{N-n}{2} + n\right) & \dots & -y\left(\frac{N-n}{2} + 2\right) \\ \vdots & \vdots & \ddots & \vdots \\ -y(N-2) & -y(N-3) & \dots & -y(N-n-1) \end{bmatrix}$$

$$\mathbf{U}_{12} = \begin{bmatrix} u(n-1) & u(n-2) & \dots & u(0) \\ u(n) & u(n-1) & \dots & u(1) \\ u(n+1) & u(n) & \dots & u(2) \\ \vdots & \vdots & \ddots & \vdots \\ u\left(\frac{N-n}{2} + n-2\right) & u\left(\frac{N-n}{2} + n-3\right) & \dots & u\left(\frac{N-n}{2} - 1\right) \end{bmatrix},$$

$$\mathbf{U}_{22} = \begin{bmatrix} u\left(\frac{N-n}{2} + n-1\right) & u\left(\frac{N-n}{2} + n-2\right) & \dots & u\left(\frac{N-n}{2}\right) \\ u\left(\frac{N-n}{2} + n\right) & u\left(\frac{N-n}{2} + n-1\right) & \dots & u\left(\frac{N-n}{2} + 1\right) \\ u\left(\frac{N-n}{2} + n+1\right) & u\left(\frac{N-n}{2} + n\right) & \dots & u\left(\frac{N-n}{2} + 2\right) \\ \vdots & \vdots & \ddots & \vdots \\ u(N-2) & u(N-3) & \dots & u(N-n-1) \end{bmatrix}$$

където

$\mathbf{Y}_{11}$, $\mathbf{Y}_{21}$, $\mathbf{U}_{12}$ и $\mathbf{U}_{22}$ са матрици на Toeplitz, а $N=3n+2l$, $l=1, 2, 3, \dots$

Стъпка 2 Пресмятат се субматриците:

$$\mathbf{G}_{11} = \mathbf{Y}_{11}^T \mathbf{Y}_{11} + \mathbf{Y}_{21}^T \mathbf{Y}_{21},$$

$$\mathbf{G}_{12} = \mathbf{Y}_{11}^T \mathbf{U}_{12} + \mathbf{Y}_{21}^T \mathbf{U}_{22},$$

$$\mathbf{G}_{21} = \mathbf{U}_{12}^T \mathbf{Y}_{11} + \mathbf{U}_{22}^T \mathbf{Y}_{21},$$

$$\mathbf{G}_{22} = \mathbf{U}_{12}^T \mathbf{U}_{12} + \mathbf{U}_{22}^T \mathbf{U}_{22}.$$

Стъпка 3 Пресмята се ковариационната матрица $\mathbf{C}$

$$C = \begin{bmatrix} \mathbf{M}_1 + \mathbf{M}_1 \mathbf{G}_{12} \mathbf{M}_2 \mathbf{G}_{21} \mathbf{M}_1 & \vdots & -\mathbf{M}_1 \mathbf{G}_{12} \mathbf{M}_2 \\ \dots\dots\dots & \dots & \dots\dots\dots \\ -\mathbf{M}_2 \mathbf{G}_{21} \mathbf{M}_1 & \vdots & \mathbf{M}_2 \end{bmatrix} \quad D = \begin{bmatrix} \mathbf{c}^\top \\ \mathbf{c}^\top \hat{\mathbf{A}} \\ \mathbf{c}^\top \hat{\mathbf{A}}^2 \\ \vdots \\ \mathbf{c}^\top \hat{\mathbf{A}}^{(N-1)} \end{bmatrix}_{(N \times n)},$$

където

$$\mathbf{M}_1 = \mathbf{G}_{11}^{-1},$$

$$\mathbf{M}_2 = (\mathbf{G}_{22} - \mathbf{G}_{21} \mathbf{M}_1 \mathbf{G}_{12})^{-1}.$$

Стъпка 4 Пресмятат се векторите $\hat{\mathbf{h}}$ и $\hat{\mathbf{a}}$ с помощта на векторно-матричната система

$$\hat{\mathbf{p}} = \mathbf{C} \begin{bmatrix} \mathbf{Y}_{11}^\top \mathbf{y}_2 + \mathbf{Y}_{21}^\top \mathbf{y}_3 \\ \mathbf{U}_{12}^\top \mathbf{y}_2 + \mathbf{U}_{22}^\top \mathbf{y}_3 \end{bmatrix}.$$

$$\hat{\mathbf{h}} = [\hat{h}_1 \quad \hat{h}_2 \quad \dots \quad \hat{h}_n]^\top = [\hat{p}_{n+1} \quad \hat{p}_{n+2} \quad \dots \quad \hat{p}_{2n}]^\top$$

,

$$\hat{\mathbf{a}} = [\hat{a}_1 \quad \hat{a}_2 \quad \dots \quad \hat{a}_n]^\top = [-\hat{p}_n \quad -\hat{p}_{n-1} \quad \dots \quad -\hat{p}_1]^\top$$

,

$$\hat{\mathbf{A}} = \begin{bmatrix} \mathbf{0} & \vdots & \mathbf{I}_{n-1} \\ \dots & \dots & \dots \\ & \hat{\mathbf{a}}^\top & \end{bmatrix}.$$

Стъпка 5 Пресмята се оценката на вектора $\mathbf{b}$ с помощта на следната система от линейни алгебрични уравнения

$$\mathbf{T} \hat{\mathbf{b}} = \hat{\mathbf{h}},$$

$$\text{където } \mathbf{T} = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 & 0 \\ -\hat{a}_n & 1 & 0 & \dots & 0 & 0 \\ -\hat{a}_{n-1} & -\hat{a}_n & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ -\hat{a}_2 & -\hat{a}_3 & -\hat{a}_4 & \dots & -\hat{a}_n & 1 \end{bmatrix}$$

е долно-триъгълна матрица на Toeplitz.

Стъпка 6 Оценява се векторът на началното състояние $\mathbf{x}_0$

$$\hat{\mathbf{x}}_0 = (\mathbf{D}^\top \mathbf{D})^{-1} \mathbf{D}^\top (\mathbf{y}_1 - \mathbf{Q} \mathbf{u}_1) = [\hat{x}_{01} \quad \hat{x}_{02} \quad \dots \quad \hat{x}_{0n}]^\top$$

където

$$\mathbf{Q} = \begin{bmatrix} 0 & 0 & \dots & 0 \\ \mathbf{c}^\top \hat{\mathbf{b}} & 0 & \dots & 0 \\ \mathbf{c}^\top \hat{\mathbf{A}} \hat{\mathbf{b}} & \mathbf{c}^\top \hat{\mathbf{b}} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{c}^\top \hat{\mathbf{A}}^{(N-2)} \hat{\mathbf{b}} & \mathbf{c}^\top \hat{\mathbf{A}}^{(N-3)} \hat{\mathbf{b}} & \dots & \mathbf{c}^\top \hat{\mathbf{b}} \end{bmatrix}_{(N \times (N-1))}.$$

Стъпка 7 Оценява се векторът на текущото състояние $\mathbf{x}(k)$:

$$\hat{\mathbf{x}}(k+1) = \hat{\mathbf{F}} \hat{\mathbf{x}}(k) + \hat{\mathbf{b}} u(k) + \mathbf{g} y(k), \quad \hat{\mathbf{x}}(0) = \hat{\mathbf{x}}_0;$$

$$\hat{\mathbf{F}} = \hat{\mathbf{A}} - \mathbf{g} \mathbf{c}^\top.$$

Векторът $\mathbf{g}$ може да се получи чрез синтез по зададени полюси. За предпочитане е $\mathbf{g}$ да се синтезира така, че матрицата $\hat{\mathbf{F}}$ да има нулеви собствени стойности или поне такива които са разположени по-навътре, в единичният кръг, от собствените стойности на $\hat{\mathbf{A}}$.

Тестване на алгоритъма

Компютърният експеримент е проведен в средата на MATLAB при следните условия:

- ❖ при зададено описание на обекта за наблюдение и входно въздействие $u(k)$, се симулира изходната му реакция $y(k)$;
- ❖ към изхода на обекта се добавя цветен шум $f(k)$;
- ❖ входното въздействие $u(k)$ и зашумената изходна реакция $y(k)$ се „подават“ на алгоритъма за наблюдение;
- ❖ на базата на входно-изходните последователности от данни, алгоритъма изчислява оценките на параметрите на обекта и вектора на състоянието.

Компютърната симулация се извършва с обект от шести ред чиято дискретна предавателна функция е

$$W_{oo}(z) = \frac{0.6z^{-1} + 0.56z^{-2} + 0.2125z^{-3} + 0.3080z^{-4} + 0.5488z^{-5} + 0.7221z^{-6}}{1 - 1.4z^{-1} + 0.7875z^{-2} - 0.2275z^{-3} + 0.035525z^{-4} - 0.002835z^{-5} + 0.00009z^{-6}}$$

а съответстващите ѝ вектори за описание в пространството на състоянията са:

$$\mathbf{a} = \begin{bmatrix} -0.00009 \\ 0.002835 \\ -0.035525 \\ 0.2275 \\ -0.7875 \\ 1.4 \end{bmatrix}; \mathbf{b} = \begin{bmatrix} 0.6 \\ 0.2 \\ 0.1 \\ 0.3 \\ 0.4 \\ 0.5 \end{bmatrix}; \mathbf{c} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix};$$

$$\mathbf{x}(0) = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}.$$

Собствените стойности на системната матрица **A** са

$$\text{eig}(\mathbf{A}) = [0.4 \quad 0.3 \quad 0.25 \quad 0.2 \quad 0.15 \quad 0.1]^T.$$

За входно въздействие $u(k)$ се използва псевдо-случайна двоична последователност (ПСДП) генерирана, в MATLAB, с израза $u = (\text{sign}(\text{randn}(127,1))) * 10$. Към симулираната изходна реакция на обекта се добавя цветен шум $f(k)$. Той е получен чрез филтриране на бял шум през филтър с предавателна функция от вида:

$$W_f(z) = \frac{1}{1 - 1.4z^{-1} + 0.7875z^{-2} - 0.2275z^{-3} + 0.035525z^{-4} - 0.002835z^{-5} + 0.00009z^{-6}}$$

Нивото на шума- η се изчислява като отношение на стандартното отклонение на шума- σ_f към стандартното отклонение на сигнала- σ_y по формулата

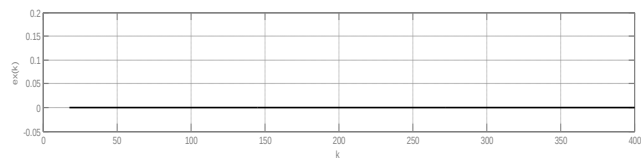
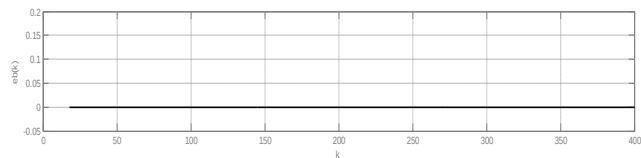
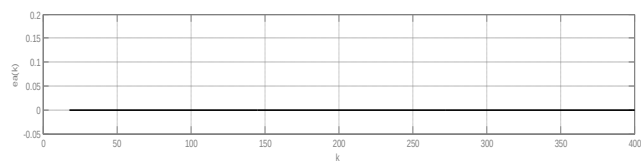
$$(6) \quad \eta = \frac{\sigma_f}{\sigma_y} 100 = 0 \div 10\%.$$

Грешката e_a при оценяване на вектора **a**, грешката e_b при оценяване на вектора **b** и грешката e_x при оценяване на вектора на състоянието $\mathbf{x}(k)$, са относителни средноквадратични грешки и се изчисляват по формулите:

$$(7) \quad e_a(k) = - \sqrt{\frac{\sum_{i=1}^n (a_i(k) - \hat{a}_i(k))^2}{\sum_{i=1}^n a_i(k)}},$$

$$(8) \quad e_b(k) = - \sqrt{\frac{\sum_{i=1}^n (b_i(k) - \hat{b}_i(k))^2}{\sum_{i=1}^n b_i(k)}},$$

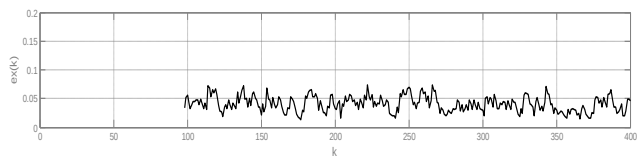
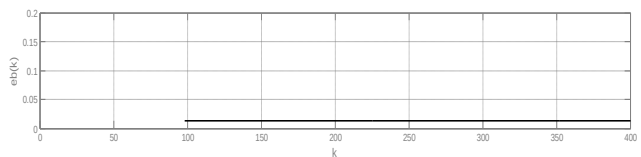
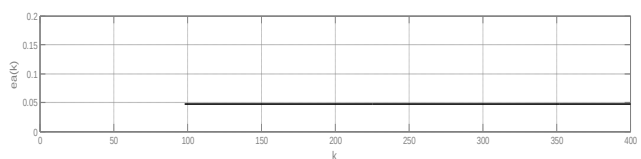
$$(9) \quad e_x(k) = - \sqrt{\frac{\sum_{i=1}^n (x_i(k) - \hat{x}_i(k))^2}{\sum_{i=1}^n x_i(k)}}.$$



фиг.1

При незашумен изход (т.е. при $f(k)=0$) и $l=0$ (т.е. при минимален брой на входно-изходните измервания - $N=3n=18$) се получават резултатите показани на фиг.1.

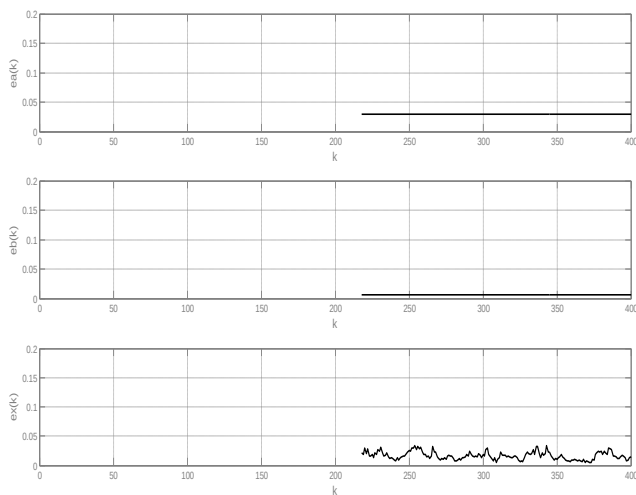
От фиг.1 е видно, че алгоритъмът започва да работи на 18-я такт, а грешките от наблюдението $e_a(k)$, $e_b(k)$ и $e_x(k)$ имат стойност нула.



фиг.2

При зашумен изход с $\eta=10.018$ и $l=40$ (т.е. при $N=3n+2l=98$) се получават резултатите показани на *фиг.2*. Алгоритъмът започва да работи на 98-я такт, а грешките от наблюдението са: $e_a(k)<0.053$, $e_b(k)<0.014$, $e_x(k)<0.081$.

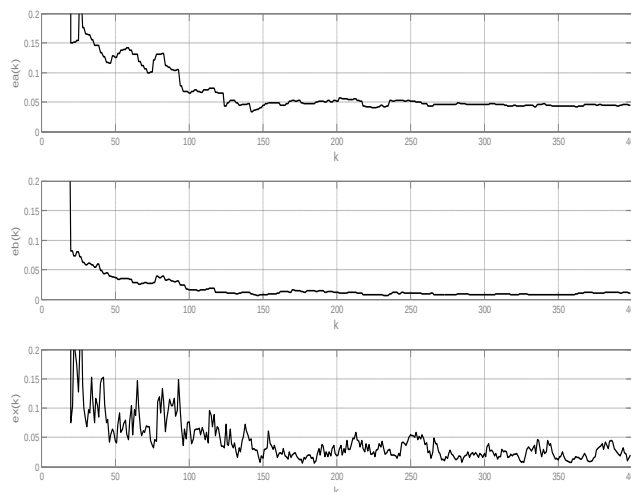
При зашумен изход с $\eta=10.018$ и $l=100$ (т.е. при $N=3n+2l=218$) се получават резултатите показани на *фиг.3*. Алгоритъмът започва да работи на 218-я такт, а грешките от наблюдението са: $e_a(k)<0.041$, $e_b(k)<0.01$, $e_x(k)<0.055$.



фиг.3

Видно е, че увеличаването на входно-изходните измервания- N води до нарастване на шумоустойчивостта на алгоритма (грешките от наблюдението намаляват), но времето за набиране на първоначалната информация става по-голямо. За да се отстрани това неудобство до известна степен, е възможно оценяването да започне при минимален брой на $N=3n$, а в последствие при постъпването на нови входно-изходни измервания непрекъснато да нараства броят на редовете на информационната матрица.

Резултатите от такъв компютърен експеримент са показани на *фиг.4*. След 150-я такт (т.е. при $N>150$, $l>66$), за грешките от наблюдението се получава $e_a(k)<0.056$, $e_b(k)<0.015$, $e_x(k)<0.059$. В края на процеса $l=191$.



фиг.4

Заклучение

От *фиг.2*, *фиг.3* и *фиг.4* много ясно се вижда, че броят на входно-изходните данни (N) има голямо значение за точността на оценяването при зашумен изход.

Достойнството на алгоритма обаче, е в начина на формирането на информационната матрица. Тук тя се формира чрез четирите подматрици Y_{11} , Y_{21} , U_{12} , U_{22} . По този начин се намалява рязко изчислителната сложност на процедурата по обръщането на матрицата G , съставена от подматриците G_{11} , G_{12} , G_{21} , G_{22} . Независимо от това колко е броят- N , на входно-изходните измервания, за оценяване на коефициентите h_i и a_i се обръщат само матриците G_{11} и $(G_{22}-G_{21}M_1G_{12})$, които винаги са с размери $(n \times n)$. При всички други случаи тази процедура е свързана с обръщането на матрица с размери поне $(N-n) \times (N-n)$.

За да бъде алгоритъмът шумоустойчив е необходимо $l>1$, а колко да нараства N би трябвало да се определя от възможностите на хардуера, с който се разполага. След като се достигне $N_{\max}$ на входно-изходните измервания, “прозорецът” от данни трябва да започне да се премества на всеки такт като отдясно се добавя новото измерване, а най-лявото отпада.

Предложеният алгоритъм за адаптивно наблюдение е най-добре да бъде използван при ниво на шума до 5%, тъй като тогава грешките са със стойности под 0.005. При високи нива на шума, за да се повиши

точността на оценяването е необходимо МНМК да бъде комбиниран с други методи за оценяване.

Тук се използва пряк (нерекурентен) метод за оценяване така, че няма проблеми относно сходимостта на процедурата.

Литература

- [1] Söderström T, P. Stoica, System identification, Prentice Hall, New Jersey, Englewood Cliffs, 1989.
- [2] Вучков И., Идентификация, София, ИК Юрапел, 1996.
- [3] Сотиров Л.Н., Теория на автоматичното управление – Част II, ТУ-Варна, 2000.
- [4] Сотиров Л. Н., В. С. Димитров, Н. Н. Николов, Дискретен адаптивен наблюдател на състояние за наблюдение в реално време, Международна конференция “Автоматика и информатика ’04”, Сборник доклади, стр.121-124, София, 2004.

За контакти:

гл.ас. инж. Никола Н. Николов
ТУ – Варна, кат. АП
Ул. Студентска №1
Тел. (052) 383-391
e-mail: nn_nikolov@abv.bg

Equation-format - 10-5-4-15-10

ELICITATION ERRORS IN ONE-DIMENSIONAL NON-MONOTONIC UTILITY FUNCTIONS

Natalia D. Nikolova

Abstract: Quasi-multimodal utility functions apply to the preferences of a fuzzy rational decision maker (one that disobeys some of the axioms of rational preference) that have several extrema within the prize range. The construction of quasi-multimodal utility functions requires finding the extreme intervals, building local utility functions in the sections with monotonic preferences (between the extrema), and then rescaling to the global utility. Local utilities are here arctg-approximated/interpolated on a set of nodes containing error either on the prize or on the utility. The utilities at the extrema (which are the reference points in the rescaling phase) are also in an interval form. The paper investigates the ways to rescale uncertainty intervals of monotonic utilities according to interval reference points. The proposed procedures are embodied into an algorithm for the construction of a quasi-multimodal utility function of a fuzzy rational decision maker. Numerical examples demonstrate its application.

Keywords: utility, non-monotonic preferences, rescaling, fuzzy rationality, interval estimates

Грешки в оценяването при построяване на едномерни немонотонни функции на полезност

Наталия Д. Николова

Резюме: Псевдо-мултимодалните функции на полезност представят предпочитанията на размито-рационален вземащ решения (който не се подчинява на част от аксиомите за рационалност), които имат няколко екстремума в множеството на наградите. За да се построи псевдо-мултимодална функция на полезност е нужно да се оценят екстремумите, да се построят локални функции на полезност в секторите с монотонни предпочитания (между екстремумите), които след това да се рескалират в глобална функция на полезност. Тук, локалните функции на полезност са arctg-апроксимирани по няколко възела, съдържащи грешка или по полезност или по наградата. Полезностите на наградите при екстремумите (които са реферативни точки при рескалирането) също са в интервална форма. Статията разглежда начините за рескалиране интервалите на неопределеност на монотонни функции на полезност на база на интервални реферативни точки. Предложените процедури са обединени в алгоритъм за построяването на псевдо-мултимодална функция на полезност на размито-рационален вземащ решения. Неговото приложение е демонстрирано в числени примери.

Ключови думи: полезност, немонотонни предпочитания, рескалиране, размита рационалност, интервални оценки

1. Introduction

Quantitative decision analysis uses measures of the expectations, preferences, and risk attitude of the decision maker (DM). With the help of these it is possible to identify, on the basis of calculations, the alternative that is most preferred for a specific DM. Preferences in particular are measured by the *utility function* $u(\cdot)$, constructed over the set of consequences. This function is constructed on the basis of subjectively elicited nodes, and its shape gives

a great deal of information regarding the risk profile of the individual (whether he is risk prone, risk averse or risk neutral).

Constructing the utility function is one of the main problems in utility theory [von Neumann, Morgenstern, 1947], which applies to problems under risk. It models alternatives are lotteries (a set of prizes along with their probabilities). The basis of the utility theory are several axioms of rational choice over lotteries [French, Insua, 2000], which also help elaborate utility elicitation techniques. They

envisage solving a preferential equation between a lottery and/or a prize during a dialog with the DM. The ideal DM obeys the rationality axioms and elicits unique point estimates of utilities. The real DM has finite discriminating abilities and elicits an uncertainty interval of estimates. As a result he partially disobeys some of the rationality assumptions and is referred to as fuzzy rational DM (FRDM) [Nikolova, et al., 2005].

In a classical setup, the utility function is constructed over a one-dimensional continuous set of prizes, e.g. monetary prizes (profits). Most typical are monotonically increasing preferences, where the increase of the prize leads to an increase in preference. Monotonically decreasing preferences on the other hand assume that the increase of the prize leads to a decrease in preference (e.g. interest rates, prices, etc.). The literature abounds of methods to elicit nodes of a monotonic utility function, such as the probability equivalence (PE) [Clemen, 1996], certainty equivalence (CE) [Farquhar, 1984], and lottery equivalence (LE) [McCord, De Neufville, 1986]. Modern techniques are the trade-off (TO) [Wakker, Deneffe, 1996], and the uncertain equivalence (UE) [Tenekedjiev, et al., 2006] methods. At a next stage, the elicited nodes are used to approximate or interpolate the utility. Approximation applies when there are only few elicited nodes and/or if their uncertainty intervals are too wide. The objective is to find an analytical form depending on the risk attitude of the DM and the type of prizes [Keeney, Raiffa, 1993].

There are sets of prizes, over which the DM expresses non-monotonic preferences, e.g. body weight, temperature, etc. This means that the most preferred or the least preferred prizes are not at the bounds of the prize range, but within it. In [Nikolova et al., 2006] such utility functions are called hill/valley quasi-unimodal (depending on whether the best or the worst prize is within the interval). In order to construct such a utility function, the DM should first find the extrema, then construct local utility functions in the sections with monotonic preferences and then rescale them to a global utility. Due to fuzzy rationality, the extreme

point is actually an extreme interval, whereas the local functions are constructed by approximation/interpolation on interval nodes (with errors on the prize or on the utility depending on the elicitation method applied). Four procedures are presented in [Nikolova et al., 2006] to elicit the hill/valley extreme interval (using bisection and golden-section search), and construct/rescale the local utilities.

Yet it is likely that in some cases there would be several extrema in the preferences of the DM, thus quasi-multimodal utility function needs to be constructed. The paper focuses on that problem by extending the algorithms and procedures from [Nikolova et al., 2006] to the case of a function with multiple extrema. It mostly focuses on the rescaling procedure, since local utilities are constructed on interval nodes, and the rescaling parameters (the utilities at the extrema) are also in an interval form. This requires approaches different from those discussed in [Nikolova et al., 2005] depending on whether the initial nodes for approximation have intervals on the prize or on the utility. All steps of this approach are combined in a generic algorithm. Numerical examples demonstrate its application.

In what follows, section 2 presents the theoretical setup of monotonic and non-monotonic utility functions, whereas section 3 formalizes the rescaling procedures of interval nodes. The generic algorithm for construction of quasi-multimodal utilities is presented in section 4, and illustrations of its application are given in section 5.

2. Monotonic and non-monotonic one-dimensional utilities

A) The monotonic case

A classical task in decision analysis is to build a utility function over a one-dimensional set of prizes X . The function is constructed in the interval $[x_{worst}; x_{best}]$, where $x_{best} = \sup(X)$, $x_{worst} = \inf(X)$.

Monotonically increasing preferences obey the following rule:

$$x_i \succ x_j \Leftrightarrow x_i > x_j, x_i \in X, x_j \in X. \quad (1)$$

Here, $\succ$ stands for the binary relation “more preferred than”. It is reasonable to elicit only several (z) nodes and construct the utility function using approximation/interpolation. The utility elicitation techniques solve preferential equations by changing one parameter until compared options (prizes and/or prizes) become indifferent [Farquhar, 1984]. Assume that a FRDM has elicited $z-2$ inner nodes of $u(\cdot)$ with coordinates $(x_{u_l}; u_l)$, $l=2, 3, \dots, z-1$ (x_{u_l} and u_l are respectively an utility quantile and an utility quantile index). The end nodes are known: $(x_{u_1}; u_1)=(x_{worst}; 0)$, $(x_{u_z}; u_z)=(x_{best}; 1)$. Methods like PE or LE select several quantiles x_{u_l} ($l=2, 3, \dots, z-1$) and elicit the corresponding $\hat{u}_l$. The FRDM elicits uncertainty intervals of the form $\hat{u}_l \in [\hat{u}_l^d; \hat{u}_l^u]$ ($l=2, 3, \dots, z-1$), where $\hat{u}_l^d$ and $\hat{u}_l^u$ are the lower and upper bounds of the uncertainty interval of $\hat{u}_l$. Other methods, like CE or UE select several utility quantiles u_l ($l=2, 3, \dots, z-1$) and elicit their corresponding $\hat{x}_{u_l}$. The FRDM elicits uncertainty intervals of the form $\hat{x}_{u_l} \in [\hat{x}_{u_l}^d; \hat{x}_{u_l}^u]$ ($l=2, 3, \dots, z-1$), where $\hat{x}_{u_l}^d$ and $\hat{x}_{u_l}^u$ are the lower and upper bound of the uncertainty interval of $\hat{x}_{u_l}$.

Monotonically decreasing preferences assume that

$$x_i \succ x_j \Leftrightarrow x_i < x_j, x_i \in X, x_j \in X. \quad (2)$$

Again it is necessary to elicit $z-2$ inner nodes of $u(\cdot)$ with coordinates $(x_{u_l}; u_l)$, $l=2, 3, \dots, z-1$, but here the end nodes are $(x_{u_1}; u_1)=(x_{best}; 1)$, $(x_{u_z}; u_z)=(x_{worst}; 0)$. The elicitation procedures follow the same principles as before, and the results are again uncertainty intervals of either form.

The next step in the analysis is to approximate/interpolate the utility function. Approximation applies when only few nodes are available or when their uncertainty intervals are too wide. The analytical form should be chosen so that to interpret correctly the data and the preferences of the DM. It should also

preserve his risk attitude [French, 1993], modeled by the local risk aversion function $r(x)=-u''(x)/u'(x)$ [Pratt, 1964]. Different forms applying for different types of risk attitude (decreasing risk aversion, decreasing risk proneness, constant risk aversion, etc.) and different prize ranges (monetary gains and losses are the standard setup for demonstration) are presented in [Keeney, Raiffa, 1993]. The work [Stoyanov, 1993] suggests using the Harrington's desirability function

$$u(x) \sim_p e^{-e^{-b-ax}} \quad (\text{here } \sim_p \text{ denotes proportionality}).$$

This dependence generates a function with a typical local risk aversion and suits to problems where consequences are either gains or losses.

Due to fuzzy rationality a problem arises of how to find estimates of the suitable parameters taking into account the uncertainty intervals of the nodes (which the classical least square method [Abdi, 2003] does not do). Therefore, [Tenekedjiev, Nikolova, 2007] discusses the use of the weighted least square method [Rao et al., 1999; Press et al., 1992] to find the unknown parameters of the analytical function. This method weighs the deviation of the model from the best subjective point estimate in a given node by the width of the uncertainty interval of the utility quantile (or utility quantile index). A goodness-of-fit χ^2 measure of the model to the data is proposed in the same source, as well as the analytical form

$$u(x) = \frac{\arctg[a(x-x_0)] - \arctg[a(x_1-x_0)]}{\arctg[a(x_z-x_0)] - \arctg[a(x_1-x_0)]}. \quad (3)$$

Here, a measures risk sensitivity, whereas $x_{0,opt}$ measures the inflex point of the function. The local risk aversion of (3) interprets the typical risk attitude of a DM, corresponding to the typical utility function [Clemen, 1996]. The utility function (3) has an analytical inverse function:

$$\begin{aligned} x(u) = & \frac{\text{tg}\{u \arctg[a(x_z-x_0)] + \\ & + (1-u) \arctg[a(x_1-x_0)]\}}{a} + x_0 \end{aligned} \quad (4)$$

The advantage of the form (3) over other analytical forms is justified in [Nikolova, 2007; Tenekedjiev, et al., 2007].

B) The non-monotonic case

Another situation arises when the DM has quasi-unimodal preferences, i.e. when there is a value x_{opt} with extreme utility (either a minimum or a maximum) within the interval of prizes [Nikolova et al., 2006]. There are two types of quasi-unimodal preferences – hill (with a maximum extremum) and valley (with a minimum extremum) preferences. Both occur due to two contradicting factors related to the analyzed variable. Difficulties arise when the DM compares values on both sides of the extreme interval – if a sufficient difference in utilities exists, the DM would be able to state preference otherwise he would be indifferent being unable to compare the options. This leads to mutual non-transitivity of preferences and even a very motivated and rational DM is not in position to define other than interval estimates of the extreme point. Since all elicitation techniques need the reference points (the prizes with extreme utility), identifying the extreme interval is mandatory.

The models of hill and valley utility functions are based on two separate sets of assumptions that refer to the discriminating abilities of the FRDM and the characteristics of the extreme interval. Two 20-step algorithms are elaborated in [Nikolova et al., 2006] (one per each type of quasi-unimodal preferences). They find the extreme interval via a dialog with the FRDM using golden section search [Kiefer, 1953] and bisection [Press, et al., 1992] methods. Golden section serves to locate the extreme interval, whereas bisection estimates its lower and upper bounds. As argued in [Nikolova et al., 2006] in an algebraic case of one-dimensional optimization, the Kiefer-Johnson method [Forsythe, et al., 1977] is probably the best, followed by golden section search and bisection. When preferential equations are solved, what matters is the number of comparisons. That is why the Kiefer-Johnson approach is inapplicable, whereas bisection is more effective than golden section search in terms of reduction of the

interval after each comparison. However it requires comparison of close prizes that is likely to generate biased estimates. That is why although less effective, the golden section search is more appropriate as it generates less biased results.

After the extreme interval is identified, the next step is to construct local utility functions in the sections with monotonic preferences. Finally, it is possible to construct the global utility function over the entire set of prizes by rescaling the local functions. Two other algorithms are proposed for that purpose in [Nikolova et al., 2006]. However, the discussion in that work only focuses on the case where there is a single extremum within the prize interval. Furthermore, the extreme prizes are only assigned point estimate utilities, which are later used in the rescaling process.

It is possible to observe preferences, where there is more than one extremum within the prize interval. Furthermore, as earlier discussed FRDMs only identify uncertainty intervals of utilities, thus the rescaling of the local utilities should be made according to interval estimates. It is necessary to investigate how the initial uncertainty in the elicited nodes combines with the uncertainty interval of the extreme utilities and what this combination leads to.

3. Rescaling monotonic functions constructed on interval nodes

The construction of quasi-multimodal utility function requires to rescale several local utilities, constructed in sections with monotonic (increasing or decreasing) preferences. The work [Nikolova et al., 2006] proposed two rescaling algorithms, which refer to the case of hill and valley quasi-unimodal functions. However, these two algorithms work only with point estimates of both the initially elicited nodes and of the reference points for rescaling. If the quasi-multimodal function of a FRDM is involved, all nodes for approximation will have an interval form, and the utilities at the reference points (the extrema or at x_{min} and x_{max}) for approximation, whose utility is different than 1 or 0 will have interval form.

That imposes different considerations in the rescaling process.

The scheme of utility elicitation assumes that a prize x_l is compared to a reference lottery $\langle x_{best} (p) x_{worst} \rangle$, and at the moment of indifference the utility of $x_l - u(x_l) = u_l$ – coincides with the probability p in the reference lottery:

$$x_l \sim \langle x_{best} (u_l) x_{worst} \rangle. \quad (5)$$

According to the expected utility rule

$$u_l = u(x_{best})u_l + u(x_{worst})(1 - u_l). \quad (6)$$

However, in the quasi-multimodal case, the utilities at some of the extreme points (in the extreme intervals, or at x_{min} and x_{max}) will be neither 1 nor 0, and will have interval form, which affects the rescaling of the initial nodes for approximation.

Assume that x_{min} and x_{max} are the boundaries of the prize range, whereas $x_{opt}^i \in [\hat{x}_{opt min}^i; \hat{x}_{opt max}^i]$ ($i=1, 2, \dots, m$) are the uncertainty intervals of the extrema. Then it is possible to divide the interval of prizes into m number of sections ($i=0, 1, \dots, m$) with monotonically increasing/decreasing preferences $[x_{min}; \hat{x}_{opt min}^1]$, $[\hat{x}_{opt max}^1; \hat{x}_{opt min}^2]$, $\dots$, $[\hat{x}_{opt max}^i; \hat{x}_{opt min}^{i+1}]$, $\dots$, $[\hat{x}_{opt max}^m; x_{max}]$ (the zero section being $[x_{min}; \hat{x}_{opt min}^1]$).

Let's analyze the i -th monotonic section $[\hat{x}_{opt max}^i; \hat{x}_{opt min}^{i+1}]$. If preferences are monotonically increasing, then $\hat{x}_{opt max}^i \equiv x_{worst,i}$, $\hat{x}_{opt min}^{i+1} \equiv x_{best,i}$. If preferences are monotonically decreasing, then $\hat{x}_{opt max}^i \equiv x_{best,i}$, $\hat{x}_{opt min}^{i+1} \equiv x_{worst,i}$ (and are the local best and worst prize). The FRDM has elicited that $u(x_{worst,i}) \equiv u_{worst,i} \in [\hat{u}_{worst,i}^d; \hat{u}_{worst,i}^u]$, $u(x_{best,i}) \equiv u_{best,i} \in [\hat{u}_{best,i}^d; \hat{u}_{best,i}^u]$.

Assume that the inner nodes for approximation have the form $\{(\hat{x}_{i,\mu_l}^d; \hat{x}_{i,\mu_l}^u; u_{i,l}) \mid l=2, 3, \dots, z_i-1\}$ (where z_i is the total number of elicited nodes in the i -th section). So the utilities $u_{i,l}$ need to be rescaled to the utilities $w_{i,l}$ according to $u_{worst,i}$ and $u_{best,i}$, thus

transforming from point estimates to uncertainty intervals. Following (6) it is obvious that since $u_{i,l}$ and $(1 - u_{i,l})$ are positive, then minimization of $w_{i,l}$ would be possible by setting $u_{worst,i}$ and $u_{best,i}$ to their lowest levels. In the same fashion, maximizing $w_{i,l}$ would be possible by setting $u_{worst,i}$ and $u_{best,i}$ to their highest levels. That results to the following:

$$\begin{cases} w_{i,l} \in [\hat{w}_{i,l}^d; \hat{w}_{i,l}^u], \text{ where} \\ w_{i,l}^d = \hat{u}_{best,i}^d u_{i,l} + \hat{u}_{worst,i}^d (1 - u_{i,l}), \\ w_{i,l}^u = \hat{u}_{best,i}^u u_{i,l} + \hat{u}_{worst,i}^u (1 - u_{i,l}). \end{cases} \quad (7)$$

The resulting nodes of the rescaled utility function $w_i(\cdot)$ thus have errors (uncertainty intervals) on both coordinates and may be formalized as $\{(\hat{x}_{i,\mu_l}^d; \hat{x}_{i,\mu_l}^u; \hat{w}_{i,l}^d; \hat{w}_{i,l}^u) \mid l=2, 3, \dots, z_i-1\}$. An appropriate point estimate of $w_{i,l}$ would be the midpoint of its uncertainty interval, i.e. $\hat{w}_{i,l} = (\hat{w}_{i,l}^d + \hat{w}_{i,l}^u)/2$.

Assume another case, where the inner nodes for approximation have the form $\{(x_{i,l}; \hat{u}_{i,l}^d; \hat{u}_{i,l}^u) \mid l=2, 3, \dots, z_i-1\}$ (elicited using PE or LE). The utility uncertainty interval $[\hat{u}_{i,l}^d; \hat{u}_{i,l}^u]$ should be rescaled to the utility uncertainty interval $[\hat{w}_{i,l}^d; \hat{w}_{i,l}^u]$ according to $u_{worst,i}$ and $u_{best,i}$, thus maintaining their interval form. Following (6) it is obvious that minimization of $w_{i,l}$ would be possible by setting $u_{worst,i}$, $u_{best,i}$ and $u_{i,l}$ to their lowest levels. In the same fashion, maximizing $w_{i,l}$ would be possible by setting $u_{worst,i}$, $u_{best,i}$ and $u_{i,l}$ to their highest levels. That results to the following:

$$\begin{cases} w_{i,l} \in [\hat{w}_{i,l}^d; \hat{w}_{i,l}^u], \text{ where} \\ w_{i,l}^d = \hat{u}_{best,i}^d \hat{u}_{i,l}^d + \hat{u}_{worst,i}^d (1 - \hat{u}_{i,l}^d), \\ w_{i,l}^u = \hat{u}_{best,i}^u \hat{u}_{i,l}^u + \hat{u}_{worst,i}^u (1 - \hat{u}_{i,l}^u). \end{cases} \quad (8)$$

The resulting nodes of the rescaled utility function $w_i(\cdot)$ thus have errors (uncertainty intervals) on only one coordinate – the utility, and may be formalized as $\{(x_{i,l}; \hat{w}_{i,l}^d; \hat{w}_{i,l}^u) \mid l=2, 3, \dots, z_i-1\}$. Unlike the previous case, an appropriate point estimate of the interval utility should be calculated as

$$\begin{cases} \hat{w}_{i,l} = \hat{u}_{best,i}^m \hat{u}_{i,l}^m + \hat{u}_{worst,i}^m (1 - \hat{u}_{i,l}^m), \\ \text{where} \end{cases} \quad (9)$$

$$\begin{cases} \hat{u}_{best,i}^m = (\hat{u}_{best,i}^d + \hat{u}_{best,i}^u)/2, \\ \hat{u}_{worst,i}^m = (\hat{u}_{worst,i}^d + \hat{u}_{worst,i}^u)/2, \\ \hat{u}_{i,l}^m = (\hat{u}_{i,l}^d + \hat{u}_{i,l}^u)/2. \end{cases}$$

4. Algorithm for construction and rescaling of quasi-multimodal utility function

All procedures that need to be implemented in order to construct a quasi-multimodal utility function are collected in the following algorithm.

Construction of a quasi-multimodal utility function of a FRDM

1. Define the boundaries of the prize range x_{min} and x_{max} .

2. Define the number m and type of the extrema.

3. Elicit the uncertainty intervals $\hat{x}_{opt}^i \in [\hat{x}_{opt min}^i; \hat{x}_{opt max}^i]$ ($i=1, 2, \dots, m$) of each extremum using the algorithms from [Nikolova et al., 2006] depending on its type.

4. Divide the interval of prizes into sections with monotonically increasing/decreasing preferences $[x_{min}; \hat{x}_{opt min}^1]$, $[\hat{x}_{opt max}^1; \hat{x}_{opt min}^2]$, $\dots$, $[\hat{x}_{opt max}^i; \hat{x}_{opt min}^{i+1}]$, $\dots$, $[\hat{x}_{opt max}^m; x_{max}]$,

5. Construct all local utility functions in the sections with monotonically increasing/decreasing preferences:

a) elicit several inner nodes for each local utility function using either LE (which generates errors on the prizes x) or UE (which generates errors on the utilities u);

b) arctg-approximate each local utility function on the elicited nodes;

c) if the arctg-approximated utility function does not well describe the preferences of the DM in a given section, then replace it by linear interpolation on the midpoints of the elicited nodes;

6. Define the global best and worst prize x_{best} and x_{worst} and elicit the utility uncertainty intervals of x_{min} , x_{max} , and $\hat{x}_{opt}^i$ ($i=1, 2, \dots, m$), different from x_{best} and x_{worst} , using either PE or LE.

7. Rescale the local utility function in the i -th section ($i=1, 2, \dots, m$) (the first section being $[x_{min}; \hat{x}_{opt min}^1]$, and the i -th section being $[\hat{x}_{opt max}^{i-1}; \hat{x}_{opt min}^i]$) with monotonically

increasing/decreasing preferences, approximated on nodes with errors on x :
– transform the initial utility values into rescaled uncertainty interval $w_{i,l} \in [w_{i,l}^d; w_{i,l}^u]$, where $w_{i,l}^d = \hat{u}_{best,i}^d u_{i,l} + \hat{u}_{worst,i}^d (1 - u_{i,l})$, $w_{i,l}^u = \hat{u}_{best,i}^u u_{i,l} + \hat{u}_{worst,i}^u (1 - u_{i,l})$ (where $u_{i,l}$ is a non-rescaled utility in the i -th section, and $l=2, 3, \dots, z_i-1$ (z_i being the total number of elicited nodes in the i -th section);

– find a point estimate of the rescaled utility as $\hat{w}_{i,l} = (\hat{w}_{i,l}^d + \hat{w}_{i,l}^u)/2$.

8. Rescale the local utility function in the i -th section ($i=1, 2, \dots, m$) with monotonically increasing/decreasing preferences, approximated on nodes with errors on u :

– transform the initial utility values into rescaled uncertainty interval $w_{i,l} \in [w_{i,l}^d; w_{i,l}^u]$, where $w_{i,l}^d = \hat{u}_{best,i}^d \hat{u}_{i,l}^d + \hat{u}_{worst,i}^d (1 - \hat{u}_{i,l}^d)$,

$w_{i,l}^u = \hat{u}_{best,i}^u \hat{u}_{i,l}^u + \hat{u}_{worst,i}^u (1 - \hat{u}_{i,l}^u)$;

– find a point estimate of the rescaled utility as $\hat{w}_{i,l}^m = \hat{u}_{best,i}^m \hat{u}_{i,l}^m + \hat{u}_{worst,i}^m (1 - \hat{u}_{i,l}^m)$, where $\hat{u}_{best,i}^m = (\hat{u}_{best,i}^d + \hat{u}_{best,i}^u)/2$,

$\hat{u}_{worst,i}^m = (\hat{u}_{worst,i}^d + \hat{u}_{worst,i}^u)/2$, $\hat{u}_{i,l}^m = (\hat{u}_{i,l}^d + \hat{u}_{i,l}^u)/2$.

9. Construct the final quasi-multimodal utility function, along with the initial and rescaled uncertainty intervals on x and/or u .

5. Numerical examples

Two numerical examples shall demonstrate the application of the algorithm from section 3.

Example 1

Let's analyze the random variable X that takes values in the interval $[100; 500]$. The preferences of a FRDM over the values of X are quasi-multimodal. The FRDM declares that there are two extrema in his preferences over X , the first being minimum extremum, the second being a maximum extremum. Using the

algorithms from [Nikolova et al., 2006], the FRDM has elicited that $x_{opt}^1 \in [\hat{x}_{opt min}^1; \hat{x}_{opt max}^1] \equiv [210; 230]$, whereas $x_{opt}^2 \in [\hat{x}_{opt min}^2; \hat{x}_{opt max}^2] \equiv [360; 380]$. That divides the interval of X into $m=2$ sections:

- section $[x_{min}; \hat{x}_{opt min}^1] \equiv [100; 210]$ with monotonically decreasing preferences;
- section $[\hat{x}_{opt max}^1; \hat{x}_{opt min}^2] \equiv [230; 360]$ with monotonically increasing preferences;
- section $[\hat{x}_{opt max}^2; \hat{x}_{opt min}^3] \equiv [380; 500]$ with monotonically decreasing preferences.

1) Constructing the local utility function in the zero section $[x_{min}; \hat{x}_{opt min}^1] \equiv [100; 210]$

UE is used to elicit 4 inner quantiles with quantile indices $u_{1,2}=0.8$, $u_{1,3}=0.6$, $u_{1,4}=0.4$, and $u_{1,5}=0.2$ (i.e. $z_1=6$). Since preferences are monotonically decreasing, then $u_{1,1}=1$, $x_{1,\mu_1} \in [\hat{x}_{1,\mu_1}^d; \hat{x}_{1,\mu_1}^u] \equiv [100; 100]$, $u_{1,6}=0$, $x_{1,\mu_6} \in [\hat{x}_{1,\mu_6}^d; \hat{x}_{1,\mu_6}^u] \equiv [210; 210]$. The elicitation results for the other nodes are as follows:

$$\begin{aligned} x_{1,\mu_2} &\in [\hat{x}_{1,\mu_2}^d; \hat{x}_{1,\mu_2}^u] \equiv [114; 128], \\ x_{1,\mu_3} &\in [\hat{x}_{1,\mu_3}^d; \hat{x}_{1,\mu_3}^u] \equiv [130; 150], \\ x_{1,\mu_4} &\in [\hat{x}_{1,\mu_4}^d; \hat{x}_{1,\mu_4}^u] \equiv [136; 164], \\ x_{1,\mu_5} &\in [\hat{x}_{1,\mu_5}^d; \hat{x}_{1,\mu_5}^u] \equiv [160; 172]. \end{aligned}$$

The optimal parameters of the arctg-approximated utility function, calculated using that data are: $a_{opt}=0.0343$, $x_{0,opt}=94.2599$, $\chi_{opt}^2=0.0506$.

2) Constructing the local utility function in the first section $[\hat{x}_{opt max}^1; \hat{x}_{opt min}^2] \equiv [230; 360]$

UE is used to elicit 3 inner quantiles with quantile indices $u_{2,2}=0.25$, $u_{2,3}=0.5$, $u_{2,4}=0.75$ (i.e. $z_2=5$). Since preferences are monotonically increasing, then $u_{2,1}=0$, $x_{2,\mu_1} \in [\hat{x}_{2,\mu_1}^d; \hat{x}_{2,\mu_1}^u] \equiv [230; 230]$, $u_{2,6}=1$, $x_{2,\mu_6} \in [\hat{x}_{2,\mu_6}^d; \hat{x}_{2,\mu_6}^u] \equiv [360; 360]$. The elicitation results for the other nodes are as follows:

$$x_{2,\mu_2} \in [\hat{x}_{2,\mu_2}^d; \hat{x}_{2,\mu_2}^u] \equiv [235; 245],$$

$$x_{2,\mu_3} \in [\hat{x}_{2,\mu_3}^d; \hat{x}_{2,\mu_3}^u] \equiv [292; 315],$$

$$x_{2,\mu_4} \in [\hat{x}_{2,\mu_4}^d; \hat{x}_{2,\mu_4}^u] \equiv [320; 330].$$

The optimal parameters of the arctg-approximated utility function, calculated using that data are: $a_{opt}=0.0081$, $x_{0,opt}=261.2118$, $\chi_{opt}^2=4.3600$.

3) Constructing the local utility function in the second section $[\hat{x}_{opt max}^2; \hat{x}_{opt min}^3] \equiv [380; 500]$

UE is used to elicit 4 inner quantiles with quantile indices $u_{3,2}=0.8$, $u_{3,3}=0.6$, $u_{3,4}=0.4$, $u_{3,5}=0.2$ (i.e. $z_3=6$). Since preferences are monotonically increasing, then $u_{3,1}=1$, $x_{3,\mu_1} \in [\hat{x}_{3,\mu_1}^d; \hat{x}_{3,\mu_1}^u] \equiv [380; 380]$, $u_{3,6}=0$, $x_{3,\mu_6} \in [\hat{x}_{3,\mu_6}^d; \hat{x}_{3,\mu_6}^u] \equiv [500; 500]$. The elicitation results for the other nodes are as follows:

$$\begin{aligned} x_{3,\mu_2} &\in [\hat{x}_{3,\mu_2}^d; \hat{x}_{3,\mu_2}^u] \equiv [415; 428], \\ x_{3,\mu_3} &\in [\hat{x}_{3,\mu_3}^d; \hat{x}_{3,\mu_3}^u] \equiv [433; 457], \\ x_{3,\mu_4} &\in [\hat{x}_{3,\mu_4}^d; \hat{x}_{3,\mu_4}^u] \equiv [447; 464], \\ x_{3,\mu_5} &\in [\hat{x}_{3,\mu_5}^d; \hat{x}_{3,\mu_5}^u] \equiv [465; 478]. \end{aligned}$$

The optimal parameters of the arctg-approximated utility function, calculated using that data are: $a_{opt}=0.0294$, $x_{0,opt}=451.4221$, $\chi_{opt}^2=0.0295$.

4) Eliciting the extreme utilities and rescaling the local utility functions to a global utility

The FRDM identified that the global best prize is $x_{best}=100$, $u(100)=1$, whereas the global worst prize is $x_{worst}=500$, $u(500)=1$. Using LE, the FRDM has elicited that the utility of x_{opt}^1 is

$$u(x_{opt}^1) = u_{opt}^1 \in [\hat{u}_{opt}^{1,d}; \hat{u}_{opt}^{1,u}] \equiv [0.10; 0.18],$$

whereas the utility of x_{opt}^2 is

$$u(x_{opt}^2) = u_{opt}^2 \in [\hat{u}_{opt}^{2,d}; \hat{u}_{opt}^{2,u}] \equiv [0.75; 0.85].$$

Following step 7 of the algorithm from section 4, it is possible to rescale the values $u_{i,l}$ into $w_{i,l}$. For example, $u_{1,2}=0.8$ rescales into $w_{1,2} \in [w_{1,2}^d; w_{1,2}^u] \equiv [0.82; 0.835]$, since

$$w_{1,2}^d = \hat{u}_{best,1}^d u_{1,2} + \hat{u}_{worst,1}^d (1 - u_{1,2}) =$$

$$1 \times 0.8 + 0.1 \times 0.2 = 0.82, \quad w_{1,2}^u = \hat{u}_{best,1}^u u_{1,2}$$

$$+ \hat{u}_{worst,1}^u (1 - u_{1,2}) = 1 \times 0.8 + 0.18 \times 0.2 = 0.835$$

. The point estimate of the rescaled utility is $\hat{w}_{i,l} = (w_{i,l}^d + w_{i,l}^u)/2 = (0.82 + 0.835)/2 = 0.828$. All other rescaled utilities are calculated in the same fashion, and the results are summarized in Table 1. Figure 1 depicts the resulting global utility function, along with the uncertainty intervals on x (the initial ones) and on u (the rescaled ones).

What fig. 1 reveals is that in two of the sections, the arctg-approximated utility well describes the elicited data. Yet in the section $[\hat{x}_{opt\max}^1; \hat{x}_{opt\min}^2] \equiv [230; 360]$ the approximated curve does not fit at all to the preferences of the DM. Therefore it is justified that the arctg-approximated utility in that section is replaced by linearly interpolated utility on the midpoints of the uncertainty intervals. The resulting curve is depicted on fig. 2.

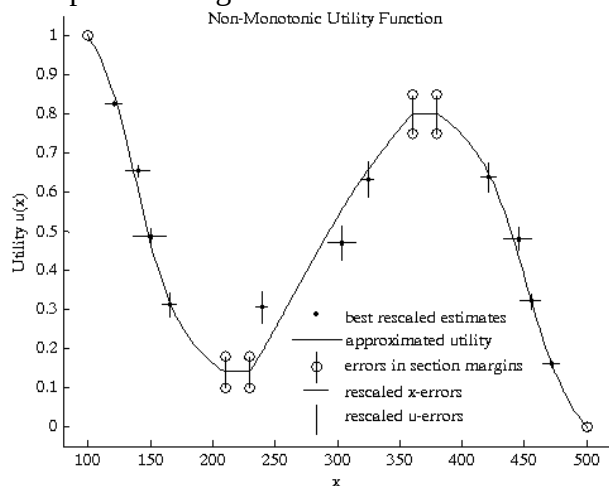


Fig. 1. The global utility function over the values of X in the interval [100; 500] from Example 1

Example 2

Let's analyze the random variable X that takes values in the interval [0; 400]. The preferences of a FRDM over the values of X are quasi-multimodal. The FRDM declares that there are two extrema in his preferences over X , the first being a maximum extremum, the second being a minimum extremum. Using the

algorithms from [Nikolova et al., 2006], the FRDM has elicited that

$$x_{opt}^1 \in [\hat{x}_{opt\min}^1; \hat{x}_{opt\max}^1] \equiv [110; 125], \text{ whereas}$$

$$x_{opt}^2 \in [\hat{x}_{opt\min}^2; \hat{x}_{opt\max}^2] \equiv [260; 280].,$$

that divides the interval of X into three sections:

– section $[x_{\min}; \hat{x}_{opt\min}^1] \equiv [0; 110]$ with

monotonically decreasing preferences;

– section $[\hat{x}_{opt\max}^1; \hat{x}_{opt\min}^2] \equiv [125; 260]$

with monotonically increasing preferences;

– section $[\hat{x}_{opt\max}^2; \hat{x}_{opt\min}^3] \equiv [280; 400]$

with monotonically decreasing preferences.

Table 1. Initial and rescaled uncertainty intervals for the global utility function from Example 1.

The bolded values are the reference points in the rescaling

i	l	$u_{i,l}$	$w_{1,2}^d$	$w_{1,2}^u$	$\hat{w}_{i,l}$
1	1	1	1	1	1
1	2	0.8	0.820	0.836	0.828
1	3	0.6	0.640	0.672	0.656
1	4	0.4	0.460	0.508	0.484
1	5	0.2	0.280	0.344	0.312
1	6	0	0.10	0.18	0.14

2	1	0	0.10	0.18	0.14
2	2	0.25	0.263	0.347	0.305
2	3	0.5	0.425	0.515	0.470
2	4	0.75	0.587	0.683	0.635
2	5	1	0.75	0.85	0.80

3	1	1	0.75	0.85	0.80
3	2	0.8	0.60	0.68	0.64
3	3	0.6	0.45	0.51	0.48
3	4	0.4	0.30	0.34	0.32
3	5	0.2	0.15	0.17	0.16
3	6	0	0	0	0

1) Constructing the local utility function in the zero section $[x_{\min}; \hat{x}_{opt\min}^1] \equiv [0; 110]$

LE is used to elicit 4 inner quantile indices of the quantiles $x_{1,2}=20$, $x_{1,3}=40$,

$x_{1,4}=60$, $x_{1,5}=80$, thus $z_1=6$ (the exact values that divide the interval $[0; 110]$ in equal section are 22, 44, 66, 88, but approximate points are chosen that the FRDM would comprehend better). Since preferences are monotonically increasing, then $x_{1,1}=0$, $u_{1,1} \in [\hat{u}_{1,1}^d; \hat{u}_{1,1}^u] \equiv [0; 0]$, $x_{1,6}=110$, $u_{1,6} \in [\hat{u}_{1,6}^d; \hat{u}_{1,6}^u] \equiv [1; 1]$. The elicitation results for the other nodes are as follows:

$$u_{1,2} \in [\hat{u}_{1,2}^d; \hat{u}_{1,2}^u] \equiv [0.04; 0.14],$$

$$u_{1,3} \in [\hat{u}_{1,3}^d; \hat{u}_{1,3}^u] \equiv [0.42; 0.58],$$

$$u_{1,4} \in [\hat{u}_{1,4}^d; \hat{u}_{1,4}^u] \equiv [0.80; 0.94],$$

$$u_{1,5} \in [\hat{u}_{1,5}^d; \hat{u}_{1,5}^u] \equiv [0.92; 0.98].$$

The optimal parameters of the arctg-approximated utility function, calculated using that data are: $a_{opt}=0.0838$, $x_{0,opt}=39.3851$, $\chi_{opt}^2=0.0145$.

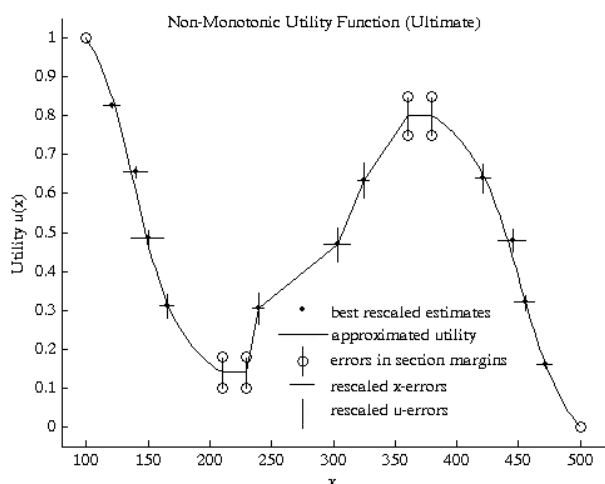


Fig. 2. Ultimate global utility function over the values of X in the interval [100; 500] from Example 1 with linear interpolation of the local utility function in the interval [230; 360]

2) Constructing the local utility function in the first section $[\hat{x}_{opt\ max}^1; \hat{x}_{opt\ min}^2] \equiv [125; 260]$

LE is used to elicit 4 inner quantile indices of the quantiles $x_{2,2}=150$, $x_{2,3}=180$, $x_{2,4}=210$, $x_{2,5}=235$, thus $z_3=6$ (the exact values that divide the interval $[125; 260]$ in equal section are 152, 179, 206, 233, but approximate points are chosen that the FRDM would

comprehend better). Since preferences are monotonically decreasing, then $x_{2,1}=125$, $u_{2,1} \in [\hat{u}_{2,1}^d; \hat{u}_{2,1}^u] \equiv [1; 1]$, $x_{2,6}=260$, $u_{2,6} \in [\hat{u}_{2,6}^d; \hat{u}_{2,6}^u] \equiv [0; 0]$. The elicitation results for the other nodes are as follows:

$$u_{2,2} \in [\hat{u}_{2,2}^d; \hat{u}_{2,2}^u] \equiv [0.78; 0.90],$$

$$u_{2,3} \in [\hat{u}_{2,3}^d; \hat{u}_{2,3}^u] \equiv [0.28; 0.42],$$

$$u_{2,4} \in [\hat{u}_{2,4}^d; \hat{u}_{2,4}^u] \equiv [0.04; 0.16],$$

$$u_{2,5} \in [\hat{u}_{2,5}^d; \hat{u}_{2,5}^u] \equiv [0.02; 0.10].$$

The optimal parameters of the arctg-approximated utility function, calculated using that data are: $a_{opt}=0.0441$, $x_{0,opt}=168.6911$, $\chi_{opt}^2=0.1021$.

3) Constructing the local utility function in the first section $[\hat{x}_{opt\ max}^2; \hat{x}_{opt\ min}^3] \equiv [280; 400]$

LE is used to elicit 4 inner quantile indices of the quantiles $x_{3,2}=300$, $x_{3,3}=330$, $x_{3,4}=350$, $x_{3,5}=380$, thus $z_3=6$ (the exact values that divide the interval $[125; 260]$ in equal section are 304, 328, 352, 376, but approximate points are chosen that the FRDM would comprehend better). Since preferences are monotonically increasing, then $x_{3,1}=280$, $u_{3,1} \in [\hat{u}_{3,1}^d; \hat{u}_{3,1}^u] \equiv [0; 0]$, $x_{3,6}=260$, $u_{3,6} \in [\hat{u}_{3,6}^d; \hat{u}_{3,6}^u] \equiv [1; 1]$. The elicitation results for the other nodes are as follows:

$$u_{3,2} \in [\hat{u}_{3,2}^d; \hat{u}_{3,2}^u] \equiv [0.04; 0.14],$$

$$u_{3,3} \in [\hat{u}_{3,3}^d; \hat{u}_{3,3}^u] \equiv [0.30; 0.42],$$

$$u_{3,4} \in [\hat{u}_{3,4}^d; \hat{u}_{3,4}^u] \equiv [0.50; 0.68],$$

$$u_{3,5} \in [\hat{u}_{3,5}^d; \hat{u}_{3,5}^u] \equiv [0.86; 0.96].$$

The optimal parameters of the arctg-approximated utility function, calculated using that data are: $a_{opt}=0.0271$, $x_{0,opt}=341.9385$, $\chi_{opt}^2=0.0286$.

4) Eliciting the extreme utilities and rescaling the local utility functions to a global utility

The FRDM identified that the global best prize is $x_{best}=400$, $u(400)=1$, whereas the global worst prize is $x_{worst}=0$, $u(0)=1$. Using LE, the FRDM has elicited that the utility of x_{opt}^1 is $u(x_{opt}^1)=u_{opt}^1 \in [\hat{u}_{opt}^{1,d}; \hat{u}_{opt}^{1,u}] = [0.76; 0.88]$, whereas the utility of x_{opt}^2 is $u(x_{opt}^2)=u_{opt}^2 \in [\hat{u}_{opt}^{2,d}; \hat{u}_{opt}^{2,u}] = [0.23; 0.33]$. Following step 8 of the algorithm from section 4, it is possible to rescale $[\hat{u}_{i,l}^d; \hat{u}_{i,l}^u]$ into $[\hat{w}_{i,l}^d; \hat{w}_{i,l}^u]$. For example, $u_{2,2} \in [\hat{u}_{2,2}^d; \hat{u}_{2,2}^u] \equiv [0.78; 0.90]$ rescales into $w_{2,2} \in [\hat{w}_{2,2}^d; \hat{w}_{2,2}^u] \equiv [0.5978; 0.825]$, since $w_{2,2}^d = \hat{u}_{best,2}^d \hat{u}_{2,2}^d + \hat{u}_{worst,2}^d (1 - \hat{u}_{2,2}^d) = 0.76 \times 0.78 + 0.23 \times 0.22 = 0.5978$, $w_{2,2}^u = \hat{u}_{best,2}^u \hat{u}_{2,2}^u + \hat{u}_{worst,2}^u (1 - \hat{u}_{2,2}^u) = 0.88 \times 0.90 + 0.33 \times 0.10 = 0.825$. The mean point $\hat{w}_{2,2}^m = \hat{u}_{best,2}^m \hat{u}_{2,2}^m + \hat{u}_{worst,2}^m (1 - \hat{u}_{2,2}^m) = 0.82 \times 0.84 + 0.28 \times 0.16 = 0.7336$, since $\hat{u}_{best,2}^m = (\hat{u}_{best,2}^d + \hat{u}_{best,2}^u)/2 = (0.76 + 0.88)/2 = 0.82$, $\hat{u}_{worst,2}^m = (\hat{u}_{worst,2}^d + \hat{u}_{worst,2}^u)/2 = (0.23 + 0.33)/2 = 0.28$, $\hat{u}_{2,2}^m = (\hat{u}_{2,2}^d + \hat{u}_{2,2}^u)/2 = (0.78 + 0.90)/2 = 0.84$.

Table 2. Initial and rescaled uncertainty intervals for the global utility function from Example 1. The bolded values are the reference points in the rescaling

i	l	$\hat{u}_{i,l}^d$	$\hat{u}_{i,l}^u$	$w_{i,l}^d$	$w_{i,l}^u$	$\hat{w}_{i,l}^m$
1	1	0	0	0	0	0
1	2	0.04	0.14	0.0304	0.1232	0.0738
1	3	0.42	0.58	0.3192	0.5104	0.4100
1	4	0.80	0.94	0.6080	0.8272	0.7134
1	5	0.92	0.98	0.6992	0.8624	0.7790
1	6	1	1	0.76	0.88	0.82

2	1	1	1	0.76	0.88	0.82
2	2	0.78	0.90	0.6434	0.8250	0.7336
2	3	0.28	0.42	0.3784	0.5610	0.4690
2	4	0.04	0.16	0.2512	0.4180	0.3340
2	5	0.02	0.10	0.2406	0.3850	0.3124
2	6	0	0	0.23	0.33	0.28

2	1	1	1	0.76	0.88	0.82
2	2	0.78	0.90	0.6434	0.8250	0.7336
2	3	0.28	0.42	0.3784	0.5610	0.4690
2	4	0.04	0.16	0.2512	0.4180	0.3340
2	5	0.02	0.10	0.2406	0.3850	0.3124
2	6	0	0	0.23	0.33	0.28

3	1	0	0	0.23	0.33	0.28
3	2	0.04	0.14	0.2608	0.4238	0.3448
3	3	0.30	0.42	0.4610	0.6114	0.5392
3	4	0.50	0.68	0.6150	0.7856	0.7048
3	5	0.86	0.96	0.8922	0.9732	0.9352
3	6	1	1	1	1	1

All other rescaled utilities are calculated in the same fashion, and the results are summarized in Table 2. Figure 3 depicts the resulting global utility function, along with the initial and rescaled uncertainty intervals.

6. Conclusions

All calculations and approximations on the utility functions in the examples were performed with the help of original program functions in MATLAB 7.0.0 environment. Four program functions handle the procedures for arctg-approximation and finding its optimal parameters depending on the type of the monotonic preferences and the errors in the elicited nodes. At the obtained optimal parameters, another set of four functions find arbitrary nodes (given one of the coordinates) from a monotonically increasing/decreasing arctg-approximated utility. For a given set of prizes, another program function finds the global utilities from the quasi-multimodal utility function. Finally, a tenth function performs the rescaling of the uncertainty intervals and constructs the quasi-multimodal utility function. All these program functions are available free of charge upon request from the author.

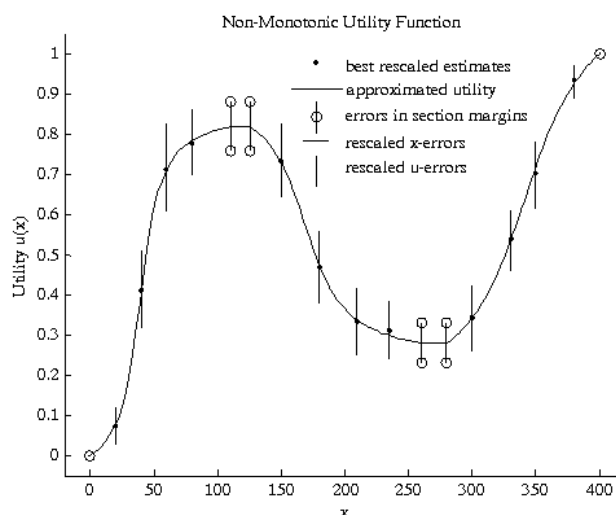


Fig. 3. The global utility function over the values of X in the interval $[0; 400]$ from Example 2

The paper discussed the construction of quasi-multimodal utility functions, where the monotonically increasing/decreasing sections of the function were constructed on nodes with interval estimates of utilities or prizes. The discussion here builds upon the approaches from [Nikolova et al., 2006] and extends them in the rescaling phase, since here the rescaling is performed according to interval utilities (of the prizes in the extrema). As demonstrated, if the elicitation nodes contain error on x , then their utilities (which are point estimates) transform into uncertainty intervals. Therefore the rescaled utility function is associated with nodes containing errors on both coordinates. In the case of elicitation nodes with errors on the utility u , the rescaled nodes contain errors only on u , but the error in the reference points in the rescaling add up to the initial one, which leads to larger uncertainty intervals (which is evident from fig. 3).

The numerical example demonstrated that in most cases the local utility functions can be arctg-approximated, yet it is possible that sometimes this form will not reflect the preferences of the DM. In that case it is always possible to replace the arctg-approximation by linear interpolation on the midpoints of the rescaled uncertainty intervals.

References

- [1]. Abdi, H. (2003). Least-Squares. – In: M. Lewis-Beck, A. Bryman, T. Futing (Eds.). *Encyclopedia for Research Methods for the Social Sciences*, Thousand Oaks (CA), Sage, 792-795.
- [2]. Clemen, R. (1996). *Making Hard Decisions: an Introduction to Decision Analysis*. Second Edition. Duxbury Press, Wadsworth Publishing Company.
- [3]. Farquhar, P. H. (1984). Utility Assessment Methods. – *Management Science*, 30(11), 1283-1300.
- [4]. Forsythe, G. E., A. Malcolm, C. B. Moler (1977). *Computer Methods for Mathematical Computations*. Prentice Hall.
- [5]. French, S. (1993). *Decision Theory: an Introduction to the Mathematics of Rationality*. Ellis Horwood.
- [6]. French, S., D. R. Insua (2000). *Statistical Decision Theory*. Arnold, London.
- [7]. Keeney, R. L., H. Raiffa (1993). *Decisions with Multiple Objectives: Preference and Value Tradeoffs*. Cambridge University Press.
- [8]. Kiefer, J. (1953) Sequential Minimax Search for a Maximum, *Proc. American Mathematical Society*, Vol. 4, pp. 502–506.
- [9]. McCord, M., R. De Neufville (1986). Lottery Equivalents': Reduction of the Certainty Effect Problem in Utility Assessment. – *Management Science*, 32, 56-60.
- [10]. Nikolova, N. D. (2007) Empirical Connection between the Number of Elicited Knots and the Quality of Analytical Approximation of a One-Dimensional Utility Function, *Machine Building and Machine Learning, Economics and Management Series*, Year II, Book 1, pp. 99-111, 2007b (in Bulgarian)
- [11]. Nikolova, N. D., A. Shulus, D. Toneva, K. Tenekedjiev (2005). Fuzzy Rationality in Quantitative Decision Analysis. – *Journal of Advanced Computational Intelligence and Intelligent Informatics*, 9(1), 65-69.

- [12]. Nikolova, N. D., K. Hirota, C. Kobashikawa, K. Tenekedjiev (2006). Elicitation of Non-Monotonic Preferences of a Fuzzy Rational Decision Maker. – *Information Technologies and Control*, Year IV, 1, 36-50.
- [13]. Pratt, J. W. (1964). Risk Aversion in the Small and in the Large. – *Econometrica*, 32, 122-136.
- [14]. Press, W. H., S. A. Teukolski, W. T. Vetterling, B. P. Flannery (1992). *Numerical Recipes – The Art of Scientific Computing*. Cambridge University Press.
- [15]. Rao, C.R., Toutenburg, H., Fieger, A., Heumann, C., Nittner, T., Scheid, S. (1999) *Linear Models: Least Squares and Alternatives*, Springer Series in Statistics.
- [16]. Stoyanov, S. (1993) *Optimization of Technological Processes*, Sofia, Tehnika.
- [17]. Tenekedjiev, K., Nikolova, N.D. (2007) *Decision Making – Subjectivity, Reality and Fuzzy Rationality*, CIELA, Sofia, Bulgaria.
- [18]. Tenekedjiev, K., N. D. Nikolova, R. Pfliegl (2006). Utility Elicitation with the Uncertain Equivalence Method. – *Comptes Rendus De L'academie Bulgare Des Sciences*, 59, Book 3, 283-288.
- [19]. Tenekedjiev, K., Nikolova, N. D., Dimitrakiev, D. (2007) Analytical One-Dimensional Utility – Comparison of Power and Arctg-Approximation, *Engineering Science*, Vol. 4, pp. 19-32.
- [20]. Von Neumann, J., O. Morgenstern (1947). *Theory of Games and Economic Behavior*. Second Edition. Princeton University Press.
- [21]. Wakker, P., D. Deneffe (1996). Eliciting von Neumann-Morgenstern Utilities when Probabilities Are Distorted or Unknown. – *Management Science*, 42, 1131-1150.

For Contacts:

: As. Prof. Natalia Danailova Nikolova, PhD,
Technical University – Varna, 1 Studentska Str.,
9010 Varna, Bulgaria, natalia@dilogos.com

ПРОБЛЕМИ ПРИ ИЗГРАЖДАНЕ НА ЕЛЕКТРОНЕН МАГАЗИН

Димитър Й. Димитракиев, Иван С. Арменски

Резюме: Статията разглежда същността и начините за изграждане на електронен магазин. Представени са преимуществата и недостатъците на този вид търговия, схематично е представена структурата на електронен магазин. Основно внимание е обърнато на последователността от проблеми, които трябва да бъдат решени при изграждане на електронен магазин, както и факторите на несигурност, които влияят върху изхода на всеки етап от изграждането.

Ключови думи: : електронен магазин, електронна търговия, несигурност, вземане на решения.

E-Store Building Problems

Dimitar Y. Dimitrakiev, Ivan S. Armenski

Abstract: The paper examines the nature and ways for e-store building. The advantages and disadvantages of this way of commerce are represented, the structure of e-store is schematic presented. We pay attention of the problems consistency, which have to be settled on e-store building, as well as the uncertainty factors, which affect the result on every stage of building.

Key words: e-store, e-commerce, uncertainty, decision making.

Увод

Развитието на информационните технологии, непрекъснатото увеличаване на ресурсите на изчислителната техника както и нарастването на пропускателната способност в Интернет водят и до нови, по-високи изисквания от страна на потребителите към вида и качеството на информацията разпространявана в мрежата. От друга страна това развитие води и до промяна на традиционните представи и технологии в бизнеса. Електронната търговия се е появила като резултат от комбинирането на тези два фактора [Dubosson, et al., 2002]. Тя дава възможност както на малките така и на големите фирми да предлагат своите продукти в Интернет, като потребителят е основен коректор за качеството и цената на предлаганите стоки и услуги [Орлов, 2006]. Всяка една фирма може да изгради собствен електронен магазин, с помощта на който да продава своите стоки.

Множество са предимствата на електронния магазин. Някои от по-важните

са очертани по-долу [Kalakota, Robinson, 2001]:

- Увеличаване обема на продажбите: магазина работи непрекъснато, няма ограничения за броя за посетителите, няма ограничения за вида и типа стоки и услуги предлагани в него;
- Разширяване на пазарния дял: в момента Интернет се ползва от повече от 65 милиона потребители, част от които се запознават със стоките и услугите предлагани в електронния магазин;
- Намаляване на разходите за поддръжка: няма работни заплати за продавачи, наеми, оборудване и наеми;
- Точно определяне на дела на конкуренцията;
- Търговия с други държави;
- Равностойна конкуренция с големи фирми: всички електронни магазини са изградени по общ принцип и подобна структура;
- Маркетингово проучване: клиентите на магазина сами показват най-актуалните за тях стоки и услуги и ценови диапазони;

- Подобряване връзката с клиента: магазина е постоянно отворен, потребителите дават своите отзиви за предлаганите продукти, задават въпроси свързани с характеристиките на продуктите и тяхната актуалност [Филлипс, 2004].

Едно от големите преимущества на електронния магазин е, че той не е ограничен в това какъв вид и колко стоки и услуги да предлага. В един електронен магазин могат да се продават както коли втора употреба, така и последен модел компютърни сървъри и лаптопи. Това гарантира увеличаване броя на посетителите на магазина и съответно шанса за продажби. По статистически данни потребителите често търсят една стока, а купуват нещо съвсем друго именно поради огромното многообразие на стоките и услугите, видени в магазина (www.911.bg).

Електронният магазин се явява голям колектив в интернет маркетинга (www.alpina.bg). Собственика и администратора на сайта във всеки един момент могат да направят справка за най-търсените продукти в техния електронен магазин, т.е. те винаги да бъдат една крачка преди желанията на техните потенциални клиенти (internet marketing в www.amazon.com) [Cotsakos, 2000].

Налице са и някои недостатъци на електронните магазини. Интернет купувача често няма точно изразен критерий за избор на стока и услуга. Ролята на интернет магазина и интернет търговията е не само да предлага тези стоки и услуги, а и да изгражда критерии за потребителско търсене. Голяма част от съвременните Интернет магазини, особено в българското web-пространство не разполагат с пълна и конкретна продуктова категоризация, точно и разбираемо ценообразуване, както и пълно описание начина на доставка на стоката до клиента. По този начин потребителят допълнително се обърква и предпочита друг сайт и точни критерии, т.е. към друг магазин с изчистен потребителски интерфейс и точни характеристики на предлаганите продукти. Голяма част от фалитите на електронните мага-

зини се дължат именно на тази проблемна политика в потребителския интерфейс и ценова политика на дизайна. Базов проблем при електронните магазини се явява и недобре оформената „витрина” на магазина. В този случай се прибегва към т.нар. свободна категоризация – card sorting. Успеваемостта на продажба е 76% при намиране на продукта от първи опит, което в интернет-търговията е много висок процент. Друг проблем е пълното информиране за характеристиките на стоките и услугите, които предлага магазина. Често при липса на информация потребителя се отказва от покупка именно поради недостатъчната информираност за продукта. Не бива обаче да се прибегва в обратния случай, когато информацията за продукта изцяло заема потребителското мислене, т.е. потребителя не е в състояние да направи избор за стока и услуга, вниманието му изцяло е заето от характеристиките на продукта, който разглежда. Правилният баланс между графичен дизайн и технически характеристики води и до повишаване на продажбите. Често фирмите изключват ролята на графичния дизайнер и web-дизайнера, което води до непрегледност и трудно възприемане на информацията, поднесена в сайта. Пълното описание на правилата за поръчка, преглед на съдържанието на потребителската кошница, както и липсата на техническа информация за желания продукт драстично намаляват доверието на потребителя към магазина, което се отразява негативно върху електронната търговия като цяло. Този проблем може да бъде разрешен с това електронните магазини да предлагат нещо повече от стоки и услуги – те трябва да предлагат преди всичко удобства за потребителя, така че да бъдат предпочетени пред конвенционалните (стандартните) магазини. Най-лесният начин за постигане на този ефект е вграждането в магазина на потребителски форуми, отзиви за работата на магазина и изграждане на пълна връзка електронен магазин – купувач. По този начин електронния магазин е в един

непрекъснат процес на развитие, като най-важният катализатор за развитието на магазина се явява клиента.

При изграждането на електронен магазин възникват редица ситуации с няколко алтернативи и нееднозначни резултати обусловени от динамичността на средата. Интуитивното решение на тези проблеми е трудно и често неадекватно поради невъзможността да се анализират всички фактори. За това е нужно да се прилагат систематични методи за количествен анализ. Статията очертава последователността от проблемни ситуации, които възникват при изграждане на електронен магазин, както и факторите на несигурност, които влияят върху изхода на всеки етап от процеса. В изложението на статията, т. 2 дава класификация на електронните магазини в три групи, т. 3 схематично представя структурата на електронните магазини, а в т. 4 са очертани проблемите и несигурността, която е налице в процеса им на изграждане.

I. Класификация на електронните магазини

Електронните магазини могат да се класифицират в следните три групи:

1) Интернет каталог – това е интернет страница (Web site) с подробна информация за вида и характеристиките на предлаганите стоки и услуги. Преди всичко се изгражда с рекламна цел. След избор на продукт потребителя има възможност да се свърже с фирмата за изпълнение на поръчката, заплащане и доставка на продукта. Пример за подобен тип електронен магазин е www.technomarket.bg (фиг. 1).

2) Електронен магазин с потребителска кошница – при него се извършва реално онлайн поръчване на продукта. Получената информация се обработва от администратора на сайта и се предава към куриерска фирма за изпълнение на поръчката. Този вид се



Фиг. 1. Електронен магазин www.technomarket.bg

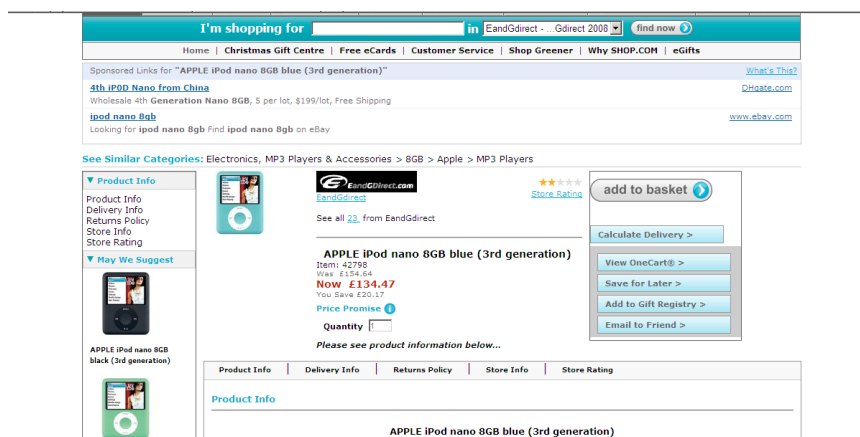
препоръчва при изграждане на магазини с неголям брой артикули или при тестови проекти. Добър пример за подобна структура е www.computer.bg (фиг. 2).

3) Автоматизиран магазин – при него автоматично се приема и обработва поръчката, плащането на продукта и заявката за доставка. Този вид е с висока

интеграция с бизнес процесите. Администратора само следи за общото функциониране на системата. Електронните магазини от този вид са най-сложни, разработват се за конкретен клиент, най-скъпи са за изграждане, но и носят най-големи приходи. Пример за такъв тип магазин е www.e-shop.uk (фиг. 3).



Фиг. 2. Електронен магазин www.computer.bg

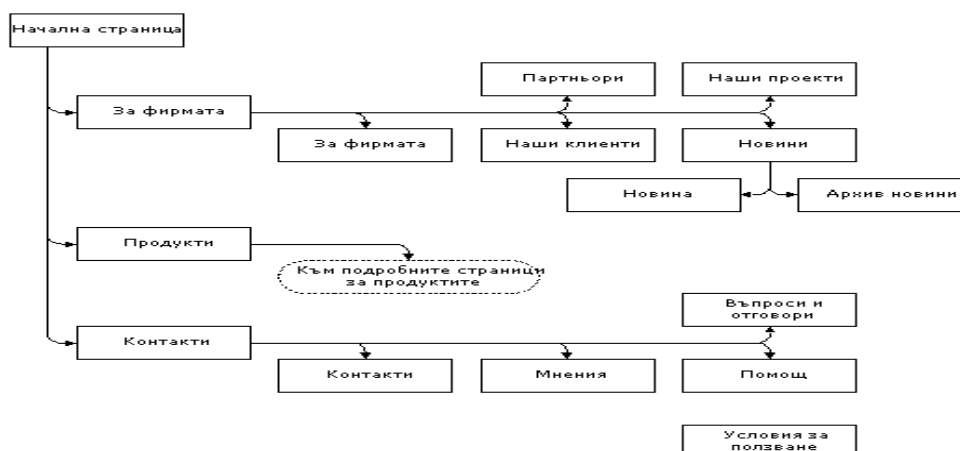


Фиг. 3. Електронен магазин www.e-shop.uk

II. Структура на електронен магазин

Всеки един интернет магазин за електронна търговия се състои от две основни части. Публична част (фиг. 4) –

служи за посещение и работа с потребители и клиенти и предоставя съответната функционалност.



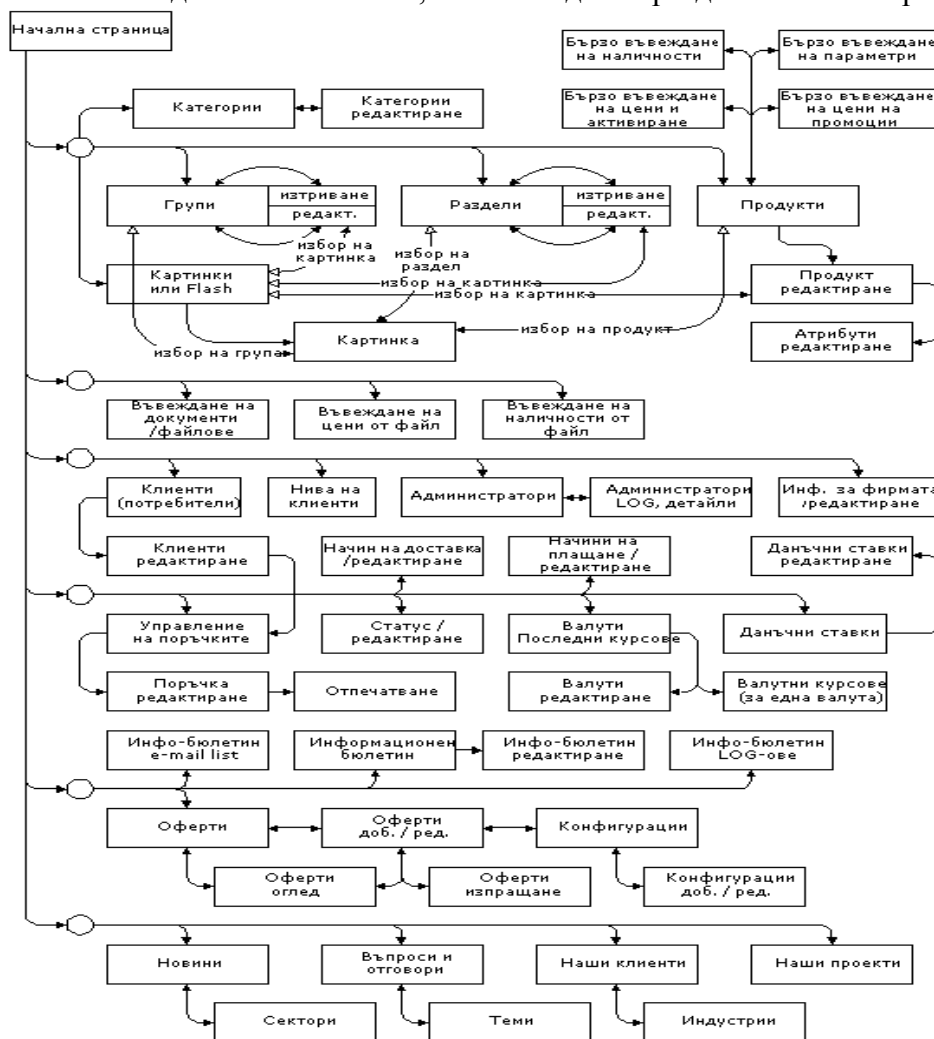
Фиг. 4. Структура на публичната част на електронен магазин

Административна част (фиг. 5) е предназначена за администраторите на системата – продуктови мениджъри въвеждащи и редактиращи информацията за продуктите, ценовите листи, търговски менажери обслужващи поръчките на клиентите.

Всяка от двете основни групи има две части – статична и динамична. Статичната област съдържа заглавната част на страницата, главното меню, подменюта и контекст към завършващата част. Статичната част на магазина дава удобство на потребителя с разположението на връзките към отделните менюта,

функциите на магазина и отделните раздели – всички те остават непроменени при работа на клиента и навигацията му в отделните страници. Динамичната област съдържа контекстно зависимата информация от съдържанието на магазина и е динамично извлечена информация от базата данни на сайта. В тази част съдържанието е най-актуално [Chaudbury, Kuilboer, 2002].

В публичната и административната част като структура страниците в електронния магазин могат да се групират в следните раздели и взаимовръзки (фиг. 6).



Фиг. 5. Структура на административната част на електронен магазин

III. Проблеми при създаване на електронен магазин

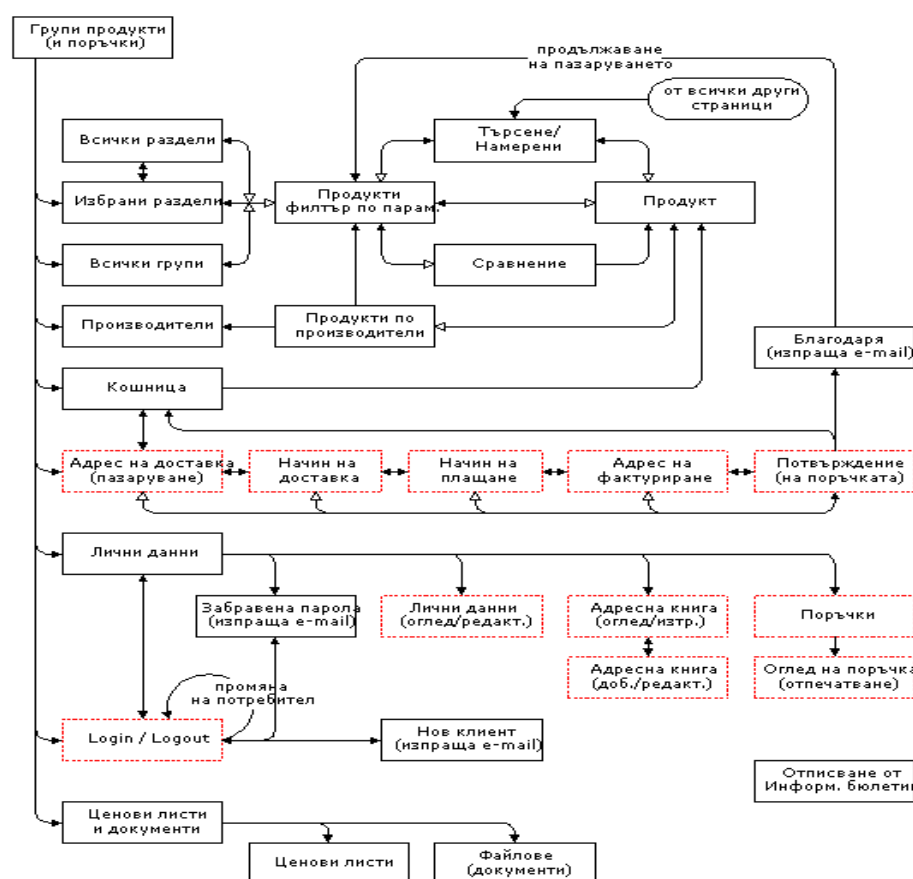
Проблемите при създаване на

електронен магазин зависят от много фактори – вид на търговската дейност (продава се по-скоро информация, отколкото реална стока), финансовите ресурси

на фирмата развиваща електронна търговия, наличния хардуер, свързаността на фирмата с Интернет както и програмистите на щат във фирмата. Основните разходи при създаване на електронен магазин могат да се разделят на следните групи: хардуер, софтуер, хостинг, маркетингови проучвания, администриране и поддръжка на софтуера и хардуера. От правилния избор за всеки един от гореизложените разходи зависи и успешния старт на електронния магазин. Преди да се вземе решение за тези разходи собственика на електронен магазин трябва да отговори на следните базови въпроси:

1. От кого и на кого ще продава сайта.
Кои са клиентските целеви групи към които магазина е насочен

2. Как ще се извършват плащанията – онлайн, по банков път, или при получаване на доставката. Възможна е и комбинация между тях
3. Как ще се организират доставките. Избор на куриерска фирма или създаване на собствена
4. Технически решения – избор на сървър, наета линия, ISP, домейн, хостинг и софтуер за изграждане на сайта
5. Екип за изграждане на сайта
6. Екип за поддръжка на сайта
7. Доизграждане на съществуващ или създаване на нов магазин
8. Избор на дизайн и потребителски интерфейс



Фиг. 6. Раздели и взаимовръзки на страниците в електронния магазин

Поради динамичността на факторите, влияещи върху успеха на електронния магазин са разработени серия от критерии за всеки един от базовите проблеми при

изграждането на електронен магазин. Тези критерии дават оценка за успеха или неуспеха на магазина в Интернет пространството. По-долу е дадено описание на

тези критерии, като са посочени и източниците на несигурност на всеки етап от решаването на проблема.

1. Целева потребителска група на магазина – в този случай собственика на магазина се стреми да обхване максимален брой посетители. Успеемостта изцяло зависи от дизайна на магазина, неговата прегледност и яснота на групата, към която са насочени стоките. Възможните фактори, влияещи върху положителната или отрицателна оценка са:

- актуалност на предлаганите стоки (представени, непредставени; детайлни характеристики, кратък преглед)
- ценова политика – в този случай конкуренцията играе решаващо значение и често обърква потребителя (връзка с пряк производител на продукта, сравнителни ценови характеристики, пълно и непълно описание на крайното ценообразуване на продукта).
- връзка на предлагания продукт със сравнителни параметри и характеристики на подобни продукти (съществува, не съществува).

2. Начин на заплащане на стоката и услугата – в този случай се получават едни от най-големите обърквания в потенциалния клиент. Разглеждат се различните видове заплащания за продукта, както и сигурността на направената транзакция [Khosrowpour, 2004]. Факторите, влияещи за вземане на решения за покупка са:

- разбираемост, неразбираемост на крайната цена
- приемане, неприемане начина на заплащане
- сигурност, несигурност на транзакцията
- време за доставка на продукта от спедиторска фирма.

3. Организация на доставките – това е един от най-възловите моменти на електронния магазин в неговия начален стадий на развитие. Електронните магазини с богат опит и дълго време на съществуване разполагат със собствени спедиторски и логистични

фирми и вериги. При тях клиента не изпитва съмнения относно времето и качеството на закупената стока. При използване на външна логистична фирма пред собственика на сайта и клиента възникват следните въпроси:

- време за доставка до клиента на поръчаната стока – кратко, нормално, дълго;
- вид и качество на получената стока – неповреден, повреден поради транспортни проблеми;
- стойност на транспортната услуга – с или без включен ДДС, константна или променлива величина зависеща от външни фактори;

4. Технически решения – Проблемите в този случай се свеждат до това дали да се наеме специалист-консултант по хардуерни и софтуерни проблеми или не. Взимането на решение зависи основно от финансови фактори поради високата квалификация на специалиста или екипа. Препоръчва се избора на консултант независещ от конкретна софтуерна или хардуерна фирма, организация или ISP.

5. Екип за изграждане на сайта – най-критичния за продължителното и успешно съществуване на магазина проблем. Обединяването на екип от специалисти в областта на програмните технологии, хардуеристи, програмни архитекти и интернет маркетинг често спира изграждането на магазина именно при решение на този проблем.

6. Екип за поддръжка на сайта – тук взимането на решение се свежда до наемането на външна фирма за поддръжка функционирането на магазина или обучението на вътрешни специалисти.

7. Доизграждане на съществуващ или създаване на нов магазин. Взимането на решение тук се свежда само до финансови критерии – оправдани ли са инвестициите за нов сайт или вече съществуващия може да се доразвие.

8. Избор на дизайн и потребителски интерфейс – от правилното разрешаване на този въпрос зависи първоначалното впечатление и времето, което ще прекара в магазина неговия посетител. Семпъл и разбираем дизайн и достатъчно дружелюбен и статичен потребителски интерфейс.

Заклучение

Електронните магазини са утвърдили се вече способи за успешна търговия на продукти, като продължават да показват ръст на интереса, посещаемостта и покупките от тях. Тук бяха разгледани подробно преимуществата и недостатъците на този вид магазини, както и видовете магазини и тяхното предназначение. На тази база се очертава добре спектъра от продукти, които имат нужните характеристики да бъдат предлагани за търговия в Интернет. Показани са примери, които демонстрират постиженията в сферата на електронните магазини в България. Очертана беше структурата и начина на изграждане на електронни магазини. Основно внимание в статията беше обърнато на проблемите, които са налице в този процес. Става ясно, че изграждането на електронен магазин е многостепенен проблем (т.е. това е процес, при който даден избор води до нов такъв във времето), като на всяка фаза са налице различни източници на несигурност, свързани както с дизайна и реализацията на магазина, така и с реакцията на

потребителите и техните предпочитания към предлаганите стоки и възможности. Бъдещите изследвания следва да бъдат насочени към моделиране и анализ на подобни многостепенни проблеми чрез техниките на количествения анализ на решения.

Литература:

- [1] Орлов, Л., Как создать электронный магазин в Интернет, Москва, Литературное агентство „Бук_Пресс”, 2006.
- [2] Филиппс, Д., PR в Интернете, ФАИР-ПРЕСС, 2004.
- [3] Chaudhury, A., Kuilboer, J-P, e-Business and e-Commerce Infrastructure, McGraw-Hill, 2002.
- [4] Cotsakos, Ch., It's Your Money: the E-TRADE Step-by-Step Guide to Online Investing, Collins, 2000.
- [5] Dubosson, M., Osterwalder, A., Pigneur, Y., eBusiness Model Design, Classification and Measurements, Thunderbird International Business Review, vol. 44, no. 1, pp. 5-23, 2002
- [6] Kalakota, R., Robinson, M., e-Business 2.0: Roadmap for Success, Addison-Wesley, 2001.
- [7] Khosrowpour, M., E-Commerce Security: Advice from Experts, Idea Group Publishing, 2004.

За контакти:

инж. д-р Димитър Йорданов Димитракиев
ddimitrakiev@yahoo.com

