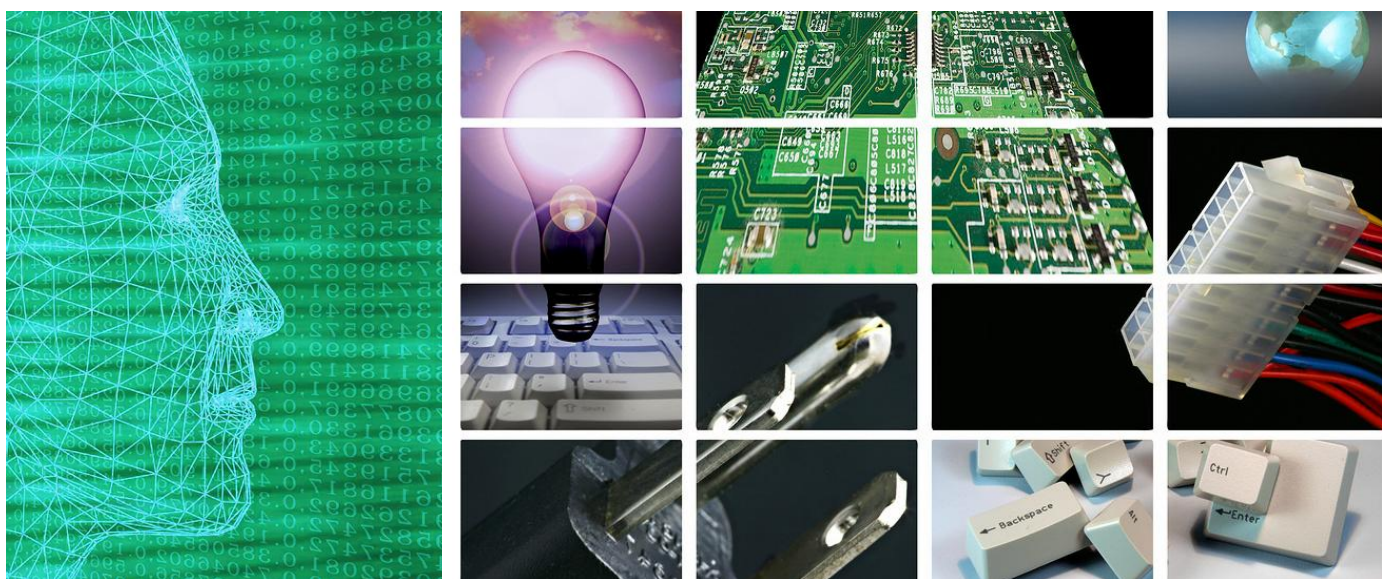


КОМПЮТЪРНИ НАУКИ И ТЕХНОЛОГИИ



Bulgaria Communications Chapter

Faculty of Computing & Automation

COMPUTER SCIENCE AND TECHNOLOGIES

Компютърни науки и технологии

Издание

на Факултета по изчислителна техника и
автоматизация
Технически университет - Варна

Гл. редактор: доц. д-р Ю. Петкова

Редакционна колегия:

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

Печат: ТУ-Варна

За контакти:

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

ISSN 1312-3335

Computer Science and Technologies

Publication

of Computing and Automation Faculty
Technical University of Varna

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

Editorial Board:

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

Printing: ТУ-Varna

For contacts:

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

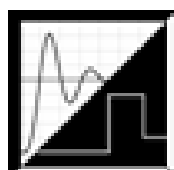
ISSN 1312-3335

СЪДЪРЖАНИЕ

CONTENTS

1	ANALYSIS OF MODERN TOOLS FOR CREATING FACE RECOGNITION SYSTEMS AND THEIR APPLICATION <i>Pavlo Anufriiev</i>	5	АНАЛИЗ СОВРЕМЕННЫХ СРЕДСТВ СОЗДАНИЯ СИСТЕМ РАСПОЗНОВАНИЯ ЛИЦ И ИХ ПРИМЕНЕНИЕ <i>Ануфриев Павел</i>	1
2	INVESTIGATION OF DIFFERENT MULTIPLIERS FOCUSED ON ULTIMATE EXECUTION SPEED IN FPGA BASED EDGE DETECTION <i>Dimitre Kromichev</i>	14	ИЗСЛЕДВАНЕ НА РАЗЛИЧНИ АЛГОРИТМИ ЗА УМНОЖЕНИЕ С ФОКУС ВЪРХУ ПОСТИГАНЕ НА НАЙ-ВИСОКА ВЪЗМОЖНА СКОРОСТ ПРИ БАЗИРАНОТО НА FPGA ОТКРИВАНЕ НА КОНТУРИ <i>Димитър Кромичев</i>	2
3	ULTIMATE EXECUTION SPEED IN FPGA BASED EDGE DETECTION: TECHNOLOGY OF ACCURATE ROUNDING IN INTEGER DIVISION WITH A VARIABLE DIVISOR <i>Dimitre Kromichev</i>	24	ПОСТИГАНЕ НА НАЙ-ВИСОКА ВЪЗМОЖНА СКОРОСТ ПРИ БАЗИРАНОТО НА FPGA ОТКРИВАНЕ НА КОНТУРИ: ТЕХНОЛОГИЯ НА ТОЧНОТО ЗАКРЪГЛЯВАНЕ ПРИ ЦЕЛОЧИСЛЕНОТО ДЕЛЕНИЕ С ПРОМЕНЛИВ ДЕЛИТЕЛ <i>Димитър Кромичев</i>	3
4	ULTIMATE EXECUTION SPEED IN FPGA BASED EDGE DETECTION: INVESTIGATING THE SPEED PARAMETERS OF ITERATIVE INTEGER DIVISION <i>Dimitre Kromichev</i>	35	ПОСТИГАНЕ НА НАЙ-ВИСОКА ВЪЗМОЖНА СКОРОСТ ПРИ БАЗИРАНОТО НА FPGA ОТКРИВАНЕ НА КОНТУРИ: ИЗСЛЕДВАНЕ НА СКОРОСТНИТЕ ПАРАМЕТРИ НА ИТЕРАТИВНОТО ЦЕЛОЧИСЛЕНО ДЕЛЕНИЕ <i>Димитър Кромичев</i>	4
5	BATTERY PERFORMANCE OPTIMIZATION USING DATA-DRIVEN MACHINE LEARNING <i>Viktor Mashkov</i>	46	ЕКСПЛОАТАЦИОННА ОПТИМИЗАЦИЯ НА АКУМУЛАТОРНИ БАТЕРИИ ПОСРЕДСТВОМ МАШИННО ОБУЧЕНИЕ <i>Виктор Машков</i>	5
6	WORKFLOW ADAPTATION FOR INTELLIGENT HUMAN-MACHINE INTERFACES <i>Miroslav Markov</i>	51	АДАПТАЦИЯ НА РАБОТНИЯ ПОТОК ПРИ ИНТЕЛИГЕНТНИ ИНТЕРФЕЙСИ ЧОВЕК-МАШИНА <i>Мирослав Марков</i>	6
7	CARDIOVASCULAR DISEASE RECOGNITION THROUGH DEEP NEURAL NETWORKS <i>Dimitar V. Krastev</i>	59	РАЗПОЗНАВАНЕ НА СЪРДЕЧНОСЪДОВИ ЗАБОЛЯВАНИЯ ЧРЕЗ НЕВРОННИ МРЕЖИ <i>Димитър В. Кръстев</i>	7

8	GAUSSIAN NAIVE BAYES CLASSIFIER <i>Neli An. Arabadzieva – Kalcheva, Ginka K. Marinova, Maya P. Todorova</i>	63	ГАУСОВ НАИВЕН БЕЙСОВ КЛАСИФИКАТОР <i>Нели Ан. Арабаджиева – Калчева, Гинка К. Маринова, Мая П. Тодорова</i>	8
9	TEXT CLASSIFICATION TASK IN MACHINE LEARNING <i>Neli An. Arabadzieva – Kalcheva, Maya P. Todorova, Ginka K. Marinova</i>	68	ЗАДАЧА ЗА КЛАСИФИКАЦИЯ НА ТЕКСТ В МАШИННОТО ОБУЧЕНИЕ <i>Нели Ан. Арабаджиева – Калчева, Мая П. Тодорова, Гинка К. Маринова</i>	9
10	EVENT TREE ANALYSIS <i>Neli An. Arabadzieva – Kalcheva, Ivelin M. Ivanov, Pavlina S. Linova</i>	75	АНАЛИЗ ЧРЕЗ ДЪРВО НА СЪБИТИЯТА <i>Нели Ан. Арабаджиева – Калчева, Ивелин М. Иванов, Павлина С. Линова</i>	10
11	METHODS OF ANALYSIS OF MODERN OPTICAL FIBERS <i>Zhejno Zhejnov, Desislava Angelova</i>	79	МЕТОДИ ЗА АНАЛИЗ НА СЪВРЕМЕННИ ОПТИЧНИ ВЛАКНА <i>Жейно Жейнов, Десислава Ангелова</i>	11
12	TRANSMISSION OF SIGNALS AND NOISE THROUGH LINEAR CHAINS CHARACTERISTICS OF A MATCHED COHERENT FILTER <i>Georgi Nikolov</i>	90	ПРЕМИНАВАНЕ НА СИГНАЛИТЕ И ШУМОВЕТЕ ПРЕЗ ЛИНЕЙНИ ВЕРИГИ И ХАРАКТЕРИСТИКИ НА СЪГЛАСУВАН ФИЛТЪР <i>Георги Николов</i>	12



АНАЛИЗ СОВРЕМЕННЫХ СРЕДСТВ СОЗДАНИЯ СИСТЕМ РАСПОЗНОВАНИЯ ЛИЦ И ИХ ПРИМЕНЕНИЕ

Ануфриев Павел

Abstract: Проведен сравнительный анализ существующих средств создания систем распознавания лиц; выявлены их достоинства и недостатки. Определены основные функциональные требования к экспериментальному образцу программной системы распознавания лиц со встроенной усовершенствованной моделью искусственного интеллекта и сформированы варианты ее использования. Представлен интерфейс экспериментальной системы, основные возможности созданного образца. Приведены результаты опытов.

Keywords: система распознавания лиц, экспериментальная система, модель, искусственный интеллект, AI, NN, CNN

ANALYSIS OF MODERN TOOLS FOR CREATING FACE RECOGNITION SYSTEMS AND THEIR APPLICATION

Pavlo Anufriev

Abstract: A comparative analysis of existing tools for creating face recognition systems was carried out; their advantages and disadvantages are revealed. The main functional requirements for an experimental sample of a face recognition software system with a built-in improved artificial intelligence model are determined and variants of its use are formed. The interface of the experimental system, the main features of the created sample are presented. The results of the experiments are presented.

Keywords: face recognition system, experimental system, model, artificial intelligence, AI, NN, CNN

1. Введение

Задача распознавания лиц является одной из актуальных задач искусственного интеллекта (AI). Системы, решающие эту задачу, могут быть использованы для автоматизации идентификации людей на изображениях, для проверки и авторизации пользователей систем или сотрудников компании, а также для многих других задач.

Для решения таких задач используются нейронные сети (NN). Они являются одним из направлений AI, эффективно справляющихся с подобными задачами, поскольку их реализация основана на мыслительной деятельности человека, что обеспечивает устойчивость к шумам и искажениям в данных. Наиболее подходящими для решения такой задачи являются сверточные нейронные сети (CNN), которые с самого начала были нацелены на распознавание изображений [1]–[3]. Кроме того, все больше современных систем распознавания лиц используют в качестве моделей именно многочисленные модификации CNN.

Учитывая все это, основной целью работы является сравнительный анализ существующих систем распознавания лиц с моделями AI и, в частности, моделями CNN и средств их проектирования для определения путей создания экспериментальной системы распознавания для образовательных организаций.

2. Системы распознавания лиц и способы их улучшения

Начиная с 1990 г., ряд компаний [4]–[8] предлагали и продолжают создавать системы, использующие нейронную сеть и другие подходы машинного обучения для решения таких разноплановых задач классификации и распознавания, как биометрическая идентификация

человека (отпечатки пальцев, лица, сетчатки глаза), компьютерное зрение, робототехника и др.

Следует отметить, что в 2013 году CNN начали использовать для задачи распознавания лиц [1], [2], [9]. Первое использование CNN позволило повысить точность неограниченного распознавания лиц в 15 раз. Таким образом, современное развитие методов создания системы распознавания на основе NN и CNN позволяет строить и практически использовать системы, работающие в режиме реального времени, не требуя значительных финансовых и материальных затрат. При этом возможно дальнейшее улучшение качественных характеристик систем, таких как точность и правильность распознавания.

В [10] предложена усовершенствованная архитектура OpenFace CNN [11] и процедура ее обучения, обеспечивающая повышение точности распознавания без снижения временных характеристик, достаточных для работы в реальном масштабе времени. В основе модификации лежат изменения подходов к ее обучению.

Выделено два направления модификации. Первое предполагает использование специальной функции ошибки Triplet Loss, которая позволяет уменьшить расстояние векторов признаков между изображениями одного и того же человека и увеличить расстояние между векторами изображений разных людей. Второе направление предполагает обучать классификатор с последующим удалением последнего классифицирующего слоя. Таким образом усовершенствованная модели распознавания строится на базе OpenFace CNN и обучается как классификатор, при этом добавлен дополнительный слой для выполнения классификации.

Также в дополнение к выбранному методу обучения предлагается реализовать усовершенствованную структуру сети. Было решено упростить его, но при этом обучить на достаточно большом наборе данных [12]. Для этого в усовершенствованной сетевой структуре были удалены 4-я и 5-я группы начальных слоев. Исследования усовершенствованной сети [10] показали, что предложенная модификация существенно сокращает время обучения без снижения качества распознавания.

Для практической проверки возможности создания системы распознавания лиц для малых организация, типа образовательных учреждений, предлагается создать экспериментальный образец такого рода системы на базе предложенной усовершенствованной модели.

3. Анализ функциональных возможностей систем распознавания лиц

При разработке экспериментальной системы для выбора ее функциональных возможностей выполнен анализ ряда инструментальных средств [4]–[8], предлагаемых на современном рынке программного обеспечения.

Далее приводится анализ систем, которые, на наш взгляд, являются лучшими продуктами в этой области [4]:

MegaMatcher SDK [5] создан для разработчиков мультибиометрических систем. Технология обеспечивает высокую надежность и скорость биометрической идентификации даже при использовании больших баз данных.

Доступный в виде SDK, он позволяет разрабатывать крупномасштабные продукты для идентификации отпечатков пальцев, радужной оболочки глаза, лица, голоса или ладони для платформ Microsoft Windows, Linux, macOS, iOS и Android.

MegaMatcher SDK доступен для скачивания, но пробная версия продукта действует 30 дней. Главный недостаток – цена продукта – от 2 500 до 5 000 евро. Следующий недостаток заключается в том, что MegaMatcher SDK не поддерживает язык программирования Python, а само приложение десктопное. Сложность использования SDK будет заключаться в адаптации библиотек на языках JAVA, C++, VB под решения исследователя AI. В

современных условиях все чаще используются кроссплатформенные и веб-решения для решения задач машинного или глубокого обучения.

Кроме того, невозможно исследовать модели AI, используемые для распознавания, обнаружения, верификации и идентификации лиц. То есть MegaMatcher SDK — не лучший выбор для последующего использования при разработке экспериментальных систем.

MegaMatcher ABIS [6] — это высокопроизводительное решение для биометрической идентификации, разработанное компанией Neurotechnology с использованием отмеченных наградами технологий [14]. Приложение готово к использованию или может быть настроено в соответствии с конкретными потребностями бизнеса и содержит все необходимые компоненты для государственных и корпоративных приложений по всему миру.

Как и в случае с MegaMatcher SDK, основным недостатком является цена продукта, а если продукт бесплатный, то есть ограничение на транзакции и срок использования. Например, чтобы установить такую систему в университете, нужно тратить не менее 500 евро в месяц, а эта система используется только для хранения информации о людях в базе данных и их проверки в статическом режиме. Данная система не предназначена для распознавания лиц в режиме видеонаблюдения.

Как и MegaMatcher SDK, MegaMatcher ABIS может использоваться только конечными пользователями. Однако благодаря анализу этой системы можно выбрать подходы относительно административного и клиентского интерфейсов и функциональных решений, которые используются в этой системе.

SkyBiometry [7] — это биометрическая технология, которая предоставляет облачное программное обеспечение для обнаружения и распознавания лиц как услугу (SaaS) и является дочерней компанией Neurotechnology. Это веб-приложение. Кроме того, SkyBiometry имеет собственный API и библиотеки для различных сред разработки. Также есть Python API.

Главный недостаток заключается в том, что нет возможности изучить модели AI, используемые в приложении. Однако, поскольку API общедоступен, ряд функционала можно использовать для разработки экспериментальной системы.

SentiVeillance SDK [8] предназначен для разработки программного обеспечения, которое выполняет биометрическую идентификацию лица, обнаруживает движущихся пешеходов, транспортные средства или другие объекты, а также выполняет автоматическое распознавание номерных знаков с использованием видеопотоков в реальном времени с цифровых камер наблюдения. SDK для видеонаблюдения использует GPU ПК или рабочей станции.

Но, на наш взгляд, как и другие системы, SentiVeillance SDK не дает возможности изучить используемые в системе модели AI. Цена SentiVeillance также достаточно велика: если модуль SDK для распознавания лиц стоит всего 200 Евро, то продукт SentiVeillance для конечного пользователя стоит уже 5000 Евро и дополнительная сумма за количество камер, которые будут использоваться для распознавания. Пробный период тоже 30 дней.

В результате проведенного анализа определены основные функции и возможности для построения экспериментальной системы.

Ориентируясь на модуль Simple Faces (Face Verification) MegaMatcher SDK для экспериментальной системы были выбраны подходы для реализации функций детекции на фотографии, на которой находится один или несколько человек. Кроме этого, детекции через камеру, идентификации и верификации человека. Создание токена изображения происходит автоматически, при котором фотография с лицом преобразуется в изображение 96 на 96 и храниться в базе.

Функция создания портрета изображения (Capture ICAO image) заслуживает отдельного внимания. Для создания портрета изображения с лицом нужно выполнять

выравнивание по горизонтали и вертикали, изменять угол наклона головы, удалить все препятствия на лице, удовлетворить условия качества самого изображения.

С учетом возможностей административной панели MegaMatcher ABIS в экспериментальной системе реализован следующий функционал:

- Выбор параметров (порогов) для идентификации и верификации (пробная версия).
- Задание биографических сведений для лиц (в разработке).
- Добавление различных ролей и разрешения доступа для пользователей системы (администратор, аудитор, оператор пропускного пункта, оператор клиентского интерфейса и т.п.) (реализовано ряд функций).
- мониторинг и добавление пользователей (реализовано).
- добавление галерей (для баз изображений) (реализовано ряд функций).
- просмотр метрик дедупликации, биометрических операций, сопоставимых метрик (планируется реализация).

Также определен клиентский интерфейс, который поддерживает:

- мониторинг и добавление пользователей (реализовано).
- идентифицировать (реализовано).
- отскакивать документ (планируется реализация).
- произвести поиск (реализовано).
- просмотр транзакций (не реализовано).
- поиск дубликатов (не реализовано).
- функционал для подключения сторонних устройств (например, считывателя отпечатков пальцев) (реализовано в случае IP камер).

Функционал экспериментальной системы для работы с видеорядом полученным от IP камер определен с учетом возможностей SentiVeillance SDK для систем видеонаблюдения.

4. Архитектура экспериментальной системы

При проектировании экспериментальной системы рассматривались следующие направления ее дальнейшего использования:

- Охрана - проведение идентификации людей без непосредственного контакта с ними, что обеспечивает безопасный доступ людей к соответствующим объектам, требующим системы безопасности.
- Научные исследования – проведение научных исследований в области AI, которые обеспечат выбор наиболее эффективных методов распознавания лиц и последующее тестирование и проверку этих методов.
- Образовательная – подготовка студентов, которые будут заниматься разработкой интеллектуальных или программных систем, поэтому разработка экспериментальной системы является образовательной задачей.

Экспериментальная система распознавания состоит из сервера распознавания, подключенного к IP-камерам или веб-камерам ПК. Он обрабатывает данные с камер через видеоускоритель, сервер баз данных и ARM разработчика. Схема узлов системы представлена на рис. 1.

Экспериментальная система создавалась с учетом следующих классов пользователей.

Сотрудники службы безопасности имеют прямой доступ к базе данных лиц, добавляют посетителей в базу данных, могут вносить их в «красный» или «зеленый» списки идентифицированных лиц. Также они контролируют доступ на объект и передвижение по нему лиц, благодаря работе систем видеонаблюдения.

Технический персонал (ученые) имеют ограниченный доступ к базе данных, имеют доступ к исходному коду системы, могут вносить в него соответствующие изменения, улучшать работу системы, модели AI, отслеживать ошибки при работе. Они занимаются

функционалом для сбора статистических данных. Они не могут редактировать «красный» и «зеленый» списки.

Сотрудники структурных подразделений могут добавлять в базу данных посетителей (например, студентов и сотрудников своего структурного подразделения).

Назначенные лица администрации объекта имеют права доступа к базе данных, имеют право следить за «красным» и «зеленым», имеют право изменять эти списки, однако с письменного согласия или под контролем начальника службы безопасности. Кроме того, они могут отслеживать статистику перемещения сотрудников, студентов и посетителей. Также они получают статистику отсутствия сотрудников. Они могут проверить статистику, учитывающую рабочее время.

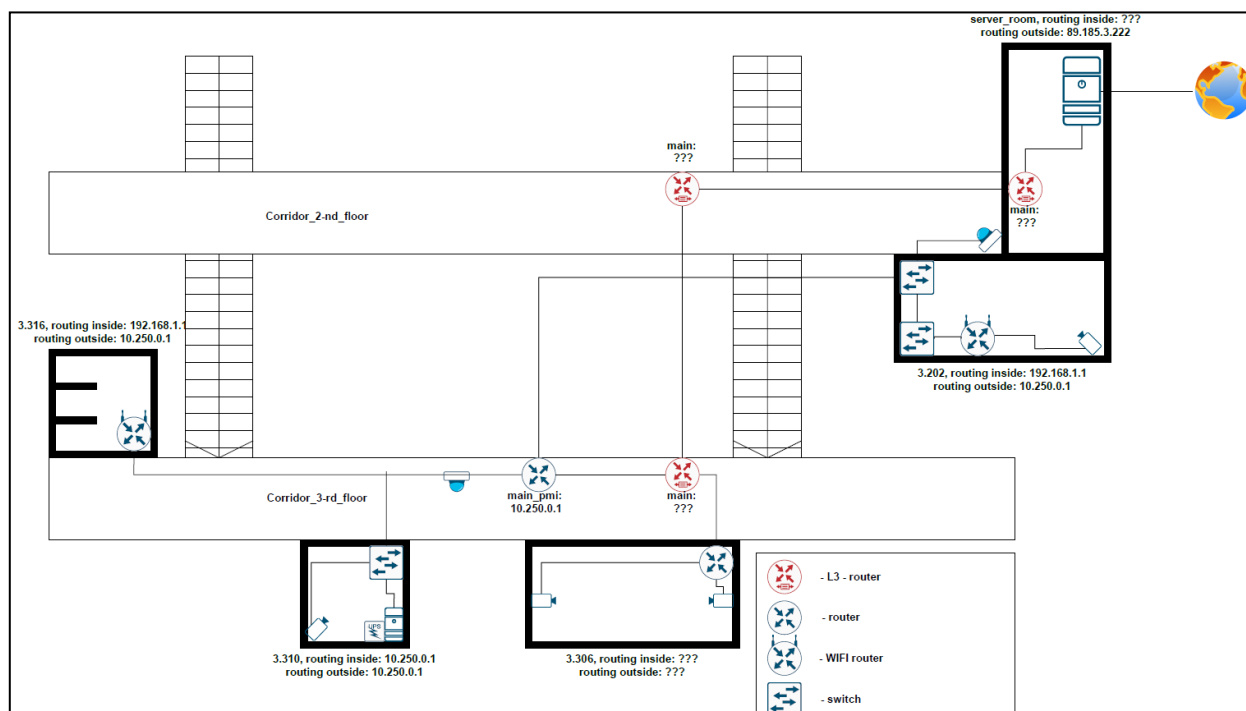


Рис. 1. Схема экспериментальной системы на определенном объекте

Функциональные требования к экспериментальной системе определяются через варианты использования. Ниже представлены списки кейсов для администратора, пользователя и ученого.

Администратор отвечает за управление пользователями и обслуживание системы.

В свою очередь, пользователю разрешено:

- идентифицировать людей на изображении.
- проверить лиц на изображении.
- сравнить лица на двух изображениях.
- сравнить человека на изображении с загруженными паспортными данными.
- идентифицировать людей во время видеозаписи.
- проверять лиц во время видеозаписи.
- просматривать информацию об идентифицируемых лицах.
- наблюдать за людьми.
- управлять подключенными устройствами (видеокамерами, веб-камерами, IP-камерами).
- выбрать устройство для мониторинга.
- делать фотографии с помощью устройства наблюдения.
- получить фрагмент записанного видео.

Ученый отвечает за настройку алгоритмов AI. То есть ему даются следующие возможности, чтобы:

- получить фрагмент записанного видео
- добавить алгоритмы AI.
- обучать алгоритмам AI.
- изменить архитектуру AI.
- проверить работу AI.
- визуализировать работу AI.
- получать статистические данные о функционировании AI.
- загружать данные для обработки и сохранять их в базе данных.
- получать информацию о загруженных данных.
- управлять библиотеками для AI.
- сравнить работу нескольких алгоритмов AI.
- управлять наборами данных (загрузка, распространение, путь и т. д.).
- получать информацию о наборах данных.
- управлять «весами».

Разработана первая версия программного обеспечения экспериментальной системы распознавания лиц со встроенной моделью AI. Открытый исходный код системы доступен в репозитории GitLab [15]

Для разработки приложения использовалась технология Docker [16], которая позволяет использовать облачную инфраструктуру для разработки, тестирования, доставки и запуска веб-приложений в средах с поддержкой контейнеризации.

Для разработки приложения использовались фреймворк Django [17] и база данных PostgreSQL, поэтому было решено, что эти технологии будут действовать как два взаимосвязанных контейнера Docker.

Django — это фреймворк, использующий архитектурный шаблон MVT и состоящий из трех компонентов: Model, View, Template.

В одном Django-приложении возможно разместить несколько разных модулей, имеющих свой функционал. Поскольку экспериментальная система должна состоять из двух программных блоков с разными задачами (отвечающих за работу с камерами и AI), было решено создать следующие три модуля:

- «main» — отвечает за навигацию между модулями и административной панелью.
- «ai_camera» — отвечает за задачи распознавания лиц с данными с IP-камер, клиентских веб-камер и загруженными фотографиями.
- «исследователи» - ответственные за работу с AI (в разработке).

Рассмотрим работу модуля «ai_camera». В качестве основных классов используются модельные компоненты модуля «ai_camera»: Camera, Person.

Модель камеры состоит из полей:

- name – имя IP-камеры.
- ip_cam_adress – IP-адрес в локальной сети.
- ip_cam_user – имя пользователя.
- ip_cam_password – пароль.
- description – описание.

Модель Person состоит из полей:

- name – имя человека.
- career – профессия (студент/сотрудник).
- image – фотография.
- image_lfw_nd_array – массив данных, сгенерированный моделью AI lfw_trained_model.

Модифицированная модель AI OpenFace была реализована в модуле «ai_camera» благодаря добавлению исходного кода модели непосредственно в код системного приложения, а также последующей загрузке в него уже обученных весов.

5. Результаты экспериментов

Рассмотрим структуру и функционал главной страницы системы для работы с IP-камерой (рис. 2).

Эта страница позволяет пользователю:

- смотреть видео с IP-камеры в режиме реального времени.
- делать скриншоты с IP-камеры и отображать их на экране параллельно с просмотром видео.
- отправлять скриншоты на сервер для их дальнейшей обработки, обнаружения и распознавания лиц и отображения их на экране.
- отображает информацию об IP-камере.
- записывать видеоряд с IP-камеры, обрабатывать его для дальнейшего распознавания и просматривать записанные и обработанные фрагменты записи.

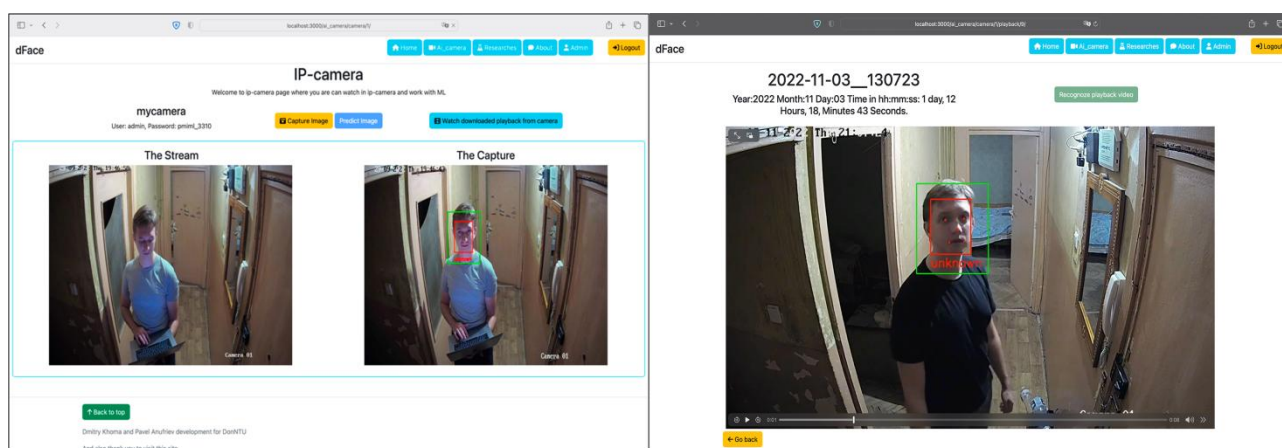


Рис. 2. Режим работы с IP-камерой

Дополнительно предусмотрена процедура отправки снимка экрана (фотографии) на сервер для обработки, обнаружения и распознавания лица (процедура распознавания). Эта процедура вызывается с помощью «Предсказать изображение» (рис. 3). Для этого был создан специальный javascript, который кодирует фотографии в формат Base64 и отправляет их на сервер для обработки. Сервер использует скрипт Python для передачи фотографии через модель AI MTCNN [18] для обнаружения лиц, и каждое лицо проходит через модифицированную модель AI OpenFace для распознавания лиц среди всех людей в модели человека. При этом, при обработке фото, используя координаты лиц, найденных благодаря MTCNN и библиотеке OpenCV, найденное лицо обводится на фото красным прямоугольником, и кроме того, только те лица, которые могут быть отправлены на распознавание модифицированным Модель OpenFace AI обведена зеленым. Если лицо было распознано и найдено в базе данных, имя человека будет написано красным цветом под лицом на фотографии, в противном случае будет написано «Неизвестно», что означает, что лицо не было найдено в базе данных. После этого список персон и их фотографии в размере 96 на 96 выводятся в отдельной таблице для более удобного взаимодействия.

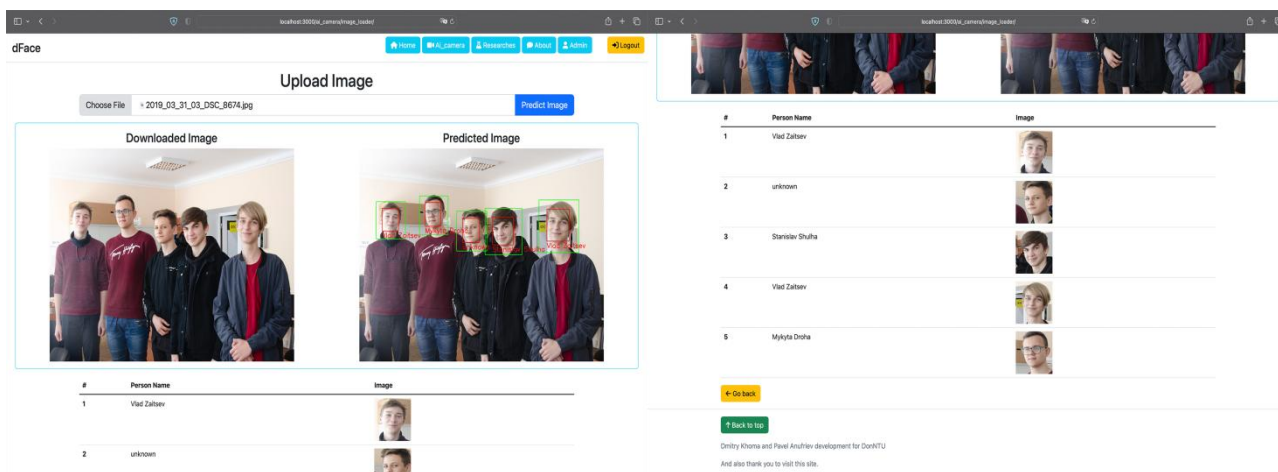


Рис. 3. Режим загрузки и обработки изображений

5. Выводы

Проведен сравнительный анализ существующих средств создания систем распознавания лиц, выявлены их преимущества и недостатки. Определены основные функциональные требования к экспериментальному образцу программной системы распознавания лиц со встроенной усовершенствованной моделью искусственного интеллекта и сформированы варианты использования. Представлен интерфейс экспериментальной системы, основные возможности созданного образца. Приведены результаты опытов.

References

- [1]. M. D. Zeiler and R. Fergus, "Visualizing and Understanding Convolutional Networks," Nov. 2013.
- [2]. C. Szegedy *et al.*, "Going Deeper with Convolutions," Sep. 2014.
- [3]. S. Ramakrishnan, "Introductory Chapter: Face Recognition - Overview, Dimensionality Reduction, and Evaluation Methods," *Face Recognition - Semisupervised Classification, Subspace Projection and Evaluation Methods*, pp. 1–6, 2016, doi: 10.5772/63995.
- [4]. "Neurotechnology." <https://www.neurotechnology.com/> (accessed Sep. 25, 2022).
- [5]. "MEGAMATCHER SDK." https://www.neurotechnology.com/megamatcher.html#nav_foldable (accessed Oct. 01, 2022).
- [6]. "MEGAMATCHER ABIS: Automated biometric identification system." <https://www.megamatcher.online/> (accessed Oct. 01, 2022).
- [7]. "SkyBiometry." <https://skybiometry.com/> (accessed Sep. 25, 2022).
- [8]. "SENTIVEILLANCE SDK." <https://www.neurotechnology.com/sentiveillance.html> (accessed Oct. 01, 2022).
- [9]. "NEUROTECHNOLOGY COMPANY INFORMATION." <https://www.neurotechnology.com/about.html> (accessed Oct. 16, 2022).
- [10]. O. Miakshyn, P. Anufriev, and Y. Bashkov, "Face Recognition Technology Improving Using Convolutional Neural Networks," in *2021 IEEE 3rd International Conference on*

Advanced Trends in Information Theory (ATIT), Dec. 2021, pp. 116–120. doi: 10.1109/ATIT54053.2021.9678722.

- [11]. “OpenFace.” <https://cmusatyalab.github.io/openface/> (accessed Oct. 20, 2022).
- [12]. Башков Є.О. and Ануфрієв П.О., “Дослідження впливу тестових баз зображень на точність систем розпізнавання облич з різними методами класифікації,” *Наукові праці Донецького національного технічного університету Серія: “Інформатика, кібернетика та обчислювальна техніка,”* vol. 2, no. 31, pp. 74–83, 2020, doi: 10.31474/1996-1588-2020-2-31-73-82.
- [13]. F. Schroff, D. Kalenichenko, and J. Philbin, “FaceNet: A unified embedding for face recognition and clustering,” in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2015, pp. 815–823. doi: 10.1109/CVPR.2015.7298682.
- [14]. “NEUROTECHNOLOGY AWARDS.” <https://www.neurotechnology.com/awards.html> (accessed Oct. 20, 2022).
- [15]. “dFace.” <https://gitlab.com/Mechetel/dFace> (accessed Oct. 16, 2022).
- [16]. “Docker overview.” <https://docs.docker.com/get-started/overview/> (accessed Oct. 20, 2022).
- [17]. “Django at a glance.” <https://docs.djangoproject.com/en/4.1/intro/overview/> (accessed Oct. 20, 2022).
- [18]. “Face Detection using MTCNN — a guide for face extraction with a focus on speed.” <https://towardsdatascience.com/face-detection-using-mtcnn-a-guide-for-face-extraction-with-a-focus-on-speed-c6d59f82d49> (accessed Oct. 16, 2022).

For contacts:

Graduate student, Anufriev, Pavlo
Department of Applied Mathematics and Informatics
Donetsk National Technical University
E-mail: pavlo.anufriev@donntu.edu.ua



INVESTIGATION OF DIFFERENT MULTIPLIERS FOCUSED ON ULTIMATE EXECUTION SPEED IN FPGA BASED EDGE DETECTION

Dimitre Kromichev

Abstract: Integer multiplication is a critical arithmetic operation for FPGA based edge detection which uses Gaussian filtering, gradient magnitude and direction. The goal of ultimate execution speed in FPGA based edge detection requires defining the values of maximum operating frequency and minimum number of clock cycles taken to secure a mathematically accurate result. The paper investigates the maximum operating frequency of different multiplication algorithms when they are implemented as a purely combinational schematic. The obtained results solve the problem of finding out whether there is a multiplier which can serve as a basis for achieving the goal of ultimate execution speed in FPGA based edge detection. Ten Intel (Altera) FPGA families are used in conducting the investigation.

Keywords: multiplication, FPGA, edge detection, ultimate execution speed, maximum operating frequency, clock cycle

1. Introduction

FPGA based multiplication has been vastly analyzed and classified [4]. Proposed is booth-2 multiplier for fast computing by reducing carry propagation time [1]. Analyzed are multiple constant multiplication using FPGA DSP blocks [3][14] and matrix multiplication using mux and Vedic multiplier with the achieved maximum frequency in Spartan 3 FPGA for 8x8 bit multiplication being 50 MHz [8]. Booth, Wallace tree, Baugh-Wooley and Montgomery multipliers are implemented in Virtex 5 with 4-bit values; the achieved clock frequencies are from 71 MHz to 148 MHz with no exact number of clock cycles being specified in obtaining the results [2]. In [24] comparison of booth, modified booth shift-and-add, array and carry save adder multiplier finds that conventional multiplication provides result in 2.06 sec, booth - in 3.09 sec, shift-and-add - in 2.31 sec., array - in 2.75 sec and modified booth multiplier - in 3.08 sec. An area-optimized and accurate signed multiplier is found to achieve up to 43% delay reduction as compared with Vivado IP multiplier [22]. Developed is multiplier regularization which restructures common multiplier algorithms into faster designs [16]. Proposed are designs for an 8-bit array multiplication aimed at reducing delay by using multiplexer [7][9]. An FPGA implementation of recursive barrel shifter multiplication is analyzed in [13]. Proposed is a method of parallel multiplication in a single DSP block [25]. Array multipliers distinguished on the basis of the used adder: (a) ripple carry adder; (b) carry look-ahead adder; (c) carry save adder are analyzed in [23]. Table based and shift-and-add constant multipliers are proposed in [11]. The study of designs for large integer multiplication show that combination multipliers offer lower delay compared to individual multipliers [15][17][18]. Approximate multipliers are investigated in terms of area, delay, energy, and degree of inaccuracy [5][6][10][12][19][20][21].

The objective of this paper is to investigate experimentally the maximum operating frequency $F_{\max}$ of different integer multiplication algorithms when they are implemented as a purely combinational schematic and executed within a single clock cycle T_{clk} . The task is to find out whether there is an algorithm which can be used in achieving the goal of ultimate execution speed in FPGA based edge detection which uses Gaussian filtering, gradient magnitude and direction. The selection of multipliers' input data widths must be in accordance with the entire range of coefficients' magnitudes in Gaussian filtering and the maximum values in gradient magnitude and

direction computations. Ten Intel (Altera) FPGA families are employed to conduct the experiments (Table 1). The tools used in the investigation are: Scilab, Intel (Altera) Quartus, TimeQuest Timing Analyzer, ModelSim. The hardware description language is VHDL. Relevant to the analyses and conclusions arrived at are gray scale images.

Table 1. Intel (Altera) FPGA families used in the investigation			
Intel (Altera) FPGA family	Technology	Intel (Altera) FPGA family	Technology
1. Cyclone	130 nm	6. Stratix	130 nm
2. Cyclone II	90 nm	7. Stratix II	90 nm
3. Cyclone III	65 nm	8. Stratix III	65 nm
4. Cyclone IV	40 nm	9. Stratix IV	40 nm
5. Cyclone V	28 nm	10. Stratix V	28 nm

2. Multiplication in FPGA based edge detection

In FPGA based edge detection which uses Gaussian filtering, gradient magnitude and gradient direction, multiplication is an indispensable integer arithmetic operation with respect to:

1) Obtaining the convolution results in computing the weighted average function by applying distributive law according to the expression

$$\sum_{i=1}^{N_c} \frac{R_{C_i}}{S_{coeff}} \quad (1)$$

where

N_c is the count of convolutions,
 R_{C_i} is a convolution result,
 S_{coeff} is the sum of all coefficients in the Gaussian filter.

2) Calculating the squares of the sides of the right triangle in Pythagoras for obtaining the value of gradient magnitude according to the expression

$$G_M = \sqrt{G_x^2 + G_y^2} \quad (2)$$

where

G_M is gradient magnitude,
 G_x is the x-gradient
 G_y is the y-gradient.

3) Calculating gradient direction according to the expressions

If ($G_y > 0 \& G_x > 0$) or ($G_y < 0 \& G_x < 0$) then

If $ G_x * 99 \geq G_y * 239$	$Dir = 0$
If $ G_x * 99 < G_y * 239 \& G_x * 169 > G_y * 70$	$Dir = 45$
If $ G_x * 169 \leq G_y * 70$	$Dir = 90$

If ($G_y > 0 \& G_x < 0$) or ($G_y < 0 \& G_x > 0$) then

$$\begin{aligned}
&\text{If } |G_x| \geq |G_y| & Dir = 0 \\
&\text{If } |G_x| < |G_y| \text{ \& } |G_x| > |G_y| & Dir = 135 \\
&\text{If } |G_x| \leq |G_y| & Dir = 90
\end{aligned} \tag{3}$$

where

Dir is the gradient direction.

3. Ultimate execution speed of FPGA based edge detection in terms of multiplication

The upper limit of $F_{\max}$ in FPGA based edge detection is defined by the embedded memory. Therefore, the reference point for deciding whether the parameter $F_{\max}$ of each of the investigated multipliers can serve as a basis for achieving the goal of ultimate execution speed in FPGA based edge detection is the maximum operation frequency of embedded memory $F_{\max}(embMem)$ in the targeted FPGA families.

The goal of ultimate execution speed in FPGA based edge detection requires that multiplication in Gaussian filtering, gradient magnitude and direction must be executed within a single clock cycle. Therefore, multiplication must be implemented as a purely combinational schematic. Hence,

$$F_{\max} = \frac{1}{T_{clk}} \tag{4}$$

where

T_{clk} is the period of the system clock,

is defined under all test conditions for

$$nT_{clk_{\min}} = \text{const} = 1 \tag{5}$$

where

$nT_{clk_{\min}}$ is minimum number of clock cycles required by the multiplier to calculate a mathematically accurate result.

Because $F_{\max}$ is defined on the basis of (5) according to

$$F_{\max} = \frac{1}{T_{clk-to-q} + T_{logic} + T_{routing} + T_{setup} - T_{skew}} \tag{6}$$

where

$T_{clk-to-q}$ is the time from clock arrival until data arrives at q ,

T_{logic} is the propagation delay through logic between flip-flops,

$T_{routing}$ is the routing delay between flip-flops,

T_{setup} is the setup time,

T_{skew} is the propagation delay of clock between the launch flip-flop and the capture flip-flop,

and because

$$F_{\max} = \frac{1}{Tclk_{\min}} \quad (7)$$

where

$Tclk_{\min}$ is the minimum length of the period of system clock,

it follows that

$$T_{\text{logic}} = T_{\text{mult}} \quad (8)$$

Thus, $Tclk_{\min}$ in (7) becomes

$$Tclk_{\min} = Tclk - to - q + T_{\text{mult}} + T_{\text{routing}} + T_{\text{setup}} + T_{\text{skew}} \quad (9)$$

Expressions (4), (5), (6), (7), (8) and (9) define $F_{\max}$ with respect to concrete values of the input data widths of the investigated multiplier. Thus, the multiplier's size is determined by the values calculated in Gaussian filtering, gradient magnitude and direction.

4. Investigating $F_{\max}$ of different multiplication algorithms

Methodology:

- Defined is $F_{\max}(\text{embMem})$ of all the embedded memory types available in the targeted FPGA families. All obtained results for $F_{\max}(\text{embMem})$ are for the highest speed grade in a particular FPGA family. The tests for all types of memory are conducted with 8-bit values. If port configuration (the depth or the width) is more than the amount a memory block can support, additional memory blocks of the same type are used.

- Ten multipliers are used in the investigation: Booth multiplier, modified Booth multiplier radix-4, Wallace multiplier, modified Wallace multiplier, Booth-Wallace multiplier, modified Booth-Wallace multiplier, Dadda multiplier, Baugh-Wooley multiplier, Braun multiplier, Vedic multiplier

- Each of the investigated multipliers is implemented as a purely combinational schematic
- All algorithms are tested for multiplying only positive integers
- Each multiplier is implemented with inputs of sizes 8x8, 16x16 and 24x24
- $F_{\max}$ for each concrete input size of each multiplier is obtained
- Tested are all combinations of operands within each of the input sizes of each multiplier
- All obtained results for $F_{\max}$ of each size of each multiplier are for the highest speed grade, in a particular FPGA family.

The obtained results are shown in Table 2, Table 3, Table 4, Table 5, Table 6, Table 7 and Table 8.

5. Analysis of results

There is a total of eight types of embedded memory in the targeted Intel (Altera) FPGA families. The experimental results presented in Table 2 show that:

Table 2. $F_{\max}(embMem)$ of embedded memory types available in the targeted Intel (Altera) FPGA families								
FPGA family	$F_{\max}(embMem)$ of embedded memory types available in the targeted Intel (Altera) FPGA families (in MHz)							
	M4K	M9K	M10K	M512	M144K	M20K	MLAB	M-RAM
Cyclone	221							
Cyclone II	252							
Cyclone III		300						
Cyclone IV		307						
Cyclone V			310				412	
Stratix	283			310				280
Stratix II	428			439				403
Stratix III		539			505		577	
Stratix IV		541			507		583	
Stratix V						510	590	

- In the FPGA families where there are two or more memory types $F_{\max}(embMem)$ of different embedded memory types is different
- In the FPGA families where there are two or more memory types, the basic memory type has lower (the lowest) $F_{\max}(embMem)$

The experimental results presented in Table 3, Table 4, Table 5, Table 6, Table 7 and Table 8 show that:

Table 3. $F_{\max}$ of 8x8 multipliers (in MHz)					
FPGA family	Booth multiplier	Modified Booth multiplier	Wallace multiplier	Modified Wallace multiplier	Booth-Wallace multiplier
Cyclone	146	178	167	181	185
Cyclone II	182	213	203	216	219
Cyclone III	228	259	249	262	265
Cyclone IV	238	266	257	268	271
Cyclone V	243	271	262	273	275
Stratix	212	244	233	247	251
Stratix II	330	360	350	363	366
Stratix III	426	455	445	459	462
Stratix IV	435	463	454	465	467
Stratix V	437	466	456	468	470

- $F_{\max}$ of 8x8 Booth multiplier is from 14.31% to 33.93% lower than $F_{\max}(embMem)$ of the basic memory type
- $F_{\max}$ of 16x16 Booth multiplier is from 59.60% to 76.47% lower than $F_{\max}(embMem)$ of the basic memory type
- $F_{\max}$ of 24x24 Booth multiplier is from 76.07% to 93.21% lower than $F_{\max}(embMem)$ of the basic memory type

Table 4. $F_{\max}$ of 8x8 multipliers (in MHz)					
FPGA family	Modified Booth-Wallace multiplier	Dadda multiplier	Baugh-Wooley multiplier	Braun multiplier	Vedic multiplier
Cyclone	188	172	163	155	139
Cyclone II	221	208	199	191	175
Cyclone III	268	255	245	237	221
Cyclone IV	273	262	253	246	232
Cyclone V	277	266	258	251	236
Stratix	254	238	229	221	205
Stratix II	368	355	346	339	322
Stratix III	464	450	441	433	419
Stratix IV	469	458	450	443	428
Stratix V	472	461	452	445	430

Table 5. $F_{\max}$ of 16x16 multipliers (in MHz)					
FPGA family	Booth multiplier	Modified Booth multiplier	Wallace multiplier	Modified Wallace multiplier	Booth-Wallace multiplier
Cyclone	52	82	72	86	90
Cyclone II	70	101	91	104	107
Cyclone III	101	125	121	128	131
Cyclone IV	107	130	126	132	134
Cyclone V	109	133	129	135	137
Stratix	90	114	110	118	122
Stratix II	156	175	171	178	181
Stratix III	195	221	217	225	227
Stratix IV	203	227	223	229	231
Stratix V	206	230	226	232	234

- $F_{\max}$ of 8x8 modified Booth multiplier is from 8.62% to 19.45% lower than $F_{\max}(\text{embMem})$ of the basic memory type
- $F_{\max}$ of 16x16 modified Booth multiplier is from 54.90% to 62.89% lower than $F_{\max}(\text{embMem})$ of the basic memory type
- $F_{\max}$ of 24x24 modified Booth multiplier is from 70.58 % to 80.54 % lower than $F_{\max}(\text{embMem})$ of the basic memory type
- $F_{\max}$ of 8x8 Wallace multiplier is from 10.58% to 24.43% lower than $F_{\max}(\text{embMem})$ of the basic memory type
- $F_{\max}$ of 16x16 Wallace multiplier is from 55.68% to 67.42% lower than $F_{\max}(\text{embMem})$ of the basic memory type
- $F_{\max}$ of 24x24 Wallace multiplier is from 72.35% to 84.61% lower than $F_{\max}(\text{embMem})$ of the basic memory type

Table 6. $F_{\max}$ of 16x16 multipliers (in MHz)					
FPGA family	Modified Booth-Wallace multiplier	Dadda multiplier	Baugh-Wooley multiplier	Braun multiplier	Vedic multiplier
Cyclone	93	76	68	61	43
Cyclone II	110	95	87	79	62
Cyclone III	133	121	117	109	92
Cyclone IV	136	125	122	115	100
Cyclone V	139	128	125	118	102
Stratix	126	109	106	99	81
Stratix II	184	170	167	163	147
Stratix III	230	216	213	205	188
Stratix IV	233	222	219	212	195
Stratix V	235	225	222	215	199

Table 7. $F_{\max}$ of 24x24 multipliers (in MHz)					
FPGA family	Booth multiplier	Modified Booth multiplier	Wallace multiplier	Modified Wallace multiplier	Booth-Wallace multiplier
Cyclone	15	43	34	47	51
Cyclone II	39	67	58	70	72
Cyclone III	50	79	70	82	85
Cyclone IV	55	84	74	86	88
Cyclone V	59	86	78	88	90
Stratix	45	74	65	78	81
Stratix II	84	113	104	116	119
Stratix III	113	142	133	145	148
Stratix IV	120	148	139	150	152
Stratix V	122	150	141	152	154

- $F_{\max}$ of 8x8 modified Wallace multiplier is from 8.23% to 18.09% lower than $F_{\max}(\text{embMem})$ of the basic memory type
- $F_{\max}$ of 16x16 modified Wallace multiplier is from 54.50% to 61.08% lower than $F_{\max}(\text{embMem})$ of the basic memory type
- $F_{\max}$ of 24x24 modified Wallace multiplier is from 70.19% to 78.73% lower than $F_{\max}(\text{embMem})$ of the basic memory type
- $F_{\max}$ of 8x8 Booth-Wallace multiplier is from 7.84% to 16.28% lower than $F_{\max}(\text{embMem})$ of the basic memory type
- $F_{\max}$ of 16x16 Booth-Wallace multiplier is from 54.11% to 59.27% lower than $F_{\max}(\text{embMem})$ of the basic memory type
- $F_{\max}$ of 24x24 Booth-Wallace multiplier is from 69.80% to 76.92% lower than $F_{\max}(\text{embMem})$ of the basic memory type

Table 8. $F_{\max}$ of 24x24 multipliers (in MHz)					
FPGA family	Modified Booth-Wallace multiplier	Dadda multiplier	Baugh-Wooley multiplier	Braun multiplier	Vedic multiplier
Cyclone	54	38	30	22	10
Cyclone II	73	62	54	46	34
Cyclone III	88	74	66	58	45
Cyclone IV	90	79	70	63	51
Cyclone V	92	81	74	67	55
Stratix	84	69	61	53	41
Stratix II	122	108	100	92	79
Stratix III	151	137	129	121	108
Stratix IV	154	143	135	128	116
Stratix V	156	145	137	130	118

- $F_{\max}$ of 8x8 modified Booth-Wallace multiplier is from 7.45% to 14.93% lower than $F_{\max}(embMem)$ of the basic memory type
- $F_{\max}$ of 16x16 modified Booth-Wallace multiplier is from 53.92% to 57.91% lower than $F_{\max}(embMem)$ of the basic memory type
- $F_{\max}$ of 24x24 modified Booth-Wallace multiplier is from 69.41% to 75.56% lower than $F_{\max}(embMem)$ of the basic memory type
- $F_{\max}$ of 8x8 Dadda multiplier is from 9.59% to 22.17 % lower than $F_{\max}(embMem)$ of the basic memory type
- $F_{\max}$ of 16x16 Dadda multiplier is from 55.88% to 65.61% lower than $F_{\max}(embMem)$ of the basic memory type
- $F_{\max}$ of 24x24 Dadda multiplier is from 71.56% to 82.80% lower than $F_{\max}(embMem)$ of the basic memory type
- $F_{\max}$ of 8x8 Baugh-Wooley multiplier is from 11.37% to 26.25% lower than $F_{\max}(embMem)$ of the basic memory type
- $F_{\max}$ of 16x16 Baugh-Wooley multiplier is from 56.47% to 69.23% lower than $F_{\max}(embMem)$ of the basic memory type
- $F_{\max}$ of 24x24 Baugh-Wooley multiplier is from 73.13% to 86.42% lower than $F_{\max}(embMem)$ of the basic memory type
- $F_{\max}$ of 8x8 Braun multiplier is from 12.74% to 29.86% lower than $F_{\max}(embMem)$ of the basic memory type
- $F_{\max}$ of 16x16 Braun multiplier is from 57.84% to 72.39 % lower than $F_{\max}(embMem)$ of the basic memory type
- $F_{\max}$ of 24x24 Braun multiplier is from 74.50% to 90.04% lower than $F_{\max}(embMem)$ of the basic memory type
- $F_{\max}$ of 8x8 Vedic multiplier is from 15.68% to 37.10 % lower than $F_{\max}(embMem)$ of the basic memory type
- $F_{\max}$ of 16x16 Vedic multiplier is from 60.99% to 80.54% lower than $F_{\max}(embMem)$ of the basic memory type

- $F_{\max}$ of 24x24 Vedic multiplier is from 76.86% to 95.47 % lower than $F_{\max}(embMem)$ of the basic memory type.

The obtained results show that modified Booth-Wallace multiplier has the highest $F_{\max}$. Unlike expressions (2) and (3), where the largest operand is 8 bits, the magnitudes of Gaussian filter coefficients in expression (1) can vary within very wide ranges. Hence, the most important fact in the comparative investigation of the ten multipliers is that $F_{\max}$ of 16x16 modified Booth-Wallace multiplier is from 2.17 times to 2.37 times lower than $F_{\max}(embMem)$ of the basic memory type. Therefore, none of the investigated multiplication algorithms can be used in achieving the goal of ultimate execution speed in FPGA based edge detection which uses Gaussian filtering.

6. Conclusions

Ten different integer multipliers of 8x8, 16x16 and 24x24 inputs are implemented in ten Intel (Altera) FPGA families. They are investigated for $F_{\max}$ under the condition of multiplication being executed within a single clock cycle. Comparative analysis of the obtained results shows that the fastest multiplier is modified Booth-Wallace. $F_{\max}$ of 16x16 imodified Booth-Wallace multiplier is from 2.17 times to 2.37 times lower than $F_{\max}$ of the basic embedded memory type. This fact leads to the conclusion that none of the investigated multiplication algorithms can be used in achieving the goal of ultimate execution speed in FPGA based edge detection which uses Gaussian filtering.

References

- [1]. Abhinav Behla, Abhilasha Gokhaleb, Neelam Sharmac, Design and Implementation of Fast Booth-2 Multiplier on Artix FPGA, International Conference on Smart Sustainable Intelligent Computing and Applications under ICITETM 2020, Computer Science 173, 2020, pp. 140-148
- [2]. Anoop C, Anu Chalil, Implementation and Analysis of Multiplication Algorithms for VLSI Applications using FPGA International Journal of Electrical, Electronics and Data Communication, Volume-5, Issue-9, Sep.-2017, pp. 48-54
- A. C. Mert, H. Azgin, E. Kalali and I. Hamzaoglu, Efficient Multiple Constant Multiplication Using DSP Blocks in FPGA, 2018 28th International Conference on Field Programmable Logic and Applications (FPL), 2018, pp. 331-3313
- [3]. Ayan Palchaudhuri, Rajat Subhra Chakraborty, High Performance Integer Arithmetic, Circuit Design on FPGA, Springer, 2016
- [4]. C. Liu, J. Han and F. Lombardi, A low-power, high-performance approximate multiplier with configurable partial error recovery, 2014 Design, Automation & Test in Europe Conference & Exhibition (DATE), Dresden, 2014, pp. 1-4
- [5]. C. Lin and I. Lin, High accuracy approximate multiplier with error correction, 2013 IEEE 1st International Conference on Computer Design (ICCD), Asheville, NC, 2013, pp. 33-38
- [6]. Dhanabalan and Tamil Selvi, FPGA Implementation of 8-bit Multiplier with Reduced Delay Time, International Journal of Computer and Communication Engineering, Vol. 2, No. 6, November 2013, pp. 412-416
- [7]. Satish S Bhairannawar, Raja K B., Venugopal K R, L. M. Patnaik, Efficient FPGA Based Matrix Multiplication Using Mux and Vedic Multiplier, International Journal of Computers & Technology, Vol 12, No. 5, 2014, pp. 3452-3462

- [8]. E. G. Walters, Array Multipliers for High Throughput in Xilinx FPGAs with 6-Input LUTs, *Computers*, vol. 5 no. 4, 2016
- [9]. E. Kalali and R. Van Leuken, Near-precise parameter approximation for multiple multiplications on a single dsp block, *IEEE Transactions on Computers*, 2021, pp. 1–6
- [10]. Florent de Dinechin, Silviu-Ioan Filip, Luc Forget, Martin Kumm, Table-Based versus Shift-And-Add constant multipliers for FPGAs, *ARITH 2019 – 26th IEEE Symposium on Computer Arithmetic*, 2019, pp. 1-8
- [11]. J. Mody, R. Lawand, R. Priyanka, S. Sivanantham and K. Sivasankaran, Study of Approximate compressors for multiplication using FPGA, 2015 Online International Conference on Green Engineering and Technologies (IC-GET), Coimbatore, 2015, pp. 1-4
- [12]. Manoj M Kamble¹, Dr. Sunita P Ugale, Analysis of different multiplication algorithm and FPGA implementation of recursive barrel shifter method for multiplication, *International Research Journal of Engineering and Technology (IRJET)*, Volume: 03 Issue: 01, 2016, pp. 1141-1147
- [13]. Martin Kumm, *Multiple Constant Multiplication Optimizations for Field Programmable Gate Arrays*, Springer, 2016
- [14]. M. Kumm, J. Kappauf, M. Istvan and P. Zipf, Resource Optimal Design of Large Multipliers for FPGAs, 2017 IEEE 24th Symposium on Computer Arithmetic (ARITH), London, 2017, pp. 131-138
- [15]. M. Langhammer and G. Baekler, High Density and Performance Multiplication for FPGA, 2018 IEEE 25th Symposium on Computer Arithmetic (ARITH), 2018, pp. 5-12
- [16]. Martin Langhammer, Bogdan Pasca, Folded Integer Multiplication for FPGAs, *FPGA '21: The 2021 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, February 2021, pp.160–170
- [17]. Rafferty, C., O'Neill, M., & Hanley, N., Evaluation of Large Integer Multiplication Methods on Hardware. *IEEE Transactions on Computers*, 2017
- [18]. S. Rehman, W. El-Harouni, M. Shafique, A. Kumar, J. Henkel and J. Henkel, Architectural-space exploration of approximate multipliers, 2016 IEEE/ACM International Conference on Computer-Aided Design (ICCAD), Austin, TX, 2016, pp. 1-8
- [19]. S. Ullah, S. Rehman, B. S. Prabakaran, F. Kriebel, M. A. Hanif, M. Shafique, A. Kumar, Area-Optimized Low-Latency Approximate Multipliers for FPGA-based Hardware Accelerators, 2018 55th ACM/ESDA/IEEE Design Automation Conference (DAC), 2018, pp. 1-6
- [20]. S. Ullah, H. Schmidl, S. S. Sahoo, S. Rehman and A. Kumar, Area-optimized Accurate and Approximate Softcore Signed Multiplier Architectures, *Transactions on Computers*, Volume 70 Issue 3, March 2021, pp. 384-392
- [21]. S. Ullah, T. D. A. Nguyen and A. Kumar, Energy-Efficient Low-Latency Signed Multiplier for FPGA-Based Hardware Accelerators, in *IEEE Embedded Systems Letters*, vol. 13, no. 2, June 2021, pp. 41-44
- [22]. Vikas Kaushik and Himanshi Saini, A Review on Comparative Performance Analysis of Different Digital Multipliers, *Advances in Computational Sciences and Technology*, Volume 10 (5), 2017, pp.1257-1272

- [23]. V. Swetha, B. Renuka, D. Lakshman Rao, S. Sridhar, K. Harika, Analysis of Different Multiplication Algorithms & FPGA Implementation, IOSR Journal of VLSI and Signal Processing (IOSR-JVSP) Volume 4, Issue 2, Ver. I, 2014, pp. 29-35
- [24]. Z. Huang, S. Zhang, and W. Wang, An efficient method of parallel multiplication on a single dsp slice for embedded fpgas, IEEE Access, vol. 7, 2019, pp. 101-108

For contacts:

Dimitre Zhivkov Kromichev, PhD
Department of Marketing and International Economic Relations
University of Plovdiv
E-mail: dkromichev@yahoo.com



ULTIMATE EXECUTION SPEED IN FPGA BASED EDGE DETECTION: TECHNOLOGY OF ACCURATE ROUNDING IN INTEGER DIVISION WITH A VARIABLE DIVISOR

Dimitre Kromichev

Abstract: The accuracy of the final integer division result depends on the accuracy of the rounding operation. In the integer division with a variable divisor, the rounding operation is realized by comparing the value of remainder with the value of divisor applying a definite rule. So far, this rule has never been addressed with respect to whether it can provide accurate results under all test conditions. This paper presents the technology of calculating rounding accurately on the basis of analyzing divisor's being an even or an odd number, thus guaranteeing exact integer division results. The analysis of accurate rounding is conducted with respect to the goal of achieving ultimate execution speed in FPGA based edge detection which uses Gaussian weighted average function as a filtering stage.

Keywords: integer division, FPGA, rounding, accuracy, mathematics, edge detection, ultimate execution speed, maximum operating frequency, clock cycles, critical path delay

1. Introduction

Although a number of works has been focused on studying various aspects of FPGA based integer division with a variable divisor [3][4][7][8][9][10][11][12][13][14] so far the problem of the accuracy of rounding operation has never been addressed and analyzed. There is no investigation of the impact of the mathematics of accurate rounding on the speed of FPGA based integer division either. Leading silicon vendors, such as Intel (Altera), do not address the problem of rounding in the guides for implementing the IP integer arithmetic functions [2] and documentation focused on speed [1][5][6].

The objective of this paper is to investigate the mathematics of accurate rounding in FPGA based integer division with a variable divisor. The task is to analyze the impact of accurate rounding on the speed characteristics of integer division with variable divisor in terms of the goal of achieving ultimate execution speed in FPGA based edge detection which relies on the Gaussian weighted average function as a filtering stage. The tools used in the investigation are: Scilab, Intel (Altera) Quartus, TimeQuest Timing Analyzer, ModelSim. The hardware description language is VHDL. Relevant to the analyses and conclusions arrived at are gray scale images.

2. The mathematics of accurate rounding in FPGA based integer division with a variable divisor

Because rounding is adding 0 or 1 to *quotient* to obtain the accurate integer division result, the value of *remainder* satisfies the inequality

$$0 \leq \text{remainder} \leq \text{divisor} - 1 \quad (1)$$

Calculating an accurate integer division result by either incrementing the *quotient* or leaving it unchanged depends entirely on

$$\{1, 2, \dots, \text{divisor} - 1\} > / < / = \frac{\text{divisor}}{2}. \quad (2)$$

Any integer division result $R_{div} \in N$ can be represented as a set of 10 sums

$$Rdiv = \{(Rdiv - 1) + 0.5, (Rdiv - 1) + 0.6, (Rdiv - 1) + 0.7, \\ (Rdiv - 1) + 0.8, (Rdiv - 1) + 0.9, Rdiv + 0, \\ Rdiv + 0.1, Rdiv + 0.2, Rdiv + 0.3, Rdiv + 0.4\}. \quad (3)$$

Mathematically, rounding is replacing a floating point number with an integer on the basis of guaranteeing proportionality between two groups of elements in set $Rdiv$. Thus, two subsets can be formed: $Rdiv1$ and $Rdiv2$. Each of them contains only the floating point addends in the sums from set $Rdiv$. The elements of each subset are placed in ascending order on the basis of magnitude:

$$Rdiv1 = \{0.5, 0.6, 0.7, 0.8, 0.9\} \\ Rdiv2 = \{0, 0.1, 0.2, 0.3, 0.4\}. \quad (4)$$

Then, it is required to compute the accurate ratio between the floating point numbers in set $Rdiv1$ and $divisor$, and set $Rdiv2$ and $divisor$. The step-by-step mechanism is:

Step #1:

$$D = \frac{divisor}{nRdiv}, \quad (5)$$

where

$nRdiv$ is the number of elements in set $Rdiv$.

Step #2:

$$0.5- > 5, 0.6- > 6, 0.7- > 7, 0.8- > 8, 0.9- > 9 \\ 0- > 0, 0.1- > 1, 0.2- > 2, 0.3- > 3, 0.4- > 4$$

$$FD1 = \{D*5, D*6, D*7, D*8, D*9\} \\ FD2 = \{D*0, D*1, D*2, D*3, D*4\}. \quad (6)$$

where

$0.5- > 5, \dots, 0.4- > 4$ is used to denote that the decimal part of each floating point number in sets $Rdiv1$ and $Rdiv2$ is taken as an integer value,

$FD1$ is a set of 5 values resulting from multiplying each of the integer values $\{5, 6, \dots, 9\}$ by D ,

$FD2$ is a set of 5 values resulting from multiplying each of the integer values $\{0, 1, \dots, 4\}$ by D .

Step #3:

$$RD = \frac{divisor}{2}$$

$$SD1 = \{SD5, SD6, SD7, SD8, SD9\} = \\ \{RD - FD1(1), RD - FD1(2), RD - FD1(3), RD - FD1(4), RD - FD1(5)\} \\ SD2 = \{SD0, SD1, SD2, SD3, SD4\} = \\ \{RD - FD2(1), RD - FD2(2), RD - FD2(3), RD - FD2(4), RD - FD2(5)\}$$

(7)

where

$SD1$ is a set of 5 values resulting from the subtraction of each of the 5 values in set $FD1$ from RD

$SD2$ is a set of 5 values resulting from the subtraction of each of the 5 values in set $FD2$ from RD .

Of the ten subtraction results in $SD1$ and $SD2$ there is always a single result which is 0. The value in $FD1$ or $FD2$ for which the subtraction result is 0 is selected as *reference value*.

Step #4:

$$\text{If } remainder \geq reference\ value \text{ then } Racc = quotient + 1. \quad (8)$$

where

$Racc$ is the accurate integer division result obtained after the execution of rounding.

Therefore, the *reference value* is the smallest value in $FD1$ or $FD2$.

Step #5:

$$\text{If } remainder < reference\ value \text{ then } Racc = quotient + 0. \quad (9)$$

Therefore, the value equal to (*reference value* – 1) is the largest value contained in $FD1$ or $FD2$.

In the integer domain of FPGA based edge detection $RD \in N$ is a must. On that basis, the relations between *remainder* and *reference value* in (8) designated as ‘equal to or larger than’ ($\geq$) can lead to division results which are wrong.

Therefore, two cases must be analyzed:

Case #1

$$divisor = 2m, \quad m \in N \quad (10)$$

In order to calculate the *reference value* the following equality must be satisfied:

$$\left(\frac{2m}{s}\right)^t - \left(\frac{2m}{2}\right) = 0, \quad (11)$$

where

s is a constant equal to the number of elements in set $Rdiv$,

t is a variable and $t = \{0,1,2,3,4,5,6,7,8,9\}$.

Hence,

$$reference\ value = m \quad (12)$$

for all possible m . In this way, it is guaranteed that *reference value* $\in N$ in all possible cases. Therefore, in this case,

$$\text{if } remainder \geq m \text{ then } Racc = quotient + 1, \quad (13)$$

Case #2

$$\begin{aligned} divisor &= 2m + 1, \\ m &\in N. \end{aligned} \quad (14)$$

In order to calculate the *reference value* the following equality must be satisfied:

$$\left(\frac{2m+1}{s}\right)t - \left(\frac{2m+1}{2}\right) = 0, \quad (15)$$

where

s is a constant equal to the number of elements in set $Rdiv$,

t is a variable and $t = \{0,1,2,3,4,5,6,7,8,9\}$.

Hence,

$$reference\ value = m + \frac{1}{2} \quad (16)$$

for all possible m . In this way, it is guaranteed that *reference value* $\in R$ in all possible cases. In the integer domain of FPGA computations this is not applicable.

To guarantee accuracy, required are two variants:

Variant #1

Calculate

$$reference\ value = m + 1. \quad (17)$$

In this scenario,

$$\text{if } remainder \geq m+1 \text{ then } Racc = quotient + 1. \quad (18)$$

Therefore, in this scenario, integer division operation requires a total of two divisions and two roundings.

Variant #2

Avoid the rounding mechanism by discarding the second term in $m + \frac{1}{2}$, thus calculating

$$reference\ value = m \quad (19)$$

The critical difference is focused here. In this scenario, it is a must that:

$$\text{only if } remainder > m \text{ then } Racc = quotient + 1. \quad (20)$$

If comparison between *remainder* and *reference value* is executed under the condition *remainder* = m , the rounding and the respective final integer division result will be wrong.

Therefore, in this scenario, an integer arithmetic operation division requires a total of two divisions and a single rounding.

Therefore, mathematically accurate rounding in the integer domain requires two different approaches on the basis of

- *divisor* is an even number
- *divisor* is an odd number.

Otherwise, if rounding is executed on the basis of one and the same comparison function ($>/\geq$) without taking into account the even/odd characteristic of *divisor*, it is guaranteed that exactly 50% of all integer division results will be wrong.

3. The schematic of accurate rounding in FPGA

Listing 1 below presents the VHDL program which executes mathematically accurate rounding. According to this program, rounding is executed as a separate computational operation within the entire computational process of the integer division. For the purpose, the calculated *quotient* and *remainder* are used as inputs and stored in registers before starting the execution of rounding. In this program, rounding takes a single clock cycle to guarantee the mathematically accurate final integer division result..

```
library IEEE;
USE IEEE.std_logic_1164.ALL;
USE IEEE.std_logic_unsigned.ALL;

Entity ROUND is port
(reset:          in std_logic;
clk:            in std_logic;
ena:            in std_logic;
remainder_in:   in std_logic_vector (7 downto 0);
quotient_in:    in std_logic_vector (7 downto 0);
divisor_in:     in std_logic_vector (7 downto 0);
Result_accurate: out std_logic_vector (7 downto 0
));
end entity ROUND;
```

architecture MATH of ROUND is

```
SIGNAL remainder: std_logic_vector (7 downto 0);
SIGNAL quotient:  std_logic_vector (7 downto 0);
SIGNAL divisor:   std_logic_vector (7 downto 0);
SIGNAL result:    std_logic_vector (7 downto 0);
SIGNAL round:     std_logic;
```

BEGIN

```
    Quotient_store:
    process(clk, reset)
    begin
        if (reset = '1') then
            quotient <= (others => '0');
        elsif (clk'event and clk = '1') then
            if ena = '1' then
                quotient <= quotient_in;
            end if;
        end if;
    end process;
```

```
    Remainder_store:
    process(clk, reset)
    begin
        if (reset = '1') then
```

```

        remainder <= (others => '0');
    elsif(clk'event and clk = '1') then
        if ena = '1' then
            remainder <= remainder_in;
        end if;
    end if;
end process;

Divisor_store:
process(clk, reset)
begin
    if (reset = '1') then
        divisor <= (others => '0');
    elsif(clk'event and clk = '1') then
        if ena = '1' then
            divisor <= divisor_in;
        end if;
    end if;
end process;

round <= '1' when ((divisor(0)='1') AND (( '0' & divisor(7 downto 1)) < remainder))
else
    '1' when (((divisor(0)='0')) AND ((( '0' & divisor(7 downto 1)) = remainder) OR (( '0'
& divisor (7 downto 1)) < remainder)))
else
    '0';

result <= quotient(7 downto 0) + ("0000000" & round);

Result_store:
process(clk, reset)
begin
    if (reset = '1') then
        Result_accurate <= (others => '0');
    elsif(clk'event and clk = '1') then
        if ena = '1' then
            Result_accurate <= result;
        end if;
    end if;
end process;
end architecture MATH;

```

Listing 1. VHDL program for mathematically accurate rounding executed within a single clock cycle

The optimal approach to checking whether *divisor* is an even or an odd number is presented in bold in Listing 1. This is the statement which must be used in the FPGA based integer division with a variable divisor in order to guarantee the accuracy of the final result, no matter what concrete integer division algorithm and what radix is implemented.

The RTL design is in Figure 1.

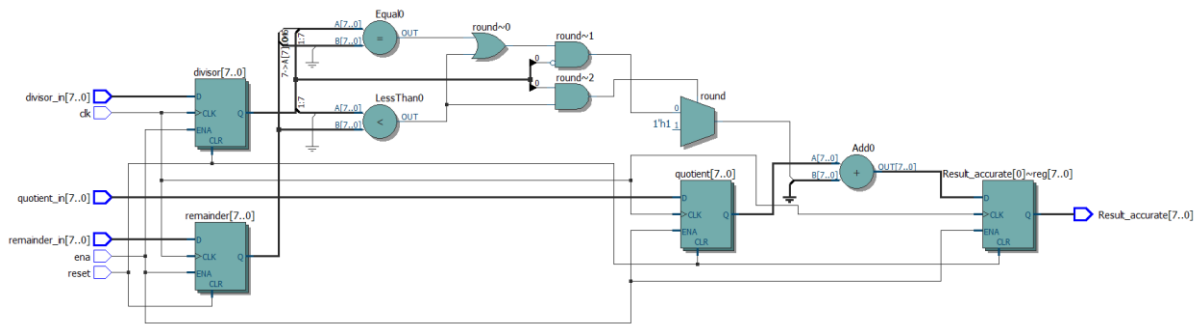


Fig 1. RTL design of mathematically accurate rounding (Source: Intel (Altera) Quartus RTL Viewer)

In the schematic in Figure 1, apart from the registers for storing *quotient* and *remainder*, there is a separate register for storing *divisor*. There is also a register for storing the final integer division result after the accurate rounding has been executed.

4. The impact of mathematically accurate rounding on the speed of FPGA based integer division with a variable divisor

The critical path (in red) of executing mathematically accurate rounding is presented in Figure 2:

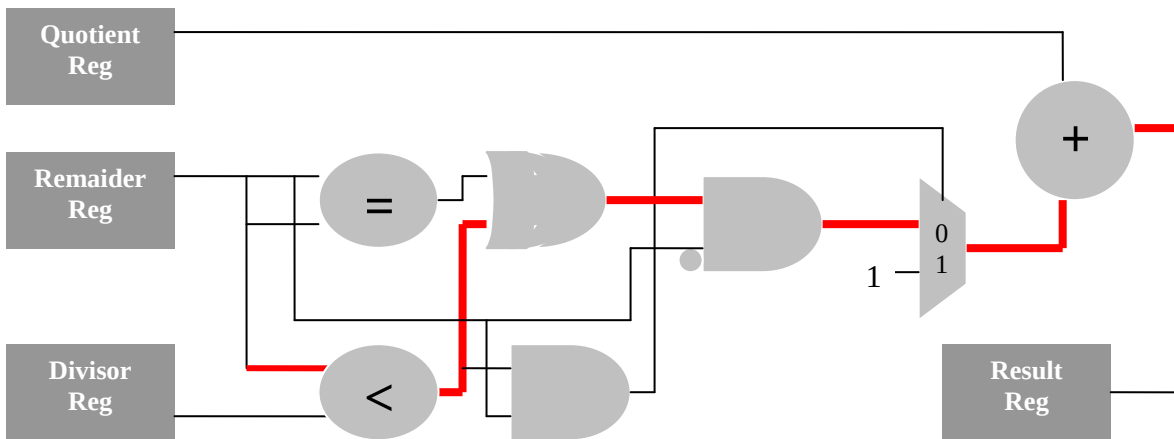


Fig. 2. Critical path of mathematically accurate rounding

The critical path delay of mathematically accurate rounding $T(CP(round))$ includes:

$$T(CP(round)) = T_{comp} + T_{or} + T_{and} + T_{mux} + T_{add} \quad (21)$$

where

T_{comp} is the propagation delay of *divisor* -bit less-than comparator,
 T_{or} is the propagation delay of OR gate,
 T_{and} is the propagation delay of AND gate,
 T_{mux} is the propagation delay of multiplexer,
 T_{add} is the propagation delay of *divisor* -bit adder.

The analysis of Figure 2 and expression (21) shows that the constituents of $T(CP(round))$ fall into two groups:

- T_{or} , T_{and} and T_{mux} are constants for a particular FPGA family
- T_{comp} and T_{add} are variables and depend on the operands' size.

Expression (21) proves that $T(CP(round))$ is a dynamic value which is proportional to the width of *divisor*. In FPGA based edge detection which uses Gaussian weighted average function, the magnitude of *divisor* is defined by the magnitudes of Gaussian filter coefficients. Hence, it follows that (21) presents a critical restriction to the maximum operating frequency of FPGA based integer division with a variable divisor and, therefore, a limitation to the ultimate execution speed of FPGA based edge detection.

Figure 2 and expression (21) show that there are two possible cases:

1) Expression (21) defines the upper limit of maximum operating frequency of FPGA based integer division with a variable divisor which can be achieved when rounding is executed within a single clock cycle. For concrete *dividend* and *divisor* maximum operating frequency depends on the number of clock cycles required to calculate *quotient* and *remainder* in the pipelined divider $npTclk(Qt)$.

$npTclk(Qt)$ is impacted by the critical path delay of rounding according to the inequality

$$T(CP_{\max}(pipeQt)) > T(CP(round)) \quad (22)$$

where

$T(CP_{\max}(pipeQt))$ is the propagation delay of the longest critical path in the pipelined divider calculating *quotient* and *remainder*.

$T(CP(round))$ is the propagation delay of the critical path of rounding.

If (22) is not satisfied, then the maximum value of $npTclk(Qt)$ has been reached. Beyond this maximum value of $npTclk(Qt)$ there is no increase in the maximum operating frequency. Therefore, it is this maximum value of $npTclk(Qt)$ that defines the optimal value of maximum operating frequency of the pipelined divider. Hence,

$$Tclk > T(CP(round)) \quad (23)$$

where

$Tclk$ is the period of the system clock.

Therefore, in case of (23) the maximum operating frequency of integer division with a variable divisor is determined by the critical path delay of the rounding operation. The maximum value of $npTclk(Qt)$ is defined exactly with respect to the concrete integer division algorithm and the maximum input values of *dividend* and *divisor*, and, therefore, the maximum value of *quotient*, which in FPGA based edge detection relying on the Gaussian weighted average function as a filtering stage is $2^8 - 1$. Up to the maximum value of $npTclk(Qt)$, increasing the number of clock cycles drives the clock frequency to a higher value. Beyond that maximum value, increasing the number of clock cycles is not effective according to (22) and (23).

2) Rounding is executed within two clock cycles. In this case, $npTclk(Qt)$ is impacted by the critical path delay of rounding according to the inequality

$$T(CP_{\max}(pipeQt)) > T(CP_{\max}(roundPipe)) \quad (24)$$

where

$T(CP_{\max}(\text{roundPipe}))$ is the propagation delay of the longest critical path in the pipelined rounding.

Hence,

$$T_{clk} > T(CP_{\max}(\text{roundPipe})) \quad (25)$$

Therefore, in case of (25) the maximum operating frequency of integer division with a variable divisor is determined by the longest critical path delay of the pipelined rounding.

The execution of rounding within two clock cycles aimed at increasing the maximum operating frequency leads a maximum value of $npT_{clk}(Qt)$ which is twice as large as the value calculated by applying (22) and (23).

Therefore, in both cases, pipelining is restricted in its application as a tool to increase maximum operating frequency of FPGA based integer division with a variable divisor by the critical path delay of mathematically accurate rounding. Thus, pipelining is not capable to guarantee a constant value of maximum operating frequency under various test conditions.

5. Conclusions

So far the problem of mathematically accurate rounding in FPGA based integer division with a variable divisor has never been addressed. This paper defines the technology for performing mathematically accurate rounding in FPGA based integer division with a variable divisor focused on the goal of achieving ultimate execution speed in FPGA based edge detection which relies on the Gaussian weighted average function as a filtering stage. The mathematics of accurate rounding is subjected to thorough analysis. It is proved that a mathematically accurate rounding and, therefore, a mathematically accurate integer division result can be guaranteed only by taking into account whether the divisor is an even or an odd number. Otherwise, half of all the calculated integer division results will be wrong. Presented is the optimal schematic of executing accurate rounding in FPGA. It is ascertained that the critical path delay of mathematically accurate rounding presents a restriction to the maximum operating frequency of FPGA based integer division with a variable divisor, and, hence, a limitation to the ultimate execution speed of FPGA based edge detection. Determined is the dependence of the maximum operating frequency of FPGA based edge detection which uses Gaussian filtering on the number of clock cycles required by the pipelined divider to calculate the quotient and remainder with respect to the critical path delay of the rounding operation. The two possible cases are considered in the analysis: rounding is executed within a single clock cycle and within two clock cycles. It is ascertained that in both cases pipelining is restricted in its application as a tool to increase maximum operating frequency by the critical path delay of mathematically accurate rounding. Pipelining is not capable to guarantee a constant value of maximum operating frequency of FPGA based integer division with a variable divisor under various test conditions.

References

- [1]. Altera Corporation, FPGA Performance Benchmarking Methodology, White Paper FPGAPBM-1.6, August 2007, ver. 1.6
- [2]. Altera Corporation. Integer Arithmetic IP Cores User Guide, 2014
- [3]. Ayan Palchoudhuri, Rajat Subhra Chakraborty, High Performance Integer Arithmetic Circuit Design on FPGA: Architecture, Implementation and Design Automation, Springer, 2016

- [4]. Dechamps, J., Sutter, G. D., Canto, E., Guide to FPGA Implementation of Arithmetic Functions, Springer Science & Business Media, 2012
- [5]. Intel Corporation, Performing Equivalent Timing Analysis Between Intel's Timing Analyzer and Xilinx's Trace, White Paper, 2021 01047-1.1 White Paper
- [6]. Intel AN 584: Timing Closure Methodology for Advanced FPGA Designs ID: 683145 Version: 2021. 10.08
- [7]. J. Kunn, S. E. Landerswartz, Advanced division algorithms based on the non-restoring with modified delay and the power dissipation corrections, 47th conference on signals and systems and digital computers, 2013, pp. 1461-1465
- [8]. Kilts, S., Advanced FPGA Design. Architecture, Implementation, and Optimization, John Wiley & Sons, Inc., New Jersey, 2007
- [9]. Rohini S. Hongal, Anita DJ, Comparative, Study of Different Division Algorithms for Fixed and Floating Point Arithmetic Unit for Embedded Applications International
- [10]. Journal of Computer Sciences and Engineering IJCSE Volume-4, Issue-9, 2016, pp. 48-54
- [11]. S. Dixit, M. Nadeem, FPGA Accomplishment of a 16- Bit Divider, Computer Science, vol. 3, no. 2, 2017, pp. 140–143
- [12]. T. Drane, W. Cheung and G. Constantinides, Correctly rounded constant integer division via multiply-add, 2012 IEEE International Symposium on Circuits and Systems (ISCAS), 2012, pp. 1243-1246.
- [13]. Uwe Meyer-Baese, Digital Signal Processing with Field Programmable Gate Arrays, Springer-Verlag Berlin Heidelberg, 2014
- [14]. Vladitiu, M., Computer Arithmetic: Algorithms and Hardware Implementations, 2012.
- [15]. Wim Vanderbauwhede, Khaled Benkrid, High-Performance Computing Using FPGAs, Springer Science & Business Media, 2013

For contacts:

Dimitre Zhivkov Kromichev, PhD
 Department of Marketing and International Economic Relations
 University of Plovdiv
 E-mail: dkromichev@yahoo.com



ULTIMATE EXECUTION SPEED IN FPGA BASED EDGE DETECTION: INVESTIGATING THE SPEED PARAMETERS OF ITERATIVE INTEGER DIVISION

Dimitre Kromichev

Abstract: Integer division is an indispensable arithmetic operation for FPGA based edge detection which uses Gaussian weighted average function as a filtering stage. The goal of ultimate execution speed requires that a Gaussian filtered image pixel must be available every clock cycle at the highest possible operating frequency, irrespective of the filter's size and the coefficients' magnitudes. In the speed domain, integer division is a most problematic operation for FPGA based edge detection. The investigation results must define whether iterative division can be used in achieving the goal of ultimate execution speed in FPGA based edge detection. The obtained experimental data are based on ten Intel (Altera) FPGA families.

Keywords: iterative integer division, FPGA, edge detection, ultimate execution speed, maximum operating frequency, clock cycles, input data width

1. Introduction

Division algorithms are traditionally grouped into two main classes: digit-by-digit algorithms and convergence algorithms [6][11][15]. In the digit-by-digit algorithms the results are computed on a digit-by-digit basis, most significant digit first. The convergence algorithms (Newton–Raphson algorithm and Taylor series expansion), require the repeated updating of an approximation to obtain the correct result which leads to significant delays. In [4] the focus is on the hardware implementation of: a) basic digit-recurrence algorithm; b) radix-2 division which encompasses non-restoring, restoring, binary recurrence, binary recurrence with carry-save adder, radix- 2^k recurrence algorithm; c) radix-B algorithm ($B > 2$). Much research is focused on hardware based digit-recurrence division – it is argued that the most widely used algorithms are restoring division and non-restoring division [2][3][4]. It is underlined that with the restoring division, where one bit of the quotient is determined in each iteration, and the non-restoring division, where the result is not stored if the subtraction goes negative, the most important limitation to speed is the propagation delay of the carry rippling through the whole calculation in each iteration [16]. It is pointed out that the execution times of digit-recurrence algorithms are incompatible with the concept of speed due to the carry's delay which is proportional to the size of both dividend and divisor [2][3][5][14]. In [12] analyzed are restoring and non-restoring digit recurrence algorithms, SRT division, multiplicative algorithm, approximation algorithms, CORDIC algorithm and continued product algorithm. In [7] proposed is an n-bit SRT divider for FPGA implementation. In [9] described is a multiply and shift algorithm as being essentially equivalent to multiplying by the inverse of the divisor with successive approximation. Described is a division algorithm in which the basic operation is subtraction. In [8] studied are division algorithms based on the non-restoring method with modified delay and the power dissipation corrections. In [13] proposed is a 16-bit divider using non-restoring method to obtain quotient bits. The quotient value in each iterative step is in binary form, partial remainders are in redundant binary representations. Iterations use carry-save adder. In [10] proposed is the radix-4 division, which only requires half the number of iterations. A functional iterative algorithm employing multiplication as the basic operation is analyzed in [7]. A multiply and shift algorithm centered on replacing the multiplying with the divisor's reciprocal and using successive approximation is proposed in [1].

The objective of this paper is to investigate experimentally the speed parameters of iterative integer division and define whether it can be used in achieving the goal of ultimate execution speed in FPGA based edge detection which relies on Gaussian weighted average function as a filtering stage. The task is to ascertain the dependence of the maximum operating frequency of iterative integer division on the technology of computing quotient and remainder, and the rounding operation – as a purely combinational schematic or pipelined, on the basis of different input data widths. The obtained experimental results must be compared with the maximum operating frequency of embedded memory in order to determine the applicability of iterative division in FPGA based edge detection focused on ultimate execution speed. The experiments are conducted on the basis of ten Intel (Altera) FPGA families. The tools used in the investigation are: Scilab, Intel (Altera) Quartus, TimeQuest Timing Analyzer, ModelSim. The hardware description language is VHDL. Relevant to the analyses and conclusions arrived at are gray scale images.

2. Integer division in FPGA based edge detection

In FPGA based edge detection which uses Gaussian filtering integer division is an indispensable arithmetic operation. The maximum operating frequency of iterative division $F_{\max}(iterDiv)$ is a function of the input data widths. The input data widths are controlled by the two basic approaches to using division in computing the weighted average function:

- The largest dividend is calculated by

$$CC * (2^8 - 1) \quad (1)$$

where

CC is the value of the central coefficient in Gaussian filter.

- The largest dividend is calculated by:

$$NF * (2^8 - 1) \quad (2)$$

where

NF is the value of the normalization factor of Gaussian filter.

Therefore, $F_{\max}(iterDiv)$ depends on two factors:

1) The Gaussian filter according to (1) and (2):

- The magnitudes of coefficients can vary in very wide ranges
- The organization of computations in the weighted average function.

2) The schematic with which the iterative division is implemented:

- Purely combinational schematic
- Pipelined computations.

The schematic of iterative division defines the minimum number of clock cycles required to obtain a mathematically accurate result $nTclk_{\min}(iterDiv)$. Thus, there are two major approaches:

1) The computation of quotient and remainder, and the rounding operation are executed within a single clock cycle. This is the worst case scenario for $F_{\max}(iterDiv)$.

2) The computation of quotient and remainder is realized as a purely combinational schematic within a single clock cycle. Quotient and remainder are stored in registers, and then the rounding operation is executed. Thus, the accurate final result is obtained on the basis of pipelined division.

With focus on the execution of the rounding operation, two possible approaches must be considered:

- Rounding takes a single clock cycle to execute

- Rounding takes 2 clock cycles to execute.

When the computation of quotient and remainder is pipelined the approaches which impact $F_{\max}(iterDiv)$ are:

- Rounding is executed within a single clock cycle together with the last stage of the pipelined computation of quotient and remainder. This is the worst case scenario for $F_{\max}(iterDiv)$ in the pipelined iterative division.
- Rounding is executed separately within a single clock cycle. In terms of $F_{\max}(iterDiv)$, this is effective only if the critical path delay of rounding is smaller than the largest critical path delay in the pipelined computation of quotient and remainder.
- Rounding is executed within 2 clock cycles. In terms of $F_{\max}(iterDiv)$, this is effective only if the largest critical path delay in the pipelined rounding is smaller than the largest critical path delay in the pipelined computation of quotient and remainder.

Because the upper limit of $F_{\max}$ in FPGA based edge detection which uses Gaussian filtering is defined by the maximum operating frequency of embedded memory $F_{\max}(embMem)$, the goal of ultimate execution speed requires that the following inequality must be satisfied:

$$F_{\max}(iterDiv) \geq F_{\max}(embMem) \quad (3)$$

Thus, the task of the investigation is: define the value of $nTclk_{\min}$ which satisfies (3) for the entire range of magnitudes of the dividend calculated using (1) and (2).

3. Investigating $F_{\max}(iterDiv)$ and $nTclk_{\min}(iterDiv)$ of FPGA based iterative division

Methodology:

- The experiments are conducted with digit recurrence radix-2 non-restoring integer division algorithm implemented in VHDL
- With respect to the range of magnitudes of Gaussian filter coefficients, the implemented input data widths of dividend and divisor are: 12x4, 16x8, 24x16, 32x24
- Experiments are conducted using all values within the range of the concrete input data widths of dividend and divisor
- The values of quotient and remainder are obtained by using both a purely combinational schematic and a pipelined computation
- To achieve optimal $F_{\max}(iterDiv)$ the experiments are conducted by executing the rounding operation separately from the computation of quotient and remainder, within a single clock cycle and within two clock cycles
- In the pipelined computation of quotient and remainder the total number of pipelining stages is defined with respect to (3)
- $F_{\max}(iterDiv)$ for the different experimental settings with respect to $nTclk_{\min}(iterDiv)$ are compared to $F_{\max}(embMem)$ of all basic memory types in the targeted Intel (Altera) FPGA families.

The obtained experimental results are presented in Table 1, Table 2, Table 3, Table 4, Table 5, Table 6, Table 7 and Table 8.

4. Analysis of results

The experimental results presented in Table 1, Table 2, Table 3, Table 4, Table 5, Table 6, Table 7 and Table 8 show that:

Table 1. $F_{\max}(iterDiv)$ of iterative divider with inputs 12x4 and rounding executed within a single clock cycle								
FPGA family	$F_{\max}(iterDiv)$ of iterative divider with inputs 12x4 according to the total number of clock cycles required to compute quotient and remainder (in MHz)							Required number of clock cycles to execute rounding
	1 clock cycle	2 clock cycles	4 clock cycles	6 clock cycles	8 clock cycles	10 clock cycles	12 clock cycles	
Cyclone	40	79	186	184	185	184	185	1
Cyclone II	70	134	275	273	272	273	273	1
Cyclone III	77	149	337	335	334	335	335	1
Cyclone IV	81	159	340	338	337	338	338	1
Cyclone V	84	161	343	341	340	341	341	1
Stratix	73	142	319	317	316	317	317	1
Stratix II	109	216	433	431	430	431	431	1
Stratix III	131	252	541	539	538	539	539	1
Stratix IV	135	267	542	540	539	540	540	1
Stratix V	137	269	552	550	549	550	550	1

Table 2. $F_{\max}(iterDiv)$ of iterative divider with inputs 12x4 and rounding executed within two clock cycles								
FPGA family	$F_{\max}(iterDiv)$ of iterative divider with inputs 12x4 according to the total number of clock cycles required to compute quotient and remainder (in MHz)							Required number of clock cycles to execute rounding
	1 clock cycle	2 clock cycles	4 clock cycles	6 clock cycles	8 clock cycles	10 clock cycles	12 clock cycles	
Cyclone	40	79	186	184	277	275	276	2
Cyclone II	70	134	275	273	352	350	351	2
Cyclone III	77	149	337	335	414	412	413	2
Cyclone IV	81	159	340	338	421	419	420	2
Cyclone V	84	161	343	341	430	428	429	2
Stratix	73	142	319	317	402	400	401	2
Stratix II	109	216	433	431	522	520	521	2
Stratix III	131	252	541	539	688	686	687	2
Stratix IV	135	267	542	540	694	692	693	2
Stratix V	137	269	552	550	703	701	702	2

1) For 12x4 iterative divider with rounding executed within a single clock cycle, if the computation of quotient and remainder takes:

- 1 clock cycle, then $F_{\max}(iterDiv) < F_{\max}(embMem)$ from 73.13% to 81.9%

- 2 clock cycles, then $F_{\max}(\text{iterDiv}) < F_{\max}(\text{embMem})$ from 47.25% to 64.25%
- 4 clock cycles, then $F_{\max}(\text{iterDiv}) > F_{\max}(\text{embMem})$ from 7.6% to 8.36% in Cyclone II-V

and Stratix I-V, and $F_{\max}(\text{iterDiv}) < F_{\max}(\text{embMem})$ by 15.83% in Cyclone

6 clock cycles, then $F_{\max}(\text{iterDiv}) > F_{\max}(\text{embMem})$ from 7.27% to 7.69% in Cyclone II-V and Stratix I-V; and $F_{\max}(\text{iterDiv}) < F_{\max}(\text{embMem})$ by 16.74% in Cyclone

Table 3. $F_{\max}(\text{iterDiv})$ of iterative divider with inputs 16x8 and rounding executed within a single clock cycle									
FPGA family	$F_{\max}(\text{iterDiv})$ of iterative divider with inputs 16x8 according to the total number of clock cycles required to compute quotient and remainder (in MHz)								Required number of clock cycles to execute rounding
	1 clock cycle	2 clock cycles	4 clock cycles	6 clock cycles	8 clock cycles	10 clock cycles	12 clock cycles	14 clock cycles	
Cyclone	19	39	75	73	74	73	74	74	1
Cyclone II	34	65	128	127	125	127	128	127	1
Cyclone III	41	77	142	141	140	141	142	141	1
Cyclone IV	44	82	159	158	157	158	159	158	1
Cyclone V	46	88	172	171	171	171	172	171	1
Stratix	37	69	129	128	127	128	129	128	1
Stratix II	57	93	170	169	169	169	170	169	1
Stratix III	68	132	252	251	250	251	252	251	1
Stratix IV	82	159	304	303	302	303	304	303	1
Stratix V	80	157	301	300	300	300	301	300	1

Table 4. $F_{\max}(\text{iterDiv})$ of iterative divider with inputs 16x8 and rounding executed within two clock cycles									
FPGA family	$F_{\max}(\text{iterDiv})$ of iterative divider with inputs 16x8 according to the total number of clock cycles required to compute quotient and remainder (in MHz)								Required number of clock cycles to execute rounding
	1 clock cycle	2 clock cycles	4 clock cycles	6 clock cycles	8 clock cycles	10 clock cycles	12 clock cycles	14 clock cycles	
Cyclone	19	39	75	73	122	120	121	121	2
Cyclone II	34	65	128	127	232	231	231	231	2
Cyclone III	41	77	142	141	269	268	268	269	2
Cyclone IV	44	82	159	158	275	274	274	275	2
Cyclone V	46	88	162	161	279	278	278	279	2
Stratix	37	69	129	128	208	206	206	208	2
Stratix II	57	93	170	169	313	311	312	312	2
Stratix III	68	132	252	251	439	437	436	438	2
Stratix IV	82	159	304	303	443	441	442	443	2
Stratix V	80	157	301	300	438	437	436	438	2

- 8 clock cycles, then $F_{\max}(iterDiv) > F_{\max}(embMem)$ from 7.1% to 7.35% in Cyclone II-V and Stratix I-V, and, $F_{\max}(iterDiv) > F_{\max}(embMem)$ by 16.28% in Cyclone
- 10 clock cycles, then $F_{\max}(iterDiv) > F_{\max}(embMem)$ from 7.27% to 7.69% in Cyclone II-V and Stratix I-V, and $F_{\max}(iterDiv) < F_{\max}(embMem)$ by 16.74% in Cyclone
- 12 clock cycles, then $F_{\max}(iterDiv) > F_{\max}(embMem)$ from 7.27% to 7.69% in Cyclone II-V and Stratix I-V, and $F_{\max}(iterDiv) < F_{\max}(embMem)$ by 16.28% in Cyclone.

Table 5. $F_{\max}(iterDiv)$ of iterative divider with inputs 24x16 and rounding executed within a single clock cycle									
FPGA family	$F_{\max}(iterDiv)$ of iterative divider with inputs 24x16 according to the total number of clock cycles required to compute quotient and remainder (in MHz)								Required number of clock cycles to execute rounding
	1 clock cycle	2 clock cycles	4 clock cycles	6 clock cycles	8 clock cycles	10 clock cycles	12 clock cycles	14 clock cycles	
Cyclone	10	16	28	26	27	26	27	26	1
Cyclone II	17	26	42	41	41	42	41	41	1
Cyclone III	26	36	50	48	49	50	50	49	1
Cyclone IV	30	39	62	60	61	61	62	62	1
Cyclone V	32	43	64	63	64	63	63	63	1
Stratix	20	28	46	45	46	46	45	45	1
Stratix II	42	55	80	79	80	80	79	79	1
Stratix III	50	73	102	100	101	102	102	101	1
Stratix IV	54	79	104	103	103	102	104	104	1
Stratix V	52	75	103	101	102	101	102	102	1

Table 6. $F_{\max}(iterDiv)$ of iterative divider with inputs 24 x16 and rounding executed within two clock cycles									
FPGA family	$F_{\max}(iterDiv)$ of iterative divider with inputs 24x16 according to the total number of clock cycles required to compute quotient and remainder (in MHz)								Required number of clock cycles to execute rounding
	1 clock cycle	2 clock cycles	4 clock cycles	6 clock cycles	8 clock cycles	10 clock cycles	12 clock cycles	14 clock cycles	
Cyclone	10	16	28	26	46	44	45	44	2
Cyclone II	17	26	42	41	70	78	78	79	2
Cyclone III	26	36	50	48	82	81	81	82	2
Cyclone IV	30	39	62	61	105	103	103	104	2
Cyclone V	32	43	64	63	106	104	104	105	2
Stratix	20	28	46	45	79	77	77	78	2
Stratix II	42	55	80	79	139	138	136	137	2
Stratix III	51	73	102	101	172	170	169	170	2
Stratix IV	54	79	104	103	178	177	177	178	2
Stratix V	52	75	105	102	176	175	175	176	2

2) For 12x4 iterative divider with rounding executed within 2 clock cycles, if the computation of quotient and remainder takes:

- 1 clock cycle, then $F_{\max}(\text{iterDiv}) < F_{\max}(\text{embMem})$ from 73.13% to 81.9%
- 2 clock cycles, then $F_{\max}(\text{iterDiv}) < F_{\max}(\text{embMem})$ from 47.25% to 64.25%

4 clock cycles, then $F_{\max}(\text{iterDiv}) > F_{\max}(\text{embMem})$ from 7.6% to 8.36% in Cyclone II-V and Stratix I-V, and $F_{\max}(\text{iterDiv}) < F_{\max}(\text{embMem})$ 15.83% in Cyclone

Table 7. $F_{\max}(\text{iterDiv})$ of iterative divider with inputs 32x24 and rounding executed within a single clock cycle									
FPGA family	$F_{\max}(\text{iterDiv})$ of iterative divider with inputs 32x24 according to the total number of clock cycles required to compute quotient and remainder (in MHz)								Required number of clock cycles to execute rounding
	1 clock cycle	2 clock cycles	4 clock cycles	6 clock cycles	8 clock cycles	10 clock cycles	12 clock cycles	14 clock cycles	
Cyclone	4	10	17	15	16	15	16	15	1
Cyclone II	10	17	28	26	26	27	27	26	1
Cyclone III	19	26	37	36	36	36	37	36	1
Cyclone IV	22	34	43	42	42	43	43	42	1
Cyclone V	26	37	46	45	46	46	45	45	1
Stratix	18	26	32	31	32	32	31	31	1
Stratix II	26	37	48	47	47	48	48	47	1
Stratix III	34	52	68	67	67	68	68	67	1
Stratix IV	39	58	77	78	78	79	78	78	1
Stratix V	37	55	79	76	77	77	76	76	1

Table 8. $F_{\max}(\text{iterDiv})$ of iterative divider with inputs 32x24 and rounding executed within two clock cycles									
FPGA family	$F_{\max}(\text{iterDiv})$ of iterative divider with inputs 32x24 according to the total number of clock cycles required to compute quotient and remainder (in MHz)								Required number of clock cycles to execute rounding
	1 clock cycle	2 clock cycles	4 clock cycles	6 clock cycles	8 clock cycles	10 clock cycles	12 clock cycles	14 clock cycles	
Cyclone	4	10	17	15	22	20	21	20	2
Cyclone II	10	17	28	25	37	35	36	36	2
Cyclone III	19	26	37	35	48	46	47	46	2
Cyclone IV	22	34	40	38	54	52	53	52	2
Cyclone V	26	37	46	44	57	55	56	55	2
Stratix	18	26	32	37	46	44	45	44	2
Stratix II	26	37	48	46	78	73	74	73	2
Stratix III	34	52	68	67	108	106	107	107	2
Stratix IV	39	58	77	76	112	110	112	112	2
Stratix V	37	55	79	77	116	114	116	116	2

- 6 clock cycles, then $F_{\max}(\text{iterDiv}) > F_{\max}(\text{embMem})$ from 7.27% to 7.69% in Cyclone II-V and Stratix I-V, and $F_{\max}(\text{iterDiv}) < F_{\max}(\text{embMem})$ by 16.74% in Cyclone
- 8 clock cycles, then $F_{\max}(\text{iterDiv}) > F_{\max}(\text{embMem})$ from 20.21% to 27.45%
- 10 clock cycles, then $F_{\max}(\text{iterDiv}) > F_{\max}(\text{embMem})$ from 19.63% to 27.25%
- 12 clock cycles, then $F_{\max}(\text{iterDiv}) > F_{\max}(\text{embMem})$ from 19.92% to 27.35%.

3) For 16x8 iterative divider with rounding executed within a single clock cycle, if the computation of quotient and remainder takes:

- 1 clock cycle, then $F_{\max}(\text{iterDiv}) < F_{\max}(\text{embMem})$ from 84.31% to 91.4%
- 2 clock cycles, then $F_{\max}(\text{iterDiv}) < F_{\max}(\text{embMem})$ from 69.21% to 82.35%
- 4 clock cycles, then $F_{\max}(\text{iterDiv}) < F_{\max}(\text{embMem})$ from 40.99% to 66.06%
- 6 clock cycles, then $F_{\max}(\text{iterDiv}) < F_{\max}(\text{embMem})$ from 41.17% to 66.95%
- 8 clock cycles, then $F_{\max}(\text{iterDiv}) < F_{\max}(\text{embMem})$ from 41.17% to 66.51%
- 10 clock cycles, then $F_{\max}(\text{iterDiv}) < F_{\max}(\text{embMem})$ from 41.17% to 66.95%
- 12 clock cycles, then $F_{\max}(\text{iterDiv}) < F_{\max}(\text{embMem})$ from 40.99% to 66.51%
- 14 clock cycles, then $F_{\max}(\text{iterDiv}) < F_{\max}(\text{embMem})$ from 41.17% to 66.51%

4) For 16x8 iterative divider with rounding executed within 2 clock cycles, if the computation of quotient and remainder takes:

- 1 clock cycle, then $F_{\max}(\text{iterDiv}) < F_{\max}(\text{embMem})$ from 84.31% to 91.4%
- 2 clock cycles, then $F_{\max}(\text{iterDiv}) < F_{\max}(\text{embMem})$ from 69.21% to 82.35%
- 4 clock cycles, then $F_{\max}(\text{iterDiv}) < F_{\max}(\text{embMem})$ from 40.99% to 66.06%
- 6 clock cycles, then $F_{\max}(\text{iterDiv}) < F_{\max}(\text{embMem})$ from 41.17% to 66.95%
- 8 clock cycles, then $F_{\max}(\text{iterDiv}) < F_{\max}(\text{embMem})$ from 14.11% to 44.79%
- 10 clock cycles, then $F_{\max}(\text{iterDiv}) < F_{\max}(\text{embMem})$ from 14.31% to 45.7%
- 12 clock cycles, then $F_{\max}(\text{iterDiv}) < F_{\max}(\text{embMem})$ from 14.5% to 45.25%
- 14 clock cycles, then $F_{\max}(\text{iterDiv}) < F_{\max}(\text{embMem})$ from 14.11% to 45.25%

5) For 24x16 iterative divider with rounding executed within a single clock cycle, if the computation of quotient and remainder takes:

- 1 clock cycle, then $F_{\max}(\text{iterDiv}) < F_{\max}(\text{embMem})$ from 89.8% to 95.47%
- 2 clock cycles, then $F_{\max}(\text{iterDiv}) < F_{\max}(\text{embMem})$ from 85.29% to 92.76%
- 4 clock cycles, then $F_{\max}(\text{iterDiv}) < F_{\max}(\text{embMem})$ from 79.8% to 87.33%
- 6 clock cycles, then $F_{\max}(\text{iterDiv}) < F_{\max}(\text{embMem})$ from 80.19% to 88.23%
- 8 clock cycles, then $F_{\max}(\text{iterDiv}) < F_{\max}(\text{embMem})$ from 80% to 87.78%
- 10 clock cycles, then $F_{\max}(\text{iterDiv}) < F_{\max}(\text{embMem})$ from 80.19% to 88.23%
- 12 clock cycles, then $F_{\max}(\text{iterDiv}) < F_{\max}(\text{embMem})$ from 80% to 87.78%
- 14 clock cycles, then $F_{\max}(\text{iterDiv}) < F_{\max}(\text{embMem})$ from 80% to 88.23%

6) For 24x16 iterative divider with rounding executed within 2 clock cycles, if the computation of quotient and remainder takes:

- 1 clock cycle, then $F_{\max}(\text{iterDiv}) < F_{\max}(\text{embMem})$ from 89.8% to 95.47%
- 2 clock cycles, then $F_{\max}(\text{iterDiv}) < F_{\max}(\text{embMem})$ from 85.29% to 92.76%

- 4 clock cycles , then $F_{\max}(\text{iterDiv}) < F_{\max}(\text{embMem})$ from 79.41% to 87.33%
- 6 clock cycles, then $F_{\max}(\text{iterDiv}) < F_{\max}(\text{embMem})$ from 80% to 88.23%
- 8 clock cycles, then $F_{\max}(\text{iterDiv}) < F_{\max}(\text{embMem})$ from 65.49% to 79.18%
- 10 clock cycles, then $F_{\max}(\text{iterDiv}) < F_{\max}(\text{embMem})$ from 65.68% to 80.09%
- 12 clock cycles, then $F_{\max}(\text{iterDiv}) < F_{\max}(\text{embMem})$ from 65.68% to 79.63%
- 14 clock cycles, then $F_{\max}(\text{iterDiv}) < F_{\max}(\text{embMem})$ from 65.49% to 80.09%

7) For 32x24 iterative divider with rounding executed within a single clock cycle, if the computation of quotient and remainder takes:

- 1 clock cycle, then $F_{\max}(\text{iterDiv}) < F_{\max}(\text{embMem})$ from 92.74% to 98.19%
- 2 clock cycles, then $F_{\max}(\text{iterDiv}) < F_{\max}(\text{embMem})$ from 89.21% to 95.47%
- 4 clock cycles, then $F_{\max}(\text{iterDiv}) < F_{\max}(\text{embMem})$ from 84.5% to 92.3%
- 6 clock cycles, then $F_{\max}(\text{iterDiv}) < F_{\max}(\text{embMem})$ from 85.09% to 93.21%
- 8 clock cycles, then $F_{\max}(\text{iterDiv}) < F_{\max}(\text{embMem})$ from 84.9% to 92.76%
- 10 clock cycles, then $F_{\max}(\text{iterDiv}) < F_{\max}(\text{embMem})$ from 84.9% to 93.21%
- 12 clock cycles, then $F_{\max}(\text{iterDiv}) < F_{\max}(\text{embMem})$ from 85.09% to 92.76%
- 14 clock cycles, then $F_{\max}(\text{iterDiv}) < F_{\max}(\text{embMem})$ from 85.09% to 93.21%

8) For 32x24 iterative divider with rounding executed within 2 clock cycles, if the computation of quotient and remainder takes:

- 1 clock cycle, then $F_{\max}(\text{iterDiv}) < F_{\max}(\text{embMem})$ from 92.74% to 98.19%
- 2 clock cycles, then $F_{\max}(\text{iterDiv}) < F_{\max}(\text{embMem})$ from 89.21% to 95.47%
- 4 clock cycles, then $F_{\max}(\text{iterDiv}) < F_{\max}(\text{embMem})$ from 84.5% to 92.3%
- 6 clock cycles, then $F_{\max}(\text{iterDiv}) < F_{\max}(\text{embMem})$ from 84.9% % to 93.21%
- 8 clock cycles, then $F_{\max}(\text{iterDiv}) < F_{\max}(\text{embMem})$ from 77.25% to 90.04%
- 10 clock cycles, then $F_{\max}(\text{iterDiv}) < F_{\max}(\text{embMem})$ from 77.64% to 90.95%
- 12 clock cycles, then $F_{\max}(\text{iterDiv}) < F_{\max}(\text{embMem})$ from 77.25% to 90.49%
- 14 clock cycles, then $F_{\max}(\text{iterDiv}) < F_{\max}(\text{embMem})$ from 77.25% to 90.95%.

Therefore, (3) is satisfied only for 12x4 divider when computation of quotient and remainder takes 4 and more clock cycles, rounding – 1 clock cycle, for Cyclone II-V and Stratix I-V; and 4 and more clock cycles, rounding - 2 clock cycles, for Cyclone II-V and Stratix I-V and 8 and more clock cycles for Cyclone. Divider with inputs 12x4 limits the value of CC in (1) up to 2^8-1 . Results prove that the number of the stages in the pipelined computation of quotient and remainder and $F_{\max}(\text{iterDiv})$ respectively are restricted by the largest critical path delay of rounding.

With respect to the goal of ultimate execution speed in FPGA based edge detection which relies on Gaussia weighted average function as a filtering stage, the two critical factors which define whether the iterative integer division can be used in the Gaussian filtering computations are:

- Very wide range of the magnitudes of Gaussian filter coefficients;
- A Gaussian filtered image pixel must be available at the output of the Gaussian filtering module every clock cycle.

On the basis of these two factors, the analysis of the interdependencies among the size of inputs, $F_{\max}(\text{iterDiv})$, $nTclk_{\min}(\text{iterDiv})$ and $F_{\max}(\text{embMem})$ presented in percentage terms proves

that iterative integer division cannot be used in FPGA based edge detection focused on achieving the goal of ultimate execution speed.

5. Conclusions

Presented is a thorough investigation of the speed parameters of iterative integer division implemented as digit recurrence radix-2 non-restoring divider with inputs 12x4, 16x8, 24x16 and 32x24, and rounding executed with a single clock cycle and within two clock cycles in ten Intel (Altera) FPGA families. The analysis of the obtained experimental results prove that iterative integer division cannot be used in achieving the goal of ultimate execution speed in FPGA based edge detection relying on Gaussian weighted average function as a filtering stage.

References

- [1]. Anusha R., Naganur Ramesh S., Design and ASIC Implementation of Efficient 8-bit Integer Division Algorithms, Annals of R.S.C.B., Vol. 25, Issue 4, 2021, pp. 12463-12474
- [2]. Ayan Palchaudhuri, Rajat Subhra Chakraborty, High Performance Integer Arithmetic Circuit Design on FPGA: Architecture, Implementation and Design Automation, Springer, 2016
- [3]. Carl Hamacher, Zvonko Vranesic, Safwat Zaky, Naraig Manjikian, Computer Organization and Embedded Systems, McGraw-Hill, 2012
- [4]. Dechamps, J., Sutter, G. D., Canto, E., Guide to FPGA Implementation of Arithmetic Functions, Springer Science & Business Media, 2012
- [5]. E. M. Kulisen, Rapid, Division Algorithms of the arithmetic design and its performance parameter comparative analysis, 2017 25th Signal Processing and Communications Applications Conference, 2017, pp. 630- 633
- [6]. Juan Jose Rodriguez Andina, Eduardo de la Torre Arnanz, Maria Dolores Valdes, FPGAs: Fundamentals, Advanced Features, and Applications in Industrial Electronics, CRC Press, [7]. 2017
- [8]. J. Kumari, M. Y. Yasin, Design and Soft Implementation of N-bit SRT Divider on FPGA through VHDL, International Journal of Application or Innovation in Engineering & Management, vol. 3, no. 4, pp. 13–19, Apr. 2015
- [9]. J. Kunn, S. E. Landerswartz, Advanced division algorithms based on the non-restoring with modified delay and the power dissipation corrections, 47th conference on signals and systems and digital computers, 2013, pp. 1461-1465
- [10]. Kilts, S., Advanced FPGA Design. Architecture, Implementation, and Optimization, John Wiley & Sons, Inc., New Jersey, 2007
- [11]. Kurt Wick, Applied Digital Logic Exercises Using FPGAs, Morgan & Claypool Publishers, 2017
- [12]. Roger Woods, John McAllister, Ying Yi, Gaye Lightbody, FPGA-based Implementation of Signal Processing Systems, Second edition, John Wiley & Sons, 2017
- [13]. Rohini S. Hongal, Anita DJ, Comparative Study of Different Division Algorithms for Fixed and Floating Point Arithmetic Unit for Embedded Applications International Journal of Computer Sciences and Engineering IJCSE Volume-4, Issue- 9, 2016, pp. 48-54

- [14]. S. Dixit, M. Nadeem, FPGA Accomplishment of a 16-Bit Divider, Computer Science, vol. 3, no. 2, 2017, pp. 140–143
- [15]. Takagi, N., Kadowaki, S., Takagi, K., A Hardware Algorithm for Integer Division, IEICE Transactions on Electronics, Communications and Computer Sciences, Volume E89-A, [17]. October 2006, pp 2874-2881
- [18]. Uwe Meyer-Baese Digital Signal Processing with Field Programmable Gate Arrays, Springer-Verlag Berlin Heidelberg, 2014
- [19]. Wim Vanderbauwhede, Khaled Benkrid, High-Performance Computing Using FPGAs, Springer Science & Business Media, 2013

For contacts:

Dimitre Zhivkov Kromichev, PhD
Department of Marketing and International Economic Relations
University of Plovdiv
E-mail: dkromichev@yahoo.com



BATTERY PERFORMANCE OPTIMIZATION USING DATA-DRIVEN MACHINE LEARNING

Viktor Mashkov

Abstract: With the world moving towards a more environmentally-friendly future a number of governing bodies have suggested moving away from ICE vehicles, and towards electric alternatives. Electric mobility offers some great advantages, but has severe drawbacks in terms of energy storage. The industry-leading lithium-ion battery cells remain relatively heavy, expensive and have a relatively short useful life. Recent advances in the world of Artificial Intelligence allow for creating machine learning models that help optimize battery performance, extend useful life, allow faster charging, more accurate range indication etc. This paper focuses on State of Charge (SoC) estimation improvements, as range anxiety is one of the largest problems faced by electric mobility. A number of lightweight machine learning models were tested on the NASA li-ion battery dataset and the capabilities of more complex, attention based models were discussed.

Keywords: Electric vehicles, EVs, Machine Learning, Artificial Intelligence, Lithium-ion Batteries

1. Introduction

The World Health Organization states that almost the entire global population (99%) breathes air that exceeds air quality limits, and threatens their health. A record number of over 6000 cities in 117 countries are now monitoring air quality, but the people living in them are still breathing unhealthy levels of fine particulate matter and nitrogen dioxide, with people in low and middle-income countries suffering the highest exposures [1]. Numerous health and environmental agencies report that a large portion of air pollution is caused by the transportation sector, with Internal Combustion Engine (ICE) vehicles being the main culprits. These facts lead to a much-needed change in the technology used for transportation with electric vehicles (EVs) being by far the highest regarded option. Electric mobility has a large number of advantages, such as higher engine efficiency, the ability to reduce air and noise pollution in cities, the potential for creating personalized urban transportation devices (e-bikes, e-scooters, etc.). On the other hand, EVs have a number of setbacks, most of which are related to the way they store energy. Batteries are, to this day, relatively expensive, have relatively low energy density, have a relatively low useful life, and create more pollution in the production process. This makes optimizing their operation vital for ensuring electric mobility has a viable use.

2. State of the problem

The most widely used method for energy storage in EVs are lithium-ion batteries. Their energy density, longevity, cost and overall performance are industry-leading. Specifics of their physical properties, however, make understanding their operation in-depth a challenge. Their discharge voltage curve, for example, remains flat for, in some cases, more than 90% of a full discharge cycle (Fig. 1)[2], which makes estimating State of Charge (SoC) a challenging task. SoC refers to the ratio between the charge stored in the battery currently and the rated battery capacity (Fig. 2).

State of Health (SoH) and Remaining Useful Life (RUL) are further operational characteristics that need to be estimated and understood in order to ensure proper vehicle function. State of Health refers to the ratio between the maximum storable energy in a battery at the moment and the rated battery capacity (usually at the beginning of the battery's life cycle) (Fig. 3).

Remaining useful life is the time after which an item's performance (in this case - a battery) drops below a certain failure threshold (Fig. 4). The three aforementioned battery performance characteristics have a nonlinear nature and depend on a large number of parameters, such as battery chemistry, current and voltage draw, peak current and voltage, ambient temperature, battery temperature etc.

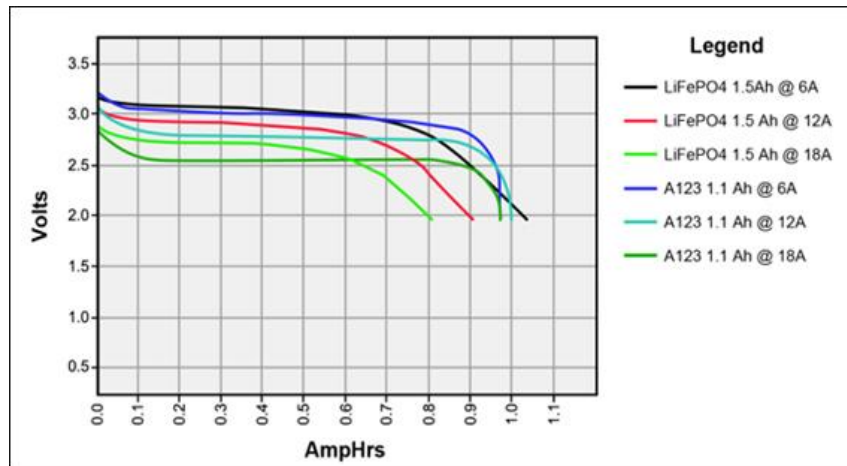


Fig. 1. Discharge voltage of lithium iron phosphate batteries.[2]

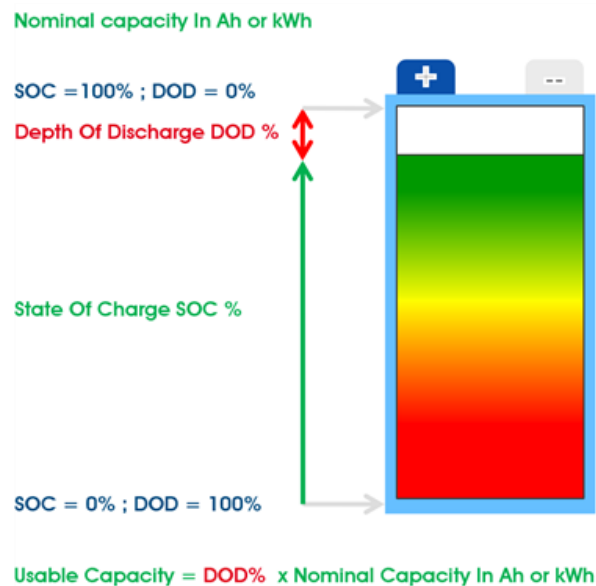


Fig. 2. Battery State of Charge. [3]

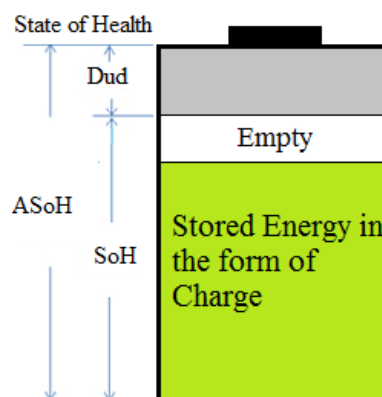


Fig. 3. Battery State of Health. ASoH - Absolute State of Health. [3]

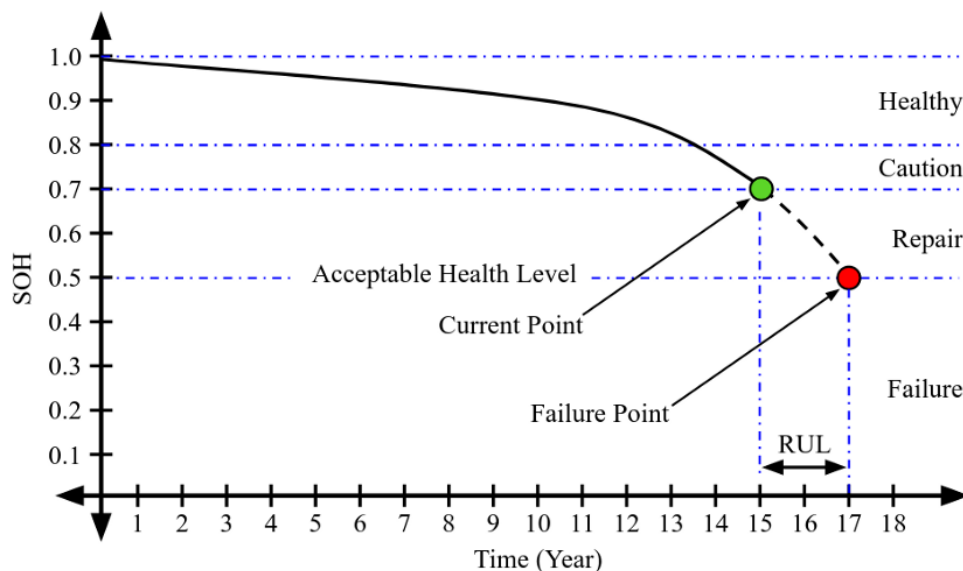


Fig. 4. Remaining Useful Life. [3]

3. Machine learning methods for battery optimization.

Our paper largely focuses on state of charge estimation, as accurately estimating a battery's state of charge (SoC) is key in transportation uses. "Range anxiety" is one of the largest problems that EV users report. In recent years researchers have started using different machine learning methods to model a battery's characteristics and therefore estimating its SoC. These methods vary from lightweight regression models that can be deployed on inexpensive hardware and provide adequate accuracy on simpler battery loading profiles to complex time-series analysis.

Lightweight models

A large number of different lightweight data-driven machine learning models were tested by different authors. In our testing, we compared 16 different lightweight regression machine learning models on the NASA battery dataset using the features load current, battery voltage and battery temperature (Table 1). We compared the models with the following parameters:

- MAE - Mean Absolute Error
- MSE - Mean Squared Error
- RMSE - Root Mean Squared
- R2 - R squared
- RMSLE - Root Mean Squared Log Error
- MAPE - Mean Absolute Percentage Error
- T - Time needed for model training

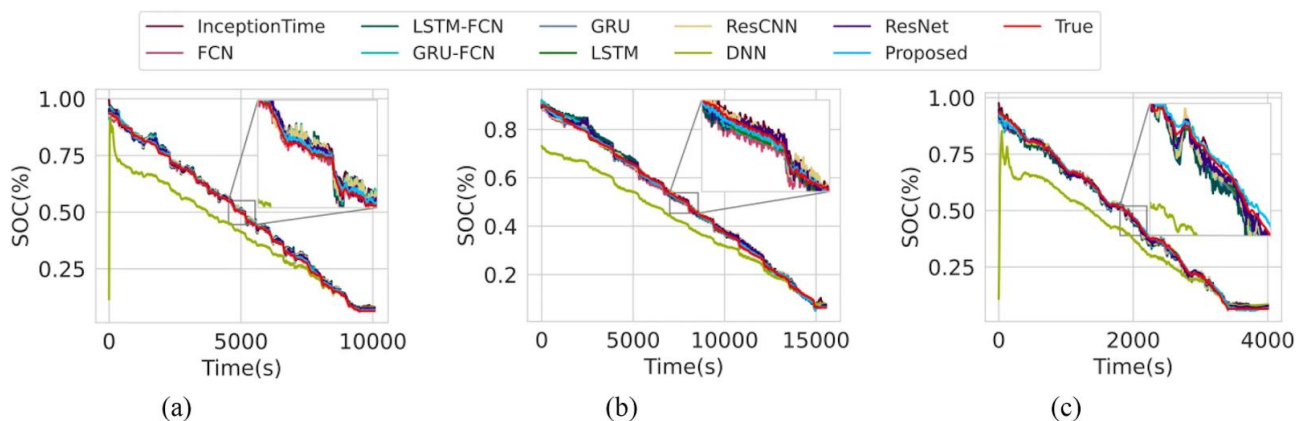
Our research concluded that the best-performing model for our use case was the Light Gradient Boosting Machine, offering the highest accuracy and the lowest required time for training.

Table 1. Sixteen regression models tested for the NASA battery dataset.

Model	MAE	MSE	RMSE	R2	RMSLE	MAPE	T (Sec)
Light Gradient Boosting Machine	1.1748	4.1872	2.0382	0.995	0.0718	7.1296	0.08
Random Forest Regressor	1.1313	4.6208	2.1428	0.9945	0.0652	2.6714	1.318
Extra Trees Regressor	1.1622	5.0014	2.2294	0.9941	0.0668	1.8929	0.873
K Neighbors Regressor	1.6748	6.3703	2.5181	0.9924	0.0877	7.1522	0.076
Decision Tree Regressor	1.454	7.7179	2.772	0.9909	0.0858	1.3091	0.053
Gradient Boosting Regressor	2.016	7.6571	2.7641	0.9909	0.1501	26.4158	0.371
AdaBoost Regressor	3.4634	17.14	4.1393	0.9797	0.3074	145.2611	0.305
Least Angle Regression	5.8084	50.0788	7.0709	0.9409	0.4594	638.1145	0.031
Bayesian Ridge	5.8091	50.0787	7.0709	0.9409	0.4593	637.5488	0.027
Linear Regression	5.8084	50.0787	7.0709	0.9409	0.4594	638.1138	0.017
Ridge Regression	5.8544	50.1542	7.0767	0.9408	0.4509	598.6082	0.026
Huber Regressor	5.3756	60.2591	7.7396	0.9289	0.5367	1042.3756	0.105
Lasso Regression	7.7429	89.289	9.4426	0.8947	0.4424	193.0764	0.028
Orthogonal Matching Pursuit	7.6742	89.2283	9.4387	0.8947	0.4321	184.6607	0.028
Elastic Net	7.9768	90.2916	9.4973	0.8934	0.4736	215.6762	0.028
Passive Aggressive Regressor	8.3438	105.6795	10.1794	0.8752	0.4355	223.8925	0.036

Complex models

Complex machine learning models for SoC estimation are most applicable for uses in larger electric vehicles (electric cars, trucks etc.). They include time-series analysis with attention-based machine learning algorithms, such as Gated recurrent unit (GRU) neural networks, Long-short term memory (LSTM) networks and, in most recent years transformer models, which tend to outperform what was previously available. The authors [5], published in 2021 in the nature journal proposes a self-supervised transformer model, which outperforms currently available machine learning models in state of charge estimation for complex battery loading that mimics standardized driving cycles. Figure 6 shows the researchers' results while comparing to the deep learning models Inception Time, FCN, LSTM-FCN, GRU-FCN, GRU, LSTM, ResCNN, DNN and ResNet.

**Fig. 5.** Proposed model by authors outperforms currently available ML models [5].

4. Conclusion

Data-driven machine learning proves to be the state-of-the-art method for accurate battery performance optimization. Our research on the NASA battery dataset showed that we can implement lightweight regression machine learning models to accurately predict low-feature use cases. These models can be implemented on cheaper hardware, making them useful for urban electric vehicles (e-bikes, e-scooters etc.). Complex loading scenarios, however, require attention-based models. Transformer regression models are industry-leading for time-series forecasting and this use case is no exception. Further research that tests different transformer-based models would be extremely valuable to the world of electric mobility and can possibly provide excellent results not only when predicting SoC, but for SoH and RUL too.

References

- [1]. <https://www.who.int/news/item/04-04-2022-billions-of-people-still-breathe-unhealthy-air-new-who-data>
- [2]. Battery University. <https://batteryuniversity.com/>
- [3]. Battery State of Charge (SOC), Wind & Sun, [http://www.windandsun.co.uk/information/batteries/battery-state-of-charge-\(soc\).aspx#.Y1_NFXbP3IU](http://www.windandsun.co.uk/information/batteries/battery-state-of-charge-(soc).aspx#.Y1_NFXbP3IU)
- [4]. Beny, M.A., Fajar, M.N.R. (2019). The Development of Remaining Useful Life Prediction System on VRLA Battery using Support Vector Regression. Bandung: Institute Technology Bandung
- [5]. Hannan, M.A., How, D.N.T., Lipu, M.S.H. et al. Deep learning approach towards accurate state of charge estimation for lithium-ion batteries using self-supervised transformer model. Sci Rep 11, 19541 (2021). <https://doi.org/10.1038/s41598-021-98915-8>

For contacts:

Assistant Prof. Viktor Mashkov
Computer Science & Engineering Department
Technical University of Varna
E-mail: viktor.mashkov@tu-varna.bg



WORKFLOW ADAPTATION FOR INTELLIGENT HUMAN-MACHINE INTERFACES

Miroslav Markov

Abstract: Following the concept of Intelligent Human-Machine Framework (iHMIfr) developed and presented recently, the paper examines the processes inherent for the concept as well as their chronology and relations. A four-layer roadmap: sensing – monitoring – decision – adaptation is used for decomposition of the interaction process, aimed at achieving of common approach for development of intelligent human-machine interfaces. The interaction trinity paradigm embedded in the concept, along with multimodality of system inputs and outputs define a combination of processes, which - related to the specificities of the particular field of implementation - enhance the creation and development of iHMI architectures.

Keywords: Human-machine interface, HMI, Human-computer interface, HCI, Human-robot interface, HRI, intelligent, adaptive, emotion-sensitive, affective, cognitive, human-centered interfaces.

1. Introduction

1.1. Analysis of previous related works

The world of the so-called intelligent machines grows extremely fast during the last decade. Many of the proposed and created exemplars definitely deserve to be addressed as intelligent, even though it is very hard for their intelligence to be measured and compared. Of course, the interaction capabilities of a machine are the strongest criterion, but since the field of human-machine interaction is quite broad, the intelligence could be expressed in myriad of ways.

One of the typical expressions of intelligence is the adaptability of the machine. In this regard, the adaptive human-machine interfaces comprise the widest category of approaches for achievement of intelligence. Basically, the adaptation of the machine behaviour corresponding to any human-related factor is a property of these interfaces. The completeness and appropriateness of the factors considered, then, could be a good point for assessment of the degree of adaptability. Workload, situation awareness and autonomy [1] are often researched as most important issues in that direction. As a consequence, broad variety of categorical definitions of HMIs could be found in the scientific literature – emotion-sensitive, affective, cognitive, human-centered, autonomous etc.

Various approaches for adaptation have been developed, depending on the application area [2-18]. In most cases, the proposed solution was focused either on the specific interaction between the human user and the machine (e.g. chatbots and voice assistants), specific mutual tasks (collaborative robots), or on the development of autonomous systems (drones, vehicles, etc.), also known as cyber-physical systems (CPS). The adaptation process for the first group usually captures the user's intent and aims to respond reasonably to resemble a human-to-human interaction [2, 3]. Technologies often support this for enhancement of the user's experience, such as avatars [5], lip-synching [6, 7], and generation of nonverbal behavior [8, 9] in order for the agent to be endowed with emotional intelligence [10] and so on. The second group develops the adaptation process around a particular task, which should be implemented mutually by humans and robots [11, 12] or by human and specific software agents [13, 14, 15]. In this case, the adaptation of the machine is mainly dedicated to achieving high accuracy of the operation, high efficiency, low error rate, minimization of the risk of injuries, etc. – features dependent on the extent of collaboration of the human-machine team. In previous research, such interaction is often represented by joint action paradigms like Social Simon Task [16]. Both groups mentioned so far mainly utilize a human-centered approach. Unfortunately, a tiny number of the research studies incorporate the impact of

the context on the human's abilities and current state. On the other hand, the third group (CPSs) [17, 18], aiming to develop maximum autonomy of the systems, perceives the context state as a target that should be dynamically recognized and offset by adaptive actions. In some cases, such strategy puts the human aside from the center and often, to a big extent even regards the users as part of the context.

1.2. Contribution of this paper

The current paper presents the behavior adaptation workflow as a combination of processes to achieve intelligent Human-Machine Interfaces as an intrinsic property of iHMIfr. In this regard, the concept of intelligent Human-Machine Interface framework (iHMIfr) [19] is followed and further elaborated. The additional development of the idea, as well as its presentation and examination from the point of view of a chronological sequence of processes, aims at conceptual formulation of a common approach for development of HMIInterfaces in different areas, independent from the level and type of the particular technologies utilized for implementation of each process. The novelty laid down in the paper consists of concatenation of the two pillars inbuilt in iHMIfr, namely – the interaction trinity “human-machine-context” and the multi-modal approach with regard to both – inputs and outputs of the system with a systematic breakdown of the processes inherent for the particular interaction.

2. Workflow Adaptation for Intelligent Human-Machine Interfaces

2.1. The Intelligent Human-Machine Interface Framework

The Intelligent Human-Machine Interface Framework (iHMIfr) generalizes the formulated vision concerning adaptive Human-Machine Interfaces [19]. In the current contribution, a four-layer roadmap of the information flow is presented, consisting of sensing – monitoring – decision – adaptation layers, and supposed to drive the machine behavior and adapt it according to the individual human needs.

There are two central concepts founding the framework. The first one is related to its multimodality – concerning both the sensors and the actuators of the system, making it more accurate and flexible in its field of implementation. The second one encompasses the processing of signals, states, and information in separate flows, passing across the framework layers and considering three main factors – the human state of mind, the dynamics in the implementation of a specific task, and the impact caused by various context-related circumstances.

2.2. Workflow Adaptation

The iHMIfr discussed above is founded on the interaction trinity "human-machine-context" and, to a considerable extent, joints the approaches mentioned above in a common framework for the development of human-machine (HMI) – including human-robot (HRI) and human-computer (HCI) interfaces. Specifically, the overall workflow is organized in the four layers (sensing, monitoring, decision, adaptation) shown in Fig.1.

Sensing Layer

The Sensing Layer consists of two interrelated and mutually dependent components – the Multi-modal Signal Acquisition and Processing Module (MMSAPM) and the Classification Module

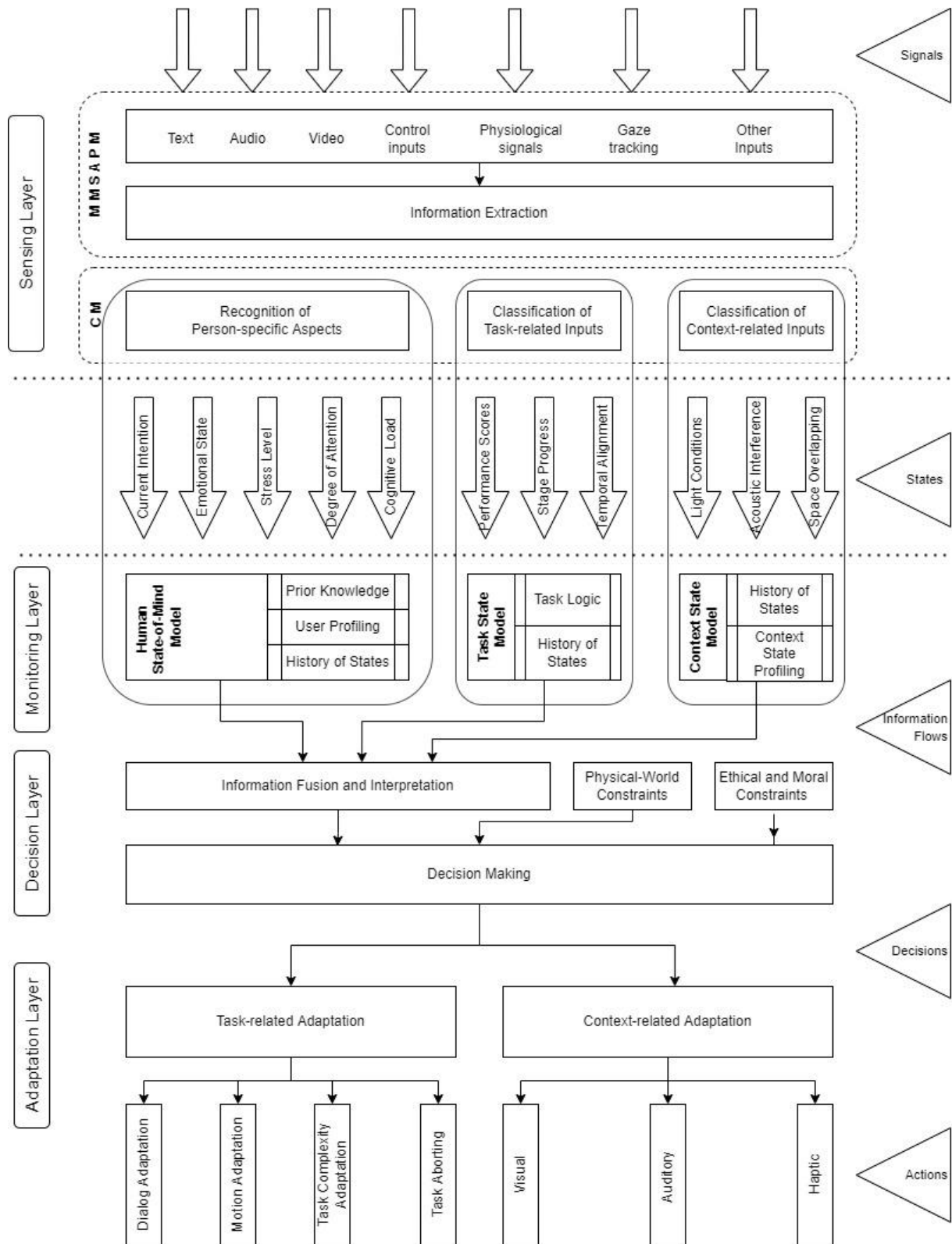


Fig.1. Workflow of iHMIfr

(CM). This layer aims to ensure real-time monitoring of various parameters related to a particular HMI ecosystem and transform them into states, determining each side's behavior. Here, the euphemism HMI ecosystem relates to all types of HMI configurations and includes at least three sides: (i) the machine (robot/ computer) itself - represented by the task it has been assigned, (ii) the

human operator with his/her features, and (iii) the surrounding environment, known as context, with its changing conditions. These three sides influence each other in cooperative or collaborative tasks.

The core function of MMSAPM is to collect a variety of signals relevant to a specific situation, which are reliable enough to reflect different aspects of the interaction. Multimodality is the key word here, and even though we have focused on a set of input signals in this framework, the list is not exhaustive and could be complemented. Furthermore, the information derived from these signals could be overlapping, which is desirable regarding the validation of a particular state and could be very useful for overcoming ambiguous or uncertain situations. Last but not least, using different input modalities allows such interactions to be adopted successfully in environments that require wearing specific safety clothing or attributes like glasses, gloves, helmets, etc.

While the utilization of inputs like text and different audio and video signals obtained from various devices is crucial for interaction, in our days, it could be defined as a commodity. In the iHMI framework, the stress level is considered an essential item in the ecosystem. Hence, the use of signals allowing machines to interpret human intention better and sense different aspects of the human state of mind is of particular interest and plays a vital role in our research. Control inputs like the position of the mouse marker or signals from the gaze tracker could be very informative for detecting human attention during the implementation of a task. Furthermore, physiological signals (such as EDA / GSR; PPG; ECG; EEG, etc.) are widely used to infer the degree of attention at a specific time and could serve as a base for estimating a human's emotional state and momentous stress level, cognitive load, and fatigue. For all these estimations to come true, the input signals undergo signal processing, feature extraction and selection, and information abstraction. The same holds for text analysis, where natural language processing (NLP) is used for guesstimating the human's intention.

The classification module (CM) transforms all the processed signals into some categories or states. The CM serves as a primary distributor of the input data into three specific streams: one – connected to recognition of human's current state of mind; second – connected to dynamic determination of task-specific states, and third – focused on real-time capturing of context-related states.

Monitoring layer

The Monitoring layer is fed with the outputs from the sensing layer – namely particular human-related, task-related, or context-related states. Its principal function is the real-time maintenance of three models, representing respectively the human state of mind, the task implementation, and the context dynamics. The Human State-of-Mind Model monitors the current human-related states and keeps a memory of the past. The prior knowledge inbuilt in the model (e.g., age, gender, level of expertise, etc.) serves as a base for elaborating user profiles relevant to the specifics of the interaction. The model generalizes all the available information and outputs a prediction for human behavior. Compared to past dynamics, each change in a particular input state can indicate a continuation or break of the current tendency. Different regression techniques are used to determine trends and subsequent extrapolation of all the monitored states. The user profiling enables the system to weigh all the changes and estimate the behavior more accurately, considering the person-specific features.

The Task State Model also uses prior knowledge, which contains the inherent logic of the task – its possible states and sequence. It also keeps the memory of the past, allowing the system to dynamically recalculate and assess performance parameters like Score, Temporal Alignment, or Stage Progress. This assessment allows a comparison between current and desired/expected parameters, which gives insights into how successful the task implementation is and hence the need for subsequent adaptation.

The functioning of the Context State Model follows a similar structure: monitoring the context-related input states and keeping their memory. The changes associated with these states

represent the variability of particular context circumstances, which also serve as a base for prediction.

Decision Layer

In order to combine all three outputs in the shape of information flows from the Monitoring layer, an entity resembling a virtual funnel is necessary. Information Fusion dynamically transforms the incoming information flows from the three abovementioned models into drafted decisions. The modular approach inbuilt in the iHMIfr framework, however, allows flexibility and interchangeability of alternative technologies, which on the other hand, allows constant development and progressive improvement of the decision capability. The decision-making process takes place as the drafted decisions are weighed against existing or imposed constraints. The constraints, in general, have two forms – restrictions and limitations. A rule-based approach could be applied to provision exclusions (e.g., in cases of emergency or force majeure).

Two types of constraints could be distinguished and involved in a decision – (i) related to the physical world (e.g., the minimum distance between human and robot during collaboration) and (ii) the moral and ethical constraints, which both the human and the machine are expected to respect.

The resulting decisions, coming out from the Decision Layer, are also distinguished in two aspects – decisions concerning the implementation of the task and those related to the context, allowing different adaptation strategies to be implemented in each direction.

Adaptation Layer

The Adaptation Layer (cf. Fig.1) consists of two parallel adaptation processes – one related to the task and one – to the context. Both processes could be considered autonomous management systems are exercising control within their scope. The output consequently consists of actions configuring two different adaptation strategies.

The task-related adaptation strategy involves actions aimed at modeling different aspects of the task in conjunction with the current human needs to achieve optimal performance, safety, and satisfaction from the interaction. Such actions include adapting the dialog flow, so the machine can respond adequately to any change in the complex multimodal inputs, in that number - undertaking the role of the proactive side when necessary. Another example is the adaptation of machine kinematics for constant prevention of undesired collisions and achievement of better accuracy in its motions. Adaptation of the Task Complexity could be achieved by enabling the system to set different levels of tutoring functionalities as a response to perceived fatigue or hesitation due to a particular emotional state, stress, or lack of attention from the human side. In cases of not finding better adaptive action, task aborting could be applied as most reasonable.

Context-related adaptation strategy consists of actions aimed at overcoming the distractive influences of the context (e.g., unfavorable light conditions or noisy environments) upon the human operator on the one hand, as well as responding to perceived human needs deriving from particular states of mind (e.g., reducing the volume of the information being submitted to the human in case of recognized significant level of cognitive load. Another reason for reaction as part of this strategy could be the reaching of unwanted or dangerous states pertained to the implemented task. The adaptation could be applied through the most appropriate channel for the situation – visual, auditory, or haptic as appointed from the decision layer or as a combination of simultaneous or queues of actions via alternative channels.

2.3. Discussion

In general, the inherent components of each workflow could be described as inputs, transformation, and outputs – of course, all of them varying in broad ranges, depending on the

implementation. In the case of the current study, the inputs collect information from the three sources participating in the interaction – human, machine (or its task properties and states), and context. In that order of thought, each of these inputs is independent and could be added, removed, or modified to ensure the system's operability. Consequently, each HMI interface developed through the framework could be equipped with the most appropriate and optimal set of inputs. Furthermore, the sensors used for data acquisition concerning the human and the context states and the technologies used for interpretation of the task states (which also should be treated as internal inputs) could be selected in the range from expensive state-of-the-art hi-tech solutions, ensuring maximum accuracy (e.g., sensors for physiological signals collection in conditions of the high-risk environment) to low-cost devices (e.g., for detection of light, noise or temperature conditions – as part of the context assessment).

Similarly, various machine learning techniques for classification and regression could be implemented concerning the transformation processes – state recognition, profiling, information fusion, decision making, etc., and various relevant solutions could be used regarding the outputs to the user.

These decisions are made after aligning with the contextual properties and identifying the specificities of the agent tasks. The future work will develop an experimental, context-based set, serving as a proof of concept to validate the iHMIfr and the workflow architecture presented in the current paper.

3. Use case

As a practical implementation of all the written above we suggest the following use case, based on human-machine collaborative task and more precisely, task requiring shared space and actions presuming the occurrence of collisions between the human and the robot. Of course, such collisions are highly dangerous and undesirable as they could potentially lead to injury or even death of the participating human operator. Let's assume that aside from any shared action schedule, defining the proper time and duration of any human and robot assigned actions the robot can dynamically evaluate and interpret the human emotional state, current level of attention and cognitive fatigue.

Multimodality of the inputs, implemented by usage of audio (microphone), visual (camera) and physiological signal (wearable) devices ensures robot's sensibilities. After further processing of the input signals the robot perceives different human states – recognized, measured and associated with a certain class. Any abnormal situation (state) recognized, by the robot leads to implementation of particular adaptive strategy forcing the machine to change its behavior. The adaptation of the behavior involves different output channels like auditory (alarm, verbal warning), visual (light signal, notification on the display) or in case of perceived danger – change of the kinematics of the robot (reducing the speed or pausing, or even aborting the current task).

Similarly, when a situation recognized as change of the context occurs – let's say increase of the environmental noise (which potentially deteriorates the human's ability to hear and understand the auditory information), the corresponding to this state strategy unlocks and the robot adapts the interaction process by respective visual notification (blinking lights, larger fonts, etc.), haptic signal (vibration), or in case of danger – again reduction of the speed or stop of the implementation.

Additionally, the product of the mutual human-robot team work (referred to as performance) is constantly assessed and in case of deviations from the normal parameters again respective adaptation strategy comes in play, so that to change the task workflow in a way ensuring more effective or safer working process (change of the speed, imposition of break, aborting the task, etc.).

4. Conclusion

The current paper presents the workflow inbuilt in the Intelligent Human-Machine Framework iHMIfr presented in [19]. The workflow processes follow a four-layer roadmap: sensing – monitoring – decision – adaptation. Through multimodal inputs, providing sensing abilities of the system, and classification technics, different human-, task- and context-related states are recognized and monitored, allowing appropriate profiling. Processes of information fusion and real-time decision-taking control and implement the execution of adequate adaptive strategy utilizing various output channels.

5. Acknowledgement

This research was supported by the Bulgarian National Science Fund (BNSF), with grant agreement FNI №KP-06-N37/18, entitled "Investigation on intelligent human-machine interaction interfaces, capable of recognizing high-risk emotional and cognitive conditions."

References

- [1]. Blakeney, R. "A Systematic Review of Current Adaptive Human-Machine Interface Research", Embry-Riddle Aeronautical University, UNSY 691 Graduate Capstone Paper, 26.09.2020, <https://www.ryanblakeney.com/home/a-systematic-review-of-current-adaptive-human-machine-interface-research>
- [2]. Mathew, A., V. Rohini, J. Paulose, "NLP-based personal learning assistant for school education.", International Journal of Electrical and Computer Engineering (IJECE), Vol.11, No.5, pp. 4522 - 4530, October 2021
- [3]. Zamanirad, Sh., B. Benatallah, C. Rodriguez, M. Yaghoubzadehfard, S. Bouguelia, and H. Brabra, "State Machine Based Human-Bot Conversation Model and Services", Lecture Notes in Computer Science, vol 12127. Springer, Cham, 2020
- [4]. Beinema, T., D. Davison, D. Reidsma, O. Banos, M. Bruijnes, B. Donval, Á. F. Valero, D. Heylen, D. Hofs, G. Huizing, R. B. Kantharaju, R. Klaassen, J. Kolkmeier, K. Konsolakis, A. Pease, C. Pelachaud, D. Simonetti, M. Snaith, V. Traver, J. van Loon, J. Visser, M. Weusthof, F. Yunus, H. Hermens, and H. op den Akker, "Agents United: An Open Platform for Multi-Agent Conversational Systems", IVA '21, September 14–17, 2021, Virtual Event, Japan
- [5]. Angga P., W. Fachri, A. Eleanita, Suryadi, R. Agushinta, "Design of Chatbot with 3D Avatar, Voice Interface, and Facial Expression", 2015 International Conference on Science in Information Technology (ICSITech), pp. 326-330, 2015
- [6]. Oneata, D., B. Lorincz, A. Stan and H. Cucu, "FlexLip: A Controllable Text-to-Lip System", Sensors, Vol. 22, Issue 11, 2022
- [7]. Fan, Y., Zh. Lin, J. Saito, W. Wang, T. Komura, "Joint Audio-Text Model for Expressive Speech-Driven 3D Facial Animation.", Proceedings of the ACM on Computer Graphics and Interactive Techniques, Vol. 5, pp.1 – 15, 2022
- [8]. Aburumman, N., M. Gillies, J. A. Ward, A. F.de C. Hamilton, "Nonverbal communication in virtual reality: Nodding as a social signal in virtual interactions", International Journal of Human-Computer Studies, Vol. 164, Art. 102819, 30 March 2022
- [9]. Hartholt, A., D. Traum, St. Marsella, A. Shapiro, G. Stratou, A. Leuski, L. Morency, J. Gratch, "All Together Now, Introducing the Virtual Human Toolkit.", Nature, vol. 418 6897, pp. 465, 2013
- [10]. Stringfellow, C., D. Gupta, "An Emotionally Intelligent Tutoring System", 2017 Computing Conference, pp. 1099-1107, 2017

- [11]. Lucci, N., A. Monguzzi, A. M. Zanchettin, P. Rocco, "Workflow modeling for human-robot collaborative assembly operations.", *Robotics and Computer-Integrated Manufacturing*, Vol. 78, Art. 102384, 2022
- [12]. Tassi, Fr., E. De Momi, A. Ajoudani, "An adaptive compliance Hierarchical Quadratic Programming controller for ergonomic human-robot collaboration.", *Robotics and Computer-Integrated Manufacturing*, Vol.78, Art. 102381, 2022
- [13]. Kostolani, D., S. Schlund, "Don't replace me, assist me!: Human-Machine Symbiosis as a Way Out of Europe's Labour Crisis", *Conference: ACM CHI22, Workshop paper for "Outsourcing Artificial Intelligence: Responding to the Reassertion of the Human Element into Automation"*, April 2022
- [14]. Lee, S., M. Jeon, "A systematic review of functions and design features of in-vehicle agents.", *International Journal of Human-Computer Studies*, Vol.165, Art. 102864, 2022
- [15]. Oestreich, H., Y. Bröker, S. Wrede, "An AdaptiveWorkflow Architecture for Digital Assistance Systems", *The 14th Pervasive Technologies Related to Assistive Environments Conference*, pp. 177–184, June 2021
- [16]. Stenzel, A., E. Chinellato, M. A. Bou, Á. P. del Pobil, M. Lappe & R. Liepelt "When humanoid robots become human-like interaction partners: co-representation of robotic actions.", *Journal of experimental psychology: Human perception and performance*, Vol. 38, Issue 5, pp. 1073–1077, 2012
- [17]. Wolf, W., G. Tech, "Cyber-physical Systems", *Computer*, pp.88 – 89, April 2009
- [18]. Jazdi, N. "Cyber-Physical Systems in the Context of Industry 4.0", *2014 IEEE International Conference on Automation, Quality and Testing, Robotics*, pp. 1-4, 2014
- [19]. Markov, M., T. Ganchev, "Intelligent human-machine interface framework", *Proceedings of ISER International Conference, Dublin, Ireland*, pp. 31-36, August, 2022

For contacts:

Assistant Professor, Miroslav Markov
 Department of Software and Internet Technologies
 Technical University of Varna
 E-mail: m.markov@tu-varna.bg



РАЗПОЗНАВАНЕ НА СЪДРЕЧНОСЪДОВИ ЗАБОЛЯВАНИЯ ЧРЕЗ НЕВРОННИ МРЕЖИ

Димитър В. Кръстев

Резюме: Публикацията съдържа анализ на експеримент за изчисляване на успеваемостта при класифициране на сърдечносъдови заболявания чрез дълбока невронна мрежа. За изследването са използвани реални данни, публикувани от University of California, Irvine. Точността на резултатите от експеримента е $\approx 90\%$.

Ключови думи: класификация, машинно обучение, невронни мрежи, сърдечносъдови заболявания.

CARDIOVASCULAR DISEASE RECOGNITION THROUGH DEEP NEURAL NETWORKS

Dimitar V. Krastev

Abstract: The publication contains an analysis of an experiment to calculate the success rate of cardiovascular disease classification using a deep neural network. The study used real data published by the University of California, Irvine. The accuracy rate of the experimental results is $\approx 90\%$.

Keywords: classification, machine learning, neural networks, cardiovascular disease.

1. Увод

С подобрението на компютърната мощност през последните десетилетия множество технологии са интегрирани към части от човешкото общество. Сферата на медицината е една такава част, където правилната диагностика на един пациент може да бъде разликата между неговото оцеляване или смъртта му. Компютърно асистираната диагностика е произлязла с цел да подпомогне бързото и точно диагностициране на пациентите и увеличаване на базовото качество на диагнозата чрез предоставяне на второ мнение спрямо диагностициращия лекар.

Проведен е експеримент за разпознаване на сърдечносъдови заболявания чрез помощта на невронни мрежи. Задачата се разглежда като класификационна, където входните данни на всеки пациент се обработват от невронна мрежа, която връща резултат, символизиращ увереността за наличие на заболяване.

2. Изложение

2.1. Използвани данни

За провеждането на експеримента са използвани данни, събрани от няколко медицински институции и публикувани от University of California, Irvine [1]. Те представляват 920 различни записа на резултати от пациенти със и без наличие на сърдечносъдови заболявания. За всеки запис са заложили 75 атрибута, но някои записи не съдържат данни за определени параметри. За провеждането на експеримента е използвано подмножество от 14 атрибута, избрани по препоръка на източника на данните.

Преди употребяване на данните за тренировъчни цели, те са подложени на предварителна обработка. Като правилен резултат за всеки пациент е взет атрибутът за сериозност на заболяването, вариращ от 0 (няма заболяване) до 4 (много тежко заболяване).

Таблица 1. Дистрибуция на пациентите според заболяване

Данни	Здрави	Леко заболяване	Средно заболяване	Тежко заболяване	Много тежко заболяване
Всички	411	265	109	107	28
Изчистени	160	56	35	35	13

2.2. Архитектура на модела

За архитектура на модела е избрана дълбока невронна мрежа. Дълбока невронна мрежа представлява невронна мрежа с повече от един междинен слой неврони и/или операции между входните параметри и изходните резултати.

Входните параметри на модела са реструктурирани, като за всеки пациент е изграден числов вектор $I = [i_1, i_2, i_3, \dots, i_{13}]$, съдържащ неговите показатели. Като резултат от работата на модела се извежда вероятността пациентът да има сърдечносъдово заболяване или да е здрав (вектор $R = [r_1, r_2]$). Функцията на модела може да бъде представена като: $R = F(I)$

Невронната мрежа е изградена от няколко последователни слоя, групирани в блокове. Всеки блок се състои от напълно свързан слой от неврони, върху който е приложена нормализация на междинните параметри. По време на обучение на невронната мрежа се прилага алгоритъм Dropout [2], чрез който стохастично отпадат неврони с цел увеличаване на точността в реални условия. Математическата операция на напълно свързан слой неврони може да се изрази чрез функцията:

$$f(x) = w^T \cdot x + b \quad (1)$$

където w е тегловната матрица, а b е отклонението.

С цел придържане на междинните параметри близо до стойността 0, предотвратяващи получаването на много голям градиент по време на тренировка, е поставен слой за нормализация. Използвана е техника, наречена BatchNormalization [3], която изчислява средното аритметично и стандартното отклонение по време на тренировка.

Последователната структура слоеве (напълно свързан слой, нормализация, отпадане) се повтаря четири пъти. При всяка итерация дълбочината на напълно свързания слой неврони намалява. Дълбочината на всеки от четирите нива е 254, 128, 64, 32.

Финалният слой на модела се състои от класификационната глава. Тя представлява напълно свързан слой с изходен вектор R_{logits} , представляващ необработените крайни стойности на резултатите. За финална операция е приложена SoftMax [4] функция. Чрез нея изходните стойности от предишния слой се мащабират до диапазон [0.0, 1.0] за получаването на изходен вектор на вероятностите R .

2.3. Метрики

По време на обучение използваното количество данни, предназначени за експеримента, е разделено на тренировъчно, валидационно и тестово подмножество в пропорции 7 към 2 към 1. За оценка на модела се прилага метод на валидация чрез матрица на объркване:

$$ConfusionMatrix = \begin{bmatrix} TruePositive & FalsePositive \\ FalseNegative & TrueNegative \end{bmatrix} \quad (2)$$

Въз основа на матрицата на объркване се изчисляват основните метрики като обща точност (Accuracy), прецизност (Precision), пълнота (Recall). Метриката за обща точност представлява отношението на правилно класифицирани примери спрямо всички примери от тестовото множество. Тя се изчислява по следната формула:

$$Accuracy = \frac{TruePositive+TrueNegative}{TruePositive+FalsePositive+TrueNegative+FalseNegative} \quad (3)$$

Метриката на прецизността измерва качеството на модела изходните резултати при класификация. В изследването чрез тази метрика е изчислена вероятността един пациент да има сърдечно съдово заболяване, ако моделът изведе наличие на такова заболяване. Дадената метрика се изразява чрез формулата:

$$Precision = \frac{TruePositive}{TruePositive+FalsePositive} \quad (4)$$

Метриката на пълнотата допълва прецизността, като показва процента на разпознати позитивни класификации спрямо всички позитивни класификации. Тази метрика е особено важна при настоящия експеримент поради това, че се работи с медицински данни и разпознаване на болести. При разглеждане на типовете грешни класификации, въпреки че и двата типа се зачитат за грешка, грешна класификация на болен като здрав е много по-критична от грешна класификация на здрав като болен. Метриката на пълнотата се изчислява чрез следната формула:

$$Recall = \frac{TruePositive}{TruePositive+FalseNegative} \quad (5)$$

3. Резултати

След обучаване на модела върху тренировъчното множество, успеваемостта на модела е изчислена върху тестовото множество за получаване на базова успеваемост върху генерализирани записи, които не са ползвани в тренировка. Тестовото множество се състои от 30 отделени от общо 299-те изчистени записи. Получената матрица на обръкване спрямо тях е следната:

Таблица 2. Матрица на обръкването върху тестовото множество		
	Позитивен (Болен)	Негативен (Здрав)
Предсказан позитив (Предсказан болен)	15	3
Предсказан негатив (Предсказан здрав)	0	12

От резултативната матрица на обръкване се извеждат метриците:

- обща точност $\approx 90\%$
- прецизност $\approx 83\%$
- пълнота $\approx 100\%$

В дадения експеримент моделът показва висока точност върху тестовото подмножество, като вниманието на модела е фокусирано върху правилната диагностика на позитивни случаи.

Заклучение

Въз основа на проведен експеримент за разпознаване на сърдечносъдови заболявания при дадени параметри от първични изследвания на пациент с помощта на невронна мрежа се извежда информация дали той има потенциално сърдечносъдово заболяване. За проведения експеримент са използвани публични данни, които са изчистени от непълни записи. Избраният модел представлява дълбока невронна мрежа от пет нива неврони и изходни вероятности за текущото състояние на пациента. Успеваемостта на модела е измерена чрез

метриките обща успеваемост, прецизност и пълнота. Експерименталният модел е трениран с приоритет да маркира пациент с потенциално сърдечно заболяване и при най-малкото съмнение поради потенциалните усложнения за пациента при неправилна диагностика. Проведеният експеримент показва, че невронните мрежи постигат добра успеваемост ($\approx 90\%$) в задачата за откриване на сърдечносъдови заболявания при тренировка върху сравнително малко множество данни (299 записа).

Литература

- [1]. Janosi A., W. Steinbrunn, M. Pfisterer & R. Detrano, (1988), Heart Disease, UCI Machine Learning Repository.
- [2]. Baldi P., P. J. Sadowski, (2013), Understanding Dropout, Advances in Neural Information Processing Systems, Curran Associates Inc., volume 26
- [3]. Bjorck N., C. P. Gomes, B. Selman & K. Q. Weinberger, (2018), Understanding Batch Normalization, Advances in Neural Information Processing Systems, Curran Associates Inc., volume 31
- [4]. Goodfellow I., Y. Bengio & A. Courville, (2016). "6.2.2.3 Softmax Units for Multinoulli Output Distributions", Deep Learning, MIT Press, pp. 180–184

За контакти:

инж. Димитър В. Кръстев
катедра „Софтуерни и интернет технологии”
Технически университет - Варна
E-mail: dimi1010100@gmail.com



**СОФТУЕРНИ И ИНТЕРНЕТ
ТЕХНОЛОГИИ**

ГАУСОВ НАИВЕН БЕЙСОВ КЛАСИФИКАТОР

Нели Ан. Арабаджиева – Калчева, Гинка К. Маринова, Мая П. Тодорова

Резюме: Целта на статията е представяне на популярния алгоритъм Гаусов Наивен Бейсов класификатор с неговите особености, предимства и недостатъци. Разгледаният алгоритъм се базира на теоремата на Бейс и Гаусовото разпределение. Изследванията показват, че Гаусовият Наивен Бейсов класификатор е относително бърз при класификация на нови екземпляри, но точността му е ниска при голям брой класове.

Ключови думи: Наивен Бейсов класификатор, Гаусов Наивен Бейсов класификатор, машинно обучение, предимства и недостатъци на алгоритъма Гаусов Наивен Бейсов класификатор

GAUSSIAN NAIVE BAYES CLASSIFIER

Neli An. Arabadzieva – Kalcheva, Ginka K. Marinova, Maya P. Todorova

Abstract: The article presents the popular Gaussian Naive Bayes classifier algorithm with its characteristics, advantages and disadvantages. The considered algorithm is based on Bayes theorem and Gaussian distribution. Research shows that the Gaussian Naïve Bayes classifier is relatively fast in classifying new instances, but its accuracy is low for a large number of classes.

Keywords: Naive Bayes classifier, Gaussian Naive Bayes classifier, machine learning, advantages and disadvantages of the algorithm Gaussian Naive Bayes classifier

1. Увод

Един от класическите алгоритми в машинното обучение е Наивният Бейсов класификатор [1, 2, 3, 4, 5, 6], който се базира на теоремата на Бейс за определяне на апостериорната вероятност за настъпване на дадено събитие.

Теорема на Бейс [7]:

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

където: $P(y=c|x)$ е апостериорна вероятност на класа

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

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

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

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

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

Известни алгоритми от типа Наивен Бейсов класификатор са: Гаусов, Бернулиев и Мултиномиален, свързани с различни предположения за разпределението на признаците.

2. Алгоритъм Гаусов Наивен Бейсов класификатор

Гаусовият Наивен Бейсов класификатор [8, 9, 10, 11, 12, 13, 14, 15, 16, 17] е свързан с предположение за нормално разпределение на признаците. Наивният аспект на алгоритъма е, че той третира всички входни величини като независими едни от други.

Функцията на вероятностната плътност на нормално разпределена случайна величина е:

$$P(x|\mu, \sigma) = \text{normpdf}(z) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}, \quad (3)$$

където

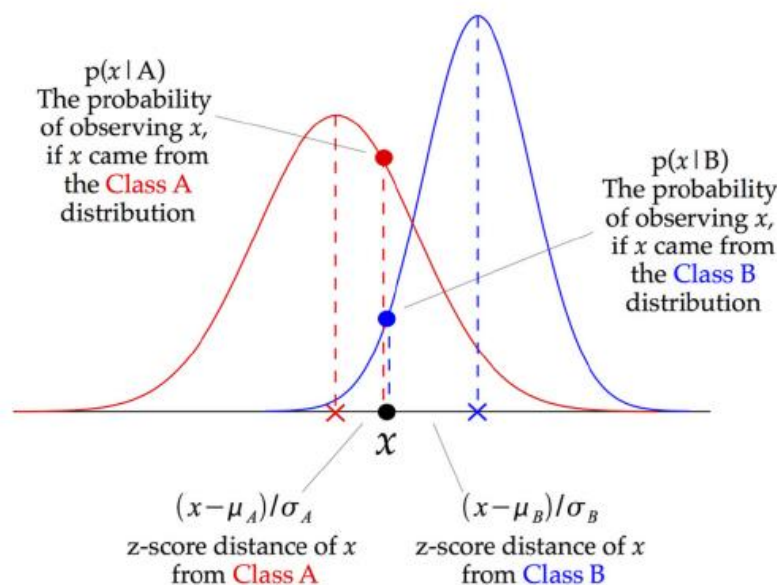
μ - математическо очакване:

$$\mu = \frac{1}{N} \sum_{i=1}^N (x_i - \mu)^2 \quad (4)$$

σ - стандартното отклонение:

$$\sigma^2 = \frac{1}{N-1} \sum_{i=1}^N (x_i - \mu)^2 \quad (5)$$

Схематично фигура 1 [15] илюстрира функционирането на Гаусовия Наивен Бейсов класификатор. При дадени два класа А и В, за всеки поотделно разглеждан признак, разстоянието z на дадена точка от центъра се изчислява както следва: $z_A = (x - \mu_A) / \sigma_A$. Разстоянието z_B се намира по подобен начин. Всеки z резултат се трансформира във вероятностна стойност, съгласно уравнението за Гаусово нормално разпределение (3).



Фиг. 1. Функциониране на Гаусовия Наивен Бейсов класификатор [15]

Разстоянията z , които се изчисляват, илюстрирани на фигура 1, са разстояния от центровете на класовете. Центърът, който е най-близък до конкретната точка от класа, ще бъде класът, към който ще бъде присвоена тази точка. Гаусовият Наивен Бейсов класификатор не изчислява обикновените евклидови разстояния до класовите центрове, а вместо това нормализира разстоянието по всяко измерение чрез вариацията по същото измерение. Присвояването на всяка точка от данни към класа, чийто център е най-близък, ще

доведе до границата на решение, която се намира на средата между двата класа и която е перпендикулярна на линията, свързваща тези центрове [15].

Уравнение (3) се изчислява за всеки признак. Тъй като Гаусовият Наивен Бейсов класификатор не отчита статистическа зависимост на признаците, затова съвместната вероятност на всички признаци е произведение от индивидуалните вероятности на всеки признак.

$$P(x_1, x_2, \dots, x_n | A) = \prod_{i=1}^n P(x_i | A) \quad (6)$$

Умножаването на голям брой малки p -стойности води до грешки при закръглянето, затова вместо произведение се използва сума на логаритмите на p -стойностите.

$$\log P(x_1, x_2, \dots, x_n | A) = \sum_{i=1}^n \log P(x_i | A) \quad (7)$$

Принадлежността на класа се определя в зависимост коя сума е по-голяма:

$\sum_{i=1}^n \log P(x_i | A)$ или $\sum_{i=1}^n \log P(x_i | B)$

Предимства:

- относително бърз при класификация на нови екземпляри [14, 15, 16];
- необходима е малка извадка, за да се обучи класификаторът [10];
- ефективен класификатор, който може да се използва в сложни реални ситуации [10, 14].

Недостатъци:

- не е добър оценител [14, 15];
- ниска скорост и точност на класификация при голям брой класове [13];
- допускането за независимост на признаците едни от други [8, 11, 15];
- класификаторът моделира всяко измерение поотделно [15].

Заклучение

Гаусовият Наивен Бейсов класификатор е базиран на теоремата на Бейс и Гаусовото нормално разпределение. Алгоритъмът е ефективен при класификация на сложни реални системи и относително бърз при класификация на нови екземпляри. Като всеки Бейсов класификатор допуска независимост на признаците едни от други, което се явява недостатък.

Изследванията показват, че Гаусовият Наивен Бейсов класификатор не е цялостен универсален метод за класифициране, който да изчислява с точност 100% всяка приложна задача.

Литература

- [1]. Калчева Н., Матева З., Бейсова теория в машинното обучение, Годишник на ТУ Варна 2016, стр. 130-133
- [2]. Калчева-Арабаджиева Н., Николов Н. Сравнителен анализ между наивния бейсов класификатор и метода на опорните вектори използващ оптимизация при класификация на български текст в машинното обучение, Списание Компютърни науки и технологии, pp. 97-105, 2017, ТУ – Варна, България.
- [3]. Калчева-Арабаджиева Н., Н. Николов, Класифициране на мнения от потребителски коментари на английски език чрез алгоритми на машинното обучение, V-та научна

- конференция с международно участие "Компютърни науки и технологии", 28-29 септември 2018 г., Списание Компютърни науки и технологии, Варна, България, том 1, стр. 120-128
- [4]. Arabadzieva – Kalcheva, Neli An., Comparative Analysis of Algorithms for Classification of Text in the Bulgarian Language in Machine Learning, International Conference “Applied Computer Technologies” ACT 2018 – Ohrid, Macedonia, pp. 6 - 11
 - [5]. Kalcheva N., M. Karova, I. Penev, Comparison of the accuracy and the execution time of classification algorithms for Bulgarian literary works, International Conference “Automatics and Informatics’2020”, Varna, Bulgaria.
 - [6]. Kalcheva N., M. Todorova, G. Marinova, Naive Bayes Classifier, Decision Tree and AdaBoost Ensemble Algorithm. Advantages and disadvantages, Naive Bayes Classifier, Decision Tree and AdaBoost Ensemble Algorithm. Advantages and disadvantages, 2020
 - [7]. Уиткин Л. В. Машинное обучение (Machine Learning) Метрические методы классификации и регрессии, 2017, стр. 6-8.
 - [8]. Bhuvaneswari, R., K. Kalaiselvi. Naive Bayesian classification approach in healthcare applications. International Journal of Computer Science and Telecommunications, vol. 3, Issue 1, 2012, pp. 106-112.
 - [9]. Bi, Z., Y. Han, C. Huang, M. Wang. Gaussian Naive Bayesian data classification model based on clustering algorithm. International Conference on Modeling, Analysis, Simulation Technologies and Applications (MASTA 2019), Advances in Intelligent Systems Research, vol. 168, 2019, pp. 396-400.
 - [10]. Kruthika, K., Rajeswari, H. Maheshappa. Multistage classifier-based approach for Alzheimer's disease prediction and retrieval. Informatics in Medicine Unlocked 14, 2019, pp. 34-42.
 - [11]. Losing, V., B. Hammer, H. Wersing. Choosing the best algorithm for an incremental on-line learning task. In European Symposium on Artificial Neural Networks (ESANN), Brügge, 2016.
 - [12]. Nikolentzos, G., P. Meladianos, F. Rousseau, Y. Stavarakas, M. Vazirgiannis. Multivariate Gaussian document representation from word embeddings for text categorization. In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: vol. 2, 2017, pp. 450-455.
 - [13]. Pandarakone, S., Y. Mizuno, H. Nakamura. A comparative study between machine learning algorithm and artificial intelligence neural network in detecting minor bearing fault of induction motors. Energies 2019, 12, pp. 2105-2119.
 - [14]. Raghunath, D., C. Usha, K. Veera, V. Manoj. Predicting heart disease using machine learning techniques. International Research Journal of Computer Science VI, no.4, 2019, pp. 149-153.
 - [15]. Raizada, R., Y. Lee. Smoothness without smoothing: why Gaussian Naive Bayes is not naive for multi-subject searchlight studies, PloS one 8, no. 7, 2013.
 - [16]. Raschka, S. Naive Bayes and text classification I - Introduction and theory. Cornell University library, Ithaca, 2014.
 - [17]. Shelke, S., B. Aksanli. Static and dynamic activity detection with ambient sensors in smart spaces. Sensors 19, no. 4, 2019, pp. 804-823.

За контакти:

доц. д-р инж. Нели Ан. Арабаджиева - Калчева
катедра „Софтуерни и интернет технологии”
Технически университет - Варна
E-mail: n_kalcheva@tu-varna.bg

ас. д-р инж. Гинка К. Маринова
катедра „Компютърни науки и технологии”
Технически университет - Варна
E-mail: gmarinova@tu-varna.bg

гл. ас. д-р Мая П. Тодорова
катедра „Софтуерни и интернет технологии”
Технически университет - Варна
E-mail: maya.todorova@tu-varna.bg



**СОФТУЕРНИ И ИНТЕРНЕТ
ТЕХНОЛОГИИ**

ЗАДАЧА ЗА КЛАСИФИКАЦИЯ НА ТЕКСТ В МАШИННОТО ОБУЧЕНИЕ

Нели Ан. Арабаджиева – Калчева, Мая П. Тодорова, Гинка К. Маринова

Резюме: В статията са описани основните етапи при решаване на задачата за класификация на текст. Първият етап на предварителна обработка на текста се състои в представяне на входния текстов поток в поредица от текстови единици, наречени токъни (tokens). Следващата стъпка е процес на построяване на числов модел на текста, удобен за по-нататъшна обработка на данните. Следва подбор на признаци, използвани при класификация, който може да бъде евристичен, основан на лингвистични знания или знания на предметни области и/или статистически. При реализацията на задачата за класификация на текст се генерира модел въз основа на входните данни и резултата, с който впоследствие се класифицира нов обект. За оценка на качеството на конструирания модели за класификация на изследваните множества от данни се прилагат различни метрики.

Ключови думи: класификация, класификация на текст, машинно обучение, точност, класификационна грешка, прецизност, чувствителност, мярка F-measure, матрица на грешките, мярка Kappa Statistic, ROC Area

TEXT CLASSIFICATION TASK IN MACHINE LEARNING

Neli An. Arabadzieva – Kalcheva, Maya P. Todorova, Ginka K. Marinova

Abstract: The article describes the main stages in solving the task of text classification. The first stage of text preprocessing consists in representing the input text stream in a series of text units called tokens. The next stage is the process of building a numerical model of the text, convenient for further data processing. The following stage is a selection of features used in classification, which may be heuristic, based on linguistic or domain knowledge, and/or statistical. In the implementation of the text classification task, a model is generated based on the input data and the output, with which a new object is subsequently classified. Different metrics are applied to evaluate the quality of the constructed classification models of the studied data sets.

Keywords: classification, text classification, machine learning, accuracy, error rate, precision, recall, F-measure, confusion matrix, Kappa Statistic, ROC Area

1. Увод

С навлизането на интернет във всеки дом информацията за мнението на потребителите при ревюта, онлайн реклами, отзиви за услуги, стоки [3], прогнозиране на цени [4, 10] става все по-важна за бизнеса. Класификацията на текст успешно се прилага при идентификация на тема, жанр [1], автори, авторски характеристики [2,3], филтриране на документи, откриване на спам [4, 5, 6, 7, 8, 9]. Нараства броят на специалисти в областта на машинното обучение, които се занимават с анализ на потребителската кошница [8], разпределение на реклами, персонални новости, съставяне на интернет каталози [6], прогнозиране на извънредни ситуации след земетресения [11]. Полезно приложение за всеки ползвател на дигитално средство е разпознаването на ръкописни букви и цифри [4, 10]. Класификацията на текст е намерила приложение и в сферата на финансите [7]: кредитен скоринг, трейдинг, класификация на облигации [10], кредитен рейтинг [12]. Реализирани са приложения за разкриване на плагиатство, двусмислие на думи, влияния на исторически писатели, намиране на препоръки, тематични текстове, проверка на документи [3]. Лингвисти по цял свят се

занимават с изследване, анализ и приложение на методите за класификация на текст при обработка на естествен език [10]. Всеки съвременен потребител на смартфон, таблет и на компютър волно и/или неволно е ползвател на приложения за класификация на текст: при търсене на информация [2, 3], машинен превод и проверка на правопис [5, 6], системи за документооборот [5].

2. Постановка на задачата за класификация

Формална постановка на задачата за класификация:

Нека:

- $X \in R^n$ – множество на обектите (входни данни)
- $Y \in R$ – множество на резултатите (изходни данни)

Законът на разпределение $P_{XY}(x,y)$ не е известен, известна е само обучаваща извадка. Двойката (x,y) се представя като реализация на $(n+1)$ – мерни случайни величини (X,Y) , зададени във вероятностното пространство.

$$\{x_1, y_1, x_2, y_2, \dots, x_N, y_N\} \quad (1)$$

където (x_i, y_i) , $i = 1, 2, \dots, N$ се явяват независими реализации на случайна величина (X, Y) .

Целта е да се намери функция $\Phi: X \rightarrow Y$, която изхождайки от стойностите на x , да предскаже y . Функцията Φ се нарича решаваща функция или класификатор.

3. Основни етапи при решаване на задачата за класификация на текст

Решаването на задачата за класификация на текст се състои от четири последователни етапа:

- Предварителна обработка на текст и индексация на документите.
- Редуциране размерността на пространството от признаци.
- Построяване и обучение на класификатор с помощта на методите на машинното обучение.
- Оценка на качеството на класификация.

А) Предварителната обработка на текст и индексация на документи

Предварителната обработка на текст включва в себе си токънизация (tokenization), която може да се представи като процес на сегментация, обособяване на отделни текстови единици, наречени токъни (tokens) [13, 14]. Този процес премахва препинателни знаци, интервали, нов ред и други символи за форматиране.

Индексацията на документите е процес на построяване на числов модел на текста, удобен за по-нататъшна обработка на данните. Често използван модел за представяне на текст е моделът речник или още наричан „торба от думи“ (bag of words (BoW)), който позволява представянето на документ в многомерен вектор от думи и тегла [2, 8, 14, 15]. С други думи, всеки документ се представя с вектор в многомерното пространство, чиито координати съответстват на номерата на думите, а значенията на координатите – на техните тегла [16]. При този модел не се отразява позицията на думите и по този начин се губи информация за семантиката и структурата на текста.

Разпространен модел на индексация е Word2Vec [16], който представя всяка дума във вид на вектор, който съдържа информация за контекста на думите.

Друг модел наречен n -грам (n -grams) [14, 16], съдържа n на брой последователни думи в текста. Ако $n=1$, n -грамите се наричат униграми (unigrams), т.е. отделни думи, а при две думи биграми (bigrams) [17].

Б) Редуциране размерността на пространството от признаци

Първа стъпка в процеса на намаляване размера на пространството е изключването на т.нар. „стоп думи“ (stopwords) [14, 15, 16], които съдържат съюзи, предлози, местоимения, наречия и други незначещи думи. Това не съкращава особено размера, но увеличава качеството на останалите думи. Премахването на близките по значение и излишните думи подобряват точността и бързодействието на алгоритмите. Например, различните морфологични форми следва да се считат за еквивалентни. Прилагането на морфологичен анализ позволява значително да се намали размерността на пространството [7]. Процес без омонимия на лексикални и граматически елементи се нарича tagging. Софтуерните средства, реализиращи този процес, са известни като POS тагери (part-of-speech taggers) [18]. В съответствие със зададен морфологичен модел всяка дума се аотира с респективната ѝ категория за част на речта. В резултат в качеството на признаци на документа се вписват значими думи, срещани в документа [16]. За българския език има разработени три тагера, които са експериментирани в системата BulTreeBank (<http://www.bultreebank.org/taggers/taggers.htm>).

Изчислителната сложност на различните методи за класификация зависи от размерността на пространството на признаците. Затова за по-ефективна работа на класификаторите често се прибегва до избор и редуциране на броя на признаците [19].

Някои класификатори игнорират често срещани, а други - рядко използвани термини [20]. Традиционно в качеството на признаци се използват честотата на срещане на думите, на които се съпоставят тегла [21].

Съществуват няколко способа за определяне на теглото на признаците на документа. Разпространени са изчислителните функции TF.IDF [2, 9, 14, 15, 16]. Числото TF.IDF - Term Frequency – Inverse Document Frequency, определя важността на определена дума в даден документ от даден корпус от документи. Колкото е по-голяма честотата на термина в текста, толкова е по-вероятно думата да е ценна за съдържанието на документа. Основната идея се състои в това, че голямо тегло получават думи с висока честота на срещане в конкретния документ и малко тегло в други документи.

Обособяването, общата значеща част в групата думи със сходен буквен състав, които имат смислово близки понятия, води до конструирането на специални програми за автоматична идентификация на общата част, която се нарича stem, а съответните програми – stemmers [15, 22]. Това е един от подходите за намаляване на размера на пространството от термини. Stemmer за български език е създаден и докладван през 2003 г. от Преслав Наков [23] и именуван BulStem.

В) Построяване и обучение на класификатор с помощта на методите на машинното обучение

Машинното обучение предполага наличието на обучаваща и контролна (тестова) извадка, т.е. $\Omega = \{d_1, d_2, \dots, d_{|\Omega|}\} \subset D$ е начална колекция от документи, където целевата функция Φ е известна за $\forall d_i$. Класификаторът Φ обучава индуктивно на основата на изявените характеристики на документите [6, 20, 21].

Документите на Ω се разделят на две непресичащи се множества:

- Обучаващо множество $T_r = \{d_1, d_2, \dots, d_{|T_r|}\}$ е множество документи, с помощта на които се създава класификатор Φ' . Φ' обучава индуктивно, основано на избраните характеристики на тези документи.

- Тестово множество $T_e = \{d_{|T_r|+1}, \dots, d_{|\Omega|}\}$ е множество документи, на които се тества ефективността на построенния класификатор. Всеки тестов документ подава на входа на класификатора Φ' и след това сравнява крайния резултат Φ' с известните значения на функцията Φ .

Г) Оценка на качеството на класификация

Класификационният модел разпределя съответния обект в един от възможните класове: вярно класифициран (True) или невярно класифициран (False). При този модел са възможни четири варианта на класификация [24]:

- True Positive (TP) (2) – действителен е клас Positive и вярно класифициран като клас Positive [16, 19];

$$TP\ Rate = \frac{TP}{TP+FN} \quad (2)$$

- True Negative (TN) (3) – действителен е клас Negative и вярно класифициран като клас Negative [16, 19];

$$TN\ Rate = \frac{TN}{TN+FP} \quad (3)$$

- False Positive (FP) (4) – действителен е клас Negative, а невярно класифициран като клас Positive;

$$FP\ Rate = \frac{FP}{FP+TN} \quad (4)$$

- False Negative (FN) (5) – действителен е клас Positive, а невярно класифициран като клас Negative.

$$FN\ Rate = \frac{FN}{FN+TP} \quad (5)$$

Матрица на грешките (Confusion matrix, contingency table) (табл.1) е квадратна матрица, чиито редове и колони отговарят на класовете. Редовете на матрицата представляват действителните екземпляри от съответния клас, а колоните - класовете, получени след прилагането на класификационния модел върху тестовите данни [25].

Таблица 1. Матрица на грешките

Клас	Positive	Negative
Positive	True Positive TP	False Negative FN
Negative	False Positive FP	True Negative TN

По главния диагонал на матрицата се намират правилно класифицираните обекти: True Positive и True Negative, а в останалите клетки на матрицата - грешно класифицираните екземпляри: False Negative и False Positive.

При крос валидация се изчисляват основно: точност (Accuracy) [16, 22, 24, 25, 26], класификационна грешка (Error Rate) [26], прецизност (Precision) [2, 8, 16, 22, 25], чувствителност (пълнота) (Recall) [2, 8, 16, 22, 25, 26] и комбинираната мярка F-measure [2, 8, 16].

Мярката точност (Accuracy) (6) се изчислява като отношението на правилно класифицирани примери към общия брой обекти от тестово множеството. Обикновено се представя в проценти.

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \quad (6)$$

Класификационната грешка (Error Rate) (7) се изчислява като отношението на неправилно класифицирани примери към общия брой обекти от тестовото множество. Обикновено се представя в проценти.

$$Error\ Rate = \frac{FP+FN}{TP+FP+TN+FN} \quad (7)$$

Прецизността (Precision) (8) е мярка, която показва доколко е точен класификаторът и дава процента на правилно класифицираните примери от съответния клас.

$$Precision = \frac{TP}{TP+FP} \quad (8)$$

Мярката чувствителност (Recall) (9) показва пълнотата или възвращаемостта на класификатора, т.е. колко от всички примери от даден клас е успял да определи като принадлежащи на този клас. Обикновено се представя в проценти.

$$Recall = \frac{TP}{TP+FN} \quad (9)$$

Мярката F-measure (10), която комбинира Precision и Recall, е тяхното претеглено хармонично средно. Обикновено се представя в проценти.

$$F\ measure = \frac{2}{1/Recall + 1/Precision} \quad (10)$$

Мярката Kappa Statistic [28] измерва степента на съгласуваност между предсказания и действителния клас. Максималната стойност на Kappa Statistic е 1, а минималната е 0, т.е. е в интервала [0;1]. При пълно съгласуване между предсказания и действителния клас стойността на Kappa Statistic е 1.

Receiver Operating Characteristic (ROC) кривите [8, 9, 12] се използват за визуална оценка на точността на класификаторите, както и за сравняване на различни класификационни модели. ROC кривата представлява двумерна диаграма, която се получава чрез изобразяване на True Positive Rate (TP Rate) по вертикалната ос и False Positive Rate (FP Rate) по хоризонталната ос.

Като количествена оценка на работата на класификационните модели се използва площта под ROC кривата - ROC Area [12]. Максималната стойност е 1, а минималната е 0. Класификационният модел е по-добър, колкото по-голяма е площта над ROC кривата. Ако ROC Area < 0.5, то оценяваният модел работи по-лошо от случайния класификатор, който е с ROC Area = 0.5.

Под бързодействие на класификатора се разбира времето, необходимо за отнасяне на документ към един от класовете. В задачите за класификация на текст бързодействието се измерва като процесорно време (в секунди) или като количество изчислителни операции, необходими за класификация. Увеличаването на точността води до снижаване на бързодействието, а съответно увеличаването на бързодействието понижава точността на работа на класификатора [16].

Мерките за оценка на качеството на класификация са инварианти по отношение на използваните класификационни алгоритми и методи.

Заклучение

Класификацията на текст е важна приложна и изследователска задача в областта на машинното обучение. В процеса на класификация на текст данните се разделят на обучаващо множество, което се използва за обучение на модела, и тестово множество, използвано за тестване на модела. Процесът на класификация има за цел да се построи модел, който еднозначно съпоставя стойностите на признаците с класа, към който принадлежи обектът.

Конструираниите модели за класификация на данни могат да се оценяват на базата на получени резултати чрез различни критерии и метрики за оценка на качеството на класификация.

Литература

- [1]. Shalev-Shwartz, S., S. Ben-David. Understanding machine learning: From theory to algorithms. Cambridge University press, 2014.
- [2]. Jurafsky, D., J. Martin. Naive Bayes and sentiment classification. Speech and Language Processing 7, 2016.
- [3]. Zechner, N. A novel approach to text classification. PhD diss., Umeå universitet, 2017.
- [4]. Воронов, А. Обзор алгоритмов машинного обучения, Курс лекций К.В. Воронцова по машинному обучений (2007-2008),
https://www.compression.ru/video/seminar/slides/2009_MachineLearning.pdf.
- [5]. Ермакова, Л. Методы классификации текстов и определения качества контента. Вестник Пермского университета 3, 2011.
- [6]. Лифшиц, Ю. Классификация текстов. 2005, URL:<http://yury.name/internet>.
- [7]. Khan, A., B. Baharudin, L. Lee, K. Khan. A review of machine learning algorithms for text-documents classification. Journal of Advances Information Technology, vol. 1, 2010, pp. 4-20.
- [8]. Murphy, K. Machine learning a probabilistic perspective. The MIT Press Cambridgr, 2012.
- [9]. Song, Y., A. Kolcz, C. Giles. Better Naive Bayes classification for high-precision spam detection. Software – Practice and Experience 39, no. 11, 2009, pp.1003–1024.
- [10]. [10]. Мелёшин, И. Анализ данных и машинное обучение в MATLAB. Машинное обучение в MATLAB, The MathWorks, 2014,
- [11]. https://matlab.ru/upload/iblock/61e/2%20Machine_Learning_with_MATLABjMeLt.pdf.
- [12]. Simpson, E., S. Reece, S. Roberts. Bayesian heatmaps: probabilistic classification with multiple unreliable information sources. In Joint European Conference on Machine Learning and Knowledge Discovery in Databases. Springer, Cham, 2017, pp. 109-125.
- [13]. Brown, I., C. Mues. An experimental comparison of classification algorithms for imbalanced credit scoring data sets. Expert Systems with Applications 39, no. 3, 2012, pp. 3446-3453.
- [14]. Kowsari, K., K. Meimandi, M. Heidarysafa, S. Mendu, L. Barnes, D. Brown. Text classification algorithms: A survey. information 10, no. 4, 2019.
- [15]. Raschka, S. Naive Bayes and text classification I - Introduction and theory. Cornell University library, Ithaca, 2014.
- [16]. Кузнецов, Р. Классификатор веб-страниц на базе SVM-Multiclass. Труды РОМИП'2006, Санкт-Петербург: НУ ЦСИ, 2006, с. 101-112.
- [17]. Батура, Т. Методы автоматической классификации текстов. Программные продукты и системы 30, no. 1, 2017.
- [18]. Taheri, S., M. Mammadov, A. Bagirov. Improving Naive Bayes classifier using conditional probabilities. In Proceedings of the Ninth Australasian, 2010.
- [19]. Найденова, К., О. Невзорова. Машинное обучение в задачах обработки естественного языка:Обзор современного состояния исследований. Ученые записки

- Казанского университета. Серия Физико-математические науки, том 150, кн. 4, 2008, с. 4-24.
- [20]. Sobran, N., A. Ahmad, Z. Ibrahim. Classification of Imabalanced dataset using conventional Naive Bayes classifier. International Conference on Artificial Intelligence in Computer Science and ICT (AICS 2013), 2013.
- [21]. Ермакова, Л. Методы классификации текстов и определения качества контента. Вестник Пермского университета 3, 2011.
- [22]. Ермакова, Л., М. Абашев, Р. Никитин, Р. Ушаков. Методы автоматической классификации текстов по функциональным стилям. Вестник Пермского университета, Математика, Механика, Информатика 4, 2014.
- [23]. Зайцев, В., Л. Линь. Способы повышения эффективности классификации документов для конечного множества языков. Вісник НТУУ «КПІ» Інформатика, управління та обчислювальна техніка №50, 2009, с. 100-104.
- [24]. Nakov, P. BulStem: Design and evaluation of inflectional stemmer for Bulgarian. In Proceeding of the Workshop on Balkan Language Resources and Tools, Thessaloniki, 2003.
- [25]. Спасова, В. Методи и алгоритми за обработка на данни в асистиращи системи за възрастни и хора с увреждания. Автореферат на дисертация за присъждане на образователната и научна степен “доктор”, ТУ – София, 2016, с. 33
- [26]. Sakhi, A., B. Mallick. High performance intrusion detection using modified k-mean & Naïve Bayes. International Journal of Research in Engineering and Technology 5, no. 8, 2016.
- [27]. McCallum, A., K. Nigam. A comparison of event models for Naive Bayes text classification. Papers from the 1998 AAAI Workshop, 1998, pp. 41-48.
- [28]. Kruthika, K., Rajeswari, H. Maheshappa. Multistage classifier-based approach for Alzheimer's disease prediction and retrieval. Informatics in Medicine Unlocked 14, 2019, pp. 34-42.
- [29]. Rossiter, D. Statistical methods for accuracy assessment of classified thematic maps. International Institute for Geo-information Science and Earth, Department of Earth Systems Analysis, Enschede, 2004.

За контакти:

доц. д-р инж. Нели Ан. Арабаджиева - Калчева
катедра „Софтуерни и интернет технологии”
Технически университет - Варна
E-mail: n_kalcheva@tu-varna.bg

гл. ас. д-р Мая П. Тодорова
катедра „Софтуерни и интернет технологии”
Технически университет - Варна
E-mail: maya.todorova@tu-varna.bg

ас. д-р инж. Гинка К. Маринова
катедра „Компютърни науки и технологии”
Технически университет - Варна
E-mail: gmarinova@tu-varna.bg

АНАЛИЗ ЧРЕЗ ДЪРВО НА СЪБИТИЯТА

Нели Ан. Арабаджиева – Калчева, Ивелин М. Иванов, Павлина С. Линова

Резюме: Анализът чрез дърво на събитията е метод за оценяване на риск, чрез който се представят взаимно изключващи се потенциални сценарии, следващи изходно събитие. Чрез разклоняване във вид на дървовидна схема, методът позволява да се изобразят причините за риска и последствията от него и въз основа на тях да се направи анализ на риска. В статията е предложен казус „Заразяване с компютърен вирус“ демонстриращ приложение на метода Анализ чрез дърво на събитията.

Ключови думи: Управление на риска, анализ чрез дърво на събитията, ЕТА

Event Tree Analysis

Neli An. Arabadzieva – Kalcheva, Ivelin M. Ivanov, Pavlina S. Linova

Abstract: Event tree analysis is a risk assessment method that is used to represent mutually exclusive potential scenarios following a trigger event. By branching in the form of a tree diagram, the method allows to depict the causes of the risk and its consequences and, based on them, to make a risk analysis. The article presents a case study "Infection with a computer virus" demonstrating the application of the Event Tree Analysis method.

Keywords: Risk Management, Event Tree Analysis, ETA

1. Увод

Процесът на управление на риска представлява съвкупност от взаимно влияещи си процеси на обмен на информация и консултиране, установяване на обстоятелства за идентифициране, анализиране, оценяване, въздействие, наблюдение, преглед на риска и докладване и записване на резултатите по подходящ начин [1, 2].

Оценяването на риска е свързано с цялостното управление на сигурността и може да се определи като последователност от процеси на идентификация, анализ и преценяване на риска по даден проект [1, 2]. Оценяването на риска може да се извършва с един или повече методи, в зависимост от сложността на проблема, характера и степента на неопределеност на оценяване на риска, както и наличие на експертен опит, знания и необходимите данни [3, 4].

В Стандарт 31010 е представено приложение на редица методи, с конкретни позовавания на други международни стандарти, в които концепцията и прилагането на методите са описани в по-големи потребности [1, 2, 3, 4]. Един от изложените методи е Анализ чрез дърво на събитията.

2. Метод Анализ чрез дърво на събитията

Методът за оценка на риска Анализ чрез дърво на събитията е индуктивен метод, който графично представя взаимно изключващи се последователности от събития, следващи изходното събитие, в съответствие с функционирането или нефункционирането на различни системи. Той може да се прилага както качествено, така и количествено [1].

От първите приложения на дървото на събитията през 60-те години на миналия век са направени много проучвания с помощта на дървета на събития, свързани с ядрената промишленост, химическата обработка и транспорта [5], както и в други области. Методът се използва главно в работата за анализ на надеждността, където моделирането на цялата

система с дърветата на отказите, би довело до неуправляеми модели [6]. Дървото на събитията, показва как един фактор, когато е съчетан или не е съчетан с други фактори, има вероятност да доведе до злополука [7]. Методът дава възможност за определяне на вероятностите на крайните събития и е приложим при оценка характеристиките на транспортните произшествия [8]. Започвайки от инициращо събитие, анализът на дървото на събитията позволява оценка на системното отклонение, като се вземе предвид функционирането или повредата на устройства за откриване, аларма, превенция, защита или намеса [9]. Методът е особено подходящ за анализ на аварийни системи, както и за системи свързани с безопасността, но е подходящ и за анализ на оперативни процедури, управленски решения и други системи, състоящи се от не-хардуерни компоненти [6].

Принципната връзка между различните категории на метода Анализ чрез дърво на събитията за оценяване на риска и факторите, присъстващи в дадена рискова ситуация, е дадена в таблица [1].

Таблица 1. Приложимост на метод Анализ чрез дърво на събитията, използван за оценяване на риска

Метод	Идентификация на риска	Анализ на риска			Преценяване на риска
		Последствия	Вероятност	Ниво на риска	
Анализ чрез дърво на събитията	Приложим	Препоръчителен	Приложим	Приложим	Неприложим

Видно от таблица 1, методът Анализ чрез дърво на събитията е приложим при идентификация на риска, препоръчителен за определяне на последствията от него, приложим при изчисляване на вероятността на риска и ниво на риска, но неприложим за преценяване на риска.

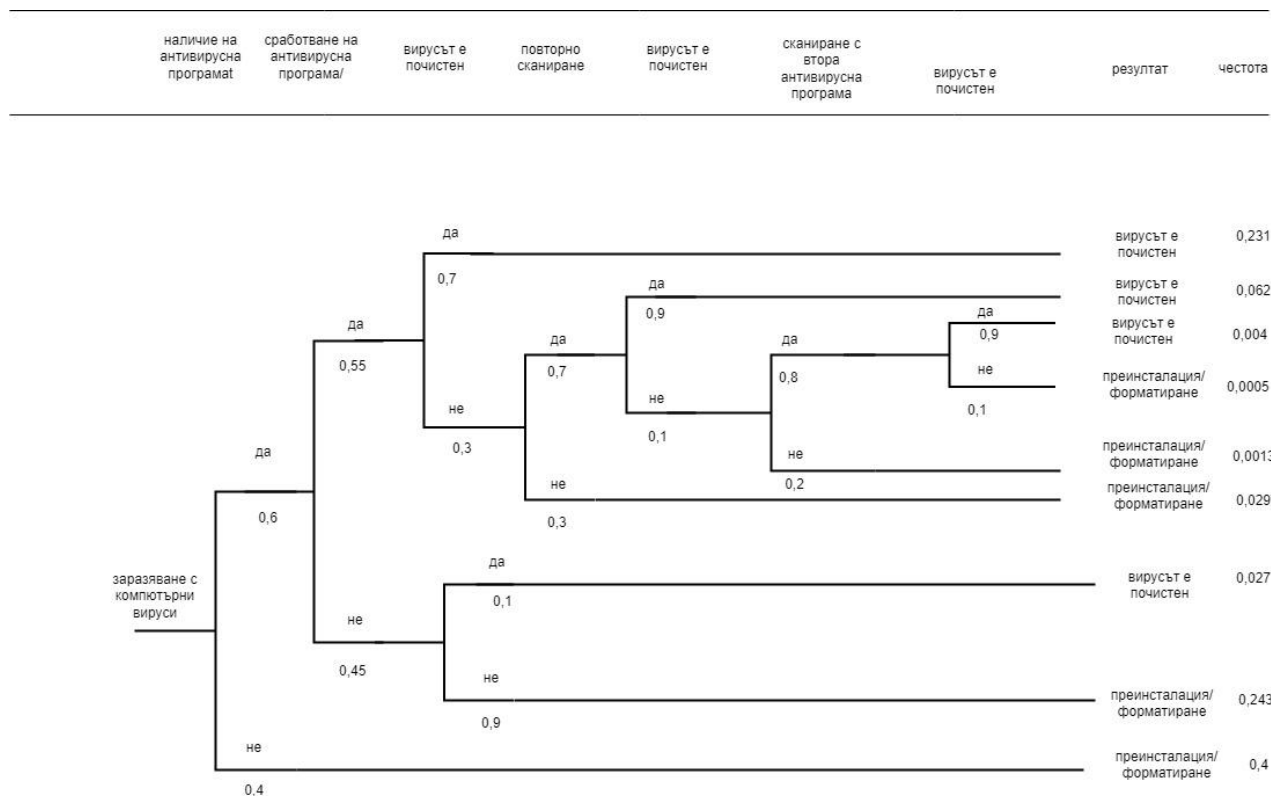
2.1. Построяване на дърво на събитията

- Построяването на дървото на събитията започва с избор на първоначално събитие, т.е. се определя начална точка на дървото.
- Вероятността за определяне на първоначалното събитие се определя чрез експертен анализ, чрез дърво на решенията или друг анализ.
- Определят се началните пътища чрез начертаване на клони на дървото, като последователно се изреждат функции или системи, които възникват в резултат на изпълнение или неизпълнение на някакво събитие.
- За всяка функция или система се начертава линия, за да представи тяхната изправност или неизправност, като пътят, който представя отказа, се спуска надолу по дървото, докато при успешно събитие се изкачва нагоре.
- За всяка функция или система може да е посочена конкретната вероятност за отказ като оценка на условната вероятност.
- Вероятностите за всеки възможен маршрут се получават като произведение на отделните условни вероятности за всяко събитие по маршрута, при условие че различните събития са независими.

2.2. Примерно практическо приложение на метода Анализ чрез дърво на събитията на конкретен казус: „Заразяване с компютърен вирус“ (фиг. 1).

- Построяването на дървото на събитията започва с определяне на първоначално събитие: „Заразяване с компютърен вирус“.

- Определят се началните пътища, като се начертават клони на дървото, като последователно се изреждат функции или системи, които възникват в резултат на изпълнение или неизпълнение на някакво събитие:
 - наличие на антивирусна програма;
 - сработване на антивирусната програма;
 - повторно сканиране;
 - сканиране с втора антивирусна програма.
- За всяка функция или система се начертава линия, за да представи тяхната изправност или неизправност, като пътят, който представя отказа, се спуска надолу по дървото, докато при успешно събитие се изкачва нагоре. В случая, когато отговорът на въпроса е „не“, клоните се спускат надолу, а при „да“ се изкачват нагоре.
- За всяка функция или система може да е посочена конкретната вероятност за отказ като оценка на условната вероятност. Например, ако вероятността за наличие на антивирусна програма е 60%, то вероятността за отсъствие на такава е 40%.
- Вероятностите за всеки възможен маршрут се получават като произведение на отделните условни вероятности за всяко събитие по маршрута, при условие че различните събития са независими. Стойностите са предложени от екип от експерти въз основа на собствен опит. Например:
 - Вероятността вирусът да е почистен е 23,71%, ако при налична антивирусна програма, тя е сработила.
 - Вероятността вирусът да е почистен е 6,2%, ако при налична антивирусна програма тя е сработила, но вирусът не се е почистил и затова се е наложило отново да се сканира.
 - Вероятността вирусът да е почистен е 0,4%, ако при налична антивирусна програма тя е сработила, но вирусът не се е почистил и затова се е наложило отново да се сканира, но вирусът не е почистен и се сканира с втора антивирусна програма.
 - Вероятността вирусът да е почистен е 2,7%, ако при налична антивирусна програма тя не е сработила.
 - Вероятността вирусът да не е почистен и се е наложило преинсталация и форматиране е 24,3%, ако при налична антивирусна програма тя не е сработила.
 - Вероятността вирусът да не е почистен и се е наложило преинсталация и форматиране е 2,97%, ако при налична антивирусна програма тя е сработила, но вирусът не се е почистил и затова се е наложило отново да се сканира.
 - Вероятността вирусът да не е почистен и да се е наложило преинсталация и форматиране е 0,13%, ако при налична антивирусна програма тя е сработила, но вирусът не се е почистил и затова се е наложило отново да се сканира, но вирусът отново не се е почистил, и се сканира с втора антивирусна програма.
 - Вероятността вирусът да не е почистен и да се е наложило преинсталация и форматиране е 40%, ако няма налична антивирусна програма и собственикът на компютъра не предприема други действия освен инсталация.



Фиг. 1. Пример за анализ чрез дърво на събитията

Заклучение

Методът Анализ на дърво на събитията представлява двоично дърво за решения, което има за цел да изследва и оценява ефектите от зададено първоначално събитие при различни условия. Методът позволява да се изобразяват влошаващи или смекчаващи събития, предизвикани в отговор на изходното събитие.

Представеният в статията пример демонстрира приложение на метода Анализ чрез дърво на събитията чрез казуса „Заразяване с компютърен вирус“. Изчислена е вероятността вирусът да бъде почистен или не в резултат от последователни действия, целящи намаляване на последствията от изходното събитие.

Литература

- [1]. БДС EN ISO 31010, Управление на риска. Методи на оценяване на риска, <https://www.bds-bg.org/>.
- [2]. Arabadzhieva – Kalcheva N., Preparation of an analysis "bow-tie" in a set case "hacker attack in it company". Scientific Conference with international participation MATTEH 2022, Conference proceedings, Vol. 2, Shumen, 2022, ISSN 1314-3921, pp. 63-67.
- [3]. Arabadzhieva – Kalcheva N., Tsankov Ts. Failure Modes and Effects Analysis – FMEA. Scientific Conference with international participation MATTEH 2022, Conference proceedings, Vol. 2, Shumen, 2022, ISSN 1314-3921, pp. 58-62.
- [4]. Arabadzhieva – Kalcheva N., Application of the method of analysis of causes and effects in a set case "computer network failure", Scientific Conference with international participation MATTEH 2022, Conference proceedings, Vol. 2, Shumen, 2022, ISSN 1314-3921, pp. 72-75
- [5]. Ramzali N., Lavasani MR., Ghodousi J., Safety barriers analysis of offshore drilling system by employing fuzzy event tree analysis, Safety science, Oct 1;78:49-59, 2015.

- [6]. Йоцов И. В., Управление на риска при осъществяване на преки инвестиции, Ejournal, vfu. bg. 2014
- [7]. Raiyan A., Das S., Islam MR., Event tree analysis of marine accidents in Bangladesh., Procedia engineering., Jan 1;194:276-83, 2017.
- [8]. Георгиев Н., Приложение на принципите за управление на безопасността при изследване и анализ на характеристиките на транспортните произшествия, Списание „Механика Транспорт Комуникации“, бр. 3, 2007.
- [9]. Mareş R., Stelea M., The application of event tree analysis in a work accident at maintenance operations, In MATEC Web of Conferences, vol. 121, p. 11013. EDP Sciences, 2017.

За контакти:

доц. д-р инж. Нели Ан. Арабаджиева - Калчева
катедра „Софтуерни и интернет технологии”
Технически университет - Варна
E-mail: n_kalcheva@tu-varna.bg

гл. ас. д-р инж. Ивелин М. Иванов
катедра „Софтуерни и интернет технологии”
Технически университет - Варна
E-mail: i.ivanov@tu-varna.bg

ас. инж. Павлина С. Линова
катедра „Софтуерни и интернет технологии”
Технически университет - Варна
E-mail: polly.linova@tu-varna.bg



**СОФТУЕРНИ И ИНТЕРНЕТ
ТЕХНОЛОГИИ**

МЕТОДИ ЗА АНАЛИЗ НА СЪВРЕМЕННИ ОПТИЧНИ ВЛАКНА

Жейно Ив. Жейнов, Десислава Д. Ангелова

Резюме: В статията се прави обзор на използваните днес оптични влакна. Описва се дизайнът на конвенционалните и фотонно кристалните оптични влакна. Разглеждат се някои техни свойства, особености и възможни приложения. Посочват се някои важни параметри на оптичното влакно за телекомуникация. Дава се класификация на конвенционалните и фотонно кристалните влакна. Разглеждат се обобщено методите за описание на разпространението на светлината в съвременните оптични вълноводи. Определят се разпространяващите се моди. Сравняват се възможностите на аналитичните и числените методи за анализ на различни типове оптични влакна с техните предимства и недостатъци. Дава се накратко методологията на анализа и някои резултати.

Ключови думи: Оптичен вълновод, фотонно кристално влакно, аналитични методи, числени методи, дизайн, разпространение на светлината, моди.

Methods of analysis of modern optical fibers

Zhejno I. Zhejnov, Desislava D. Angelova

Abstract: The article provides an overview of the optical fibers in use today. The design of the conventional and photonic crystal optical fibers is described. Some of their properties, features and possible applications are considered. Some important parameters of optical fiber for telecommunication are pointed. A classification of conventional and photonic crystal fibers is given. The methods for describing the propagation of light in modern optical waveguides are considered in general. The propagating modes are determined. The possibilities of analytical and numerical methods for the analysis of different types of optical fibers are compared with their advantages and disadvantages. The methodology of the analysis and some results are briefly given.

Keywords: Optical waveguide, photonic crystal fiber, analytical methods, numerical methods, design, light propagation, modes.

1. Увод

Оптичното влакно е изработено като тънък диелектричен цилиндър, в който се въвежда светлина, която взаимодейства с материала в него. Оптичните влакна се ползват за измерване на различни параметри, за усилване на сигнали, пренасяне на информация, за изработка на лазери. В телекомуникациите те са среда за прехвърляне на информация между две точки, като трансокеански кабели и кабели до дома, което позволява прехвърляне на данни с висока скорост. Те се употребяват в сензори за измерване на физически величини: температура, въртене, акустични и/или други величини. Оптичните сензори могат да работят нормално в екстремни среди с висока температура и високо напрежение. Освен това оптичните влакна се ползват за усилване на сигнали, за прехвърляне на енергия, предаване на изображения и др.

Технологията за оптични влакна се изследва активно от физици и инженери. Вложени са огромни ресурси и средства за подобряване на свойствата на оптичните влакна и изследване на нови видове влакна за различни цели като медицина, спектроскопия, сензори и телекомуникации.

По-нататък ще бъдат разгледани конвенционалните и други видове оптични влакна, основните закони и параметри, свързани с тяхната работа. Ще бъдат обсъдени основните оптични свойства на влакната, които влияят на производителността на системата, като

затихване, дисперсия и нелинейност. Разглежда се насочващият механизъм на светлината при различните видове фотонно кристални влакна (PCF) и техните специални свойства.

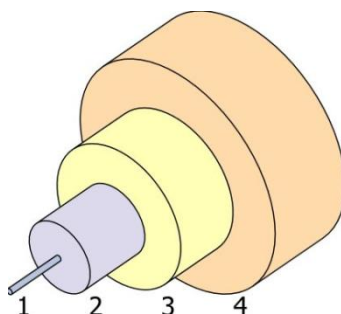
Анализът на оптичните влакна и устройства изисква провеждането на експериментални или числени изследвания. Експериментът отнема време и изисква големи физически ресурси. Теоретичното изследване се основава на числени симулации. Ползват се адаптирани аналитични или числени методи в зависимост от изследваното влакно. Целта е да се разбере как промяната на различните физически параметри на влакното и дизайна влияе върху оптичните му свойства. С анализ на представените систематични изследвания може да се реализира дизайн с желаните свойства на оптичното влакно.

2. Изложение

Конвенционалните технологии за оптични влакна със стъклена (силициева) сърцевина са усъвършенствани значително от първото успешно създадено стъклено влакно през 1957 г. През 1970 г. е произведено оптично влакно с ниски загуби - под 20dB/km. Разработването на легиран с ербий усилвател за оптично влакно през 80-те години на миналия век намали цената на комуникацията на дълги разстояния, тъй като оптичният сигнал може да се усилва директно, без да се преобразува в електрически. Така намалява броят на повторителите по оптичната линия.

Днес оптичните влакна са една от най-добрите комуникационни среди за комуникации на дълги разстояния. Те имат много ниски загуби, устойчиви са на електромагнитни смущения и имат широка честотна лента, като по този начин осигуряват най-високи скорости на предаване на данни. Оптичните влакна се използват в подводни интернет кабели, които осигуряват надежден високоскоростен интернет. В оптичните телекомуникации спектърът на дължината на вълната е разделен на три прозореца, съответно на дължини на вълните 850nm, 1300nm и 1550nm. Оптичните влакна и интегрираните оптични вълноводи намират широко приложение и в други области.

Оптичното влакно представлява цилиндричен диелектричен вълновод - фиг. 1.



Фиг. 1. Типична структура на силициево оптично влакно
1 - сърцевина, 2 – обвивка, 3 - буфер, 4 - защитно покритие

То е изградено от сърцевина с висок индекс на пречупване, в която се разпространява светлината. Сърцевината е заобиколена от обвивка с малко по-нисък индекс на пречупване от сърцевината и покрития, които осигуряват механична защита на влакното от физическо увреждане и предпазват стъклената повърхност от влага и прахови частици.

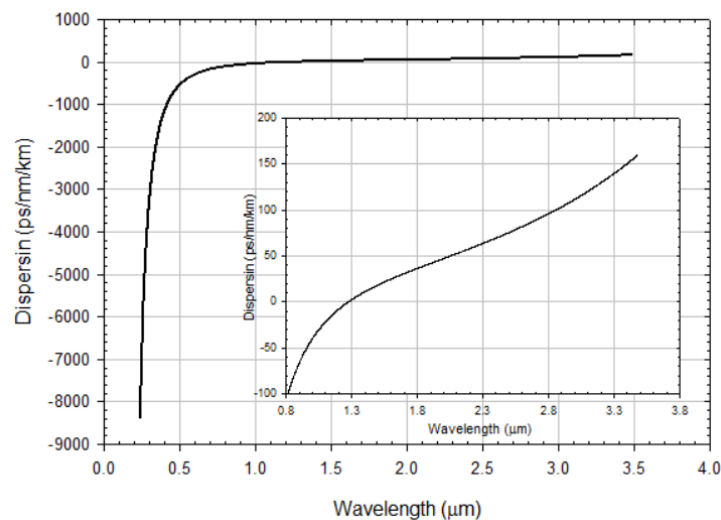
Размерът на сърцевината и разликата в индекса на пречупване между сърцевината и обвивката играят основна роля при определяне на свойствата на влакното. Важен оптичен параметър на всяка прозрачна среда е индексът на пречупване n , който се определя като отношение на скоростта на светлината в материала v и скоростта на светлината във вакуум c .

$$n = \frac{v}{c} \quad (1)$$

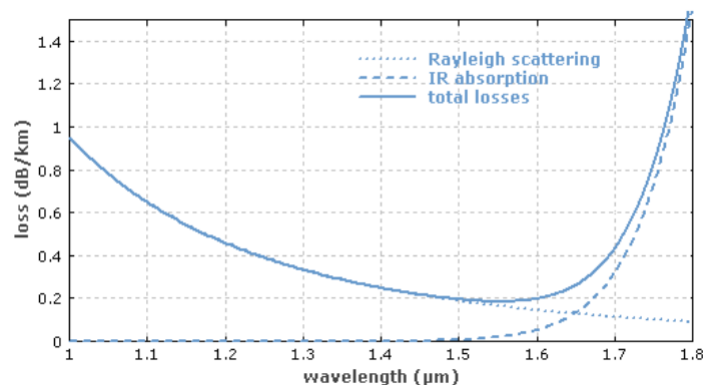
Промяната на индекса на пречупване с дължината на вълната води до разширяването на продължителността на импулса (по време) и дължина (пространствено). Индексът на пречупване на разтопения силициев диоксид силно зависи от дължината на вълната. Промяната в индекса на пречупване при по-ниски дължини на вълната е много рязка, което води до висока стойност на дисперсията.

Разпространението на светлината в оптично влакно може да бъде обяснено чрез закона на Снелиус. Част от светлинните лъчи, пътуващи от сърцевината към обвивката, достигат повърхността ядро/обвивка под ъгъл, по-голям от критичния. Това предизвиква пълно вътрешно отражение. Тези лъчи се отразяват обратно в сърцевината и не преминават в обвивката. Така се удържа светлината в сърцевината на влакното. Всички други лъчи се пречупват и излизат от ядрото и не се разпространяват в него.

В оптичните системи две основни свойства ограничават възможностите за предаване: затихването (загуба на мощност) и дисперсията (разширението на импулса). И двете могат да имат вредно въздействие върху телекомуникациите. Техните стойности трябва да бъдат малки или нулеви, така че ефектът им да е поносим. Дисперсията на силициев материал е показана на фиг. 2, а вътрешните загуби - на фиг. 3 [1].



Фиг. 2. Дисперсия на материал разтопен силиций.



Фиг. 3. Вътрешни загуби на материал разтопен силиций

Някои приложения изискват различна от нула или по-голяма стойност на дисперсията. Такъв е случаят с нелинейните ефекти, при които е необходима ненулева дисперсия, за да

има значително въздействие върху сигнала. Големи стойности на дисперсията са необходими в специално проектираните с дисперсия влакна, за да се компенсира дисперсията на конвенционалните телекомуникационни оптични влакна. Дисперсията трябва да се контролира, за да отговаря на необходимото приложение. Независимо от начина на производство има два вида оптични влакна: едномодови влакна (Single Mode Fiber) и многомодови влакна (MultiMode Fiber).

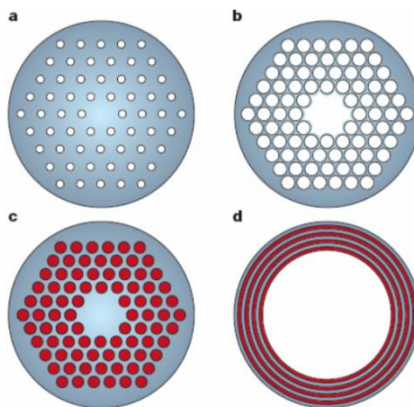
Радиусът на ядрото на SMF е много по-малък отколкото при MMF. В SMF само основната мода може да се разпространява през ядрото, докато броят на модите в MMF е приблизително $(V^2/2)$, където V параметърът е [2]:

$$V = \frac{2\pi a}{\lambda} \sqrt{n_1^2 - n_2^2} \quad (2)$$

В (2) a е радиусът на сърцевината на влакното, λ е дължината на светлинната вълна, n_1 е коефициентът на пречупване на сърцевината, а n_2 - коефициентът на пречупване на обвивката на влакното.

MMF изпитват междумодова дисперсия. Едномодовите влакна имат по-ниски загуби и по-ниска дисперсия от MMF. Обаче MMF имат по-висока ефективност при свързване от SMF. Въпреки че SMF са по-скъпи от MMF при производството, те са за предпочитане за комуникации на дълги разстояния, тъй като SMF поддържа само основната мода, поради което загубите и дисперсията са по-ниски. Това изисква по-малко стъпала за регенерация на сигнала и по-ниска обща цена.

Дизайнът на PCF е различен от конвенционалните силициеви влакна. PCF могат да бъдат изработени от един материал с двуизмерни фотонни кристали или периодични решетки от въздушни отвори, успоредни на оста на влакното, за да се получи форма на обвивка и сърцевина [3]. Няколко вида PCF са показани на фиг. 4: (a) влакно със сърцевина от чист Si и въздушни отвори в обвивката; (b) влакно с въздушни тръби в сърцевината, където светлината се удържа заради ефекта на лентата на пропускане; (c) сърцевината е направена от чист силициев диоксид, а дупките в обвивката са пълни с течност с висок индекс на пречупване и (d) кухи цилиндрични многослойни влакна с изцяло масивна обвивка - влакна на Bragg.



Фиг. 4. PCF влакна с различен дизайн.

Фотонните кристални влакна могат да бъдат класифицирани в две основни категории според механизма на разпространение на светлината: влакна с висок индекс на пречупване на сърцевината, в които разпространението на светлината става по начин, подобен на конвенционалните влакна; вторият тип PCF е направляващи влакна с нисък индекс на пречупване на сърцевината, известни като влакна с куха сърцевина /НС/. Ядрото на НС влакната е направено от материал с по-нисък индекс от обвивката и светлината се насочва

или чрез PBG – фотонни забранени зони или антирезонансен ефект. HC влакната са допълнително подкласифицирани в HC-PBF, влакна на Bragg, влакна на Kagome и HC-NCF /Hollow-Core Negative Curvature Fiber/ - влакна с куха сърцевина с отрицателни кривини на границите на сърцевината, където светлината е ограничена в сърцевина с по-нисък индекс на пречупване от обвивката. HC-NCF влакната са създадени през 2010 г.

Броят на модите, поддържани от PCF, се определя подобно на конвенционалните оптични влакна с V параметъра. Формулата се манипулира леко, за да отговаря на PCF, тъй като няма добре дефинирана сърцевина или постоянна стойност на индекса на пречупване за обвивката. Коефициентът на пречупване на обвивката е функция на дължината на вълната, която се променя в зависимост от взаимодействието на светлината с двата материала (сърцевина и въздух) на обвивката. При по-дълги вълни светлината се разпространява вътре в отворите и води до промяна на индекса на пречупване в зависимост от дизайна. Формулата за V_{PCF} е показана по-долу [4].

$$V_{PCF} = \frac{2\pi A}{\lambda} \sqrt{n_1^2 - n_{cl}^2(\lambda)} \quad (3)$$

където $n_{cl}(\lambda)$ е среден индекс на пречупване на обвивката на PCF.

В едномонов режим насочването на светлината зависи от индекса на пречупване на сърцевината и средния индекс на обвивката на фотонния кристал, който е функция на дължината на вълната. В конвенционалните оптични влакна има гранична честота, над която влакното вече не е едномоново. Областта за работа в едномонов режим в PCF се разширява, тъй като средният индекс на пречупване на фотонния кристал зависи от дължината на вълната. При по-къса дължина на вълната модовото поле е по-концентрирано в стъкления материал, тъй като дължината на вълната се увеличава, полето се разширява повече във въздушните отвори. Следователно при по-къси дължини на вълната средният индекс на обвивката е близък до индекса на материала. Това води до малка разлика в индекса между сърцевината и обвивката, така че стойността на V_{PCF} да е малка дори при по-къси дължини на вълните. Поради зависимостта на коефициента на пречупване от дължината на вълната, PCF може да бъде проектиран да бъде едномонов, когато е изпълнено условието [4]:

$$V_{PCF} < \pi \quad (4)$$

Анализът на оптични устройства е сложен процес, който изисква огромно количество математически операции. Времето, необходимо за извършването им, е голямо и е невъзможно да се работи без помощта на изчислителни ресурси. Поради това се използват интензивно както аналитични, така и числени методи.

Аналитичните методи се прилагат за решаване на електромагнитни проблеми. Те се използват при прости конструкции като плочи, вкопани и ребрени вълноводи. Аналитичните методи дават точни решения, но те са ограничени и неприложими за реални случаи, включващи сложни геометрии и нехомогенни области. Аналитичните методи бяха по-приложими преди навлизането на компютрите. Те обаче имат важна роля в развитието на числените методи.

Има различни аналитични методи, използвани за получаване на решения за прости оптични вълноводи. Анализът за сложни оптични вълноводни структури с нехомогенност в пространствената област не е възможен. Тогава се правят приближения и предположения за проблеми, които трябва да бъдат решени с числени методи.

Апроксимационните или числените методи, както показва името, се основават на приближаване на решението чрез прилагане на числени или физични предположения към проблема, който представлява интерес. Това внася известна допустима грешка. Направените приближения ги класифицират в два основни класа.

Първият клас са асимптотичните методи. Тези методи изискват фундаментално приближение в уравненията на Максвел. Апроксимацията във физиката ограничава използването на тези методи за общи случаи. Те обикновено се използват за големи метални и проводящи предмети, което е недостатък. Техните предимства са ниските изчислителни ресурси и високата ефективност [5].

Вторият клас числени методи са изцяло вълновите методи. При тези методи приближението се прави числено и няма първоначално физическо приближение. Тези методи се класифицират допълнително според използваната формула, интегрална или диференциална, според домейна, в който работят, по време или честота, като някои от методите могат да бъдат приложени в двете области. Те включват три основни стъпки за решаване на проблема. Първо проблемът трябва да бъде дефиниран чрез контролно уравнение, което се получава от уравненията на Максвел. Втората стъпка е дискретизация на домейна в неприпокриващи се поддомейни, за да се представят произволни нехомогенни диелектрични вълноводи, така че всеки поддомейн или елемент да е хомогенен. Дискретизацията води до набор от уравнения, които са описани в матрична форма. Последната стъпка изисква решение на матрицата от неизвестни с помощта на ефективни алгоритми. Най-популярните изцяло вълнови методи са описани накратко по-долу.

Методите на честотната област са подходящи за изследване на теснолентови приложения в стационарно състояние, докато методите на времевата област са подходящи за широколентови преходни приложения. Дискретизацията в интегралната формула включва важната повърхност на проблема, като по този начин се намаляват изчислителните ресурси. Това води обаче до лошо представяне на сложни структури и нехомогенни материали. Интегралната формула може да се използва за изследване на проблеми с разсейване и отворен домейн. В диференциалните формули дискретизацията включва пълния обем. Това изисква повече изчисления, но може ефективно да представи нехомогенни среди и сложни структури. Диференциалните формули могат да бъдат приложени в граничен проблем.

Няма универсален избор на най-добър метод при всички случаи. Всеки метод има предимства и недостатъци и се избира подходящият за всеки отделен случай. Всеки метод обаче се различава в предварителната обработка на уравнението на Максвел за получаване на крайната формула.

Често срещан аспект при изцяло вълновите методи е да се дискретизира системният домейн на поддомейни. По този начин решението става почти независимо от системата [6].

Търси се представяне на компонентите E и H на вълната, разпространяваща се по оста z като хармонична функция на времето:

$$E(x, y, z) = E_v(x, y) \exp(-j\beta_v z) \text{ и} \quad (5)$$

$$H(x, y, z) = H_v(x, y) \exp(-j\beta_v z), \quad (6)$$

където E е напрегнатостта на електрическото поле, а H е напрегнатостта на магнитното поле, v е индекс на съответната мода (тип вълна) [2].

Нека в (5) предположим, че търсената вълнова функция Φ е апроксимирана с функцията Φ_a . При това се получава резидуална (остатъчна) грешка R , като чрез намаляване на разликата между Φ и Φ_a усреднената грешка R може да бъде намалена до нула в цялата анализирана област.

$$\frac{\partial^2 \Phi_a}{\partial x^2} + \frac{\partial^2 \Phi_a}{\partial y^2} + k_0^2 (\epsilon_r - n_{eff}^2) \Phi_a = R \quad (7)$$

Въвеждаме тегловна функция Ψ и получаваме:

$$\iint \Psi R \, dx dy = 0. \quad (8)$$

Уравнението за елемент с номер e в отделните елементи на областта е:

$$\left[\int_{\Gamma_e} \Psi_e \frac{\partial \Phi a_e}{\partial n} d\Gamma \right] - \iint_e \left(\frac{\partial \Psi_e}{\partial x} \frac{\partial \Phi a_e}{\partial x} + \frac{\partial \Psi_e}{\partial y} \frac{\partial \Phi a_e}{\partial y} \right) dx dy + \iint_e \Psi_e k_0^2 (\varepsilon_r - n_{eff}^2) \Phi a_e \, dx dy = 0. \quad (9)$$

Тук Ψ_e и Φ_e в елемента с номер e се разширяват чрез използване на същите базови функции:

$$\Phi a_e = \sum_i^{M_e} \Phi_{ei} N_i = [N_e]^T [\Phi_e], \quad (10)$$

$$\Psi_e = \sum_i^{M_e} \Phi_{ei} N_i = [N_e]^T [\Phi_e], \quad (11)$$

където M_e е броят на възлите в елемента e . $[N_e]$ и $[\Phi_e]$ се задават като набор коефициенти и кореспондират с фигурната функция и полевия компонент.

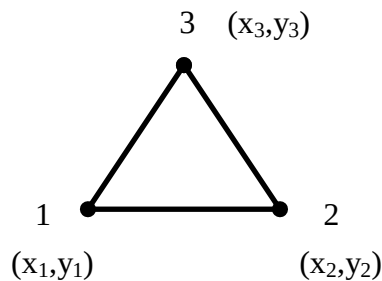
$$[N_e] = [N_1 \, N_2 \, N_3 \, \dots \, N_{M_e}]^T, \quad (12)$$

$$[\Phi_e] = [\Phi_1 \, \Phi_2 \, \Phi_3 \, \dots \, \Phi_{M_e}]^T. \quad (13)$$

Замествайки (10), (11), (12), (13) в (9) окончателно се получава матрично уравнение:

$$([A] - \lambda^2 [B])\{\Phi\} = 0, \text{ където } \{\Phi\} = \sum_e \{\Phi_e\}, \quad \{A\} = \sum_e \{A_e\}, \quad \{B\} = \sum_e \{B_e\}. \quad (14)$$

При 2D анализ за апроксимация на разпределението на полето се използват триъгълни елементи и полиномиална апроксимация. Истинската вълнова функция се апроксимира с линейна функция или с квадратна функция. Апроксимацията с полином от по-висок ред води до по-точни резултати, обаче усложнява анализа и изисква повече памет. Широко използвани са триъгълни елементи от първи ред с 3 възела - фиг. 5.



Фиг. 5. Триъгълен елемент от първи ред

Например за триъгълен елемент от първи ред вълновата функция $\Phi(x,y)$ в произволни координати (x,y) вътре в елемента се разширява чрез функциите на формата N_1, N_2, N_3 .

Полетата Φ_1, Φ_2, Φ_3 във върховете на триъгълника (възли 1,2,3) я представят така:

$$\Phi(x,y) = N_1 \Phi_1 + N_2 \Phi_2 + N_3 \Phi_3 = [N]^T \{\Phi\}, \quad \text{където} \quad (15)$$

$$[N]^T = [N_1 \, N_2 \, N_3]^T, \quad \{\Phi\} = (\Phi_1 \, \Phi_2 \, \Phi_3)^T. \quad (16)$$

Функциите на формата са линейни функции на координатите на точката (x,y) и координатите на върховете на елемента и могат да бъдат лесно пресметнати. Чрез заместване на съответните изрази в (9) се извеждат стойности за елементите на матриците A и B за триъгълния елемент e . Главното матрично уравнение е:

$$([K] - \beta^2 [M])\{\Phi\} = \{0\}, \text{ където } \beta = k_0 \eta_{eff} \text{ е константата на разпространение.} \quad (17)$$

Стойността на η_{eff} се определя от компонентата на полето и ПП n_r . Матриците, участващи в уравнението, са:

$$[K] = \sum_e \{(-\eta_e^2 ([A_e] + [B_e]) + k_0^2 \xi_e^2 [C_e])\}, \quad (18)$$

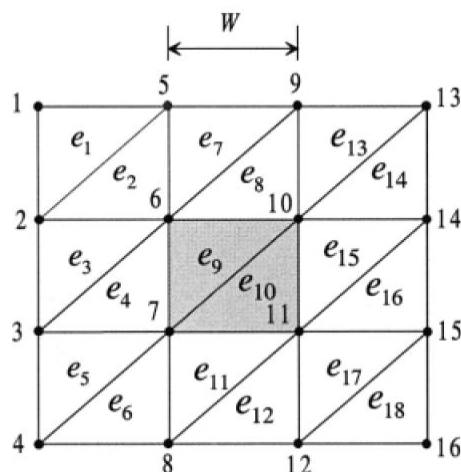
$$[M] = \sum_e -\eta_e^2 [C_e], \quad (19)$$

$$\{\Phi\} = \sum_e \Phi_e, \text{ като матриците на коефициентите се пресмятат така:} \quad (20)$$

$$[A_e] = \iint_e \frac{\partial [N_e]}{\partial x} \frac{\partial [N_e]^T}{\partial x} dx dy, [B_e] = \iint_e \frac{\partial [N_e]}{\partial y} \frac{\partial [N_e]^T}{\partial y} dx dy, [C_e] = \iint_e [N_e] [N_e]^T dx dy. \quad (21)$$

Граничните условия са две. Едното изисква вълновите функции на границата Γ да са 0, а другото – вариацията на вълновата функция да бъде много малка:

$$\frac{\partial \Phi_i}{\partial n} = 0, \quad \Phi_i = 0. \quad (22)$$



Фиг. 6. Мрежа от триъгълни елементи

Ще покажем алгоритъма за получаване на стойностите на полето за случай на потопен оптичен вълновод, който е разделен на 18 триъгълни елемента от I ред $e_1, e_2, e_3, \dots, e_{18}$. Сърцевината с ширина w включва 2 елемента – e_9 и e_{10} . Фиг. 6 показва начина на разделянето на напречното сечение на вълновода на сегменти, като са означени номерата на възлите. Правим следните стъпки:

1. Покриваме цялата анализирана област с мрежа от триъгълни елементи.
2. Определяме координатите на възлите /възел i е с координати (x_i, y_i) / и локалните координати на елементите.
3. Пресмятаме матриците $[A_e]$, $[B_e]$, $[C_e]$ за всеки триъгълен елемент e .
4. Съгласно с (21), (18), (19) попълваме глобалните матрици $[K]$ и $[M]$, чийто редове и колони кореспондират с номерата на елементите и локалните координати.
5. За всеки възел, принадлежащ на повече от 1 триъгълен елемент (на фиг. 5 напр. възел 6 е включен в 6 триъгълника), се попълват глобалните матрици $[K]$ и $[M]$ за всички триъгълни елементи.
6. Включваме (22) в $[K]$ и $[M]$.

7. Пресмятаме β или n_{eff} чрез решаване на матричното уравнение (17).

Методът на моментите (МоМ), известен също като метод на граничния елемент, има предимство при проблеми, включващи отворени региони. Прилага се най-вече в честотна област. Това е най-широко използваният метод в антенното инженерство. Математическата формулировка, представяща електромагнитните полета, е реализирана в МоМ чрез дискретизиране на интегралното уравнение. Той е изчислително ефективен, но изисква сложна математика в сравнение с други методи. Това е предпочитаният метод за разсейване на честотната област и проблеми с излъчването.

Методът на крайните разлики (FDM) е най-широко използваният метод при електромагнитни проблеми поради това, че е ясен и лесен за прилагане. Домейнът е разделен на мрежа и решението се намира за всяка точка от мрежата. Той прилага диференциалните уравнения на Максуел, за да изведе вълновото уравнение и да се намери решение за модите.

Диференциалният оператор се заменя с оператори за разлика чрез използване на формула за централна разлика. Има друг метод, който споделя същата концепция за свързване, който е известен като метод на крайната разлика във времевата област (FDTD). Използва се при анализа на проблеми във времевата област чрез директно дискретизиране на уравненията на Максуел. Двата метода са много ефективни, тъй като изискват малко операции върху всяка точка от мрежата [7]. Недостатъкът на FDM е, че мрежата трябва да бъде еднаква, за да може методът да работи добре.

Метод на крайните елементи (FEM) използва стълбична апроксимация на границите, които не са в съответствие с мрежата. Решението е достъпно само във възлите.

Методът на крайните елементи може да представя точно сложни домейни, тъй като дискретизира домейна в мрежа от неструктурирани елементи и по този начин приближава по-добре границите. Типичният елемент е триъгълник в 2D и тетраедър в 3D. FEM е разработен да се прилага основно при проблеми в честотната област. Може да се приложи във времевата област. FEM е ефективен за микровълнови устройства и симулации на проблеми със собствените стойности [8].

3. Заключение

FEM, FDM и МоМ са основните числени методи, използвани в електромагнитния анализ. За анализ на модерни оптични влакна и вълноводи със сложна структура на сърцевината и обвивката се избира един от тези методи. При произволните граници на PCF структурите FEM се предпочита, защото прави по-добро приближение на границите им.

Литература

- [1]. Снайдер А., Дж. Лав. Теория оптических волноводов. Превод от англ.. Радио и связь, 1987.
- [2]. Kawano K., T. Kitoh. Introduction to optical waveguide analysis. John Wiley and Sons, Inc., 2001.
- [3]. Russell, P. St.J. Photonic crystal fibers, // Science, 2003, vol. 299, pp. 358–362.
- [4]. Quimby, R. S. Photonics and Lasers: An Introduction. John Wiley & Sons, 2006.
- [5]. Sheng, X., W. Song, *Essentials of Computational Electromagnetics*. Singapore: John Wiley & Sons, 2012.
- [6]. Garg, R. *Analytical and Computational Methods in Electromagnetics*. Norwood: Artech House, 2008.

- [7]. Rylander, T., P. Ingelström, A. Bondeson, *Computational Electromagnetics*. New York: Springer, 2013.
- [8]. Davidson, D. B. *Computational Electromagnetics for RF and Microwave Engineering*. Cambridge: Cambridge University Press, 2010.

За контакти:

Доц. д-р Жейно Ив. Жейнов
катедра „Компютърни науки и технологии”
Технически университет – гр. Варна
E-mail: zh_viv@tu-varna.bg



ПРЕМИНАВАНЕ НА СИГНАЛИТЕ И ШУМОВЕТЕ ПРЕЗ ЛИНЕЙНИ ВЕРИГИ И ХАРАКТЕРИСТИКИ НА СЪГЛАСУВАН ФИЛТЪР

Георги Николов

Резюме: Статията е предназначена да запознае аудиторията с определения и алгоритми, изследващи процесите в линейните радиотехнически вериги чрез математически модели. Извършва се анализ на поведението на сигналите и шумовете в тях. Разглеждат се методите за потенциалното обработване на същите и обосновка на целта за оптимална филтрация чрез съгласуван филтър.

Ключови думи: линейна верига, преходна характеристика, импулсна характеристика, корелационна функция, сигнал, шум, съгласуван филтър.

TRANSMISSION OF SIGNALS AND NOISE THROUGH LINEAR CHAINS CHARACTERISTICS OF A MATCHED COHERENT FILTER

Georgi Nikolov

Abstract: The article is designed to acquaint the audience with definitions and algorithms that study the processes in linear radio circuits through mathematical models. An analysis of the behavior of the signals and the noises in them is performed. The methods for their potential processing and justification of the goal for optimal filtration through a coherent filter are considered.

Key words: linear circuit, transient response, impulse response, correlation function, signal, noise, coherent filter.

1. Въведение

При анализа на преминаването и обработката на сигналите и шумовете през радиотехническите линейни вериги (системи) се налага подходът по зададени величини (входен сигнал, входен шум) да се определят крайните резултати от реализираните процеси на изхода на тези вериги (системи).

Според общото определение „линейна верига“ е такова устройство, което извършва линейно преобразуване на постъпващите на входа му сигнали. Тези сигнали може да носят полезна информация, но може и да са с характера на смущения и шумове. В линейните вериги се извършват линейни операции като усилване, филтриране, предаване по „дълги линии“. Основна отличителна характеристика на линейните вериги е, че операциите, които те изпълняват, не променят спектъра на сигнала.

В сравнение с тях нелинейните вериги (системи) като детектори, умножители /делители на честота, ограничители, извършват нелинейни операции, водещи до промяна на спектъра на сигнала.

Аналогична е формулировката в общия анализ, определяща характера както на входните полезни сигнали като случайни величини, така и на въздействащите шумове. Подходът може се сведе до представянето им с краен на брой реализации и последващото им интегриране, но в настоящата статия задачата се акцентира върху постъпването на входа на линейната радиотехническа верига на сигнал с познат спектър, като се ще се търси величината на енергийния спектър на изходния сигнал (шум).

2. Свойства, характеристики и параметри, определящи преминаването на сигналите през линейни вериги и оптималната им филтрация

Енергийният спектър на сигнала [1], [2], [4], [8] $G(\omega)$ по своята същност се представя като квадратична стойност на комплексната спектрална плътност [5], [7], [8] $\Phi(j\omega)$:

$$G(\omega) = |\Phi(j\omega)|^2 \quad (1)$$

В редица случаи намирането на $G(\omega)$ е свързано със затруднения, за чието избягване е достатъчно определянето на корелационната функция [2], [8] $K(\tau)$.

В ролята на променливата τ е разликата от моментите във времето, между които се търси взаимната връзка

$$\tau = t_2 - t_1$$

Математическият апарат, способстващ анализа, е Теоремата на Винер-Хинчин [4], [6], [7], която определя връзката между автокорелационната функция на стационарния случаен процес и неговия спектър.

Разглежда се една конкретна реализация на сигнала $Sr(t)$:

$$K(\tau) = \lim_{T \rightarrow \infty} \frac{1}{T} \int_{-\infty}^{+\infty} Sr(t) \cdot Sr(t + \tau) dt ,$$

която за реален сигнал за време T е:

$$K(\tau) = \lim_{T \rightarrow \infty} \frac{1}{T} \int_{-T/2}^{+T/2} Sr(t) \cdot Sr(t + \tau) dt \quad (2)$$

Използват се две свойства на комплексната спектрална плътност [5], [7], [8] $\Phi(j\omega)$:

- Първо – правилото на изместването (сдвига/shift [3], [7], [9]), заключаващо, че ако сигналът се измести „назад“ по оста на времето с величина τ , то това съответства на умножение на нейната комплексна спектрална плътност с величина $e^{-i\omega\tau}$.

Доказателство:

Нека е зададена реализация на сигнала $Sr(t)$. Изместена във времето на τ , тя има вида:

$$\int_{-\infty}^{+\infty} Sr(t - \tau) \cdot e^{-i\omega t} dt ,$$

като при смяна на променливата $t - \tau = \theta$, следва $t = \tau + \theta$ и $dt = d\theta$.

$$\begin{aligned} \int_{-\infty}^{+\infty} Sr(t - \tau) \cdot e^{-i\omega t} dt &= \int_{-\infty}^{+\infty} Sr(\theta) \cdot e^{-i\omega(\theta + \tau)} d\theta = \int_{-\infty}^{+\infty} Sr(\theta) \cdot e^{-i\omega\theta} \cdot e^{-i\omega\tau} d\theta = \\ &= e^{-i\omega\tau} \int_{-\infty}^{+\infty} Sr(\theta) \cdot e^{-i\omega\theta} d\theta = \Phi(j\omega) \cdot e^{-i\omega\tau} \end{aligned} \quad (3)$$

- Второ – Теоремата на Пърсивал [2], [7], заключаваща, че ако две реализации на сигнала $Sr(t)$ и $Se(t)$ удовлетворяват Принципа на Дирихле (който гласи, че ако m -брой обекта са разположени в n -брой клетки и ако $m > n$, то поне една от клетките съдържа не по-малко от m/n обекта, а също така в поне една друга клетка има не повече от m/n обекта) и имат комплексни спектрални плътности $\Phi_r(j\omega)$ и $\Phi_e(j\omega)$, то за тях може да се запише:

$$\int_{-\infty}^{+\infty} Sr(t) \cdot Se(t) dt = \int_{-\infty}^{+\infty} \Phi_r(j\omega) \cdot \Phi_e^*(j\omega) d\omega$$

Корелационната функция [2], [8] добива вида:

$$Kr(\tau) = \lim_{T \rightarrow \infty} \frac{1}{T} \int_{-T/2}^{+T/2} Sr(t) \cdot Sr(t + \tau) dt = K(\tau) = \lim_{T \rightarrow \infty} \frac{1}{T} \int_{-T/2}^{+T/2} \Phi_r(j\omega) \cdot \Phi_r^*(j\omega) \cdot e^{i\omega\tau} d\omega$$

Знакът пред $j\omega\tau$ е „плюс“, защото не е „минус“, а $\Phi_e^* = \Phi_r^*(j\omega)$ в случая на заместване $Se = Sr(t+\tau)$ и т.н.

Имайки предвид, че $S^* \cdot S = |S|^2$, се получава:

$$Kr(\tau) = \lim_{T \rightarrow \infty} \frac{1}{T} \int_{-\infty}^{+\infty} |\Phi r(j\omega)|^2 \cdot e^{i\omega\tau} d\omega = \int_{-\infty}^{+\infty} Gr(\omega) \cdot e^{i\omega\tau} d\omega \quad (4)$$

Следователно корелационната функция [2], [8] на реализацията по времето може да се получи и е свързана с енергийния спектър на сигнала [1], [2], [4], [8] чрез Обратното преобразование на Фурие [1], [2], [3], [5], [8], [9].

Енергийния спектър $Gr(\omega)$ може да се получи от $Kr(\tau)$ чрез Правото преобразование на Фурие.

$$Gr(\omega) = \int_{-\infty}^{+\infty} Kr(\tau) \cdot e^{-i\omega\tau} d\tau$$

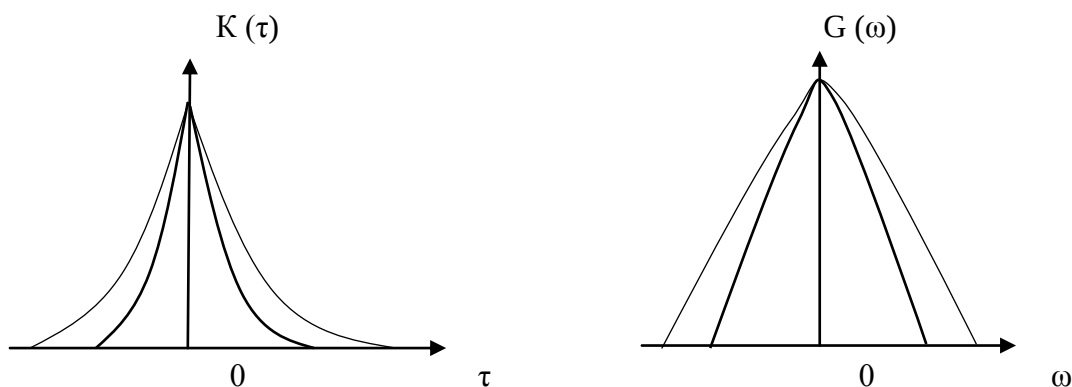
Ако се вземе предвид математическото очакване $M \{Kr(\tau)\}$, за целия сигнал се получават двата начина на изразяване с Теоремата на Винер-Хинчин [4], [6], [7]:

$$G(\omega) = \int_{-\infty}^{+\infty} K(\tau) \cdot e^{-i\omega\tau} d\tau \quad (5)$$

$$K(\tau) = \int_{-\infty}^{+\infty} G(\omega) \cdot e^{i\omega\tau} d\omega \quad (6)$$

Ако сигналите са стационарни, тогава функциите са четни и може да се запише:

$$K(\tau) = 2 \int_0^{\infty} G(\omega) \cdot \cos \omega\tau d\omega \quad \text{и} \quad G(\omega) = 2 \int_0^{\infty} K(\tau) \cdot \cos \omega\tau d\tau$$



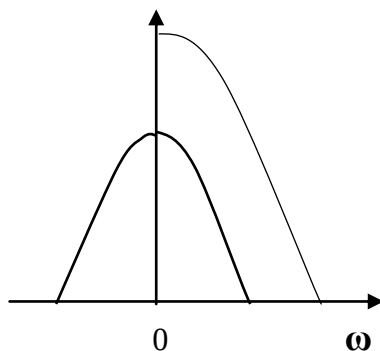
Фиг. 1. Корелационна функция на реализацията и енергиен спектър на сигнала

$G(\omega)$ се нарича още математически спектър на сигнала [1], [8], [9].

С цел практическата насоченост на анализа се въвежда понятието физически енергиен спектър, защото за $G(\omega)$ няма да може да се осъществят честоти с отрицателна стойност ($\omega < 0$) и се означава с $Go(\omega)$.

$$Go(\omega) = 4 \int_0^{\infty} K(\tau) \cdot \cos \tau d\tau \quad (7)$$

$$Go(\omega) \quad G(\omega)$$



Фиг. 2. Физически енергиен спектър на сигнала

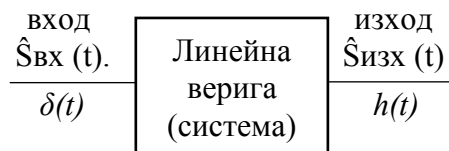
Както вече бе упоменато, при анализа на радиотехническа верига/система се налага по зададен входен сигнал да се прецени какво ще се получи като изходен такъв (на изхода на системата).

Във връзка с това се оценяват две основни характеристики:

- Предавателна характеристика [2], [8], [9] – отношението на комплексните амплитуди на изходния и входния сигнали:

$$K(j\omega) = \frac{S_{\text{изх}}(t)}{S_{\text{вх}}(t)} \quad (8)$$

- Импулсна характеристика [2], [8], [9] $h(t)$ – реакцията на радиотехническата система във времето, когато за входен сигнал се използва делта-функцията $\delta(t)$.



Фиг. 3. Входни и изходни величини на линейна верига

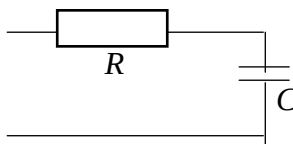
На изхода $h(t)$, което по същество е изходният сигнал, има същата размерност, съответна на тази на входния сигнал. Доказва се, че $K(j\omega)$ и $h(t)$ са взаимно свързани с Правото и Обратното преобразования на Фурие[1], [2], [3], [5], [8], [9]:

$$h(t) = \int_{-\infty}^{+\infty} K(j\omega) \cdot e^{i\omega t} d\omega \quad (9)$$

$$K(j\omega) = \int_{-\infty}^{+\infty} h(t) \cdot e^{-i\omega t} dt \quad (10)$$

Ако в една линейна система, която има $h(t)$ и $K(j\omega)$, колкото е необходимо $h(t)$ да се приближава до $\delta(t)$ функцията, толкова трябва да се разширява честотната лента и обратно.

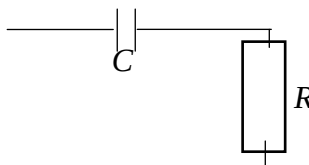
Приближавайки се до $\delta(t)$, $h(t)$ ще бъде със стремеж до единичен безкраен скок – т.е. сигнал с безкраен спектър.



Фиг. 4. Интегрираща верига

$$K(j\omega) = \frac{1}{1 + j\omega R.C} \quad (11)$$

$$h(t) = \frac{1}{R.C} e^{-t/R.C} \quad (12)$$



Фиг. 5. Диференцираща верига

$$K(j\omega) = \frac{j\omega R.C}{1 + j\omega R.C} \quad (13)$$

$$h(t) = \delta(t) - \frac{1}{R.C} e^{-t/R.C} \quad (14)$$

Доколкото задачата касае намирането на енергийния спектър на сигнала (или шума) [1], [2], [4], [8] на изхода на линейната система, ако се знае $K(j\omega)$, решението използва равенството:

$$G_{изх}(\omega) = G_{вх}(\omega) \cdot |K(j\omega)|^2 \quad (15)$$

Оттук чрез Обратното преобразование на Фурие [1], [2], [3], [5], [8], [9] може да бъде намерена корелационната функция [2], [8] на сигнала.

Нека на входа има „бял шум“ със съответния си енергиен спектър $G_{вх}(\omega) = \frac{N_0}{2}$,

тогава изходната му величина е: $G_{изх}(\omega) = \frac{N_0}{2} |K(j\omega)|^2$

$$G_{изх}(\omega) = N_0 \cdot |K(j\omega)|^2 \quad (16)$$

Мощността на шума в този случай е:

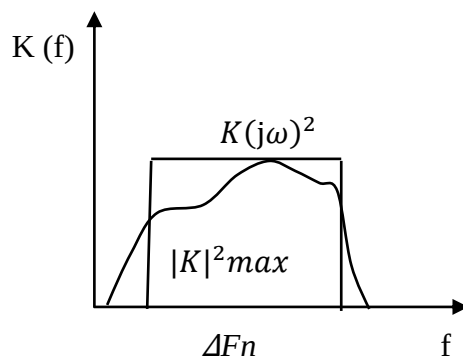
$$P_n = \int_{-\infty}^{+\infty} G_{изх}(\omega) d\omega = N_0 \int_0^{\infty} |K(j\omega)|^2 d\omega$$

Важен критерий за оценка на процесите е „шумовата лента на пропускане“ ΔF_n , определена от израза:

$$\Delta F_n = \frac{\int_0^{\infty} |K(j\omega)|^2 d\omega}{|K|^2_{max}} \quad (17)$$

Окончателният израз за мощността има вида:

$$P_n = N_0 \cdot |K|^2_{max} \cdot \Delta F_n \quad (18)$$



Фиг. 6. Корелационна функция на сигнала, отнесена към шумова лента на пропускане

На показаните графики (Фиг. 6.) лицата на двете геометрични фигури са еднакви, което онагледява ролята на шумовата лента на пропускане ΔF_n .

Практически свързано, шумовата лента на пропускане е характеристиката на линейните радиотехнически вериги (системи), използвани в качеството на честотни филтри.

С оглед направените до тук разсъждения, може да се премине към разглеждане на взаимовръзките на оптималната филтрация [1], [2], [8].

В основата си тя се базира като функционалност на съгласувания филтър [2], [3], [8].

Нека на входа на линейната радиотехническа верига постъпва сума $u_{вх}(t)$ на позната реализация на сигнала $Sr_{вх}(t)$, смесена с шум $n(t)$.

$$u_{вх}(t) = Sr_{вх}(t) + n(t)$$

Доколкото веригата е линейна, то изходното напрежение ще бъде сума от напреженията, предизвикани от преминаването на сигнала и шума през нея.

По същество в линейните вериги се използва принципа на суперпозицията. В радиотехниката в редица случаи се поставя задачата за отделяне на сигнала от шума, като за конкретно предназначение на дадено устройство тя се формулира различно.

Най-интересна е формулировката при зададена входна реализация на сигнала $Sr_{вх}(t)$ да се намери предавателната функция на линейната верига $Kr(j\omega)$, така че на изхода и в момент от време $t = to$, наречен „момент на отчета“, отношението на модула на напрежението на изходния сигнал към средноквадратичната стойност на шума да е максимално (19).

Такава линейна радиотехническа верига се определя като съгласувана с реализацията $Sr(t)$ или още се нарича „съгласуван филтър“ за $Sr(t)$.

$$\max \frac{|Sr_{изх}(t)|}{\sqrt{Pn}} \quad (19)$$

Използват се вече познатите определения:

$$Pn = N_0 \int_0^{\infty} |K(j\omega)|^2 d\omega \quad \text{или} \quad Pn = \frac{N_0}{2} \int_{-\infty}^{+\infty} |K(j\omega)|^2 d\omega$$

$$Gr_{изх}(\omega) = Gr_{вх}(\omega) \cdot |K(j\omega)|^2$$

Известно е, че $Gr(\omega) = |\Phi(j\omega)|^2$, откъдето следва:

$$|\Phi_{изх}(j\omega)|^2 = |\Phi_{вх}(j\omega)|^2 \cdot |K(j\omega)|^2$$

$$\Phi_{изх}(j\omega) = \Phi_{вх}(j\omega) \cdot K(j\omega)$$

В момента to изходната реализация $Sr_{изх}(t)$ ще има вида $Sr_{изх}(to)$.

Като се използва Обратното преобразование на Фурие, за реализацията в момента to се получава:

$$Sr_{изх}(to) = \int_{-\infty}^{+\infty} \Phi_{изх}(j\omega) \cdot e^{i\omega to} d\omega \quad (20)$$

Замества се $\Phi_{изх}(j\omega)$ с неговото равно и резултатът е:

$$Sr_{изх}(to) = \int_{-\infty}^{+\infty} \Phi_{вх}(j\omega) \cdot K(j\omega) \cdot e^{i\omega to} d\omega$$

Отношението на изходния сигнал в момента to към средноквадратичното отклонение на шума е:

$$\frac{|Sr_{изх}(to)|}{\sqrt{Pn}} = \frac{\int_{-\infty}^{+\infty} \Phi_{вх}(j\omega) \cdot K(j\omega) \cdot e^{i\omega to} d\omega}{\sqrt{\frac{N_0}{2} \int_{-\infty}^{+\infty} |K(j\omega)|^2 d\omega}}$$

Използва се неравенството на Шварц-Бунейковски [2], [7], (21), (22):

$$\int_a^b A(x) \cdot B(x) dx \leq \sqrt{\int_a^b \{ |A(x)|^2 \cdot |B(x)|^2 \} dx} \quad (21)$$

$$\int_a^b A(x) \cdot B(x) dx \leq \sqrt{\int_a^b |A(x)|^2 dx \int_a^b |B(x)|^2 dx} \quad (22)$$

Неравенство се изпълнява винаги при произволни функции $A(x)$ и $B(x)$, а равенство има само когато $B(x) = K \cdot \dot{A}(x)$, където $\dot{A}(x)$ е комплексно спрегнатата функция на $A(x)$.

При полагане:

$$A(x) = \Phi r_{\text{вх}}(j\omega) \cdot e^{i\omega t_0}$$

$$B(x) = K(j\omega)$$

се получава:

$$\frac{|Sr_{\text{изх}}(t_0)|}{\sqrt{Pn}} \leq \frac{\sqrt{\int_{-\infty}^{+\infty} |\Phi r_{\text{вх}}(j\omega) \cdot e^{i\omega t_0}|^2 d\omega \cdot \int_{-\infty}^{+\infty} |K(j\omega)|^2 d\omega}}{\sqrt{\frac{N_0}{2} \int_{-\infty}^{+\infty} |K(j\omega)|^2 d\omega}}$$

$$\frac{|Sr_{\text{изх}}(t_0)|}{\sqrt{Pn}} \leq \sqrt{\frac{2}{N_0} \int_{-\infty}^{+\infty} |\Phi r_{\text{вх}}(j\omega) \cdot e^{i\omega t_0}|^2 d\omega} = \sqrt{\frac{2}{N_0} \int_{-\infty}^{+\infty} |\Phi r_{\text{вх}}(j\omega)|^2 d\omega}$$

$|\Phi r_{\text{вх}}(j\omega)|^2 d\omega$ е енергията на реализацията на сигнала Er .

Крайният резултат, показващ отношението сигнал/шум, е това на енергията на сигнала към спектралната плътност на шума и има вида:

$$\frac{|Sr_{\text{изх}}(t_0)|}{\sqrt{Pn}} \leq \sqrt{\frac{2 \cdot Er_{\text{вх}}}{N_0}} \quad (23)$$

Импулсната характеристика на съгласувания филтър [2], [3], [8] за реализацията на сигнала $Sr(t)$ се извежда с помощта на Обратното преобразование на Фурие [1], [2], [3], [5], [8], [9]:

$$hr(t) = \int_{-\infty}^{+\infty} Kr(j\omega) \cdot e^{i\omega t} d\omega$$

$$Kr(j\omega) = K \cdot \Phi r^*(j\omega) \cdot e^{-i\omega t}$$

За момента на отчета t_0 отношението сигнал/шум е максимално, а после е с тенденция за намаляване.

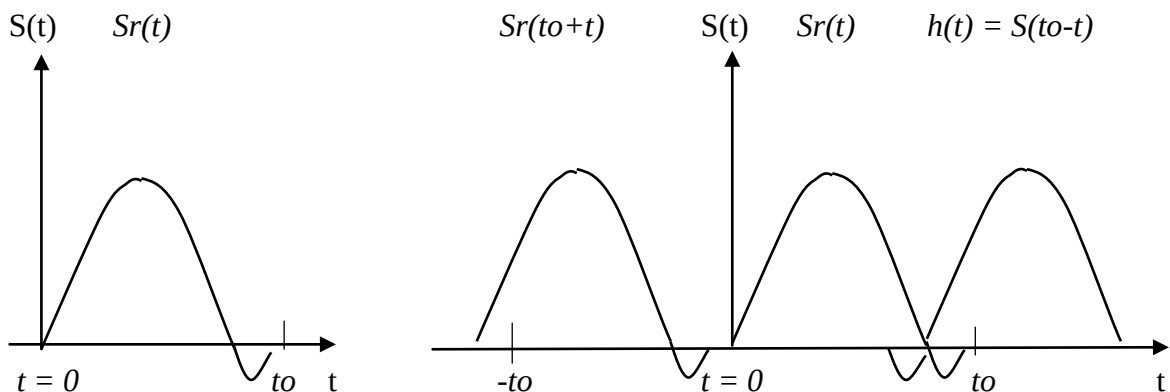
$$hr(t) = K \cdot \int_{-\infty}^{+\infty} \Phi r^*(j\omega) \cdot e^{-i\omega(t_0-t)} d\omega,$$

Като се отчете $\Phi r^*(j\omega) = \Phi r(-j\omega)$, се получава:

$$hr(t) = K \cdot \int_{-\infty}^{+\infty} \Phi r(j\omega) \cdot e^{i\omega(t_0-t)} d\omega$$

$$hr(t) = K \cdot S(t_0-t) \quad (24)$$

Импулсната характеристика на съгласувания филтър [2], [3], [8] е реализацията, която му се подава, отместена във времето на интервал t_0 .



Фиг. 7. Импулсна характеристика и сигнал, изместен по време

Формата на импулсната характеристика е с формата на сигнала, изместен по време t_0 и възниква въпросът за физическата възможност за изпълнение на съгласуван филтър с такава импулсна характеристика [2], [8], [9]. Поставят се някои условия.

$$h(t) = 0 \text{ при } t < 0 \quad (25)$$

Съгласуваният филтър е практически осъществим ако $Sr(t) = 0$ при $t > t_0$, т.е. ако сигналът Sr е завършил в момента t_0 .

3. Заключение

В резултат на представените изследвания се налагат следните изводи:

1) Отношението на сигнала на изхода на линейната верига към мощността на шума се определя от отношението на енергията на сигнала към спектралната плътност на шума. Отношението показва, че резултатът на изхода зависи от това, какво се подава на входа. При произволни функции $A(x)$ и $B(x)$ винаги ще има неравенство, т.е. част от енергията на сигнала се губи.

Равенство ще има само при върховото условие $K(j\omega) = K \cdot \Phi r^*_{vx}(j\omega) \cdot e^{i\omega t_0}$, където $e^{i\omega t_0}$ е коефициентът на закъснение.

При изпълнение на горното условие за предавателната характеристика на изхода на линейната верига ще се установи максимално отношение сигнал/шум.

2) Ако на изхода на съгласувания филтър се цели получаване на максимално отношение на изходния полезен сигнал към шума, то предавателната характеристика $K(j\omega)$ трябва да бъде пропорционална на функцията на комплексната спектрална плътност [5], [7], [8] на входния сигнал.

Пряката връзка с физически смисъл се онагледява с равенството на модулите:

$$|K(j\omega)| = K \cdot |\Phi r^*_{vx}(j\omega)| \quad (26)$$

Модулът на амплитудно-честотната характеристика на съгласувания филтър по форма съвпада с модула на комплексната спектрална плътност на сигнала [5], [7], [8], което означава, че се осигурява пропускане на най-интензивните по мощност участъци от спектъра на полезния сигнал и се потискат участъците, в които съставните са с малка амплитуда и шумът преобладава. Т.е. амплитудно-честотната характеристика е такава, че да пропуска честоти, чиито мощности имат най-голям дял при създаване на енергията на сигнала, а фазовата му характеристика е такава, че съгласуваният филтър допълва началните фази на всички спектрални съставни до стойност $2\pi f_0$ в момент $t = t_0$. В този момент всички изходни съставни имат „нулева“ начална фаза и се сумират по аритметичен закон. Резултатът е един голям „пик“ на изходния сигнал в този момент, но неговата форма обаче значително е изменена в сравнение с тази на входния.

Това може да се докаже и с анализ на изходния сигнал чрез Обратното преобразование на Фурие:

$$\begin{aligned} Sr_{изх}(t) &= \int_{-\infty}^{+\infty} \Phi r_{изх}(j\omega) \cdot e^{i\omega t} d\omega \\ Sr_{изх}(t_0) &= \int_{-\infty}^{+\infty} \Phi r_{vx}(j\omega) \cdot K(j\omega) \cdot e^{i\omega t_0} d\omega = \\ &= K \cdot \int_{-\infty}^{+\infty} [\Phi r_{vx}(j\omega) \cdot \Phi r^*_{vx}(j\omega) \cdot e^{-i\omega t_0}] \cdot e^{-i\omega t} d\omega \\ Sr_{изх}(t_0) &= K \cdot \int_{-\infty}^{+\infty} |\Phi r_{vx}(j\omega)|^2 \cdot e^{-i\omega(t-t_0)} d\omega \end{aligned} \quad (27)$$

3) На изхода на съгласувания филтър няма да има фазови отношения между отделните съставни на сигнала.

4) При съпоставяне на импулсната характеристика на съгласувания филтър $h(t)=0$ при $t<0$ и $hr(t)=K \cdot Sr(t_0 - t)$ следва:

$$hr(t)=K \cdot Sr(t_0 - t) = Sr(t_0 - t) = h(-t) \quad (28)$$

Това означава, че съгласуваният филтър почти винаги не е точно съгласуван (т.е. той е квази-съгласуван).

За да се получи най-голямо отношение сигнал/шум при оптимална филтрация в линейните радиотехнически вериги винаги се използват различията в спектрите на сигналите и шумовете. Затова цялата полезна информация на сигнала се съдържа в спектъра му.

Благодарности

Методологията на разсъжденията, основните акценти и резултатите, представени в статията, са в пряка връзка с академичното ниво на преподаване и висока компетентност от гл. ас. инж. Петър Костов за задълбочено изучаване на дисциплината „Обща теория на свръзките“.

Благодарности към доц. д-р Валерия Симеонова, доц. д-р инж. Пламен Радински и доц. д-р инж. Иван Христов за обективната съпричастност и оценката на използвания математически апарат, практическото значения на теоретичните взаимовръзки в настоящата статия и пригодността им за аргументация при научни изследвания.

Литература

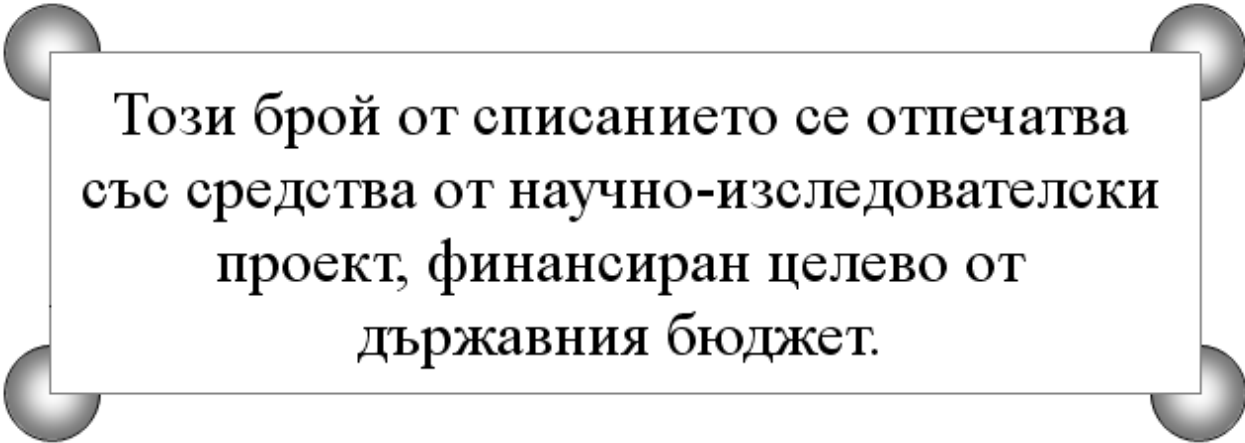
- [1]. Lee E. D., D. G. Messerschmitt. Digital communication. – 2. ed. – Boston etc.: Kluwer Acad. Publishers, 1994.
- [2]. Oppenheim Alan V. , Willsky Alan S. Massachusetts Institute of Technology with Nawab S. Hamid, Signals and systems, second edition, Boston University 1997.
- [3]. Meddins Bob. Introduction to digital signal processing, School of Information Systems University of East Anglia, UK 2000.
- [4]. Leon W. Couch I. I. Digital and Analog Communications Systems - 6 ed. - Prentice Hall, New Jersey, 2001.
- [5]. Halsall Fred, Multimedia Communications: Applications, Networks, Protocols and Standards, Pearson Education, Asia, Second Indian reprint 2002.
- [6]. Dennis Ward Ricker. Echo Signal Processing - Springer, 2003.
- [7]. Васюков В.Н Теория электрической связи учебник / В. Н. Васюков, Новосибирск: Изд-во НГТУ, 2005.
- [8]. Андреев Р.Н., Краснов Р.П., Чепелев М.Ю. Теория электрической связи: курс лекций: учебное пособие для вузов. – Горячая линия-Телеком, 2014.
- [9]. Васюков В.Н. Общая теория связи: учебник / В.Н. Васюков. – Новосибирск: Изд-во НГТУ, 2017.

За контакти:

инж. Георги С. Николов
катедра „Комуникационна техника и технологии“
Технически университет - Варна
e-mail: gnikolov@tu-varna.bg

ИЗИСКВАНИЯ ЗА ОФОРМЯНЕ НА СТАТИИТЕ ЗА СПИСАНИЕ "КОМПЮТЪРНИ НАУКИ И ТЕХНОЛОГИИ"

- I. Статиите се представят разпечатани в два екземпляра (оригинал и копие) в размер до 6 страници, формат А4 на адрес: Технически университет – Варна, ФИТА, ул. „Студентска” 1, 9010 Варна, както и в електронен вид на имейл адреси: ned.nikolov@tu-varna.bg или yulka.petkova@tu-varna.bg.
 - II. Текстът на статията трябва да включва: УВОД (поставяне на задачата), ИЗЛОЖЕНИЕ (изпълнение на задачата), ЗАКЛЮЧЕНИЕ (получени резултати), БЛАГОДАРНОСТИ към сътрудниците, които не са съавтори на ръкописа (ако има такива), ЛИТЕРАТУРА и информация за контакти, включваща: научно звание и степен, име, организация, поделение (катедра), e-mail адрес.
 - III. Всички математически формули трябва да са написани ясно и четливо (препоръчва се използване на Microsoft Equation).
 - IV. Текстът трябва да бъде въведен във файл във формат WinWord 2000/2003 с шрифт Times New Roman. Форматирането трябва да бъде както следва:
 1. Размер на листа - А4, полета: ляво - 20мм, дясно - 20мм, горно - 15мм, долно - 35мм, Header 12.5мм, Footer 12.5мм (1.25см).
 2. Заглавие на български език - размер на шрифта 16, удебелен, главни букви.
 3. Един празен ред - размер на шрифта 14, нормален.
 4. Имена на авторите - име, инициали на презиме, фамилия, без звания и научни степени - размер на шрифта 14, нормален.
 5. Два празни реда - размер на шрифта 14, нормален.
 6. Резюме и ключови думи на български език, до 8 реда - размер на шрифта 11, нормален.
 7. Заглавие на английски език - размер на шрифта 12, удебелен.
 8. Един празен ред - размер на шрифта 11, нормален.
 9. Имена на авторите на английски език - размер на шрифта 11, нормален.
 10. Един празен ред - размер на шрифта 11, нормален.
 11. Резюме и ключови думи на английски език, до 8 реда - размер на шрифта 11, нормален.
 12. Основните раздели на статията (Увод, Изложение, Заключение, Благодарности, Литература) се формират в едноколонен текст както следва:
 - a. Наименование на раздел или на подраздел - размер на шрифта 12, удебелен, центриран, един празен ред преди наименованието и един празен ред след него - размер на шрифта 12, нормален;
 - b. Текст - размер на шрифта 12, нормален, отстъп на първи ред на параграф – 10 мм; разстояние от параграф до съседните (Before и After) за целия текст – 0.
 - c. Цитиране на литературен източник - номер на източника от списъка в квадратни скоби;
 - d. Текстът на формулите се позиционира в средата на реда. Номерация на формулите - дясно подравнена, в кръгли скоби.
 - e. Фигури - центрирани, разположение спрямо текста: "Layout: In line with text". Номер и наименование на фигурата - размер на шрифта 11, нормален, центриран. Отстояние от съседните параграфи – 6 pt.
 - f. Литература – всеки литературен източник се представя с: номер в квадратни скоби и точка, списък на авторите (първият автор започва с фамилия, останалите – с име), заглавие, издателство, град, година на издаване, страници.
 - g. За контакти: научно звание и степен, име, презиме (инициали), фамилия, организация, поделение (катедра), e-mail адрес, с шрифт 11, дясно подравнено.
- Образец за форматиране можете да изтеглите от адрес <http://cs.tu-varna.bg/> - Списание КНТ, Spisanie_Obrazec.zip.



Този брой от списанието се отпечатва
със средства от научно-изследователски
проект, финансиран целево от
държавния бюджет.