НАУЧНО-ИЗДАТЕЛЬСКИЙ ЦЕНТР «АКАДЕМИЯ ЕСТЕСТВОЗНАНИЯ»

SCIENTIFIC PUBLISHING CENTER «ACADEMY OF NATURAL HISTORY»

НАУЧНОЕ ОБОЗРЕНИЕ • ТЕХНИЧЕСКИЕ НАУКИ № 4

SCIENTIFIC REVIEW • TECHNICAL SCIENCES 2021

Журнал «Научное обозрение. Технические науки» зарегистрирован Федеральной службой по надзору в сфере связи, информационных технологий и массовых коммуникаций. Свидетельство ПИ № ФС77-57440
ISSN 2500-0799

Двухлетний импакт-фактор РИНЦ – 0,270
Пятилетний импакт-фактор РИНЦ – 0,242

Учредитель, издательство и редакция: ООО НИЦ «Академия Естествознания», Почтовый адрес: 105037, г. Москва, а/я 47 Адрес редакции и издателя: 410056, Саратовская область, г. Саратов, ул. им. Чапаева В.И., д. 56

Founder, publisher and edition: LLC SPC Academy of Natural History, Post address: 105037, Moscow, p.o. box 47 Editorial and publisher address: 410056, Saratov region, Saratov, V.I. Chapaev Street, 56

Подписано в печать 29.10.2021
Дата выхода номера 30.11.2021
Формат 60×90 1/8

Типография ООО НИЦ «Академия Естествознания», 410035, Саратовская область, г. Саратов, ул. Мамонтовой, д. 5

Signed in print 29.10.2021
Release date 30.11.2021
Format 60×90 8.1

Typography LLC SPC «Academy Of Natural History» 410035, Russia, Saratov region, Saratov, 5 Mamontovoi str.

Технический редактор Байгузова Л.М. Корректор Галенкина Е.С., Дудкина Н.А.

Тираж 1000 экз.
Распространение по свободной цене
Заказ НО 2021/4
© ООО НИЦ «Академия Естествознания»

Журнал «НАУЧНОЕ ОБОЗРЕНИЕ» выходил с 1894 по 1903 год в издательстве П.П. Сойкина. Главным редактором журнала был Михаил Михайлович Филиппов. В журнале публиковались работы Ленина, Плеханова, Циолковского, Менделеева, Бехтерева, Лесгафта и др.

Journal «Scientific Review» published from 1894 to 1903. P.P. Soykin was the publisher. Mikhail Filippov was the Editor in Chief. The journal published works of Lenin, Plekhanov, Tsiolkovsky, Mendeleev, Bekhterev, Lesgaft etc.

М.М. Филиппов (M.M. Philippov)

С 2014 года издание журнала возобновлено Академией Естествознания

From 2014 edition of the journal resumed by Academy of Natural History

Главный редактор: М.Ю. Ледванов
Editor in Chief: M.Yu. Ledvanov

Редакционная коллегия (**Editorial Board**)
А.Н. Курзанов **(A.N. Kurzanov**)
Н.Ю. Стукова **(N.Yu. Stukova)**
М.Н. Бизенкова **(M.N. Bizenkova)**
Н.Е. Старчикова **(N.E. Starchikova)**
Т.В. Шнуровозова **(T.V. Shnurovozova)**

НАУЧНОЕ ОБОЗРЕНИЕ • ТЕХНИЧЕСКИЕ НАУКИ

SCIENTIFIC REVIEW • TECHNICAL SCIENCES

www.science-education.ru

2021 г.

В журнале представлены научные обзоры, статьи проблемного и научно-практического характера

The issue contains scientific reviews, problem and practical scientific articles

СОДЕРЖАНИЕ

Технические науки (05.09.00, 05.11.00, 05.12.00, 05.13.00)

СТАТЬИ

СОЗДАНИЕ ПРОГРАММЫ ПОСТРОЕНИЯ ДИАГРАММ НАПРАВЛЕННОСТИ РУПОРНЫХ АНТЕНН СРЕДСТВАМИ ЯЗЫКА PYTHON

Ильичев В.Ю., Юрик Е.А. *5*

ПРОГРАММНЫЙ СЕРВИС СБОРА ИНФОРМАЦИИ ОБ ЭНЕРГОПОТРЕБЛЕНИИ МОБИЛЬНОГО УСТРОЙСТВА

Гуляева С.А., Гришунов С.С., Белов Ю.С. *10*

ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО КОМПЛЕКСА ОБНАРУЖЕНИЯ И РАСПОЗНАВАНИЯ ДОРОЖНЫХ ЗНАКОВ В ПОТОКОВОМ ВИДЕО

Смольянинов В.А., Белов Ю.С. *16*

ЧИСЛЕННОЕ ИССЛЕДОВАНИЕ ТОЧЕЧНОГО ОТОБРАЖЕНИЯ ДУФФИНГА

Молчанов А.Н., Ильичев В.Ю. *22*

CONTENTS

Technical sciences (05.09.00, 05.11.00, 05.12.00, 05.13.00)

ARTICLES

CREATION OF PROGRAM OF DIRECTIONAL DIAGRAMS OF HORN ANTENNAS BY MEANS OF PYTHON LANGUAGE

Ilichev V.Y., Yurik E.A. *5*

SOFTWARE SERVICE CARRYING OUT THE COLLECTION OF INFORMATION ON THE POWER CONSUMPTION OF THE MOBILE DEVICE

Gulyaeva S.A., Grishunov S.S., Belov Yu.S. *10*

DESIGNING A SOFTWARE COMPLEX FOR TRAFFIC SIGN DETECTION AND RECOGNITION IN THE STREAMING VIDEO

Smolyaninov V.A., Belov Yu.S. *16*

NUMERICAL STUDY OF DUFFING POINT MAPPING

Molchanov A.N., Ilichev V.Yu. *22*

СТАТЬИ

УДК 004.43:621.396.677.73

СОЗДАНИЕ ПРОГРАММЫ ПОСТРОЕНИЯ ДИАГРАММ НАПРАВЛЕННОСТИ РУПОРНЫХ АНТЕНН СРЕДСТВАМИ ЯЗЫКА PYTHON

Ильичев В.Ю., Юрик Е.А.

Калужский филиал ФГОУ ВО «Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)», Калуга, e-mail: patrol8@yandex.ru

Рупорные пирамидальные апертурные антенны обладают рядом достоинств, делающих их незаменимыми при использовании для излучения или приёма волн сверхвысокой частоты (СВЧ). В частности, они имеют совершенную и легко корректируемую диаграмму направленности. Однако для осуществления расчёта размеров элементов антенны и построения диаграмм направленности (в разных плоскостях – горизонтальной и вертикальной, а также объёмной 3D диаграммы) существует необходимость в разработке программных средств на основе современных языков программирования. Все указанные задачи можно осуществить с помощью универсального, свободно распространяемого языка программирования Python при подключении к нему специальных библиотек. В ходе описываемой работы был разработан алгоритм расчёта основных размеров и характеристик антенны и написан программный код для визуализации диаграмм направленности. Для трёхмерного отображения диаграммы создана методика на основе применения команд библиотеки Matplotlib.pyplot для Python. Созданная программа позволяет рассчитывать рупорные пирамидальные антенны для любой частоты сигнала СВЧ. Разработанный при создании программы код визуализации может быть использован для отображения двухмерных и трёхмерных диаграмм направленности любого типа антенн. В заключение сделаны выводы по результатам исследования, а также намечены направления будущих разработок при решении задач проектирования и оптимизации новых типов антенн.

Ключевые слова: апертурная антенна, рупорная антенна, СВЧ, диаграмма направленности антенны, язык Python, модуль Matplotlib

CREATION OF PROGRAM OF DIRECTIONAL DIAGRAMS OF HORN ANTENNAS BY MEANS OF PYTHON LANGUAGE

Ilichev V.Y., Yurik E.A.

Kaluga Branch of Bauman Moscow State Technical University, Kaluga, e-mail: patrol8@yandex.ru

Horn pyramidal aperture antennas have a number of advantages that make them indispensable when used to emit or receive ultra-high frequency (micro-) waves. In particular, they have a perfect and easily corrected directional pattern. However, in order to calculate the size of antenna elements and construct directional-style diagrams (in different planes – horizontal and vertical, as well as volumetric 3D diagram), it is essentially necessary to develop software based on modern programming languages. All these tasks can be accomplished using the universal, freely distributed Python programming language when connecting special libraries to it. In the course of the described work, an algorithm was developed for calculating the main dimensions and characteristics of the antenna and a program code was written for visualizing directional patterns. For three-dimensional display of the chart the technique on the basis of use of commands of Matplotlib.pyplot library for Python is created. The created program allows calculating horn pyramidal antennas for any frequency of the microwave signal. The visualization code developed when creating the program can be used to display two-dimensional and three-dimensional radiation patterns of any type of antenna. Conclusions were drawn from the results of the study, as well as directions for future developments in solving the problems of designing and optimizing new types of antennas.

Keywords: aperture antenna, horn antenna, microwave, antenna pattern, Python language, Matplotlib module

Рупорные антенны являются одним из видов апертурных антенн, обеспечивающих направленность излучения за счёт формирования специально спроектированной формы их раскрытия. В основном они применяются в среднем (сантиметровом) диапазоне волн СВЧ, хотя могут использоваться также в миллиметровом и в дециметровом диапазонах.

Достоинствами данного типа антенн в указанной области применения являются [1]:

– простота, прочность и технологичность конструкции;

– удобство подключения к волноводным линиям;

– совершенная диаграмма направленности с отсутствием заднего лепестка и очень малыми боковыми лепестками;

– возможность достижения высокой мощности излучения;

– малые потери мощности и высокий КПД;

– наличие расчётных формул, обеспечивающих хорошее соответствие теоретических и практических результатов.

Обычно для возбуждения таких антенн используют простейший тип волн («волны основного типа») H_{10}, поле которых похоже на поле в плоском конденсаторе.

Наиболее совершенной формой рупорной антенны с точки зрения достижения

максимума указанных достоинств является рупорная антенна пирамидального типа, рассматриваемая в данном исследовании.

Целью данной работы являлась разработка методов вычисления основных размеров и характеристик рупорной антенны пирамидального типа, а главное – построения диаграмм направленности в полярной системе координат в горизонтальной *H* и вертикальной *E* плоскостях, а также объёмной (3*D*) диаграммы направленности с использованием библиотек языка программирования Python.

Данный метод должен отличаться простотой, удобством использования, наглядностью. Программные коды, написанные на языке Python, в частности для построения полярных и объёмных диаграмм направленности, должны быть применимы также для исследования других типов антенн.

При этом необходимо отработать технологию построения только главного, центрального, лепестка диаграммы направленности, так как задний лепесток у такого типа антенн практически отсутствует, а боковые лепестки очень малы, к тому же существуют исследованные способы для дальнейшего снижения их уровня [2].

Материалы и методы исследования

Алгоритм разработанной программы состоит из следующих блоков:

1. Импорт необходимых модулей языка программирования Python: Numpy для создания массивов и заполнения их данными, необходимыми для визуализации результатов расчёта [3]; Matplotlib. pyplot для создания полярных двухмерных диаграмм и трёхмерной фигуры диаграммы направленности [4].

2. Расчёт основных размеров рупорной антенны для заданной частоты *f, Гц* в порядке, описанном в [5, 6]. Эта методика основана на достижении максимального КНД (коэффициента направленного действия антенны), т.е. на проектировании так называемого оптимального рупора [7].

Данный расчёт состоит в последовательном определении λ – длины волны излучаемого сигнала, ширины и высоты прямоугольного волновода *a* и *b* (рис. 1), горизонтальной и вертикальной сторон раскрытия раструба рупора $a_р$ и $b_р$, длины рупора в горизонтальной плоскости $L_Н$ и в вертикальной плоскости $L_Е$.

3. Определение значения КНД D_0 в направлении максимального излучения антенны, исходя из которого, после построения диаграммы направленности антенны (и тем самым нахождения относительного КНД при любом горизонтальном и вертикальном угле), можно найти КНД в любом направлении.

4. Расчёт диаграммы направленности рупорного излучателя в предположении, что поле в раскрытии рупора является синфазным и амплитудное поле сигнала совпадает с полем сечения питающего волновода. При данном расчёте принят очень небольшой шаг изменения угла в горизонтальной и вертикальной плоскостях (для плавности получаемых графиков); диапазон углов изменяется от 0 до 2π рад.

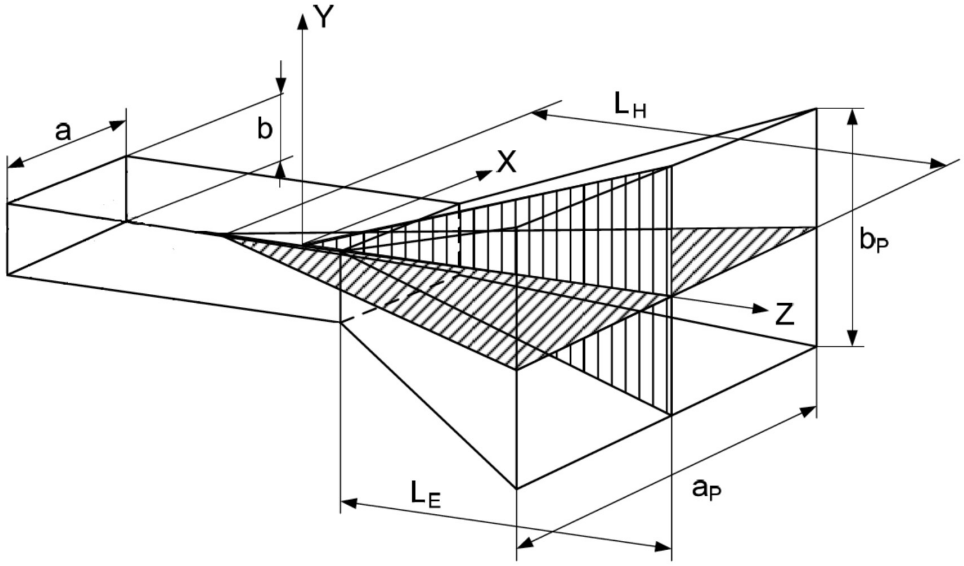


Рис. 1. Эскиз и расчётные характеристики пирамидальной рупорной антенны

Рассчитанные значения относительных КНД при разных углах (радиусы на плоскостных диаграммах направленности в полярных системах координат) заносятся в отдельные массивы Numpy для горизонтальной и вертикальной плоскостей (*H* и *E*).

5. С помощью функций модуля Matplotlib.pyplot изображаются диаграммы направленности антенны в полярных координатах (в зависимости от угла) в плоскостях *H* и *E*.

6. Создаётся сетка в декартовых координатах *X*, *Y*, *Z*, в узлах которой необходимо определить значения относительных КНД путём пересчёта их значений из полярной системы координат с применением тригонометрических функций (которые вычисляются также с использованием модуля Numpy).

7. С помощью функций модуля Matplotlib.pyplot в системе координат *X*, *Y*, *Z* по значениям в узлах сетки создаётся поверхность, изображающая объёмную диаграмму направленности рассчитываемой рупорной антенны.

Результаты исследования и их обсуждение

Для демонстрации результатов работы разработанной программы в качестве примера была рассчитана пирамидальная рупорная антенна конструкции, изображённой на рис. 1, с частотой излучаемой волны $f = 3 \cdot 10^9$ Гц (длина волны $\lambda = 10$ см).

В результате расчёта получились следующие размеры волновода и рупора: $a = 70$ мм, $b = 31{,}5$ мм, $a_р = 95$ мм, $b_р = 63$ мм, $L_H = 30$ мм, $L_E = 20$ мм. Значение КНД в направлении максимального излучения антенны $D_0 = 4{,}81$.

График диаграммы направленности в горизонтальной плоскости *H* изображён на рис. 2.

Как видно из рисунка, антенна обеспечивает хорошую направленность в направлении раскрытия рупора (наибольший относительный КНД, как и следовало ожидать, наблюдается при угле 0 рад, т.е. в направлении оси *Z*).

Аналогично была рассчитана и построена диаграмма направленности в вертикальной плоскости *E* (рис. 3).

В вертикальной плоскости наблюдается также хорошая направленность антенны.

По обеим полученным диаграммам направленности пирамидального рупора можно сделать вывод, что они являются сужающимися, в отличие, например, от диаграмм направленности *H*- или *E*-секториальных рупоров.

На рис. 4 показана объёмная диаграмма направленности в декартовой системе координат *X*, *Y*, *Z* (оси соответствуют рис. 1).

Данный рисунок отличается высокой наглядностью, и на нём отчётливо видно, что наибольшая интенсивность КНД наблюдается по «углам» рупора антенны. Указанную поверхность, полученную средствами модуля Matplotlib.pyplot, можно вращать, подробно рассматривая с разных сторон и увеличивая интересующие исследователя детали.

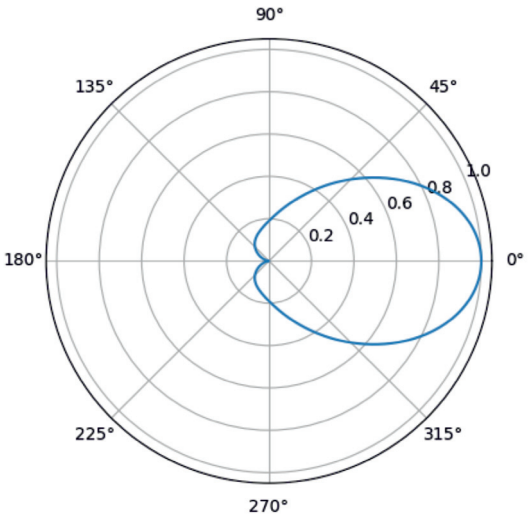


Рис. 2. Диаграмма направленности рассчитанной рупорной антенны в горизонтальной плоскости H

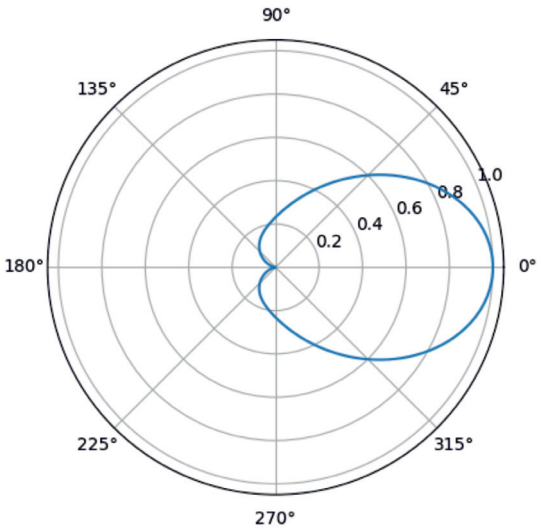


Рис. 3. Диаграмма направленности рассчитанной рупорной антенны в вертикальной плоскости E

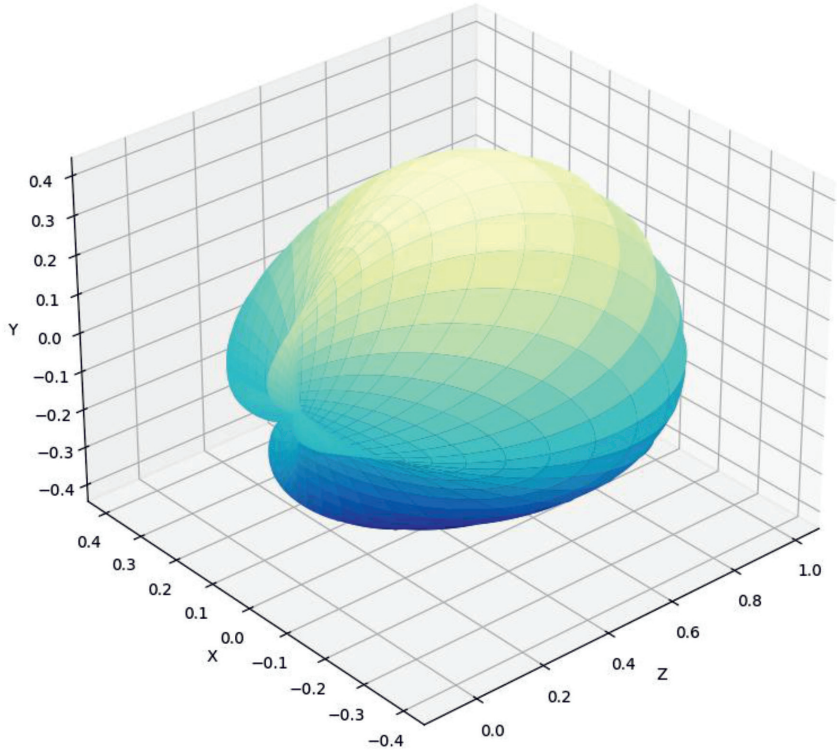


Рис. 4. Объёмная (3D) диаграмма направленности рассчитанной рупорной антенны

Расчёты, подобные вышеприведённому, проводились и для других частот (длин волн) излучаемого антенной сигнала – объёмная диаграмма направленности при этом имеет похожую форму. Её можно корректировать, изменяя размеры волновода, углы раскрытия и длину горизонтальной и вертикальной сторон рупора (все эти параметры поддаются принудительному изменению вручную внутри кода программы). Однако, чем больше изменять размеры по сравнению с рассчитанными программой, тем сильнее будет уменьшаться КНД в направлении максимального излучения антенны.

Заключение

При увеличении длины излучаемых антенной волн размеры волновода и особенно раскрытия рупора могут сильно увеличиваться (например, при длине волны 1 м размеры раскрытия получаются равными $a_р = 0{,}95$ м, а $b_р = 0{,}63$ м, длины сторон рупора $L_Н = 0{,}3$ м, а $L_Е = 0{,}2$ м). Естественно, такие рупоры из-за их громоздкости применять на практике сложно (хотя их массу можно снизить, применяя материалы с низкой плотностью, например, фольгированный текстолит).

Для уменьшения длины рупора, оставляя размеры раскрытия прежними, в нём можно установить так называемую радиолинзу [8] – диэлектрическую вставку, выравнивающей распределение фазы (при этом КНД не снижается). Для расчёта рупора с радиолинзой также подходит представленный выше порядок расчёта и используемые модули и функции языка Python; только формулы, используемые в программе, будут иметь иную форму.

В ходе выполнения работы были выполнены следующие задачи:

– рассмотрены особенности и достоинства пирамидальных рупорных апертурных антенн, работающих в СВЧ-диапазоне;

– сформулирована цель и выбраны средства решения задач, возникающих в ходе проводимой научно-исследовательской работы;

– описан алгоритм программы для расчёта антенны, с рассмотрением особенностей применения функций языка Python;

– приведён пример расчёта размеров и характеристик антенны, произведён краткий анализ полученных с помощью программы двухмерных (в двух плоскостях) и трёхмерных диаграмм направленности;

– даны рекомендации по доработке конструкции рупорной антенны при необходимости снижения её габаритных размеров.

Созданная в результате работы программа позволяет автоматизировать, ускорить и сделать более наглядным процесс проектирования рупорных антенн [9, 10].

Разработанные программные методы рекомендуется применять для получения диаграмм направленности и других типов антенн. Также можно, например, подробно исследовать влияние размеров антенны на форму поверхностей направленности, добиваясь необходимых характеристик.

Проведённое исследование показывает удобство применения выбранного современного языка программирования Python и его библиотек при проектировании и исследовании параметров антенн. Рекомендуется использование этого языка для создания и прочих радиотехнических приложений. С его помощью возможна также обработка данных в реальном времени, автоматическое размещение информации на сайты и многое другое.

Также полученные результаты могут явиться отправной точкой для тех, кто собирается начинать или продолжать исследования в данной области. В частности, авторы планируют в ходе дальнейших работ разработать методику создания конструктивных конфигураций, а также расчёта диаграмм направленности и прочих характеристик фрактальных антенн, с целью их совершенствования, в том числе с проведением экспериментов и их обработкой с помощью модулей языка Python [11, 12].

Список литературы

1. Мительман Ю.Е., Абдуллин Р.Р., Сычугов С.Г., Шабунин С.Н. Антенны и устройства СВЧ: расчет и измерение характеристик. М.: Юрайт, 2017. 138 с.

2. Ставцев Б.С. Способ снижения уровня боковых лепестков диаграммы направленности рупорной сверхширокополосной антенны // Труды Военно-космической академии имени А.Ф. Можайского. 2017. № 657. С. 131–140.

3. Пылов П.А., Протодьяконов А.В. Использование и представление массивов в библиотеке Numpy // Инновации. Наука. Образование. 2020. № 23. С. 258–266.

4. Ильичев В.Ю., Гридчин Н.В. Визуализация масштабируемых 3D-моделей с помощью модуля Matplotlib для Python // Системный администратор. 2020. № 12 (217). С. 86–89.

5. Воскресенский Д.И., Гостюхин В.Л., Максимов В.М., Пономарев Л.И. Устройства СВЧ и антенны. Изд. 2, доп. и перераб. М.: Радиотехника, 2006. 376 с.

6. Заикин И.П., Тоцкий А.В., Абрамов С.К. Проектирование антенных устройств систем связи: учеб. пособие. Харьков: Нац. аэрокосм. ун-т «Харьк. авиац. ин-т», 2007. 78 с.

7. Белоусов А.Г., Ганичев А.А., Драч В.Е., Чухраев И.В. Рупорный облучатель с диаграммой направленности специальной формы // Вопросы радиоэлектроники. 2012. Т. 1. № 3. С. 43–50.

8. Алымов Н.Л., Горшков А.А., Кочетков В.А., Солдатиков И.В., Ханарин И.М., Черкасов А.Е Асимптотические, электродинамические методы и модели проектирования радиолинз как элементов диаграммообразующих схем антенных решеток РЭС СВЧ- И КВЧ-диапазонов // Известия Тульского государственного университета. Технические науки. 2019. № 2. С. 13–29.

9. Драч В.Е., Кузнецова Д.А., Самбуров Н.В., Чухраев И.В. Расширение полосы частот рупорной антенны Н-образного сечения // Электромагнитные волны и электронные системы. 2018. Т. 23. № 1. С. 19–23.

10. Драч В.Е., Луганская А.И. Моделирование влияния преграды на характеристики рупорной антенны // Вестник Рязанского государственного радиотехнического университета. 2018. № 65. С. 19–25.

11. Ильичев В.Ю., Юрик Е.А. Создание программы расчёта упорных подшипников скольжения на языке Python // Научное обозрение. Технические науки. 2020. № 3. С. 14–18.

12. Ильичев В.Ю., Юрик Е.А. Анализ массивов данных с использованием библиотеки Pandas для Python // Научное обозрение. Технические науки. 2020. № 4. С. 41–45.

УДК 004.45

ПРОГРАММНЫЙ СЕРВИС СБОРА ИНФОРМАЦИИ ОБ ЭНЕРГОПОТРЕБЛЕНИИ МОБИЛЬНОГО УСТРОЙСТВА

Гуляева С.А., Гришунов С.С., Белов Ю.С.

Московский государственный технический университет имени Н.Э. Баумана, филиал, Калуга, e-mail: fn1-kf@mail.ru

В наши дни можно утверждать, что телефон перестал быть устройством только для звонков. Он позволяет нам оплачивать покупки, находить правильную дорогу, вызывать такси. Ситуация, в которой у вас садится элемент питания, становится одной из самых стрессовых. Остаться ночью на незнакомой улице без телефона довольно неприятно. При этом расход аккумуляторной батареи растет во многом как следствие расширения возможностей. Чтобы справиться с проблемой разряда батареи, необходимо сначала понять энергетическое поведение приложений и систем. Было разработано приложение, которое направлено на анализ информации о мощности устройства и событиях и на предоставление пользователям подробных данных о поведении энергии. Сервис состоит из серверной части, которая предоставляет информацию о мощности различных приложений, и клиентской стороны, которая анализирует информацию о мощности и событиях для конкретных приложений. В статье оценивается работа данного сервиса, позволяющего собирать данные об энергопотреблении устройства, на планшете и смартфоне. Предоставляются данные о том, как работает сервис для трех групп пользователей: сценарий конечного пользователя, сценарий разработчика приложений, сценарий разработчика системы. Также описаны подходы к тестированию и протестирован разработанный сервис.

Ключевые слова: энергоэффективность устройств, тестирование, анализ, клиент-серверное приложение, подходы к тестированию

SOFTWARE SERVICE CARRYING OUT THE COLLECTION OF INFORMATION ON THE POWER CONSUMPTION OF THE MOBILE DEVICE

Gulyaeva S.A., Grishunov S.S., Belov Yu.S.

Bauman Moscow State Technical University, Kaluga branch, Kaluga, e-mail: fn1-kf@mail.ru

Nowadays it can be argued that the telephone is no longer a device for making calls. It allows us to pay for purchases, find the right road, call a taxi. The situation in which your battery is low becomes one of the most stressful. Staying on an unfamiliar street at night without a phone is rather unpleasant. At the same time, battery consumption is increasing largely as a result of expanding capabilities. To deal with the problem of battery drain, you must first understand the energy behavior of applications and systems. An application has been developed that aims to analyze information about device power and events and to provide users with detailed data on energy behavior. The service consists of a server side, which provides information about the capacity of various applications, and a client side, which analyzes information about capacity and events for specific applications. This paper evaluates the work of this service, which allows you to collect data on the energy consumption of a device on a tablet and a smartphone. It provides data on how the service works for three groups of users: end-user script, application developer script, system developer script. Approaches to testing are also described, and the developed service is tested.

Keywords: energy efficiency of devices, testing, analysis, client-server application, approaches to testing

В настоящее время мобильные устройства, такие как планшеты и смартфоны, стали важной частью нашей повседневной жизни. Нет никаких сомнений, что приложения делают нашу жизнь более удобной и красочной, но не стоит забывать, что они потребляют большое количество энергии на мобильных устройствах и значительно влияют на время автономной работы устройства. Конечному пользователю необходимо знать, какое приложение является более энергозатратным при одном и том же наборе функций. Энергоэффективные приложения более конкурентоспособны на рынке.

Цель исследования: провести тестирование и анализ разработанного приложения.

В ходе работы был разработан сервис, который представляет собой профилировщик и анализатор мощности прикладного уровня. Как показано на рис. 1, сервис возвращает конечным пользователям информацию о мощности соответствующих приложений и дает им дополнительные рекомендации при выборе интересующих их приложений. Для разработчиков приложений, помимо информации о мощности аналогичных приложений, собранной с сервера, клиент также показывает информацию о событиях их разработок, поэтому можно легко найти проблемы с питанием приложений. С точки зрения разработчиков систем, детальная информация о мощности системы, предоставляемая клиентом, полезна им для выбора механизмов энергосбережения [1].

Описание подходов к тестированию. Сервис получает системную информацию от двух классов, PowerProfile и BatteryStats,

и вычисляет энергопотребление приложения на основе моделей мощности. Чтобы проверить точность считывания данных из класса PowerProfile, сначала сравнивается файл ресурсов между различными версиями ОС Android. Кроме того, были написаны собственные тесты, чтобы проверить, согласуются ли результаты с данными, записанными в файле. В основном тестировались яркость, процессор, подключение сокета и файловый ввод / вывод. В экспериментах эталонные приложения запускались на переднем плане, в то время как работа остальных приложений была остановлена для сохранения точности расчетов. Исходя из полученных данных, можно рассчитать мощность, а также информацию о потребляемой энергии. Файл ресурсов содержит данные о мощности различных состояний экрана, Wi-Fi, процессора, bluetooth и так далее. Были запущены тесты, чтобы сравнить собранные данные с информацией, содержащейся в файле. Кроме того, произведен расчет данных ввода-вывода файла, а результаты помещены в систему расчета мощности, которая не поставляется в файле ресурсов.

Мобильные устройства, которые использовались для этих экспериментов – это Huawei Mate 10 Pro и Samsung Galaxy Tab A 8.0. Их аппаратные параметры представлены в табл. 1, включающей информацию о датчиках. Оба этих устройства используют ОС Android 8.0. Чтобы обеспечить репрезентативные результаты, выбранные приложения охватывают большинство категорий. Они имеют более миллиона установок и входят в топ-100, согласно Google Play.

Тестирование сервиса. Были сделаны три тематических исследования, чтобы проиллюстрировать, как различные типы пользователей могут воспользоваться преимуществами сервиса.

Сценарий конечного пользователя. Конечный пользователь обычно хочет, чтобы батарея работала дольше, не слишком часто заряжаясь. Помимо экономии энергии с помощью операционной системы или отключения неиспользуемых устройств, эта цель может быть достигнута путем установки энергосберегающих приложений. Разработанный сервер поддерживает большое количество данных о мощности приложений с информацией об аппаратной платформе.

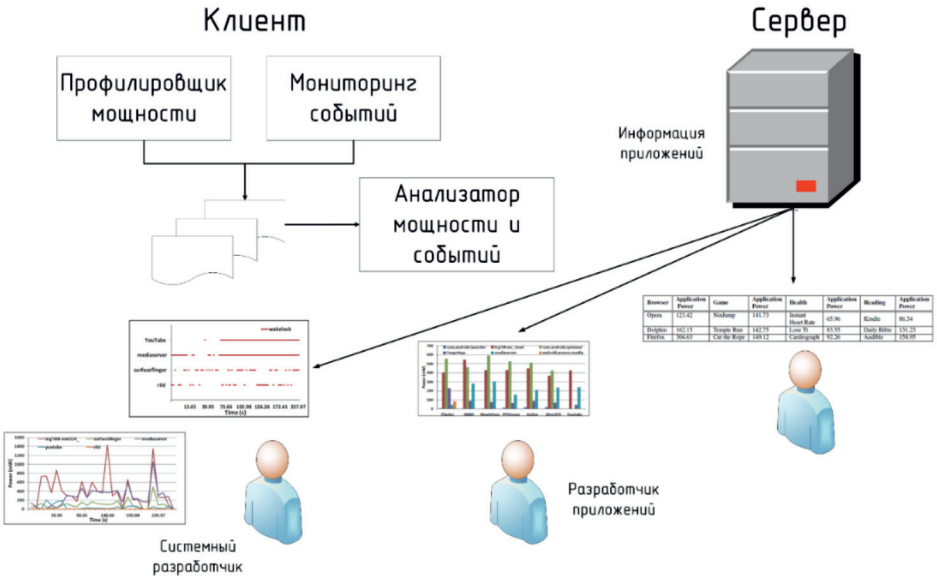


Рис. 1. Схема работы сервиса

Таблица 1

Экспериментальные платформы

Компонент	Huawei Mate 10 Pro	Samsung Galaxy Tab A 8.0
Процессор	8 ядер, 4 x A73 (2,36 ГГц) + 4 x A53 (1,8 ГГц)	Qualcomm Snapdragon 429
Частота (ГГц)	2	2
RAM (Гб)	4	2

Конечные пользователи могут запрашивать эту информацию и искать категорию приложения, которое они хотят установить, возвращаемые данные ранжируются по энергопотреблению приложений. Кроме характеристик приложения, таких как пользовательский интерфейс и специальные функциональные возможности, улучшающие пользовательский опыт, конечные пользователи также могут учитывать энергопотребление [2].

Возьмем в качестве примера браузер, предположим, что пользователи хотят установить Opera на свое устройство. Они могут просто отправить тип устройства и имя приложения Opera на сервер. В табл. 2 перечислены данные нескольких приложений, хранящиеся на сервере.

Отмечено, что данные по мощности рассчитываются в соответствии с общей ситуацией использования. В этом эксперименте были выбраны шесть популярных вебсайтов, включая YouTube.com, Wikipedia.org., Amazon.com, открыли их один за другим и каждый раз прокручивали вниз, чтобы увидеть всю информацию [3]. Можно увидеть, что Opera потребляет меньше энергии, чем Firefox, что делает ее более конкурентоспособной. Чтобы выяснить причины этого, необходимо проанализировать информацию о событиях и необработанные данные журнала питания Opera и Firefox, результаты показывают, что их мощность процессора имеет большую разницу. Firefox может делать больше обработки и вычислений для улучшения пользовательского опыта.

Сценарий разработчика приложений. С одной стороны, сервер предоставляет данные о мощности приложений, которые впоследствии используются разработчиками приложений. С другой стороны, разработчики могут получать информацию о событиях от клиента, что дает направление оптимизации с точки зрения энергопотребления. В этом разделе используется видеоприложение в качестве примера, чтобы показать, как работает сервис [4].

Чтобы учесть влияние, которое приложение может оказать на систему, сервер не только предоставляет информацию о каждом приложении, но и дает другие четыре наиболее энергозатратных процесса каждого приложения и сравнивает их объединение. Таким образом, на рис. 2 сравниваются шесть процессов. Данные собираются из Huawei Mate 10 Pro, и можно увидеть, что системные процессы, поддерживающие приложения, потребляют гораздо больше энергии, чем само приложение. Например, systemui отвечает за рисование пользовательского интерфейса, mediaserver обеспечивает звуковую и другую поддержку мультимедиа. С точки зрения целевого приложения YouTube находится в хорошей ситуации, его мощность ниже, чем у других. На рис. 3 представлена информация о событиях и изменении мощности приложения YouTube как на Huawei Mate 10 Pro, так и на Samsung Galaxy Tab A 8.0. Эта информация помогает разработчикам понять проблему питания. Для информации о событиях ось x – это время; ось y – это процессы, которые генерируют эти события [5]. Записанные события включают в себя wakelock (блокировка сна), датчик, экран и т.д. Была найдена только информация wakelock в этом сценарии, процесс mediaserver появился в обоих устройствах, a rild, surface flinger занимают wakelock на Huawei Mate 10 Pro, a systemui на стороне Samsung Galaxy Tab A 8.0. YouTube также занимает wakelock в течение длительного времени, как показано на рис. 3, а и в, разработчики могут анализировать свой код для улучшения использования wakelock, например, выпускать wakelock в состоянии паузы приложения. Изменение мощности процессов показано на рис. 3, б и д, которые занимают событие или имеют высокое энергопотребление, YouTube потребляет высокую мощность, когда запускается приложение, в то время как mediaserver периодически достигает высшей точки.

Таблица 2

Сравнение мощности различных приложений

Браузер	Мощность приложения (мВт)	Игра	Мощность приложения (мВт)
Opera	123.42	NinJump	141.73
Dolphin	162.15	Temple Run	142.75
Firefox	304.63	Cut the Rope	149.12
Здоровье	Мощность приложения (мВт)	Чтение	Мощность приложения (мВт)
Instant Heart Rate	65.96	Kindle	86.34
Lose Tr	83.55	Daily Bible	131.23
Cardiograph	92.26	Audible	158.95

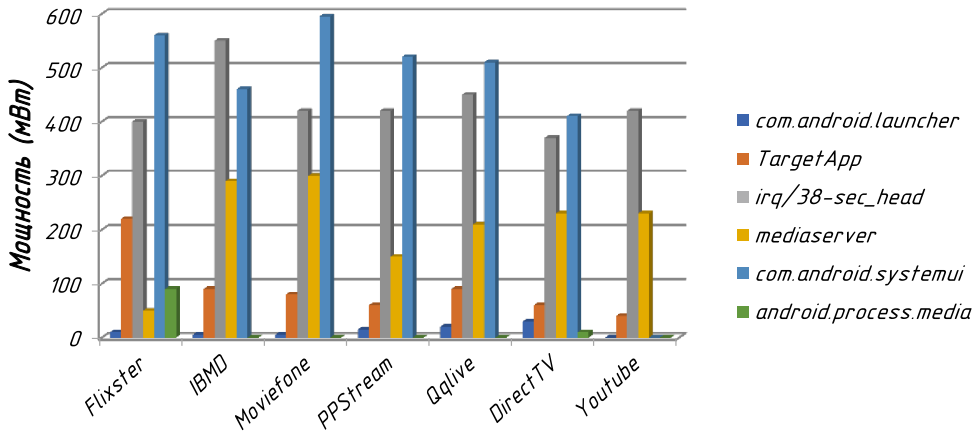
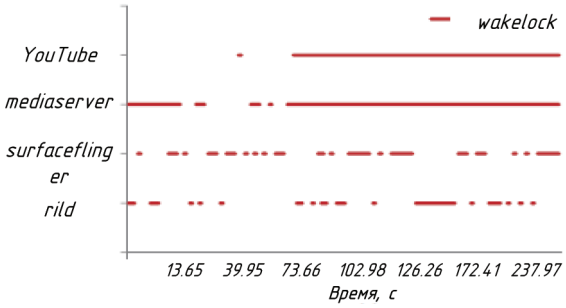
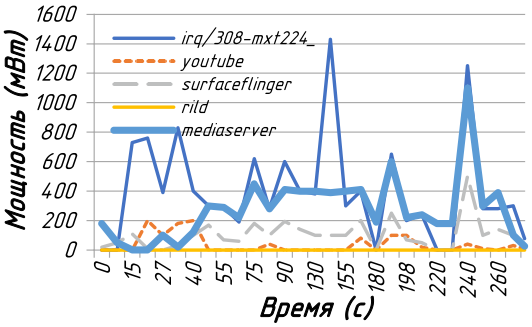


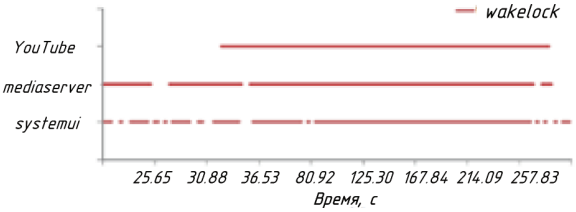
Рис. 2. Сравнение мощности семи видеоприложений



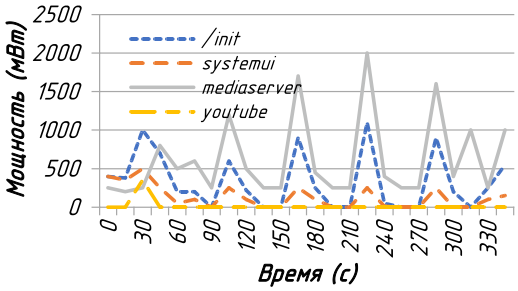
(а) информация о событиях YouTube, запущенного на Huawei Mate 10 Pro



(б) Информация об изменении мощности на Huawei Mate 10 Pro



(в) информация о событиях YouTube, запущенного на Samsung Galaxy Tab A 8.0



(д) Информация об изменении мощности на Samsung Galaxy Tab A 8.0

Рис. 3. Сравнение событий YouTube и информации о мощности

Проанализировав информацию об использовании ресурса в начале YouTube, было обнаружено, что он передавал сетевые пакеты и имел дело с пользовательскими входами (например, touch, click). Никаких других лишних данных не обнаружено. Еще одной причиной столь высокого энергопотребления является система подготовки к началу новой деятельности.

Сценарий разработчика системы. Разработчики систем больше заботятся о потреблении энергии всей системой, а не о конкретном приложении или аппаратном компоненте. Сервис предоставляет информацию о мощности всех процессов, запущенных в системе, что точно помогает знать всю информацию о потребляемой энергии. Раньше было выяснено, что системные процессы потребляют гораздо больше энергии, чем само целевое приложение. Чтобы показать, что это обычная проблема, был проведен еще один эксперимент с игровыми приложениями [6].

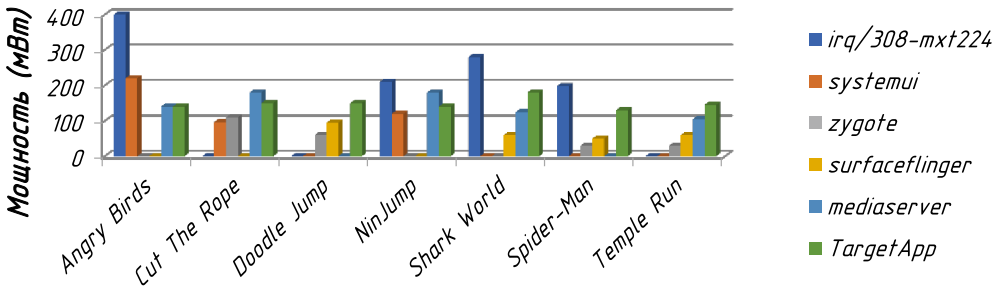
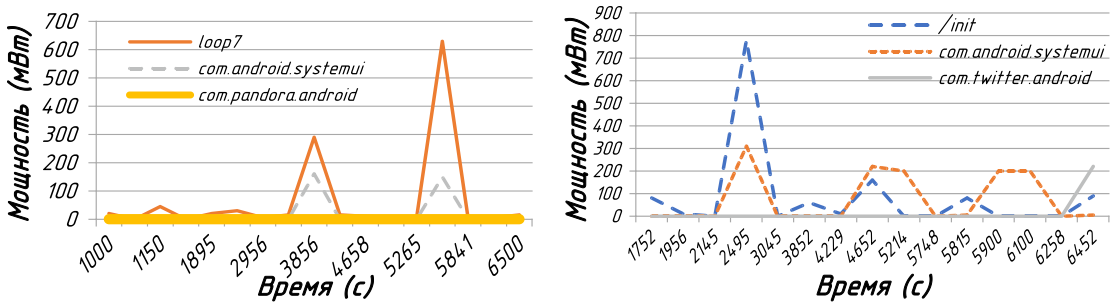


Рис. 4. Сравнение мощности семи игровых приложений



(а) Изменение мощности на Huawei Mate 10 Pro *(б) Изменение мощности на Samsung Galaxy Tab A 8.0*

Рис. 5. Сравнение информации о питании устройств в режиме «сна» с фоновыми приложениями

Были проанализированы 7 популярных игр: Angry Birds, Cut the Rope, NinJump и так далее. Энергопотребление каждой игры и нескольких соответствующих системных процессов на Huawei Mate 10 Pro демонстрируется на рис. 4. Можно наблюдать, что системные процессы, такие как irq/308-mxt224, mediaserver и zygote, потребляли много энергии, следовательно, они не были незначительными по сравнению с целевыми приложениями.

Кроме активных приложений, фоновые приложения также являются основной проблемой для разработчиков систем. В экспериментах сначала проверялась ситуация, когда существуют только системные процессы и не открываются никакие другие приложения. Чтобы представить реальный случай, когда пользователи используют эти устройства, были проведены эксперименты с приложениями, работающими в фоновом режиме. Наиболее распространенная ситуация, которую он представляет, это переход пользователей в спящий режим, их мобильные устройства находятся в спящем состоянии, не выходя из всех открытых приложений [7]. На рис. 5 показано изменение мощности трех основных энергоемких процессов Huawei Mate 10 Pro и Samsung Galaxy Tab A 8.0 в режиме «сна» с незамкнутыми приложениями. Прежде чем перевести устройства в режим «сна», были открыты facebook, twitter, youtube, angrybirds и pandora и каждое использовалось в течение нескольких минут. Из рисунка можно узнать, что мощность большинства приложений невелика, в то время как системные процессы все еще потребляют достаточное количество энергии. Поэтому разработчики систем должны уделять больше внимания оптимизации этих фоновых процессов и служб.

Заключение

Сервис состоит из серверной части, которая предоставляет информацию о мощности различных приложений, и клиентской стороны, которая анализирует информацию о мощности и событиях для конкретных приложений. Были показаны примеры поиска первопричин большого энергопотребления, например чрезмерное использование wakelock [8]. Анализ информации о мощности приложений полезен для многих исследований, связанных с энергией/мощностью на мобильных устройствах, и выводы, полученные из наблюдений, указывают на несколько потенциальных направлений оптимизации.

Список литературы

1. Гуляева С.А., Белов Ю.С. Подходы к оптимизации энергопотребления программно-аппаратных компонентов смартфона // E-Scio: Электронное периодическое издание «E-Scio.ru». 2020. № 7. С. 520–529.

2. Гуляева С.А., Ефимов П.В., Белов Ю.С. Влияние уровней архитектуры мобильных устройств на общее потребление энергии // E-Scio: Электронное периодическое издание «E-Scio.ru». 2019. № 12. С. 669–674.

3. Park S., Kim D., Cha H. Reducing energy consumption of alarminduced wake-ups on android smartphones. Proceedings of the 16th International Workshop on Mobile Computing Systems and Applications. 2015. P. 33–38.

4. Benkhelifa E., Welsh T., Tawalbeh L., Jararweh Y. User Profiling for Energy Optimisation in Mobile Cloud Computing. Procedia Computer Science. 2015. No. 52. P. 1159–1165.

5. Tawalbeh L., Basalamah A., Mehmood R., Tawalbeh H. Greener and Smarter Phones for Future Cities: Characterizing the Impact of GPS Signal Strength on Power Consumption. IEEE Access. 2016. No. 4. P. 858–868.

6. Qian H., Andresen D. Extending mobile device's battery life by offloading computation to cloud. Proceedings of the Second ACM International Conference on Mobile Software Engineering and Systems. 2015. P. 150–151.

7. Bolla R., Khan R., Parra X., Repetto M. Improving smartphones battery life by reducing energy waste of background applications. Next Generation Mobile Apps, Services and Technologies (NGMAST), Eighth International Conference. 2014. P. 123–130.

8. Sewook Park, Dongwon Kim, Hojung Cha. Reducing energy consumption of alarm- induced wake-ups on android smartphones. Proceedings of the 16th International Workshop on Mobile Computing Systems and Applications. 2015. P. 33–38.

УДК 004.942

ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО КОМПЛЕКСА ОБНАРУЖЕНИЯ И РАСПОЗНАВАНИЯ ДОРОЖНЫХ ЗНАКОВ В ПОТОКОВОМ ВИДЕО

Смольянинов В.А., Белов Ю.С.

Московский государственный технический университет имени Н.Э. Баумана, филиал, Калуга, e-mail: fn1-kf@mail.ru

В данной статье описывается проектирование системы поиска и распознавания дорожных знаков в потоковом видео на основе библиотеки OpenCV и сверточной нейронной сети. Представленная модель использует последовательность видеокадров в качестве входных данных. Сначала выполняется перевод видеокадра в цветовое пространство HSV. Затем к HSV-изображению применяется фильтр размытия с ядром 5x5. Получившееся размытое изображение бинаризуется по порогу. Морфологические операции, проводимые над двоичным изображением, удаляют шумы. Следующим этапом проводится контурный анализ, в результате которого отбирается самый большой по площади. Затем из исходного изображения извлекается область с потенциальным знаком. На вход классификатору подаются видеокадры, полученные с помощью алгоритма поиска дорожных знаков. В результате классификации пользователь получает конечную информацию о детектированном дорожном знаке. Основное внимание уделяется тестированию производительности модели. Экспериментально было оценено качество работы алгоритмов поиска и распознавания дорожных знаков на примере видеопотока реального времени, сгенерированного при нормальном дневном освещении. Эксперимент показал, что точность детекции составляет 87,42%, точность распознавания – 97,64%. Общий процент работы системы составляет 80,52%.

Ключевые слова: поиск дорожных знаков, распознавание дорожных знаков, OpenCV, сверточная нейронная сеть

DESIGNING A SOFTWARE COMPLEX FOR TRAFFIC SIGN DETECTION AND RECOGNITION IN THE STREAMING VIDEO

Smolyaninov V.A., Belov Yu.S.

Bauman Moscow State Technical University, Kaluga branch, Kaluga, e-mail: fn1-kf@mail.ru

This article describes the design of a system for searching and recognizing traffic signs in streaming video based on the OpenCV library and a convolutional neural network. The presented model uses a sequence of video frames as input. First, the video frame is translated into the HSV color space. Then a 5x5 kernel blur filter is applied to the HSV image. The resulting blurred image is threshold binarized. Morphological operations performed on a binary image remove noise. The next step is contour analysis, as a result of which the largest area is selected. Then the area with a potential sign is extracted from the original image. The input to the classifier is video frames obtained using the traffic sign search algorithm. As a result of the classification, the user receives the final information about the detected road sign. The focus is on testing the performance of the model. The quality of the algorithms for the search and recognition of road signs was experimentally evaluated using the example of a real-time video stream generated under normal daylight illumination. The experiment showed that the detection accuracy is 87.42%, the recognition accuracy is 97.64%. The overall percentage of the system is 80.52%.

Keywords: traffic sign detection, traffic sign recognition, OpenCV, convolutional neural network

В настоящее время системы поиска и распознавания дорожных знаков привлекают к себе большое внимание. Цель таких систем – осведомить водителей о приближающихся знаках на дороге и предупредить их о возможных опасностях. Дорожные знаки включают полезную и важную визуальную информацию, такую как ограничение максимальной скорости, сужение полосы движения, направление движения, расстояние до места назначения, а также опасные или необычные условия, чтобы помочь водителю во время движения. Если водитель не замечает дорожный знак или не понимает содержащуюся в нем информацию, то разумно предположить, что условия движения становятся неоптимальными. Данная система может уменьшить эту возможность и значительно помочь водителю, обнаруживая и распознавая попутные дорожные знаки.

Цель исследования: разработать комплекс обнаружения и распознавания дорожных знаков с помощью методов, минимизирующих выявленные недостатки существующих систем.

Описание модели

Изучив основные аспекты в области распознавания дорожных знаков, можно сделать вывод о том, что целесообразно объединять несколько алгоритмов для решения задачи распознавания [1]. Например, объединение алгоритмов цветовой сегментации с основным функционалом библиотеки компьютерного зрения OpenCV позволит максимизировать процент корректного распознавания и свести к минимуму недо-

статки алгоритмов. Работа системы обнаружения и распознавания дорожных знаков происходит в два этапа: детектирование дорожных знаков; распознавание дорожных знаков. Общая модель системы распознавания дорожных знаков состоит из ряда взаимосвязанных модулей.

Веб-камера. В этом исследовании для сбора входных данных использовалась веб-камера Defender G-lens 2597 (рис. 1). Благодаря CMOS сенсору с разрешением 2 мегапикселя камера выдает детализированное широкоформатное изображение. Автоматическая подстройка светочувствительности позволяет получать высококачественное изображение даже при недостаточном освещении. Нет необходимости в постоянной настройке фокуса при перемещении в кадре: камера делает это автоматически.

Цветовая сегментация. Сегментация цвета является важным шагом, на котором с изображения (видеокадра) устраняются все фоновые объекты и неважная информация, и генерируется двоичное изображение, содержащее дорожные знаки и любые другие схожие объекты [2]. Результат цветовой сегментации показан на рис. 1.

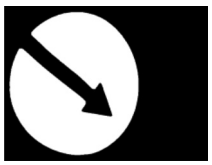


Рис. 1. Результат цветовой сегментации

Детектирование дорожного знака реализуется посредством контурного анализа. На предобработанном двоичном изображении выполняется поиск контуров потенциального дорожного знака с помощью функции cv2.findContours(). Затем найденные контуры отображаются на изображении (cv2.drawContours()). Далее происходит поиск наибольшего по площади контура и поиск области, в которой он располагается (cv2.boundingRect()) [3]. Завершается этап детекции вырезанием области с потенциальным дорожным знаком (рис. 2).

Рис. 2. Детектированный дорожный знак

Набор обучающих данных. Реализация эксперимента проходила с использованием набора данных, сгенерированного на основе российской базы дорожных знаков (RTSD). Для генерации базы данных RTSD был привлечен персонал из организации «Геоцентр-Консалтинг» [4]. Набор обучаемых данных для предлагаемой модели содержит 42689 изображений с 53 различными классами. Изображения неравномерно распределены между этими классами, поэтому модель может предсказывать одни классы с большей точностью, чем другие. Исходя из этого, рекомендуется сделать набор обучающих данных более согласованным, увеличив набор данных с помощью изменения существующих изображений. Такой подход называется искусственным увеличением данных. Изображения, содержащиеся в исходном наборе данных RTSD-3, имеют динамический диапазон измерений: от 16×16×3 до 128×128×3, поэтому его нельзя передать непосредственно в модель. Используя библиотеку компьютерного зрения OpenCV, была установлена размерность (40×40) всех изображений в наборе данных с помощью метода cv2.resize (img, (40, 40)) (рис. 3).

Рис. 3. Фрагмент набора изображений 40×40 для каждого из 53 классов

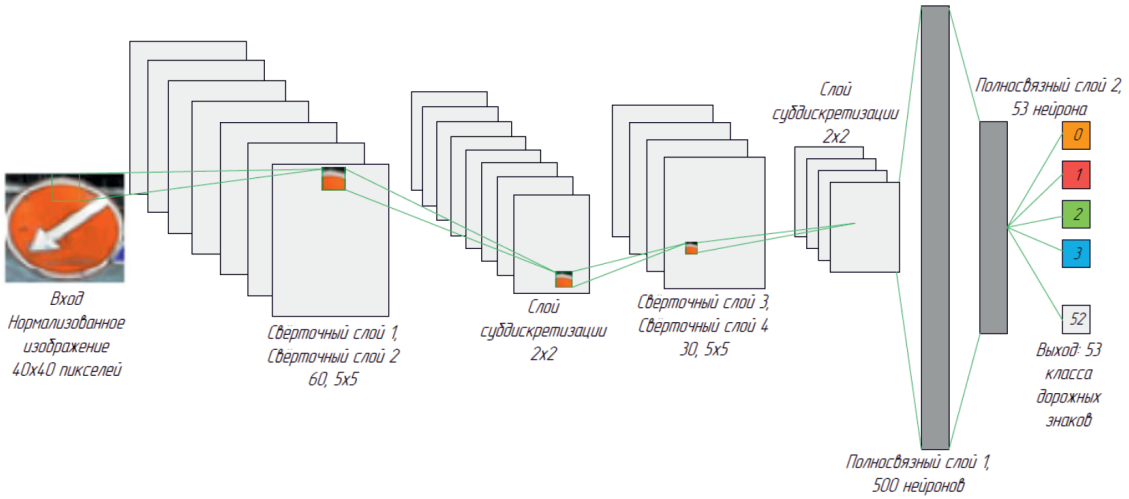


Рис. 4. Архитектура сверточной нейронной сети

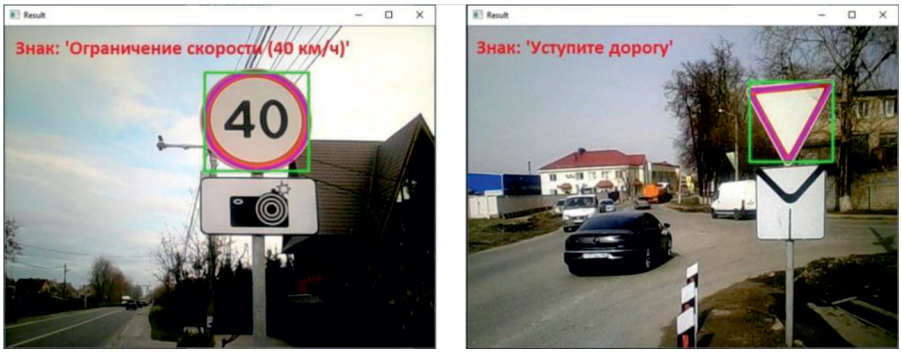


Рис. 5. Пример работы системы распознавания дорожных знаков

Сверточная нейронная сеть. Сверточная нейронная сеть, изображенная на рис. 4, спроектирована на основе модели распознавания рукописных цифр [5] и состоит из четырех сверточных слоев и двух полносвязных. Первые два слоя свертки содержат 60 фильтров, каждый из которых имеет размерность 5×5. Последние два слоя содержат 30 фильтров с размером ядра 3×3. Функция выпрямленных линейных единиц (ReLu) используется в качестве функции активации для всех сверточных слоев. После каждых двух слоев свертки используются слои Maxpooling, размер пула которых составляет 2×2. Чтобы уменьшить вероятность переобучения, после второго слоя maxpooling и первого слоя Dense добавляется 50% ный Dropout (отсев). Первый полносвязный слой содержит 500 нейронов, второй – 53 нейрона, что равняется количеству классов дорожных знаков. Последняя функция активации – это функция softmax для классификации.

Классификация. Этап классификации (распознавания) выполняется с помощью сверточной нейронной сети. На вход классификатору подаются как тестовая выборка изображений дорожных знаков, так и изображения, полученные с помощью алгоритма поиска дорожных знаков. В результате классификации пользователь получает конечную информацию о детектированном дорожном знаке (рис. 5).

Анализ полученных результатов. За 30 эпох обучения нейронной сети точность на обучающих данных составила 0,9623, ошибка на обучающих данных уменьшилась до 0,1304; точность на проверочных данных достигла результата 0, 9918, а ошибка на проверочных данных снизилась до 0,0319. С помощью библиотеки Matplotlib, для наглядности, были построены графики точности и ошибки обучения (рис. 6).

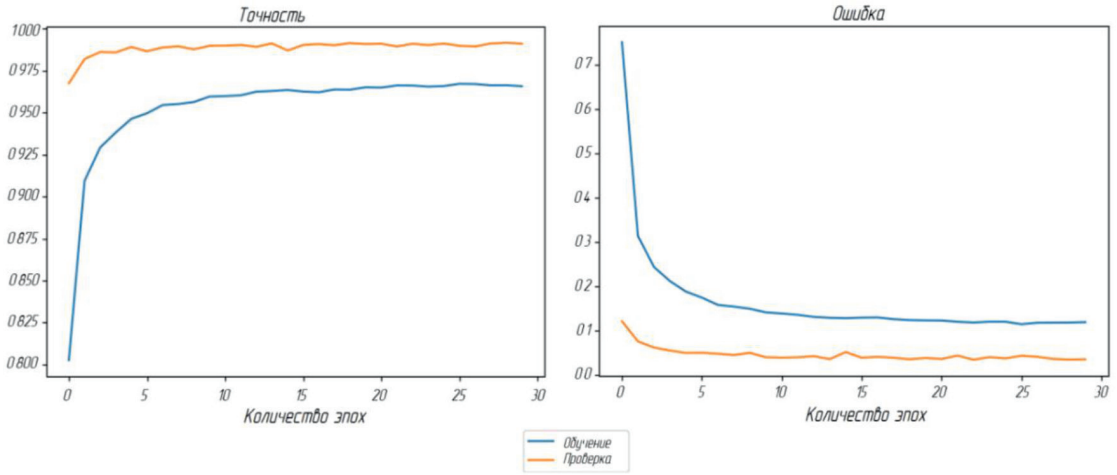


Рис. 6. Показатели точности и ошибки на протяжении 30 эпох обучения

Таблица 1

Результаты работы алгоритма обнаружения в потоковом видео

Форма контура знака	Общее количество знаков на видео	Количество корректно обнаруженных знаков	Верное обнаружение (%)
Треугольник	276	220	79,71
Четырехугольник	180	148	82,22
Окружность	160	140	87,5
Всего	616	508	82,47

Реализация модели обеспечивает точность в 99,22 %, тем самым превышая результат из исследования [6], в котором сверточная нейронная сеть, обученная на основе базы немецких дорожных знаков (GTSRB), достигает точности 95 %.

Тестирование модели. Вследствие своих недостатков предлагаемый алгоритм поиска дорожных знаков неустойчив к некоторому роду шумов: частичное перекрытие знаков, низкое качество видео ввиду малого разрешения камеры, а также плохой ее стабилизации и недостаточной цветопередачи. В режиме реального времени алгоритм корректно детектирует 82,47 % дорожных знаков.

Согласно табл. 2 средний показатель качества распознавания знаков на изображениях, полученных с видеопотока, составляет 97,64 %, что является доверительным показателем. Результаты распознавания знаков ограничения скорости были объединены ввиду отсутствия всех четырех наименований этого класса в видеопотоке.

Стоит обратить внимание на то, что процент классификации изображений, извлеченных из видеопотока, немного ниже, чем процент классификации изображений, сгенерированных при тестировании. Это расхождение может быть оправдано тем, что изображения, содержащиеся в обучающем наборе, имеют то же происхождение, что и тестовые синтезированные изображения, в то время как реальные изображения, в свою очередь, имеют другое отношение к изображениям из обучающего набора данных [7]. Согласно результатам оценки качества работы разработанной системы поиска и распознавания объектов дорожной инфраструктуры, общий процент работы предлагаемой модели составил 80,52 %. Исходя из проведенного эксперимента, можно сделать вывод о том, что детектор является слабым местом в модели и нуждается в доработке.

В табл. 3 показано время выполнения обнаружения и распознавания. Процессорная обработка алгоритмов модели занимает 1299,46 мс для всех вариантов разрешения изображений.

Стоит отметить, что разрешение входного кадра положительно сказывается на времени работы алгоритмов: чем ниже разрешение изображения, тем быстрее модель его обрабатывает. Однако с уменьшением разрешения входного видеокадра снижается качество работы модели.

Таблица 2

Результат работы алгоритма обнаружения дорожных знаков в видеопотоке

Порядковый номер класса	Наименование класса	Распознавание (%)
27–31	Знаки ограничения скорости	96,96
22	Въезд запрещен	98,09
26	Обгон запрещен	97,47
36	Движение прямо	97,83
38	Движение налево	98,1
37	Движение направо	97,06
39	Движение прямо или направо	98,12
40	Движение прямо или налево	97,82
41	Объезд препятствия справа	97,99
42	Объезд препятствия слева	97,07
47	Пешеходный переход	97,94
8	Уступить дорогу	97,64
14	Дети	97,63
19	Примыкание второстепенной дороги справа	97,29

Таблица 3

Производительность предлагаемой модели

Разрешение кадра	Обнаружение (мс)	Распознавание (мс)	Всего (мс)	Точность детекции (%)	Количество кадров / с
1920 x 1080	352,26	135,45	487,71	82,47	2,05
1632 x 918	252,03	95,69	347,72	82,12	2,88
1344 x 756	167,99	64,97	232,96	80,71	4,29
1056 x 594	97,33	40,16	137,49	77,73	7,27
768 x 432	49,02	20,92	69,94	70,7	14,30
480 x 270	15,47	8,17	23,64	42,7	42,30
Всего	934,1	365,36	1299,46		

Заключение

Для решения поставленной задачи поиска и распознавания объектов дорожной инфраструктуры целесообразно объединить несколько алгоритмов. Например, комбинирование алгоритмов цветовой сегментации с анализом контуров позволит повысить точность распознавания и минимизировать недостатки каждого из алгоритмов.

Была предложена модель системы поиска и распознавания дорожных знаков в потоковом видео на основе библиотеки OpenCV и сверточной нейронной сети. Детектор основан на алгоритмах компьютерного зрения. На вход классификатору подаются видеокадры, полученные с помощью алгоритма поиска дорожных знаков. В результате классификации пользователь получает конечную информацию о детектированном дорожном знаке.

В ходе экспериментов был продемонстрирован весь функционал, определенный на шаге конструирования модели. Эксперименты показали, что детектор дорожных знаков, основанный на алгоритмах компьютерного зрения, неустойчив к некоторому роду шумов: частичное перекрытие знаков, низкое качество видео ввиду малого разрешения камеры, а также плохой ее стабилизации и недостаточной цветопередачи.

Таким образом, можно сделать вывод о том, что детектор недостаточно эффективен для решения поставленной задачи в реальном времени. В дальнейшем планируется улучшить алгоритмы детектора, что позволит минимизировать количество ложных срабатываний.

Список литературы

1. Смольянинов В.А., Белов Ю.С. Сравнительный анализ алгоритмов поиска дорожных знаков // Высокие технологии и инновации в науке: сборник избранных статей Международной научной конференции. 2020. С. 186–190.

2. Смольянинов В.А., Гришунов С.С., Белов Ю.С. Подходы к проектированию и разработке программного детектора дорожных знаков // Сборник избранных статей по материалам научных конференций ГНИИ «Нацразвитие»: материалы Международных научных конференций. 2020. С. 145–149.

3. Bradski G., Kaehler A. Learning OpenCV: Computer Vision with the OpenCV Library. Sebastopol: O'Reilly Media, 2008. 571 p. [Electronic resource]. URL: http://bookre.org/reader?file=1077338 (date of access: 23.04.2021).

4. Шахуро В., Конушин А. Российская база изображений автодорожных знаков // Компьютерная оптика. 2016. № 2. С. 294–300.

5. Shawon A., Mahmud F., Md. Jamil-Ur Rahman, Arefin Zaman M.M. Bangla Handwritten Digit Recognition Using Deep CNN for Large and Unbiased Dataset. International Conference on Bangla Speech and Language Processing (ICBSLP). IEEE, 2018.

6. Class Data Science Project 2020 Traffic Signs Recognition [Electronic resource]. URL: https://medium.com/dataflair/class-data-science-project-for-2020-traffic-signs-recognition-12b09c131742 (date of access: 23.04.2021).

7. Yasmina D., Karima R., Ouahiba A. Traffic signs recognition with deep learning. 2018 International Conference on Applied Smart Systems (ICASS), Medea, Algeria, 2018. P. 1–5. DOI: 10.1109/ICASS.2018.8652024.

УДК 004:51

ЧИСЛЕННОЕ ИССЛЕДОВАНИЕ ТОЧЕЧНОГО ОТОБРАЖЕНИЯ ДУФФИНГА

Молчанов А.Н., Ильичев В.Ю.

Калужский филиал ФГОУ ВО «Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)», Калуга, e-mail: patrol8@yandex.ru

Одним из современных численных методов исследования нелинейных динамических систем является использование так называемых точечных отображений. Он основывается на описании дискретных состояний системы с помощью простых алгебраических уравнений, поэтому является удобным для алгоритмической реализации. Язык программирования Python обладает широким набором средств, позволяющих исследовать эволюционное изменение разных параметров точечных отображений, а также выявлять важные для науки закономерности. Это особенно актуально в настоящее время, когда теория точечных отображений позволяет смоделировать всё большее количество реальных физических процессов, даже таких, для описания которых не существует точных математических приёмов. В данной работе описывается часто применяемое точечное отображение Дуффинга, а также алгоритмические приёмы и модули языка Python, использованные для разработки исследовательских программ. Созданы две программы, позволяющие решить следующие задачи: для любых начальных параметров системы Дуффинга и набора используемых аргументов уравнений строить двухмерное фазовое отображение; откладывать положение формируемых точек системы по осям координат в зависимости от номера итерации; строить кривые распределения вероятности значений координат. Также решена задача получения комплексной картины, позволяющей наглядно отображать области устойчивого и хаотического поведения системы Дуффинга (карты динамических режимов). Коды написанных программ размещены для свободного распространения и модификации на сайте авторов. Это позволяет всем желающим изучить основные принципы, лежащие в основе теории точечных отображений, и приёмы их реализации на Python.

Ключевые слова: итерация, точечное отображение, отображение Дуффинга, динамические нелинейные системы, язык Python, карта динамических режимов

NUMERICAL STUDY OF DUFFING POINT MAPPING

Molchanov A.N., Ilichev V.Yu.

Kaluga branch of Bauman Moscow State Technical University, Kaluga, e-mail: patrol8@yandex.ru

One modern numerical method of studying nonlinear dynamic systems is the use of point mappings. It is based on the description of discrete states of system using simple algebraic equations, therefore, it is convenient for algorithmic implementation. Python programming language has a wide range of tools that allow you to investigate the evolutionary change of different parameters of point mappings, as well as identify patterns important to science. This is especially true at present, when the theory of point mappings allows you to model an increasing number of real physical processes, even those that do not have exact mathematical techniques to describe. This work describes the often used Duffing mapping, as well as algorithmic techniques and Python language modules used to develop research programs. Two programs have been created to solve the following problems: for any initial parameters of Duffing system and the set of arguments of the equations, build a two-dimensional phase mapping; postpone the position of the generated points of the system along the coordinate axes depending on the iteration number; plot the probability distribution curves of coordinate values. The problem of obtaining a complex picture that allows you to clearly display areas of stable and chaotic behavior of the Duffing system (dynamic mode maps) was also solved. Codes of written programs are posted for free distribution, study and modification on the authors' website. This allows everyone to study basic principles underlying the theory of point mappings and techniques for implementing them on Python.

Keywords: iteration, point mapping, Duffing mapping, dynamic non-linear systems, Python language, dynamic mode map

Большая область современных исследований посвящена определению характеристик динамических нелинейных систем [1], так как они наиболее точно описывают многие реальные физические объекты. При этом всё более широко используется метод точечных отображений [2], при котором параметры системы вычисляются через равные промежутки времени. Этот метод позволяет простыми математическими приёмами смоделировать эволюцию сложных систем.

Благодаря рассмотренным особенностям, применение точечных отображений для исследования сложных динамических нелинейных систем является одним из наиболее перспективных приёмов. Он позволяет выявить зоны устойчивого и неустойчивого поведения систем [3], исследовать траектории изменения их параметров, многие вероятностные характеристики, причём даже для систем, описываемых уравнениями, не решаемыми точными математическими методами.

Родоначальником метода точечных отображений является знаменитый учёный А. Пуанкаре, а в трудах Д. Биркгофа дан-

ный метод стал основным инструментом для изучения поведения динамических систем, и с тех пор сфера его использования всё более расширяется.

Целью данной работы является создание программных средств для вычисления характеристик одного из видов точечных отображений – Дуффинга (иногда называемого отображением Холмса). Физически оно описывает процесс движения одномерной (точечной) частицы в потенциале (например, в электромагнитном поле). Оно помогает смоделировать такие разнообразные системы, как микромеханические структуры, наномеханические изоляторы, роторы, призмы в потоках жидкости, траектории движения крыльев насекомых, характеристики систем ультразвуковой резки и многие другие.

Программные коды для исследования систем Дуффинга написаны с использованием одного из наиболее популярных, наглядных и простых в освоении и применении языков Python и создаются для использования, доработки, изучения всеми желающими, поэтому размещаются для свободной загрузки на сайте авторов [4]. Данные разработки дополняют ранее произведённые исследования других типов динамических систем [5].

Материалы и методы исследования

Для многих динамических систем характерна особенность, что при некотором наборе начальных значений факторов и коэффициентов уравнений интерполяции они демонстрируют устойчивость, которая выражается в совпадении точек, получаемых в процессе итераций. Также есть области таких начальных значений факторов и коэффициентов уравнений, при которых система демонстрирует хаотичное поведение (то есть не возвращается в прежнее состояние). Однако обычно этот хаос также некоторым образом упорядочен (поэтому его часто называют детерминированным хаосом), то есть формируемые точки не уходят далеко от некоторых областей притяжения – аттракторов [6].

Рассмотрим теперь, что представляет собой исследуемое точечное отображение Дуффинга [7] и каким образом оно формируется. Данная динамическая система относится к классу двухмерных (то есть имеет два изменяемых фактора x и y), и её точечное отображение поэтому формирует некую фазовую картину (для получения которой необходимо разработать программу) на плоскости. Разрабатываемая программа должна показывать зависимость значений факторов x и y от номера итерации (условного времени, номера очередной формируемой на плоскости точки) и позволять исследовать некоторые вероятностные характеристики распределения параметров. Ставится также задача численного определения областей устойчивости и хаоса точечного распределения Дуффинга и отображения их в графическом виде.

Формулы для вычисления итераций отображения Дуффинга выглядят следующим образом:

$$x_{n+1} = y_n;\ \ y_{n+1} = -bx_n + ay_n - y_n^3,$$

где a, b – постоянные множители, n – номер итерации вычислений.

Видим, что положение каждой последующей точки отображения зависит от предыдущей и т.д. Поэтому необходимо также задать координаты начальной точки.

Рассмотрим функции программы исследования отображения Дуффинга:

1. Импорт библиотек Tkinter и PIL – для формирования интерфейса; Matplotlib [8], Seaborn [9] – для обработки и вывода результатов в графическом виде.

2. Создание пользовательского интерфейса.

3. Создание функции, заполняющей массивы факторов x, y, а также номеров итерации n.

4. Написание функции, осуществляющей вывод в графическом виде фазовой траектории, формируемой точками с вычисляемыми при множестве итераций координатами (x, y).

5. Создание функции, формирующей зависимости координат x, y от номеров итерации n.

6. Вычисление и построение статистических распределений факторов x, y.

Интерфейс созданной программы представлен на рис. 1.

Рисунок показывает вид программы с параметрами по умолчанию. Двухмерная фазовая траектория из точек, получаемая при нажатии на кнопку «Траектория», представлена на рис. 2.

Может показаться, что сформированная картина не является хаосом, так как видны линии, напоминающие участки синусоиды. На самом же деле каждая последующая итерация (точка) расположена достаточно далеко от предыдущей. Однако, как было отмечено выше, чем больше итераций отображается, тем больше заметно тяготение координат точек к определённым областям – аттракторам (при малом числе итераций это не так видно). Следовательно, реализованная функция «Траектория» позволяет исследователю проследить влияние

различных параметров, задаваемых с помощью интерфейса программы, на картину, формируемую точками на плоскости. Если построить отображение для коэффициентов a и b, при которых поведение системы устойчиво, то на графике будут присутствовать лишь одна или несколько дискретных точек, причём для любого количества итераций количество дискретных точек будет одинаково. То есть система всегда возвращается к первоначальному состоянию.

Результаты выполнения следующей функции программы «Перемещения по осям» (реализованной с помощью модуля Seaborn для Python) приведены на рис. 3 (выводятся отдельно графики для x, y, но показан только один из них, так как они получаются одинаковыми).

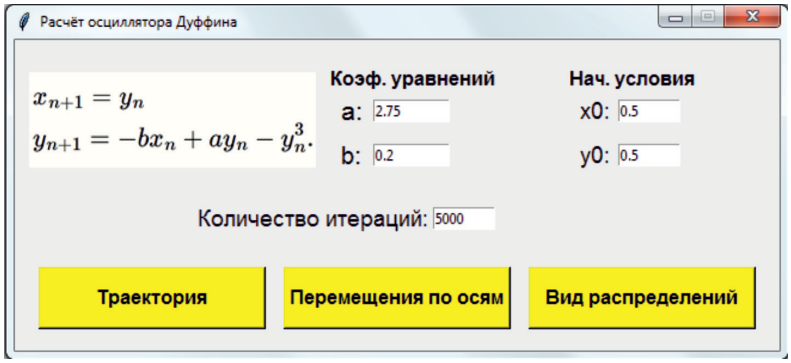


Рис. 1. Интерфейс программы для исследования точечного отображения Дуффинга

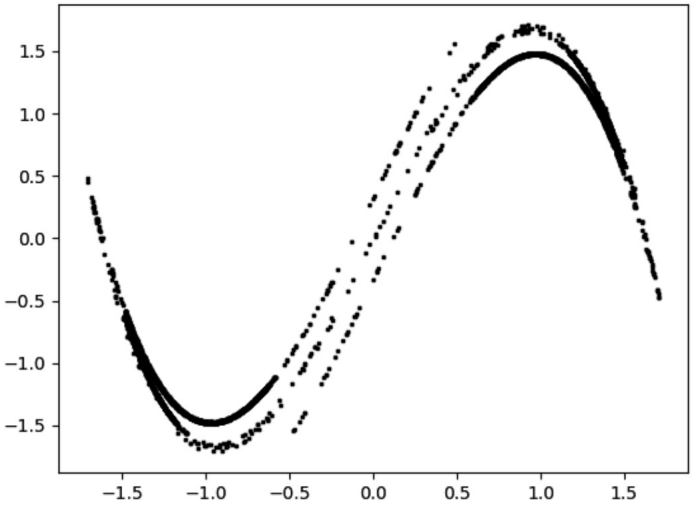


Рис. 2. Картина двухмерной фазовой траектории отображения Дуффинга, формируемая с параметрами программы по умолчанию

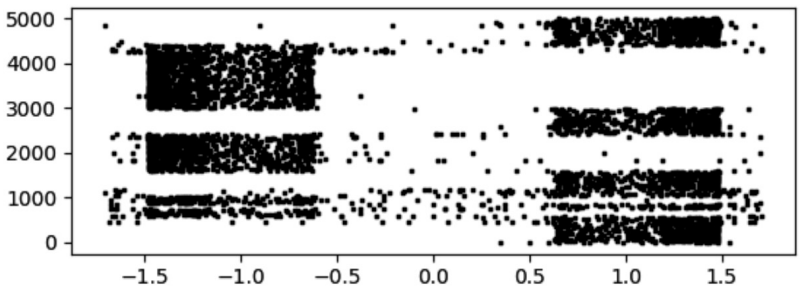


Рис. 3. Результаты вычисления перемещений по осям x, y в зависимости от номера итерации

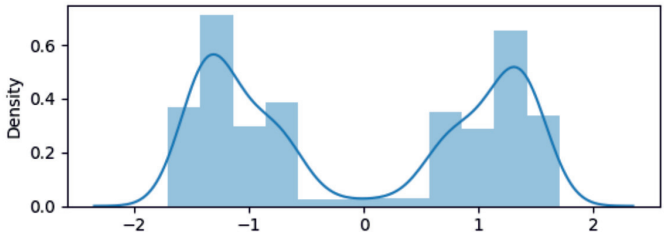


Рис. 4. Распределение значений x, y, формируемых при итерациях

Данный рисунок показывает, как координаты *x*, *y* зависят от номера итерации – виден «перескок» координат из одной области значений в другую. Также проявляется и тяготение координат к определённым областям. Данное распределение точек по диапазонам значений можно показать с помощью третьей функции программы «Вид распределений» (рис. 4).

В данном случае распределение по факторам *x*, *y* является одинаковым (симметричным), а наиболее вероятные значения факторов находятся вблизи областей с модами -1,25 и 1,25. То же самое видно и на рис. 2 и 3, но не столь явно. При других значениях задаваемых в интерфейсе программы, распределение и модальные значения могут быть другими, и программа позволяет их однозначно идентифицировать в каждом рассматриваемом случае.

Затем разрабатывалась программа для определения устойчивости отображения Дуффинга при изменении параметров *a* и *b*. Цвет точек графика показывает количество итераций, через которое картина возвращается к первоначальной. Если на каждой итерации система возвращается в одну и ту же точку с заданной точностью ε – цвет точки, соответствующий сочетанию *a* и *b*, устанавливается красным. Если возвращение происходит через одну итерацию (цикл возврата равен 1) – цвет оранжевый и т.д. до фиолетового цвета радуги, соответствующего циклу возврата 7. Если цикл больше 7, то считается, что система проявляет хаотичное поведение, и точка, соответствующая сочетанию *a* и *b*, окрашивается в чёрный цвет. Также возможны случаи, когда координаты *x*, *y* при заданном в настройках программы количестве итераций уходят в бесконечность – этот процесс называется слоением к бесконечности [10].

Определённое количество итераций до возврата в первоначальную точку называется циклом системы, а сама точка называется неподвижной точкой кратности *n*. В разрабатываемой программе кратность определяется только по восьми последним итерациям. Задав шаг движения по осям *a* и *b* и выводя разноцветные точки, соответствующие состоянию системы, получим диаграмму, называемую картой динамических режимов системы Дуффинга. Этот процесс осуществлён с помощью второй программы, код которой состоит из следующих блоков:

1. Импорт графической библиотеки Matplotlib.

2. Задание пределов изменения коэффициентов точечного отображения Дуффинга *a* и *b*.

3. Присвоение значения переменной ε, указывающей допустимую разницу между сравниваемыми значениями координат системы Дуффинга на последней и предыдущих итерациях (вплоть до восьмой).

4. Основная часть программы, предназначенная для выявления описанной выше возможной цикличности итераций или хаотичного поведения системы и окрашивания точки, соответствующей каждому сочетанию коэффициентов *a* и *b*, в цвет, характеризующий степень устойчивости системы. Если факторы *x* или *y* уходят в бесконечность, то вычисление итерационного цикла прерывается и устанавливается белый цвет точки.

5. Осуществляется перебор значений коэффициентов *a* и *b* с заданным шагом.

6. Вывод в файл точечного графика, на котором цвет каждой точки, соответствующий разным координатам *a* и *b* и характеризующий степень устойчивости системы.

На рис. 5 приведены полученные с помощью второй программы карты динамических режимов системы Дуффинга при разных исходных данных.

Результаты исследования и их обсуждение

С помощью полученного рисунка наглядно видны зоны устойчивого и хаотичного поведения системы Дуффинга. Можно выделить четыре основные области:

1. Чёрная зона в центре, в котором эволюция системы хаотична. Аргументы, для которых выше было получено хаотическое отображение, лежат на границе этой области, но при этом существует много других сочетаний *a* и *b*, при которых система проявляет хаотичное поведение. Исследование характеристик системы при этом возможно с помощью первой разработанной программы.

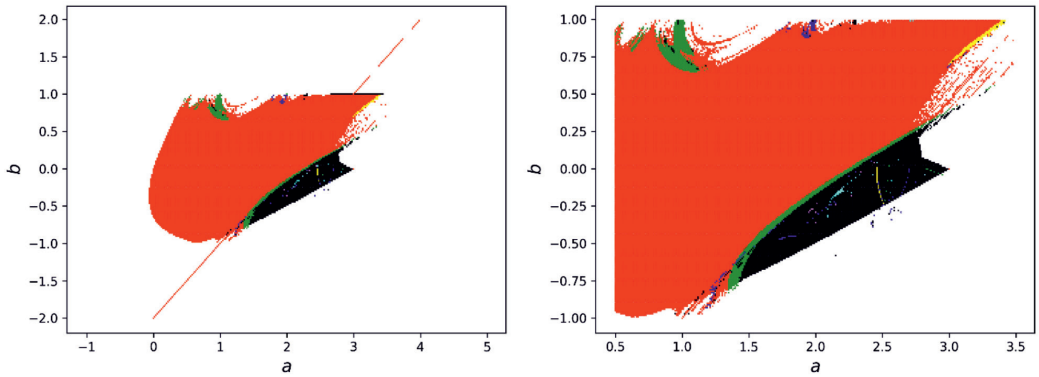


Рис. 5. Карты динамических режимом отображения Дуффинга

2. В центре карты динамических режимов располагается довольно чётко очерченная область устойчивого поведения системы с кратностью возврата 1 (красного цвета), хотя и за её пределами есть отдельные точки, где система столь же устойчива.

3. Зоны хаоса и полной устойчивости разделяет зелёная «линия», с кратностью возврата 4, которую можно условно назвать «областью колебаний».

4. Описанные зоны окружены областью стремления факторов к бесконечности. Это объясняется тем, что в итерационные уравнения входит степенной член.

По рис. 5 можно также сделать вывод, что параметры ε и *t* в исследуемых пределах почти не влияют на получаемую картину.

Методики исследования точечного отображения Дуффинга, описанные в данной статье, можно с успехом использовать также для исследования прочих видов точечных отображений, однако это потребует модификации кодов разработанных программ. Такая осознанная модификация позволит учёным глубже понять смысл и возможности применения одного из перспективнейших методов исследования – с использованием точечных отображений.

Заключение

Любой тип известных на данный момент точечных отображений позволяет описывать определённые классы реальных физических явлений, причём в данном исследовании показано, насколько наглядным является такой метод. Он является примером использования современных компьютерных математических алгоритмов для моделирования сложных физических систем (в данном случае динамических нелинейных). В описываемом исследовании разработаны две компьютерные программы, обладающие достаточно широкими возможностями, в том числе для визуального отображения результатов. Показано преимущество языка Python, обладающего всем необходимым набором модулей для организации обозначенных в целях статьи методах исследования.

При проведении исследований с помощью разработанной программы, строящей карту динамических режимов, установлен новый факт. Принято полагать, что хаотическое поведение системы Дуффинга наблюдается при следующих значениях коэффициентов отображения: $a = 2{,}75$ и $b = 0{,}2$. Однако, как показывают описанные исследования, существует достаточно обширная область значений, при которых наблюдается хаос.

Описанная методика должна помочь при совершенствовании современных методов исследования сложных систем с помощью доступных программных методов.

Список литературы

1. Бендерская Е.Н. Нелинейные динамические системы как основа новых интеллектуальных систем // Стохастическая оптимизация в информатике. 2012. Т. 8. № 2. С. 136–150.

2. Антоновская О.Г. Метод точечных отображений в моделировании системы с комбинированным частотно-фазовым управлением: сборник материалов VI Международной научной конференции, посвященной памяти Б.А. Рогозина. 2018. С. 91–93.

3. Ильичев В.Ю. Разработка программы для исследования аттрактора Лоренца и ее использование // Сложные системы. 2021. № 1 (38). С. 56–63.

4. Программы для исследования точечного отображения Дуффинга. [Электронный ресурс]. URL: http://turbopython.ru/Duffing (дата обращения: 21.07.2021).

5. Ильичев В.Ю., Витчук П.В. Исследование динамических характеристик пластинчатых муфт // Известия Тульского государственного университета. Технические науки. 2016. № 4. С. 217–225.

6. Грибов А.Ф., Краснов И.К. Элементарное введение в теорию одномерных дискретных динамических систем // Modern European Researches. 2019. № 5. С. 49–56.

7. Дубровский А.Д. Природа хаоса в консервативных и диссипативных системах осциллятора Дюффинга – Холмса // Дифференциальные уравнения. 2010. Т. 46. № 11. С. 1652–1656.

8. Ильичев В.Ю. Разработка программных средств увеличения изображений с использованием их фрактальных свойств // Системный администратор. 2021. № 1–2 (218–219). С. 124–127.

9. Сердалиев М.Т., Султамурат О.Д. Анализ данных на основе алгоритмов // Polish Journal of Science. 2020. № 34–2 (34). С. 34–39.

10. Богатырёв А.Б. Комбинаторный анализ отображений периодов: топология двумерных слоев // Математический сборник. 2019. Т. 210. № 11. С. 24–57.