УДК 519.683.8 DOI 10.46973/0201–727X_2023_1_136

Е. А. Альтман, А. В. Александров

АНАЛИЗ ЗАВИСИМОСТИ БЫСТРОДЕЙСТВИЯ БЫСТРОГО ПРЕОБРАЗОВАНИЯ ФУРЬЕ ОТ ОБЪЕМА ОБРАБАТЫВАЕМЫХ ДАННЫХ

Аннотация. Приведен краткий анализ зависимости быстродействия алгоритмов расчета быстрого преобразования Фурье от объема обрабатываемых данных. Все существующие алгоритмы в работе не учитывают аппаратные возможности вычислительных устройств в полном объеме. В результате неоптимального использования памяти процессоров для различных объемов данных количество операций может сильно возрастать. Визуальный анализ графиков зависимостей производительности быстрого преобразования Фурье (БПФ) от его размера позволяет предположить кусочно-непрерывный характер этой зависимости и причины возникновения различных непрерывных участков графика. При разработке реализаций алгоритмов, использующих БПФ, необходимо получить метод численной оценки границ участков графиков. Разрабатываемый метод позволяет с высокой степенью достоверности определить границы участков на графике зависимости производительности БПФ от его размера, на которых изменяется производная графика. Алгоритмы цифровой обработки сигналов, использующие БПФ, желательно разрабатывать таким образом, чтобы размер БПФ не превышал границу, определяемую объемом кэш-памяти первого уровня.

Ключевые слова: быстрое преобразование Фурье, кусочно-линейная аппроксимация, вычислительная эффективность, производительность, быстродействие, архитектура аппаратных средств, границы участков.

Для цитирования: Альтман, Е. А. Анализ зависимости быстродействия быстрого преобразования Фурье от объема обрабатываемых данных / Е. А. Альтман, А. В. Александров // Вестник Ростовского государственного университета путей сообщения. – 2023. – № 1. – С. 136–143. – DOI 10.46973/0201–727X_2023_1_136.

Введение

Быстрое преобразование Фурье (БПФ) помимо непосредственного вычисления дискретного преобразования Фурье (ДПФ) находит применение и для быстрого вычисления других операций. Хорошо известны и широко применяются алгоритмы вычисления с помощью БПФ свертки, корреляции, перемножения матриц и других операций [1].

Обычно операции, вычисление которых ускоряется с помощью БПФ, имеют квадратичную зависимость сложности операции от объема обрабатываемых данных. Если N – это количество чисел в исходных данных, то количество требуемых для вычисления операций равно $C \cdot N^2$, где C – это некоторый коэффициент. Применение БПФ позволяет изменить формулу для количества вычислительных операций на $C' \cdot N\log(N)$, при этом C' в несколько раз больше, чем C. Эффективность применения БПФ зависит от соотношения коэффициентов C, C' и N, обычно алгоритм с использованием БПФ показывает наибольшую эффективность при достаточно большом значении N.

Вопрос о целесообразности использования БПФ для быстрой реализации операций свертки осложняется еще двумя факторами.

Во-первых, часть алгоритмов позволяют использовать БПФ различной длительности. Например, для вычисления свертки через БПФ существуют различные подходы к применению БПФ (перекрытие с накоплением, перекрытие с суммированием), причем в каждом из подходов возможно применение БПФ различной длины.

Во-вторых, при практической реализации БПФ его реальное быстродействие зависит от количества требуемых вычислительных операций нелинейно. Данная зависимость носит сложный характер и зависит от ряда факторов, определяемых архитектурой системы. Прежде всего это количество регистров процессоров, объем кэш-памяти каждого уровня, объем оперативной памяти и др.

В данной статье исследуется вопрос о виде зависимости практического быстродействия БПФ от размера преобразования, в частности, вопрос нахождения границ участков этой зависимости, на которых применение БПФ наиболее эффективно.

Обзор проблемы

Теоретическое быстродействие алгоритмов БПФ принято определять по количеству арифметических операций, требуемых для выполнения преобразования. Классическому алгоритму БПФ Кули – Тьюки требовалось выполнить примерно $5N\log(N)$ вещественных арифметических операций. Предложенный в 1968 году алгоритм Split Radix FFT снизил это число до $4N\log(N)$. В 2007 году это число удалось снизить до $(\frac{34}{9} \approx 3,78)N\log(N)$ [2].

Практическое быстродействие реализаций алгоритмов БПФ в некоторой степени коррелирует с теоретическим, однако не в меньшей степени зависит от архитектуры вычислительного устройства, качества реализации и степени оптимизации программного обеспечения. Хотя зависимость времени выполнения БПФ от его размера носит такой же характер $(C'' \cdot N\log(N))$, коэффициент C'' может быть определен только экспериментально.

В [3] для определения практического быстродействия БПФ используется величина, названная *производительность БПФ* (FFT performance). Данная величина измеряется в мегафлопсах (mflops, flops – FLoating-point OPerations per Second, количество операций с плавающей точкой в секунду). Численно эта величина для комплексного преобразования определяется как

$$mflops = \frac{5N\log(N)}{t},$$

а для вещественного как

$$mflops = \frac{2.5N\log(N)}{t},$$

где t – время выполнения одного преобразования БПФ в микросекундах.

Физический смысл производительности БПФ заключается в том, что эта величина показывает количество вычислительных операций алгоритма БПФ, выполняемых вычислительным устройством за одну секунду.

В идеальном вычислительном устройстве, в котором нет задержек в доставке операндов к арифметико-логическому устройству (АЛУ), производительность БПФ была бы константой, определяемой количеством и быстродействием АЛУ. В реальном устройстве из-за ограничений в количестве регистров процессора, пропускной способности шины, объема кэш-памяти и др. АЛУ не загружается полностью и производительность падает.

Рассмотрим примеры зависимостей быстродействия БПФ от объема данных, представленные на рис. 1. Расчет проводился с помощью ЭВМ на базе процессора Intel i3 10105 3,7 ГГц. На этом рисунке по оси абсцисс в логарифмическом масштабе отложен размер БПФ, по оси ординат – производительность БПФ в mflops. На рисунке показаны производительность прямого БПФ для комплексных вещественных чисел с двойной точностью.

Обе зависимости имеют явно выраженные участки с различными наклонами графика. На первом участке производительность быстро нарастает, на втором не изменяется или медленно падает, на третьем падает быстро.

Можно предположить, что такой вид зависимости связан с архитектурой памяти устройства. На первом участке ограничения производительности связаны с недостаточным объемом операндов, чтобы загрузить все АЛУ вычислительного устройства. На втором участке вступает ограничение, связанное с объемом одного из уровней кэш-памяти, на третьем – аналогичное ограничение на другом уровне архитектуры памяти. Из этого предположения следует, что точность определения изломов на границах участков играет существенную роль в корректном определении зависимости быстродействия БПФ от его размеров.

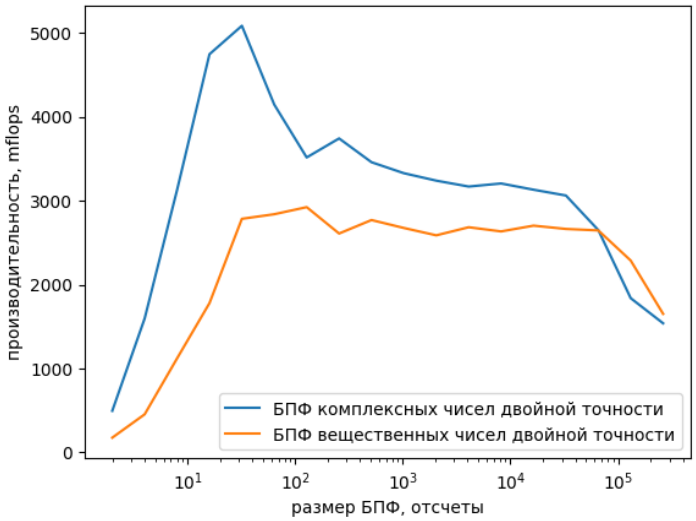

Рис. 1. Сравнение производительности БПФ для комплексных и вещественных чисел с двойной точностью

Для проверки нашего предположения рассмотрим еще одну пару графиков зависимости производительности от размера БПФ, представленную на рис. 2, где показаны зависимости БПФ для чисел с одинарной и двойной точностью.

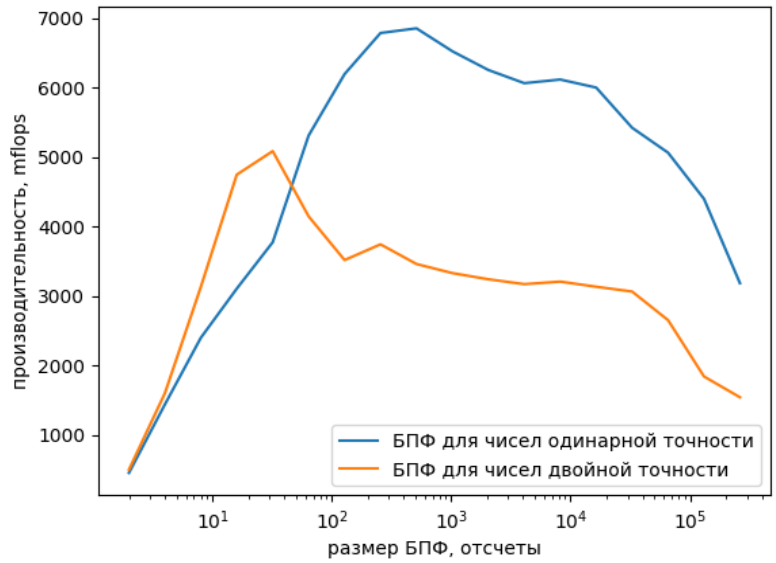

Рис. 2. Смещение границ участков в зависимости от типа данных

Данное предположение согласуется с тем, что границы участков графика зависимости БПФ для вещественных чисел сдвинуты в большую область относительно границ участков БПФ для комплексных чисел (из-за этого в конце мы наблюдаем пересечение графиков). Очевидно, это происходит вследствие того, что комплексные числа занимают в два раза больше места в памяти, чем вещественные.

На рис. 2 мы опять наблюдаем, что границы участков для БПФ, работающего с меньшими объемами данных (с одинарной точностью), смещены вправо относительно границ участков для БПФ, работающего с большими объемами данных (с двойной точностью).

Предлагаемый метод

Визуальный анализ графиков зависимостей производительности БПФ от его размера позволяет предположить кусочно-непрерывный характер этой зависимости и причины наличия различных непрерывных участков графика. Для получения научно-обоснованных выводов и практического их применения при разработке реализаций алгоритмов, использующих БПФ, необходимо получить метод

численной оценки границ участков графиков (далее по тексту – метод) и определить достоверность этой оценки.

В открытом доступе имеется достаточно большая база результатов измерений производительности БПФ [4]. Хотя ее размер недостаточен для использования методов машинного обучения, данных достаточно для статистической проверки разрабатываемого нами метода.

Критерием достоверности метода будет минимальное значение среднеквадратического отклонения получаемых им оценок границ участков для разных алгоритмов БПФ при заданном вычислительном устройстве и типе преобразования БПФ (далее для краткости под термином «отклонения» будем подразумевать такого рода отклонения). Алгоритм применялся для расчетов свертки с использованием различных методов БПФ и аппаратных электронно-вычислительных устройств различных архитектур (одно и многоядерных).

В качестве базового варианта получения численных оценок границ участков графиков было рассмотрено применение известных алгоритмов аппроксимации кусочно-непрерывных функций. В качестве реализации таких алгоритмов выбрана библиотека для кусочно-линейной аппроксимации pwlt (piecewise linear functions) языка Python.

В результатах оценки границ на графиках производительности БПФ с помощью кусочно-линейной аппроксимации наблюдались существенные отклонения. Анализ показал, что они возникают из-за особенностей кусочно-линейной аппроксимации, которые можно увидеть на примере графика производительности комплексного преобразования Фурье библиотекой fftw3 (рис. 3).

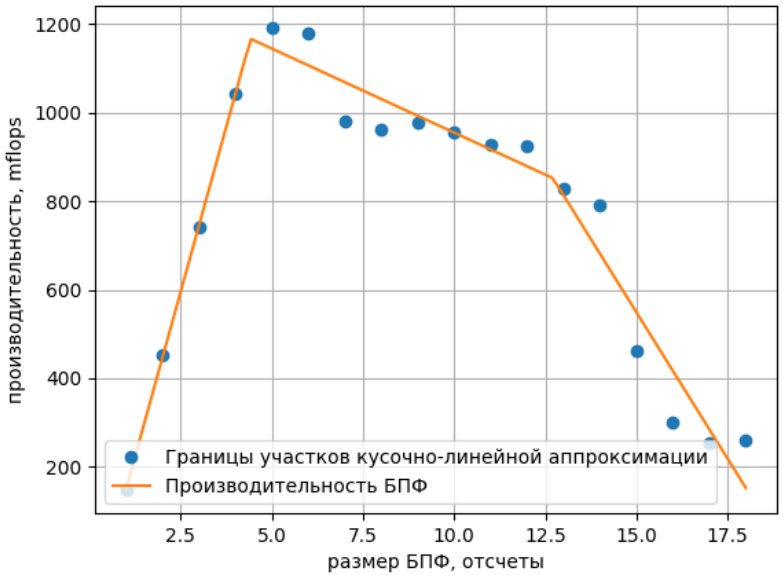

Рис. 3. Ошибка определения границ участков алгоритмов кусочно-линейной аппроксимации

Очевидно, что в данном случае кусочно-линейная аппроксимация ошибочно определила, как первую, так и вторую границы участка графика.

Подобные ошибки при определении границ встречались при различных настройках алгоритма аппроксимации. В частности, увеличение количества линейных участков приводило к увеличению отклонений.

Улучшить качество аппроксимации можно с помощью перехода к кусочно-нелинейной аппроксимации. Ввиду отсутствия в свободном доступе реализаций универсальных алгоритмов для кусочно-нелинейной аппроксимации была реализована собственная функция, использующая полный перебор границ.

Результаты оценки границ графиков производительности БПФ с помощью кусочно-нелинейной аппроксимации также содержат большое количество отклонений. Например, на рис. 4 приведена кусочно-квадратичная аппроксимация графика производительности комплексного преобразования Фурье библиотекой fftw3.

Анализ результатов первых двух вариантов метода показывает, что алгоритмы кусочной аппроксимации в общем виде не позволяют достаточно точно оценить границы участков. Поэтому было решено разработать метод, учитывающий особенности исследуемых зависимостей.

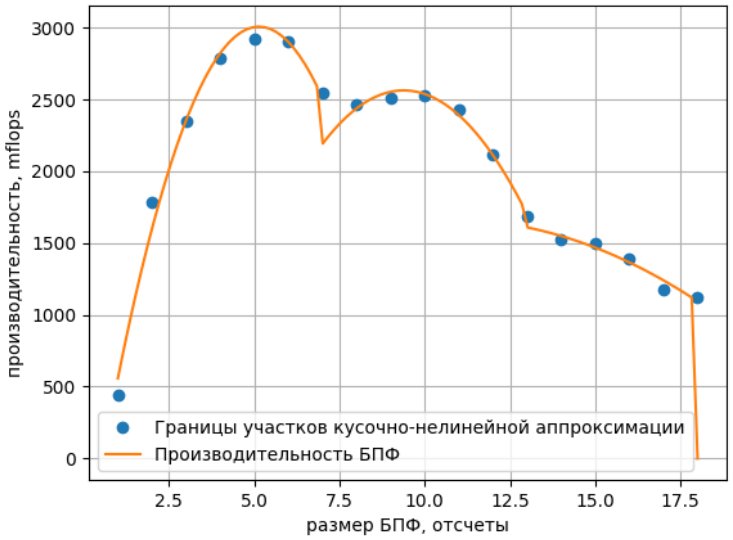

Рис. 4. Границы участков БПФ для кусочно-нелинейного алгоритма

Первой особенностью зависимости является рост графика до определенной границы, после которой начинается медленный спад. Определить такую границу можно несколькими способами (по двум, трем, четырем точкам, с использованием различных коэффициентов). По результатам проведенных расчетов наилучшие результаты показало следующее условие для границы:

$$y_i < y_{i-1} \cdot th, \tag{1}$$

где y – производительность БПФ с индексом i;

i – граничный индекс отсчета;

th – экспериментально определяемый пороговый коэффициент.

В результате экспериментов наименьшее отклонение было при $th = 0{,}95$.

Второй особенностью зависимости является существенный спад производительности в конце графика. Для определения границы этого спада предлагается использовать следующую экспериментально подобранную формулу для оценки изменения производной графика:

$$(y_{i-2} - y_{i-1}) < (y_i - y_{i+1}) \cdot th. \tag{2}$$

В окончательном виде предлагаемый метод формулируется следующим образом:

– с начала последовательности отсчетов сигнала производится поиск отсчета, который удовлетворяет условию (1); индекс этого отсчета будет являться границей первого участка графика;

– поиск второй границы участка будет начинаться со следующего отсчета за найденным на первом этапе и продолжаться до тех пор, пока не будет выполняться условие (2); это будет вторая граница.

Экспериментальное исследование

Основные результаты исследований представлены в табл. 1 (первая граница) и табл. 2 (вторая граница). Исследования проводились с применением библиотек: Intel Integrated Performance Primitives (IPPS) [5], FFTW 3 [6], Spiral [7], Ooura [8] и оптимизированной библиотеки БПФ для RISC процессоров 2.0. Значения в табл. 1 и 2 соответствуют размеру БПФ.

В строках указаны различные процессоры ЭВМ, для которых производились вычисления. В столбцах указаны методы, применяемые для реализации БПФ.

Таблица 1

Первая граница

Тип процессора	IPPS	FFTW3	Spiral	Ooura	FFTs for RISC 2.0
1.266 GHz Intel Pentium 3	512	128	64	32	16
1.6 GHz Intel Pentium M (Banias)	64	128	64	32	16
3.0 GHz Intel Xeon Core Duo (Woodcrest)	1024	128	32	32	16
3.60 GHz Intel Xeon Pentium 4 (Prescott)	1024	64	32	32	16

Таблица 2

Вторая граница

Тип процессора	IPPS	FFTW3	Spiral	Ooura	FFTs for RISC 2.0
1.266 GHz Intel Pentium 3	8192	1024	512	256	128
1.6 GHz Intel Pentium M (Banias)	512	1024	512	256	128
3.0 GHz Intel Xeon Core Duo (Woodcrest)	16384	1024	512	256	256
3.60 GHz Intel Xeon Pentium 4 (Prescott)	8192	512	1024	256	128

В экспериментах использовались наиболее быстродействующие реализации БПФ, они приведены в порядке убывания быстродействия.

Результаты и выводы

Стабильность представленных результатов позволяет с высокой степенью достоверности утверждать, что предлагаемый метод определяет границы участков на графике зависимости производительности БПФ от его размера, на которых происходит значительное изменение скорости обработки сигнала.

Вторая граница находится на уровне, порядок которого соответствует размеру кэша данных первого уровня, равного 32 кбайт для всех рассматриваемых процессоров. Это вполне согласуется с тем фактом, что после второй границы начинается быстрое падение производительности БПФ, и дополнительно подтверждает правильность предлагаемого метода.

По поводу первой границы можно высказать предположение, что из-за небольшого объема данных на первом участке графика происходит неполная загрузка вычислительных элементов процессора.

Положение первой и второй границ коррелирует с общим быстродействием реализаций БПФ: чем дальше граница, тем более быстродействующей оказывается реализация. Данный факт подчеркивает важность эффективной работы с архитектурой памяти вычислительного устройства при разработке вычислительных алгоритмов, работающих с большими объемами данных.

Реализация IPPS заметно отличается от остальных, и определенные для нее границы не вполне соответствуют приведенным выше выводам. Данная реализация от фирмы Intel разработана для процессоров Intel с учетом особенностей их архитектуры. Очевидно, что разработчики этой реализации предприняли специальные меры, чтобы сгладить ограничения архитектуры памяти этих процессоров.

Алгоритмы цифровой обработки сигналов, использующие БПФ, желательно разрабатывать таким образом, чтобы размер БПФ не превышал границу, определяемую доступным объемом данных кэш-памяти первого уровня.

Список литературы

1 Система скрытой передачи информации на базе квазиортогональных сигналов / Е. А. Альтман, А. Г. Малютин, Ю. В. Романов [и др.] // Успехи современной радиоэлектроники. – 2012. – № 11. – С. 26–31. – ISSN 2070–0784.

2 **Steven, G. J.** A modified split–radix FFT with fewer arithmetic operations / G. J. Steven, M. Frigo // IEEE Transactions on Signal Processing. – 2007. – No. 55(1). – P. 111–119.

3 FFT Benchmark Methodology // FFTW : [website]. – URL: https://fftw.org/speed/method.html (date of access: 12/23/2022).

4 FFT Benchmark Results // FFTW : [website]. – URL: https://fftw.org/speed/ (date of access: 12/23/2022).

5 Intel Integrated Performance Primitives Reference Manual // NACAD. – URL: http://www.nacad.ufrj.br/online/intel/Documentation/en_US/ipp/ippsman.pdf (date of access: 12/23/2022).

6 **Frigo, M.** FFTW / M. Frigo // FFTW – URL: https://www.fftw.org/fftw3.pdf date of access: 12/23/2022).

7 SPIRAL User Manual. // SPIRAL Team : [website]. – URL: https://spiral–software.github.io/spiral–software/ (date of access: 12/23/2022).

8 General Purpose FFT (Fast Fourier/Cosine/Sine Transform) Package // Research Institute for Mathematical Sciences (RIMS) : [website]. – URL: https://www.kurims.kyoto-u.ac.jp/~ooura/fft.html (date of access: 12/23/2022).

References

1 System of hidden information transmission on the basis of quasi-orthogonal signals / E. A. Altman, A. G. Malyutin, Y. V. Romanov [et al.] // Advances of modern radioelectronics. – 2012. – No. 11. – P. 26–31. – ISSN 2070–0784.

2 **Steven, G. J.** A modified split–radix FFT with fewer arithmetic operations / G. J. Steven, M. Frigo // IEEE Transactions on Signal Processing. – 2007. – No. 55(1). – P. 111–119.

3 FFT Benchmark Methodology // FFTW : [website]. – URL: https://fftw.org/speed/method.html (date of access: 12/23/2022).

4 FFT Benchmark Results // FFTW : [website]. – URL: https://fftw.org/speed/ (date of access: 12/23/2022).

5 Intel Integrated Performance Primitives Reference Manual // NACAD. – URL: http://www.nacad.ufrj.br/online/intel/Documentation/en_US/ipp/ippsman.pdf (date of access: 12/23/2022).

6 **Frigo, M.** FFTW / M. Frigo // FFTW. – URL: https://www.fftw.org/fftw3.pdf (date of access: 12/23/2022).

7 SPIRAL User Manual // SPIRAL Team : [website]. – URL: https://spiral–software.github.io/spiral–software/ (date of access: 12/23/2022).

8 General Purpose FFT (Fast Fourier/Cosine/Sine Transform) Package // Research Institute for Mathematical Sciences (RIMS) : [website]. – URL: https://www.kurims.kyoto-u.ac.jp/~ooura/fft.html (date of access: 12/23/2022).

E. A. Altman, A. V. Aleksandrov

ANALYSIS OF THE DEPENDENCE OF THE FAST FOURIER TRANSFORM PERFORMANCE ON THE AMOUNT OF PROCESSED DATA

Abstract. This paper gives brief analysis the algorithms performance dependence for computing the fast Fourier transform on the volume of processed data. All existing algorithms in work do not take into account hardware capabilities of computing devices in full. As a result of non-optimal use of processor memory for different data volumes, the number of operations can greatly increase. Visual analysis of graphs dependence FFT performance on its size allows us to suggest a piecewise continuous nature of this dependence and the reasons for various noncontinuous sections of the graph. When developing implementations of algorithms that use FFT, it is necessary to obtain a method for numerical evaluation of the boundaries of the graph sections. The method under development allows to determine with a high degree of reliability the boundaries of sections on the FFT performance dependence graph, on which the derivative of the graph changes. Algorithms of digital signal processing that uses FFT, it is desirable to

develop in such a way that the FFT size does not exceed the boundary determined by the data cache size of the first level of memory cache.

Keywords: fast Fourier transform, piecewise linear approximation, computational efficiency, performance, speed, hardware architecture, area boundaries.

For citation: Altman, E. A. Analysis of the dependence of the fast Fourier transform performance on the amount of processed data / E. A. Altman, A. V. Aleksandrov // Vestnik Rostovskogo Gosudarstvennogo Universiteta Putey Soobshcheniya. – 2023. – No. 1. – P. 136–143. – DOI 10.46973/0201–727X_2023_1_136.

Сведения об авторах

Альтман Евгений Анатольевич
Омский государственный университет путей сообщения (ОмГУПС),
кафедра «Автоматика и системы управления»,
кандидат технических наук, доцент,
e-mail: AltmanEA@gmail.com

Александров Александр Владимирович
Омский государственный университет путей сообщения (ОмГУПС),
кафедра «Автоматика и системы управления»,
аспирант, старший преподаватель,
e-mail: Alexandrov_A_V@mail.ru

Information about the authors

Altman Eugeniy Anatolievich
Omsk State Transport University (OSTU),
Chair «Automation and Control Systems»,
Candidate of Engineering Sciences, Associate Professor,
e-mail: AltmanEA@gmail.com

Aleksandrov Aleksander Vladimirovich
Omsk State Transport University (OSTU),
Chair «Automation and Control Systems»,
Postgraduate Student, Senior Lecturer,
e-mail: Alexandrov_A_V@mail.ru