УДК 519.246.27

DOI: 10.15827/0236-235X.125.159-166

Дата подачи статьи: 22.05.18

2019. Т. 32. № 1. С. 159–166

Специализированное программное обеспечение измерительной системы для оперативного оценивания спектрального состава многокомпонентных процессов

*В.Н. Якимов¹, д.т.н., профессор, yvnr@hotmail.com**А.В. Машиков¹, старший преподаватель, arintelligence@yandex.ru**А.В. Желонкин¹, студент, zhelonkin.ru@yandex.ru*¹ Самарский государственный технический университет, г. Самара, 443100, Россия

В статье рассматривается разработка ПО измерительной системы, предназначенной для проведения оперативного спектрального анализа многокомпонентных процессов. При разработке данного ПО выбрана парадигма модульного построения программных компонентов. Такой подход позволил обеспечить гибкость, расширяемость и масштабируемость ПО, а также заменяемость программных компонентов и возможность их повторного использования.

В соответствии с выбранным подходом ПО измерительной системы состоит из модуля реализации алгоритма спектрального анализа, модуля визуализации измерительных данных и конвертора обработки данных, обеспечивающего взаимодействие между этими модулями. Структура ПО основана на многоуровневой организации вычислительной среды. В соответствии с этим выделены уровни представления и обработки данных. Это позволило отделить метрологически значимые и незначимые программные компоненты и обеспечить защиту измерительной информации от непреднамеренных и преднамеренных изменений. Уровень представления реализован на языке Java, а уровень обработки на языке C++. Выбор языка Java обусловлен наличием компонент поддержки цифровых технологий отображения данных и пользовательского интерфейса, что позволило сделать приложение функционально гибким в использовании, осуществлять управление процессом спектрального анализа, включая операции ввода данных, визуализации результатов в графическом и табличном представлениях.

Процедуры вычисления спектральных оценок реализованы в модуле метрологически значимого ПО на языке C++, позволившего применить подход многопоточного программирования для повышения производительности обработки данных в ходе спектрального анализа.

Ключевые слова: многокомпонентный процесс, аналого-стохастическое квантование, знаковый сигнал, оперативный спектральный анализ, программное обеспечение, измерительная система, многоуровневая структура.

Техническая диагностика работоспособности механических систем и оборудования во многих случаях требует оперативного оценивания *спектральной плотности мощности* (СПМ) многокомпонентных процессов. Одним из основных классических подходов к оцениванию СПМ является коррелограммный метод. Этот метод относится к наиболее устойчивым (робастным) методам спектрального анализа [1]. В настоящее время оценивание СПМ согласно коррелограммному методу, как правило, осуществляется в цифровом виде. При этом современные средства измерений, предназначенные для спектрального анализа, представляют собой высокоавтоматизированные измерительные системы, в которых операции, связанные с получением оценок СПМ, выполняются программным способом [2–5].

Программный способ реализации коррелограммного метода оценивания СПМ не вызывает существенных трудностей. Однако в своем классическом исполнении коррелограммный метод, во-первых, требует предварительного оценивания *корреляционной функции* (КФ), а во-вторых, его программная реализация связана с необходимостью выполнения большого числа цифровых операций умножения при вычислении оценок как КФ, так и СПМ. Все это ведет к снижению оперативности оценивания СПМ. Отсюда следует актуальность задачи разработки цифрового алгоритма оценивания СПМ, программная реализация которого

обеспечивала бы вычислительную эффективность спектрального анализа.

Метод решения

Для решения поставленной задачи предложено использовать *бинарное знаковое аналого-стохастическое квантование* (БЗАСК) в качестве первичного преобразования непрерывного во времени анализируемого многокомпонентного процесса $X(t)$ [6–8]:

$$z(t) = \begin{cases} +1, & X(t) \geq \xi(t), \\ -1, & X(t) < \xi(t), \end{cases}$$

где $\xi(t)$ – *вспомогательный случайный сигнал* (ВСС).

ВСС $\xi(t)$ является равномерно распределенным и изменяется в пределах от $-\xi_{\max}$ до $+\xi_{\max}$. Здесь $\xi_{\max} \geq |X_{\sup}|$, где $|X_{\sup}|$ – наибольшее по абсолютной величине значение динамического диапазона, в пределах которого может принимать свои значения многокомпонентный процесс $X(t)$.

Особенностью БЗАСК является возможность осуществлять двухуровневое преобразование непрерывных процессов без систематической погрешности независимо от их статистических свойств. При этом его результат $z(t)$ можно рассматривать как непрерывный во времени знаковый сигнал, ограниченный по уровню значениями -1 и $+1$.

В ходе выполнения двух процедур БЗАСК предварительно ^оцентрированной реализации $x(t)$ многоком-

понтного процесса $X(t)$, то есть после удаления постоянной составляющей, сформируем на интервалах времени $0 \leq t \leq T$ и $0 \leq t \leq 2T$ знаковые сигналы $z_1(t)$ и $z_2(t)$ соответственно.

Найдем оценку КФ [4]:

$$\hat{R}_{XX}(\tau) = \xi_{\max}^2 T^{-1} \int_0^T z_1(t) z_2(t + \tau) dt.$$

Тогда коррелограммная оценка СПМ с учетом четности КФ будет следующей:

$$\hat{S}_{XX}(f) = 2\xi_{\max}^2 T^{-1} \int_0^T z_1(t) \int_t^{T+t} z_2(\tau) \cos 2\pi f(\tau - t) d\tau dt. \quad (1)$$

Для однозначного представления сигналов $z_1(t)$ и $z_2(t)$ во времени достаточно знать только их значения $z_1(t_0)$ и $z_2(t_0)$ в начальный момент времени квантования и моменты времени, в которые каждый из них пересекает нулевой уровень:

$$\{t_i^{z_1} : 1 \leq i \leq I - 1\} \text{ и } \{t_j^{z_2} : 1 \leq j \leq J - 1\}, \quad (2)$$

где $t_0^{z_1} = t_0^{z_2} = t_0 = 0$, $t_I^{z_1} = T$ и $t_J^{z_2} = 2T$.

На интервалах времени, определяемых моментами времени (2), знаковые сигналы $z_1(t)$ и $z_2(t)$ остаются постоянными. С учетом этого интегралы в (1) могут быть вычислены аналитически. Будем вычислять оценки СПМ на частотах $f_n = n\Delta f$, где $\Delta f = 1/T$ – предельное разрешение по частоте. Тогда после вычисления интегралов в (1) получаем:

$$\hat{S}_{XX}(n\Delta f) = \frac{A(\Delta f)}{n^2} \sum_{i=0}^I (-1)^i \lambda_i z_2(t_i^{z_1}) B_{i,n}, \quad (3)$$

$$B_{i,n} = C_i + \sum_{j=m(i)+1}^{m(i)+r(i)} (-1)^{j-m(i)} \cos 2\pi n\Delta f (t_j^{z_2} - t_i^{z_1}), \quad (4)$$

$$A(\Delta f) = z_1(t_0) \frac{\pi^{-2} \xi_{\max}^2}{\Delta f},$$

$$\lambda_i = \begin{cases} 1, & i = 0, i = I; \\ 2, & 1 \leq i \leq I - 1; \end{cases}$$

$$C_i = \begin{cases} 0, & r(i) - \text{четное}; \\ 1, & r(i) - \text{нечетное}; \end{cases}$$

$$\{t_{m(i)+1}^{z_2}, t_{m(i)+2}^{z_2}, \dots, t_{m(i)+r(i)}^{z_2}\} \in \{t_j^{z_2} : 0 \leq j \leq J\}.$$

Отсчеты времени $\{t_{m(i)+1}^{z_2}, t_{m(i)+2}^{z_2}, \dots, t_{m(i)+r(i)}^{z_2}\}$ соответствуют тем моментам, в которые знаковый сигнал $z_2(t)$ меняет свое значение на интервале от $t_i^{z_1} = t_{m(i)}^{z_2}$ до $t_i^{z_1} + T = t_{m(i)+r(i)+1}^{z_2}$.

Соотношения (3) и (4) с учетом коэффициента $A(\Delta f)$, который остается постоянным для заданного разрешения по частоте $\Delta f = 1/T$, определяют процедуру дискретного вычисления оценок СПМ. Нетрудно заметить, что реализация этих соотношений не требует предварительного оценивания КФ анализируемого процесса, а также практически не связана с выполнением цифровых операций умножения. Основными

операциями являются алгебраическое суммирование и вычитание отсчетов косинусоидальной функции $\cos 2\pi n\Delta f (t_j^{z_2} - t_i^{z_1})$. Отсутствие необходимости в вычислении оценок КФ и переход к более простым алгебраическим операциям существенно повышают оперативность цифровых процедур в ходе вычисления оценок СПМ.

Программная реализация алгоритма вычисления оценок СПМ

На основе (3) и (4) разработан алгоритм вычисления оценок СПМ в цифровом виде, который, в свою очередь, стал основой разработки специализированного ПО измерительной системы для спектрального анализа.

Необходимость в создании специализированного ПО измерительной системы, реализующего оценивание спектральных составляющих многокомпонентных процессов, вызвана тем, что существующие проприетарные программные продукты либо имеют избыточный функционал для решения узкоспециализированных задач, либо не обеспечивают гибкости в предоставлении, поддержке и модификации библиотек, удовлетворяющих специфическим требованиям.

Разработанное ПО состоит из трех основных частей: системного, общего прикладного и метрологически значимого ПО.

Системное ПО представляет собой совокупность программных средств, обеспечивающих общее управление аппаратно-программными компонентами измерительной системы.

Общее прикладное ПО предназначено для выполнения типовых процедур обработки и обмена данными, которые предусмотрены процессом измерений.

Согласно требованиям к ПО средств измерений (ГОСТ 8.654-2015), метрологически значимое ПО представляет собой совокупность программных модулей, выполняющих функции сбора, передачи, обработки, хранения и представления измерительной информации.

Подобного рода разделение ПО измерительной системы предоставляет возможность модификации его метрологически незначимых частей (пользовательского интерфейса, системных библиотек, процедур взаимодействия с операционной средой и периферийными устройствами) без нарушения его соответствия утвержденному ПО и защиты от случайных или непреднамеренных изменений данных, получаемых из метрологически значимых частей ПО.

Для реализации метрологически значимых и незначимых частей в структуре ПО измерительной системы выделены уровни представления и обработки данных. На рисунке 1 приведена обобщенная схема многоуровневой структуры ПО измерительной системы.

Уровень представления соответствует метрологически незначимой части ПО и отвечает за графическое

представление данных и интерфейс взаимодействия с пользователем.

Уровень обработки данных соответствует метрологически значимой части ПО и содержит модуль реализации алгоритма вычисления оценок СПМ.

Уровень представления реализован на строго типизированном объектно-ориентированном языке программирования Java, а уровень обработки – на статически типизированном языке программирования общего назначения C++. Использование языка Java обеспечило создание программных конструкций с возможностью повторного использования. В свою очередь, C++ позволил применить подход многопоточ-

ного программирования для повышения производительности обработки данных.

При разработке ПО измерительной системы была выбрана модульная парадигма организации структуры ПО. Это позволило обеспечить выполнимость таких критериев, как адаптивность и расширяемость системы, масштабируемость процесса разработки, заменяемость программных компонентов и возможность их повторного использования [9, 10]. ПО состоит из следующих основных модулей: модуль визуализации измерительных данных, модуль реализации алгоритма вычисления оценки СПМ и конвертор обработки данных между пользовательским интерфейсом и модулем реализации алгоритмов.

Основу пользовательского интерфейса составляет модуль графической визуализации данных, позволяющий отображать полученные результаты обработки данных в графической форме.

Модуль анализа данных представляет инструмент спектрального анализа на основе разработанного алгоритма вычисления оценок СПМ.

Модуль восстановления сигнала позволяет восстановить гармонические составляющие многокомпонентного процесса после проведения спектрального анализа.

Реализация программного кода модулей анализа данных и восстановления сигнала осуществляется через блок модулей конвертирования данных между языками программирования Java и C++. Использование связующих модулей позволяет заменить их при необходимости использования под другим графическим интерфейсом. Взаимодействие модулей ПО и организация обмена данными показаны на рисунке 2.

Дополнительно разработаны модуль моделирования многокомпонентного процесса, который обеспечивает формирование аддитивной смеси гармонических составляющих и широкополосного шума с заданными характеристиками, и модуль имитации БЗАСК. Эти модули предназначены для отладки и тестирования разработанного ПО в ходе его расширения и модификации.

Модульная архитектура ПО измерительной системы позволила обеспечить принцип разделения ответственности программных компонентов, где каждый компонент отвечает за выполнение только своей задачи и взаимодействует с компонентами других уровней по установленным связям через разработанные интерфейсы. При этом учтено требование конструктивной и функциональной однородности программных модулей [11, 12].

Пользовательский интерфейс системы реализован на языке Java в среде разработки IntelliJ IDEA 2017.1.4 на основе технологии JavaFX. Для удобства все элементы представления вынесены в fxml-разметку, а обработка событий в контроллеры.

Общая структура интерфейса приложения представлена на рисунке 3 в виде диаграммы классов универсального языка моделирования UML:

- класс MainController является корневым обработчиком событий пользовательского интерфейса;
- класс SignalTabController представляет общий обработчик вкладки сигналов и выборки целевых обработчиков;
- классы RandomProcessController и ExcelProcessController определяют программный и графический интерфейсы модуля загрузки данных анализируемого процесса;
- класс SubSignalsTabController представляет обработчик вкладки создания вспомогательных сигналов;
- класс AnalysisTabController представляет обработчик вкладки анализа данных колебательного процесса.

Основной задачей являлась разработка метрологически значимого ПО, которое реализует разработанный алгоритм вычисления оценок СПМ. ПО приложения, реализующего данный алгоритм, написано на языке C++ в среде разработки VisualStudio 2017. При этом используются библиотека xInt для чтения электронных таблиц форматаxlsx, а также механизм JavaNativeInterface (JNI) для запуска кода под управлением виртуальной машины Java (JVM), который написан на языке C++. Для большей гибкости код, реализу-

Fig. 3. A flowchart of the user interface of the information measuring system software

ющий алгоритм, разделен на две части: Core – модуль реализации процедур выполнения алгоритма; Java-Bridge – модуль преобразования java-объектов в стандартные типы данных языка C++.

Согласно концепции объектно-ориентированного программирования, основу информационного обеспечения измерительной системы составили четыре класса-контейнера.

1. Классы-контейнеры для передачи данных о моделируемом процессе:

- класс QuantumBridge используется для загрузки нативных библиотек подсчета и анализа данных;
- класс JNIRandomProcessModel хранит общие данные о модели анализируемого процесса: параметры гармонических составляющих, закон распределения шума, среднее значение и отклонение;
- класс JNIRandomProcessComponent хранит данные гармонической составляющей: тип гармонического компонента, амплитуда, частота и фаза.

2. Классы-контейнеры для передачи данных о загружаемом реальном сигнале из Excel-файла:

- класс JNIExcelProcessData хранит общие данные о загружаемом сигнале: путь до файла, расширение файла, тип аргумента, интервал дискретизации, номер колонки аргумента, коэффициент децимации, массив данных о загружаемых массивах;
- класс JNIExcelProcessComponent хранит данные о загружаемом из файла массиве: номер колонки, номер первой строки, количество строк.

3. Классы-контейнеры для получения данных сигналов и их анализа:

- класс JNIExcelSeriesResearch хранит общие данные о загруженных сигналах: матрица значений

сигналов, массив контейнеров с результатами анализов по каждому сигналу;

- класс JNIExcelSeriesComponent хранит данные об анализе сигнала: максимальное и минимальное значения, абсолютное максимальное значение, математическое ожидание, дисперсия, среднее квадратическое отклонение, коэффициент корреляции, массив центрированных значений.

4. Класс-контейнер JNIFourierData хранит общие данные для спектрального анализа: массив сигналов, постоянная составляющая, среднее значение, отклонение, амплитуда, период и фаза вспомогательного сигнала, время анализа, интервал опроса, минимальное и максимальное значения частоты.

Общая структура адаптера для модуля подсчетов представлена на рисунке 4.

Перечисленные классы являются абстрактными. Они определяют интерфейсы по взаимодействию отдельных компонент ПО измерительной системы друг с другом и реализуют базовые операции по обработке данных. Они также являются основой для реализации способов загрузки данных исследуемого процесса, алгоритмов подготовки и анализа путем переопределения объявленных процедур.

Наличие различных функциональных компонентов ПО обусловило разграничение интерфейса метрологически значимого ПО на интерфейс пользователя и программный интерфейс. При проектировании пользовательского интерфейса особое внимание уделено вопросам пространственного представления информации. Это качественный признак, определяющий, насколько правильно спроектированы структура ПО и эргономичный для пользователя интерфейс. Логиче-

ски организованная информация обеспечивает удобную и единообразную навигацию как в пределах одного окна, так и в целом по множеству окон и вкладок. Пользовательский интерфейс ПО информационной измерительной системы показан на рисунке 5.

Вкладка «Сигнал» позволяет работать в двух режимах путем выбора типа источника данных:

- «загрузка реального сигнала», который позволяет загрузить реальный процесс из текстового файла или электронной таблицы формата *xlsx*;
- «модель реализации случайного процесса», который формирует тестовый процесс.

В режиме тестирования имеется возможность задать число и параметры гармонических составляющих (амплитуду, частоту, период и начальную фазу), а также выбрать вид аддитивного шума (равномерный или нормальный).

На рисунке 6 показана вкладка «Анализ», на которой задаются параметры для выполнения спектрального оценивания. На этой же вкладке расположен модуль визуального отображения графической информации. При этом имеется возможность выбора частотного диапазона для детального просмотра спектральных оценок с помощью двойного ползунка. Также выводится таблица численных значений результатов вычисления оценок СПМ с разрешением по частоте $\Delta f = 1/T$.

Результаты экспериментального тестирования

Экспериментальные исследования алгоритма и тестовые испытания ПО проводились с использованием

специально разработанного модуля имитационного моделирования. Имитация центрированной реализации $\overset{\circ}{x}(t)$ многокомпонентного процесса $X(t)$ осуществлялась в виде суммы статистически независимых гармонических компонент и аддитивного шума $e(t)$.

Параметры гармонических компонент (значения частот и амплитуд) подбирались так, чтобы по их спектрам можно было судить о разрешающей способности и надежности идентификации спектральных составляющих. Амплитуды гармонических компонент принимали значения от нуля до единицы и интерпретировались как нормированные, их частоты также задавались нормированными: $f_k^H = f_k / f_{\max}$, где $0 < f_k < f_{\max}$ и $f_{\max}$ – верхняя частота диапазона, в границах которого имитировалось вычисление оценок СПМ. Нормированные частоты обеспечило постоянство частотного диапазона спектрального оценивания от нуля до 0,5. Начальные фазы φ_i задавались в соответствии с равномерным законом распределения в пределах их периода следования. Шум $e(t)$ представлял собой белый шум, который имел нулевое математическое ожидание и дисперсию σ_e^2 . В частности, модель реализации $\overset{\circ}{x}(t)$ содержала семь гармонических компонент, параметры которых приведены в таблице. При этом дисперсия шума была равна единице.

Результаты анализа модели реализации $\overset{\circ}{x}(t)$ представлены на верхнем графике рисунка 6. Спектральные линии четко разрешимы по частоте, сильные гармонические компоненты не маскируют слабые. Даже

Рис. 5. Пользовательский интерфейс. Вкладка «Сигнал»

Fig. 5. User interface. Signal tab

Рис. 6. Пользовательский интерфейс. Вкладка «Анализ»

Fig. 6. User interface. Analysis tab

при высоком уровне шума (дисперсия равна единице) гармонические компоненты с нормированной амплитудой 0,1 четко различимы на его фоне.

Параметры гармонических компонент

модели реализации $x(t)$

Parameters of harmonic components

of the signal model $x(t)$

k	f_k^H	α_k^H	α_k/σ_e , дБ
1	0,22	0,1	-20
2	0,23	0,3	-10
3	0,24	0,5	-6
4	0,25	1,0	0
5	0,26	0,5	-6
6	0,27	0,3	-10
7	0,28	0,1	-20

Заключение

Задача разработки цифрового алгоритма оценивания СПМ на основе коррелограммного метода, программная реализация которого обеспечивала бы вычислительную эффективность спектрального анализа, по своему характеру являлась комплексной и потребовала решения ряда сопутствующих задач.

Прежде всего был разработан цифровой алгоритм, который не требует предварительного оценивания КФ анализируемого процесса. При этом в отличие от классического подхода к цифровой реализации

коррелограммного метода он не требует выполнения многочисленных операций цифрового умножения, связанных с обработкой многоуровневых отсчетов анализируемого процесса. Исключение процедуры вычисления оценок КФ и переход к более простым алгебраическим операциям суммирования и вычитания повышают оперативность цифровых процедур вычисления оценок СПМ.

Дополнительно повышение вычислительной эффективности разработанного алгоритма спектрального анализа решается на программном уровне за счет выбора концепции многоуровневой архитектуры построения ПО для его практической реализации. В результате использования данной концепции ПО измерительной системы имеет модульную структуру, где модули отвечают за выполнение только своей задачи и взаимодействуют с вычислительной средой через программные интерфейсы, представляющие собой классы-контейнеры. Такой подход обеспечил гибкость, расширяемость, масштабируемость и заменяемость программных компонентов, а также возможность их повторного использования.

Работа выполнена при финансовой поддержке РФФИ, грант № 19-08-00228-А.

Литература

1. Марпл.-мл. С.Л. Цифровой спектральный анализ и его приложения. М.: Мир, 1990. 584 с.
2. Tan Li, Jiang Jean. Digital signal processing fundamentals and applications. Acad. Press, 2013, 896 p. DOI: 10.1016/C2011-0-05250-X.
3. Прохоров С.А., Столбова А.А. Программный комплекс ана-

лиза неэквидистантных временных рядов на основе непрерывного вейвлет-преобразования // Программные продукты и системы. 2017. № 4. С. 668–671. DOI: 10.15827/0236-235X.120.668-671.

4. Чижилова Л.А. Принципы проектирования модульной архитектуры программного обеспечения авиационной тематики // Программные продукты и системы. 2017. № 2. С. 291–300. DOI: 10.15827/0236-235X.118.291-300.

5. Luo Fa-Long, Williams W., Rao R., Narasimha R., Montpetit Marie-José. Trends in signal processing applications and industry technology. *IEEE Signal Process Mag.* 2012, vol. 29, pp. 171–174. DOI: 10.1109/MSP.2011.943129.

6. Макс Ж. Методы и техника обработки сигналов при физических измерениях. М.: Мир, 1983. Т. 1. 312 с.

7. Мирский Г.Я. Характеристики стохастической взаимосвязи и их измерения. М.: Энергоиздат, 1982. 320 с.

8. Якимов В.Н. Цифровой комплексный статистический анализ на основе знакового представления случайных процессов // Изв. СамНЦ РАН. 2016. Т. 18. № 4. С. 1346–1353.

9. Гамма Э., Хелм Р., Джонсон Р., Влиссидес Д. Приемы объектно-ориентированного проектирования. Паттерны проектирования. СПб: Питер, 2016. 366 с.

10. Eeles P., Cripps P. *The process of software architecting*. Addison Wesley, 2010, 432 p.

11. Якимов В.Н., Горбачев О.В. Программное обеспечение системы измерения амплитудных спектров колебательных процессов // Программные продукты и системы. 2013. № 2. С. 166–171.

12. Якимов В.Н., Горбачев О.В. Программно-аппаратное обеспечение системы оценки амплитудного спектра многокомпонентных процессов // Приборы и техника эксперимента. 2013. № 5. С. 49–55.

Software & Systems

DOI: 10.15827/0236-235X.125.159-166

Received 22.05.18

2019, vol. 32, no. 1, pp. 159–166

Specialized software of the measuring system for the operative estimating the spectral composition of multicomponent processes

V.N. Yakimov¹, Dr.Sc. (Engineering), Professor, yvnr@hotmail.com

A.V. Mashkov¹, Senior lecturer, arintelligence@yandex.ru

A.V. Zhelonkin¹, Student, zhelonkin.ru@yandex.ru

¹ Samara State Technical University, Samara, 443100, Russian Federation

Abstract. The paper considers the problem of software development of a measuring system for operative spectral analysis. The software development includes a modular paradigm for organizing software components. It allows ensuring flexibility and extensibility of the system, development process scalability, interchangeability of software components and the possibility of their reuse.

According to the chosen approach, the visualization system software consists of a module of a spectral analysis algorithm, a module for measuring data visualization, and a data processing converter that provides communication between these modules. The software structure is based on a multi-level organization of the computing environment. The levels of data representation and processing are highlighted. This allows separating metrologically significant and insignificant parts and protecting measuring information against unintentional and deliberate changes. The presentation level is implemented in Java, and the processing level is in C++. The choice of Java is due to the availability of support components for digital display technologies and user interface, which made it possible to make the application functionally flexible in use, to manage the spectral analysis process including data entry operations, graphical and tabular visualization of results.

The procedures of calculating spectral estimates are implemented in the metrologically significant software module in C++, which enable multithreaded programming to improve data processing performance during the spectral analysis.

Keywords: random process, analog-stochastic quantization, sign-function signal, operative spectral analysis, software, information-measuring system, multi-level structure.

Acknowledgements. The work has been financially supported by RFBR, project no. 16-08-00269-A.

References

1. Marple S.L. *Digital spectral analysis: with applications*. Englewood Cliffs, NJ, Prentice Hall Publ., 1987, 492 p. (Russ. ed.: Moscow, Mir Publ., 1990, 584 p.).

2. Li T., Jean J. *Digital signal processing fundamentals and applications*. 2nd ed., Academic Press, 2013, 896 p. DOI: 10.1016/C2011-0-05250-X.

3. Prokhorov S.A., Stolbova A.A. A software package for nonuniform time series analysis based on continuous wavelet transformation. *Software & Systems*. 2017, vol. 4, pp. 668–671 (in Russ.). DOI: 10.15827/0236-235X.120.668-671.

4. Chizhikova L.A. Design principles of modular software architecture in aviation. *Software & Systems*. 2017, vol. 2, pp. 668–671 (in Russ.). DOI: 10.15827/0236-235X.118.291-300.

5. Luo Fa-Long, Williams W., Rao R., Narasimha R., Montpetit M.-J. Trends in signal processing applications and industry technology. *IEEE Signal Process Mag.* 2012, vol. 29, pp. 171–174. DOI: 10.1109/MSP.2011.943129.

6. Max J. *Methodes et techniques de traitement du signal et applications aux mesures, physiques*. Masson, 1981, vol. 1, 312 p. (Russ. ed.: Moscow, Mir Publ., 1983, 312 p.).

7. Mirsky G.Ya. *Characteristics of stochastic relations and their measurement*. Moscow, Energoizdat Publ., 1982, 320 p.

8. Yakimov V.N. Digital complex statistical analysis based on sign-function representation of random processes. *Izvestia of Samara Scientific Center of RAS*. 2016, vol. 18, no. 4, pp. 1346–1353 (in Russ.).

9. Gamma E., Helm R., Johnson R., Vlissides J. *Design patterns: elements of reusable object-oriented software*. Addison Wesley Longman Publ., 1994, 416 p.

10. Eeles P., Cripps P. *The process of software architecting*. Addison Wesley Publ., 2010, 432 p.

11. Yakimov V.N., Gorbachev O.V. Software for vibration processes amplitude spectrum measurement system. *Software & Systems*. 2013, vol. 2, pp. 166–171 (in Russ.).

12. Yakimov V.N., Gorbachev O.V. Firmware of the amplitude spectrum evaluating system for multicomponent processes. *Instruments and Experimental Techniques*. Springer US Publ., 2013, vol. 56, no. 5, pp. 540–545.