

Дмитрий Михайлович Клионский, доцент кафедры математического обеспечения и применения ЭВМ Санкт-Петербургского государственного электротехнического университета «ЛЭТИ» им. В.И. Ульянова (Ленина), кандидат технических наук, klio2003@list.ru

Олег Владимирович Перчёнок, м.н.с. кафедры высшей математики Санкт-Петербургского государственного электротехнического университета «ЛЭТИ», кандидат технических наук, olegpercb@gmail.com

СИСТЕМА ЭЛЕКТРОННОГО ОБУЧЕНИЯ, ОБЕСПЕЧИВАЮЩАЯ ВЕРИФИКАЦИЮ РЕШЕНИЯ ЗАДАЧ В ОБЛАСТИ ЦИФРОВОЙ ОБРАБОТКИ СИГНАЛОВ ПО ПРЕДМЕТНО-ОРИЕНТИРОВАННОМУ ОПИСАНИЮ ИХ УСЛОВИЙ

В системах автоматизированного контроля знаний, применяемых в российских учреждениях высшего профессионального образования, обычно используется принцип тестирования: студенту предлагается несколько вариантов ответа, из которых он выбирает один или несколько. При этом тестирующую систему «не интересует» непосредственно алгоритм решения, которым пользовался студент. Тем самым становится возможной ситуация, когда студент может наугад выбрать вариант ответа, который, если повезёт, окажется верным.

На факультете компьютерных технологий и информатики СПбГЭТУ «ЛЭТИ» в рамках Программы стратегического развития (ПСР) на 2012–2016 гг. создаётся научно-педагогическая школа методов и технических средств цифровой обработки сигналов (ЦОС). В данной школе предполагается использование принципиально иного подхода к контролю знаний студентов

и иных учащихся (курсы повышения квалификации и прочее), основанного на проверке алгоритма решения задачи, что исключает возможность угадывания правильного ответа.

Ранее проводилась разработка систем электронного обучения в соответствии с обозначенным подходом, но в других предметных областях. В качестве примера можно

привести систему «Конструктор комбинаторных коллекций», обеспечивающую автоматическую верификацию решений задач по комбинаторике на основе формального описания условия¹, а также систему WiseTasksGeometry², осуществляющую аналогичную поддержку задач на геометрические построения³.

В статье рассматривается подход к автоматизации верификации задач в области ЦОС с использованием предметно-ориентированного языка системы WiseTasks DSP. Решение задач происходит на языке MATLAB⁴. Важно отметить, что система WiseTasksDSP предназначена не только для изучения ЦОС и проверки знаний студентов по ЦОС, но и для освоения соответствующих программных средств MATLAB, реализующих цифровую фильтрацию, спектральный анализ, многоскоростную обработку сигналов, квантование и другие операции ЦОС.

¹ Богданов М.С. Автоматизация проверки решения задач по формальному описанию ее условия. Компьютерные инструменты в образовании, № 4 (2006), СПб, АНО КИО. С. 51–57.

² Wise tasks — умные задачи (англ.).

³ Перченко О.В. Разработка системы электронного обучения для автоматической верификации решений задач по описанию условия на предметно-ориентированных языках. Материалы XVIII Международной научно-методической конференции «Современное образование: содержание, технологии, качество». ООО «Технолит», СПб, 2012. Т. 1. С. 223–224.

Перченко О.В., Поздняков С.Н., Посов И.А. Автоматизация проверки решения геометрических задач по описанию их условий на предметно-ориентированном языке. АНО «КИО», Компьютерные инструменты в образовании (№ 1, 2012). С. 37–44.

⁴ Дьяконов В.П. MATLAB 7.*/R2006/2007. Самоучитель. М.: «ДМК-Пресс», 2008.

Существующие пособия по изучению MATLAB можно классифицировать по трём типам:

- 1) набор заданий для самостоятельного выполнения в среде MATLAB;
- 2) видеодемонстрация решения задач в среде MATLAB;
- 3) интерактивный видеоролик решения задач в среде MATLAB.

Верификацию решений задач обеспечивают пособия третьего типа, например, тренажёр решения задач по обработке аудиосигналов на сайте Mathworks⁵. Однако такие задачи не предполагают написание М-файлов, а поддерживают только ввод решения в командной строке. Поэтому данный тренажёр позволяет решать только простые задачи, решение которых умещается в 2–3 строки. Кроме того, верификация в данном случае представляет собой посимвольное сравнение введённых тестируемым строк с эталонным решением, что отсекает возможность решения задачи разными способами.

Например, такая простая задача, как передискретизация (изменение частоты дискретизации) звука, записанного в матрицу y , с частоты 44100 Гц на 16000 Гц имеет единственное правильное решение с использованием тренажёра Mathworks:

$$yI = \text{resample}(y, 16000, 44100).$$

Следующее фактически правильное решение при данном подходе будет признано неверным:

$$fs = 44100; fs_new = 16000; yI = \text{resample}(y, fs_new, fs).$$

При решении данной задачи в системе WiseTasksDSP верификация будет состоять в следующем: на различных исходных сиг-

⁵ www.mathworks.com

налах y будет проверяться правильность выдаваемых скриптом студента выходных сигналов y/l . Если решение студента прошло все тесты, то оно признаётся верным. При этом не важно, каким именно способом студент решал задачу.

Модель жизненного цикла системы, поддерживающей самопроверяемые задачи

В данном разделе описывается общий подход к разработке систем автоматической верификации задач⁶.

Модель жизненного цикла системы изображена на рис. 1 в форме UML-диаграммы деятельности⁷. В данной диаграмме действуют следующие актёры:

- *метапреподаватель* — опытный преподаватель, глубоко владеющий предметной областью и являющийся инженером по знаниям (умеет строить онтологии);
- *преподаватель* — пользователь системы электронного обучения, составляющий задачи для системы, анализирующий результаты их решения и имеющий возможность обратной связи с программистом и преподавателем;
- *программист* — разработчик системы электронного обучения, взаимодействующий в процессе работы с метапреподавателем и преподавателем;
- *студент* — пользователь системы электронного обучения, который решает в ней задачи и имеет возможность обратной связи с преподавателем.

Описание действий модели приведено в табл. 1.

⁶ Перченко О.В. Модель жизненного цикла системы электронного обучения, обеспечивающей автоматическую проверку задач// Известия СПбГЭТУ «ЛЭТИ». 2013. № 7. С. 117–124.

⁷ Леоненков А.В. Самоучитель UML 2. СПб. БХВ-Петербург. 2007. 576 с.

Таблица 1

Описание модели жизненного цикла системы

Действие	Описание
Составление онтологии предметной области	Метапреподаватель составляет на основе своих знаний о предметной области её онтологию, которая будет лежать в основе предметно-ориентированных языков описания условия и решения. При этом он принимает во внимание замечания преподавателя, возникшие в результате эксплуатации системы.
Разработка предметно-ориентированных языков	Метапреподаватель разрабатывает предметно-ориентированные языки описания условия задач и описания решения задач на основе онтологии предметной области, онтологии абстрактных задач и абстрактной модели языка.
Проверка языков на реализуемость	Программист проверяет возможность реализации языков описания условия задач и описания решения задач. Если у него есть замечания, он сообщает их метапреподавателю, иначе приступает к проектированию и реализации новой системы или модификации существующей.

Окончание Таблицы 1

Проектирование и реализация системы	Программист проектирует, реализует и тестирует новую систему электронного обучения или модифицирует существующую систему. При этом он принимает во внимание замечания преподавателя, возникшие в результате эксплуатации системы.
Составление задач	Преподаватель составляет новые задачи на предметно-ориентированном языке описания условий (или корректирует ранее составленные задачи), принимая во внимание вопросы студентов, возникшие при решении ранее составленных задач, и выводы, сделанные им в процессе анализа результатов решения задач студентами. При наличии замечаний по языку преподаватель сообщает их метапреподавателю. При наличии замечаний по интерфейсу преподаватель сообщает их программисту. При отсутствии замечаний преподаватель даёт студентам задание, состоящее в решении задач с помощью системы.
Решение задач	Студент решает в системе электронного обучения задачи, составленные преподавателем. Если у него возникают вопросы в процессе решения, он задаёт их преподавателю, иначе вызывает автоматическую верификацию решений, результаты которой становятся доступны преподавателю.
Анализ результатов	Преподаватель анализирует результаты автоматической верификации решений студентов. Выводы, сделанные в ходе анализа, используются преподавателем для составления новых или корректировки ранее составленных задач.

Представленная на рис. 1 (с. 96) модель жизненного цикла позволяет спроектировать систему электронного обучения на основании онтологии предметной области, онтологии абстрактных задач, модели абстрактного предметно-ориентированного языка для описания условий задач⁸.

Онтология предметной области

В процессе разработки системы была составлена онтология предметной области

для задач по ЦОС. Онтология была создана в системе Protégé⁹. Иерархия классов данной онтологии изображена на рис. 2 (с. 96), описание классов приведено в табл. 2 (с. 97). В онтологии присутствует одно свойство типа данных (DataTypeProperty) `hasVerifier`, имеющее в качестве области определения (Domain) класс `DSPStatement`, в качестве области значения (Range) тип `string`.

⁸ Генерация математических задач и верификация решений в автоматизированных системах поддержки обучения / Манцеров Д.И., Перченко О.В., Поздняков С.Н. и др. СПб. Изд-во СПбГЭТУ ЛЭТИ. 2012. 154 с.

⁹ <http://protege.stanford.edu/>

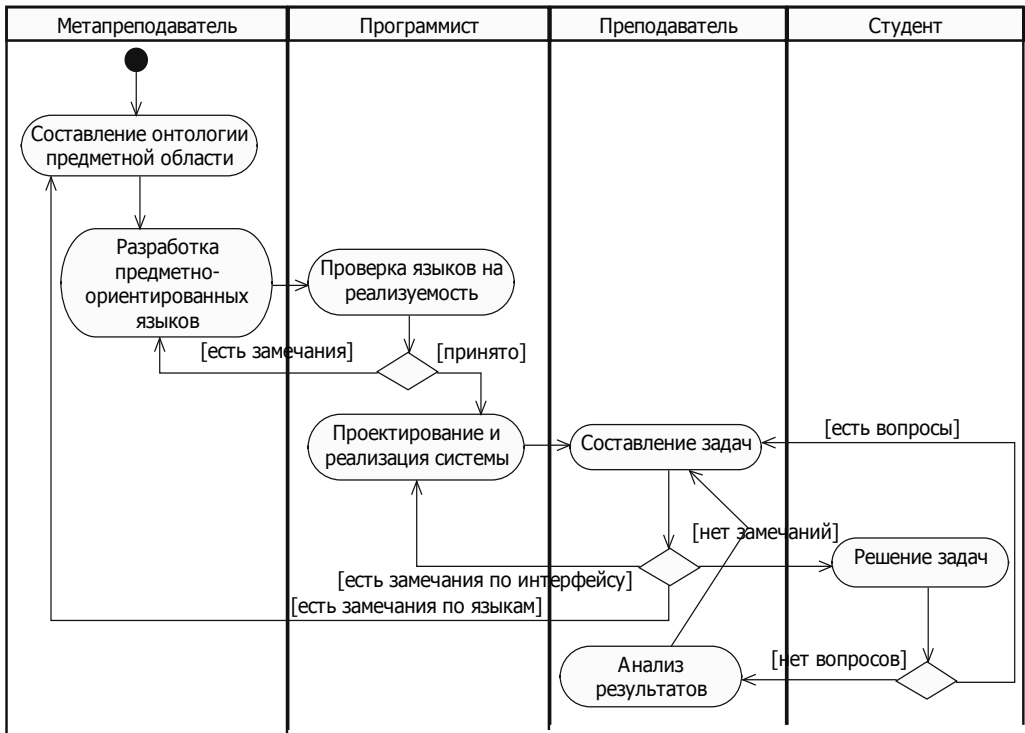


Рис. 1. Модель жизненного цикла системы электронного обучения

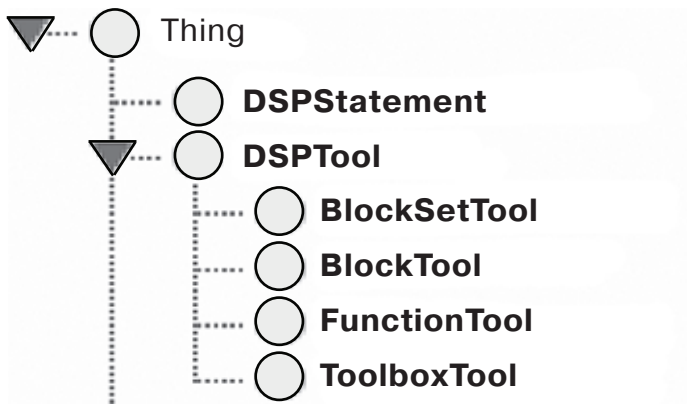


Рис. 2. Иерархия классов онтологии предметной области ЦОС

Таблица 2

Описание классов онтологии предметной области задач по ЦОС

Класс	Описание
DSPStatement	Структурированное условие задачи
DSPTool	Инструмент для решения задач по ЦОС в пакете MATLAB
BlockSetTool	Библиотека элементов Simulink
BlockTool	Элемент Simulink
FunctionTool	Функция MATLAB
ToolboxTool	Библиотека функций MATLAB

Описание системы WiseTasksDSP

Система WiseTasksDSP включает в себя модуль преподавателя и модуль студента¹⁰. В модуле преподавателя педагог вводит условие задачи на предметно-ориентированном языке. Задача по ЦОС имеет следующую структуру:

- заголовок;
- текстовое условие;
- поясняющее изображение к текстовому условию;
- инструменты;
- верификатор.

Наибольший интерес применительно к ЦОС представляют последние две структурные единицы (инструменты и верификатор).

¹⁰ Перченко О.В. Тренажер по решению задач в области цифровой обработки сигналов с помощью инженерно-математического пакета MATLAB [Текст]/Клионский Д.М., Перченко О.В. // Материалы XIX Международной научно-методической конференции «Современное образование: содержание, технологии, качество», 24 апреля 2013. Тезисы доклада. СПб, 2013. Т. 1. С. 149–151.

Перченко О.В. Автоматизация верификации решения задач в области цифровой обработки сигналов [Текст]/Перченко О.В., Клионский Д.М. // Компьютерные инструменты в образовании. СПб. 2012. № 3. С. 28–37.

Инструменты — список функций MATLAB (из стандартного набора) и элементов Simulink, доступных студенту при решении задачи. Также в него можно включать группы функций и элементов. Примеры: «функция fft» (вычисление прямого ДПФ с помощью алгоритмов БПФ), «функция ifft» (вычисление обратного ДПФ с помощью алгоритмов БПФ), «набор функций Signal Processing Toolbox» (библиотека функций для цифровой обработки сигналов), «элемент Сумматор» (элемент Simulink, предназначенный для суммирования входов). При решении задачи студенты также могут использовать любые арифметические функции и операции, поддерживаемые программной системой MATLAB.

Верификатор — логическая функция MATLAB, проверяющая решение ученика путём перебора значений параметров и возвращающая значение «истина», если решение признано верным на всех итерациях циклов. Особенностью функции является обязательное присутствие макрокоманды «Решение», на место которой при проверке подставляется скрипт-решение студента.

Решение задачи осуществляется в модуле студента. Этот модуль позволяет загрузить задачу, ввести скрипт-решение, используя только разрешённые инструменты,

и вызвать проверку, т.е. запустить верификатор с подставленным в него скриптом-решением.

Далее приведён пример структурированной задачи по ЦОС. Она посвящена нахождению верхней граничной частоты спектра аналогового (непрерывного) сигнала и проверке выбранной частоты дискретизации дискретного сигнала на допустимость для точного (без потерь информации) восстановления исходного (аналогового) сигнала из дискретного на основе теоремы Котельникова¹¹. При определении верхней граничной частоты спектра аналогового сигнала используется 10%-ный критерий (на практике также используются и другие критерии определения верхней граничной частоты), т.е. за верхнюю граничную частоту принимается та частота, при которой значение амплитудного спектра сигнала составляет 10% от максимума.

Условие задачи может быть сформулировано на предметно-ориентированном языке следующим образом:

Заголовок. Определение верхней граничной частоты спектра аналогового сигнала и проверка выбранной частоты дискретизации дискретного сигнала в соответствии с теоремой Котельникова.

Текстовое условие. Дан исходный аналоговый сигнал $s(n)$ длины N (отсчёты хранятся в массиве S , T_{an} — шаг по времени для значений аналогового сигнала). Частота дискретизации дискретного сигнала F_s . С помощью MATLAB определить верхнюю граничную частоту спектра аналогового сигнала (F_{up}) по 10%-ному критерию и прове-

рить, подходит ли данная частота дискретизации для представления аналогового сигнала в дискретном виде без потерь информации. Если подходит, присвоить переменной isOK значение true. Если не подходит, присвоить isOK=false и задать минимальную целую частоту дискретизации F_{snew} , для которой выполняется условие теоремы Котельникова.

Инструменты. Набор функций Signal Processing Toolbox

Функция fft

Функция max

Верификатор:

```
function ver=ver_freqs()
M=10; % Количество тестов.
N=100; % Количество отсчётов в массиве
сигнала.
eps=0.001;% Константа для сравнения
истинных частот и значений
% частот, найденных студентом.
thresh=0.1;% Порог в соответствии с 10%-
ным критерием.
T_an=2.2e-5;% Шаг по времени для значе-
ний аналогового сигнала.
Fs=1000; % Частота дискретизации, Гц.
flag=false;% Флаг правильности прохож-
дения i-го теста.
for i = 1:M% Цикл для тестов.
S=rand (N,1);% Массив случайных
чисел с равномерным распределением
% (массив значений аналогового сиг-
нала)
% Макрос для подстановки скрипта-ре-
шения студента.
```

Решение

```
% Проверка правильности решения
W=abs(fft(S));% Амплитудный спектр
сигнала
```

¹¹ Сергиенко А.Б. Цифровая обработка сигналов. 3-е издание. СПб.: БХВ-Петербург, 2011.

```

Wmax=max(W);% Максимальное значение амплитудного спектра
Wup=thresh*Wmax;% Значение амплитудного спектра по заданному критерию
for j= 1:N% Ищем наименьшую частоту, на которой амплитудный спектр не
    if W(j)<=Wup% превышает порога
        Fup1=1/(T_an*N)*(j-1);% Истинная верхняя граничная частота спектра
        break;% сигнала по заданному критерию
    end
end
if abs(Fup1-Fup) < eps% Сравниваем истинную граничную частоту с ответом студента
    if Fs>=2*Fup1% Проверяем выполнение условия теоремы Котельникова
        flag = isOK;
    else% Если оно не выполняется,
        Fsnew1 = ceil(2*Fup1);% то находим истинную наименьшую целую частоту
        % дискретизации, удовлетворяющую условию теоремы Котельникова
        flag = ~isOK & (abs(Fsnew1-Fsnew)<eps);
    end
end
else
    flag=false;
end
if ~flag% Если тест пройден неправильно, то серия тестов прерывается
    break;
end
end
ver=flag;% Если решение правильно прошло все тесты, то оно верно.

```

Задача, введённая в модуле преподавателя, приведена на рис. 3 (с. 99). Ввод условия

осуществляется с помощью структурированного редактора предметно-ориентированных языков¹².

Для решения задачи необходимо её загрузить в модуль студента. Скрипт-решение для рассматриваемой задачи может быть следующим:

```

W=abs(fft(S));% Находим амплитудный спектр сигнала
Wmax=max(W);% Определяем максимальное значение амплитудного спектра
Wup=0.1*Wmax;% Определяем значение амплитудного спектра по заданному критерию
for j= 1:N% Ищем наименьшую частоту, на которой амплитудный спектр не
    if W(j)<=Wup% превышает порога
        Fup=1/(T_an*N)*(j-1);% Верхняя граничная частота
        break;
    end
end
if Fs>=2*Fup% Проверяем выполнение условия теоремы Котельникова
    isOK=true;
else % Если оно не выполняется,
    Fsnew = ceil(2*Fup);% то находим наименьшую целую частоту
    % дискретизации, удовлетворяющую условию теоремы Котельникова
    isOK = false;
end

```

Введя решение, студент отправляет его на проверку, заключающуюся в запуске функции. В ходе проверки приведённая

¹² Перченко О.В., Поздняков С.Н., Посов И.А. Автоматизация проверки решения геометрических задач по описанию их условий на предметно-ориентированном языке. АНО «КИО». Компьютерные инструменты в образовании (№1, 2012). С. 37–44.

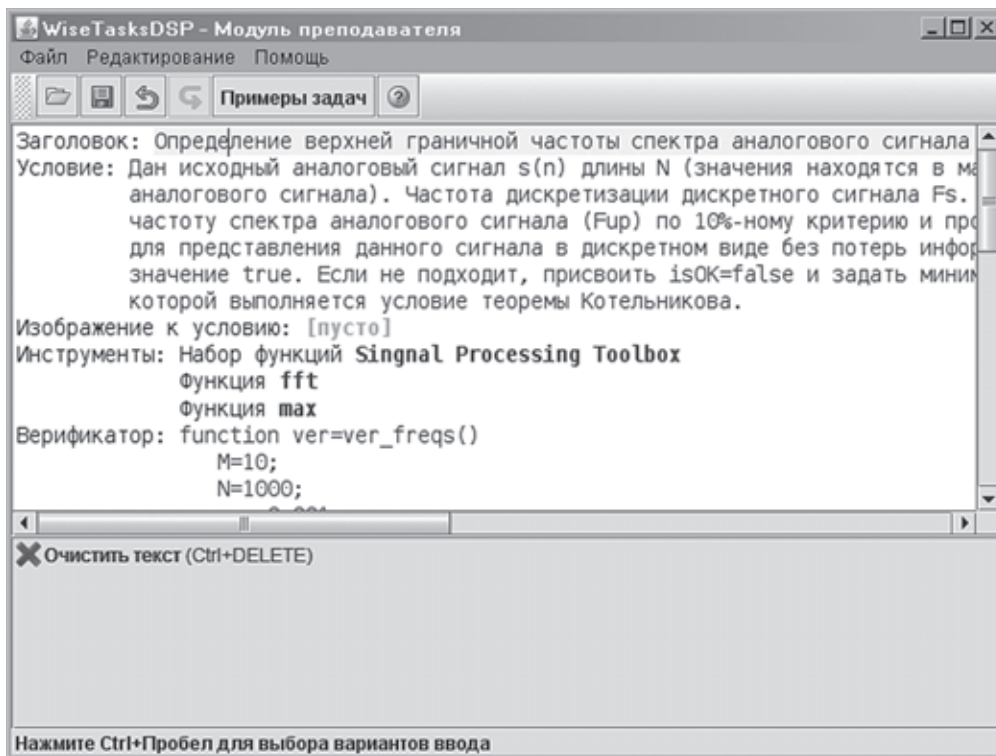


Рис. 3. Модуль преподавателя системы WiseTasksDSP

в листинге 2 функция-верификатор вначале определяет начальные условия для выполнения тестов. Далее в цикле организуется тестирование решения студента. На каждом тесте выполняются следующие шаги:

- 1) определение начальных условий для выполнения решения;
- 2) вызов скрипта-решения;
- 3) проверка правильности ответов, определённых в п. 2.

Рассчитанные ответы зависят от массива отсчётов сигнала, задаваемого во время выполнения каждого теста случайным образом, поэтому студенту бессмысленно сво-

дить скрипт-решение к простому присваиванию переменным F_s и F_{up} числовых значений. В этом случае ответы будут неверны наверняка.

Заключение

В статье обозначен общий подход к проектированию систем автоматической верификации решений задач, продемонстрированы возможности системы WiseTasksDSP для решения учебных задач. В качестве примера приведена одна из часто встречающихся учебных задач по ЦОС, связанная с определением верхней граничной частоты спектра

аналогового сигнала и проверкой частоты дискретизации дискретного сигнала в соответствии с теоремой Котельникова.

Система WiseTasksDSP может также использоваться для решения ряда других известных в ЦОС задач. К их числу относятся, например, поиск оптимальных по методу наименьших квадратов коэффициентов линейного предсказания, синтез КИХ-фильтров с АЧХ, наименее отличающейся от эталонной в смысле заданного критерия, синтез адаптивных фильтров¹³, построение оптимального классификатора в двумерном про-

странстве¹⁴ и пр. Система WiseTasksDSP также является перспективной для многих реальных научных задач, сложность которых на несколько порядков выше, по сравнению с теми задачами, что используются в учебных целях.

Настоящая статья и проведенная научная работа поддержаны в рамках проекта РФФИ «Мой первый грант» (Соглашение № 14-07-31250/14) и проекта Министерства образования и науки Российской Федерации в рамках договора № 02.G25.31.0058 от 12.02.2013.

¹³ Сергиенко А.Б. Цифровая обработка сигналов. 3-е издание СПб.: БХВ-Петербург, 2011.

Солонина А.И., Арбузов С.М. Цифровая обработка сигналов. Моделирование в MATLAB — СПб.: БХВ-Петербург, 2008.

Айфичер Э., Джервис Б. Цифровая обработка сигналов. М.—СПб.—Киев: Вильямс, 2004.

Солонина А.И., Улахович Д.А., Арбузов С.М., Соловьева Е.Б. Основы цифровой обработки сигналов. 2-е изд. СПб.: БХВ-Петербург, 2005.

Солонина А.И., Клионский Д.М., Меркучева Т.В., Перов С.Н. Цифровая обработка сигналов и MATLAB. СПб.: БХВ-Петербург, 2013.

¹⁴ Айвазян С.А., Мхитарян В.С. Прикладная статистика в задачах и упражнениях: Учебник для вузов. М.: Юнити-Дана, 2001.