

УДК 519.876.5

ИССЛЕДОВАНИЕ СРЕДСТВ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ МНОГОМЕРНЫХ СИГНАЛОВ В СИСТЕМАХ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА РЕАЛЬНОГО ВРЕМЕНИ

Ким Т.А., Арещенков Д.А., Сотников А.А.

ФГБОУ ВО «Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)», Москва, e-mail: kimta@bmstu.ru

Увеличение количества сложных технических систем, обрабатывающих многомерные сигналы, актуализирует проблему их тестирования, которое возможно осуществить наиболее полно с помощью имитационного моделирования. Имитационная модель воспроизводит поведение окружения для тестируемой системы и должна удовлетворять определенным требованиям. При этом перед разработчиками модели возникает необходимость определения оптимального набора параметров модели, обеспечивающих удовлетворение модели конкретным предъявленным требованиям – ее адекватность решаемой задаче. Целью работы является исследование аппаратных и программных средств имитационного моделирования, поддерживающих работу систем искусственного интеллекта в реальном времени. Для имитационного моделирования в системах реального времени предпочтительнее использование вычислительных средств специального назначения, многофункциональные модульные платформы имеют преимущество перед другими аппаратными средствами в виде применимости к большому классу задач и меньшей трудоемкости разработки и моделирования. Проведен сравнительный анализ программных средств, написаны программы для оценки скорости и точности выполнения операций цифровой обработки сигналов. Среди программных средств выделены языки программирования, подходящие для прототипирования имитационной модели многомерных сигналов (LabVIEW, Python, MATLAB, Julia) и для ее реализации в системах искусственного интеллекта реального времени (C/C++).

Ключевые слова: имитационное моделирование, многомерные сигналы, программно-аппаратные средства, искусственный интеллект, реальное время

INVESTIGATION OF TOOLS FOR SIMULATION MODELING OF MULTIDIMENSIONAL SIGNALS IN REAL-TIME ARTIFICIAL INTELLIGENCE SYSTEMS

Kim T.A., Areschenkov D.A., Sotnikov A.A.

Bauman Moscow State Technical University, Moscow, e-mail: kimta@bmstu.ru

An increase in the number of complex technical systems that process multidimensional signals actualizes the problem of their testing, which can be carried out most fully using simulation modeling. The simulation model reproduces the behavior of the environment for the system under test and must meet certain requirements. The purpose of the paper is to investigate hardware and software simulation tools that support the operation of artificial intelligence systems in real time. Results: for simulation modeling in real-time systems, it is preferable to use special-purpose computing tools; multifunctional modular platforms have an advantage over other hardware in applicability to a larger class of tasks and less laborious development and modeling. A comparative analysis of software tools has been carried out, programs have been written to evaluate the speed and accuracy of performing digital signal processing operations. Among the software tools, programming languages are selected that are suitable for prototyping a simulation model of multidimensional signals (LabVIEW, Python, MATLAB, Julia) and for its implementation in real-time artificial intelligence systems (C/C++).

Keywords: simulation modeling, multidimensional signals, software and hardware tools, artificial intelligence, real time

Зачастую испытание сложных технических систем на определенных этапах жизненного цикла значительно эффективнее проводить с помощью имитационной модели. Такой подход позволяет снизить стоимость и время испытаний, обеспечивает более полное тестовое покрытие. При этом необходимым условием успешного завершения испытаний является адекватность модели.

Под имитационной моделью авторы понимают программное обеспечение, которое воспроизводит поведение некоторой реальной системы. Имитационная модель позволяет в комфортных для испытателя условиях многократно повторять эксперименты с различными значениями параметров и по-

лучать подробную статистику функционирования системы [1]. Отличительной чертой имитационного моделирования также является возможность управления скоростью течения процессов: время имитации может замедляться в случае с быстропротекающими явлениями или ускоряться для моделирования систем с медленными процессами. Как известно, модель имеет следующие свойства и характеристики [2]:

1) адекватность. Модель должна обеспечивать соответствие объекту в части исследуемых свойств;

2) полнота. Модель должна предоставлять всю необходимую информацию об исследуемом объекте;

3) гибкость. Модель должна предоставлять возможность воспроизведения различных ситуаций в широком диапазоне изменения условий и параметров;

4) чувствительность. Модель должна реагировать на изменения входных данных и корректировать значения выходных данных в соответствии с этими изменениями;

5) точность. Выходные значения модели должны отличаться от выходных значений исследуемого объекта не более чем на заданную величину;

6) устойчивость. Модель должна оставаться адекватной при любых допустимых значениях входных данных и всех возможных параметрах самой модели;

7) масштабируемость. Модель должна поддерживать увеличение объема входных данных, внешних воздействий и выходных значений с сохранением адекватности;

8) производительность. Модель должна выполнять не меньше определенного объема операций за заданный промежуток времени.

Очевидно, что в общем случае одновременное достижение наилучших показателей для всех характеристик невозможно. Например, увеличение точности модели снижает ее производительность. При этом перед разработчиками модели возникает необходимость определения оптимального набора параметров модели, обеспечивающих удовлетворение модели конкретным предъявленным требованиям – ее адекватность решаемой задаче. В свою очередь, такого рода параметрическая перестройка модели накладывает требования на средства имитационного моделирования. Как аппаратная, так и программная их составляющая должны иметь возможность обеспечения характеристик модели во всем динамическом диапазоне.

Схема разработки имитационной модели и ее основные этапы представлены на рис. 1.

Этап 1 – Постановка задачи. Определение структуры исследуемой системы, входных и выходных параметров объекта моделирования, диапазонов их изменения, совокупности ограничений и допущений, требований к характеристикам модели.

Этап 2 – Разработка алгоритмического и математического обеспечения. Описание объектов, среды взаимодействия и процессов средствами математических и алгоритмических моделей и методов. Определение оптимальных методов решения задачи. Предварительная оценка диапазона значений характеристик модели. Методы имитационного моделирования подробно изложены в литературе [3–5].



Рис. 1. Схема разработки имитационной модели

Этап 3 – Выбор средств моделирования. Производится выбор программно-аппаратных средств реализации моделей и методов. При выборе средств моделирования учитывается:

- полнота математического аппарата,
- точность вычислений,
- производительность вычислений,
- масштабируемость реализации,
- простота и легкость освоения,
- доступность и стоимость.

Этап 4 – Разработка программного обеспечения. Реализация (кодирование) математического и алгоритмического обеспечения с использованием выбранных программно-аппаратных средств моделирования.

Этап 5 – Верификация. Оценка соответствия результата разработки программного обеспечения. Неудовлетворительные результаты верификации могут стать основанием для изменения средств моделирования.

Этап 6 – Валидация. Оценка соответствия результата разработки математического и алгоритмического обеспечения. Оценка адекватности полученной модели.

Выбор средств имитационного моделирования должен учитывать особенности имитационной модели и является важным этапом при разработке модели. Как аппаратные, так и программные средства должны иметь возможность обеспечения характеристик модели во всем динамическом диапазоне. Неправильный выбор средств имитационного моделирования является одной из причин неудовлетворительных результатов верификации и валидации модели.

Данная статья посвящена исследованию средств имитационного моделирования многомерных сигналов в системах искусственного интеллекта реального времени.

Анализ аппаратных средств имитационного моделирования

Для реализации имитационных моделей многомерных сигналов наряду с вычислительными средствами общего назначения целесообразно применять специализированные вычислительные средства (процессор цифровой обработки сигналов, программируемая логическая интегральная схема), а также устройства формирования и преобразования электрических сигналов.

Вычислительные средства общего назначения находят применение в задачах общего управления и диспетчеризации имитационной модели, а также реализации интерфейса взаимодействия с пользователем.

Процессор цифровой обработки сигналов (ПЦОС) представляет собой специализированный микропроцессор, предназначенный для решения задач цифровой обработки сигналов (далее – ЦОС) [6]. Система команд арифметико-логического устройства ПЦОС оптимизирована для выполнения базовых команд ЦОС, что позволяет существенно увеличить быстродействие по сравнению с вычислителем общего назначения даже при более низкой тактовой частоте ПЦОС.

Программируемая логическая интегральная схема (далее – ПЛИС) представляет собой логическое устройство, логика работы которого задается пользователем посредством программирования [7]. Поскольку ПЛИС последних поколений оснащаются устройствами быстрого умножения, то это позволяет выполнять операцию «умножения с накоплением» с минимальными аппаратными затратами. Наличие микропроцессора в кристалле ПЛИС создает дополнительные возможности в области ЦОС.

В настоящее время существует ряд стандартных многофункциональных модульных платформ (далее – ММП), представляющих собой некоторый стандарт построения аппаратного обеспечения, а также большой набор серийно выпускаемых модулей, соответствующих стандарту. Используя ММП, пользователь может подобрать необходимый набор модулей, содержащих как вычислительные средства общего назначения, так и специализированные вычислители, а также необходимые для решения конкретной задачи формирователи и преобразователи электрических сигналов для сопряжения с реальными объектами. В зависимости от характеристик ММП и инструментов для отладки такое средство может использоваться одновременно для исследований, прототипирования и реализации систем имитационного моделирования. Такой инструмент позволяет снизить трудоемкость моделирования и повысить степень адекватности получаемых результатов за счет использования готовых аппаратных реализаций, позволяющих моделировать сложные процессы.

Разработка собственной ММП требует большого количества времени, ресурсов и квалифицированных специалистов, поэтому имеет смысл использование существующих ММП с отлаженной архитектурой. Недостатком таких средств является их относительно высокая стоимость.

Анализ программных средств имитационного моделирования

Реализация имитационных моделей должна учитывать следующие особенности разработки программного обеспечения:

1) алгоритмы моделирования многомерных сигналов являются параллельными, то есть предполагающими выполнение более чем одного преобразования в каждый момент времени;

2) объем обрабатываемых данных не является детерминированным и требует динамического распределения оперативной памяти;

3) большое количество вычислительных операций связано с математическим аппаратом цифровой обработки сигналов (преобразование Фурье, расчет функции взаимной корреляции, цифровая фильтрация и т.д.);

4) поддержка на уровне библиотек методов искусственного интеллекта (нейросети, глубокое обучение, эволюционное моделирование, экспертные системы);

5) детерминированное время выполнения операций для обеспечения режима реального времени.

С учетом указанных особенностей авторами проведены исследования современных программных средств имитационного моделирования: C/C++, MATLAB, Python, Julia, LabVIEW.

В качестве критериев сравнения были сформулированы следующие:

1) скорость выполнения базовых операций ЦОС (расчет автокорреляционной функции (далее – АКФ) многомерного сигнала, выполнение прямого и обратного быстрого преобразования Фурье (далее – БПФ) многомерного сигнала;

2) погрешность выполнения прямого и обратного БПФ, которая отражает точность вычислений и влияет на точность модели;

3) наличие инструментов визуализации результатов, позволяющих преобразовывать сигналы в изображения и строить графики для анализа результирующих и промежуточных данных;

4) наличие готовых библиотек цифровой обработки сигналов и библиотек для работы с искусственным интеллектом,

позволяющих сократить время на разработку и реализацию алгоритмов;

5) поддержка выполнения алгоритмов в режиме реального времени;

6) стоимость и сложность развертывания сред программирования.

Для каждого из языков была установлена среда разработки, подключены необходимые библиотеки для ЦОС и визуализации данных, написан код программ с расчетом АКФ и выполнением прямого и обратного БПФ. В качестве исходного сигнала использовался двумерный массив размерностью 1000x1000 точек, заполненный случайными вещественными числами от нуля до единицы.

Написанные программы состоят:

- из чтения исходного двумерного сигнала из файла,
- сохранения времени до выполнения операций ЦОС,
- выполнения операций ЦОС,
- сохранения времени после выполнения операций ЦОС,
- вывода разницы между временными метками.

Таблица 1

Результаты сравнительного анализа программных средств

Критерий сравнения	Julia	Python	MATLAB	C	C++	LabVIEW
Время расчета АКФ, с	0,24	0,32	0,55	0,062	0,076	1,12
Время выполнения БПФ, с	0,086	0,087	0,07	0,034	0,039	0,09
Погрешность выполнения БПФ	1,39E-10	1,55E-10	1,64E-10	1,60E-10	1,60E-10	2,55E-10
Инструменты визуализации результатов	Plots	Matplotlib	Встроены в среду	gnuplot	Matplotlib-cpp, Matplot++, Gnuplot, HippoDraw	Встроенные модули, Advanced Plotting Toolkit
Библиотеки для ЦОС	DSP, Signal Operators	Scipy	Встроены в среду, Signal Processing Toolbox	fftw	fftw, SigPack	Signal Processing Toolkit
Библиотеки для ИИ	Flux, Knet, MXNet, TensorFlow	Keras, Caffe, Scikitlearn, TensorFlow, OpenCV, Dlib	Deep Learning Toolbox	FANN	Tiny_dnn, Apache. SINGA, Dlib, OpenCV, TensorFlow, Caffe	Deep Learning Toolkit
Стоимость	Бесплатно	Бесплатно	Платно	Бесплатно	Бесплатно	Платно
Сложность развертывания	Средней сложности	Средней сложности	Легко	Средней сложности	Средней сложности	Легко
Поддержка Real-Time	Возможно	Затруднительно	Возможно	Да	Да	Да

Приведенный в статье фрагмент исходного кода для C/C++ содержит только измерение времени расчета АКФ без процедуры чтения двумерного сигнала из файла и визуализации рассчитанной АКФ.

В конце программы с выполнением БПФ также сравниваются исходный сигнал и двумерный массив, полученный после прямого и обратного БПФ. Для этого вычисляется разность по модулю между каждым из соответствующих элементов, которая суммируется по всему массиву для получения погрешности выполнения БПФ. Погрешность вычислений возникает из-за ограниченной точности представления чисел в ЭВМ. Для достоверности сравнения в каждом из языков использовался формат представления числа с плавающей точкой, занимающий в памяти 8 байт.

Результаты сравнительного анализа рассмотренных программных средств представлены в табл. 1. Фрагменты исходных

кодов и визуализация рассчитанной АКФ приведены в табл. 2 и 3.

Сравнение вычислительной производительности рассматриваемых программных средств было проведено при выполнении трех операций ЦОС: расчета автокорреляционной функции (АКФ), выполнения прямого и обратного быстрого преобразования Фурье (БПФ). При сопоставлении исходного сигнала и полученного после прямого и обратного БПФ оценена точность выполнения операций. Результаты сравнения отражены в гистограммах на рис. 2.

По результатам измерений времени выполнения операций ЦОС особенно выделяются языки C и C++. Остальные языки между собой мало отличаются по скорости выполнения БПФ, а АКФ быстрее рассчитывается с помощью Julia и Python.

Погрешность выполнения БПФ для всех языков имеет одинаковый порядок, где наименее точный результат (LabVIEW) отличается от наиболее точного (Julia) в два раза.

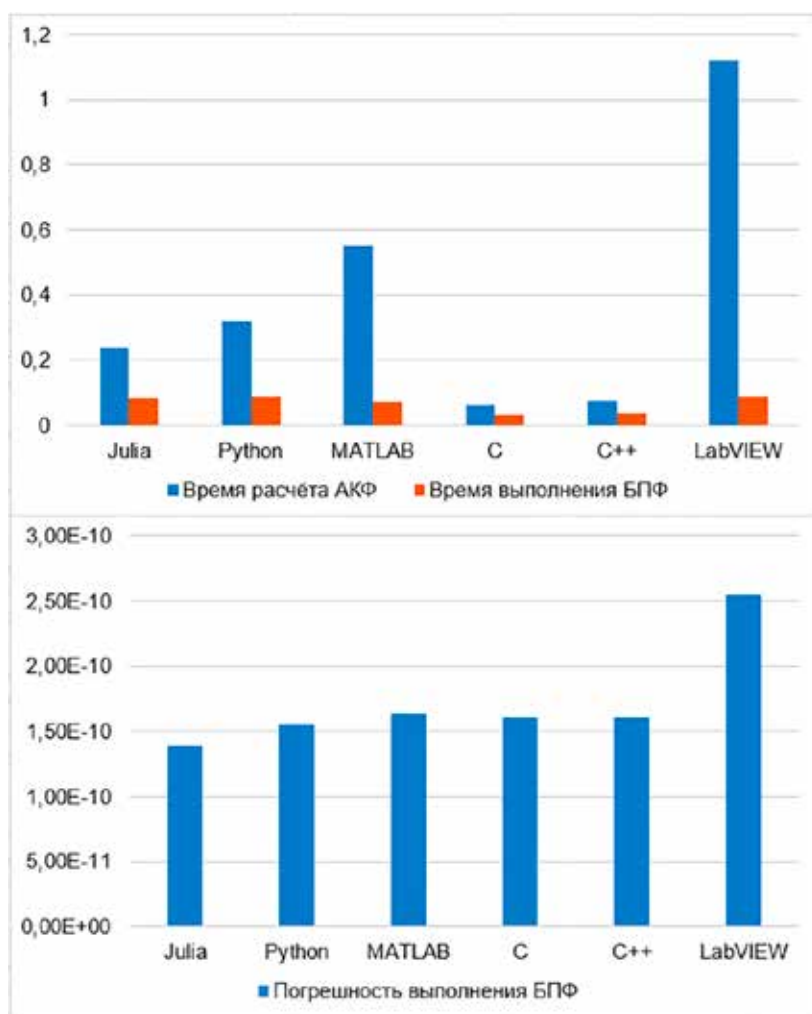


Рис. 2. Сравнение времени выполнения и погрешности алгоритмов АКФ и БПФ

Таблица 2

Исходный код вычисления и визуализация АКФ для МАТЛАВ, Python и Julia

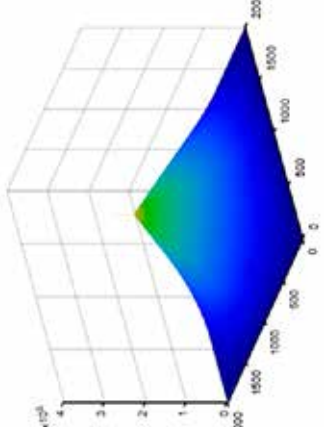
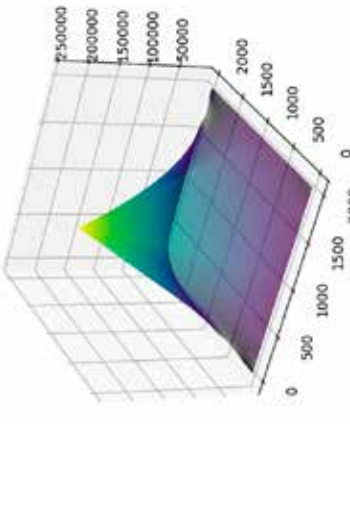
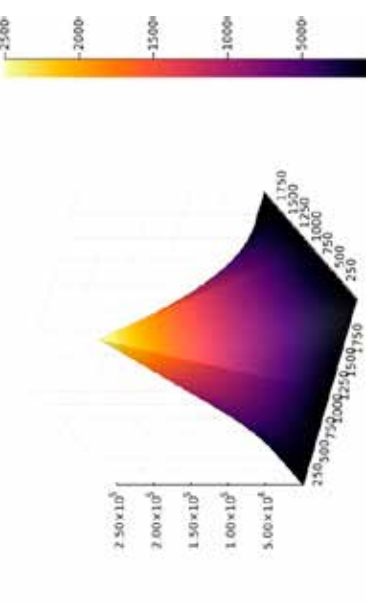
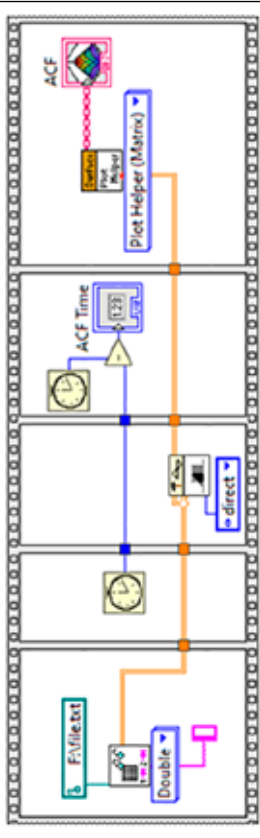
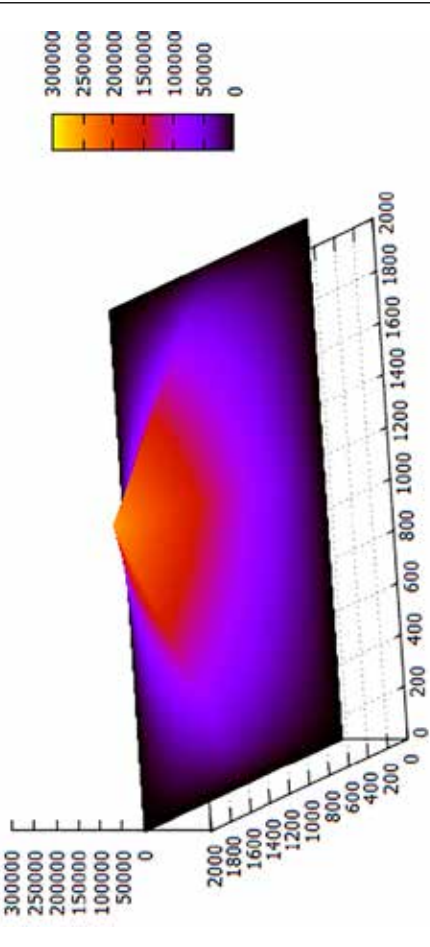
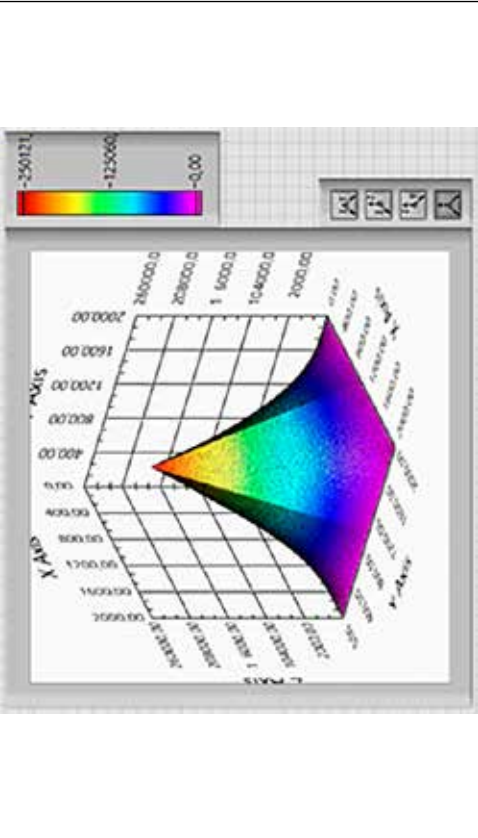
MATLAB	Python	Julia
<pre>import xcorr2_fft.* sig = readmatrix('file.txt'); tic acf = xcorr2_fft(sig, sig); toc figure; surf(acf, 'EdgeColor', 'none');</pre>	<pre>import numpy as np from scipy import signal import time as t import matplotlib.pyplot as plt with open('file.txt', 'r') as f: l = [[float(num) for num in line.split(' ')] for line in f] sig = np.array(l) st_t = t.perf_counter() acf = signal.convolve(sig, sig, mode='full', method='fft') end_t = t.perf_counter() print("ACF calculation time:", end_t - st_t) x = np.linspace(1, len(acf), len(acf)) y = np.linspace(1, len(acf), len(acf)) X_corr, Y_corr = np.meshgrid(x, y) ax = plt.axes(projection='3d') ax.plot_surface(X_corr, Y_corr, acf, rstride=1, cstride=1, cmap='viridis') plt.show()</pre>	<pre>using DelimitedFiles using Plots using DSP sig = Array{Float64, 2}(undef, 1000, 1000) sig = readlm("file.txt", '\Float64', '\n') acf_time = @elapsed begin acf = conv(sig, sig) end println("Время расчёта АКФ:", acf_time) x_corr = Y_corr = 1:1:size(acf,1) surface(x_corr, Y_corr, acf)</pre>
		

Таблица 3

Исходный код вычисления и визуализация АКФ для C/C++ и LabVIEW

C/C++	LabVIEW
<pre> begin = clock(); plan = fftw_plan_dft_r2c_2d(sig_rows, sig_cols, (double*)sig, (fftw_complex*)fdata, FFTW_ESTIMATE); fftw_execute(plan); fftw_destroy_plan(plan); CompMulR(fdata); //do a complex multiply plan2 = fftw_plan_dft_c2r_2d(sig_rows*2, sig_cols*2, (fftw_complex*)fdata, (double*)acf, FFTW_ESTIMATE); fftw_execute(plan2); fftw_destroy_plan(plan2); end = clock(); time_spent = (double)(end - begin) / CLOCKS_PER_SEC; printf(«ACF calculation time:%f seconds\n», time_spent); </pre>	
	

Заключение

В данной статье были рассмотрены современные средства имитационного моделирования сигналов.

Для реализации системы имитационного моделирования многомерных сигналов в системах ИИ с поддержкой реального времени предпочтительнее использовать вычислительные устройства специального назначения. Кроме того, прием и формирование физических сигналов требует использования дополнительной аппаратуры, например ММП.

Выбор программного средства реализации имитационной модели нельзя определить однозначно. В том случае, если целью является разработка имитационной модели реального времени с жесткими требованиями по времени выполнения вычислений, из рассматриваемых программных средств наиболее подходящими являются языки программирования C и C++. Программное средство LabVIEW RT может стать эффективным средством прототипирования имитационной модели на базе ММП, так как обеспечивает значительно более «низкий порог входа» разработчика по сравнению с языками C и C++, в том числе обеспечивает поддержку драйверов большой номенклатуры готовых аппаратных модулей и библиотеки программных функций. Такие программные средства, как Python, Julia и MATLAB, с учетом поддержки их библиотеками большого количества необхо-

димых для имитационного моделирования многомерных сигналов функций, но в силу большого времени выполнения операций могут эффективно применяться на стадии прототипирования имитационной модели, разработки алгоритмического и математического обеспечения в тех случаях, когда допустимо пренебрежение ограничениями режима реального времени.

Исследование выполнено за счет гранта Российского научного фонда № 22-11-00049, <https://rscf.ru/project/22-11-00049/>.

Список литературы

1. Эльберг М.С., Цыганков Н.С. Имитационное моделирование: учебное пособие. Красноярск: Издательство СФУ, 2017. 128 с.
2. Строгалев В.П., Толкачева И.О. Имитационное моделирование: учебное пособие. М.: Издательство МГТУ им. Н.Э. Баумана, 2013. 280 с.: ил.
3. Гуренко В.В., Климов С.М., Пролетарский А.В., Смирнова Е.В., Сотников А.А., Сюзев В.В. Методы имитации сигналов в научных задачах моделирования информационно-управляющих систем реального времени: монография / кол. авторов; под ред. В.В. Сюзева. М.: РУСАЙНС, 2021. 326 с.
4. Быков В.В. Цифровое моделирование в статистической радиотехнике. М.: Советское радио, 1971. 328 с.
5. Борисов Ю.П., Цветнов В.В. Математическое моделирование радиотехнических систем и устройств. М.: Радио и связь, 1985. 176 с., ил. (Библиотека радиоинженера «Современная радиоэлектроника»).
6. Сараджишвили С.Э. Цифровая обработка многомерных сигналов: учебное пособие. СПб.: СПбПУ. 2014. 125 с.
7. Сюзев В.В., Смирнова Е.В., Пролетарский А.В. Алгоритмы многомерного имитационного моделирования случайных процессов // Компьютерная оптика. 2021. Т. 45. № 4. С. 627–637.