

20. Höppner F. Discovery of temporal patterns – learning rules about the qualitative behavior of time series, In: *Proc. of the 5th European Conference on Principles and Practice of Knowledge Discovery in Databases, Lecture Notes in Artificial Intelligence 2168*, Springer, 2001, pp. 192-203.
21. Ultsch A. Unification Based Temporal Grammar. In: Technical Report No. 37, Philipps-University Marburg, Germany, 2004.
22. Mörchen F., Ultsch A. Mining Hierarchical Temporal Patterns in Multivariate Time Series. Available at: <http://citeseerx.ist.psu.edu/viewdoc/download>.
23. Suchkova L.I., Starikov E.S. К вопросу о методе выявления закономерностей в данных мониторинга [To the question about the method of identifying patterns in data monitoring], *Materialy 14 Mezhdunarodnoy konferentsii «Izmerenie, kontrol', informatizatsiya»* [Proceedings of the 14 International conference "Measurement, control, information"]. Barnaul: AltGTU, 2013, pp. 155-157.
24. Starikov E.S. Formalizatsiya opisaniya zakonomernostey v gruppakh vremennykh ryadov [Formalization of the description of regularities in groups of time series], *Sbornik po rezul'tatam XXI zaachnoy nauchnoy konferentsii «Mezhdunarodnyy nauchno-issledovatel'skiy zhurnal»* [Proceedings of the XXI according to the results of the correspondence scientific conference "international research journal"]. Ekaterinburg, 2013, No. 11 (18), Part 1, pp. 131-132.
25. Starikov E.S., Suchkova L.I. Modification universal temporal grammar for the description of the legitimacies in data, *Academic science-problems and achievements X vol. 1 spc Academic*. USA, North Charleston, 2016, pp. 63-65.

Статью рекомендовал к опубликованию д.т.н, профессор С.В. Соколов.

Стариков Егор Сергеевич – ФГБОУ ВО «Алтайский государственный технический университет им. И.И. Ползунова»; e-mail: yegor.inc@live.ru; 656038, г. Барнаул, проспект Ленина, 46; тел.: +79619902171; аспирант.

Сучкова Лариса Иннокентьевна – e-mail: lis@agtu.secna.ru; тел.: +73852290786; кафедра информатики, вычислительной техники и информационной безопасности; д.т.н.; профессор.

Starikov Yegor Sergeevich – Altay State Technical University; e-mail: yegor.inc@live.ru; 46, Lenin prospekt, Barnaul, 656065, Russia; phone: +79619902171; postgraduate student.

Suchkova Larisa Innokentievna – e-mail: lis@agtu.secna.ru; phone: +73852290786; the department of informatics, computer facilities and information security; dr. of eng. sc.; professor.

УДК 004.4'42

DOI 10.23683/2311-3103-2017-6-212-223

М.Ю. Поленов, В.С. Лапшин, С.М. Гушанский

РЕАЛИЗАЦИЯ МОДУЛЯ КОНВЕРТАЦИИ МОДЕЛЕЙ ДЛЯ СРЕДЫ MATLAB*

Рассматривается реализация модуля конвертации программных моделей, представленных в формате среды Matlab и на языке программирования C++. Данный модуль и его дополнительные компоненты разработаны на кафедре вычислительной техники Инженерно-технологической академии Южного федерального университета (ЮФУ). На сегодняшний день весьма актуальной является проблема организации взаимодействия программных комплексов моделирования, в которых исследователи ведут свои разработки по созданию программных моделей различных сложных технических систем. Одним из самых востребованных комплексов является среда разработки Matlab. Также, практически каждая современная среда моделирования поддерживает программные модели, реализованные на языке программирования C++. Таким образом, для организации наиболее эффективного процесса разработки программных моделей различных технических систем, исследователям необходимы инструменты, которые

* Работа выполнена в рамках проектной части госзадания Минобрнауки России № 2.3928.2017/4.6 в Южном федеральном университете.

позволят быстро и эффективно преобразовывать самые популярные форматы как моделей отдельных компонент, так и всей модели системы. В данной работе рассматривается разработка такого инструментария для конвертации моделей на базе программного комплекса «Мультитранслятор», который был создан ранее на кафедре вычислительной техники ЮФУ. Конвертер представляет собой два взаимодействующих трансляционных модуля. Данный принцип организации конвертирования кодов моделей позволяет эффективно модернизировать разработанные средства, позволяя дополнять или изменять трансляционные модули. Также проанализированы возможные тупиковые ситуации, возникающие в ходе трансляции программного кода моделей и выработаны механизмы их преодоления. В итоге, произведены эксперименты по конвертации различных программных моделей в ходе которых была произведена оценка работоспособности разработанного модуля.

Модель; конвертация моделей; Мультитранслятор; трансляционный модуль.

M.Yu. Polenov, V.S. Lapshin, S.M. Gushanskiy

IMPLEMENTATION OF MODELS CONVERSION MODULE FOR MATLAB

In this work the implementation of program models' converting module for models in the Matlab format and in C ++ programming language is considered. This module and its additional components are developed at the Department of Computer Engineering in Engineering and Technological Academy of the Southern Federal University (SFedU). Today, the problem of the interaction organization for the simulation program complexes where researchers develop program models of different complex technical systems is very urgent. One of the most popular complex is Matlab simulation environment. Also, practically each modern simulation environment supports the program models implemented in C++ language. Thus, for the organization of the most effective development process of program models of different technical systems the tools are necessary for developers, which will allow to transform quickly and effectively the most popular formats of separate components of models and whole system model. In this paper the development of such converting tools on the basis of the Multitranslator complex, which was developed earlier at Department of Computer Engineering, is described. The converter consists of two interacting translation modules. This principle of the conversion organization allows upgrading effectively the developed tools by adding or changing translation modules. The possible impasses arising during program code translation are also analyzed and mechanisms of their overcoming developed. As a result, experiments on converting different program models for the assessment of operability of the developed program model have been made.

Model; models' conversion; Multitranslator; translation module.

Введение. Для реализации моделирования сложных технических систем в настоящее время применяется множество инструментальных средств. Некоторые из этих средств имеют большую популярность и широко используются исследователями, другие появились совсем недавно и еще не достаточно распространены. Некоторая часть пакетов являются универсальными и используется для моделирования технических, и не только технических, систем. Другие имеют узкую специализацию в какой-либо предметной области. Возможности многих пакетов в значительной степени аналогичны и подходы к решению одних и тех же задач моделирования у них достаточно схожи [1].

Процесс создания моделей систем в различных пакетах моделирования в значительной степени основан на использовании существующих библиотек моделей [1, 2]. Такие библиотеки обычно содержат достаточно большое число элементов различной сложности. Однако не редко возникают ситуации, когда необходимая модель не реализована в рамках выбранного пакета среды виртуального моделирования. Решением данной проблемы может стать:

- ◆ создание новой модели с ее последующим подключением к библиотеке среды моделирования;
- ◆ организация трансляции и импорта внешней модели, реализованной на одном из языков моделирования или программирования, не поддерживаемым выбранной средой.

Поскольку, освоение определенного профессионального пакета моделирования связано со значительными временными затратами, очевидно, что для большинства пользователей выбор второго варианта является более предпочтительным, поскольку он лишает необходимости повторения ранее выполненных исследований в процессе построения модели и ее отладки [2–10].

Из всего многообразия существующих инструментальных средств, позволяющих проводить моделирование технических систем в различных доменах, можно выделить среду Matlab [11–13], которая является одной из самых популярных и программируемая в которой происходит на одноименном языке Matlab [14].

Этот программный комплекс активно пользуется большим числом исследователей, разработчиков систем, инженеров и математиков. Командный язык Matlab имеет весьма простой синтаксис, сравнительно прозрачен и позволяет сконцентрировать внимание на поставленной задаче, а не затрачивать много времени на написание и отладку различных функций. Это привело к тому, что в среде Matlab уже реализовано множество различных алгоритмов для решения различных задач. Но одним из недостатков данной среды является то, что язык Matlab является интерпретируемым, т.е. при выполнении кода, написанного на этом языке, каждая команда сначала проверяется на наличие ошибок, расшифровывается, и лишь затем ядро программного комплекса выполняет соответствующие действия. Это требует большого количества ресурсов компьютера и приводит к значительному замедлению при выполнении программы модели. Такой принцип реализации объясняется тем, что среда Matlab разрабатывалась первую очередь для удобного и эффективного использования исследователями, и не ориентировалась на высокую производительность. То, что язык Matlab является интерпретируемым, позволяет производить вычисления и сразу анализировать получаемые результаты, но с другой стороны, это заметно снижает производительность системы [14].

Тем не менее, очень часто возникают ситуации, когда реализованные и отлаженные в среде Matlab инженерные решения (модели) необходимо использовать в проектах [15], где высокая производительность и ресурсоемкость являются необходимыми и ключевыми условиями. Поскольку язык C++ является широко распространенным, существуют его реализации практически на всех платформах, поэтому, самым эффективным решением этой проблемы является перевод программы с языка Matlab на язык C++ и создание инструментального программного модуля осуществляющего такой перевод. Далее рассматриваются подход к организации такого модуля и особенности его программной реализации.

Разработка конвертора моделей для среды Matlab. Перед рассмотрением разработанного модуля-конвертора кодов моделей необходимо отметить, что среда Matlab имеет в своем составе аналогичные утилиты для трансляции кода моделей в наиболее распространенные языки программирования высокого уровня (C++, Java, C# и др.). Одной из таких утилит является продукт Matlab Compiler от Matlab. Он переводит код с языка Matlab в код на языке C++, а также имеет удобный интерфейс, встроенный в собственный интерфейс Matlab. Но, имеется ряд причин, из-за которых эффективно использовать полученный оттранслированный код не представляется возможным. Первой из таких причин является то, что получаемый после трансляции код, содержит множество обращений к библиотекам Matlab [15–18]. По этой причине, дальнейшая модернизация и сопровождение полученной модели в сторонней среде моделирования весьма затруднительна. Также стоит отметить, что код полученный с помощью утилиты Matlab Compiler предназначен только лишь для последовательного исполнения, что существенно уменьшает производительность.

Еще одним аналогом разработанного модуля можно считать программный комплекс Semi|Works. Данный интерпретатор полностью написан на платформе .NET, а также имеет свой собственный парсер. Но, хотя возможностей данного комплекса хватает для частичного решения поставленной проблемы, его можно назвать скорее демонстрационным, так как он реализует лишь командную строку Matlab, то есть может интерпретировать только простые выражения. Возможности интерпретации моделей систем нет.

Принимая во внимание указанные выше аспекты была поставлена задача разработки конвертера для среды Matlab, который будет включать в себя два трансляционных модуля и позволит преобразовывать программный код, написанный на командном языке M-code, в код на языке C++ и обратно.

Рассмотрим вначале саму постановку задачи. Как известно, транслятор должен трансформировать строки написанной на входном языке L исходной программы SP(L) в строки генерируемой программы OP, представленной на выходном языке R:

$$SP(L) \rightarrow OP(R). \quad (1)$$

В связи с этим функцию трансляции программ, написанных на языке L в эквивалентные программы на языке R, можно определить как:

$$OP(R) := T_{LR}[SP(L)]. \quad (2)$$

Если система моделирования должна работать с несколькими языками, например, такими как M-язык, Modelica, XML, C++ и др., то при классическом подходе ее подсистема импорта моделей должна содержать набор соответствующих трансляторов, работающих в соответствии с выражением (2). Иными словами, такая система описывается соотношением:

$$OP(R) := T_{LjR}[SP(L_j)],$$

где T_{LjR} – функция преобразования (транслятор) с входного языка L_j , $L_j \in \{L_1, L_2, \dots, L_N\}$ на выходной язык R.

Традиционный подход к разработке трансляторов состоит в том, что функции преобразования T_{LjR} разрабатываются и оптимизируются для каждого конкретного входного языка L_j отдельно:

$$\begin{aligned} OP(R) &:= T_{L1R}[SP(L_1)], \\ OP(R) &:= T_{L2R}[SP(L_2)], \\ OP(R) &:= T_{LN R}[SP(L_N)]. \end{aligned} \quad (3)$$

При этом критерием оптимизации, как правило, служит время трансляции и объем оперативной памяти, занимаемый компилирующей программой.

Однако при таком подходе любые, даже незначительные изменения в языке программирования приводят к необходимости разработки нового транслятора. В случае систем моделирования, при использовании которых пользователю зачастую нужен не весь набор операторов некоторого языка, а всего лишь его подмножество, причем изменяемое самим пользователем, нужен более гибкий метод трансляции, позволяющий оперативно осуществлять требуемые изменения функции трансляции T_{LR} .

В качестве такого гибкого метода трансляции в данной работе предлагается использовать развиваемый авторами метод многоязыковой трансляции, названный мультитрансляцией [1, 2, 14]. Идея мультитрансляции основана на том, что в отличие от традиционного подхода, в соответствии с которым независимо от специфики языка любой транслятор можно считать функциональным преобразователем, построенным на основе системы продукций $P(L, R)$:

$P(L, R) = \{S_i^L, A_i^R\}$, $i=1..k$, где S_i^L – левая (ситуационная) часть отдельно взятого правила i из системы продукций P для разбора текста на языке L; A_i^R – его правая (действенная) часть, выполняющая синтез программы на языке R; k – количество правил в системе продукций. Поэтому сам процесс трансляции формально можно определить и так:

$$\begin{aligned} OP(R) &= T_{P(L,R)}[SP(L)], \\ P(L,R) &= \{S_i^L, A_i^R\}, i = 1..k. \end{aligned} \quad (4)$$

С этой точки зрения классический подход к разработке трансляторов состоит в том, что соответствующая языкам L и R система продукционных правил формируется и оптимизируется совместно с системами грамматического разбора (лексического и синтаксического анализа). В отличие от этого метод многоязыковой трансляции основан на отделении процедур синтеза требуемой системы продукций от системы грамматического разбора. Более того, эта система строится инвариантной относительно наборов продукционных правил [1]. На основе данных принципов была реализована программная среда – Мультитранслятор [8], которая и была выбрана в качестве средств реализации разрабатываемого конвертера на основе проведенного анализа средств автоматизации построения трансляторов.

В данной программной среде, общая структура трансляционного модуля представляется набором глобальных объектов, набором правил и действий, набором процедур и функций, а также главной структурой трансляционного модуля процедурой main. В процедуре main описывается инициализация входных и выходных файлов для перевода, инициализация объектов переменных, структур, настройка параметров грамматического разбора, подготовка входного текста для разбора, запуск грамматического разбора.

На данный момент завершена разработка модуля, представляющего собой конвертер с языка C++ на язык Matlab, схема перевода моделей с его помощью представлена на рис. 1.

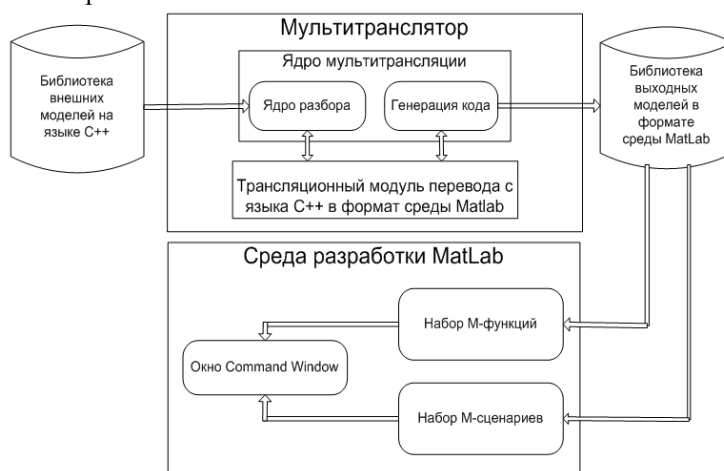


Рис. 1. Схема перевода моделей

Как видно из представленной схемы, исходная программная модель подвергается разбору в ядре Мультитранслятора на основе разработанного трансляционного модуля [15–16]. Затем, встроенный в Мультитранслятор генератор кода формирует программную модель, оформленную в формате среды Matlab, которая помещается в библиотеку выходных моделей. Далее выходная модель, представленная набором M-функций и M-сценариев, загружается в среду разработки Matlab, где в дальнейшем, возможно, будет запускаться или, если это необходимо, модернизироваться.

Разработка второго модуля, предполагает реализацию транслятора перевода программного кода из языка Matlab на язык C++. Таким образом, два разработанных трансляционных модуля будут образовывать конвертер, позволяющий пользователю динамично конвертировать программный код и использовать полученные результаты в различных средах разработки программного обеспечения.

Перед началом реализации модуля был проведен анализ основных проблем возникающих при трансляции программных кодов с языка среды Matlab [17]. Первой такой проблемой является различный принцип адресации в массивах. Нумерация массива в Matlab начинается с единицы, а в C++ с нуля. Для решения этой проблемы планируется в ходе трансляции менять индекс массива. Чтобы выходной код оставался внешне схожим с исходным кодом транслируемой программы, аналогичным образом будет изменяться и индекс цикла. Также, существенные различия присутствуют в синтаксисе вызова функций.

Стандартный вызов функции на языке Matlab выглядит следующим образом:

```
[<out1 >, <out2 >...] = <name >(<in1 >, <in2 >, ...)
```

где inX означают входные параметры, и outX – результаты функции [16]. При такой схеме вызова функции в Matlab, возвращающей несколько значений, возникает проблема при прямой трансляции этой структуры в формат языка C++, так как функции в C++ могут возвращать не более одного значения. Одним из возможных вариантов решения данной проблемы может стать использование передачи параметров по ссылке. Функции на языке Matlab вида:

```
[<out1 >, <out2 >, ...] = <name >(<in1 >, <in2 >, ...)
```

будут конвертироваться в функции на C++ вида:

```
<name >(<in1 >, <in2 >, ... , &<out1 >, &<out2 >, ...);
```

где &<out1 > и &<out2 > являются аналогами выходных параметров функции в Matlab.

Принцип определения глобальных переменных в Matlab также отличается от C++. Интерпретатор языка Matlab все переменные, указанные без ключевого слова global, считает локальными, даже если они не встречаются в рамках какой-либо функции. Для указания переменных, которые должны сохранять свое значение между всеми функциями и всеми файлами используется ключевое слово global. Для того чтобы такие переменные можно было использовать без особых трудозатрат и без сильных изменений кода на языке C++, в процессе трансляции будет генерироваться файл, в котором будут храниться все переменные, глобальные по логике исполнения программы [21]. Этот файл будет подключаться в файле, содержащем функцию main(), до подключения остальных файлов, которые были исходно написаны на языке Matlab. Таким образом, эти переменные становятся доступными во всех функциях и исполняют ту же роль, что исполняли глобальные переменные в исходной программе.

Язык Matlab является интерпретируемым [21], что позволяет использовать некоторые преимущества, такие, например, как динамическая типизация. Это позволяет никогда не указывать тип переменной при ее создании – интерпретатор вычисляет тип для результата выражения и приписывает переменной именно этот тип. Таким образом, тип переменной может меняться по ходу программы, что невозможно в программах на языке C++. Для того чтобы корректно обрабатывать такие ситуации, было решено в процессе трансляции генерировать временные переменные с аналогичными именами.

В итоге, основываясь на различиях в особенностях построения программных моделей на языке Matlab и C++, рассмотренных выше, были разработаны методы и алгоритмы, позволяющие выполнять корректную трансляцию программных кодов моделей, которые и представляют научную новизну предложенного и реализованного в данной работе подхода к организации трансляции моделей.

Экспериментальная проверка синтезированного трансляционного модуля. Для проверки работоспособности и оценки функционирования синтезированного трансляционного модуля перевода с языка C++ в формат среды Matlab, были проведены эксперименты по трансляции исходных C++ программ в программы моделей на языке M-code, представляющие собой набор m-файлов. Оценка резуль-

татов работы сгенерированных m-файлов осуществлялась в среде Matlab. В качестве примера за основу был выбран алгоритм Дейкстры [19] для нахождения кратчайших путей из одной изначально заданной вершины графа до всех остальных.

В результате трансляции исходной программы представленной на языке C++ при помощи трансляционного модуля Мультитранслятора, было сгенерировано два m-файла:

- ◆ М-функция, реализующая непосредственно алгоритм Дейкстры, имеет имя «Dijkstra» и содержит в себе реализацию данной функции для входного файла в формате выходной среды;
- ◆ М-сценарий, в котором реализуется М-функция с необходимыми начальными значениями.

М-функция имеет вид:

```
function Dijkstra(GR , st)
V = 6;
distance=[];
int64 count;
int64 index;
int64 i;
int64 u;
m=st;
visited = [];
for i=1:1:V
    distance (i) = intmax;
    visited(i) =false;
end
distance(st) = 0 ;
for count=1:1:V
    min=intmax;
    for i=1:1:V
        if ( visited(i) == false && distance(i) <= min )
            min = distance(i);
            index = i;
        end
        u = index ;
        visited(u) = true;
    end
    for i=1:1:V
        if (visited(i) == false && GR(u,i) && distance(u) ~=
intmax && distance(u) + GR(u,i) < distance(i))
            distance(i)=distance(u)+ GR(u,i);
        end
    end
end
disp ('Стоимость пути из начальной вершины до остальных:\t\n');
for i=1:1:V
    if (distance(i) ~= intmax)
        fprintf ('%f > %f = %f \n ', m , i , distance(i) );
    else
        fprintf ( 'маршрут не найден\n' );
    end
end
end
```

Для оценки корректности трансляции приведем сравнение результатов моделирования исходной модели с результатами оттранслированной модели. Результаты моделирования приведены ниже на рис.2 и рис. 3.

Как видно из рисунков, результаты работы программ совпали. Это свидетельствует о том, что трансляция кода прошла корректно.

```

C:\Users\Иван\Рабочий стол\Иван\Иван.exe
Начальная вершина >> 1
Стоимость пути из начальной вершины до остальных:
1.000000 > 1.000000 = 0.000000
1.000000 > 2.000000 = 1.000000
1.000000 > 3.000000 = 4.000000
1.000000 > 4.000000 = 10.000000
1.000000 > 5.000000 = 2.000000
1.000000 > 6.000000 = 10.000000
>>
    
```

Рис. 2. Результаты работы программы на языке C++

```

Начальная вершина
1
Стоимость пути из начальной вершины до остальных:\t\n
1.000000 > 1.000000 = 0.000000
1.000000 > 2.000000 = 1.000000
1.000000 > 3.000000 = 4.000000
1.000000 > 4.000000 = 10.000000
1.000000 > 5.000000 = 2.000000
1.000000 > 6.000000 = 10.000000
>>
    
```

Рис. 3. Результаты работы программы на языке Matlab

С целью промежуточной проверки и оценки функционирования разрабатываемого трансляционного модуля перевода моделей с языка среды Matlab в формат моделей на языке C++ [18], были также проведены эксперименты по трансляции исходных моделей среды Matlab в подпрограммы на языке C++. Тестирование сгенерированных сpp-файлов проводилось в среде Microsoft Visual Studio.

В качестве тестового примера был использован алгоритм скользящего среднего [20]. Скользящее среднее – общее название для семейства функций, значения которых в каждой точке определения равны среднему значению исходной функции за предыдущий период. Скользящие средние обычно используются с данными временных рядов для сглаживания краткосрочных колебаний и выделения основных тенденций или циклов. Математически скользящее среднее является одним из видов свертки и поэтому его можно рассматривать как фильтр низких частот, используемых в обработке сигналов.

В качестве исходной была взята программная реализация данного алгоритма в среде Matlab. В результате трансляции этой программы средствами разрабатываемого трансляционного модуля был сгенерирован следующий код:

```

void moving_average(double * input, double *&result, int size,
int periods)
{
    double sum = 0;
    for (int i = 0; i < size; i++)
    {
        if (i < periods) {
            sum += input[i];
            result[i] = (i == periods - 1) ? sum/(double)periods: 0;
        } else {
            sum = sum - input [i - periods] + input [i];
            result[i] = sum / (double)periods;
        }
    }
    return
    
```



```
}  
void main()  
{double *values = new double[100]; // the input  
double *averages = new double[100]; // the output  
values[0] = 55;  
values[1] = 113;  
values[2] = 92.6;  
...  
values[9] = 23;  
moving_average(values, averages, 100, 5);  
}
```

Для проверки работоспособности полученного кода, с помощью высокоуровневой библиотеки визуализации пользовательских данных DISLIN [22], было произведено моделирование полученного результата. Результаты моделирования представлены на рис. 4.

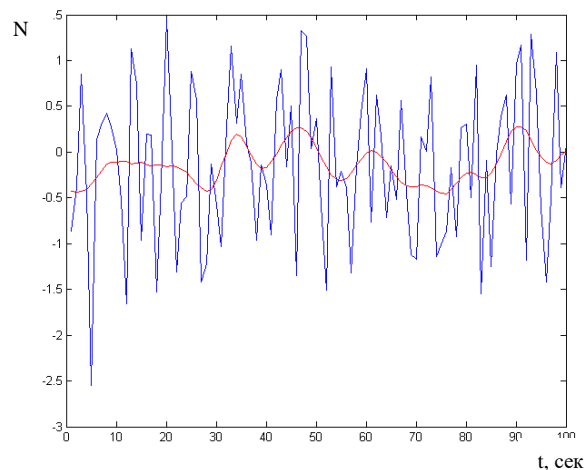


Рис. 4. Результаты работы оттранслированной модели на языке C++

Как видно из рис. 4, при изменении управляющего параметра, изменяется степень сглаженности исходного сигнала, что свидетельствует о корректной работе сгенерированной модели, а значит и корректной работе разрабатываемого трансляционного модуля для конвертера моделей.

Заключение. Следует отметить, что существующие на сегодняшний день механизмы преобразования форматов моделей не предоставляют возможности эффективного использования оттранслированных моделей в сторонних средах разработки. Описанный в данной работе подход и созданный конвертер позволят расширить сферу применения разработанных проектов в системе Matlab и упростить пользователям разработку программных моделей на языке C++. За счет своей модульной организации, данный конвертер может легко модернизироваться и преобразовываться с расширением своего функционала.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Чернухин Ю.В., Гузик В.Ф., Поленов М.Ю. Многоязыковая трансляция средств виртуального моделирования. – Ростов-на-Дону: ЮНЦ РАН, 2009. – 368 с.
2. Mens T., Van Gorp P. A taxonomy of model transformation // Electronic Notes in Theoretical Computer Science, Netherlands. – 2006. – Vol. 152. – P. 125-142.
3. Breunese A.P.J., Top J.L., Broenink J.F., Akkermans J.M. Libraries of Reusable Models: Theory and Application // Simulation. – 1998. – Vol. 71. – P. 7-22.

4. Чернухин Ю.В., Гузик В.Ф., Поленов М.Ю. Подход к формированию внешних библиотек сред виртуального моделирования на базе мультязыковой трансляции // Вестник компьютерных и информационных технологий. – 2008. – № 10. – С. 2-12.
5. Robinson S., Nance R.E., Paul R.J., et al. Simulation model reuse: definitions, benefits and obstacles // Simulation Modelling Practice and Theory. – 2004. – No. 12. – P. 479-494.
6. Поленов М.Ю. Организация распределенных инструментальных средств поддержки многократно используемых моделей // Известия ЮФУ. Технические науки. – 2013. – № 7 (144). – С. 201-207.
7. Chernukhin Y., Polenov M., Vemulapally C., Solodovnik E., Mantooth A., Dougal R. Deploying Modelica Models into Multiple Simulation Environments // Proceedings of IEEE International Behavioral Modeling and Simulation Conference (BMAS 2005), IEEE, 2005. – P. 134-139.
8. Chernukhin Yu., Guzik V., Polenov M. Multilanguage Translation Usage in Toolkit of Modeling Systems // WIT Transactions on Information and Communication Technologies. – 2014. – Vol. 58, VOL. 1. – P. 397-404.
9. Поленов М.Ю., Гушанский С.М., Николава А.Ю., Курмалеев А.О. Организация взаимодействия систем моделирования с распределенной библиотекой моделей // Информатизация и связь. – 2015. – № 3. – С. 78-82.
10. Polenov M., Guzik V., Gushanskiy S., Kurmaleev A. Development of the Translation Tools for Distributed Storage of Models // Proceedings of 9th IEEE International Conference on Application of Information and Communication Technologies (AICT 2015). – 2015. – P. 30-34.
11. MATLAB. MathWorks, Inc. – URL: www.mathworks.com (дата обращения: 15.03.2017).
12. Simulink. ЦИТМ Экспонента. – URL: <http://matlab.ru/products/simulink> (дата обращения: 20.03.2017).
13. Смоленцев Н.К. Создание Windows-приложений с использованием математических процедур MATLAB. – М.: ДМК-Пресс, 2008. – 456 с.
14. Богословский Н.А., Климов Ю.А., Савельев А.В., Шалыга Д.К. Разработка экспериментального комплекса суперкомпьютерного моделирования на основе кода на языке Matlab // Программные системы: теория и приложения: электронный научный журнал. – 2013. – Т. 40, № 4. – С. 21-42.
15. Joisha P., Kanhere A., Shenoy N. The Design and Implementation of a Parser and Scanner for the MATLAB Language in the MATCH Compiler. 1999. – URL: <http://www.ece.northwestern.edu/csrc/pjoisha/Publications/CPDC-TR-9909-017.pdf> (дата обращения: 27.03.2017).
16. Поленов М.Ю., Лапшин В.С. Трансляционный модуль перевода моделей с языка Си в формат среды MatLab // Информационные технологии, системный анализ и управление (ИТСАиУ-2015): Сборник трудов XIII Всероссийской научной конференции молодых ученых, аспирантов и студентов. – Ростов-на-Дону: Изд-во ЮФУ, 2016. – Т. 3. – С. 89-93.
17. Поленов М.Ю., Лапшин В.С. Организация трансляции внешних моделей для среды Matlab // Сборник трудов II Всероссийской научно-технической конференции «Фундаментальные и прикладные аспекты компьютерных технологий и информационной безопасности». – Ростов-на-Дону: Изд-во ЮФУ, 2016. – Т. 3. – С. 69-72.
18. Лапшин В.С., Поленов М.Ю., Гузик В.Ф. Организация трансляционного модуля перевода моделей среды Matlab на язык C++ // Информационные технологии, системный анализ и управление (ИТСАиУ-2016): Сборник трудов XIV Всероссийской научной конференции молодых ученых, аспирантов и студентов. – Ростов-на-Дону: Изд-во ЮФУ, 2016. – Т. 2. – С. 309-312.
19. Алгоритм Дейкстры. Академия Microsoft: Структуры и алгоритмы компьютерной обработки данных. НОУ "Интуит". – URL: <http://www.intuit.ru/studies/courses/648/504/lecture/11475> (дата обращения: 10.03.2017).
20. Арифметическое скользящее среднее. – URL: <https://prog-cpp.ru/moving-average/> (дата обращения: 20.03.2017).
21. The difference between a compiled and interpreted language. – URL: <http://www.programmerinterview.com/index.php/general-miscellaneous/whats-the-difference-between-a-compiled-and-an-interpreted-language/> (дата обращения: 20.03.2017).
22. The Data Plotting Software DISLIN. – URL: <http://www.mps.mpg.de/dislin/online-manual> (дата обращения: 20.03.2017).

REFERENCES

1. Chernukhin Yu.V., Guzik V.F., Polenov M.Yu. Mnogoyazykovaya translyatsiya sredstv virtual'nogo modelirovaniya [Multi-language stream of a virtual simulation]. Rostov-on-Don: YuNTs RAN, 2009, 368 p.
2. Mens T., Van Gorp P. A taxonomy of model transformation, *Electronic Notes in Theoretical Computer Science, Netherlands*, 2006, Vol. 152, pp. 125-142.
3. Breunese A.P.J., Top J.L., Broenink J.F., Akkermans J.M. Libraries of Reusable Models: Theory and Application, *Simulation*, 1998, Vol. 71, pp. 7-22.
4. Chernukhin Yu.V., Guzik V.F., Polenov M.Yu. Podkhod k formirovaniyu vneshnikh bibliotek sred virtual'nogo modelirovaniya na baze mul'tiyazykovoy translyatsii [Approach to the formation of the external library environments virtual simulation on the basis of multilingual translation], *Vestnik komp'yuternykh i informatsionnykh tekhnologiy* [Vestnik of Computer and Information Technology], 2008, No. 10, pp. 2-12.
5. Robinson S., Nance R.E., Paul R.J., et al. Simulation model reuse: definitions, benefits and obstacles, *Simulation Modelling Practice and Theory*, 2004, No. 12, pp. 479-494.
6. Polenov M.Yu. Organizatsiya raspredelennykh instrumental'nykh sredstv podderzhki mnogokratno ispol'zuemykh modeley [Organization of the distributed tools of reusable models support], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2013, No. 7 (144), pp. 201-207.
7. Chernukhin Y., Polenov M., Vemulapally C., Solodovnik E., Mantooth A., Dougal R. Deploying Medica Models into Multiple Simulation Environments, *Proceedings of IEEE International Behavioral Modeling and Simulation Conference (BMAS 2005)*, IEEE, 2005, pp. 134-139.
8. Chernukhin Yu., Guzik V., Polenov M. Multilanguage Translation Usage in Toolkit of Modeling Systems, *WIT Transactions on Information and Communication Technologies*, 2014, Vol. 58, VOL. 1, pp. 397-404.
9. Polenov M.Yu., Gushanskiy S.M., Nikolava A.Yu., Kurmaleev A.O. Organizatsiya vzaimodeystviya sistem modelirovaniya s raspredelennoy bibliotekoy modeley [The organization of interaction of modeling systems with distributed library models], *Informatizatsiya i svyaz'* [Informatization and Communication], 2015, No. 3, pp. 78-82.
10. Polenov M., Guzik V., Gushanskiy S., Kurmaleev A. Development of the Translation Tools for Distributed Storage of Models, *Proceedings of 9th IEEE International Conference on Application of Information and Communication Technologies (AICT 2015)*, 2015, pp. 30-34.
11. MATLAB. MathWorks, Inc. Available at: www.mathworks.com (accessed 15 March 2017).
12. Simulink. TsITM Eksponenta [Simulink. The CITMA of the Exhibitor]. Available at: <http://matlab.ru/products/simulink> (accessed 20 March 2017).
13. Smolentsev N.K. Sozdanie Windows-prilozheniy s ispol'zovaniem matematicheskikh protsedur MATLAB [Creating Windows-based applications using mathematical procedures MATLAB]. Moscow: DMK-Press, 2008, 456 p.
14. Bogoslovskiy N.A., Klimov Yu.A., Savel'ev A.V., Shalyga D.K. Razrabotka eksperimental'nogo kompleksa superkomp'yuternogo modelirovaniya na osnove koda na yazyke Matlab [Development of the experimental complex supercomputer-based simulation code in Matlab], *Programmye sistemy: teoriya i prilozheniya: elektronnyy nauchnyy zhurnal* [Software systems: theory and applications: electronic scientific journal], 2013, Vol. 40, No. 4, pp. 21-42.
15. Joisha P., Kanhere A., Shenoy N. The Design and Implementation of a Parser and Scanner for the MATLAB Language in the MATCH Compiler. 1999. Available at: <http://www.ece.northwestern.edu/cpdc/pjoisha/Publications/CPDC-TR-9909-017.pdf> (accessed 27 March 2017).
16. Polenov M.Yu., Lapshin V.S. Translyatsionnyy modul' perevoda modeley s yazyka Si v format sredey MatLab [Translation module the translation models of the C language in MatLab format], *Informatsionnye tekhnologii, sistemnyy analiz i upravlenie (ITSAiU-2015): Sbornik trudov XIII Vserossiyskoy nauchnoy konferentsii molodykh uchenykh, aspirantov i studentov* [Information technology, system analysis and management (Idayu-2015): proceedings of the XIII all-Russian scientific conference of young scientists, postgraduates and students]. Rostov-on-Don: Izd-vo YuFU, 2016, Vol. 3, pp. 89-93.

17. *Polenov M.Yu., Lapshin V.S.* Organizatsiya translyatsii vneshnikh modeley dlya sredy Matlab [Organization of external translation models for Matlab], *Sbornik trudov II Vserossiyskoy nauchno-tekhnicheskoy konferentsii «Fundamental'nye i prikladnye aspekty komp'yuternykh tekhnologiy i informatsionnoy bezopasnosti»* [Proceedings of the II all-Russian scientific-technical conference "Fundamental and applied aspects of computer technologies and information security"]. Rostov-on-Don: Izd-vo YuFU, 2016, Vol. 3, pp. 69-72.
18. *Lapshin V.S., Polenov M.Yu., Guzik V.F.* Organizatsiya translyatsionnogo modulya perevoda modeley sredy Matlab na yazyk C++ [Organization of translation module of translation models from Matlab to C++], *Informatsionnye tekhnologii, sistemy analiz i upravlenie (ITSAiU-2016): Sbornik trudov XIV Vserossiyskoy nauchnoy konferentsii molodykh uchenykh, aspirantov i studentov* [Information technology, system analysis and management (Idayu-2016): proceedings of the XIV all-Russian scientific conference of young scientists, postgraduates and students]. Rostov-on-Don: Izd-vo YuFU, 2016, Vol. 2, pp. 309-312.
19. *Algoritm Deykstry. Akademiya Microsoft: Struktury i algoritmy komp'yuternoy obrabotki dannykh. NOU "Intuit"* [The Dijkstra's Algorithm. Academy Microsoft: Structures and algorithms of computer data processing. NOU "Intuit"]. Available at: <http://www.intuit.ru/studies/courses/648/504/lecture/11475> (accessed 10 March 2017).
20. *Arifmeticheskoe skol'zyashchee srednee* [Arithmetic moving average]. Available at: <https://prog-cpp.ru/moving-average/> (accessed 20 March 2017).
21. *The difference between a compiled and interpreted language.* Available at: <http://www.programmerinterview.com/index.php/general-miscellaneous/whats-the-difference-between-a-compiled-and-an-interpreted-language/> (accessed 20 March 2017).
22. *The Data Plotting Software DISLIN.* Available at: <http://www.mps.mpg.de/dislin/online-manual> (accessed 20 March 2017).

Статью рекомендовал к опубликованию д.т.н., профессор В.И. Божич.

Поленов Максим Юрьевич – Южный федеральный университет; e-mail: mypolenov@sfedu.ru; 347928, г. Таганрог, пер. Некрасовский, 44; тел.: 88634371550; кафедра вычислительной техники; к.т.н.; доцент.

Лапшин Вячеслав Сергеевич – e-mail: lapshin@sfedu.ru; тел.: 88634371656; кафедра вычислительной техники; магистрант.

Гушанский Сергей Михайлович – e-mail: smgushanskiy@sfedu.ru; кафедра вычислительной техники; к.т.н.; доцент.

Polenov Maxim Yuryevich – Southern Federal University; e-mail: mypolenov@sfedu.ru; 44, Nekrasovskiy, Taganrog, 347928, Russia; phone: +78634371550; the department of computer engineering; cand. of eng. sc.; associate professor.

Lapshin Vyacheslav Sergeevich – e-mail: lapshin@sfedu.ru; phone: +78634371656; the department of computer engineering; graduate student.

Gushanskiy Sergey Mikhailovich – e-mail: smgushanskiy@sfedu.ru; the department of computer engineering; cand. of eng. sc.; associate professor.

УДК 004.032

DOI 10.23683/2311-3103-2017-6-223-233

В.Ф. Гузик, С.М. Гушанский, М.Ю. Поленов, В.С. Потапов

**РЕАЛИЗАЦИЯ КОМПЬЮТЕРНОГО МОДЕЛИРОВАНИЯ СИСТЕМЫ
С ЧАСТИЦЕЙ В ОДНОМЕРНОМ И ДВУХМЕРНОМ ПРОСТРАНСТВЕ
НА КВАНТОВОМ УРОВНЕ**

Описывается понятие пропускной способности, характерной для квантового канала передачи данных при наличии внешних шумов. Также выполнено компьютерное моделирование системы с частицей в одномерном и двухмерном пространстве на квантовом уровне с соответствующей программной реализацией и визуализацией результатов. В рамках