

Разработка блока системы управления роботом с применением инженерного пакета SciLab

А.Б. Колкер, Д.А. Ливенец, Кошелева А.И., Жмудь В.А.

1. ВВЕДЕНИЕ

Задача, решенная в данной работе – создание специализированной встраиваемой операционной системы на базе ядра *Linux*. В настоящее время ширится использование Linux-подобной операционной системы при создании информационной платформы технического устройства. В рамках этой задачи описан процесс создания специализированной встраиваемой операционной системы для функционирования на аппаратных ресурсах одноплатного компьютера, а также модулей сопряжения с *SciLab* и *WiFi* каналами связи.

2. РАЗРАБОТКА СТРУКТУРНОЙ СХЕМЫ МОДЕЛИ

Разрабатываем систему управления роботом, используя визуальные методы программирования в среде инженерных расчетов *SciLab*.

Описание используемого оборудования и объекта управления.

Для разработки системы управления выбрано следующее оборудование: одноплатный компьютер фирмы-производителя *IEI Technology incorp.* Модель *WAFER-945GSE2*. Основные характеристики приведены в приложении П.8. Данный компьютер имеет следующие параметры и характеристики:

– CPU - *Intel® Atom™ Processor N270 1.6 GHz with a 533MHz FSB*;

– Цифровые входы/выходы 8-бит цифровые входы/выходы (4-бита вход, 4-бита выход);

– ОЗУ - *DDR2 SDRAM*;

Ethernet - Dual Realtek RTL8111CP PCIe GbE controllers.

Объектом управления является гусеничное шасси, на которое крепится блок питания и одноплатный компьютер.

Сигналы управления передаются через канал *Wi-Fi* от персонального компьютера. Осуществлено сопряжение объекта управления, обеспечена передача сигналов управления через канал *Wi-Fi*.



Рис. 1. Одноплатный компьютер WAFER-945GSE2

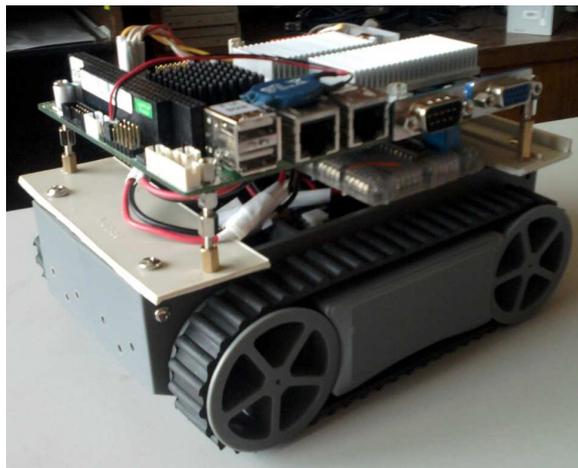


Рис. 2. Объект управления

3. РАЗРАБОТКА СТРУКТУРНОЙ СХЕМЫ

Необходимо обеспечить запуск и останов двигателя платформы по нажатию кнопки, изменение направления движения, регулировку скорости. Скорость движения должна изменяться без скачков и резких перепадов.

В соответствии с вышеуказанными требованиями была разработана схема, представленная на рис. 3.

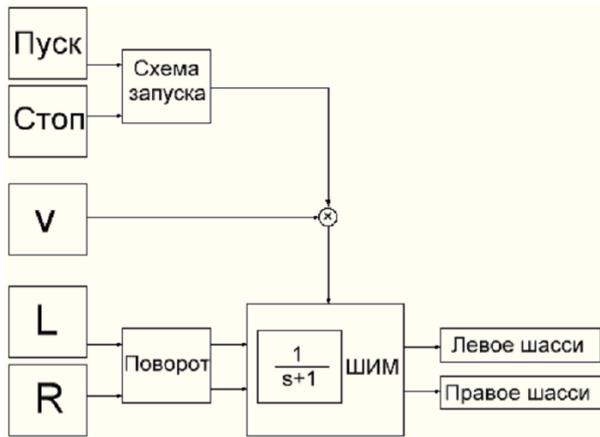


Рис. 3. Структурная схема общая

По нажатию кнопки Пуск на широтно-импульсный модулятор (ШИМ) подается импульс разрешающий движение. На ШИМ подается значение скорости, формируются управляющие тактирующие сигналы отдельно на правое и левое шасси. Схема запуска представлена на рис. 4.

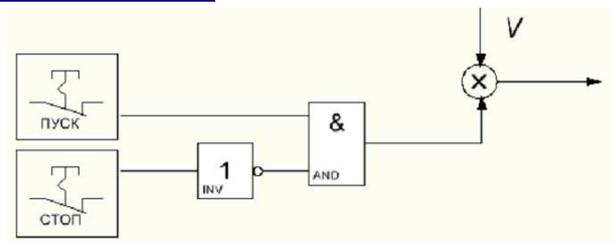


Рис. 4. Схема запуска

Кнопки Стоп и Пуск не могут быть нажаты одновременно. Запуск системы осуществляется, когда нажата кнопка Пуск и не нажата кнопка Сброс. Система выключена либо если не была нажата кнопка Пуск, либо если нажата кнопка Сброс.

На рис. 5 показана схема переключения направления представлена логика управления поворотом платформы. Чтобы платформа повернула, одно шасси снижает скорость до минимальной, а второе продолжает вращаться с прежней скоростью. Значения управляющих сигналов представлены в Таблице 1.

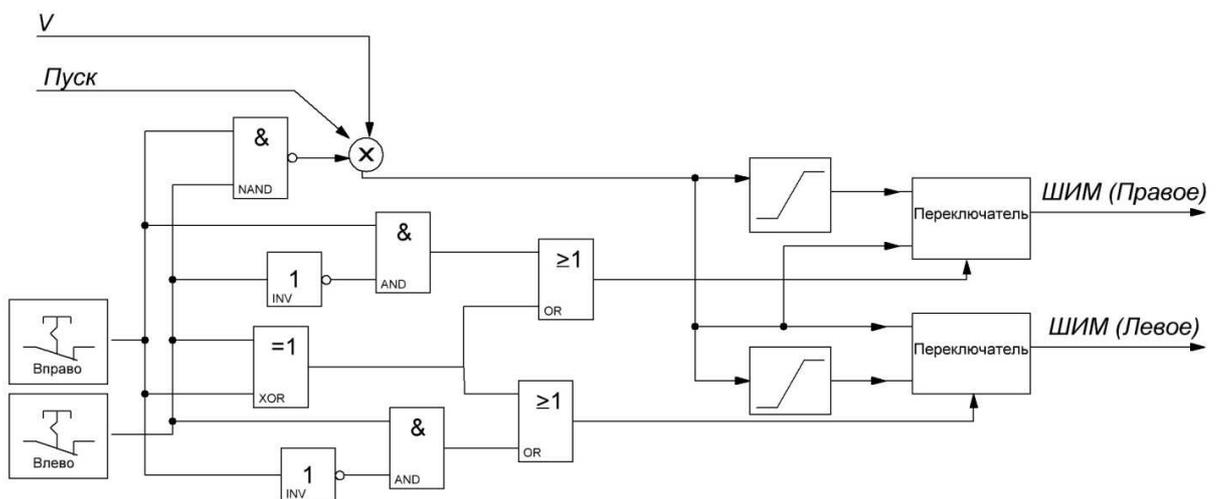


Рис. 5 . Схема переключения направления

Таблица 1

Таблица управляющих сигналов

Влево	Вправо	Выход ШИМ
1	1	Недопустимая комбинация
1	0	Поворот влево
0	1	Поворот вправо
0	0	Движение вперед (назад)

Таким образом, на выходе блока управления поворотом необходимо получить следующую временную диаграмму на рис. 6.

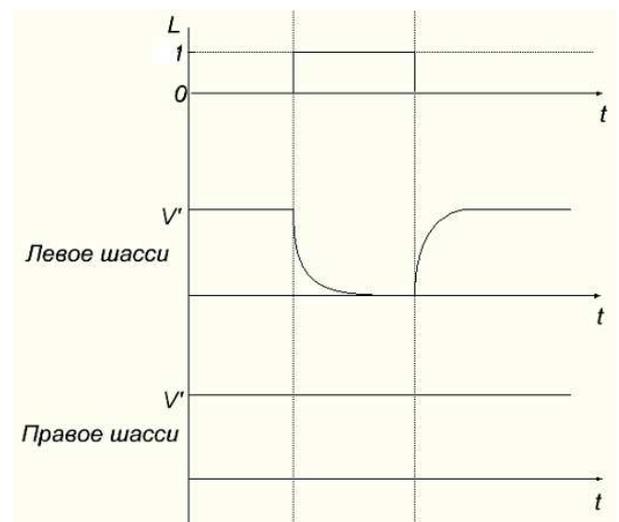


Рис. 6. Временная диаграмма шасси при

повороте

Сигнал с выхода схемы, обеспечивающей поворот платформы, поступает на ШИМ шасси.

ШИМ реализован на основе оператора, который возвращает значение периодической функции, был выбран оператор $\cos(u)$. Длительность периода сигнала с выхода оператора $\cos(u)$ изменяется в зависимости от

значения скорости: чем быстрее необходимо двигаться, тем чаще ШИМ формирует сигналы. Сигнал с выхода блока $\cos(u)$ поступает на компаратор, который регистрирует переход сигнала через 0, в момент перехода через ноль формируется прямоугольный импульс. В зависимости от знака сигнала скорости сигналы поступают на выход «Вперед» или «Назад».

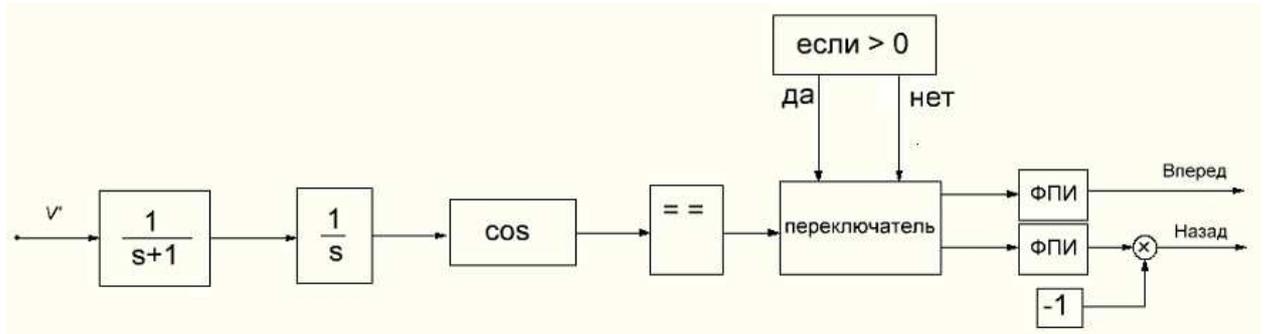


Рис. 7. Широтно-импульсный модулятор

Таким образом, на выходе блока управления необходимо получить на выходе результаты, представленные на рис. 8.

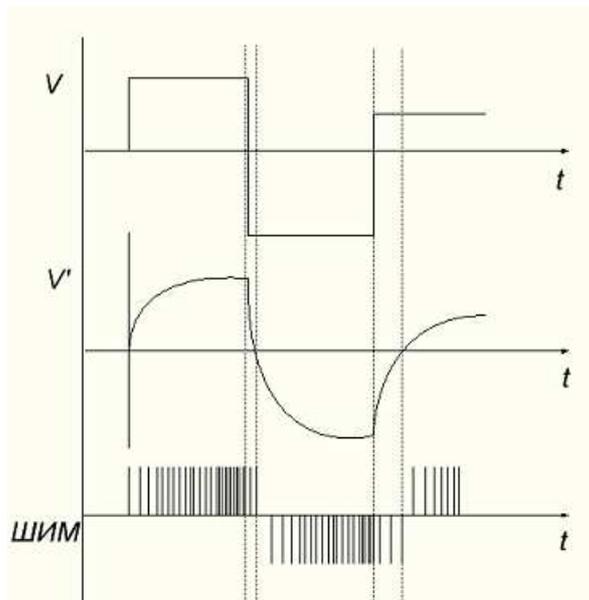


Рис. 8. Временная диаграмма блока управления

4. РАЗРАБОТКА ФУНКЦИОНАЛЬНОЙ СХЕМЫ

Значения скорости, направления, сигналов пуск и стоп, программируются в блоке *Signal builder*. Вместо этих блоков могут использоваться блоки *Sensor*, которые считывают сигнал с датчиков. Логические операции выполняются блоком *Logical op*, параметры блока позволяют выбрать логическую операцию (*OR*, *XOR*, *NOT*, *AND*, *NAND*, *NOR*), задать количество входов, тип входных данных (выходной сигнал будет тот же тип данных).

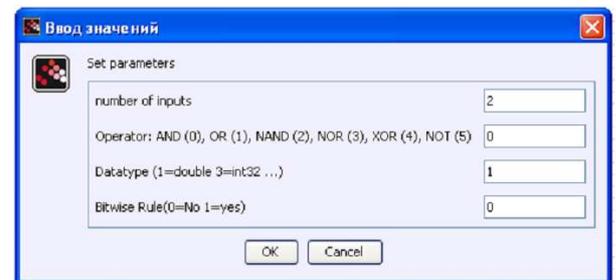


Рис. 9. Блок *Logical op*

Входные данные поступают в формате *Real double*, а для выполнения логических операций над данными и получения корректного результата необходимо конвертировать тип данных сигнала. Для этого в системе используется блок *Convert to*.

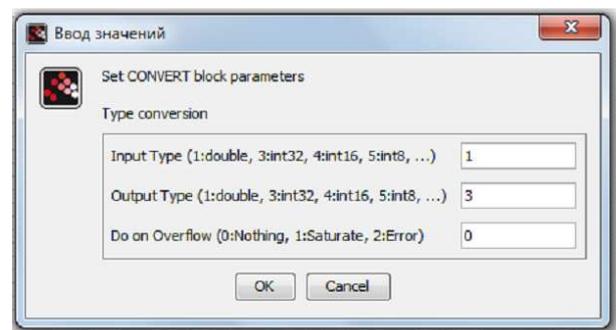


Рис.10. Блок *Convert to*

Данный блок имеет следующие параметры для настройки: тип входящих данных, тип данных, который необходимо получить, а так же параметр, отвечающий за переполнение.

На выходе системы сигнал равный значению скорости, если нажата кнопка Пуск, либо нулю, если нажата кнопка Стоп или не нажата кнопка Пуск, а так же необходимо исключить ситуацию, когда нажаты обе кнопки направления «Вправо» и «Влево». Схема запуска представлена на рис.

11. Схема управления поворотом представлена на *рис. 12.*

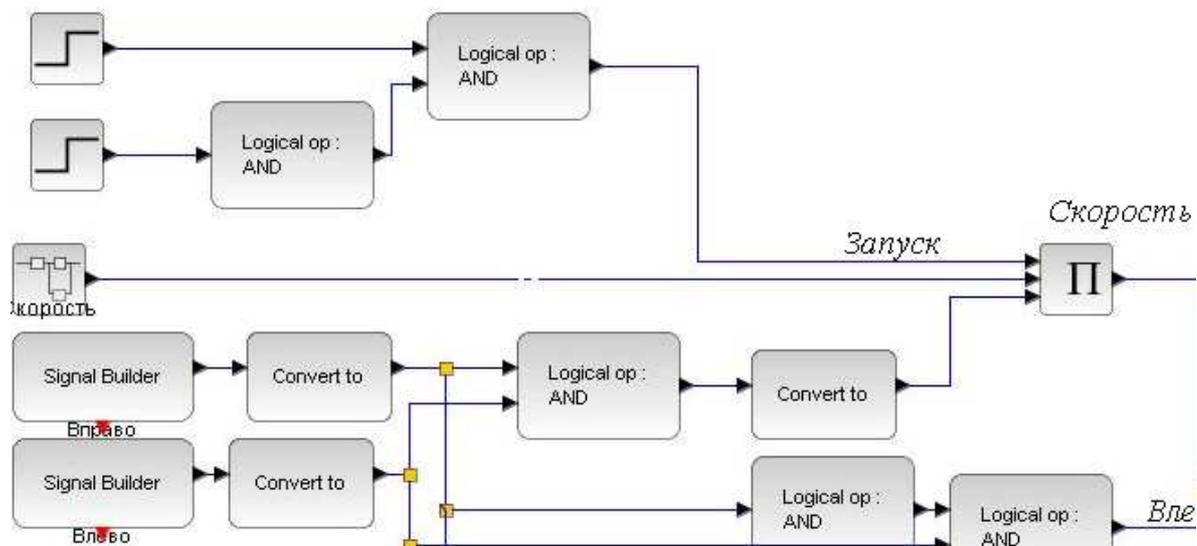


Рис. 11. Схема запуска

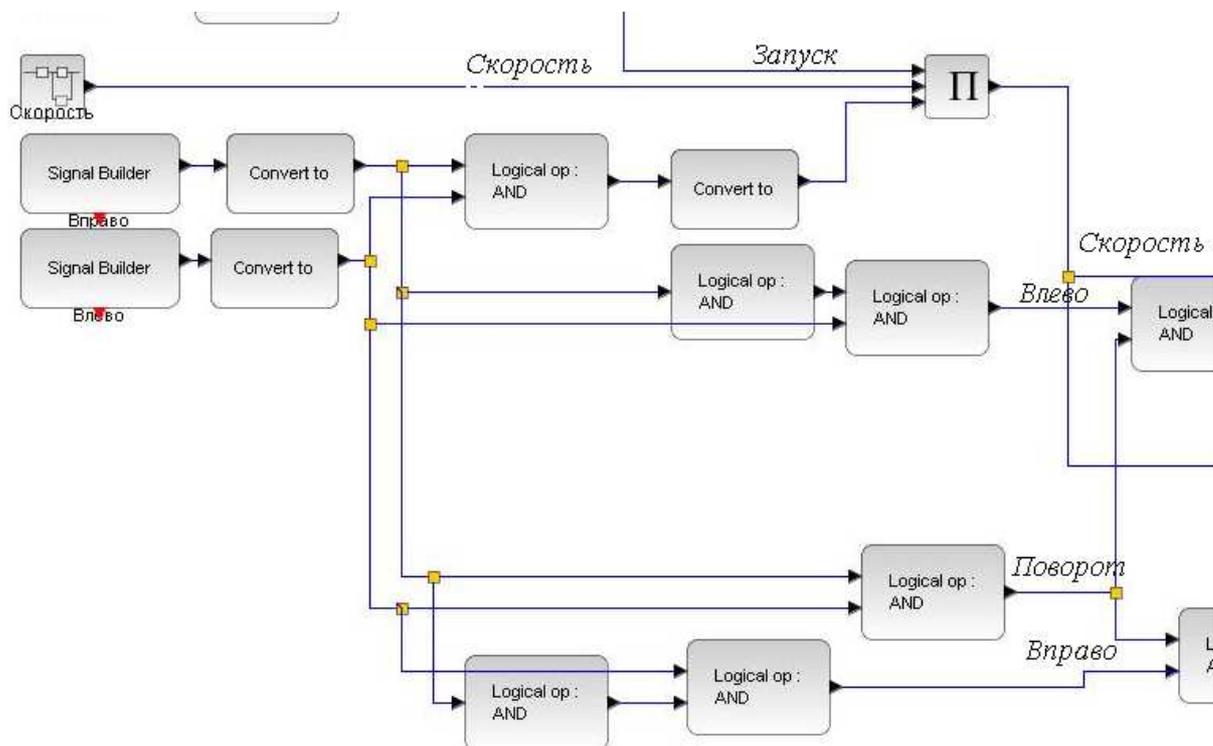


Рис.12. Схема управления поворотом

Если поступает сигнал «Влево» («Вправо»), то скорость вращения левого (правого) шасси снижается до 0, а скорость вращения правого понижается до допустимого уровня. На функциональной схеме этот принцип реализуется с помощью блока Switch (*рис. 13*), который подает на ШИМ правого шасси либо заданную скорость при движении прямо, либо допустимую скорость при повороте налево. Блок Switch

сравнивает значение входа 2 с параметром «а», если правило сравнения выполняется, то на выход блока подается вход 1, если не выполняется, тогда вход 3. Также можно выбрать тип входных и выходных данных, правило сравнения, правило перехода через 0. Сигнал Поворот становится 1, если оператор нажал любую кнопку поворота, и равен 0, если машина едет прямо.

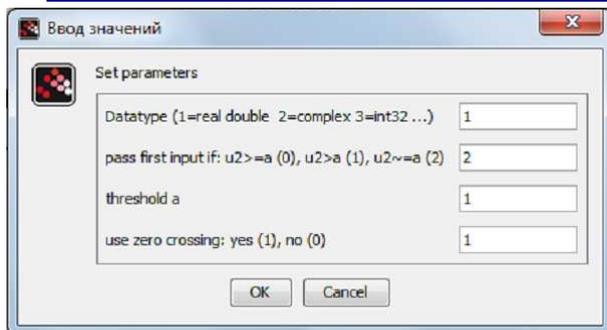


Рис. 13. Блок *Switch*

Для того, чтобы избежать заноса на прохождении поворота, использован блок ограничения скорости *Saturation* (рис. 14. Блок *Saturation*). Задаваемые пользователем параметры: нижний предел, верхний предел, правило перехода через 0. На рис.15. приведена схема управления шасси.

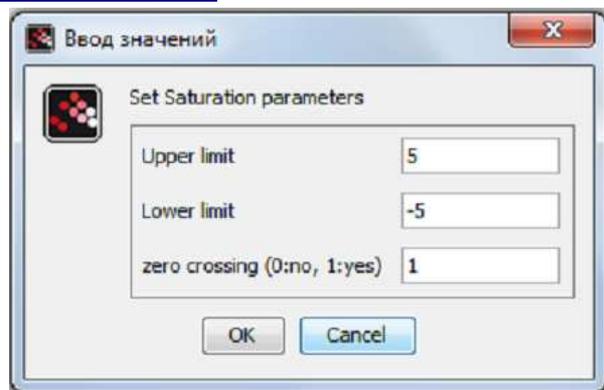


Рис.14. Блок *Saturation*

Сигнал от схемы управления шасси поступает на ШИМ шасси. Для каждого выполняется отдельный модуль. В качестве сглаживающего резкие скачки переключения скорости блока выбрано апериодическое звено первого порядка.

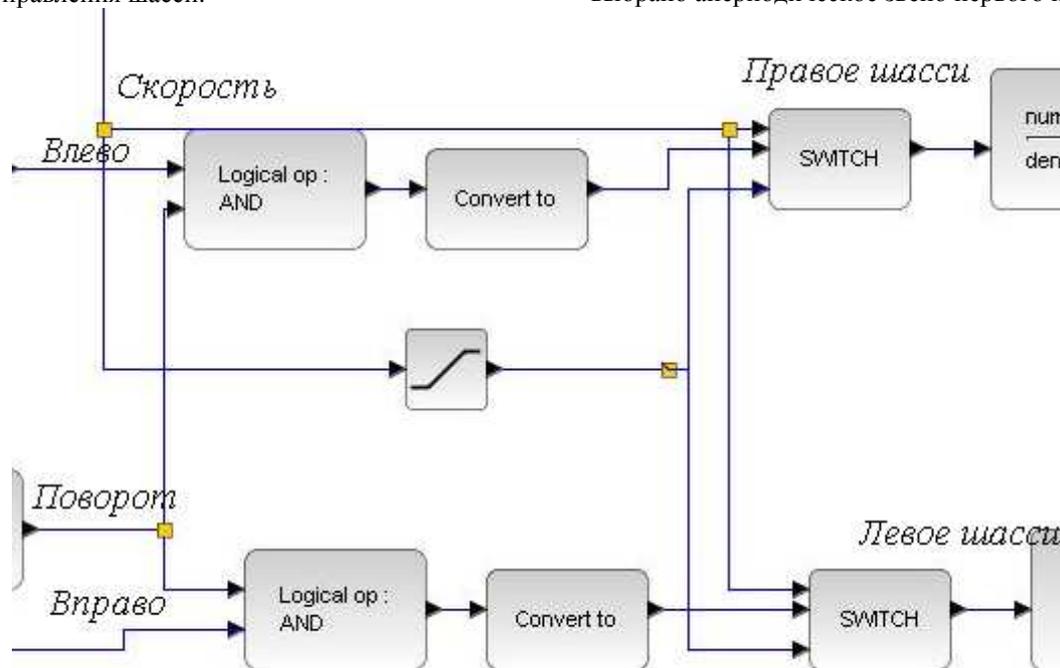


Рис.15. Схема управления правым (левым) шасси

Поскольку программа с выхода апериодического звена передает на блок функции $\cos(u)$ не значение, а скорость изменения сигнала, то в ходе моделирования было принято решение использовать в схеме интегратор. Коэффициент отношения значения скорости к частоте тактирования шасси устанавливается в блоке $\cos(u)$. Схема широтно-импульсного модулятора представлена на рис. 16. Модель системы объединяется в подсистему «*Super Block*» для

дальнейшей компиляции. На конечном этапе моделирования схема имеет вид рис. 17. Блок управления был скомпилирован, создан исполняемый файл с открытым кодом, таким образом, программу можно по необходимости корректировать, встраивать в систему управления.

Таким образом, модель системы управления примет вид, показанный на рис. 18.

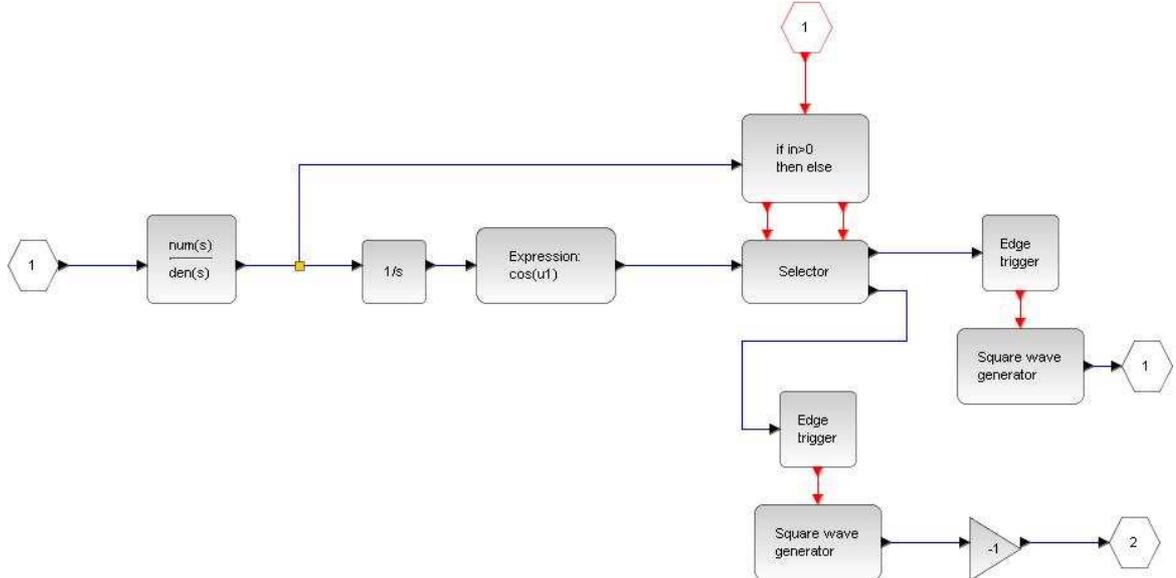


Рис.16. Широтно-импульсный модулятор

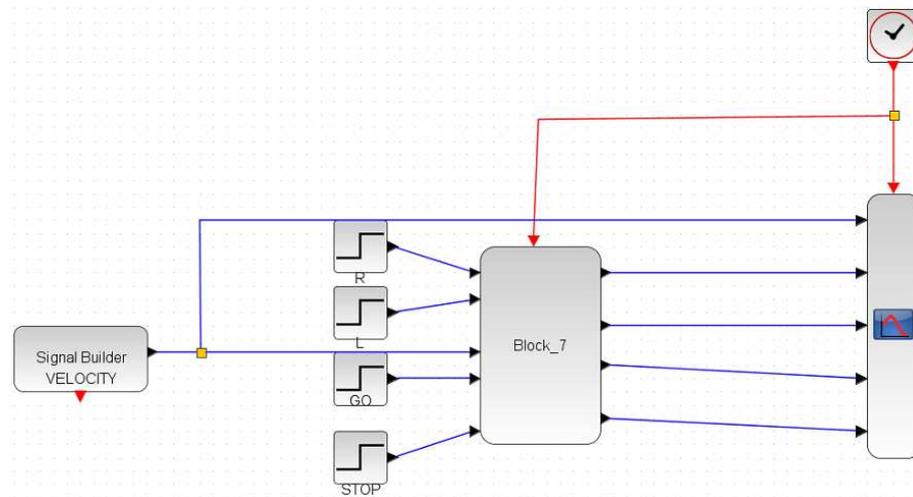


Рис.17. Схема управления

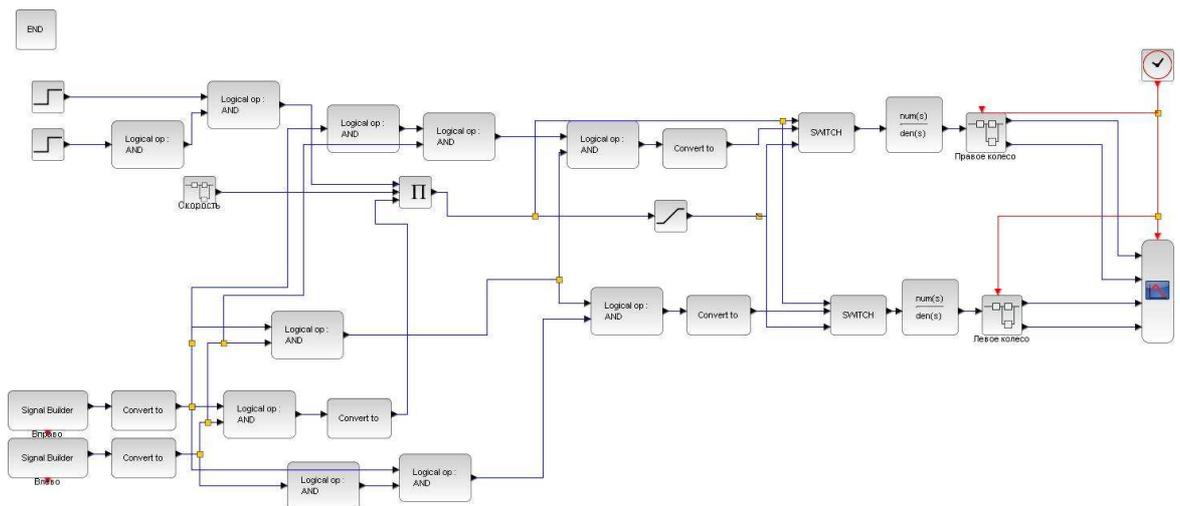


Рис. 18. Функциональная схема модели управления

Модель системы объединяется в подсистему «Super Block» для дальнейшей компиляции. Встроенная команда CodeGen создает исполняемый файл, который представляет собой

программу на языке Си. В программе имеется виртуальный осциллограф, который позволяет наблюдать результат выполнения созданной системы управления.

5. РЕЗУЛЬТАТЫ МОДЕЛИРОВАНИЯ

На вход системы подается значение скорости, представленное на *рис. 19*. Изменения направления представлены на диаграмме *рис. 20*. В результат моделирования системы на выходе ШИМ тактирующих шасси наблюдаются сигналы, представленные на *рис. 21*.

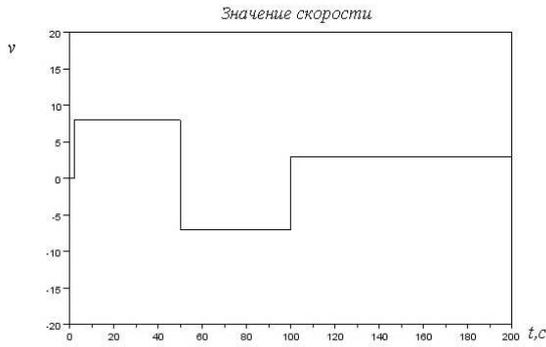


Рис. 19. Значение скорости с блока логического программирования сигнала для отладки системы управления

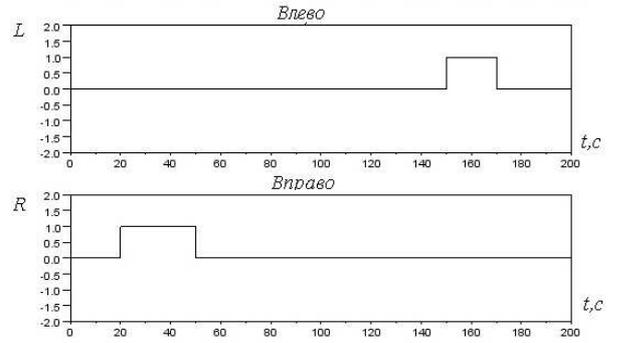


Рис. 2. Изменения направления движения

В результате моделирования системы на вход широтно-импульсного модулятора шасси передаются измененное значение скорости, график представлен на *рис. 22*.

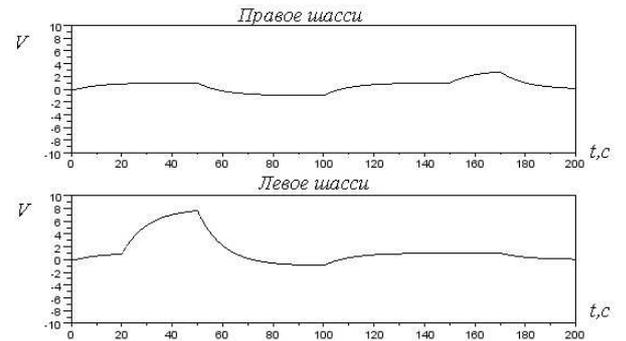


Рис. 21. Значение скорости на входе ШИМ

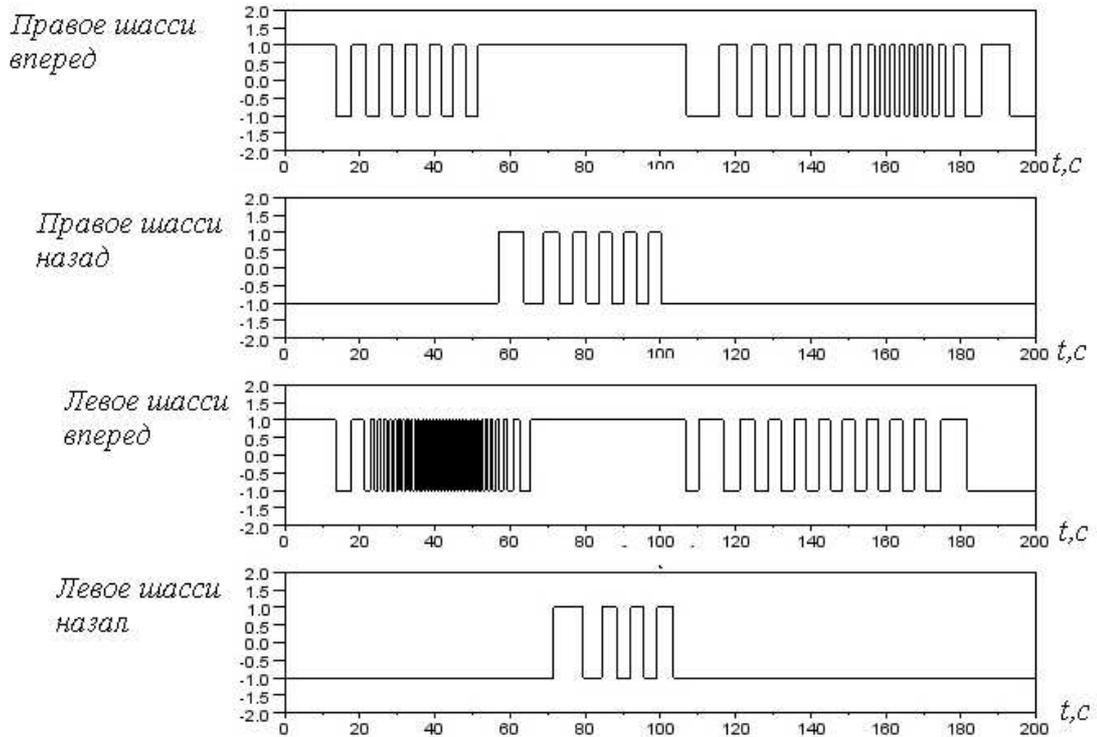


Рис. 22. Выходные сигналы

По результатам моделирования можно сделать вывод, что реализованный блок

управления выполняет заданные требования. Перед запуском программы необходимо

установить следующие параметры. Задать скорость управления, момент изменения направления движения (поворот вправо или влево), коэффициент передачи скорости к частоте тактирования шасси, а так же коэффициент сглаживания скачков при смене значения скорости. Скорость управления задается в блоке Signal builder Velocity, момент изменения направления вправо Signal builder R, момент изменения направления влево Signal builder L, коэффициент сглаживания устанавливается в параметрах апериодического звена, а коэффициент передачи скорости к частоте тактирования в параметрах блока периодической функции.

Таблица 2

Таблица управляющих сигналов

Направление	Двигатель правого шасси	Двигатель левого шасси	Движение	В десятичной системе
11	1	1	вперед	15
01	1	1	вправо	7
10	1	1	влево	11
00	1	1	назад	3
00	0	0	стоп	0

Однако для запуска данной модели на объекте управления необходимо внести корректировку, т.к. для управления заданным объектом на выходе системы необходимо получать 4-битный код.

Для запуска модели на объекте управления необходимо модифицировать схему, так как

управлять двигателями шасси возможно только получая на выходе следующие двоичные комбинации.

Значения сигналов управления приведены в Таблице 2.

6. РАЗРАБОТКА СТРУКТУРНОЙ МОДЕЛИ ДЛЯ ЗАПУСКА ОБЪЕКТА УПРАВЛЕНИЯ

В структурную схему, представленную на рис. 23, внесены изменения типа данных, получаемых на выходе. Схема управления направлением движения представлена на рис. 24.

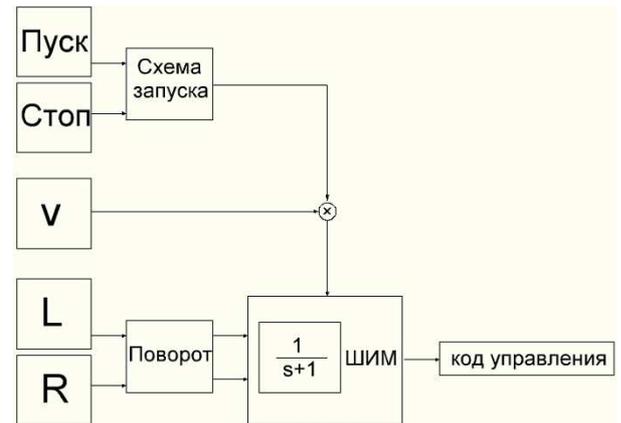


Рис. 23. Структурная схема

Значения управляющих сигналов представлены в Таблице 3. Поскольку платформа разворачивается, вращая шасси в разные стороны, то при повороте нет необходимости управлять каждым шасси отдельно.

Функциональная схема собрана на ранее описанных блоках, представлена на рис. 25. Модель системы объединяется в подсистему «Super Block» для дальнейшей компиляции.

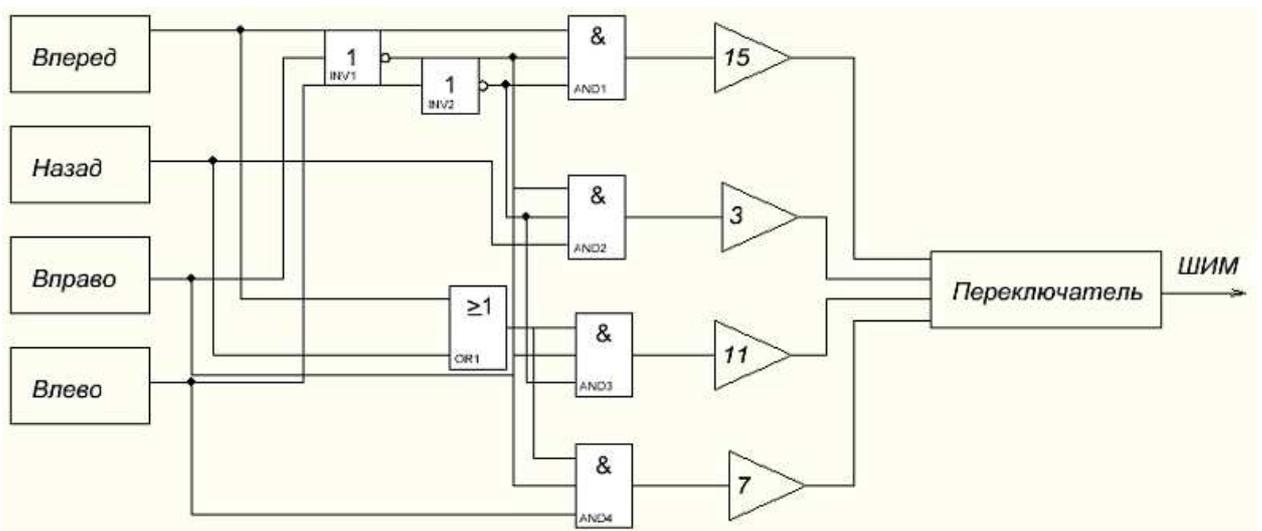


Рис. 24. Схема управления направлением

Таблица управляющих сигналов

Вперед	Назад	Влево	Вправо	Направление	Выход
0	0	0	0	Стоп	0
0	0	0	1		
0	0	1	0		
0	0	1	1		
0	1	0	0	назад	3
0	1	0	1	вправо назад	7
0	1	1	0	влево	11
0	1	1	1	нерабочая комбинация	-
1	0	0	0	вперед	15
1	0	0	1	вперед вправо	7
1	0	1	0	вперед влево	11
1	0	1	1	нерабочая комбинация	-
1	1	0	0	нерабочая комбинация	-
1	1	0	1		
1	1	1	0		
1	1	1	1		

Результаты моделирования показаны на рис. 26 – 28.

На вход системы подается сигнал скорости,

изображенный на рис. 26.

Эти диаграммы отражают работоспособность модели в заданных условиях.

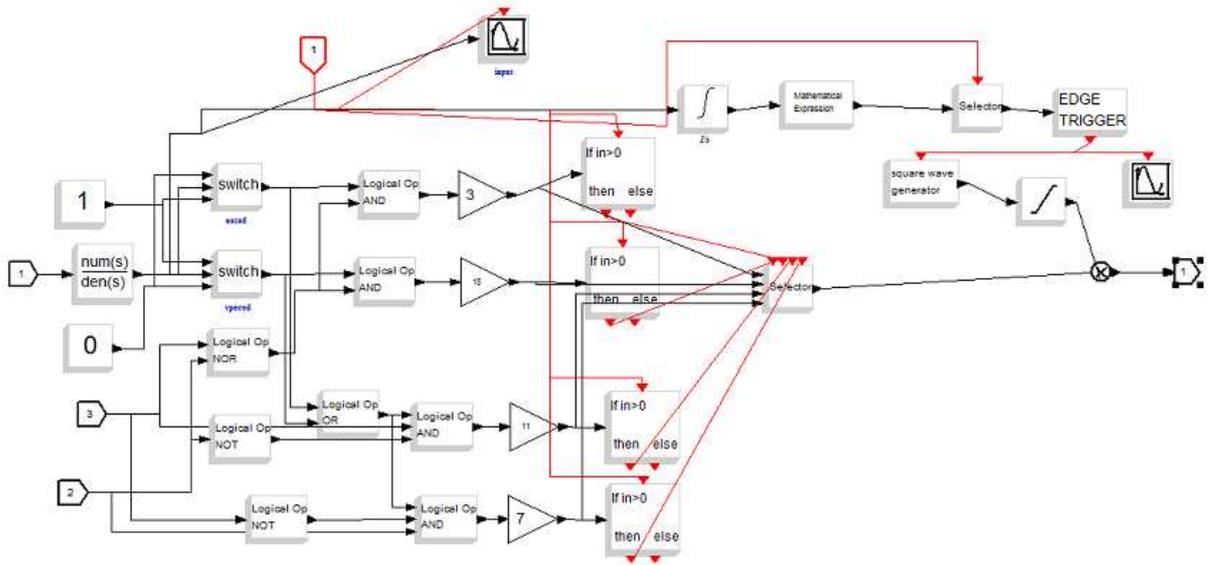


Рис. 25. Функциональная схема

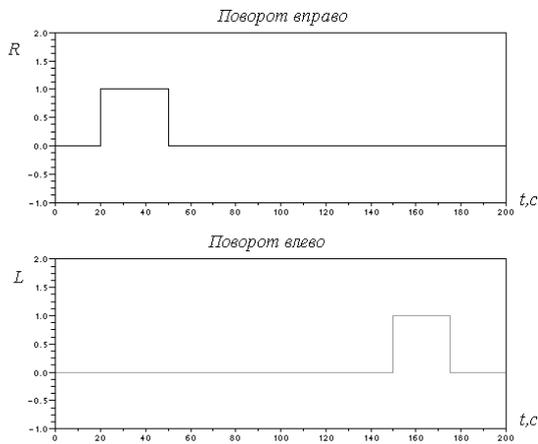


Рис. 26. Входные сигналы

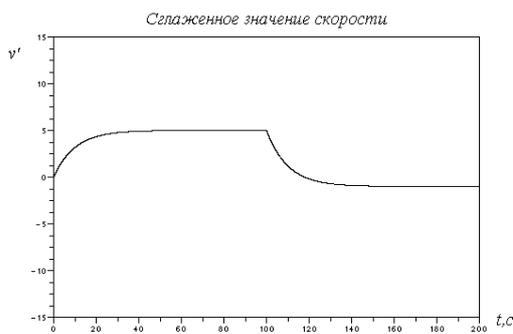


Рис. 27. Сглаженное значение скорости

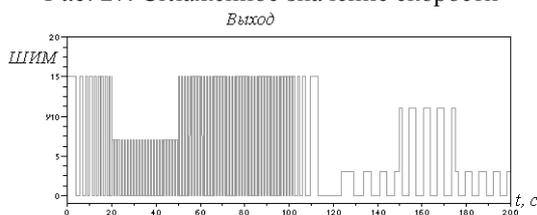


Рис. 28. Выход ШИМ

7. ОТЛАДКА И КОМПИЛЯЦИЯ

Для компиляции и запуска программы на объекте управления используется схема,

показанная на рис. 29.

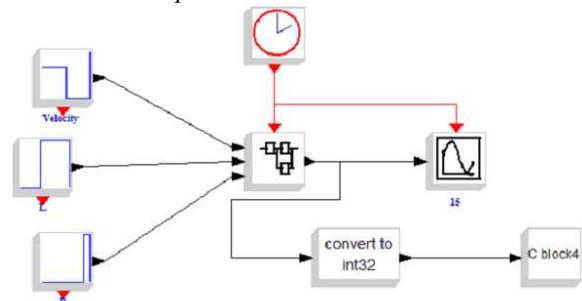


Рис. 29. Схема для тестирования

Здесь *C block 4* блок, передающий сигнал на драйвер управления шасси через канал *Wi-Fi*. Блок является пользовательской функцией, написанной на языке Си. Суперблок модели транслируется в Си код, который затем компилируется при помощи компилятора *GCC*. Сгенерированный *C* текст может быть откомпилирован для операционных систем *Linux* и *Windows* без изменений, т.к. соответствует стандарту *POSIX*.

В качестве входных сигналов использовались следующие блоки:

- заранее запрограммированная последовательность команд (жесткая логика);
- команды, переданные при помощи *WiFi*.

В случае использования заранее запрограммированной последовательности команд модель объединяется в суперблок высшего уровня (включающий блоки-источники сигнала). Блоки – источники сигнала должны быть сконфигурированы: скорость управления, момент изменения направления движения (поворот вправо или влево), коэффициент скорости к частоте тактирования шасси, а так же коэффициент сглаживания скачков при смене значения скорости. Скорость управления задается в блоке *Signal builder*

Velocity, момент изменения направления вправо *Signal builder R*, момент изменения направления влево *Signal builder L*, коэффициент сглаживания устанавливается в параметрах апериодического звена, а коэффициент передачи скорости к частоте тактирования в параметрах блока периодической функции.

Результаты моделирования дали положительные результаты: платформа движется в заданном направлении, обрабатывается изменение скорости.

ЗАКЛЮЧЕНИЕ

Система *SciLab* позволяет как моделировать систему управления объектом и исследовать данные, полученные на виртуальном объекте, так и использовать данную модель для управления реальным объектом в виде исполняемого приложения. Компиляция исполняемого приложения осуществляется в два этапа:

- трансляция структуры модели в Си текст;
- компиляция Си текста в исполняемый код при помощи компилятора *GCC*.

Си текст хорошо читается, создается согласно стандартам *POSIX* и может быть подвергнут изменениям. Изменения разрешены лицензией *GNU*, в рамках которой развивается данный проект. Исходный код структурирован, интуитивно понятен. В работе использована литература [1–9].

Работа выполнена по заданию Министерства образования и науки РФ, проекты № 7.559.2011 (Темплан) и ГК № П761 от 20.05.2010.

ЛИТЕРАТУРА

- [1] Свободные математические программы URL : <http://openmath.ucoz.ru/>
- [2] Имитационное моделирование системы управления с использованием пакета SciLab: дипломный проект: рук. А.Б. Колкер; исполн. А.И. Кошелева, 2010. - 44 с. , 2010г.
- [16] User manual : URL: http://www.ieiworld.com/files/file_pool/0B329000195966240272/file/WAFER-945GSE2_UMN_v2.00.pdf
- [3] RTAI - Real Time Application Interface Official Website. URL: <https://www.rtai.org/>
- [4] Scicos: Block diagram modeler/simulator. URL: <http://scicos.org/>
- [5] The Free Platform for Numerical Computation. URL: <http://www.scilab.org/support/documentation>
- [6] SciLab refence manual. Scilab Group INRIA Meta2 Project/ENPC Cergrene. - Domaine de Voluceau – Rocquencourt: - BP 105 - 78153 Le Chesnay Cedex (France). 2001.
- [7] Nikoukhah R., Steer S. SCICOS - A Dynamic System Builder and Simulator. 1998.

- [8] Modeling and Simulation in Scilab/Scicos with ScicosLab 4.4 Справочное пособие / Stephen L. Campbell и др. – Springer, 2010.
 - [9] Campbell S. L. Modeling and Simulation in Scilab/Scicos /S. L. Campbell, J.P. Chancelier, R. Nikoukhah – NY 10013, USA: Springer Science , 2006.
 - [10] Галынский В.М. Свободно распространяемые системы компьютерной алгебры и возможности их применения в образовании / Белорусский Государственный Университет Минск, Беларусь. – 2006.
 - [11] W. R. Stevens, S. A. Rago. Advanced Programming in the UNIX® Environment: Second Edition / Addison Wesley Professional, 2005.
 - [12] Стивенс У. UNIX: Разработка сетевых приложений / СПб.: Питер, 2007.
 - [13] Марк Дж. Рочкинд. Программирование для UNIX. Наиболее полное руководство Advanced UNIX Programming / БХВ-Петербург, Русская Редакция, 2005.
 - [14] Н. Мэтью, Р. Стоунс. Основы программирования в Linux / БХВ-Петербург, 2009.
 - [15] SciLab refence manual. Scilab Group INRIA Meta2 Project/ENPC Cergrene. - Domaine de Voluceau – Rocquencourt: - BP 105 - 78153 Le Chesnay Cedex (France). 2001.
 - [16] Nikoukhah R., Steer S. SCICOS. A Dynamic System Builder and Simulator,1998.
 - [17] Modeling and Simulation in Scilab/Scicos with ScicosLab 4.4 Справочное пособие / Stephen L. Campbell и др. – Springer, 2010.
 - [18] Scicos: Block diagram modeler/simulator. [Электронный ресурс] – URL: <http://scicos.org/>
 - [19] The Free Platform for Numerical Computation [Электронный ресурс] URL: <http://www.scilab.org/support/documentation>
 - [20] CHIPINFO – крупнейший справочник по микросхемам. URL: <http://www.chipinfo.ru/dsheets/ic/155/In1.html>
 - [21] DatasheetCatalog.com – свободный онлайн-справочник электронных компонентов и полупроводников. URL: <http://www.datasheetcatalog.org/datasheet/texasinstruments/1293d.pdf>
 - [22] IEL, the industrial computer provider focus on industrial boards, display, embedded system, digital signage, medical solution, pos, networking platforms. URL: http://www.ieiworld.com/product_groups/industrial/content.aspx?gid=00001000010000000001&cid=09050665109937740140&id=08323535591322447606
 - [23] Advantech, solutions in the e-world computing and web-based automation fields. URL: http://www.advantech.com/products/PCM-4153/mod_A8209479-19F4-47AD-8446-84FBD131D005.aspx
 - [24] DFRobot-An Online Opensource Robot. URL: [http://www.dfrobot.com/wiki/index.php?title=Tank_RP5_Chassis_\(SKU:ROB0007\)](http://www.dfrobot.com/wiki/index.php?title=Tank_RP5_Chassis_(SKU:ROB0007))
- Колкер Алексей Борисович** – доцент кафедры Автоматики НГТУ, к.т.н.,
e-mail: fiery77@yandex.ru
Ливенец Дмитрий Александрович – магистр, окончил кафедру Автоматики НГТУ,
e-mail: dmitry.livenets@gmail.com
Кошелева Алёна Игоревна – магистр, окончила кафедру Автоматики НГТУ,
e-mail: kosheleva.alvona@mail.ru
Жмудь Вадим Аркадьевич – заведующий кафедрой Автоматики НГТУ, д.т.н.,
e-mail: оао_nips@bk.ru