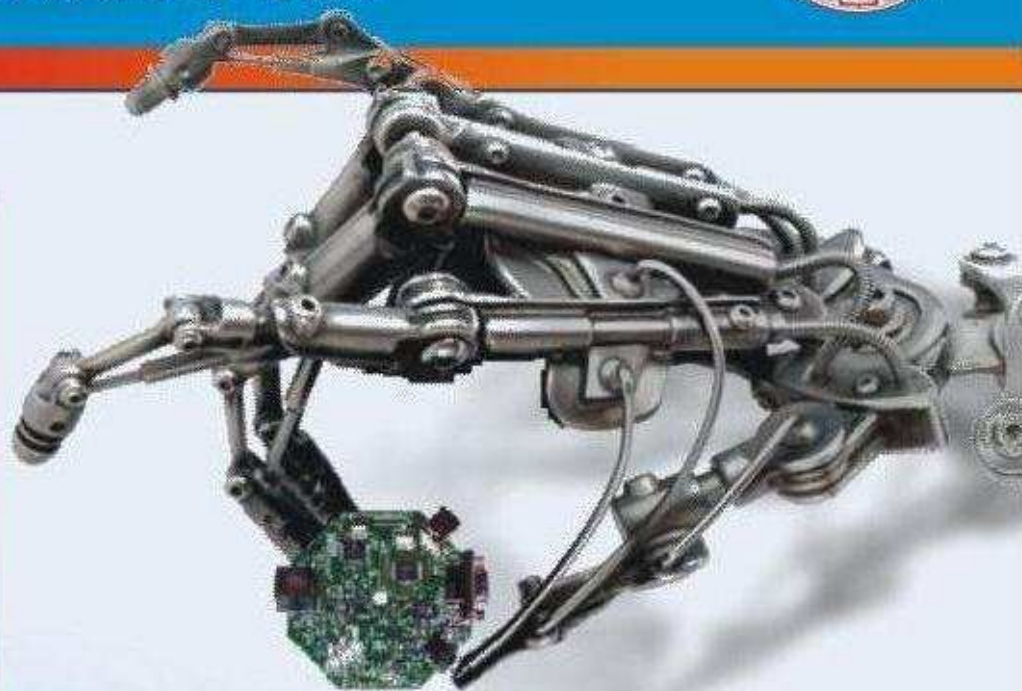




2012 - №1 (1)

## В НОМЕРЕ

- I. Автоматика  
Автоматизированные системы  
Измерительные средства и системы
- II. Робототехника  
Программные средства и системы  
Свободное программное обеспечение  
Открытое программное обеспечение
- III. Инновационные технологии  
Перспективные технологии  
Моделирование систем  
Облачные технологии  
Национальная программная платформа
- IV. Популярные статьи  
Введение в специальность  
К сведению авторов



**Главный редактор д.т.н., профессор В.И. Гужов,**  
Новосибирский государственный технический университет (НГТУ)

**Заместитель главного редактора д.т.н., доцент В.А. Жмудь,**  
Новосибирский государственный технический университет, ОАО «НИПС»

**Ответственный секретарь д.т.н., доцент Г.А. Французова,**  
Новосибирский государственный технический университет

**Редакционная коллегия:**

- А.С. Востриков** Профессор кафедры Автоматики НГТУ, д.т.н., профессор
- S.S. Mahapatra** Professor in the Department of Mechanical Engineering, NIT Rourkela,  
**India**
- H. Roth** Head of the Department of Automatic Control Engineering of  
University if Siegen, **Germany** , professor
- В.В. Губарев** Заведующий кафедрой Вычислительной техники НГТУ,  
профессор, д.т.н., профессор
- А.А. Воевода** Профессор кафедры Автоматики НГТУ, д.т.н.
- Е.В. Рабинович** Профессор кафедры Вычислительной техники НГТУ, д.т.н.,  
профессор
- V. J. Savsani** Associate Professor at B. N. Gardi college of engineering and  
technology, Surat, India
- Б.Д. Борисов** Заведующий лабораторией Института Лазерной физики СО РАН  
(ИЛФ СО РАН), д.т.н.
- Б.В. Поллер** Заведующий лабораторией Института Лазерной физики СО РАН  
(ИЛФ СО РАН), д.т.н.
- А.В. Ляпидевский** Директор ОАО «Новосибирский институт программных систем»  
(ОАО «НИПС»), к.э.н.
- В.И. Гололобов** Руководитель лаборатории ОАО «НИПС», к.т.н.
- М.В. Калинин** Руководитель отдела ОАО «НИПС»
- Н.Р. Рахимов** Руководитель лаборатории Оптоэлектронного метода  
неразрушающего контроля» СГГА., профессор, д.т.н.

**Учредители журнала:**

Федеральное государственное бюджетное образовательное учреждение Высшего профессионального образования «Новосибирский государственный технический университет»

Сайт организации: [www.nstu.ru](http://www.nstu.ru)

Открытое акционерное общество «Новосибирский институт программных систем»

Сайт организации: [www.nips.ru](http://www.nips.ru)

**Адреса редакции:**

630092, Россия, Новосибирск, просп. Карла Маркса, 20,  
ФГБОУ ВПО НГТУ, Факультет Автоматики и вычислительной техники, декану

630090, Россия, Новосибирск, просп. Академика Лаврентьева, 6/1,  
ОАО «НИПС», заместителю директора по науке

e-mail: [oaonips@bk.ru](mailto:oaonips@bk.ru)

web: <http://ait.cs.nstu.ru/content/ape>

[http://www.nips.ru/index.php?option=com\\_content&task=view&id=424&Itemid=437](http://www.nips.ru/index.php?option=com_content&task=view&id=424&Itemid=437)



## Оглавление

Анализ метода проектирования робастного регулятора методом двойной итеративной параллельной численной оптимизации.....	7
Радиочастотный метод измерения сверхмалых перемещений и вибраций.....	17
Исследование оптоэлектронного метода и разработка устройств для мониторинга жидких сред .....	35
Оптоэлектронный метод контроля физико-механических параметров поверхностей твердых материалов .....	43
Обоснование выбора программного обеспечения для робототехники.....	51
Разработка системы управления стенда для исследования переходных процессов вакуумного высоковольтного выключателя.....	65
Разработка блока системы управления роботом с применением инженерного пакета SciLab.....	73
Исследование вариантов создания интеллектуальных систем робототехники на базе одноплатных компьютеров и свободных операционных систем .....	84
Преимущества свободного программного обеспечения для стратегических технологий .....	101
Апология Теории Автоматического Управления .....	109
Будущее программы двойных магистерских дипломов по автоматике: взаимодействие НГТУ с вузами России, Украины, Болгарии, Чехии и Франции.....	134
Требования к публикациям в научном электронном журнале «Автоматика и программная инженерия» .....	138

**I**  
**Автоматика**  
**Автоматизированные системы**  
**Измерительные средства и**  
**системы**

### Содержание раздела:

- В.А. Жмудь, А.Н. Заворин, А.В. Полищук, О.Д. Ядрышников. Анализ метода проектирования робастного регулятора методом двойной итеративной параллельной численной оптимизации. С. 7–16.
- В.А. Жмудь, Д.О. Терешкин, А.В. Ляпидевский, А.В. Захаров. Радиочастотный метод измерения сверхмалых перемещений и вибраций. С. 17–34.
- Е.Ю. Кутенкова, Б.Н. Рахимов, Т.В. Ларина, Ш.И. Мадумаров. Исследование оптоэлектронного метода и разработка устройств для мониторинга жидких сред. С. 35–42.
- Т.В. Ларина, Е.Ю. Кутенкова, В.А. Жмудь, Д.Д. Алижанов. Оптоэлектронный метод контроля физико-механических параметров поверхностей твердых материалов. С. 43–48.

# Анализ метода проектирования робастного регулятора методом двойной итеративной параллельной численной оптимизации

В.А. Жмудь, А.Н. Заворин, А.В. Полищук, О.Д. Ядрышников  
Novosibirsk State Technical University (NSTU); Novosibirsk State University (NSU),

**Аннотация:** Решается задача синтеза для объекта с запаздыванием, изменяющим свои параметры приблизительно в 3 раза. Задача решается путем минимизации функционального критерия качества, определения неблагоприятных сочетаний параметров объекта и повторной оптимизации набора систем, включающих объекты с выявленными неблагоприятными параметрами. Метод пояснен примером из практики.

**Ключевые слова:** синтез регуляторов, робастное управление, численная оптимизация, критерии оптимизации

## 1. ВВЕДЕНИЕ

Управление динамическими объектами в замкнутом контуре существенно усложняется, если параметры объекта известны недостаточно точно, или изменяются под действием неконтролируемых факторов, хотя и остаются в известном интервале. В этом случае необходимо применение методов робастного управления, обеспечивающих успешное управление объектом при всех возможных вариантах изменения его параметров. В настоящее время широко известны следующие методы робастного регулирования:  $H_\infty$ -синтез;  $H_2$ -синтез;  $LQG$ -синтез;  $LQR$ -синтез [1–4].

Один из методов робастного управления состоит в оптимизации при одновременном моделировании совокупности систем с одинаковыми регуляторами, но разными объектами, описываемыми наиболее характерными моделями из множества возможных моделей этого объекта. Главная идея этого подхода состоит в том, чтобы одинаковыми регуляторами обеспечить устойчивость замкнутых систем не только с объектом с номинальными параметрами, но и с выбранными образцами моделей объекта с измененными параметрами. Если при численной оптимизации регулятора для единственного объекта в качестве критерия оптимальности используется интегральный критерий качества системы, то при численной оптимизации одного регулятора для множества объектов следует

использовать сумму этих критериев.

Примером системы автоматического регулирования (САР) с таким объектом является САР температуры пара в теплоэнергетических установках [5]. Это САР острого пара (температура пара на выходе из энергетического котла) и САР промежуточного перегрева (температура пара на выходе из цилиндра высокого давления турбины). Регулирование температуры пара относится к важнейшим задачам автоматизации теплоэнергетических процессов, поскольку ее решение обеспечивает более долгий срок эффективной эксплуатации оборудования и увеличивает КПД теплоэнергетической установки. В настоящее время настройка таких регуляторов остается большой технической проблемой. Перспективным является решение поставленной задачи на основе исследований численными методами оптимизации. В данной работе исследуется инерционный контур САР температуры пара с параметрами объекта, меняющимися в зависимости от режима работы теплоэнергетической установки.

## 2. ПОСТАНОВКА ЗАДАЧИ

Объект управления имеет известную модель в виде передаточной функции  $W_0(s)$ , например [5]:

$$W_0(s) = \frac{k}{(T_1s + 1)(T_2s + 1)} e^{-sT_d}. \quad (1)$$

Параметры этой передаточной функции изменяются в широких пределах. Необходимо рассчитать регулятор, обеспечивающий устойчивое и точное управление объектом не только при номинальном значении передаточной функции, но также и при всех ее возможных изменениях. Качество регулятора оценивается по формальным критериям, таким как перерегулирование, длительность переходного процесса до достижения наперед заданной малой величины ошибки, интеграл от модуля или от квадрата ошибки и так далее.

### 3. МЕТОД РЕШЕНИЯ ПОСТАВЛЕННОЙ ЗАДАЧИ

Для обеспечения требуемого качества системы при различных значениях параметров объекта предлагается использовать множество объектов, описываемых соотношением (1) с различными значениями параметров, как показано на рис. 1. Для каждого из таких объектов моделируется идентичный регулятор. Его параметры вычисляются методом оптимизации по критерию, в который входят ошибки всех систем множества.

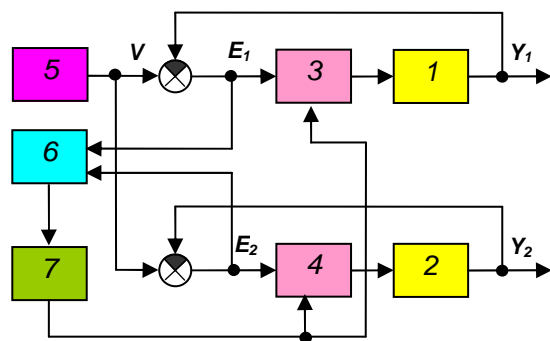


Рис.1. Схема оптимизации регулятора: 1, 2 – модели объекта с разными значениями параметров; 3, 4 – идентичные модели регуляторов; 5 – формирователь тестового воздействия; 6 – анализатор качества систем; 7 – оптимизатор параметров регулятора

Структуру регулятора зададим в виде последовательного корректирующего устройства:

$$W_R(s) = k_P + \frac{k_I}{s} + k_D s, \quad (2)$$

Расчет параметров регулятора (2) при фиксированном значении параметров передаточной функции (1) может быть осуществлен их оптимизацией при моделировании системы в одной из известных программ [5–7]. Для получения робастного регулятора следует оптимизировать его коэффициенты при одновременном использовании такого регулятора в нескольких объектах, отличающихся параметрами. В этом случае регулятор используется для каждой системы один и тот же, а критерий качества вычисляется как сумма критериев каждой системы.

Пусть  $e_1(s)$  – ошибка регулирования номинального объекта,  $e_2(s)$  – ошибка регулирования возмущенного объекта. Численная оптимизация коэффициентов регулятора (2) на основе интегрального

критерия, зависящего от функции переходного процесса на ступенчатое или иное воздействие, например, на вход системы. В этом случае для решения задачи необходимо задать зависимость целевой (стоимостной) функции от ошибки регулирования. Тогда функционалы оценки качества формируем в виде:

1. Интеграл от суммы модулей ошибки:

$$\Psi_1 = \int_0^T [|e_1(t)| + |e_2(t)|] dt, \quad (3)$$

2. Интеграл от суммы модулей ошибки, умноженной на время:

$$\Psi_2 = \int_0^T [|e_1(t)| + |e_2(t)|] t dt, \quad (4)$$

3. Интеграл от суммы функций, зависящих от ошибки и от перерегулирования:

$$\Psi_3 = \int_0^T [|r_1(t)| + |r_2(t)|] t dt, \quad (5)$$

где  $r_1$  и  $r_2$  формируются с помощью соотношений:

$$r_i = |e_i(t)| + 100 \max\{0, [y_i(t) - C]\}. \quad (6)$$

Величина перерегулирования задается константой  $C$ : а)  $C = 1,0001$ ; б)  $C = 1,05$ ; в)  $C = 1,1$ ; г)  $C = 1,2$ . В общем случае  $C = 1 + \delta$ , где  $\delta$  – максимально допустимое перерегулирование.

### 4. ВЫБОР ПРОГРАММНЫХ СРЕДСТВ

Для сравнения будем использовать два различных программных средства: *MATLAB* и *VisSim* [9]. В программе *MATLAB* 7.11.0 оптимизация выполняется с помощью блока *Signal Constraint*, принцип действия которого основан на методе Монте-Карло [9]. В данном блоке задаются требования к виду переходного процесса, и по этому виду выполняется численная оптимизация параметров регулятора (1).

Начальная схема моделирования данной системы в программе *VisSim* 6.0 представлена на рис.2. Время симуляции взято 250 с (не менее утроенной длительности сумм всех постоянных времени).



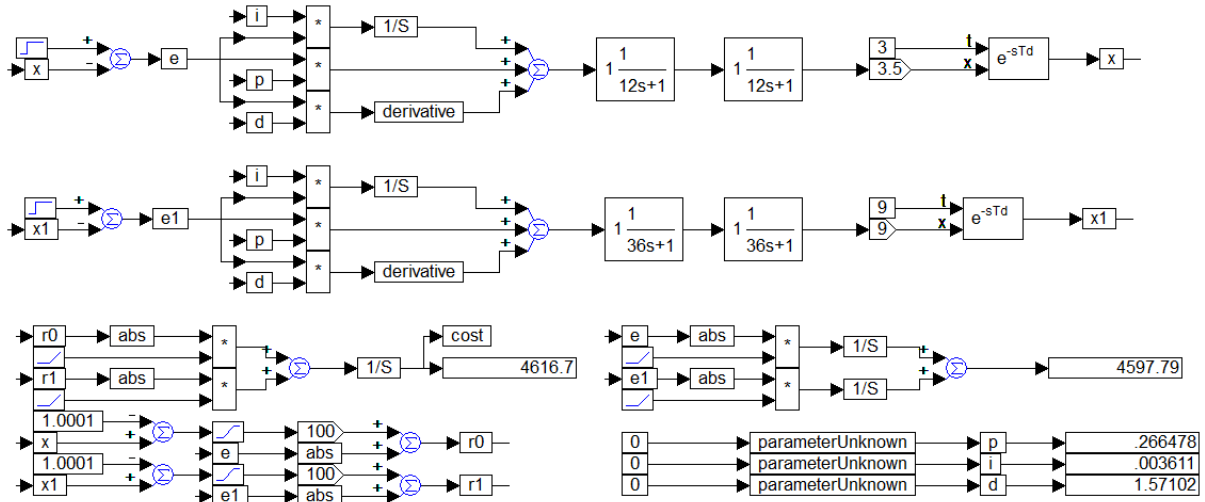


Рис.2. Схема моделирования в ПО VisSim 6.0

### 5. РЕЗУЛЬТАТЫ ОПТИМИЗАЦИИ ПО ДВУМ ПАРАЛЛЕЛЬНЫМ СИСТЕМАМ С ДВУМЯ ИЗМЕНЯЮЩИМИСЯ ПАРАМЕТРАМИ

Зафиксируем значения  $T_1 = T_2 = 24$  с. Также зафиксируем  $C = 1.0001$ . Осуществим оптимизацию регулятора для двух объектов с крайними значениями параметров: при наименьших значениях коэффициента усиления передаточной функции и постоянной времени звена запаздывания ( $k = 3,5; T_d = 3$ ); при наибольших значениях коэффициента усиления передаточной функции и постоянной времени звена запаздывания ( $k = 9; T_d = 9$ ). Далее, осуществляя девиацию варьируемых параметров в различных направлениях, определим качество переходных процессов с оптимизируемыми регуляторами. Время симуляции взято 200 с (не менее утроенной длительности сумм всех постоянных времени).

На рис. 3–5 представлены переходные процессы с различными регуляторами, рассчитанными в программе VisSim 6.0, а на рис. 6–8 – с регуляторами, рассчитанными в программе MATLAB 7.11.0. На рисунках приняты следующие обозначения:  $x$ :  $k = 3,5, T_d = 3$ ;  $x_1$ :  $k = 9, T_d = 9$ ;  $x_2$ :  $k = 3,5, T_d = 9$ ;  $x_3$ :  $k = 9, T_d = 3$ ;  $x_4$ :  $k = 6, T_d = 6$ , где  $x_i$  – выход системы с соответствующими варьируемыми параметрами.

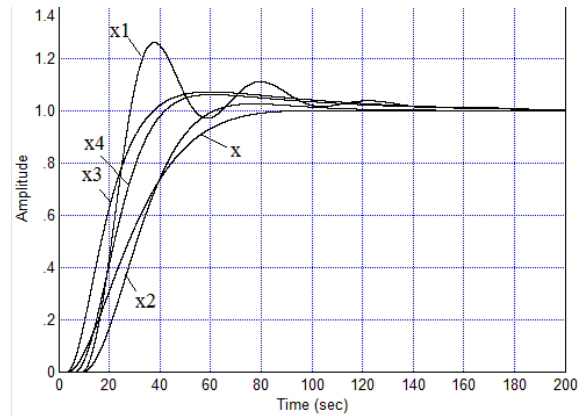


Рис. 3. Переходные процессы объекта (1) с регулятором настроенным по критерию качества  $\Psi_1$

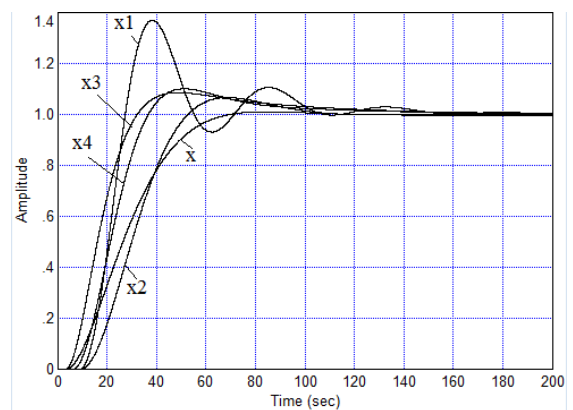


Рис. 4. Переходные процессы объекта (1) с регулятором настроенным по критерию качества  $\Psi_2$

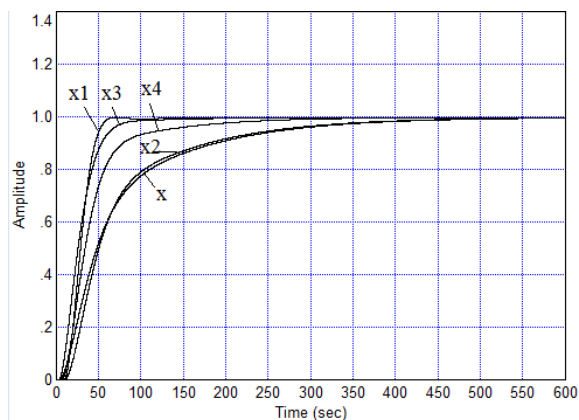


Рис. 5. Переходные процессы объекта (1) с регулятором настроенным по критерию качества  $\Psi_3$

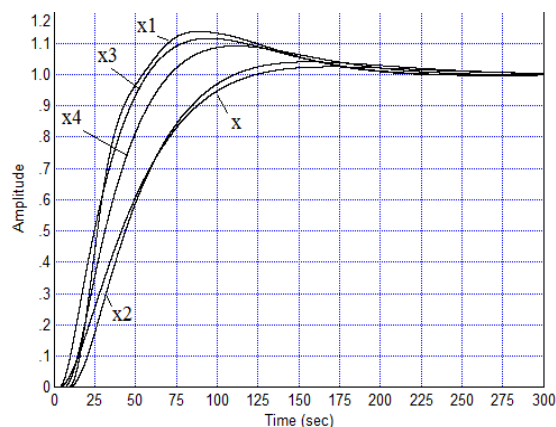


Рис. 7. Переходные процессы объекта (1) с регулятором настроенным методом **DO-II**

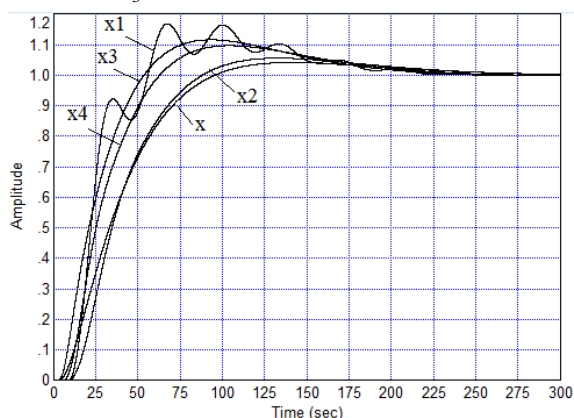


Рис. 6. Переходные процессы объекта (1) с регулятором настроенным методом **DO-I**

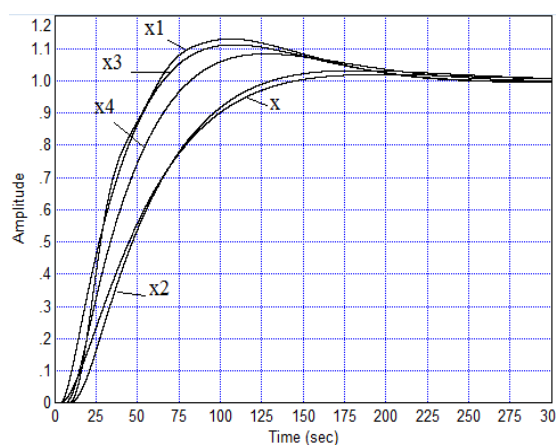


Рис. 8. Переходные процессы объекта (1) с регулятором настроенным методом **DO-III**

Из рис. 3–8 видно, что во всех рассмотренных случаях управление остается устойчивым. Однако рассмотрены далеко не все варианты значений параметров. При неблагоприятном сочетании параметров объектов переходные процессы могут стать неустойчивыми. Поэтому необходимо рассматривать все возможные сочетания. Поскольку это практически невозможно, следует, как минимум, рассмотреть все крайние значения из области допустимых значений этих параметров, а также их среднее значение.

## 6. РАСШИРЕНИЕ ВАРИАНТОВ СОЧЕТАНИЯ ПАРАМЕТРОВ

Рассмотрим все крайние исследуемые варианты параметров объекта, которые представлены в Таблице 1. Для стартовой оптимизации используем значения параметров, представленных вариантами  $x$  и  $x_1$ . Кроме того, целесообразно рассмотреть несколько вариантов значений  $C$ , соответствующих перерегулированию от 0 до 20%.

Таблица 1. Варианты параметров объекта

	$x$	$x_1$	$y_{11}$	$y_{12}$	$y_{13}$	$y_{21}$	$y_{22}$	$y_{23}$	$y_{31}$	$y_{32}$	$y_{33}$	$y_{34}$	$y_4$
$k$	3,5	9	3,5	3,5	9	9	9	3,5	3,5	3,5	9	9	6
$T_d$	3	9	3	9	3	9	3	9	3	9	9	3	6
$T_1$	12	36	12	12	12	12	36	36	36	12	12	12	24
$T_2$	12	36	36	12	12	36	36	36	36	36	12	36	24

Одновременная оптимизация регулятора для всех возможных сочетаний параметров объектов излишне трудоемка. Поэтому предлагается на первом этапе осуществить оптимизацию регулятора для малого числа объектов с крайними значениями параметров. Далее в системе с полученными регуляторами осуществим перебор всех вариантов параметров объектов и выявим те сочетания параметров, которые наиболее отрицательно сказываются на устойчивости системы и качестве переходных процессов. Далее будем использовать эти варианты для следующей итерации совместной оптимизации регулятора.

В Таблице 2 представлены результаты оптимизации параметров ПИД-регулятора. На рис. 3–6 представлены переходные процессы в объектах с параметрами  $x$  и  $x_1$  из таблицы 1 и полученными регуляторами, рассчитанными оптимизацией в программе VisSim 6.0.

Таблица 2. Параметры регуляторов

Критерий качества	Параметры ПИД регулятора		
	$K_p$	$K_I$	$K_D$
$\Psi_1$	1.2098	0.0156	7.8468
$\Psi_2$	1.0273	0.0157	6.0002
$\Psi_3(a)$	0.2807	0.0025	1.9542
$\Psi_3(б)$	0.8876	0.0043	10.8686
$\Psi_3(в)$	1.0632	0.007	8.552
$\Psi_3(г)$	0.9104	0.0126	7.0024
$\Psi_3(г, б)$	0.293	0.0044	1.563

На рис. 9–14 представлены переходные процессы с различными регуляторами, рассчитанными в программе VisSim 6.0. На рисунках приняты следующие обозначения:  $x$ :  $k = 3.5$ ,  $T_d = 3$ ,  $T_1 = T_2 = 12$ ;  $x_1$ :  $k = 9$ ,  $T_d = 9$ ,  $T_1 = T_2 = 36$ , где  $x_i$  – выход системы с соответствующими варьируемыми параметрами. Из рис. 9–14 видно, что система является устойчивой при крайних варьируемых параметрах системы, но чтобы выявить является ли система робастной с рассчитанными регуляторами, необходимо исследовать ее поведение при девиации параметров в соответствии с вышеизложенной схемой. Необходимо только знать, каким требованиям должна удовлетворять система и в соответствии от этих требований выбрать одну из систем, переходные процессы которых представлены на рис. 9–14. В случае применения критериев  $\Psi_1$  и  $\Psi_2$  обеспечено высокое быстродействие (время переходного процесса остается в пределах

180 с), но перерегулирование достигает 40%. При использовании критерия  $\Psi_3(a)$  перерегулирование остается пренебрежимо малым, но время переходного процесса достигает 700 с. Критерии  $\Psi_3(б)$  и  $\Psi_3(в)$  существенно не улучшили ситуацию и время переходного процесса соответственно составило 600 с и 400 с. Использование критерия  $\Psi_3(г)$  позволило ускорить переходные процессы в системе до 200 с при 20% перерегулировании. Допустим, заявленным требованиям удовлетворяет система, с параметрами регулятора, оптимизируемыми по критерию  $\Psi_3(г)$ . Необходимо дальнейшее исследование системы с целью определения ее робастности. На рис. 15–18 представлены переходные процессы системы с регулятором, оптимизируемым по критерию  $\Psi_3(г)$  при девиации параметров.

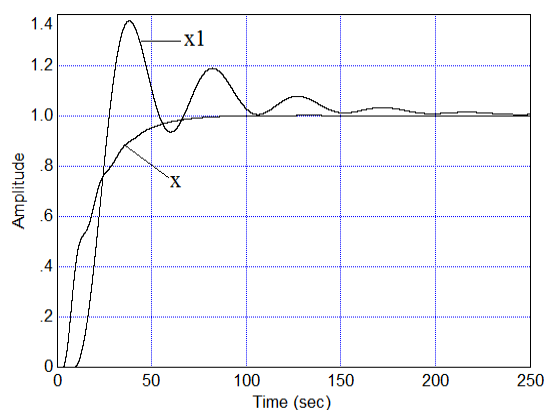


Рис.9. Переходные процессы объекта с регулятором по критерию  $\Psi_1$

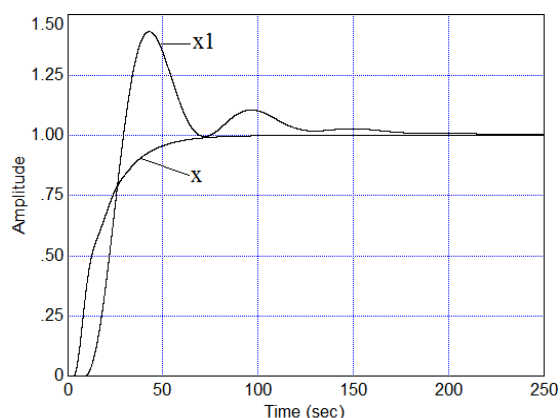


Рис.10. Переходные процессы объекта с регулятором по критерию  $\Psi_2$

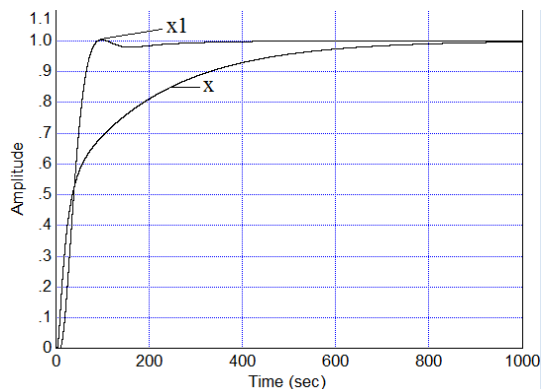


Рис.11. Переходные процессы объекта с регулятором по критерию  $\Psi_3(a)$

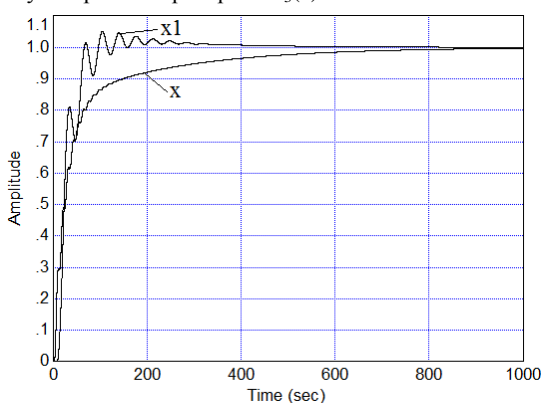


Рис.12. Переходные процессы объекта с регулятором по критерию  $\Psi_3(б)$

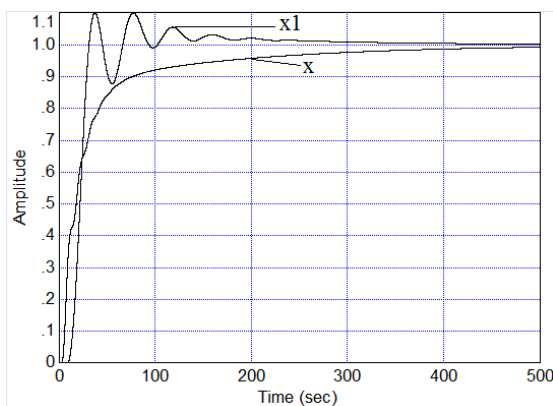


Рис.13. Переходные процессы объекта с регулятором по критерию  $\Psi_3(в)$

Из рис. 15–17 видно, что система выходит из устойчивого состояния для случаев  $y_{12}$ ,  $y_{13}$ ,  $y_{21}$  и  $y_{23}$ , а значит, данный регулятор не является робастным, поэтому ансамбль объектов следует дополнить моделями с параметрами объектов, которые вывели систему из состояния устойчивости.

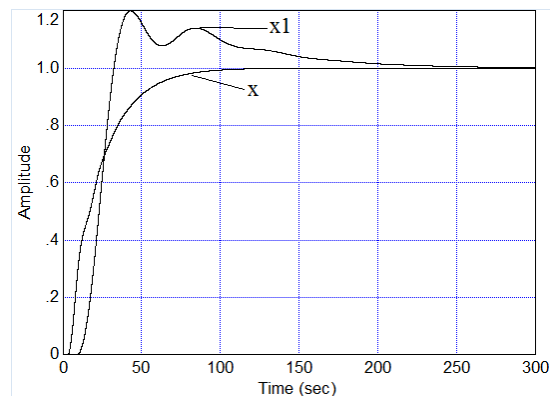


Рис.14. Переходные процессы объекта с регулятором по критерию  $\Psi_3(г)$

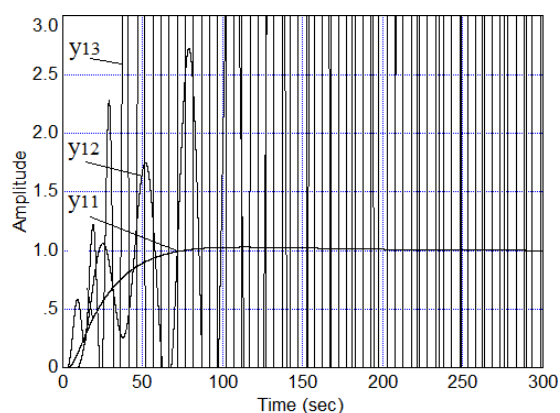


Рис. 15. Переходные процессы объекта с регулятором по критерию  $\Psi_3(г)$

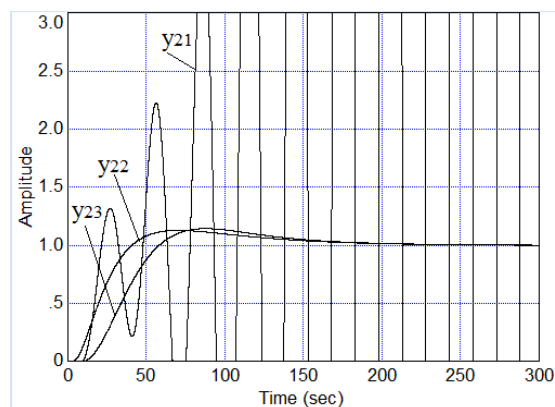


Рис. 16. Переходные процессы объекта с регулятором по критерию  $\Psi_3(г)$

Дополняя наборную схему моделирования системами с объектами, принимающими эти критические значения, используя критерий  $\Psi_3(г, б)$ , то есть используется критерий с допустимым перерегулированием 20% и моделируется одновременно 6 систем с различными 6

сочетаниями параметров объектов, получим новый результат оптимизации регулятора (см. последнюю строку в таблице 2). Шесть сочетаний параметров объектов получено добавлением четырех выявленных проблемных сочетаний к исходным двум сочетаниям. На рис. 18 приведены графики переходных процессов с полученным регулятором.

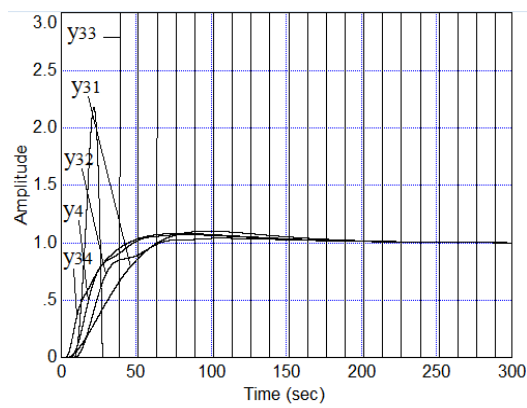


Рис. 17. Переходные процессы объекта с регулятором по критерию  $\Psi_3(\gamma)$

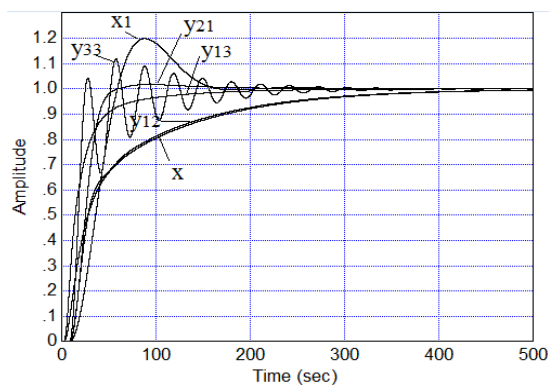


Рис. 18. Переходные процессы объекта с регулятором по критерию  $\Psi_3(\gamma)$ , уточненному по шести параллельным моделям объекта, т.е. по критерию  $\Psi_3(\gamma, \delta)$

## 7. ДЕТАЛЬНОЕ ИССЛЕДОВАНИЕ И ДАЛЬНЕЙШАЯ ОПТИМИЗАЦИЯ

На рис. 18 представлены переходные процессы только с пятью вариантами сочетаний параметров объекта. На самом деле таких сочетаний гораздо больше. В таблице 1 дано 13 вариантов, из которых последний представляет собой набор средних значения всех параметров, а первые 12 вариантов – сочетания крайних значений.

Целесообразно поставить вопрос о том, насколько полно данная выборка представляет все возможные варианты сочетаний, которых может быть бесконечное множество.

Можно предположить, что конкретные значения

постоянных времени минимально-фазовой части модели объекта не столь существенны, как их сумма, то есть вместо рассмотрения в отдельности  $T_1$  и  $T_2$  можно рассмотреть их сумму. Это предположение требует проверки. Также целесообразно решить вопрос, может ли быть понижена величина перерегулирования.

Предлагается следующий метод решения поставленных вопросов.

1. Моделирование систем со всеми вариантами сочетаний крайних значений параметров. Наборы параметров представлены в Таблице 3.

2. К этим вариантам добавим вариант, содержащий средние значения постоянных времени, (колонка Г в Таблице 3) и сравним результат с результатом моделирования при этом же значении их суммы, когда одна из этих величин принимает минимальное значение, а вторая – максимальное значение (колонка С).

3. Для удобства рассмотрения будем объединять графики по четыре, номера соответствующих рисунков даны в строке «Рис» Таблицы 3, соответствие цвета и параметров объекта указаны в колонках таблицы 3 (К – красный, С – синий, Г – голубой, Ч – черный).

4. Результаты моделирования показаны на рис. 19 – 23.

Таблица 3. Параметры объекта к рис. 19 – 26

	К	С	Г	Ч	К	С	Г	Ч
$k$	3,5	3,5	3,5	3,5	3,5	3,5	3,5	3,5
$T_d$	3	3	3	3	9	9	9	9
$T_1$	12	12	24	36	12	12	24	36
$T_2$	12	36	24	36	12	36	24	36
Рис	19, 23				20, 24			
$k$	9	9	9	9	9	9	9	9
$T_d$	3	3	3	3	9	9	9	9
$T_1$	12	12	24	36	12	12	24	36
$T_2$	12	36	24	36	12	36	24	36
Рис	21, 25				22, 26			

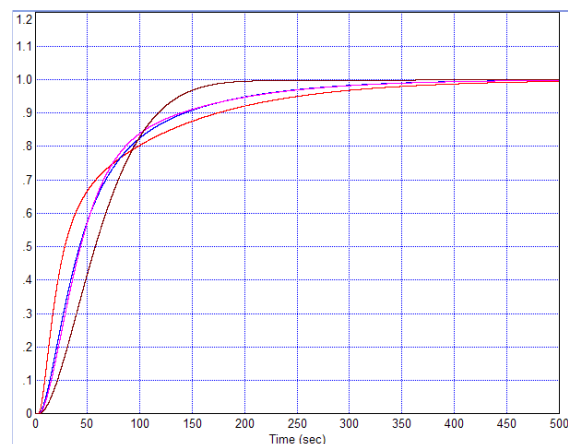


Рис. 19. Переходные процессы объекта с регулятором  $\Psi_3(\gamma, \delta)$  и параметрами из табл. 3



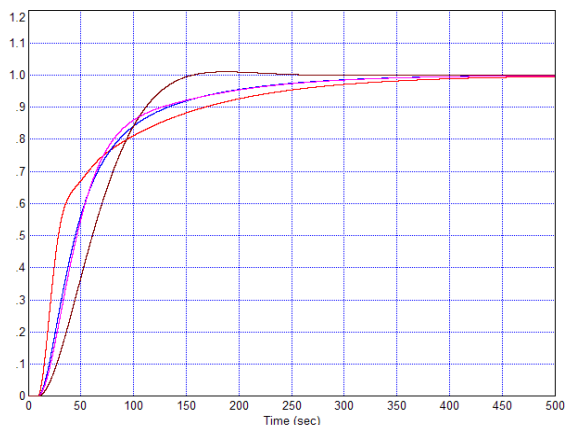


Рис. 20. Переходные процессы объекта с регулятором  $\Psi_3(\gamma, \delta)$  и параметрами из табл. 3

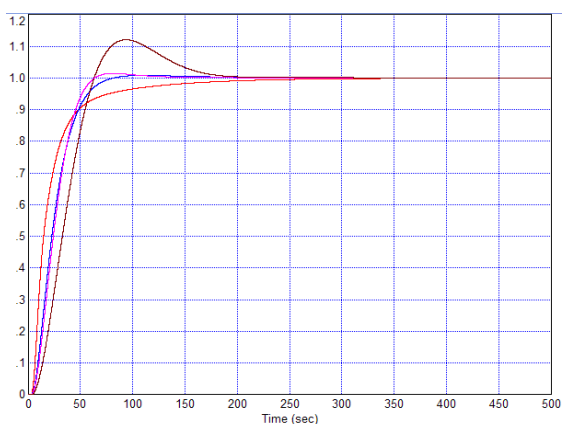


Рис. 21. Переходные процессы объекта с регулятором  $\Psi_3(\gamma, \delta)$  и параметрами из табл. 3

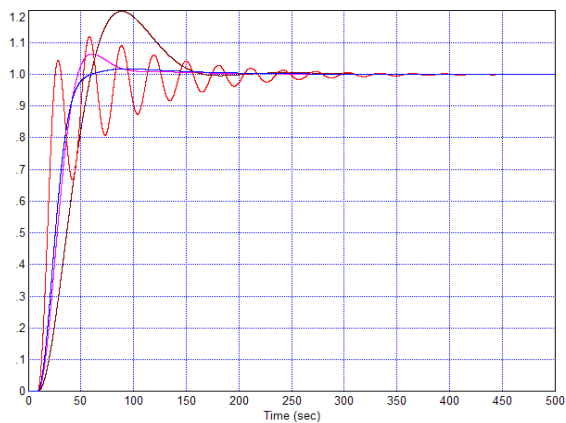


Рис. 22. Переходные процессы объекта с регулятором  $\Psi_3(\gamma, \delta)$  и параметрами из табл. 3

Из анализа рис. 19–22 можно сделать следующие выводы:

1. Предположение, что наиболее важна лишь сумма постоянных времени  $T_1$  и  $T_2$  подтвердилось, соответствующие графики почти везде сливаются, кроме рис. 22, где голубой график характеризуется перерегулированием около 5%, а синий график имеет перерегулирование менее 1%.

2. Наиболее опасными с позиции потери устойчивости являются соотношения параметров,

показанные на рис. 22 черным и красным графиком. Эти наборы параметров характеризуются максимальными значениями запаздывания и коэффициента усиления и крайними значениями суммы постоянных времени минимально-фазовых частей модели объекта. В единственном случае перерегулирование достигает 20% (рис. 22).

3. Наименьшее быстродействие получается при наименьшем значении коэффициента усиления объекта (рис. 19–20).

На этом основании можно предложить попытаться снизить перерегулирование, например, до 5–15 %, причем, для окончательного проектирования регулятора применить следующие меры:

1. Задать в критерии (6) соответствующее значение константы  $C = 1,05$  или  $C = 1,15$ .

2. Использовать лишь самые неблагоприятные сочетания параметров объектов, а именно: максимальные значения коэффициента усиления и запаздывания, и крайние значения суммы  $T_1$  и  $T_2$ .

3. Для обеспечения поиска коэффициентов в ограниченной области (и предотвращения фатальной ошибки, вызванной чрезмерным возрастанием какого-либо коэффициента) целесообразно в критерий ввести дополнительный член, принимающий большое значение в случае недопустимого возрастания любого коэффициента регулятора. С этой целью используются нелинейности типа «зона нечувствительности» с граничными значениями  $\pm 20$ , последовательно с ними включенным выпрямителем и сумматором с большим коэффициентом (100), результат добавляется под интеграл, вычисляющий функцию (5), в этом случае получаем новую функцию:

$$\Psi_4 = \int_0^T [ |r_1(t)|t + |r_2(t)|t + q ] dt \quad (7)$$

$$q = R(k_p) + R(k_I) + R(k_D) \quad (8)$$

$$R(k_N) = 100 \max \{ |k_N| - 100, 0 \} \quad (9)$$

Схема оптимизации в программе *VisSim* показана на рис. 23. Результат окончательной оптимизации регулятора представлен в Таблице 4. Переходные процессы приведены на рис. 24 – 27.

Таблица 4. Итоговые параметры регулятора

Критерий качества	Параметры ПИД регулятора		
	$K_p$	$K_I$	$K_D$
$\Psi_4$	0.275	0.00374	1.461

По результатам можно сделать следующие выводы:

1. Снижение перерегулирования ниже 15% обеспечено для всех сочетаний крайних значений параметров объекта при использовании регулятора из Таблицы 4.

2. Использование только двух моделей объектов (после выявления наиболее критических сочетаний параметров) оказалось достаточным для получения надежного результата.

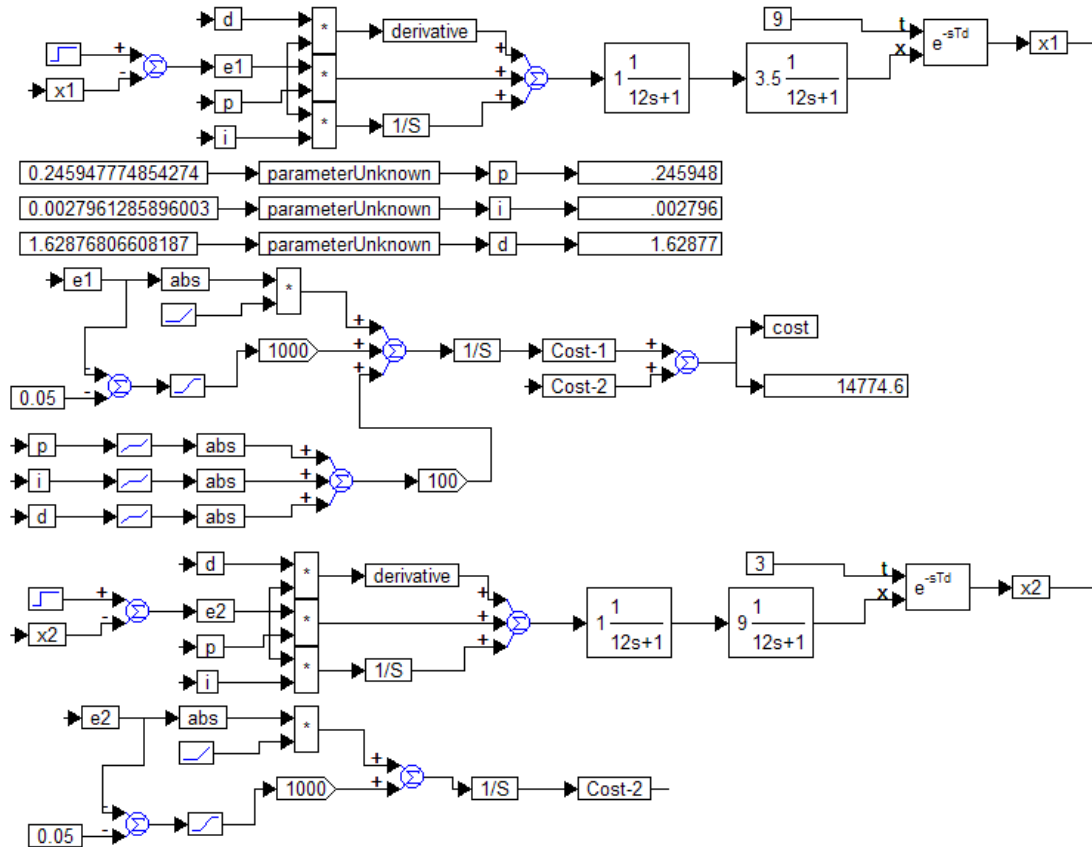


Рис. 23. Схема для численной оптимизации регулятора  $\Psi_4$

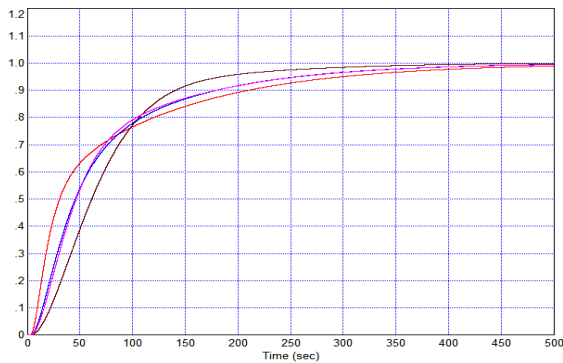


Рис. 24. Переходные процессы объекта с параметрами из табл. 3 и регулятором  $\Psi_4$  (табл.4)

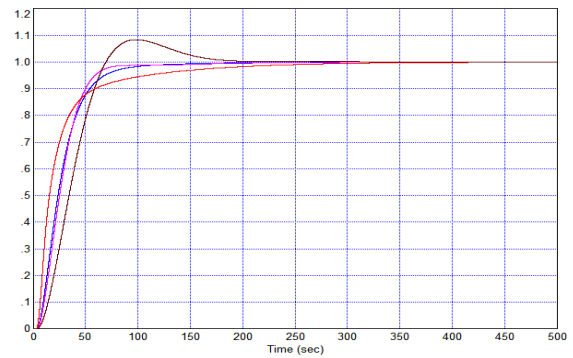


Рис. 26. Переходные процессы объекта с регулятором  $\Psi_3(\gamma, 6)$  и параметрами из табл. 3

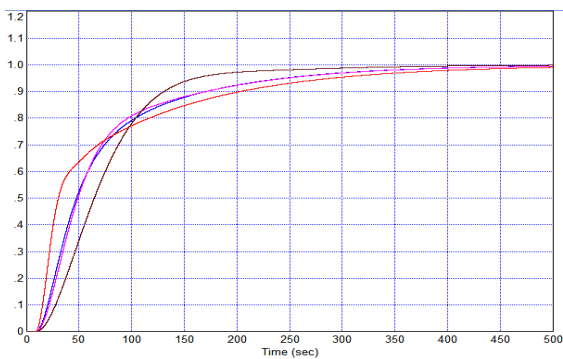


Рис. 25. Переходные процессы объекта с регулятором  $\Psi_3(\gamma, 6)$  и параметрами из табл. 3

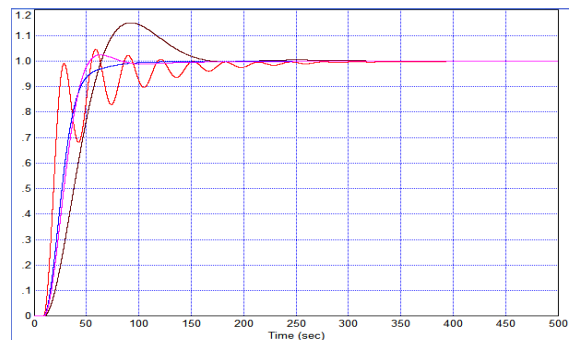


Рис. 27. Переходные процессы объекта с регулятором  $\Psi_3(\gamma, 6)$  и параметрами из табл. 3

## ЗАКЛЮЧЕНИЕ

Отыскание критических наборов параметров объекта с включением этих наборов в ансамбль объектов, используемых для оптимизации общего регулятора, дают методику численной оптимизации для расчета робастного регулятора, обеспечивающего астатическое регулирование с перерегулированием не более 15% и минимальной длительностью переходного процесса в этих условиях (400 с).

Работа выполнена по заданию Министерства образования и науки РФ, проекты № 7.559.2011 и ГК № П761.

## СПИСОК ЛИТЕРАТУРЫ

[1] Методы классической и современной теории автоматического управления: учебник в 5 т. / Под ред. **К. А. Пупкова, Н. Д. Егупова**. М.: Изд-во НГТУ им. Н. Э. Баумана. 2004. т. 5. Методы современной теории автоматического управления.

[2] **Никифоров В. О.** Адаптивное и робастное управление с компенсацией возмущений — СПб: Наука. — 282 с.

[3] Методы классической и современной теории автоматического управления: учебник в 5 т. / Под ред. **К. А. Пупкова, Н. Д. Егупова**. М.: Изд-во НГТУ им. Н. Э. Баумана. 2004. т. 3. Синтез регуляторов систем автоматического управления.

[4] **Doyle J.C., Glover K., Khargonekar P.P., Francis B.A.** State-space solutions to standart  $H_2$  and  $H_\infty$  control problems // IEEE Transactions on Automatic Control AC-34, №8, 1989.

[5] Динамические характеристики энергоблока 180 МВт. Котлоагрегат ТПЕ-214 (Еп-670-13,8-545КТ). Турбина паровая Т-180/210-130. ОАО «НовосибирскЭнерго», Новосибирск, 2001.

[6] **V.A. Zhmud, O.D. Yadryshnikov, A.N. Zavorin** The study of the influence of high-frequency part of logarithmic frequency response curve to the quality of the feedback controls system // Proceedings of DST-RFBR-Sponsored Second Indo-Russian Joint Workshop on Computational Intelligence and Modern Heuristics in Automation and Robotics. NSTU, Novosibirsk, Russia. 9th – 12th September 2011. Новосибирск, НГТУ. pp. 97–101.

[7] **V. Zhmud, O. Yadryshnikov, A. Zavorin, E. Prokhorenko, R.V. Rao.** The Improvement Of The Qualitative Characteristics Of Control Systems With Feedback At Use  $PI^2D^2$ -Regulator // Proceedings of the Fifth International

Conference on Advances in Mechanical Engineering (ICAME-2011), June 06-08, 2011. Surat – 395 007, Gujarat, India. pp. 280-287.

[8] **Воевода А.А., Жмудь В.А.** Сходимость алгоритмов оптимизации регулятора для объекта с ограничителем и запаздыванием. Научный вестник НГТУ, 2007. № 4(29). С. 179 – 184.

[9] **Дьяконов В.П.** VisSim+Mathcad+MATLAB. Визуальное математическое моделирование. – М.:СОЛОН-Пресс, 2004. – 384с.



Вадим Жмуд – заведующий кафедрой Автоматики в НГТУ, профессор, доктор технических наук, автор более 200 научных статей, включая 10 патентов и 6 учебных пособий. Область научных интересов и компетенций – теория автоматического управления, электроника, лазерные системы, оптимизация, измерительная техника.

E-mail: [oao\\_nips@bk.ru](mailto:oao_nips@bk.ru)

Олег Ядрышников, аспирант кафедры Автоматики НГТУ, автор более 10 научных статей. Область научных интересов и компетенций – теория автоматического управления, оптимальные и адаптивные системы, оптимизация, многоканальные системы.

E-mail: [oleg\\_yadr@mail.ru](mailto:oleg_yadr@mail.ru)



Александр Заворин, аспирант кафедры Автоматики НГТУ, автор более 10 научных статей. Область научных интересов и компетенций – теория автоматического управления, оптимальные и адаптивные системы, оптимизация, многоканальные системы.

E-mail: [zavorin@ngs.ru](mailto:zavorin@ngs.ru)



Александр Полищук, аспирант кафедры Автоматики НГТУ, автор 10 научных статей. Область научных интересов и компетенций – теория автоматического управления, оптимальные и адаптивные системы, оптимизация, многоканальные системы.

E-mail: [aleksandrpolishchuk@gmail.com](mailto:aleksandrpolishchuk@gmail.com)

# Радиочастотный метод измерения сверхмалых перемещений и вибраций

*В.А. Жмудь, Д.О. Терешкин, А.В. Ляпидевский, А.В. Захаров  
НГТУ, ОАО «НИПС»*

**Аннотация:** Описано устройство измерительной техники, которое может использоваться в геодезии, строительстве, системах контроля состояния сложных инженерных сооружений: ГЭС, плотин, мостов и др. для выполнения высокоточных бесконтактных измерений.

**Ключевые слова:** измерения перемещений, измерения вибраций.

## ВВЕДЕНИЕ

Для высокоточных бесконтактных измерений малых перемещений и вибраций применяют методы, основанные на известной скорости распространения оптических волн или радиоволн. Оптические устройства, чаще всего лазерные, характеризуются более высокой точностью, но требуют прямой видимости объекта и отсутствия помех оптического характера, включающих резкие перепады, влажности или давления, создающие дифракционные неоднородности. Радиотехнические измерители характеризуются, как правило, меньшей точностью, но большей дальностью, а также чувствительностью к отражающим свойствам объекта и его форме и высокой зависимостью результатов от проводимости окружающих предметов. Анализ показал значительное преимущество радиочастотных измерителей для целей непрерывного мониторинга малых перемещений и вибраций в натуральных условиях, однако, эксперименты с первыми образцами таких устройств не позволили достичь желаемой точности измерений. В связи с этим предложены и анализируются усовершенствованные методы и аппаратура для измерения сверхмалых перемещений и вибраций радиотехническим путем вне помещений на расстояниях более 1 км.

## 1. ОБСУЖДЕНИЕ АНАЛОГОВ И ПРОТОТИПОВ

Измерительные устройства преобразуют измеряемое линейное расстояние во время задержки отраженного сигнала, которое затем измеряется специальными электронными

средствами – измерителями интервалов или фаз. Недостатком оптических измерительных устройств при их использовании вне помещений является зависимость точности измерений от метеоусловий, влияющих на скорость распространения оптических волн в воздухе, а также необходимость прямой оптической видимости между источником излучения и приемником излучения. Недостатком радиотехнических измерительных устройств является сложность конструирования источника узконаправленного излучения и отражателя радиоволн и сложность настройки таких устройств, а также возникновение помех от сторонних объектов, которые в некоторых случаях невозможно устранить.

Решение задачи непрерывного мониторинга состояния сложных инженерно-технических сооружений, например, плотин, может быть решено одновременным использованием измерителей всех типов. При этом оптический измеритель можно использовать для контроля по оптическим путям, обеспеченным первоначальным проектированием и проходящим внутри самой конструкции.

Зависимость результата измерения от состояния атмосферы редко обсуждается в литературе, однако, этот недостаток является существенным для оптических измерителей. Рассмотрим, например, оптический измеритель линейных перемещений, схема которого приведена на рис.1 [1]. В этом измерителе излучение от источника оптического излучения через передающую оптическую схему поступает в среду распространения излучения, где распространяется до объекта и обратно к приемной оптической системе. Фотоприемник преобразует оптический сигнал в электрический, который поступает на средство обработки, где определяются соотношения фаз, частот, или иных характеристик принятого сигнала, по которым вычислительное средство вычисляет расстояние от измерителя перемещений до объекта. На прохождение через среду в прямом и обратном направлении оптическое излучение затрачивает некоторое время вследствие конечной скорости распространения света в этой

среде. Измерение этой задержки позволяет измерять расстояние до объекта. Для измерения этой задержки может использоваться сравнение с сигналом, поступающим на этот же или на дополнительный фотоприемник, минуя путь к объекту или обратно. Или же в сигнале, распространяющемся к объекту и обратно, могут содержаться несколько модуляционных частот. Одинаковая задержка по времени в

сигналах на различных частотах приводит к разным фазовым сдвигам в принимаемых сигналах, что позволяет вычислить величину задержки во времени. Это вычисление осуществляет средство обработки, а средство вычисления преобразует ее выходные сигналы в форму, удобную для восприятия человеком или регистрирующим устройством.

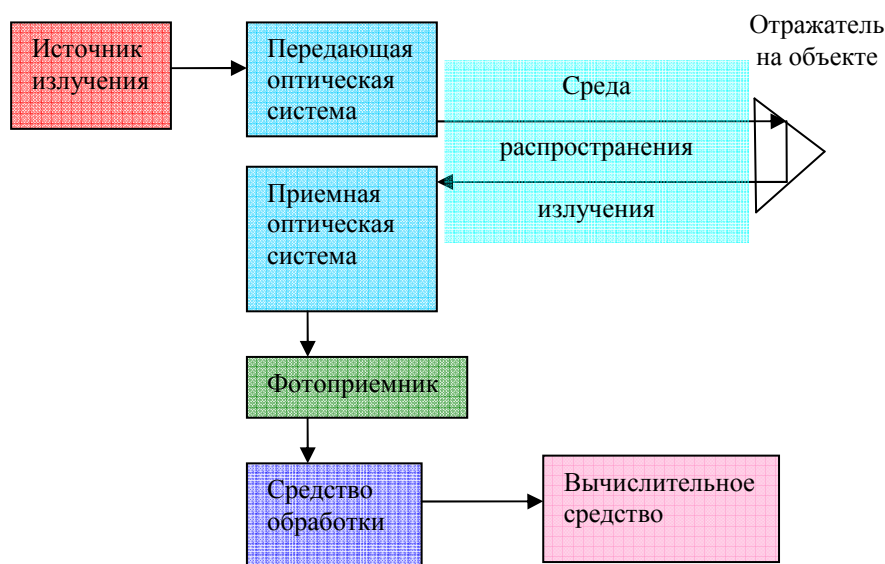


Рис. 1. Обобщенная схема оптического измерителя перемещений и вибраций [1]

Недостатком этого измерителя является зависимость результата измерения от метеоусловий, таких как давление и температура, изменяющих скорость света в среде, которой является воздух, что снижает точность измерений. Другим недостатком является необходимость прямой оптической видимости на всей измерительной длине между передающей и приемной системами.

Радиотехнический измеритель в идеале может быть прикреплен к контролируемым

частям конструкции и использоваться без дальнейших настроек в круглосуточном режиме работы. Такое изделие найдет широкое применение при обеспечении измерений с погрешностью менее 1 мм на расстояниях более 1 км. В этом случае оно превзойдет оптические устройства по техническим характеристикам и по удобству эксплуатации.

Рассмотрим, например, линейных перемещений, схема которого приведена на рис.2 [2].

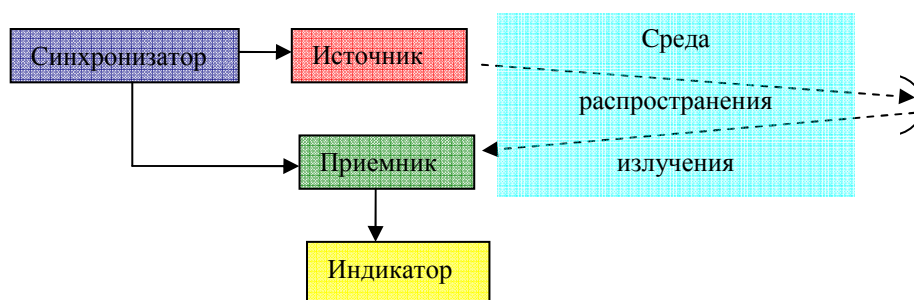


Рис. 2. Обобщенная схема радиотехнического измерителя перемещений [2]

Этот измеритель линейных перемещений

работает следующим образом. Синхронизатор



формирует связанные по времени сигналы на входы источника и приемника источника электромагнитных волн. Источник по сигналу синхронизатора со своей антенны посылает сигнал, часть которого отражается объектом и поступает на антенну приемника. Время поступления этого сигнала на приемник сравнивается со временем поступления сигнала от синхронизатора, и по запаздыванию во времени поступления отраженного сигнала по сравнению с сигналом синхронизатора  $\Delta T$  определяется расстояние до объекта, равно произведению скорости распространения локационного сигнала  $V$  на запаздывание  $\Delta T$ . Результат измерения отображается на индикаторе. Таким образом, определяется расстояние между местом размещения источника и приемника и объектом.

Недостатком этого устройства является

низкая точность измерений, обусловленная тем, что в нем используется сигнал радиотехнического диапазона, отраженный от объекта сложной формы, что вызывает искажение формы отраженного сигнала. Применение специального отражателя довольно сложно и при этом не достаточно эффективно, в связи с противоречивостью требований его узконаправленного действия и малого искажения фазы.

На рис. 3 показана схема другого радиотехнического измерителя перемещений, разработанного и исследованного нами, который можно считать прототипом предлагаемого устройства [3]. Далее на рисунках среда распространения излучения не показывается, поскольку понятно, что она всегда присутствует.

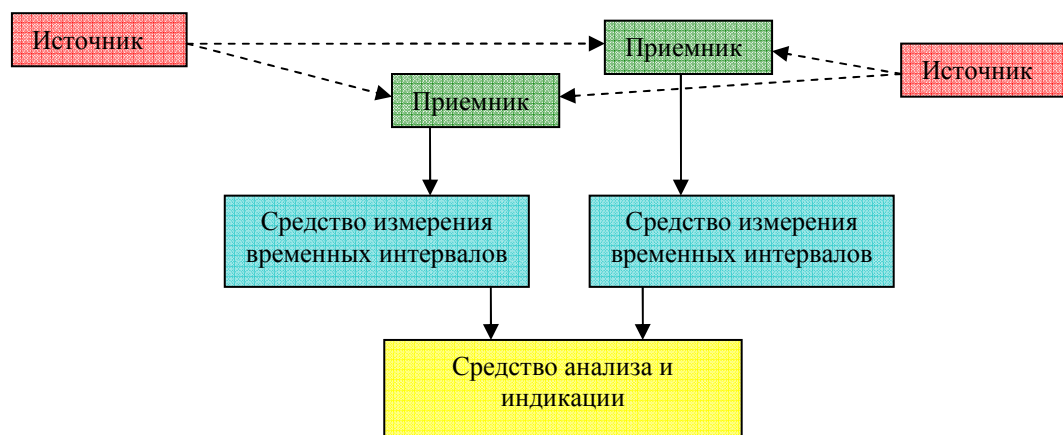


Рис.3. Радиотехнический измеритель перемещений [3]

Этот измеритель линейных перемещений работает следующим образом. Источники излучают радиосигналы, которые легко могут быть отличены друг от друга, причем каждый из таких сигналов содержит периодически появляющиеся признаки времени или фазы встроенного генератора или эталона времени. Каждый приемник воспримет сигнал каждого источника, различая их по характерным признакам. Предположительно, таким признаком отличия сигналов от разных передатчиков является различная несущая частота. Принятые сигналы вместе с содержащимися в них признаками времени поступают на средства измерения временных интервалов. Каждое из средств измерения временных интервалов определяет разность моментов этих признаков. Обозначим время обнаружения временного признака в сигнале от передатчика с номером  $N$  на приемнике с номером  $M$  как  $t_{NM}$ . Тогда, если приемники и

источники выстроены в прямую линию, начинающуюся первым источником, проходящую через первый приемник, затем через второй приемник и после этого через второй источник, то отличие разности моментов возникновения временных признаков  $\Delta t = (t_{11} - t_{21}) - (t_{21} - t_{22})$  пропорционально расстоянию между приемниками.

Эта разница зависит только от искомого расстояния между приемниками, а также от скорости распространения радиоволны в среде. Это позволяет вычислить величину искомого расстояния, не используя отражатели сигнала.

Таким образом, устройство позволяет измерять расстояние между антеннами двух приемников. В устройстве не требуется отражатель электромагнитных волн радиочастотного диапазона, что упрощает его конструкцию и удешевляет, устраняет зависимость результата измерения от качества такого отражателя. Это позволяет повысить

точность измерений. В результате достигается упрощение системы и повышение точности измерений.

Второй источник выполняется в виде ретранслятора сигнала от первого источника, который работает на другой несущей частоте. В этом случае признаком времени каждого сигнала может служить любая метка одинаковой огибающей. При этом оба приемника содержат два селекторных каскада, два демодулятора и в качестве средства измерения временных интервалов содержат дифференциальный фазометр.

Недостатком прототипа является недоста-

точно высокая точность измерений вследствие различного вклада разных приемных трактов приемников, поскольку каждый из приемников осуществляет прием двух сигналов на различных несущих частотах. По этой причине для приема сигналов от разных приемников применяются разные электронные аналоговые тракты, содержащие селекторные каскады, узкополосные фильтры, усилители или иные узлы, настраиваемые на различные частоты. Задержки сигналов в этих каскадах могут существенно отличаться, и эти отличия изменяются со временем, а также в зависимости от температуры и других внешних факторов.

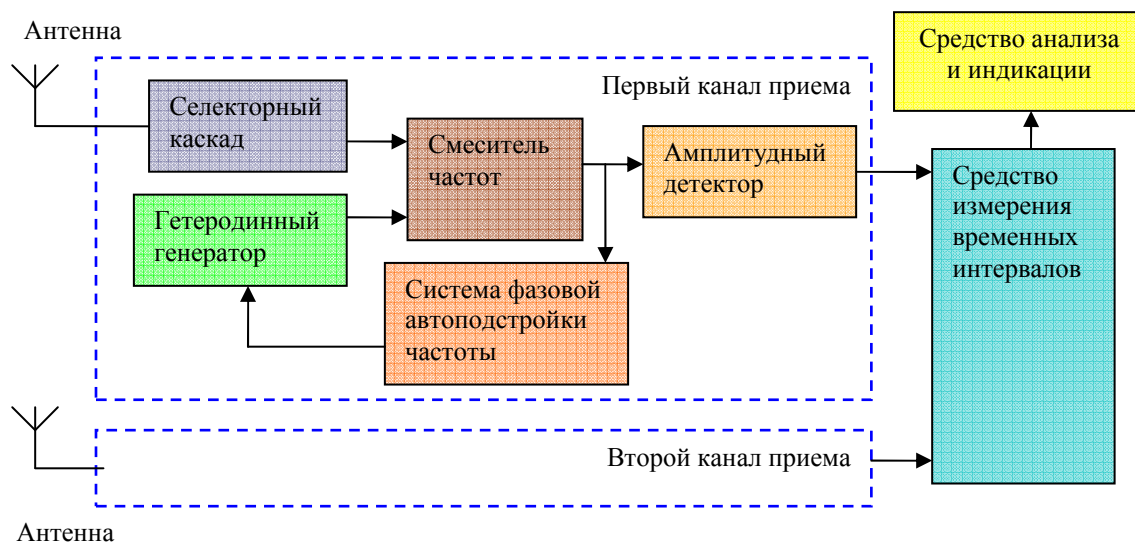


Рис.4. Пример радиоприемного тракта измерителя перемещений [3]

На рис. 4 показан пример радиоприемного тракта измерителя перемещений. Как правило, в таком тракте используется гетеродинный генератор, частота которого с помощью системы фазовой автоподстройки подстраивается к частоте принимаемого сигнала, что позволяет повысить чувствительность системы. При этом фаза обрабатываемого далее сигнала зависит не только от фазы принятого сигнала, но и от фазы гетеродинного генератора. Поскольку в каждом приемном тракте на различных несущих частотах используются разные гетеродинные генераторы, их фазы не совпадают, что вносит дополнительную ошибку в результат измерений. Таким образом, дальнейшее повышение точности требует устранения этого источника ошибки.

## 2. ПРЕДЛАГАЕМОЕ УСТРОЙСТВО И ПРИНЦИП ЕГО ДЕЙСТВИЯ

Кардинально данная проблема может быть решена лишь в том случае, если для приема двух сигналов от двух разных передатчиков используется полностью один и тот же тракт,

что возможно лишь в случае использования одной и той же несущей частоты. Однако в этом случае затруднительно разделить сигналы от разных передатчиков. Для решения этой проблемы предлагается осуществлять такую модуляцию, которая сохраняет свои временные признаки даже при сложении различных сигналов.

Схема предлагаемого измерителя показана на рис. 5. Поставленная задача решается тем, что предлагается использовать одну и ту же несущую частоту в обоих передатчиках, а возможность отличия принятых сигналов обеспечить путем использования уникальных модулирующих функций в каждом из передатчиков.

Предлагаемый измеритель работает следующим образом. Источники излучают радиосигналы на одной и той же несущей частоте, но с различными модулирующими функциями. Каждая из двух модулирующих функций (МФ) формируется как функция времени на выходе соответствующего генератора. Каждая МФ содержит периодически появляющиеся признаки времени (ПВ) от

встроенного или внешнего генератора или эталона времени. Эти генераторы воздействуют на входы источников излучения, осуществляя разную и хорошо отличимую модуляцию передаваемых в эфир сигналов, чья несущая частота совпадает. Эти сигналы могут быть приняты фотоприемниками только совместно в виде их суммы с коэффициентами и с задержками во времени, зависящими от расстояния между передатчиками и приемниками. В принимаемой смеси этих сигналов требуется отличить друг от друга их специфические признаки времени, для чего необходимы специальные методы, реализованные с помощью специализированных устройств, имеющихся в каждом тракте обработки на выходе приемника. Предварительно сигнал, принятый приемником, преобразуется в цифровую последовательность

его значений с помощью аналого-цифрового преобразователя. Поскольку прием смеси сигналов осуществляется в каждом приемнике единственным трактом, то все вносимые этим трактом задержки одинаковы для обоих сигналов этой смеси. Поэтому никакие изменения запаздывания в этих трактах, включая задержки в аналого-цифровых преобразователях, не будут вносить вклада в изменение разности времени между характерными ПВ от разных источников. Дальнейшая обработка сигналов осуществляется цифровым путем, поэтому все вносимые задержки при обработке сигналов могут быть точно определены и учтены в вычислении окончательного результата разности времен поступления ПВ от первого и второго источников.



Рис. 5. Схема предлагаемого измерителя перемещений

Для этой цели используются средства распознавания ПВ, показанные на рис. 6. Эти средства распознавания ПВ выявляют по отдельности ПВ двух компонент. Они работают следующим образом. В каждом канале имеется генератор, формирующий выходной сигнал,

изменяющийся во времени, который является точной копией соответствующей модулирующей функции, но время запуска этого генератора может управляться внешним сигналом.

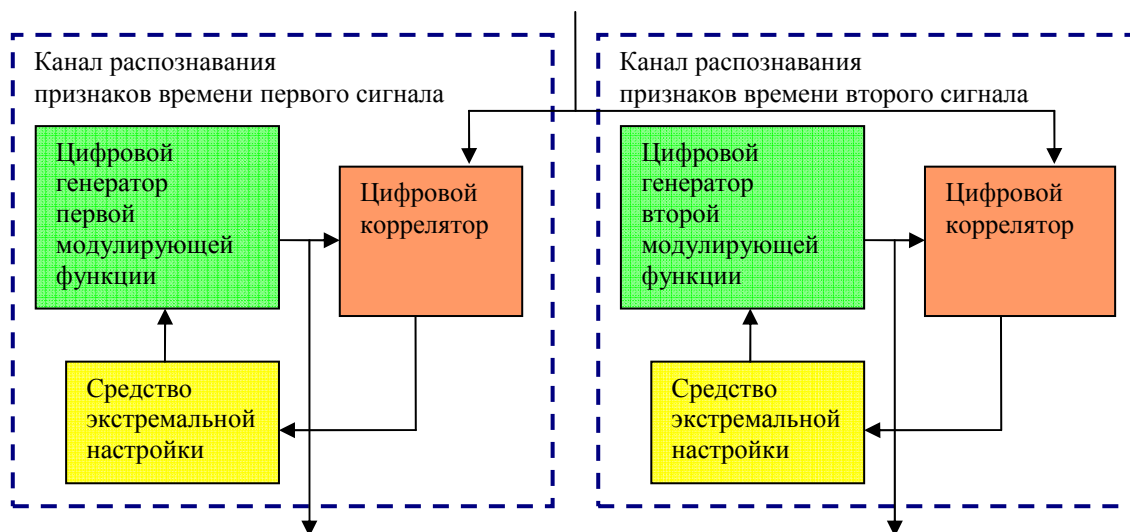


Рис. 6. Схема предлагаемого средства распознавания признаков времени

Выходной сигнал этого генератора поступает на один вход коррелятора, а на другой его вход поступает входной сигнал средства распознавания ПВ. Выходной сигнал коррелятора анализируется средством экстремальной настройки, которое на основе анализа этого сигнала изменяет время запуска генератора так, чтобы сигнал на выходе коррелятора был максимальный. В результате действия этого средства экстремальной настройки генератор оказывается настроенным синхронно компоненте с той же формой, которая имеется во входном сигнале средства распознавания ПВ. Поэтому выходной сигнал генератора может служить источником сигнала, содержащего признак времени, наилучшим образом настроенный на соответствующую компоненту в принятом приемником сигнале. Средство распознавания признаков времени может иметь один или два выхода. Если это средство имеет один выход, то какой-то заданный признак его выходного сигнала, например передний фронт импульса, отмечает момент появления признака времени в сигнале от первого источника, а другой заданный признак, например, задний фронт импульса, отмечает момент появления признака времени в сигнале от второго источника. Также средство распознавания признаков времени может иметь два отдельных выхода, в этом случае на первом выходе определенный признак, например передний фронт импульса, отмечает момент появления признака времени в сигнале от первого источника, а на втором выходе такой же признак отмечает момент появления признака времени в сигнале от второго источника.

Практическая реализация этого устройства

может быть осуществлена на быстром микропроцессоре с соответствующей программой или на специализированном цифровом вычислительном устройстве на основе программируемых логических матриц. Средствами измерителей временных интервалов, как и в прототипе, могут служить счетчики времени или фазометры при условии обеспечения требуемой точности. Средством анализа и индикации, как и в прототипе, может служить персональный компьютер, оснащенный соответствующими средствами связи и программой для обработки сигналов. Источники сигналов могут быть выполнены как обычные радиопередатчики, использующие амплитудную, фазовую или иную модуляцию. При этом любой или каждый источник может содержать тракт автоматической подстройки несущей частоты к какой-либо внешней частоте или к ее кратной доле путем применения традиционной фазовой автоподстройки. Это позволяет обеспечить точное равенство несущих частот, что обеспечит точное равенство временных задержек, вносимых всеми аналоговыми элементами приемников. Генератор каждой МФ может быть выполнен, как последовательно включенные формирователь фиксированной цифровой последовательности и цифроаналоговый преобразователь.

Таким образом, за счет единственности тракта аналоговой обработки (от антенны приемника до выхода аналого-цифрового преобразователя) и применения цифровой обработки средством распознавания ПВ, исключается ошибка, порождаемая в прототипе за счет неодинаковости задержек различных аналоговых трактов обработки сигналов. В

результате точность измерений возрастает.

### 3. ПРАКТИЧЕСКАЯ РЕАЛИЗАЦИЯ ПРЕДЛОЖЕННОГО УСТРОЙСТВА

На рис. 7 показана функциональная схема каждого из приемников с последующим трактом обработки смеси радиосигналов. Для реализации этого устройства может быть использован серийный приемник, содержащий антенну с усилителем, квадратурно-балансный демодулятор, гетеродин, формирующий когерентную и квадратурную компоненты, АЦП и контур управления усилением и частотой гетеродина. Оригинальной программно-аппаратной частью является только средство обработки сигналов, содержащее устройство для распознавания двух признаков времени и цифровой фазометр.

**Метод качественно разных огибающих.** На рис. 8 показан пример двух видов амплитудной модуляции сигналов, каждый из которых снабжает модулируемый сигнал признаками времени. Верхний сигнал имеет модуляцию треугольными импульсами, нижний – прямоугольными импульсами, длительности периодов модуляции соотносятся как взаимно простые числа, несущая частота одна и та же.

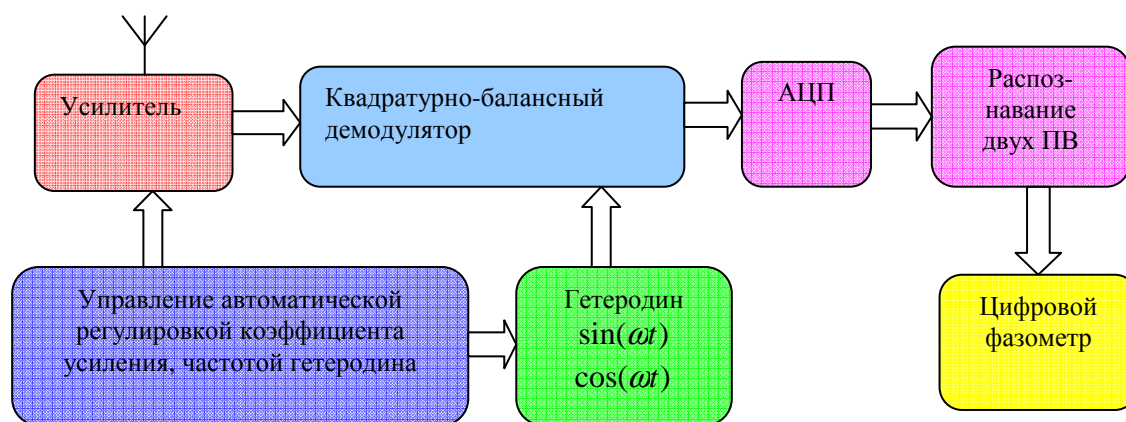


Рис 7. Функциональная схема приемника смеси радиосигналов

**Метод поочередного молчания.** На рис. 11 показана пара сигналов, формируемых по методу, который можно назвать методом поочередного молчания. В этом случае периоды передачи чередуются с периодами пауз. При этом длительности передачи и длительности пауз от различных передатчиков таковы, что достоверно при любых постоянных сдвигах времени между этим сигналами сохраняются интервалы времени, на которых присутствует только сигнал от первого передатчика, и участки, на которых присутствует только сигнал

Наряду с использованием коррелятора или вместо него могут быть использованы иные методы обработки сигналов. Например, момент времени достижения минимума и максимума сигналом с треугольной модуляцией может быть определен, например, методом максимального правдоподобия. Это же относится к определению момента переднего и заднего фронтов сигнала с прямоугольной модуляцией.

На рис. 9 показан результат суммирования сигналов, показанных на рис. 8, а также результаты детектирования этого сигнала по отдельности с когерентным и с квадратурным гетеродином, которые показаны разным цветом. Результат суммирования двух нижних сигналов показан на рис. 10. В этом сигнале можно усмотреть признаки каждого из исходных сигналов в раздельности. Момент резкого изменения амплитуды соответствует фронту или срезу модулирующего сигнала прямоугольных импульсов (т.е. несет временные признаки сигнала второго передатчика), момент излома огибающей несет признаки времени сигнала первого передатчика. Этот метод можно назвать методом качественно разных огибающих.

от второго передатчика. Это обеспечивается выбором периодов повторения, которые относятся в некотором некрatном соотношении, а также выбором длительностей сигналов. Некратное соотношение, например, число « $\pi$ » или « $e$ », позволяет обеспечивать повышение точности при увеличении времени усреднения и накопления статистического сигнала детектирования.

На рис. 12 показаны сигналы, получаемые в результате квадратурной демодуляции суммы сигналов, показанных на рис. 11. Видно, что в



полученных сигналах можно указать интервалы, когда отсутствуют оба сигнала, а также интервалы, на которых присутствует, как минимум, один из сигналов. Очевидно, что, как правило, вначале появляется лишь один из сигналов, далее следует интервал, на котором присутствуют оба сигнала, на следующем

интервале присутствует лишь второй сигнал. Даже если такая последовательность время от времени нарушается, ситуация может легко распознаваться вследствие характерных, заранее известных и некротных длительностей первого и второго сигнала.

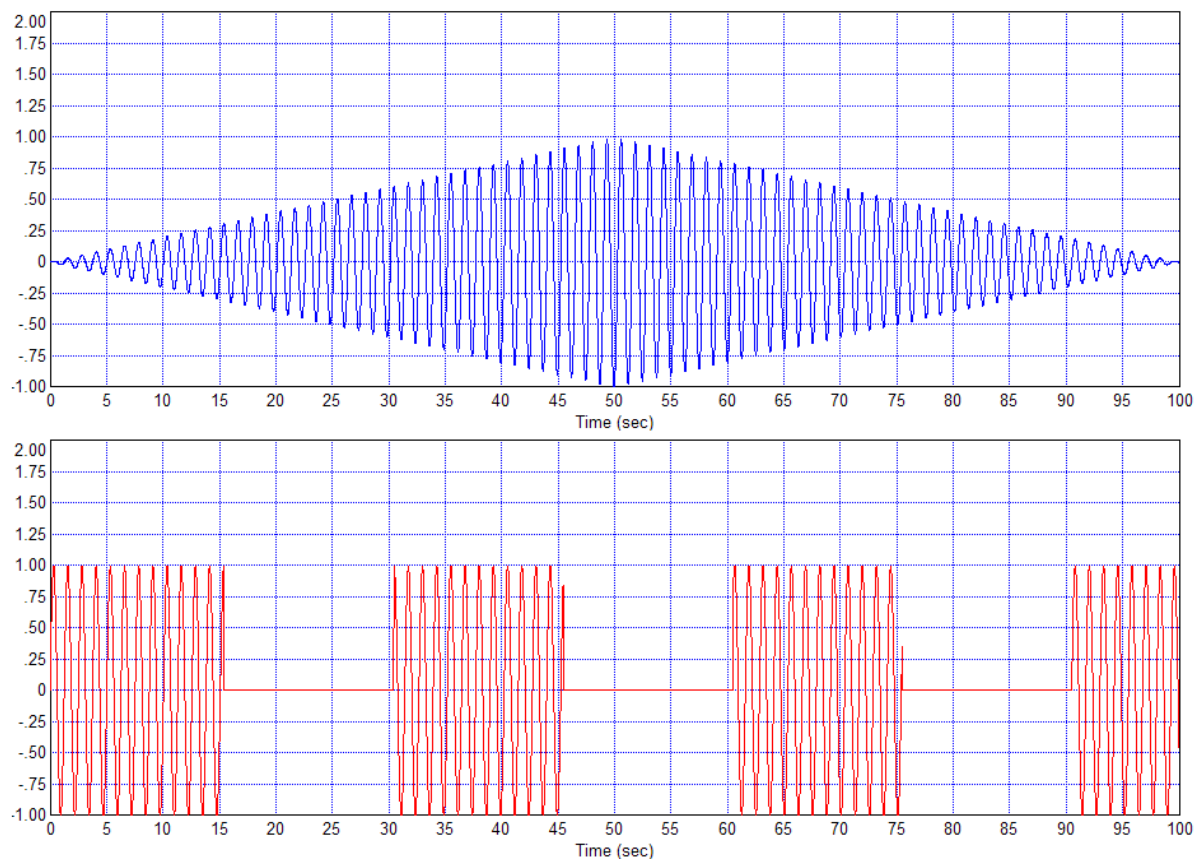


Рис. 8. Пример двух видов модуляции, снабжающих признаком времени

**Метод периодического инвертирования фазы.** Вместо периодического отключения сигнала может быть использовано периодическое инвертирование его фазы. На рис. 13 показаны соответствующие сигналы передатчика. Периодичность инверсии является характерным признаком данного передатчика, периоды инвертирования соотносятся как иррациональные числа. Фазометр определяет фазу получаемого сигнала, при этом измеренная

разность фаз изменяется скачками в различные моменты времени, соответствующие моментам изменения фаз на первом и на втором передатчиках. По характерному периоду каждого из скачков при вторичной обработке можно выявить признаки первого или второго передатчика, что может служить признаком времени (признаком запаздывания) от каждого передатчика в отдельности.

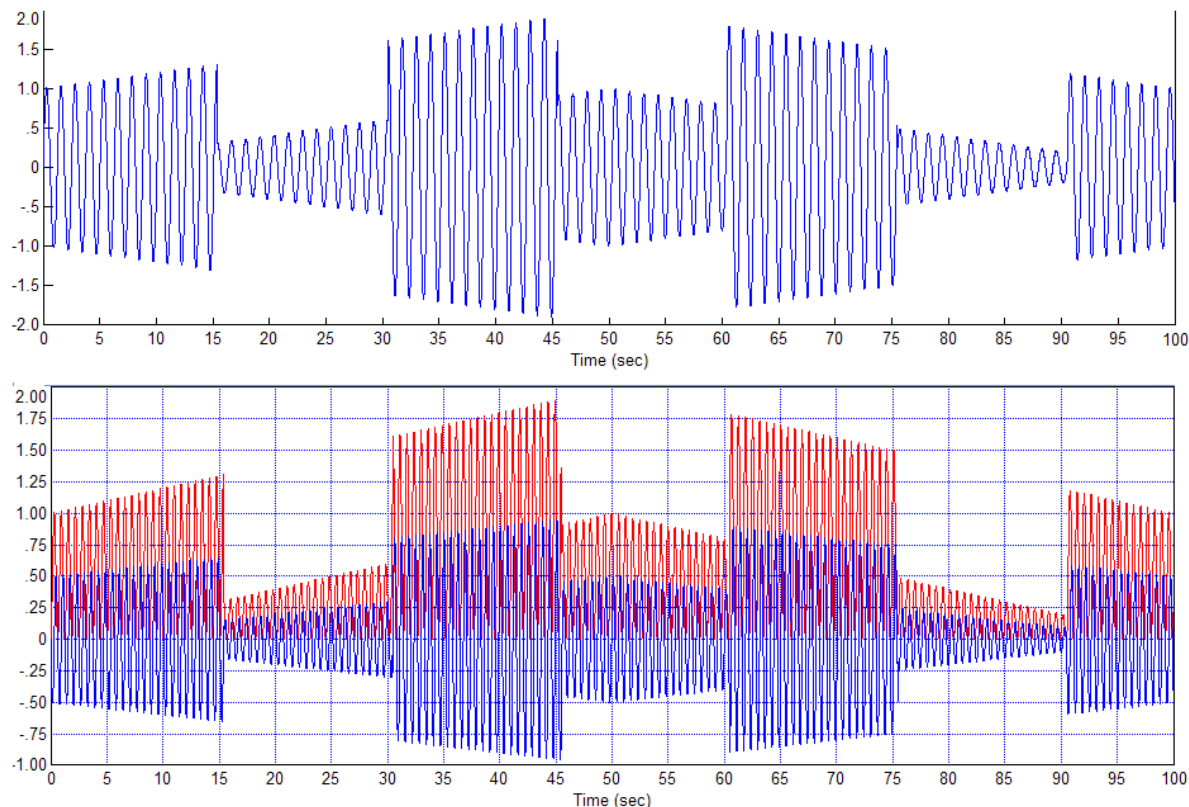


Рис. 9. Сигнал, являющийся суммой сигналов рис. 8, а также результаты детектирования этого сигнала

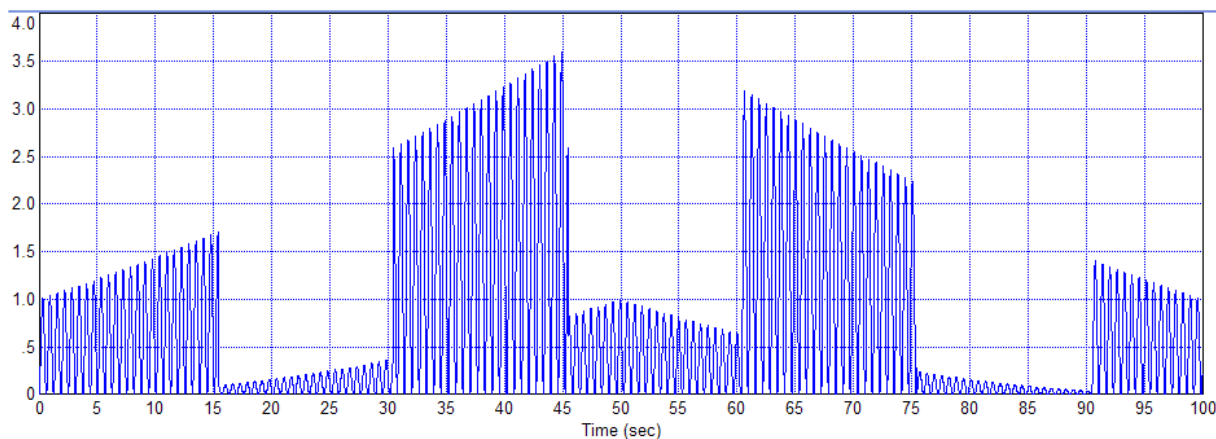


Рис. 10. Сумма двух нижних сигналов, показанных на рис. 9

**Метод объединения видов модуляции.**  
Суть этого метода состоит в комплексном использовании одновременно нескольких видов модуляции для более уверенного выделения ее признаков. Например, можно предложить для сигналов по методу качественно разных огибающих ввести дополнительно скачкообразную амплитудную и (или) фазовую

модуляцию для первого и (или) для второго сигналов. К примеру, сигнал с треугольной огибающей может в момент достижения нуля изменять фазу на 180 градусов, и при этом сигнал с прямоугольной огибающей после очередной паузы изменять фазу на 90 градусов. Также может быть использована модуляция фазы только для одного из указанных сигналов.

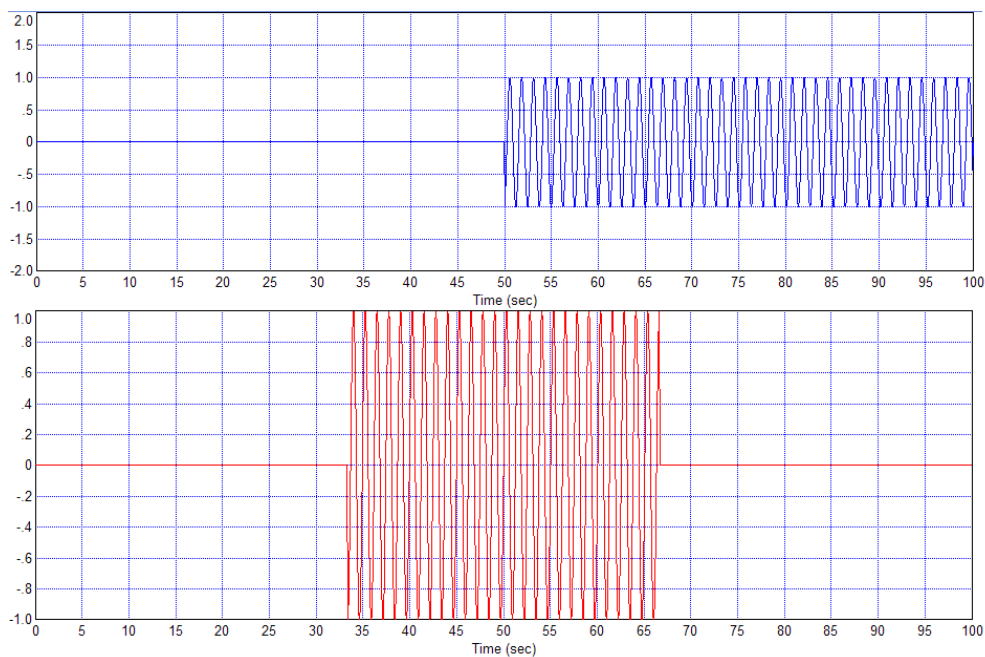


Рис. 11. Сигналы по методу поочередного молчания

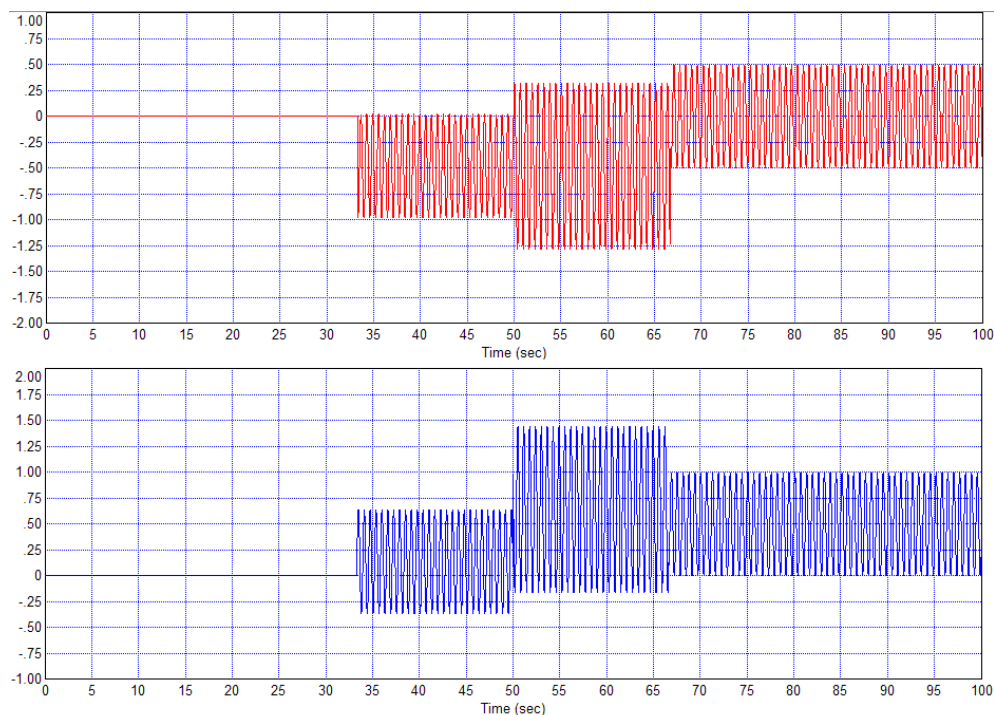


Рис. 12. Компоненты сигналов при детектировании (когерентная и квадратурная)

Итогом всех обсуждаемых технических решений является тот факт, что смесь сигналов от двух различных передатчиков, осуществляющих передачу сигналов на одной и той же несущей частоте, можно осуществлять одной антенной, одним трактом первичной обработки сигналов. В этом случае все виды задержек от антенны и от тракта обработки, включая случайные девиации свойств фильтров

и девиации фазы гетеродинного генератора. Получаемые разности фаз или разности временных признаков в групповом сигнале могут быть идентифицированы и отнесены к конкретным передатчикам. При этом разность времен поступления этих временных признаков несет информацию о различной дальности этих передатчиков по отношению к приемной антенне.

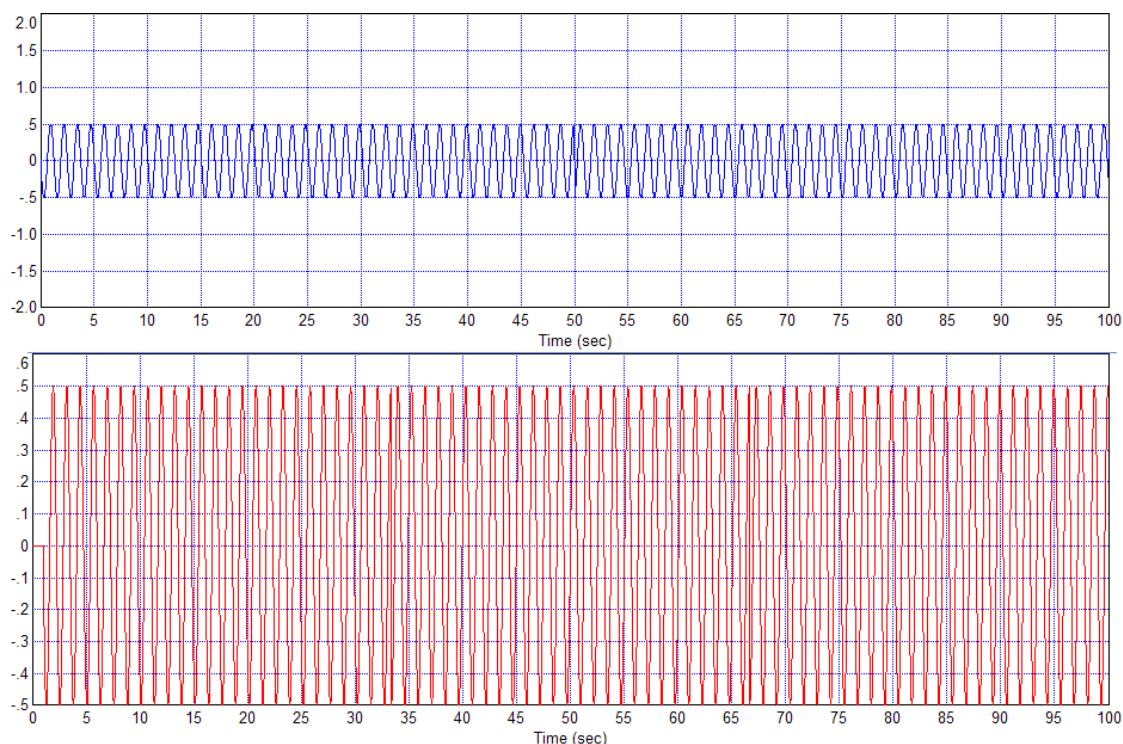


Рис. 13. Сигналы передатчиков по методу периодического инвертирования фазы

#### 4. ОБЕСПЕЧЕНИЕ ЭФФЕКТИВНОСТИ РАБОТЫ СРЕДСТВА ЭКСТРЕМАЛЬНОЙ НАСТРОЙКИ

Выбор метода модуляции способен повлиять на эффективность действия средства экстремальной настройки. Одна из причин недостаточной точности может состоять в высокой погрешности определения точных моментов поступления признаков времени.

Признак времени может содержаться в явном или неявном виде в форме огибающей функции сигнала, передаваемого каждым из передатчиков. Явным признаком времени может служить, например, фронт импульса или момент инвертирования фазы передаваемого сигнала. Неявным признаком может служить, например, точное значение относительно медленно изменяющейся гармонической огибающей функции. Трудность точного определения явного признака времени состоит в следующем. Во-первых, он является одним из видов быстрого изменения сигнала, и, следовательно, спектр сигнала, содержащего этот признак, очень широкий. При передаче по радиочастотному каналу, имеющему, как правило, не слишком широкий спектр, часть исходного спектра не проходит через приемопередающий спектр. Оставшийся сигнал приобретает искажения, вызывающие изменение точного момента определения

характерного признака времени, например, фронт импульса растягивается, а момент переключения фазы сглаживается соответствующим переходным процессом. Во-вторых, если применять, например, корреляционный метод отыскания признака времени, то алгоритм реализации этого метода чрезвычайно усложняется, поскольку зависимость корреляционной функции от ошибки настройки коррелятора существенно не плавная, поэтому производная этой функции по ошибке многократно изменяет знак. Это приводит к тому, что возможны ошибочные настройки на локальные экстремумы корреляционной функции. Трудность точного определения неявного признака связана с несущественным изменением сигнала ошибки настройки при небольшой ошибке, то есть с низкой чувствительностью метода.

Предлагаемое техническое решение направлено на повышение точности измерителя.

Поставленная задача решается тем, что предлагается использовать одновременно два признака времени, один из которых – неявный и служит для робастного (грубой, приближенного) отыскания признака времени и настройки на него, а второй – явный и служит для точного определения признака времени, например, путем настройки фазы собственного (гетеродинного) генератора на этот признак времени. С этой целью в устройство введены

два генератора гладкой модуляции и генератор признаков времени на входах каждого передатчика, а также между выходом каждого приемника и входом каждого средства

измерения временных интервалов введены параллельно включенные два канала обработки сигналов.

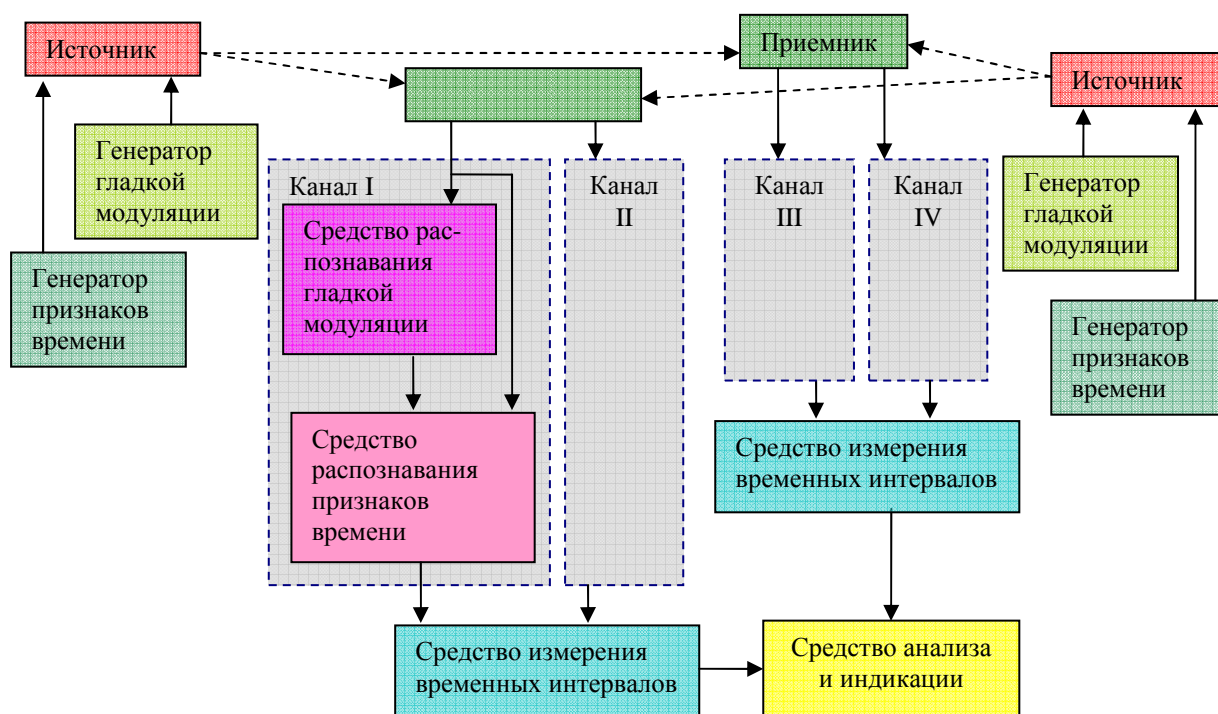


Рис. 14. Схема устройства с использованием двух видов модуляции

– Схема этого измерителя приведена на рис.14.

Источники радиосигналов снабжены двумя генераторами для модуляции передаваемых сигналов: генератором гладкой модулирующей функции и генератором признаков времени. Для возможности различения сигналов приемниками все указанные генераторы формируют сигналы, отличающиеся многими признаками, что должно обеспечить малую их взаимную корреляцию. Каждый из источников, таким образом, излучает сигналы с индивидуальными признаками. Каждый из двух передаваемых сигналов имеет два вида модуляции. Каждая модуляция осуществляется сигналами с выходов соответствующих генераторов. Генератор гладкой модуляции формирует сигнал, осуществляющий гладкую модуляцию. Эта модуляция содержит неявные признаки времени и служит для грубой (предварительной) настройки каналов обработки сигналов. Генератор признаков времени формирует сигналы, содержащие явные признаки времени. Эти сигналы служат для модуляции, которая используется для точного определения моментом поступления признаков времени. Разные виды модуляции могут вводиться путем

их суммирования или любым иным путем, например, одна модуляция может быть амплитудной, а другая – частотной. Каждый из приемников принимает сигнал каждого из источников в форме их смеси (суммы с коэффициентами). Принятые сигналы от каждого источника подаются на два индивидуальных канала обработки. Каждый из каналов обработки сначала путем распознавания гладкой модуляции для грубой, предварительной настройки определяет моменты поступления неявных признаков времени, содержащихся в гладкой модуляции. После этого путем распознавания признаков времени более точно определяет моменты поступления явных признаков времени, вводимых в передаваемые сигналы генераторами признаков времени. Результатом действия каналов обработки сигналов является точное определение моментов поступления явных признаков времени. Сигналы, отмечающие эти моменты, поступают на входы средств измерения временных интервалов, которые определяют разность времени поступления явных признаков времени от различных приемников. Результаты измерений поступают на средство анализа и индикации.

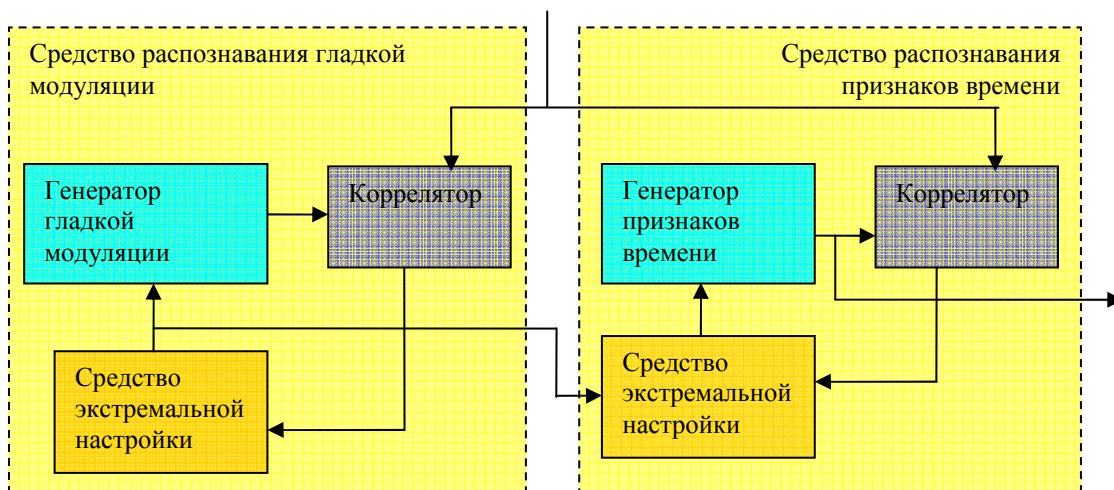


Это средство вычисляет разность этих результатов, которая пропорциональна расстоянию между приемниками, как описано выше.

Наличие в принятых сигналах гладкой модуляции позволяет легко осуществить грубую предварительную настройку, что исключает настройку на локальные экстремумы

и повышает надежность и быстродействие системы. Наличие в принятых сигналах явных признаков времени позволяет осуществлять более точную окончательную настройку, что повышает точность измерения.

Каждый из каналов обработки сигналов может быть, например, выполнен по схеме, показанной на *рис. 15*.



*Рис. 15.* Структура канала обработки сигналов

Каждый канал содержит средство распознавания гладкой модуляции и средство распознавания признаков времени, причем, входной сигнал каждого из этих каналов поступает на каждое из этих средств, и, кроме того, дополнительный сигнал грубой настройки поступает из средства распознавания гладкой модуляции на вход средства распознавания признаков времени. Этот сигнал позволяет более успешно (быстро и надежно) определять приближенное значение моментов поступления признаков времени.

Например, средство распознавания гладкой модуляции может содержать включенные петли генератор гладкой модуляции, коррелятор и средство экстремальной настройки. Генератор гладкой модуляции полностью идентичен соответствующему генератору, подключенному только к одному из передатчиков. Коррелятор сравнивает сигнал с выхода генератора гладкой модуляции с сигналом, поступающим на другой его вход с выхода приемника. Выходной сигнал корреляции характеризуется гладкой зависимостью от ошибки настройки, производная этого сигнала по ошибке плавно изменяет свою величину и редко изменяет свой знак. Это позволяет средству экстремальной настройки надежно и безошибочно настроить

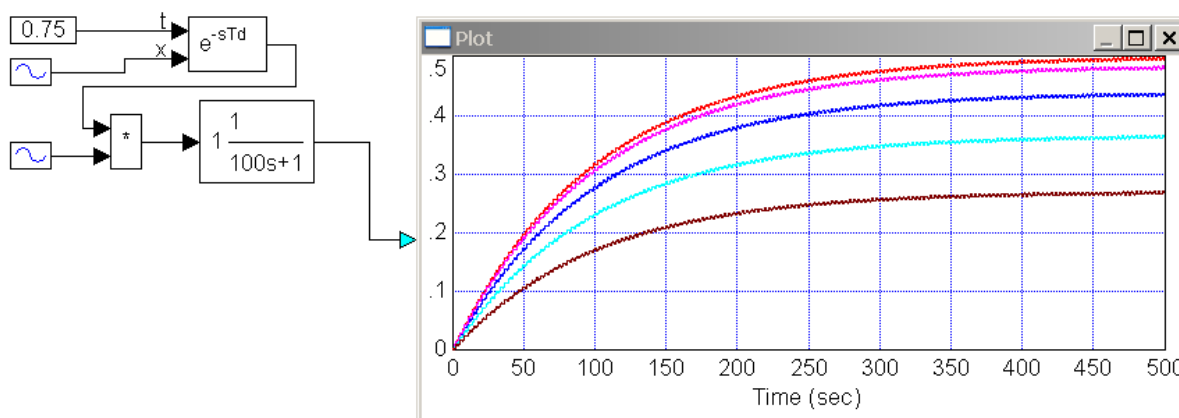
фазу генератора гладкой модуляции, обеспечив близкое соответствие этой фазы фазе принятого сигнала. Сигнал автоподстройки с выхода средства экстремальной настройки используется средством распознавания признаков времени для предварительной настройки. Средство распознавания признаков времени может работать по такому же принципу и иметь аналогичную структуру, то есть содержать генератор признаков времени, идентичный генератору признаков времени, коррелятор и средство экстремальной настройки. Это средство экстремальной настройки использует информацию, поступающую с выхода средства распознавания гладкой модуляции для предварительной настройки генератора признаков времени. Это ускоряет процесс настройки и исключает настройку генератора признаков времени на локальный экстремум, следовательно, повышает точность определения моментов поступления признаков времени.

## **5. МОДЕЛЬНОЕ ТЕСТИРОВАНИЕ ПРЕДЛОЖЕННОГО МЕТОДА**

Эффективность предложенного метода исследована путем моделирования в программе *VisSim*. Результаты моделирования показаны на *рис. 16–20*. Выходной сигнал коррелятора

имитируется конструкцией, содержащей перемножитель сигналов и низкочастотный фильтр на его выходе. Устройство запаздывания позволяет получать произвольный сдвиг фаз перемножаемых сигналов для исследования влияния этого сдвига на выходной сигнал коррелятора. При использовании только гармонического сигнала, как показано на *рис. 16*, зависимость установившегося уровня выходного сигнала плавно зависит от относительного сдвига фаз перемножаемых сигналов. Это позволяет относительно просто

организовать поисковую процедуру для отыскания оптимальной настройки численными методами, которая соответствует максимальному значению выходного сигнала коррелятора. Наряду с этим достоинством, такой сигнал обладает существенным недостатком, состоящем в том, что при небольших изменениях величины запаздывания (при небольших ошибках настройки фаз) выходной сигнал коррелятора изменяется незначительно.



*Рис. 16.* Монотонная зависимость выходного сигнала коррелятора от запаздывания



*Рис. 17.* Немонотонная зависимость выходного сигнала коррелятора от запаздывания

На *рис. 17* приведен пример использования псевдослучайного сигнала с той же целью, с которой ранее применялся генератор гармонического сигнала. Как видно из *рис. 17*, полное совпадение фаз перемножаемых сигналов вызывает резкое возрастание выходного сигнала коррелятора. На *рис. 18* показаны в укрупненном масштабе сигналы при различных значениях запаздывания, то есть при различной величине ошибки настройки фаз. По этим сигналам затруднительно сделать верное заключение о величине и/или знаке фазовой ошибки, поскольку отсутствует явная закономерность между знаком ошибки настройки фазы и знаком выходного сигнала

коррелятора. На *рис. 19* показан пример результата использования суммы двух сигналов, один из которых является псевдослучайным, а другой – гармонический. Результат сочетает оба достоинства и свободен от обоих недостатков, а именно: зависимость выходного сигнала коррелятора от ошибки относительно плавная с экстремумом в точке наиболее точной настройки, причем экстремум выражен весьма существенно, выходной сигнал возрастает в три раза в сравнении с сигналом при неточной настройке (на значение длительности шага интегрирования, установленного в программе моделирования).

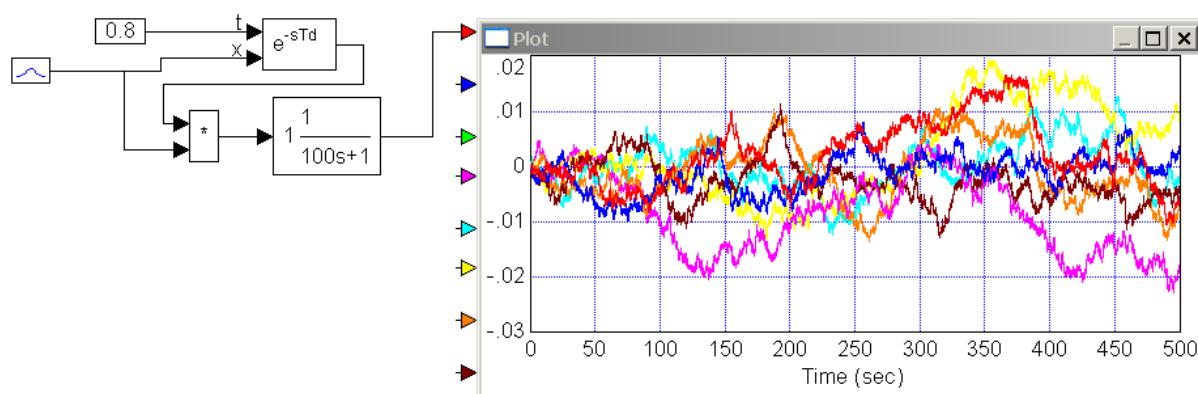


Рис. 18. То же, что на рис. 17, в крупном масштабе

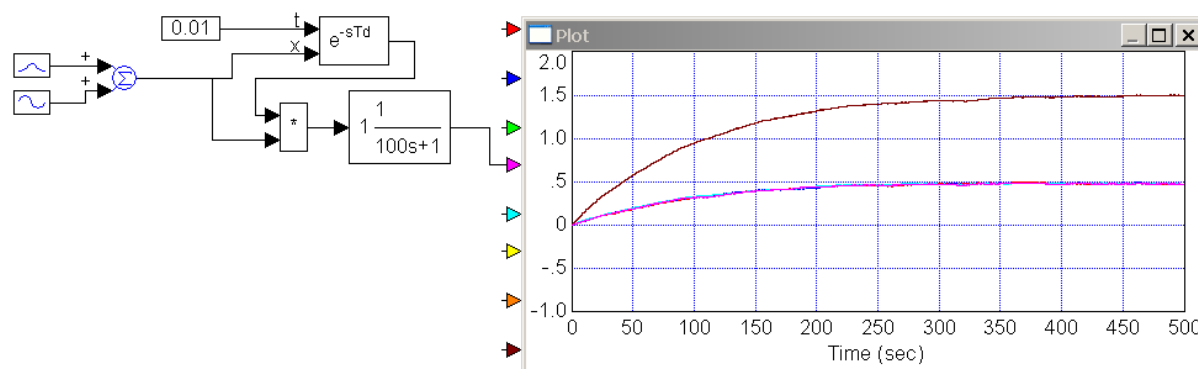


Рис. 19. Получение монотонной зависимости от запаздывания с резким (немонотонным) пиком при полном совпадении фаз путем суммирования гармонического и псевдослучайного процессов (модель выходного сигнала коррелятора)

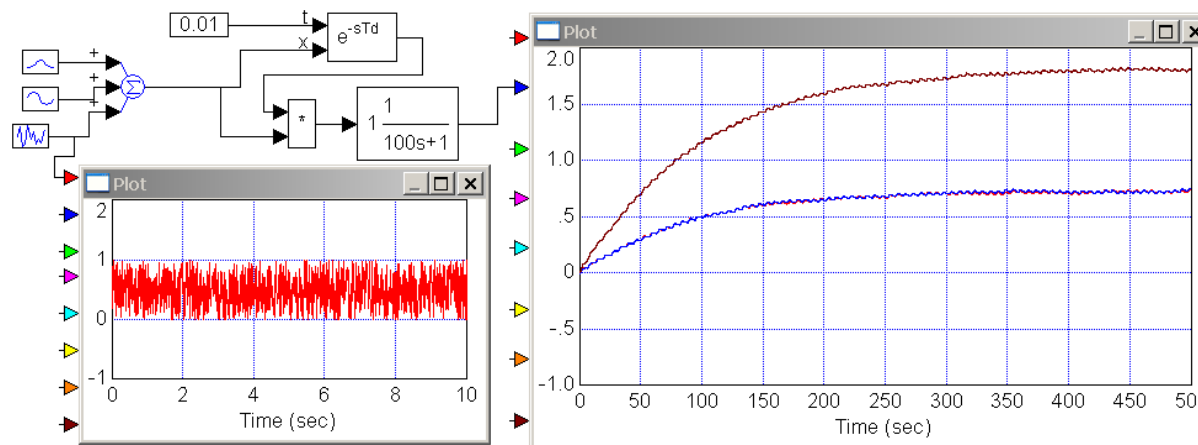


Рис. 20. Использование двух псевдослучайных сигналов и одного гармонического

На рис. 20 показан результат использования суммы двух псевдослучайных сигналов и одного гармонического сигнала. Результат получается аналогичным. На рис. 21 показан характерный вид такой зависимости в осях: ошибка по фазе (ось X, нулевая ошибка соответствует значению 8) – выходной сигнал коррелятора (ось Y).

Отметим отличительные особенности разработки. Все известные радиочастотные

измерители действуют на принципе отражения или рассеяния радиочастотного сигнала от объекта. Недостатком таких измерителей является сложность создания источника узконаправленного излучения и отражателя радиоволн и сложность настройки таких устройств, а также возникновение помех от сторонних объектов, которые невозможно устранить.



Рис. 21. Зависимость выходного сигнала коррелятора от настройки (условные единицы) фазы

В предложенной системе достигаются следующие преимущества:

1. Не требуется прямая видимость источников и приемников.

2. Не используется отражение или рассеяние сигналов.

3. Понижена зависимость точности от погодных изменений атмосферных характеристик по сравнению с оптическими устройствами.

Сделаем оценку потенциальных отличий по точности указанного метода и альтернативных радиотехнических схем. К ним отнесем устройства с отражением или рассеянием сигналов и устройства на основе космических систем (ГЛОНАСС, GPS). Схема расположения элементов измерительной системы при использовании отражения или рассеяния показана на рис. 22. Основная проблема такой схемы – практическая сложность создания плоского фронта волны методом отражения или рассеяния. Схема применения предложенного измерителя показана на рис. 23, а схема с использованием космических систем показана на рис. 24.

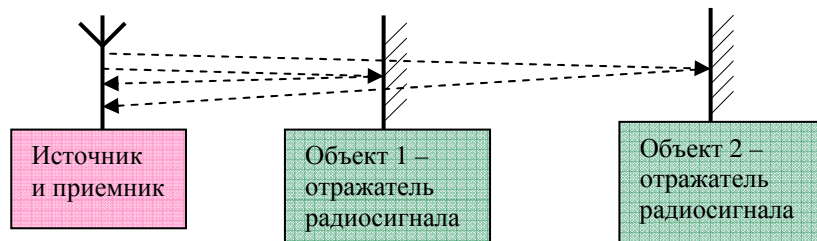


Рис. 22. Схема расположения элементов измерителя при использовании отражения или рассеяния

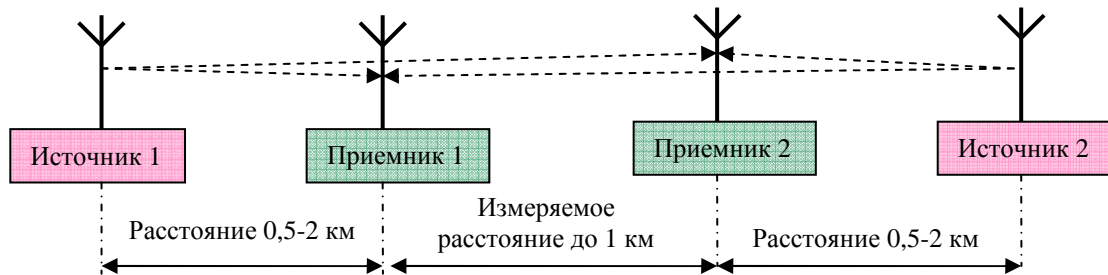


Рис. 23. Схема расположения элементов измерителя

Радиотехнический метод на основе космических систем принципиально характеризуется погрешностью в 30 раз большей. Однако наземные эксперименты с предложенными системами могут служить основой для развития подобной методики измерений с помощью космических систем.

Действительно, если наземными системами измеряются приращения на базе 1 км, то абсолютная погрешность в 1 мм соответствует относительной погрешности  $10^{-6}$ . Если же измерения осуществляются космическими

системами, например, при высоте 20 км, дальность до спутника под углом  $45^\circ$  составляет примерно 30 км, поэтому при той же самой абсолютной погрешности требуется обеспечить относительную погрешность в 30 раз меньше, то есть  $0,33 \cdot 10^{-7}$ .

Таким образом, обоснование возможности использования этого метода в космических системах состоит в следующем: собственно конкретный измеритель – это система из двух приемников и средства обработки сигналов. Одни и те же сигналы от передатчиков могут

быть использованы для неограниченного количества измерителей. Система не требует «общения», обмена сигналами с передатчиками. Спутник не должен откликаться на запрос

каждой такой измерительной системы, достаточно лишь отправки стандартизованных уникальных сигналов. Такая система теоретически может быть внедрена в космосе.

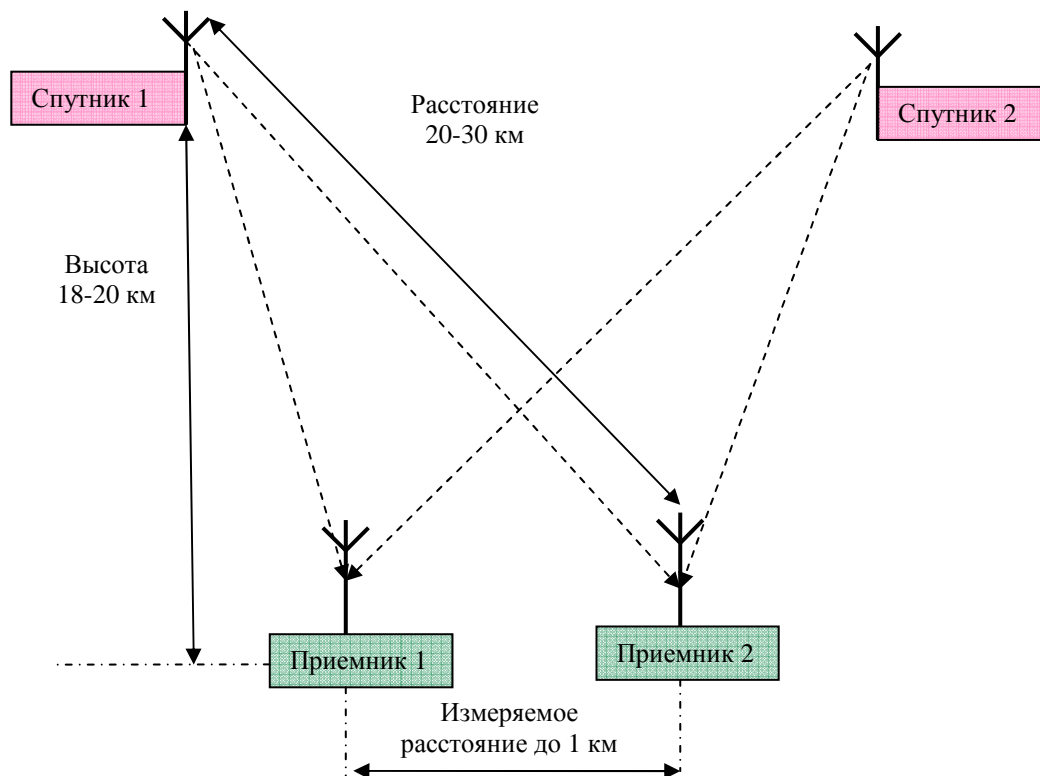


Рис. 24. Схема расположения элементов измерителя в случае использования космических систем

Поэтому разработанную методику измерений можно при необходимости привести в систему ГЛОНАСС, если она докажет высшую точность. При сохранении относительной погрешности такой перенос увеличит абсолютную погрешность в 20-30 раз. Аналогично, если методики, имеющиеся в системе ГЛОНАСС перенести на поверхность, то при сохранении относительной погрешности, абсолютная погрешность может быть снижена в 20-30 раз. Если учесть, что в космической системе имеется существенная неоднородность (плотности и температуры) атмосферы на протяжении трассы измерительных излучений, то отличие может превысить указанные расчетные отличия дополнительно еще в несколько раз.

Основная задача последующих исследований состоит в дальнейшем повышении точности измерений радиотехническим методом.

Таким образом, предложенный метод эффективно решает задачу обеспечения надежной работы средства экстремальной

настройки, поскольку обеспечивает одновременно и плавную зависимость выходного сигнала коррелятора от ошибки настройки и резкое увеличение этого сигнала при настройке с высокой точностью.

## ЗАКЛЮЧЕНИЕ

Работа поддержана грантом по проекту «Исследование предельных точностей оптических методов измерения параметров движения и мехатронных методов управления движением и разработка новых робототехнических и электромеханических систем», Темплан, проект № 7.559.2011 и при поддержке Министерства образования и науки в рамках ФЦП «Научные и научно-педагогические кадры инновационной России» на 2009-2013 гг., мероприятие 1.2.2, конкурс 630П, ГК № П 761 от 20.05.2010.

## ЛИТЕРАТУРА

[1] М.И. Мусяков, И.Д. Миценко. Оптико-электронные



системы ближней дальнометрии. Москва: Радио и связь. 1991. с.4.

[2] А.П. Сиверс, Н.А. Сулов, В.И. Метельский. Основы радиолокации. Ленинград: Судпромгиз. 1959 г. с.9.

[3] Измеритель линейных перемещений, патент на полезную модель № 87252, опубликовано 27.09.2009, Бюл. № 27.



Вадим Жмудь – заведующий кафедрой Автоматики в НГТУ, профессор, доктор технических наук, автор 200 научных статей. Область научных интересов и компетенций – теория автоматического управления, электроника, лазерные системы, оптимизация, измерительная техника.

E-mail: [oao\\_nips@bk.ru](mailto:oao_nips@bk.ru)



Денис Терешкин, аспирант кафедры Автоматики НГТУ, автор 15 научных статей. Область научных интересов и компетенций – теория автоматического управления, лазерная физика, программные системы.

E-mail: [toadin.dirihle@gmail.com](mailto:toadin.dirihle@gmail.com)



Александр Ляпидевский, директор ОАО «НИПС», автор более 50 научных статей. Область научных интересов и компетенций – программные системы и средства, инновационные технологии.

E-mail: [nips@nips.ru](mailto:nips@nips.ru)



Антон Захаров, аспирант кафедры Автоматики НГТУ, автор 10 научных статей. Область научных интересов и компетенций – программные системы и средства.

E-mail: [zaharov\\_av@mail.ru](mailto:zaharov_av@mail.ru)

# Исследование оптоэлектронного метода и разработка устройств для мониторинга жидких сред

Е.Ю. Кутенкова<sup>1</sup>, Б.Н. Рахимов<sup>2</sup>, Т.В. Ларина<sup>1</sup>, Ш.И. Мадумаров<sup>3</sup>

<sup>1</sup> ФГБОУ ВПО СГГА, <sup>2</sup> ТУИТ, Ташкент, <sup>3</sup> ФГБОУ ВПО НГТУ

**Аннотация.** Приведен анализ существующих средств мониторинга жидких сред. Предлагается новая конструкция многофункционального оптоэлектронного фотоколориметра для автоматического контроля. Отличительными особенностями которого является то, что стационарно расположенная кювета представляет собой склеенный из двух половин цилиндр с центральным отверстием, куда заливается исследуемая жидкость. Наличие  $n+1$  оптопар позволяет контролировать наличие и содержание  $n$  компонентов в составе жидкостей.

**Ключевые слова:** Светоизлучающий диод, фотоприемник, оптопара, кварцевая кювета, полупрозрачные жидкости, сумматор, блок обработки фотоэлектрического сигнала, измерительный прибор, оптоэлектронная система.

В настоящее время оптоэлектронные методы контроля [1 - 4] с успехом применяются для количественного и качественного анализа различных веществ, например, для определения параметров полупрозрачных жидких сред (нефтепродуктов, растительного масла, глицерина, соков, напитков, мочи, крови и т.п.). Они обладают такими существенными преимуществами по сравнению с другими физико-химическими методами анализа, такими как высокая точность и чувствительность, экономичность являются их преимуществами.

Суть оптоэлектронного мониторинга состоит в том, что любое вещество отражает, поглощает, или излучает свет. При этом в зависимости от химического состава вещества и количественного соотношения составляющих его элементов изменяется интенсивность светопоглощения, угол отражения и другие характеристики взаимодействия светового излучения и вещества.

Оптические устройства для контроля физико-химических параметров жидкостей в зависимости от физических явлений, изменяющих параметры светового потока, прошедшего через исследуемую среду, можно подразделить на три группы:

поляриметрические, рефрактометрические, фотометрические.

Фотометрический метод в основном применяется для анализа жидких веществ и растворов. В этом методе либо сравниваются света анализируемой жидкости с эталоном, либо оценивается степень поглощения светового излучения.

Определяя цвета жидких сред можно контролировать концентрацию того или иного компонента. Это производится либо визуально, либо при помощи оптоэлектронных устройств, превращающих, с начала, электрическую энергию в световую, а после взаимодействия света с контролируемой средой – обратно в электрическую. В соответствии с этим различают фотометрический визуальный метод анализа, называемый часто колориметрическим, и метод анализа с применением оптронов – фотоколориметрический.

Одним из наиболее развивающихся направлений в этой области является создание оптоэлектронных анализаторов – фотоколориметров. Бесспорно, фотоколориметры применяются в различных областях народного хозяйства, например, на предприятиях водоснабжения, в металлургической, химической, нефтеперерабатывающей, пищевой промышленности, в сельском хозяйстве и других областях народного хозяйства. для непрерывного или дискретного контроля физико-химических параметров веществ и материалов. В частности, фотоколориметры предназначены для измерения коэффициентов пропускания и оптической плотности водных растворов, а также для измерения скорости изменения оптической плотности вещества и определение концентрации определенного вещества в растворе.

Широкое использование фотоколориметры получили в медицине для определения концентраций биопрепаратов в биологических жидкостях и контроля за состоянием человека.

В последние годы значительно возросли требования, предъявляемые к оперативности и качеству медицинских услуг. Однако нехватка финансирования в медицине не позволяет

вывести эту важнейшую отрасль на новый качественный уровень. В частности, наличие на рынке медицинской техники широкого выбора оборудования для лабораторных исследований зарубежного производства не позволяет решить проблему снабжения отечественных биохимических и клинико-диагностических лабораторий современной техникой из-за ее большой стоимости. При этом отсутствуют современные отечественные разработки в данной области. Кроме того, зарубежное лабораторное оборудование ориентировано на использование дорогостоящих реактивов производства западноевропейских фирм, что при массовых исследованиях делает практически невозможным эксплуатацию таких приборов в рядовых отечественных медицинских учреждениях. Отсутствие квалифицированного технического персонала по обслуживанию зарубежного лабораторного оборудования еще более усугубляет эту ситуацию.

Эксплуатация таких приборов крайне затруднена, например, в условиях полевой лаборатории, в машине скорой помощи, при профосмотрах и выездах, что объясняется их громоздкостью и необходимостью работы от электросети. Отсутствие на рынке специализированных малогабаритных портативных приборов с возможностью работы не только от электросети, но и от бортовой сети автомобиля или аккумуляторов затрудняет работу медперсоналу.

Главной частью традиционного фотоколориметра, определяющей энергопотребление, надежность и массогабаритные показатели является оптический блок на основе лампы накаливания, который практически во всех фотометрических приборах является закрытым и имеет конструктивную защиту от влияния условий внешней освещенности. Эксплуатация таких приборов сопряжена с дополнительными манипуляциями при каждом измерении и, соответственно, с существенными затратами времени при проведении массовых измерений.

Разработка малогабаритного оптического блока открытого типа на основе полупроводниковой оптоэлектронной пары и переход от конструктивных средств защиты оптического канала к схемотехническим позволит кардинально уменьшить массу и габариты прибора повысить его надежность и увеличить срок службы. В связи с этим особую значимость приобретает разработка электронной измерительной части прибора, позволяющей эффективно работать в условиях меняющейся внешней освещенности.

В настоящее время наибольшее применение нашли отечественные фотоколориметры серийного выпуска такие, как колориметры фотоэлектрические концентрационного типа КФК-2МП [5], КФК-2 (рис. 1), содержащие излучатель, светофильтр, кювету (кварцевую или стеклянную) с исследуемым раствором, приемник оптического излучения (ПОИ), электронный блок обработки сигналов и измерительный прибор.

Эти колориметры предназначены для измерения в отдельных участках диапазона длин волн 315 - 80 нм, выделяемых светофильтрами, коэффициентов пропускания и оптической плотности жидкостных растворов и твердых тел, а также определения концентрации веществ в растворах методом построения градуировочных графиков. Колориметр позволяет производить измерения коэффициентов пропускания рассеивающих взвесей, эмульсий и коллоидных растворов в проходящем свете.



Рис. 1. Фотоколориметр КФК-2

Кроме того в мировой практике для этих целей получил распространение фотометр ЮНИКО 1201 (рис. 2), который предназначен для измерения коэффициентов пропускания, оптической плотности и концентрации растворов. Однолучевой спектрофотометр специально разработан для широкого использования в лабораториях всех отраслей промышленности, решения аналитических задач в таких отраслях, как клиническая химия, биохимия, защита окружающей среды, энергетика, металлургические, химические, пищевые лаборатории, лаборатории воды и сточной воды, нефтехимии и в других сферах контроля качества.

По техническим характеристикам и возможностям ЮНИКО 1200 (1201) полностью

заменяет широко применяемые в лабораторной практике фотоколориметры и фотометры фотоэлектрические типа ФЭК и КФК-2,-3,-5 и др. Например, спектрофотометр обеспечивает определение содержания веществ в различных растворах – меди, железа, хлора, серебра; определение содержания сахара, билирубина, глюкозы, холестерина, креатина в крови и моче; мочевины, общего белка, щелочей, фосфатов в химических растворах.

Фотометр ЮНИКО 1200 (1201) работает в спектральном интервале 325 – 1000 нм.

Прибор специально адаптирован для отечественных условий эксплуатации и выпускается с учетом российских лабораторных требований.



Рис. 2. Фотометр ЮНИКО 1201

Для данных устройств характерны следующие недостатки: невысокая чувствительность и неточность измерений, за счет несовершенства конструкции кюветы и кюветодержателя, которая предполагает необходимость извлечения кюветы из прибора при проведении каждого последующего анализа для того, чтобы заполнить очередной пробой, кроме того поверхности кювет приходится мыть и протирать от потеков исследуемой жидкости.

Промышленностью выпускаются автоматические фотоэлектрические колориметры для непрерывного контроля за цветом нефтепродуктов на технологических потоках, в которых используют дифференциальный метод измерения светопоглощения. В качестве сравнительного образцового средства чаще всего используется светофильтр, изготовленный из цветных стекол или специально подобранная жидкость с равноценной спектральной характеристикой. Выходное напряжение, подаваемое на регистрирующий прибор, изменяется пропорционально изменению отношения оптических плотностей контролируемого

продукта и образцового средства и измеряется при помощи оптической системы с электрической компенсацией.

Для контроля нефтепродуктопроводов разработан и используется специализированный автоматический фотоэлектрический колориметр погружного типа для осуществления контроля за границами раздела последовательно перекачиваемых бензинов [1].

Отверстие, находящееся в верхней части корпуса 1 фотоэлектрического датчика (рис. 1) [1], которое закрывается крышкой 9 с прокладкой 8, обеспечивает доступ жидкости во внутреннюю полость датчика. У корпуса с двух сторон имеются штепсельные разъемы 4 и 13. Прозрачные защитные окна 7 и 10, которые изготавливают из стекла или пластмассы, пропускающие световой поток в видимой и ИК-областях спектра герметично вклеены в стенки корпуса. Два источника излучения: 5 для работы в видимой области спектра, 3 для калибровки показаний датчика в ИК-области установлены перед защитным окном 7. Интерференционный светофильтр 6, пропускает видимую часть спектра и устанавливается между источником излучения 5 и окном 7. Инфракрасный интерференционный светофильтр 2 ограничивает поток излучения на источник 3.

Фоторезисторы 12 и 14, воспринимающие световой поток видимой части спектра при контроле полупрозрачных жидкостей и инфракрасное излучение при калибровке датчика находятся перед защитным окном 10. Светофильтр 11 пропускает излучение в видимом (зеленом) и инфракрасном диапазонах спектра и находится перед фоторезистором 12, а светофильтр 15 - излучение в видимом (красном) и инфракрасном диапазонах спектра, он установлен перед фоторезистором 14. К штепсельным разъемам 4 и 13 с помощью проводов подключены источники излучения 3, 5 и фоторезисторы 12, 14

Повышение чувствительности и точности контроля за цветом перекачиваемых полупрозрачных жидкостей обеспечивается за счет использования зеленого и красного светофильтров. Например, для коммерческого и премиального бензинов отношение пропусканий излучения красного и зеленого светофильтров изменяется от 0,9 до 7, т.е. максимальный предел отношений на границах раздела бензинов достаточно высокий - 8:1. Также в состав фотоэлектрического колориметра входят электронное устройство, блок питания и регистрирующий прибор.



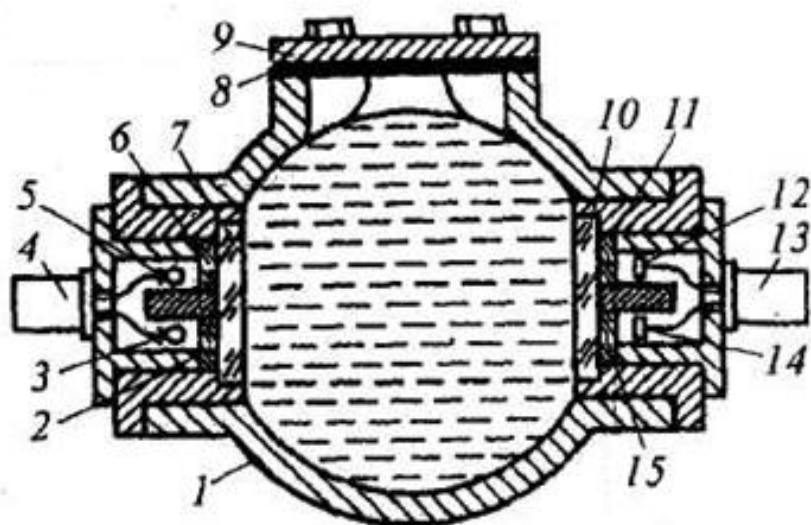


Рис 3. Конструкция фотоэлектрического датчика

Основными недостатками рассмотренного фотоэлектрического датчика являются сложность конструкции из-за необходимости использования светофильтров, что кроме того вносит определенные погрешности, а также важность обеспечения герметичности при контроле.

Авторами была предложена конструкция колориметра [6–8], которая позволяла устранить выше перечисленные недостатки за счет стационарного положения колориметра, встроенного в технологический поток полупрозрачных жидкостей. Кроме того кювету предлагалась изготавливать в виде прозрачной линзы-шара, внутри которой имеется по оси симметрии имеется цилиндрическое отверстие, куда с помощью специальных стоек закрепляется, например, четырехгранный или цилиндрический стержень с посеребренными отражающими поверхностями. Но данная конструкция имеет существенный недостаток: технологическая сложность изготовления линзы-шара, особенно полированной поверхности центрального отверстия.

Оптоэлектронный многопараметровый колориметр не имеет вышеперечисленных недостатков, его гораздо проще изготовить, так как кварцевая кювета состоит из двух полуцилиндрических трубок 15 (рис. 4, 5) [9, 10]. Плоское зеркало с двухсторонним отражением 8 приклеивается между двумя полуцилиндрами. Стеклоградированная воронка 18, куда заливается контролируемая жидкость 9, присоединяется к кювете. Кран 19,

находящийся снизу конструкции, служит для перекрытия жидкости. Вся конструкция помещена в корпус 18 (рис. 4). Датчик можно установить на линии технологического процесса, т.е. контролировать жидкости (соки, напитки, пиво и т.д.), протекающие через трубу по стрелке, указанной на рис. 4.

Кроме того, оптоэлектронный многопараметровый колориметр (рис. 5) состоит из задающего генератора 1 (источника импульсного питания), коммутатора 2, опорного лазерного диода 3, измерительных лазерных диодов 4, 5, 6, 7,  $(n-1)_{из}$ ,  $n_{из}$ , измерительных фотоприемников 11, 12, 13, 14,  $(n-1)_{пр}$ ,  $n_{пр}$ , блока обработки фотоэлектрического сигнала 16, регистрирующего прибора 17 (например, ЭВМ). При включении, задающий генератор 1, вырабатывает прямоугольные импульсы 8 - 10 Гц. Разделенные импульсы через коммутатор - переключатель оптронов 2 подаются попеременно на лазерные диоды 4, 5, 6, 7,  $(n-1)_{из}$ ,  $n_{из}$ , которые имеют определенные спектральные характеристики, соответствующие спектральным характеристикам  $n$  компонентов (параметров), содержащихся в полупрозрачных жидкостях. При этом оптопары могут включаться последовательно. Лазерный диод 3 и измерительный фотоприемник 10 образуют опорную оптопару, которая предусмотрена для того, чтобы сравнивать измерительные сигналы с сигналом опорного канала и обеспечивать достоверность результатов измерения. При заполнении цилиндрического отверстия кюветы 15 контролируемой жидкостью 9 она облучается



лазерными диодами. В первом положении переключателя, поток излучения лазерного диода 4 фокусируется, проходит через контролируемую жидкость 9, отражается от одной из двух поверхностей плоского зеркала 8, и вновь проходя через контролируемый образец 9, далее попадает на измерительный фотоприемник 11. Затем сигналы поступают в блок обработки фотоэлектрического сигнала

(БОФС) - 16, где реализуется отношение сигналов этого измерительного потока и компенсационного - от диода 3 и измерительного фотоприемника 10. Во втором положении переключателя 2 подключается вторая оптопара, процесс измерения повторяется, и так далее происходит для остальных оптопар.

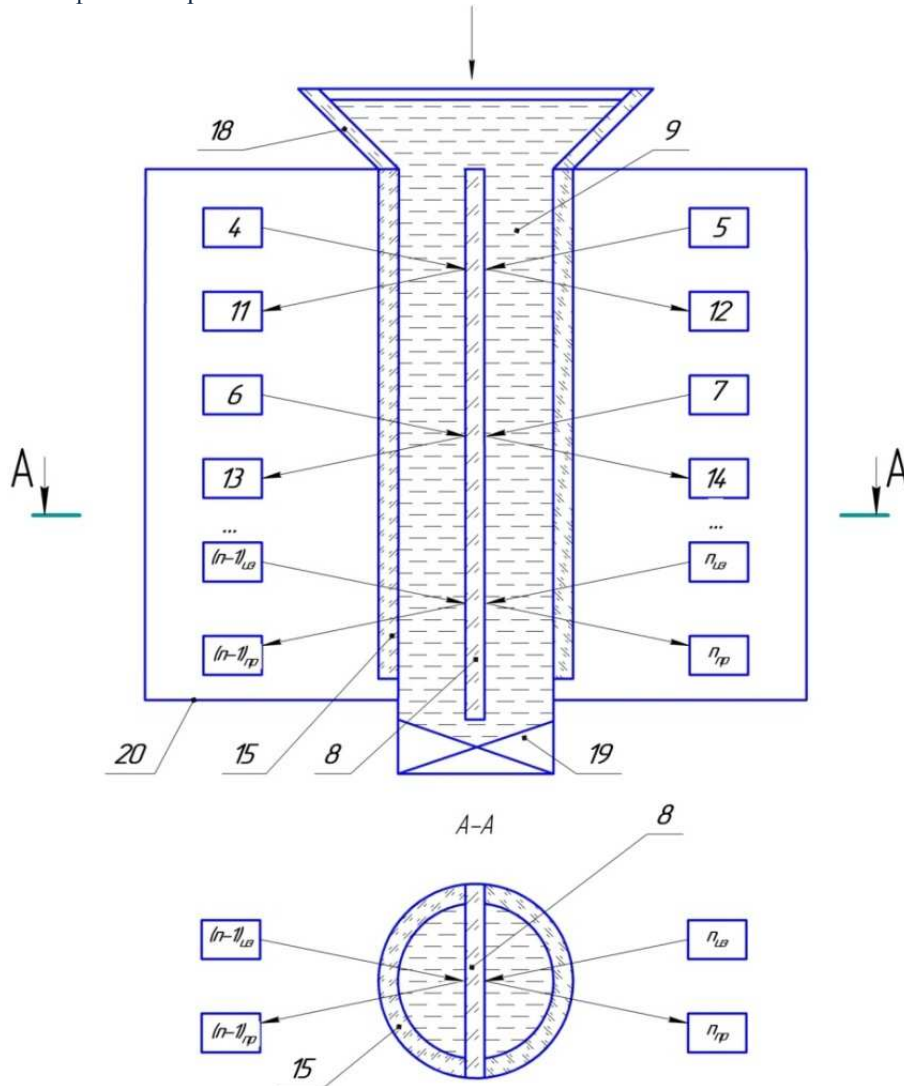


Рис. 4. Конструкция датчика оптоэлектронного многопараметрового колориметра

Полученный сигнал отношения пропорционален величине коэффициента пропускания и оптической плотности жидких сред или прозрачных твердых тел. За счет использования  $n$  оптопар в конструкции датчика можно определить наличие и концентрации веществ (параметров) в растворе. Полученные результаты подаются на измерительный прибор или ЭВМ 17, по показанию которого судят о параметрах жидких сред при этом необходимо наличие предварительно определенных градуи-

ровочных характеристик. ЭВМ обеспечивает автоматическое выполнение аналого-цифровых измерительных преобразований, вычислительных процедур, выдачу полученной информации, формирование командной и другой служебной информации, необходимой для функционирования оптоэлектронных многофункциональных автоматических контрольно-измерительных систем.

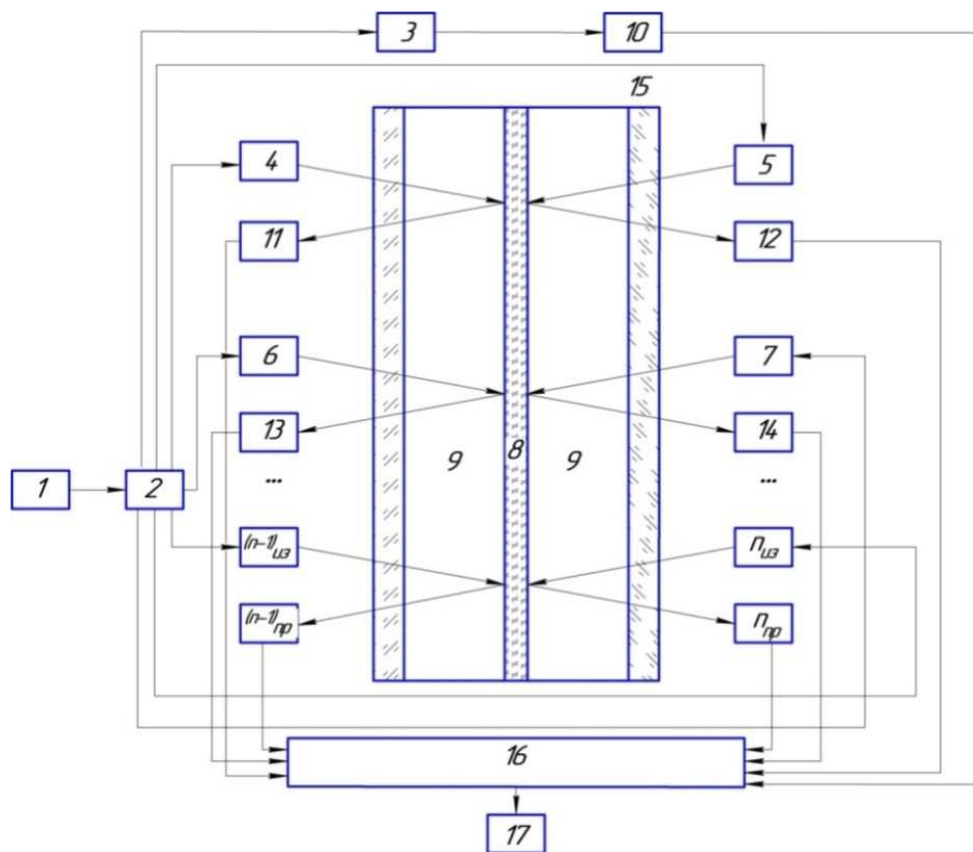


Рис. 5. Блок-схема оптоэлектронного многопараметрового колориметра

Для обеспечения эффективности работы предлагаемых устройств [9 - 10] были экспериментально определены характеристики основных элементов колориметров, такие как проходная характеристика оптронной пары с открытым оптическим каналом (ООК). Для сравнения результатов расчета коэффициента передачи оптического сигнала (ОС) с экспериментальным значением необходима характеристика оптронной пары без ООК. Были сняты характеристики оптронных пар с использованием схемы, показанной на рис. 6, где в качестве светоизлучающих диодов (СИД) использовались следующие: синие (С), зеленые (З), красные (К), инфракрасные (ИК) (рис. 2, 3). В качестве приемника излучения использовался фоторезистор СФ2-4, охватывающий широкую спектральную область. Результаты эксперимента, который проводился при комнатной температуре ( $t = 20 \text{ }^\circ\text{C}$ ).

представлены на рис. 7 в виде графической зависимости тока фотоприемника от тока источника излучения. Кривые, полученные в результате эксперимента, показывают, что характеристики оптронной пары достаточно близки к линейной зависимости.

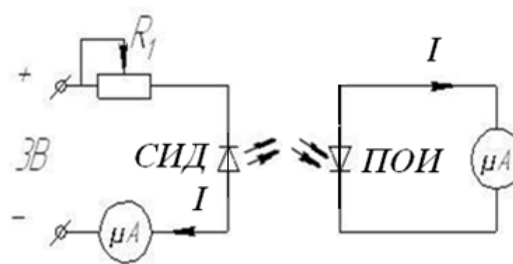


Рис. 6. Схема снятия характеристики оптронной пары

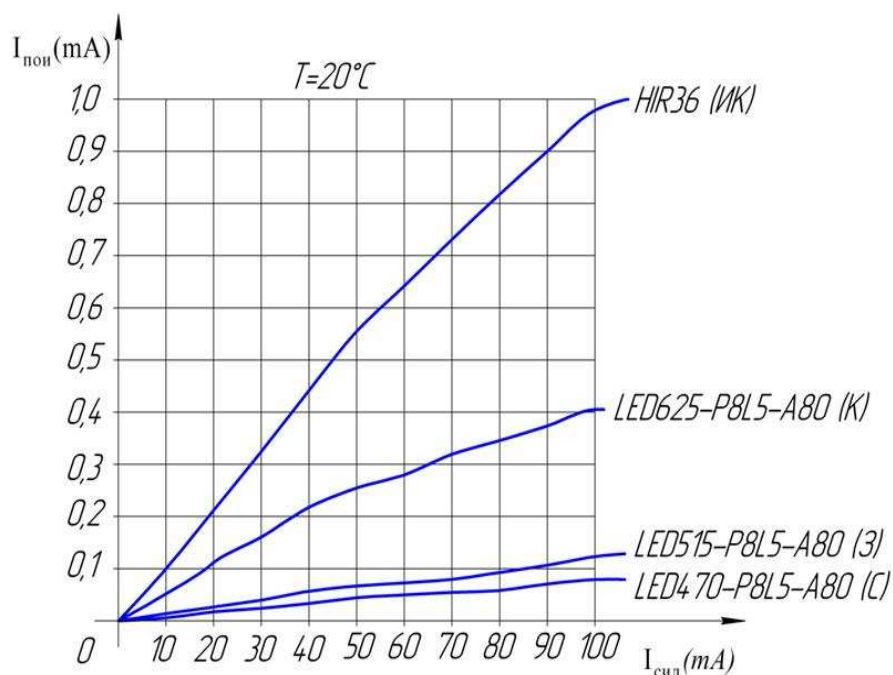


Рис. 7. Характеристики оптронной пары

Эксперимент позволил для каждого спектрального диапазона выявить светодиоды, для которых характеристика идет круче и эти светодиоды обладают большей мощностью излучения по сравнению с остальными. Для датчика имеет смысл брать в качестве источника излучения те светодиоды, которые имеют более крутую характеристику из всех исследованных для соответствующей определенной спектральной области (синих, зеленых, красных, инфракрасных). В результате проведенных экспериментов в качестве источников оптического излучения было принято решение использовать следующие светодиоды: синий «Пиранья» LED470-P8L5-A80, зеленый «Пиранья» LED515-P8L5-A80, красный «Пиранья» LED625-P8L5-A80, инфракрасный «Пиранья» HIR36.

### ЗАКЛЮЧЕНИЕ

Предлагаемый авторами оптоэлектронный многопараметровый имеет небольшие габариты ( $200 \times 50 \times 200$ ), но за счет двукратного прохождения излучения через исследуемую полупрозрачную жидкость обеспечивает точность проводимых измерений. Стационарное расположение кюветы, через которую пропускается жидкость снижает погрешности измерений, которые неизбежны при переустановке датчика. Усовершенствованная конструкция кюветы, позволит упростить ее

изготовление и снизить затраты на стадии ее производства. Наличие n+1 оптопар позволяет контролировать наличие и содержание n компонентов в составе жидкостей, что делает анализ более объективным.

Работа поддержана грантом по проекту «Исследование предельных точностей оптических методов измерения параметров движения и мехатронных методов управления движением и разработка новых робототехнических и электромеханических систем», Темплан, проект № 7.559.2011.

### ЛИТЕРАТУРА

- [1] Неразрушающий контроль: справочник: В 7т./под общ.ред. В.В. Клюева. Т. 6: Магнитные методы контроля, Оптический контроль, Радиоволновый контроль / И.Н. Ермолов. – М.: Машиностроение, 2004.
- [2] Рахимов Н.Р., Касымохунова А.М., Усманов Ш.. Оптоэлектронные средства неразрушающего контроля физико-химических параметров жидких сред / журнал «Техническая диагностика и неразрушающий контроль». – Киев, 2001. № 3. – С. 40–42.
- [3] Рахимов, Н.Р. Оптоэлектронные системы на основе эффекта НПВО для контроля технологических параметров нефти и нефтепродуктов / Н.Р. Рахимов, Л.Ф. Парфирьев // Изв. вузов. Приборостроение, 2006. - № 1. – С. 41 – 45
- [4] Мирзамахмудов Т.М., Рахимов Н.Р., Гафуров У.А., Зокиров Р.З., Атакулов О.Х.. Устройство для определения оптических параметров жидких сред / Авторское свидетельство № 1693482 от 23.11.91 г.
- [5] Глухов В.С. Термохимические методы и приборы для анализа жидких сред. – Армавир, АГПИ, 1998.

© **АВТОМАТИКА И ПРОГРАММНАЯ ИНЖЕНЕРИЯ. 2012, №1(1)**

[6] Пат. 2413201 Российская Федерация, МПК51 G01N 21/03. Оптоэлектронный фотоколориметр/ Рахимов Б.Н., Ушаков О.К., Кутенкова Е.Ю., Ларина Т.В.; заявитель и патентообладатель Государственное образовательное учреждение высшего профессионального образования «Сибирская государственная геодезическая академия». - №2009146659/28; заявл. 15.12.2009; опубл. 27.02.11, Бюл. №6. - 4 с. : ил.

[7] Рахимов, Б.Н. Оптоэлектронный автоматический колориметр/Б.Н. Рахимов, О.К. Ушаков, Е.Ю. Кутенкова, Т.В. Ларина // Приборы и техника эксперимента, 2011. - № 5. - С. 161 – 162.

[8] Rakhimov, B. N. Aspects of Improvement of Automatic Optoelectronics Colorimeter Receiver of Optical Radiation (ROR) / B. N. Rakhimov, E. U. Kutenkova, P. V. Petrov // Proceedings of the RFBR and DST Sponsored "The Second Russian-Indian Joint Workshop on Computational Intelligence and Modern Heuristics in Automation and Robotics" 10 – 13 September, 2011. Additional Volume sponsored by NSTU, Department of Automation, p. 32 – 33.

[9] Заявка № 2011153209 Российская федерация. Оптоэлектронный многопараметровый колориметр/Б.Н. Рахимов и др.; Государственное образовательное учреждение высшего профессионального образования «Сибирская государственная геодезическая академия».

[10] Ушаков, О.К. Автоматический колориметр на основе оптоэлектронных преобразователей/ О.К. Ушаков, Е.Ю. Кутенкова, А.Н. Серьезнов / Интерэкспо ГЕО – Сибирь-2012. VIII Междунар. науч. конгр., 10 – 20 апреля 2012 г. Новосибирск : Междунар. науч. конф. «Специализированное приборостроение, метрология, теплофизика, микротехника, нанотехнологии»: сб. материалов в 2 т. Т.1. – Новосибирск : СГГА, 2012. – С. 146 - 151.

*Елена Юрьевна Кутенкова* - доцент кафедры ТОП ФГБОУ ВПО "СГГА".

E-mail: [kutenkova.elena@yandex.ru](mailto:kutenkova.elena@yandex.ru)

*Рахимов Бахтиержон Нематович* - аспирант ТУИТ, Ташкент.

E-mail: [nerah@rambler.ru](mailto:nerah@rambler.ru)

*Татьяна Вячеславовна Ларина* - доцент кафедры ТОП ФГБОУ ВПО "СГГА".

E-mail: [larina\\_t\\_v@mail.ru](mailto:larina_t_v@mail.ru)

*Шерзод Ильхомович Мадумаров* - аспирант НГТУ кафедры Автоматики.

E-mail: [fergana\\_10@mail.ru](mailto:fergana_10@mail.ru)

# Оптоэлектронный метод контроля физико-механических параметров поверхностей твердых материалов

Т.В. Ларина, Е.Ю. Кутенкова, В.А. Жмудь, Д.Д. Алижанов

**Аннотация.** В статье рассмотрен оптоэлектронный неразрушающий метод контроля усталости металлических конструкций. Также предлагается оптоэлектронная система, которая состоит из датчика и электронного блока. Представляется один из вариантов выполнения датчика, изготовленный в виде полутрубчатой основы или полого полуцилиндра, в которую установлены три пары Y-образных и три обычных подводных и шесть отводящих оптоволокон.

**Ключевые слова:** источники и приемники оптического излучения, оптоволокно, оптоэлектронная система, зернистость, шероховатость поверхности, дефектность, цветность.

Контроль усталости металлических поверхностей – это одна из самых трудоемких технологических операций в машиностроении, самолетостроении и т. п., ибо ни одна деталь не может быть изготовлена без измерения ее технических характеристик. В связи с усложнением и требованием неуклонного повышения надежности новой техники трудоемкости контрольных операций в промышленности резко увеличиваются. Например, в развитых странах затраты на мониторинг металлических поверхностей составляют в среднем 1,5...3,5% от стоимости выпускаемой продукция, а в отдельных т.н. наукоемких отраслях – до 20% [1].

Детали, подвергающиеся воздействию переменных напряжений, разрушаются при напряжениях, значительно меньших значений предела прочности, а иногда и предела пропорциональности материала. Явление разрушения под действием переменных напряжений называется усталостью материала. Процесс постепенного накопления повреждений под действием переменных напряжений, приводящий к изменению свойств материала, образованию трещин и разрушению детали, называется усталостным разрушением (усталостью).

Механизм усталостного разрушения связан с неоднородностью материала: различных размеров и конфигураций отдельных зерен, направлением их кристаллографических плоскостей, наличием неоднородных фаз, включений, дефектов кристаллической решетки (вакансий, дислокаций), остаточных напряжений [2-5].

Изменение состояния материала при усталостном процессе отражается на таких его параметрах и

свойствах, как зернистость, шероховатость, дефектность (усадочная рыхлость, газовые пузыри, пустоты и трещины) и цветность поверхности.

Контроль этих качественных параметров металлических поверхностей осуществляется рядом методов разрушающего и неразрушающего контроля, основными видами которого являются [6]:

- радиационный - выявление неисправностей на основе анализа изменения ионизирующего излучения, пропускаемого через объект контроля;
- магнитный - так называемая магнитнопорошковая дефектоскопия, используется для выявления дефектов металлических конструкций;
- визуально-измерительный, с которого и начинается проведение неразрушающего контроля, позволяет отметить наиболее грубые, визуально заметные дефекты и нарушения;
- капиллярный (капиллярная дефектоскопия) - предназначен для обнаружения открытых дефектов, выходящих на поверхность;
- тепловой - для оценки качества объекта применяется его инфракрасная съемка;
- акустический.

В настоящее время в качестве лабораторного метода в основном используется визуально-измерительные методы с применением электронной микроскопии - просвечивающей и сканирующей. Они перекрывают весь диапазон необходимых в исследовательской и аналитической практике увеличений: от фрактографии изломов, до атомной структуры. Наиболее важным (и обязательным) в практике научных исследований является метод просвечивающей электронной микроскопии. Универсальность этого метода определяется не только уникальной шириной диапазона увеличений, но и тем, что характеристики микроструктуры дополняются характеристиками элементного состава в самых малых объемах и кристаллоструктурными характеристиками для тех же объемов.

Метод растровой электронной микроскопии позволяет:

- изучать микротопографию (фрактографию) поверхностей разрушенных образцов (изломов) после испытания, и проведя классификацию видов разрушения, в дальнейшем по виду излома судить об энергоемкости процесса разрушения;
- изучать поверхность металлоконструкций с целью обнаружения микротрещин;
- давать количественную оценку шероховатости поверхности металла с помощью



специальных профилограмм, основанных на измерении тока отраженных электронов;

– изучать кинетику роста трещины в образце и отвечать на вопросы, где зародилась магистральная трещина, каково направление и скорость ее распространения;

– изучать влияние фазовых выделений на механизмы разрушения;

– исследовать связь микротопографии изломов образцов с исходной структурой для выявления мест предпочтительного зарождения и распространения трещин;

– изучать морфологию фазовых составляющих и их объемную конфигурацию на поверхности травленных микрошлифов и изломов.

Если растровая электронная микроскопия может объяснить, как произошло разрушение в исследуемом материале изделия, как металлическая поверхность детали откликается на термопластическое воздействие внешней среды, то просвечивающая электронная микроскопия может объяснить, почему это происходит, как этому способствует структурно-фазовое состояние материала. Но, к сожалению, электронная микроскопия ограничена в своих возможностях по исследованию и диагностике поверхности. Несмотря на огромные плюсы, которые она имеет, существует несколько неоспоримых недостатков:

– дорого стоит и сам электронный микроскоп, и его обслуживание;

– подготовка материала для работы с микроскопом отнимает много времени и требует высокой квалификации персонала;

– исследуемые образцы под действием пучка электронов постепенно разрушаются. Поэтому, если требуется детальное изучение образца, необходимо его фотографировать;

– необходимость достаточного вакуума для получения относительно хорошего разрешения;

– отсутствие возможности просмотра больших образцов.

Из вышеперечисленных методов одним из наиболее распространенных видов неразрушающего автоматического контроля является акустический метод. Его популярность обусловлена легкостью введения и распространения ультразвуковых волн в объекты контроля и достаточно эффективным выявлением дефектов (таких как трещины, царапины, поры и др.), а также высокой чувствительностью ультразвуковых волн к изменению структуры и свойств материалов, и возможностью контроля больших толщин (до 2-5 м в зависимости от структуры металла). Помимо этого данный метод безопасен для окружающих, обладает высокой производительностью, малой стоимостью контроля и возможностью автоматизации (при простой геометрической форме изделия) [7].

Главными недостатками данного метода для правильного проведения контроля и получения объективных результатов являются квалификация, теоретическая и практическая подготовка специалиста. Также необходимо создать

акустический контакт через жидкую среду и существуют ограничения по чистоте обработки поверхности. Характерна малая чувствительность при контроле крупнозернистых материалов, отсутствует наглядность и отмечены сложность расшифровки результатов контроля, оценка размеров и формы дефектов с большими погрешностями.

В данной работе предлагается оптоэлектронный неразрушающий метод контроля усталости металлических конструкций, который представляется перспективным в отношении простоты и удобства. В данном методе применяется принцип сравнения качественных параметров исследуемого объекта с эталоном, в роли которого могут применяться образцовые стандартизованные меры зернистости, шероховатости и дефектности. Теоретические и экспериментальные исследования для зеркальной составляющей отраженного потока для шероховатой поверхности со случайным распределением элементов профиля показывают, что возможно применение этого метода для определения шероховатости металлических поверхностей с  $R_z$  до 0,2 мкм.

Основной измеряемой оптической характеристикой при данном методе выступает интенсивность отраженного светового потока от поверхности контролируемого объекта (КО), которая сравнивается с величиной аналогичной характеристики эталона. При этом измеряемой электрической характеристикой выступает сила тока или напряжение на приемнике оптического излучения (ПОИ). Они определяются выражениями  $U_{\Phi} = S_{VEe} \cdot E_e$  и  $I_{\Phi} = S_{IEv} \cdot E_v$ , где  $U_{\Phi}$  и  $I_{\Phi}$  – соответственно, напряжение фотосигнала и фототок;  $S_{VEe}$  и  $S_{IEv}$  – соответственно, вольтовая чувствительность к облученности и токовая чувствительность к освещенности;  $E_e$  и  $E_v$  – облученность и освещенность.

При исследовании цветности определяющей выступает четкость спектра излучения светоизлучающего диода (СИД) и монохроматическая чувствительность ПОИ, для других параметров – интегральная чувствительность ПОИ.

Происходит сравнение измеряемого параметра КО и с любым из ряда образцовых, хранящихся в запоминающем устройстве (ЗУ) и сравнение их в блоке обработки фотозлектрического сигнала (БОФС). КО подсвечивается соответствующим идентичным СИД, который питается импульсным током от задающего генератора (ЗГ) через коммутатор и разделяющий триггер (РТ). Далее результирующий сигнал из ПОИ сопоставляется в БОФС с любым из сигналов, хранящихся в ЗУ. Затем оба сигнала или их разность подаются на измерительный прибор (ИП) или в ЭВМ для хранения и дальнейшей обработки. Удобство данного решения в том, что сопоставление полученного сигнала с эталонным позволяет быстро и с достаточной точностью определить количественные значения параметров и отнести их к тому или иному классу (зернистости, шероховатости

и т.п.).

К настоящему времени разработан ряд оптоэлектронных приборов для неразрушающего контроля различных веществ и материалов [8-10].

На рис. 1 представлена структурная схема оптоэлектронной системы, а на рис. 2 - один из вариантов выполнения датчика.

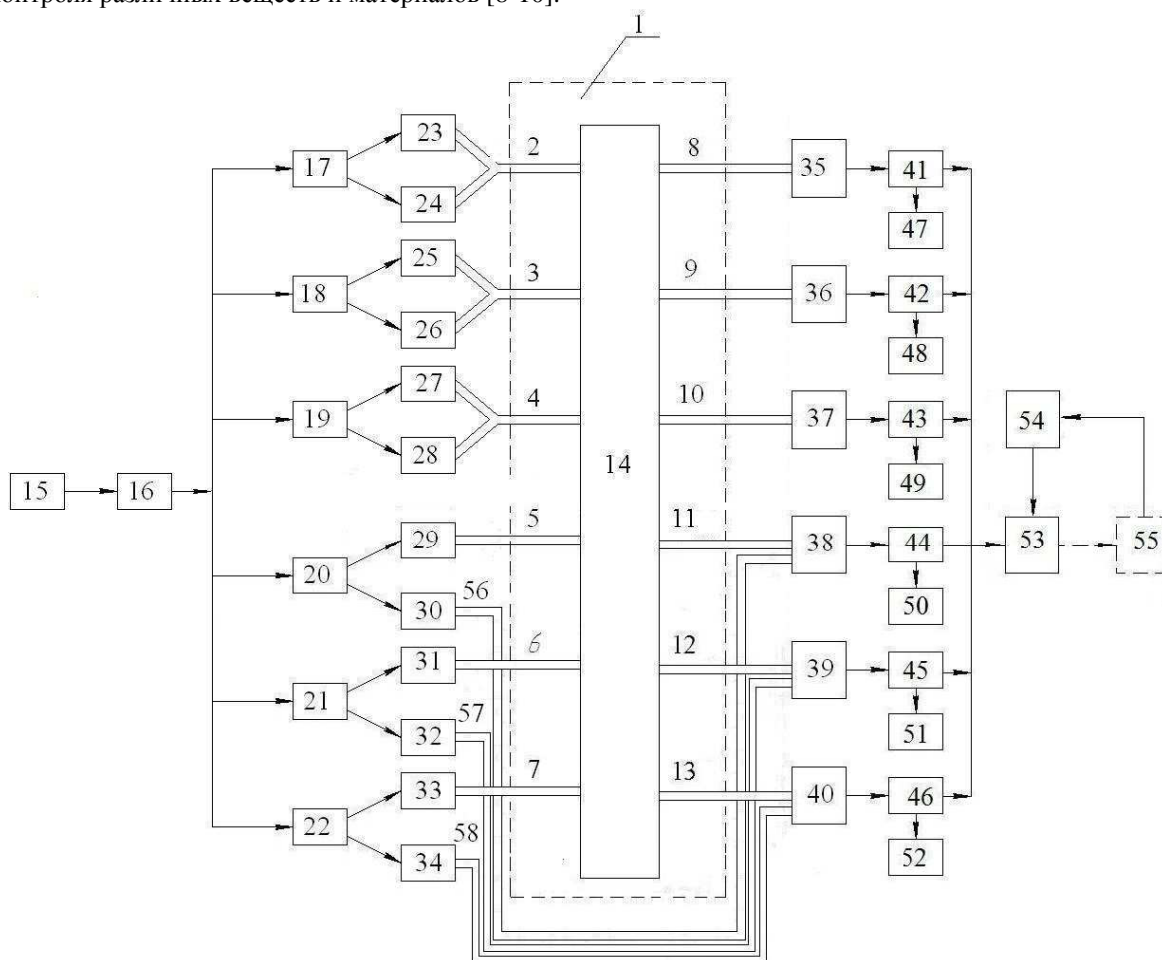


Рис. 1. Структурная схема оптоэлектронной системы для контроля усталости металлических конструкций

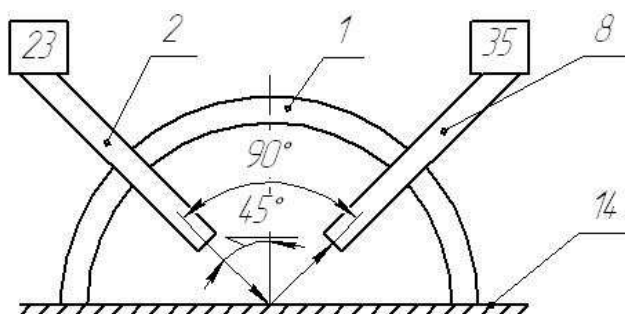


Рис. 2. Вариант выполнения датчика

Оптоэлектронная система скомплектована следующим образом, она включает в себя разветвленную структуру, содержащую шесть самостоятельных ветвей (цепочек). Каждая цепочка отвечает за контроль своего параметра металлической конструкции, например, первая, вторая и третья - за контроль цветности, четвертая - за шероховатость поверхности, пятая - за зернистость и шестая - за дефектность.

Датчик выполнен в виде полутрубчатой основы или полого полуцилиндра 1, в которую установлены

три пары Y - образных 2-4 и три обычных 5-7 подводящих оптоволокон, шесть отводящих 8-13 оптоволокон, а три опорные оптоволокна 56-58 входят в состав электронного блока.

Электронный блок включает в себя ЗГ 15, коммутатор 16, шесть триггеров 17-22, шесть выходов которых соединены с шестью измерительными СИД 23, 25, 27, 29, 31, 33 вторые шесть выходов - с компенсационными СИД 24, 26, 28, 30, 32, 34. Все измерительные СИД соединены подводящими оптоволоконками 2-7 с КО 14, а

компенсационные СИД соединены иначе. Например, СИД 24, 26 и 28 соединены через КО с соответствующими ПОИ, а компенсационные СИД 30, 32 и 34 соединены с соответствующими ПОИ 38, 39 и 40 оптическими опорными волокнами 56, 57 и 58 минуя КО 14. Выходы каждого из шести ПОИ 35 - 40 подсоединены к входу соответствующего блока сравнения 41 - 46, выход каждого из которых соединен с соответствующим ИП 47 - 52. Затем электронный блок включает в себя БОФС 53, ЗУ 54 и измерительную систему, например, ЭВМ - 55.

Устройство работает следующим образом. ЗГ 15 вырабатывает прямоугольные импульсы, которые подаются на вход коммутатора 16. Разделенные импульсы далее подаются на вход шести идентичных триггеров 17-22, шесть выходов которых соединены с шестью измерительными СИД 23, 25, 27, 29,31, 33, вторые три выхода – с компенсационными СИД 24, 26, 28, 30, 32, 34, импульсы от триггеров поступают на соответствующие СИД.

КО 14 заключен в полутрубу или в полый полуцилиндр 1, с подводными Y- образными 2 - 4 и тремя обычными 5 – 7 оптическими волокнами облучается двумя световыми потоками измерительным и компенсационным.

Оптоволокна заключены в полутрубчатый кожух из мягкой резины для необходимой ориентации датчика и светоизоляции оптического канала и расположены они, например, под углом 45°, относительно нормали к КО в точке отражения при контроле цветности, и под углом  $\approx 30^\circ$  - при контроле шероховатости поверхности и др.

При контроле цветности оптическое излучение отражается от КО 14 и отводящими оптическими волокнами 8 - 10 подается на ПОИ 35-37, которые преобразуют оптические сигналы в электрические. Данные ПОИ, работают на длинах волн  $\lambda_1=680$  нм,  $\lambda_2=560$  нм и  $\lambda_3=450$  нм.

Благодаря прохождению света через подводные 2-4 и отводящие 8-10 оптоволокна подается и принимается узкий пучок излучения, который дает возможность контроля параметров цветности.

Физический смысл заключается в следующем: цветовые параметры определяются как объективное свойство предметов, проявляющееся в спектральном составе исходящего от них (пропускаемого, отражаемого) излучения и воспринимаемое как осознанное зрительное ощущение. В этом определении даны два аспекта – физический и психофизиологический, неразрывно связанные друг с другом.

Современная теория распознавания цвета основана на однозначно установленном факте трихроматности зрения человека, т.е. зрительный аппарат содержит три вида рецепторов, каждый из которых преимущественно реагирует на красный, зеленый или синий цвет.

Согласно этому, цветовой параметр математически выражается вектором в трехмерном

цветовом пространстве, причем начало этого вектора совпадает с началом цветовой координатной системы (ЦКС). Если в качестве основных цветов использовать единичные векторы трех цветов красного  $r_n$ , зеленого  $g_n$  и синего  $b_n$ , то любой цвет можно выразит в виде:

$$C = R r_n + G g_n + B b_n \quad (1)$$

где R, G, B – три основных цвета.

На *рис. 3* показаны кривые сложения цветов систем RGB (а) и XYZ (б). RGB-система называется эмпирической, которая используется в качестве основных цветов чистого спектрального излучения красного ( $\lambda=680$  нм), зеленого ( $\lambda=560$  нм) и синего ( $\lambda=450$  нм) цветов. А XYZ называется феноменологической системой. Смысл этих систем заключается в том, что спектрально чистый цвет с  $\lambda=600$  нм воспринимается глазом как состоящий из красного и зеленого компонентов в соотношении 14:3, излучение с  $\lambda=450$  нм воспринимается в виде соотношения 7,5:1:35 красного, зеленого и синего цветов соответственно [11].

В системе RGB кривые красного  $r_n$ , зеленого  $g_n$  и синего  $b_n$  построены так, что для каждого из трех основных цветов отличаются от нуля лишь ординатой одной кривой. Принципиальный недостаток этих кривых сложения заключается в наличии отрицательного участка у кривой. При измерении цветовых координат вычитание цвета реализовать невозможно, поэтому в колориметрии это система не применяется.

Таким образом, задача контроля цветовых параметров на основе XYZ системы осуществляется тремя способами. Первый – это визуальное сопоставление измеряемого цвета с эталонным. Эталон выбирается из заранее составленного цветового атласа или компьютерным программированием каждого из цветов. Второй – это спектрофотометрирование, наблюдаемого излучения и расчет координат X, Y, Z. Третий – это непосредственное измерение координат X, Y, Z при использовании трех эталонных ПОИ, спектральные функции, чувствительности которых в точности соответствуют кривым  $Y_x$ ,  $Y_y$ ,  $Y_z$ . Этот способ, несомненно, является наиболее перспективным, так как он удовлетворяет требованиям технологического процесса.

Для разработки по этому способу устройство должно выполнить следующие операции:

- выделение анализирующего излучения на трех длинах волн (красный, зеленый и синий);
- прием отражающих через КО потоков излучения и преобразование их в фотоэлектрический сигнал;
- обработка и сравнение фотоэлектрического сигнала.

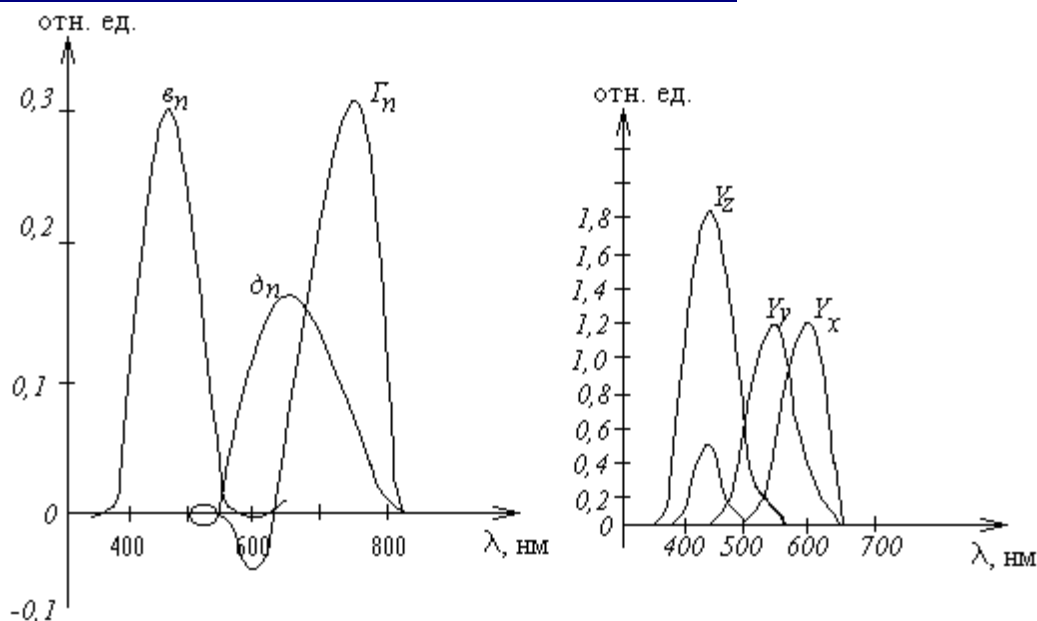


Рис.3 показаны кривые сложения цветов систем RGB (а) и XYZ(б)

Назначение каждой из трех оптопар поясняется следующим образом: КО облучается двумя световыми потоками с длинами волн  $\lambda_1$  и  $\lambda_2$ , один из которых является измерительным излучением, а другой - компенсационным.

Пусть на КО падает световой поток  $\Phi_{0\lambda}$ . Освещенный слой разделит упавший на него световой поток на три части:

$\Phi_{\lambda 1}$  – отраженный от поверхности и попадающий на ПОИ, из которой пришел и падающий поток;

$\Phi_{\lambda 2}$  – отраженный от поверхности и не падающий на ПОИ;

$\Phi_{\lambda 3}$  – поглощенный поток, который в веществе слоя превратится в теплоту или другую форму энергии.

В соответствии с законом сохранения энергии сумма световых потоков равна падающему потоку:

$$\Phi_{0\lambda} = \Phi_{\lambda 1} + \Phi_{\lambda 2} + \Phi_{\lambda 3} \quad (2)$$

или

$$\frac{\Phi_{\lambda 1}}{\Phi_{0\lambda}} + \frac{\Phi_{\lambda 2}}{\Phi_{0\lambda}} + \frac{\Phi_{\lambda 3}}{\Phi_{0\lambda}} = 1 \quad (3)$$

Для окрашенных веществ эти коэффициенты зависят от спектрального состава падающего излучения. Для монохроматического излучения с определенной длиной волны (как спектр излучения светоизлучающего диода), их обозначить  $\rho(\lambda)$ ,  $\tau(\lambda)$  и  $\alpha(\lambda)$ .

$$\lambda = \frac{1}{2} \Delta \lambda \quad (4)$$

Спектральную зависимость этих коэффициентов удобно изображать графически.

Предлагаемое устройство обладает повышенной точностью распознавания цвета за счет трех оптопар, которые контролируют три цветовых параметра, соответствующие параметрам X, Y, Z.

Устройство обладает повышенной точностью контроля за счет трехмерного измерения с длинами волн  $\lambda_1 = 680$  нм,  $\lambda_2 = 560$  нм,  $\lambda_3 = 450$  нм, когда в

качестве эталонных ПОИ применены многоцветные фоторезисторы. При необходимости сигнал с выхода БОФС можно подать в систему автоматического контроля.

Далее сигналы попадают на соответствующие блоки сравнения 41 - 43, где берется отношение двух сигналов (измерительного и компенсационного) и затем измерительной системой 47 - 49 определяются оттенки трех цветов. Процесс измерения цветности на данном этапе можно закончить.

Следующие параметры контроля, такие как шероховатость, баллы зернистости и наличие дефектов (трещин, рыхлости) измеряются иначе. Например, сигнал от измерительного СИД 29 через подводящее оптоволокно 5 попадает на КО 14, и, отражаясь от него, поступает по отводящему оптоволокну 11 на ПОИ 38, который отвечает за шероховатость и работает на длинах волн  $\lambda_4 = 860$  нм (ПОИ 39 – отвечает за зернистость и работает на длинах волн  $\lambda_5 = 1350$  нм, ПОИ 40 – отвечает за дефектность и работает на длинах волн  $\lambda_6 = 2800$  нм).

На ПОИ 38 также попадает второй сигнал от компенсационного СИД 30 через опорное оптоволокно 56, минуя КО 14, и затем оба сигнала сравниваются между собой в блоке сравнения 44. Контроль двух других параметров выполняется аналогичным образом.

Физический механизм системы поясняется следующим образом. Например, для определения шероховатости через 29 СИД подается измерительный сигнал в виде светового потока  $\Phi_{04}$ , а отраженный сигнал  $\Phi_4$  при этом равен:

$$\Phi_4 = \Phi_{04} \cdot e^{-k} \quad (5)$$

где  $e$  – основание натурального логарифма;  
 $k$  – коэффициент поглощения и рассеивания, зависящий от шероховатости.

Второй компенсационный поток  $\Phi_{05}$  от СИД 30 попадает через 56 опорное оптоволокно на ПОИ 38, при этом сигнал  $\Phi_{05}$  будет иметь вид:

$$\Phi_5 = \Phi_{05} \quad (6)$$



Если начальные потоки идентичны, то есть  $\Phi_{04} \approx \Phi_{05}$ , то отношение сигналов имеет вид:

$$\frac{\Phi_{15}}{\Phi_{14}} = \frac{\Phi_{15}}{\Phi_{14}} \cdot e^{\epsilon} \quad (7)$$

Сигнал будет прямо пропорционален:

$$\frac{\Phi_{15}}{\Phi_{14}} = e^{\epsilon} \quad (8)$$

Основной измеряемой оптической характеристикой при данном методе выступает интенсивность отраженного светового потока от поверхности контролируемого объекта (КО), которая сравнивается с величиной аналогичной характеристики эталона.

Затем шесть сигналов могут поступать на БОФС 53, где сопоставляются с любым из ряда образцовых, хранящихся в ЗУ 54. Далее шесть сигналов или их отношение подаются на измерительную систему или в ЭВМ 55. При необходимости сигнал с выхода БОФС можно подать в систему автоматического контроля.

На основе данного метода разработана оптоэлектронная система для контроля усталости металлических конструкций. Прибор имеет высокие технические характеристики, что делает возможным его применение в исследовательских целях: напряжение питания – 220 В; частота – 50 Гц; потребляемая мощность – не более 5 Вт; длина волны источников излучений,  $\lambda_1 = 2800 \text{ нм}$ ,  $\lambda_2 = 1350 \text{ нм}$ ,  $\lambda_3 = 860 \text{ нм}$ ,  $\lambda_4 = 680 \text{ нм}$ ,  $\lambda_5 = 560 \text{ нм}$ ,  $\lambda_6 = 450 \text{ нм}$ ; предел измерения цвета (ЦНТ) - 0,010Б; чувствительность (ЦНТ) – 0,01; шероховатость, Ra = 80..0,2 мкм; зернистость, номер зерна (баллы) – от минус 3 до 14; дефектность (усадочная рыхлость, газовые пузыри, пустоты и трещины) – контролируется на выходе (проверяется, имеется ли); габариты прибора – 400 × 100 × 200 мм<sup>3</sup>; масса – не более 3,5 кг.

Это устройство позволяют заменить визуальный контроль технологических параметров исследуемой поверхности по ГОСТ 9378-93 сравниваемой с эталонными образцами, причем расчеты показали, что стоимость устройства окажется в несколько раз ниже стоимости зарубежных приборов. При этом объективность, оперативность и точность контроля данным устройством намного выше, чем при визуальном контроле. Также устройство можно встраивать и в автоматизированное оборудование для одновременного контроля нескольких технологических параметров металлических поверхностей как шероховатость, зернистость, цветность и дефективность.

## ЛИТЕРАТУРА

- [1] Клюев, В. В. Приборы для неразрушающего контроля материалов и изделий/ В. В. Клюев.- М.: Машиностроение, 1986. - 382 с.
- [2] Терентьев, В. Ф. Усталость металлических материалов. М., Наука. 2003. 254.
- [3] В. Ф. Терентьев. О пределе выносливости металлических материалов. Металловедение и термическая обработка металлов. N7. 2004. 15–19.
- [4] Москвичев В.В. Трещиностойкость и механические свойства конструкционных материалов, 2002, 335 с.
- [5] Трошенко В.Т. Трещиностойкость металлов при циклическом нагружении, 1987, 255 с.
- [6] Ермолов, И.Н. Неразрушающий контроль: Справочник. В 7-ми т. / Под ред. В.В. Клюева. Т. 6. Магнитные методы контроля, Оптический контроль, Радиоволновый контроль. М.: Машиностроение, 2004.
- [7] Н. А. Семашко, В. И. Шпорт, Б. Л. Марьин. Акустическая эмиссия в экспериментальном материаловедении. М., Машиностроение. 2002. 239.
- [8] Пат. №2428728 Российская Федерация, МПК 51G01J 3/46. Анализатор цвета поверхности твердых материалов/ Рахимов Б.Н., Ушаков О.К., Кутенкова Е.Ю., Ларина Т.В., заявитель и патентообладатель Государственное образовательное учреждение высшего профессионального образования «Сибирская государственная геодезическая академия». - № 2009147534/28; заявл. 21.12.2009; опубл. 20.09.2011 Бюл. № 26. –7с.: 3 ил.
- [9] Ушаков, О.К. Физические основы применения оптоэлектронного метода с использованием световодов для контроля качественных параметров металлических поверхностей / О.К. Ушаков, Н.Р. Рахимов, Т.В. Ларина, Е.Ю. Кутенкова, В.А. Плиско // ГЕО – Сибирь - 2010: сб. материалов VI Междунар. науч. конгр. «ГЕО – Сибирь - 2010», 19 - 29 апреля 2010 г., Новосибирск.- Новосибирск: СГГА, 2010. – Т.5., ч.1. – С. 179 – 185.
- [10] Ларина, Т.В. Исследование оптонов открытого канала для контроля качественных параметров металлических поверхностей // ГЕО – Сибирь - 2011: сб. материалов VII Междунар. науч. конгр. «ГЕО – Сибирь - 2011», 19 - 29 апреля 2011 г., Новосибирск.- Новосибирск: СГГА, 2011. – Т.5., ч.1. – С. 120 – 125.
- [11] Джадд, Д. Цвет в науке и технике/Д. Джадд, Г. Вышецки.- М.: Мир, 1978. -592 с.

**Ларина Татьяна Вячеславовна**, доцент, ФГБОУ ВПО «СГГА», каф. ТОП.

e-mail: [larina\\_t\\_v@mail.ru](mailto:larina_t_v@mail.ru)

**Кутенкова Елена Юрьевна**, доцент, ФГБОУ ВПО «СГГА», каф. ТОП ИОиОТ.

e-mail: [kutenkova.elena@yandex.ru](mailto:kutenkova.elena@yandex.ru)

**Вадим Аркадьевич Жмудь**, Новосибирский Государственный Технический Университет (НГТУ), д.т.н., профессор заведующий кафедрой Автоматики,

e-mail: [oao\\_nips@bk.ru](mailto:oao_nips@bk.ru)

**Алижанов Донёрбек Дилшодович**, аспирант НГТУ Российская Федерация – 643 Новосибирская обл., г. Новосибирск 630087, ул. Немировича-Данченко 138, к.508.

e-mail: [doni.al@mail.ru](mailto:doni.al@mail.ru)

**II**  
**Робототехника**  
**Программные средства и системы**  
**Свободное программное**  
**обеспечение**  
**Открытое программное обеспечение**



### Содержание раздела:

- А.Б. Колкер, Д.А. Ливенец, Кошелева А.И. [Обоснование выбора программного обеспечения для робототехники](#). С. 51–64.
- А.Б. Колкер, Д.А. Ливенец, Кошелева А.И. [Разработка системы управления стенда для исследования переходных процессов вакуумного высоковольтного выключателя](#). С. 65–72.
- А.Б. Колкер, Д.А. Ливенец, Кошелева А.И., В.А. Жмудь. [Разработка блока системы управления роботом с применением инженерного пакета SciLab](#). С. 73–83.
- А.Б. Колкер, Д.А. Ливенец, Кошелева А.И., В.А. Жмудь. [Исследование вариантов создания интеллектуальных систем робототехники на базе одноплатных компьютеров и свободных операционных систем](#). С. 84–98.

# Обоснование выбора программного обеспечения для робототехники

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

**Аннотация:** В работе дан обзор основных программных средств для графического программирования в целях робототехники.

**Ключевые слова:** Программное обеспечение, робототехника, *MatLab*, *Octave*, *Macsyma*, *Mathcad*, *Scilab*, *LabVIEW*.

## 1. ВВЕДЕНИЕ

Создание роботов требует использования программных средств моделирования, тестирования и отладки управляющих программ. Важными критериями этого выбора являются функциональные возможности, ресурсные ограничения, стоимость продукта, лицензионные ограничения, доступность литературы. В данной работе обосновывается выбор свободного программного обеспечения для этих целей.

В работе дано сравнение преимуществ и недостатков инженерных и математических пакетов для визуального построения системы управления роботами с позиции поставленной задачи конструирования алгоритма управления использованием визуальных методов программирования. В итоге требуется получение исполняемого кода, запускаемого под управлением операционной системы *Linux*. Совместимость с ОС *Windows* желательна, но не обязательна.

## 2. ОСОБЕННОСТИ ИСПОЛЬЗОВАНИЯ ПРОГРАММЫ MATLAB

Рассмотрим преимущества и недостатки использования программы *MatLab* (*MATrix LABoratory*).

Минимальные требования к системе:

- процессор *Pentium III*, 4, *Xeon*, *Pentium M*; *AMD Athlon*, *Athlon XP*, *Athlon MP*;
- 256 Мбайт оперативной памяти (рекомендуемое количество 512 Мбайт);
- 400 Мбайт дискового пространства (только для самой системы *MatLab* и ее *Help*);
- операционная система *Microsoft Windows 2000 (SP3)/XP*.

Стоимость лицензии *MatLab* для коммерческих организаций составляет около \$ 2880, библиотеки визуального моделирования *Simulink* - \$ 4220, необходимые библиотеки докупаются отдельно (к примеру, стоимость библиотеки *Real-Time WindowsTarget* для модуля *Simulink* составляет \$ 3000).

*MatLab* – одна из старейших, тщательно проработанных и проверенных временем систем

автоматизации математических расчетов, которая построена на расширенном представлении и применении матричных операций [1] (см. рис. 1). Принципы и функциональные возможности программы по работе с матрицами создатели языка выразили в лозунге «думай векторно» (англ. *Think vectorized*).

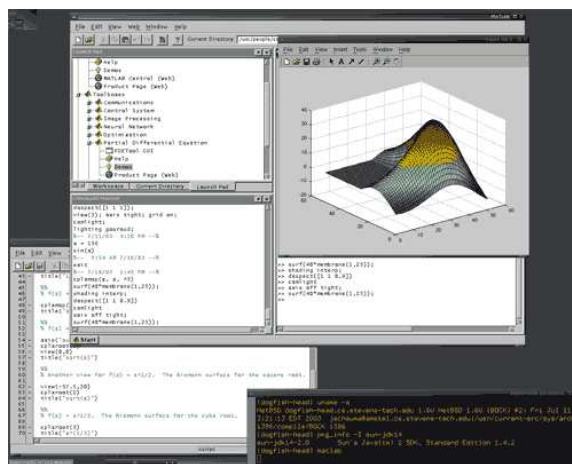


Рис.1. Интерфейс *MatLab*

*MatLab* был разработан в конце 1970-х годов Кливом Моулером (англ. *Cleve Moler*), занимающим должность декана факультета компьютерных наук в Университете Нью-Мексико, с целью разработки было избавить студентов факультета от необходимости изучения языка программирования Фортран и дать возможность использовать программные библиотеки *Linpack* и *EISPACK*. Вскоре новый язык распространился среди других университетов и очень заинтересовал учёных, работающих в области прикладной математики. В сети Интернет можно найти версию 1982 года, написанную на Фортране, распространяемую с открытым исходным кодом. Инженер Джон Литтл (англ. *John N. Little*) был привлечен к разработке проекта во время визита Клива Моулера в Стэнфордский университет в 1983 году. Джона Литтла заинтересовал новый язык тем, что обладает большим коммерческим потенциалом. Разработчиками коммерческой версии *MatLab* стали Джон Литтл Клив Моулер и Стив Бангерт (англ. *Steve Bangert*), переписав программу на языке C. В 1984 они основали компанию *The MathWorks* для дальнейшего развития. Изначально переписанные на C библиотеки выпускались под именем *JACKPAC*. Первоначально *MatLab* предназначался для

проектирования систем управления (основная специальность Джона Литтла), но вскоре завоевал популярность во многих других научных и инженерных областях. Уже тогда *MatLab* также широко использовался в сфере образования для преподавания линейной алгебры и численных методов.

Язык *MatLab* является высокоуровневым интерпретируемым языком программирования, который включает основанные на матрицах структуры данных, широкий спектр функций, интегрированную среду разработки, объектно-ориентированные возможности и интерфейсы к программам, написанным на других языках программирования [1].

Программы, созданные с помощью пакета *MatLab*, бывают двух типов – скрипты и функции. Функции имеют входные и выходные аргументы, собственное рабочее пространство для хранения промежуточных результатов вычислений и переменных. Скрипты используют общее рабочее пространство. Как и скрипты, так и функции не компилируются в машинный код и сохраняются в виде текстовых файлов. Существует также возможность сохранять *pre-parsed* программы – функции и скрипты, обработанные в вид, удобный для машинного исполнения. В общем случае такие программы выполняются гораздо быстрее обычных, особенно если функция содержит команды построения графиков.

Несмотря на то, что изначально *MatLab* предназначалась исключительно для вычислений, в дополнение к усовершенствованным вычислительным средствам у фирмы *Waterloo Maple* по лицензии для *MatLab* было приобретено ядро символьных преобразований, а также появились библиотеки, позволившие *MatLab* расширить области применения и обеспечившие уникальные для математических пакетов функции.

Для *MatLab* имеется возможность создавать специальные наборы инструментов, расширяющих его функциональность. Наборы инструментов представляют собой коллекции функций, написанных на языке *MatLab* для решения специализированных задач. Компания *Mathworks* поставляет наборы инструментов, которые используются во многих областях [2]:

- цифровая обработка сигналов, изображений и данных: *DSP Toolbox*, *Image Processing Toolbox*, *Wavelet Toolbox*, *Communication Toolbox*, *Filter Design Toolbox* – наборы функций, позволяющих решать широкий спектр задач обработки сигналов, изображений, проектирования цифровых фильтров и систем связи;

- системы управления: *Control Systems Toolbox*,  *$\mu$ -Analysis and Synthesis Toolbox*, *Robust Control Toolbox*, *System Identification Toolbox*, *LMI Control Toolbox*, *Model Predictive Control Toolbox*, *Model-Based Calibration Toolbox* – наборы функций, облегчающих анализ и синтез динамических систем, проектирование,

моделирование и идентификацию систем управления, включая современные алгоритмы управления, такие как робастное управление,  $H_\infty$ -управление, ЛМН-синтез,  $\mu$ -синтез;

- финансовый анализ: *GARCH Toolbox*, *Fixed-Income Toolbox*, *Financial Time Series Toolbox*, *Financial Derivatives Toolbox*, *Financial Toolbox*, *Datafeed Toolbox* – наборы функций, позволяющие быстро и эффективно собирать, обрабатывать и передавать различную финансовую информацию;

- анализ и синтез географических карт, включая трёхмерные: *Mapping Toolbox*;

- сбор и анализ экспериментальных данных: *Data Acquisition Toolbox*, *Image Acquisition Toolbox*, *Instrument Control Toolbox*, *Link for Code Composer Studio* – наборы функций, позволяющих сохранять и обрабатывать данные, полученные в ходе экспериментов, в том числе в реальном времени. Поддерживается интеграция широкого спектра научного и инженерного измерительного оборудования;

- набор инструментов для визуализации и представления данных: *Virtual Reality Toolbox* – позволяет создавать интерактивные миры и визуализировать научную информацию с помощью технологий виртуальной реальности и языка *VRML*;

- средства разработки: *MatLab Builder for COM*, *MatLab Builder for Excel*, *MatLab Builder for NET*, *MatLab Compiler*, *Filter Design HDL Coder* – наборы функций, которые позволяют создавать независимые приложения из среды *MatLab*;

- взаимодействие с внешним программным обеспечением: *MatLab Report Generator*, *Excel Link*, *Database Toolbox*, *MatLab Web Server*, *Link for ModelSim* – это наборы функций, позволяющие сохранять данные в различных видах таким образом, чтобы другие программы могли с ними работать;

- инструменты работы с базами данных: *Database Toolbox*;

- научные и математические пакеты: *Bioinformatics Toolbox*, *Curve Fitting Toolbox*, *Fixed-Point Toolbox*, *Fuzzy Logic Toolbox*, *Genetic Algorithm and Direct Search Toolbox*, *OPC Toolbox*, *Optimization Toolbox*, *Partial Differential Equation Toolbox*, *Spline Toolbox*, *Statistic Toolbox*, *RF Toolbox* – наборы специализированных математических функций, которые позволяют решать широкий спектр научных и инженерных задач, включающих разработку генетических алгоритмов, решения задач в частных производных, целочисленные проблемы, оптимизацию систем и другие;

- инструменты для синтеза и анализ нейронных сетей: *Neural Network Toolbox*;

- инструменты для построения и анализа нечётких множеств: *Fuzzy Logic Toolbox*;

- библиотека символьных вычислений: *Symbolic Math Toolbox* – инструменты для

символьных вычислений с возможностью взаимодействия с символьным процессором программы *Maple*.

Перечислим совместимые с *MatLab* программы на уровне языка программирования

- *GNU Octave*;
- *FreeMat*;
- *Maxima*;
- *Scilab*;
- Таблицы *Excel*;
- *Maple*.

В состав *MatLab* включается библиотека визуального программирования *Simulink*, что позволяет построить логическую схему системы управления, используя только стандартные блоки. После конструирования схемы можно детально проанализировать ее работу. Сигналы, значения параметров и атрибуты схемы указываются либо непосредственно на самой схеме, либо введены из текстовой консоли. Использование готовых моделей позволяет управлять набором данных и быстро перепрофилировать модели. *MatLab* работает с различными типами данных: целыми, вещественными, с фиксированной запятой, матрицами, векторами, комплексными, а так же работает с пользовательскими типами данных.

*MatLab* предоставляет широкие возможности для программирования. Библиотека *C Math* (компилятор *MatLab*) является объектной и имеет свыше 300 процедур обработки данных на языке Си. При работе с этим программным продуктом можно использовать как процедуры самой *MatLab*, так и стандартные процедуры языка Си, используя компилятор *C Math*, можно встраивать любые процедуры *MatLab* в готовые приложения [1].

Библиотеки *MatLab* отличаются высокой скоростью численных вычислений. Матрицы являются основой автоматического решения и составлений уравнений состояния динамических объектов и систем. *MatLab* один из наиболее мощных универсальных интегрированных пакетов компьютерной математики [1].

Компилятор пакета *MatLab* позволяет из *m*-функций создавать:

- автономные приложения, выполняются если *MatLab* не установлен на компьютере;
- *C/C++* библиотеки совместного использования, использующиеся без установленного пакета *MatLab* на системе конечного пользователя.

Компилятор поддерживает большое количество команд и пакетов *MatLab*, однако, не все. Полный список ограничений можно найти на сайте производителя *MatLab*. Компиляция кодов в *MatLab* требует хорошей подготовки программиста и глубокого знания принципов компиляции.

Компилятор *MatLab* требует наличия хотя бы одного из следующих *C/C++* компиляторов:

- *Lcc C*, является только *C*-компилятором;
- *Borland C++* версии 5.3, 5.4, 5.5, 5.6 [3];

- *Microsoft Visual C/C++* версии 6.0, 7.0, 7.1.

Компилятор *MatLab* версии 4, в отличие от своих предшественников, генерирует только код интерфейса (оболочки), т.е. не транслирует *m*-функции в двоичный или *C/C++* код, но при этом создает специальный файл по технологии *Component Technology File (CTF)*, который содержит объединения различных пакетов, необходимых для обеспечения работы *m*-функции. Попутно компилятор *MatLab* зашифровывает данный файл уникальным (неповторяющимся) 1024-битным ключом.

*MatLab* – инженерный пакет высокого уровня, тщательно отлаженный, широко используемый во многих сферах научных исследований, инженерных расчетах и проектировании систем автоматического управления. Основным недостатком этой программы является стоимость лицензии: набор платформы плюс библиотека *Simulink* обойдется коммерческой организации примерно в \$7000. Можно отметить, что часто при использовании этой программы приходится работать с большим количеством окон, с которыми удобнее работать на двух мониторах (рис. 2).

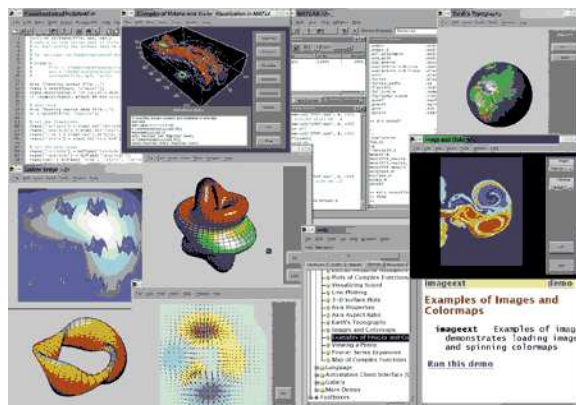


Рис. 2. Рабочий экран пользователя *MatLab*

Программа имеет хорошую справочную систему, меню *help*, объем фирменной документации достигает почти 5 тысяч страниц, однако, это делает ее трудно обозримой.

Пакет является проприетарным (коммерческим) продуктом и не имеет открытого программного кода. Так даже скомпилированные пользовательские программы зашифрованы уникальным (неповторяющимся) 1024-битным ключом, таким образом, эти скомпилированные приложения на языке Си не могут редактироваться;

Система *MatLab* не имеет развитых средств создания пользовательского интерфейса, какими обладают системы визуального программирования *Microsoft Visual C++*, *Borland C++Builder* и др., поэтому создаваемое системой *MatLab* автономное приложение не отвечает требованиям профессионального программирования.



Система *MatLab* используется в технике, широко используется в науке и образовании. Предоставляет возможности для анализа данных, охватывающие практически все области математики (матрицы и линейная алгебра, многочлены и интерполяция, математическая статистика и анализ, обработка данных, дифференциальные уравнения, разреженные матрицы, целочисленная арифметика). А так же система *MatLab* применяется для разработки алгоритмов, предоставляя удобные средства для разработки алгоритмов, включающие высокоуровневые с использованием концепций объектно-ориентированного программирования, имеются необходимые средства интегрированной среды разработки, включая отладчик и профилиер. Имеются функции для работы с целыми типами данных, которые облегчают создание алгоритмов для микроконтроллеров и других приложений, где это необходимо.

Этот инженерный пакет предоставляет огромные возможности визуализация данных. В составе пакета имеется достаточное количество функций для построения графиков, в том числе трёхмерных, визуального анализа данных и создания анимированных роликов.

Встроенная среда разработки предоставляет возможность создавать пользовательские графические интерфейсы с различными элементами управления, такими как кнопки, поля ввода и другими. С помощью компонента *MatLab-Compiler* эти графические интерфейсы могут быть преобразованы в самостоятельные приложения [4].

Программа *MatLab* является одной из самых обширных программ компьютерной математики, вобравшей опыт, методы и правила математических вычислений, накопленные за тысячи лет развития математики. Прилагаемую к системе обширную документацию вполне можно рассматривать как фундаментальный многотомный электронный справочник по математическому обеспечению [1].

### 3. ВОЗМОЖНОСТИ ИСПОЛЬЗОВАНИЯ ПРОГРАММЫ OCTAVE

*Octave* был задуман (в 1988 году) как программное обеспечение для студентов позволяющее изучить принципы химической конструкции реактора Джеймсом Б. Ролингом из Университета Висконсин-Мэдисон и Джоном Г. Экердтом из Университета Техаса. Изначально данный программный продукт разрабатывался как специализированный инструмент для решения проблем химической конструкции реактора. Позже, увидев недостатки такого подхода, авторы решили попытаться создать более гибкий инструмент.

Поскольку студенты больше времени уделяли изучению языка программирования *Fortran*, а не самим принципам машиностроения для химических реакций, было принято решение

модифицировать среду *Octave* как интерактивный инженерный пакет. Таким образом, удалось добиться того, что студенты осваивали данный программный продукт в течение нескольких часов, и большее внимание уделяли непосредственно реализации поставленных задач.

Первый альфа-релиз состоялся 4 января 1993 года, а версия 1.0 была выпущена 17 февраля 1994 года. С тех пор, *Octave* претерпел несколько крупных изменений, на данный момент программа входит в состав *Debian GNU / Linux* и *SuSE Linux* дистрибутивов.

Очевидно, что *Octave* сейчас гораздо больше, чем просто еще один пакет курсов ограниченного использования в пределах высшей школы. Сами авторы о своем проекте говорят следующие: «Хотя наши первоначальные цели были несколько расплывчаты, мы знали, что мы хотели создать нечто, что позволит студентам решать проблемы реалистично, мы создавали продукт для того, чтобы они могли использовать *Octave* для многих других вещей, кроме проблем химической конструкции реактора». Сегодня тысячи людей во всем мире используют *Octave* для преподавания, исследований и коммерческих приложений.

*Octave* представляет интерактивный командный интерфейс (интерпретатор *Octave*), реализованный в ОС *Windows* и *Linux*, более просто аналог *MatLab*.

Математический пакет *GNU Octave* выпускается под лицензией *GNU GPL* (Открытое лицензионное соглашение для свободной Unix-подобной операционной системы). Приложение может работать с различными операционными системами [5]. Переменные и данные представляются в виде матриц. Синтаксис языка программирования и формат команд аналогичен *Scilab*, *MatLab*. Помимо встроенных математических функций существует мощный инструментарий для создания пользовательских функций. У *GNU Octave* существуют различные графические интерфейсы: *Kalculus - MatLab*-подобный интерфейс, *Xoctave*, отдельный плагин для интеграции *GNU Octave* в распространенную графическую среду *NetBeans IDE*.

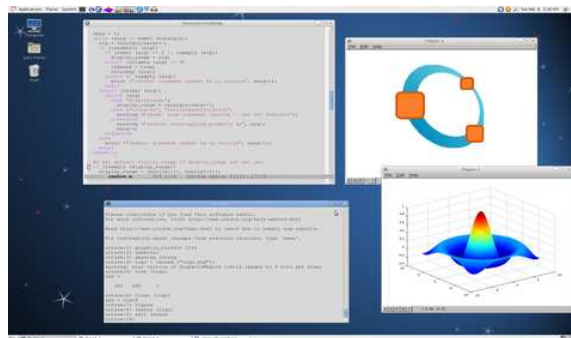


Рис. 3. Интерфейс *Octave*

Программа *KOctave* представляет собой более продвинутый графический интерфейс для



системы *Octave*. При использовании программы *KOctave* интерфейс *Octave* становится практически таким же, как и у *MatLab*.

Синтаксис языка программирования и формат команд аналогичен *Scilab*, *MatLab*, позволят реализовать алгоритм любой сложности.

Функции *Octave* позволяют создавать одно-, двух- и трехмерные графики и поверхности. Для построения графиков и поверхностей в *Octave* используется пакет *gnuplot*, а также этот пакет может быть использован как самостоятельная программа для построения графиков.

*Octave* содержит функции для численного и аналитического решения нелинейных уравнений и систем, а также для интегрирования и дифференцирования [7].

*Octave* имеет мощные и гибкие функции для решения задач оптимизации.

Сложные оптимизационные задачи в *Octave* решают с помощью пакета расширений *Minimization* для *GNU Octave*.

Для решения обыкновенных дифференциальных уравнений и систем в *Octave* существует достаточно много функций, в том числе и жёстких.

Множество функций для решения дифференциальных уравнений находится в пакете расширений *odepkg*. С кратким описанием функций этого пакета на английском языке и примерами можно ознакомиться на странице производителя [6].

*Octave* имеет специальный набор инструментов для анализа и обработки данных эксперимента, задач идентификации, разработки и тестирования математических алгоритмов.

Пакет инженерных вычислений *Octave* предоставляет следующие минимальные требования к аппаратуре:

- процессор *Pentium III, 4, Xeon, Pentium M; AMD Athlon, Athlon XP, Athlon MP*;
- 256 Мбайт оперативной памяти (рекомендуется 512 Мбайт);
- 400 Мбайт дискового пространства;
- операционная система *Microsoft Windows 2000 (SP3)/XP, Linux*;

Программа *Octave* является свободно распространяемым программным продуктом.

*Octave* не имеет встроенных блоков или библиотек для визуального программирования и моделирования, таких как *Simulink (MatLab)* или *XCos (SciLab)*.

Для перевода кода *Octave* в Си-код в этом программном продукте предусмотрен компилятор *Octave Compiler*. Однако, имеется высокое несоответствие между языками, что не всегда позволяет найти однозначное соответствие функциям языка *Octave* и *C, C++*.

Кроме того, если вы пытаетесь получить большой прирост производительности, то компиляция *C*-кода скорее всего не даст этого, потому что код, который будет получен после компиляции, использует все виды динамических и огромных вычислительных функций. На сайте

производителя [6] в разделе Поддержка есть оговорка, что наилучшая производительность достигается, если писать программу изначально на языке *C, C++*.

Сформулируем недостатки *Octave* для его применения в системах управления робототехническим устройством:

- не имеет библиотек визуального моделирования;
- этот программный продукт предназначен для решения математических задач численно, то есть путем расчета значения в памяти компьютера. Это означает, что она не всегда может дать решение с достаточной точностью;
- компилятор *C*-кода работает некорректно.

Пакет *Octave* рекомендуется как инструмент для решения математических задач в курсах «Высшая математика», «Математический анализ», «Линейная алгебра», «Аналитическая геометрия», а также во многих специальных курсах, в которых приходится решать задачи вычислительной математики.

Применяется для разработки и тестирования математических алгоритмов.

Несколько примеров использования этого инженерного пакета. *NASA* с помощью *Octave* разрабатывала систему стыковки кораблей. На базе *Octave Jaguar Racing* разработала систему отображения и анализа данных, передаваемых от болидов Формулы 1, а ученые Шеффилдского университета используют этот математический пакет для разработки программного обеспечения, чтобы распознавать раковые клетки.

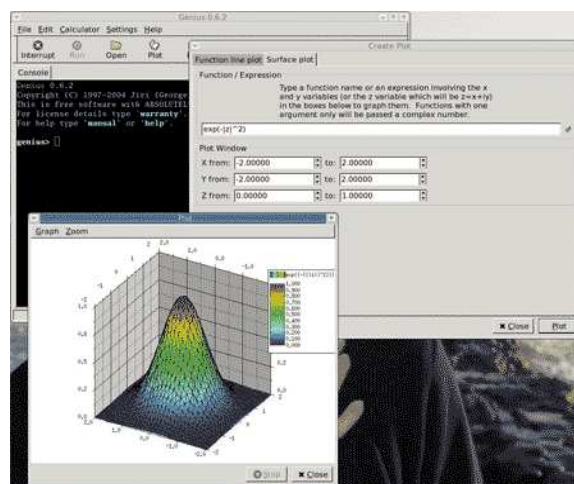


Рис. 4. Построение поверхностей в программе *Octave*

#### 4. ПЕРСПЕКТИВЫ ПРИМЕНЕНИЯ MACSYMA (MASSACHUSETTS COMPUTATION SYMBOLIC ALGEBRA)

Программа *Macysma* – одна из первых математических программ, оперирующих символьной математикой, то есть предназначенных не только для численных, но и для аналитических расчетов.

Проект был инициирован в июле 1968 года Карлом Энгельманом (*Carl Engelman*), Вильямом Мартином (*William A. Martin*) (интерфейс пользователя, отображение выражений, арифметика полиномов) и Джоэлем Мозесом (*Joel Moses*) (механизм упрощения выражений, неопределённые интегралы: эвристики/алгоритм Риша) в Массачусетском технологическом институте (США). Бил Мартин был руководителем проекта до 1971 года, а Мозес следующие десять лет. *Macsyma* была написана на языке программирования *Maclisp*, и являлась, в некоторых случаях, ключевым мотиватором для улучшения этого диалекта Лиспа (*LISP*, *LIST Processing language* — «язык обработки списков»; современное написание: Lisp) в области числовых вычислений, эффективной компиляции, и дизайна самого языка. *Maclisp* работал в основном на компьютерах PDP-6 и PDP-10, а кроме того под операционной системой Multics и на Лисп-машинах. В то время *Macsyma* была одной из самых больших, если не самой большой программой на лиспе.

Программа занимала некоторое время лидирующую позицию среди универсальных математических программ. Благодаря своим сильным сторонам – линейной алгебре и дифференциальным уравнениям - она не потеряла популярности. Свое второе рождение программа *Macsyma* получила в 1992 году, когда возникла компания *Macsyma Inc.*, представившая обновленную, удобную, эффективную программу *Macsyma* и сопутствующую ей программу *PDEase2D* [8].

*Macsyma* – развитой аппарат линейной алгебры и дифференциальных уравнений. Система ориентирована на прикладные расчеты и не предназначена для теоретических исследований в области математики [8].

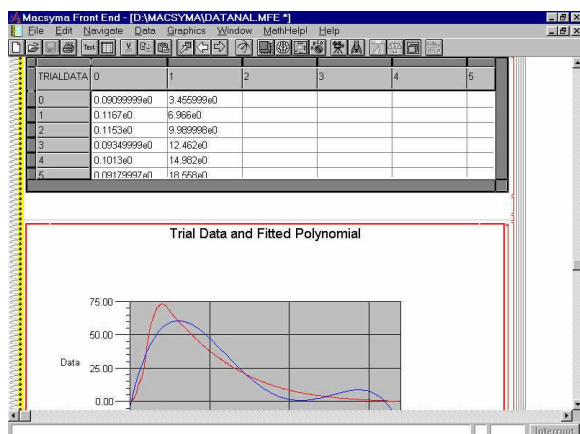


Рис. 5. Интерфейс пользователя в виде рабочей тетради

В связи с этим в программе отсутствуют или сокращены разделы, связанные с теоретическими методами (теория чисел, теория групп, и др.). Числа в математических выражениях по умолчанию предполагаются действительными. Это позволяет получать аналитические решения для многих вычислений, встречающихся в

прикладных задачах, (таких как алгебраические преобразования и упрощения, интегрирование, решение дифференциальных уравнений и так далее), для которых в комплексной области решения не существуют.

Одно из главных преимуществ *Macsyma* перед другими универсальными математическими пакетами является то, что пользователь может аналитически и численно решать большое количество различных типов уравнений в частных производных [9].

Программный продукт, как и другие пакеты символьной математики, имеет средства, как процедурного программирования, так и программирования по заданному правилу. Программа имеет открытую архитектуру, так что большинство команд, хранящихся в командных файлах (с расширением *.mac*) может быть прочитано и изменено пользователем. Пользователь может программировать свои команды, пополняя библиотеку *Macsyma*.

Математический пакет имеет развитой интерфейс, позволяет открывать несколько рабочих документов, внедрять один документ в другой и вводить гипертекстовые связи, как внутри документа, так и между несколькими документами.

Программа имеет довольно хороший текстовый редактор, позволяющий редактировать параграфы, отдельные слова и отдельные символы, изменять тип шрифта.

*Macsyma* работает в интерактивном режиме, команды вводятся из командной строки, и тут же выводится результат.

Почти все команды *Macsyma* в библиотечных файлах загружаются автоматически, что очень удобно, т.к. нет необходимости помнить, какие команды находятся в библиотечных файлах и их имена и загружать эти библиотечные файлы. Имеется удобное окно просмотра (браузер) математических функций.

Программа позволяет строить двух- и трехмерные графики, включая параметрические графики кривых и поверхностей, а также графики векторных полей и анимацию.

Для решения специализированных задач программа имеет специальные пакеты расширений [10]:

- графика - содержит достаточное количество видов графиков для иллюстрации прикладных и научных расчетов, однако число настраиваемых атрибутов графиков в *Macsyma* гораздо больше, чем у конкурентов;
- геометрический пакет - команды этого пакета позволяют вычислять длины кривых, площади и объемы двух-, трех- и многомерных фигур, а так же более сложные геометрические задачи;
- прикладная математика - включает большое количество команд и дополнительные пакеты комбинаторики, теории вероятностей, математической статистики и статистической обработки данных. Позволяет получать

аналитические решения для многих вычислений, встречающихся в прикладных задачах, (таких как алгебраические преобразования и упрощения, интегрирование, решение дифференциальных уравнений и так далее). *Macysma* имеет сопоставимые с *MatLab* средства для преобразования алгебраических выражений, для решения уравнений, неравенств и систем, вычисления интегралов, сумм и произведений;

- векторное исчисление – набор инструментов, позволяющий вычислять и упрощать выражения, содержащие координаты дву- и трехмерных векторов при помощи независимых от координат векторных операторов (скалярное и векторное произведение, градиент, дивергенция, Лапласиан);

- интегральные преобразования – пакет, содержащий функции только для прямого и обратного преобразования Лапласа и Фурье, что достаточно для большинства прикладных задач;

- обыкновенные дифференциальные уравнения - набор функций для решения аналитически линейных и нелинейных дифференциальных уравнений первого и второго порядка и системы линейных дифференциальных уравнений первого порядка. Имеются приближенные методы (разложение в ряд Тейлора и три метода возмущений для решения большого класса обыкновенных дифференциальных уравнений). Численные методы включают классические алгоритмы Рунге-Кутты а также алгоритмы решения жестких дифференциальных уравнений. Программа не имеет средств для аналитического решения дифференциальных уравнений выше второго порядка;

- Интегральные уравнения - позволяет решать интегральные уравнения, программа использует около десятка методов для интегральных уравнений первого и второго рода, с фиксированными и переменными пределами интегрирования.

- обработка данных - набор инструментов содержит функции для анализа, обработки и фильтрации больших массивов данных, а также графического представления и аппроксимации;

- *Numkit* - дополнительный пакет для *Macysma 2.2* существенно увеличивает скорость выполнения операций с действительными и комплексными числами с плавающей запятой, используется для вычислений с матрицами, полиномами, при обработке данных и при решении систем линейных уравнений. Этот пакет распространяется за дополнительную плату и функционирует только на 32-разрядных платформах (*Windows 95* или *NT*);

- пакет *PDEase2D 3.0* – предназначен для численного решения двумерных задач, выраженных системой дифференциальных уравнений в частных производных. Для решения применяется метод конечных элементов. Программа позволяет решать систему до 32

связанных уравнений, не считая граничных условий.

Программа *Macysma* имеет возможность использовать для работы данные программ:

- *MatLab*;
- *Mathematic*;
- *Maple*;

Так как этот программный продукт разрабатывался как своего рода математический калькулятор, он не имеет библиотек для визуального программирования процессов.

*Macysma 2.2* имеет встроенный компилятор - все программы могут быть скомпилированы в двоичный код, включая подпрограммы по заданному правилу.

Генерация *FORTRAN*а и *C*: инженерный пакет генерирует коды *FORTRAN*а и *C*, включая управляющие операторы (циклы итерации, *if-then* ветвления), определения *subroutine* и *function*, описания типов переменных, включая матрицы, сегментацию выражений и возможность задания оптимизации общих частей выражений. Возможно одновременное использования двух языков *FORTRAN* (или *C*) с языком *Macysma* в одном файле [8].

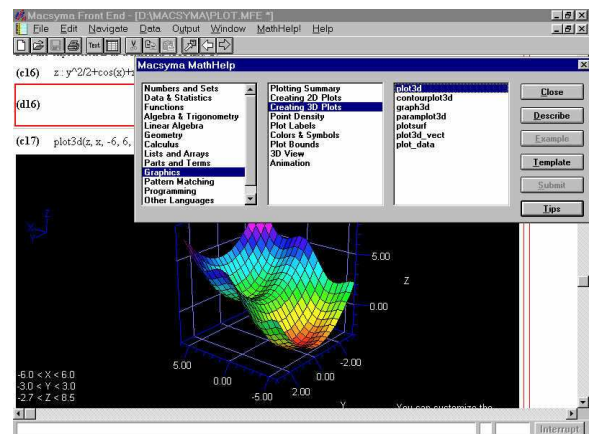


Рис. 6. Интерфейс программы *Macysma*

Сформулируем недостатки проекта *Macysma*: по своему назначению программа является «умным калькулятором» и не может быть использована для анализа процессов, а так же нет библиотек для визуального моделирования.

Инженерный математический программный продукт *Macysma* может быть применим для решения различных задач в учебном процессе средней и высшей школы, а так же для ведения расчетов и визуализации данных в научных работах.

Пакет предъявляет следующие минимальные требования к системе:

- *Macysma 2.2* работает на платформе *Intel* под управлением *Windows* или *Windows for Workgroups 3.1* или *3.11*, *Windows 95* или *Windows NT 3.51* или *4.0*.
- для платформы *UNIX* разработана *UNIX-Macysma*.



- 16-битная версия для 16-битной *Windows* 3.1 и 32-битная версия для 32-битной *Windows* 95 / *NT*.

- Оперативная память: для *Windows* 3.1 – минимум 8 Мбайт основной памяти и 24 Мбайта виртуальной памяти; для *Windows* 95 минимум 12 Мбайт основной памяти и 16 Мбайт виртуальной памяти.

Стоимость лицензия версии *Pro* с сопутствующим пакетом *NumKit* составляет \$349, пакет *PDEase2D* 3.0 стоит \$ 999 долл., *Macsyma* + *PDEase2D* 3.0 - \$ 1319.

## 5. ПЕРСПЕКТИВЫ ИСПОЛЬЗОВАНИЯ ПРОГРАММЫ MATHCAD

*Mathcad* был задуман и первоначально написан Алленом Раздовом из Массачусетского технологического института (MIT), являющегося соучредителем компании *Mathsoft*, которая с 2006 года является частью корпорации *PTC* (*Parametric Technology Corporation*) [11].

*Mathcad* имеет простой и интуитивный для использования интерфейс пользователя. Для ввода формул и данных можно использовать как клавиатуру, так и специальные панели инструментов.

Некоторые из математических возможностей *Mathcad* (версии до 13.1 включительно) основаны на подмножестве системы компьютерной алгебры *Maple* (*MKM*, *Maple Kernel Mathsoft*). Начиная с 14 версии – использует символьное ядро *MuPAD*.

*Mathcad* – программное средство, среда для выполнения на компьютере разнообразных математических и технических расчетов, снабженная простым в освоении и в работе графическим интерфейсом, которая предоставляет пользователю инструменты для работы с формулами, числами, графиками и текстами. В среде *Mathcad* доступны около сотни операторов и логических функций, предназначенных для численного и символьного решения математических задач различной сложности, что позволяет выполнять численные и символьные вычисления, производить операции с скалярными величинами, векторами и матрицами, автоматически переводить одни единицы измерения в другие [12].

Инженерный пакет *Mathcad* имеет следующие возможности:

- решение дифференциальных уравнений, в том числе и численными методами;
- построение двумерных и трёхмерных графиков функций (в разных системах координат, контурных, векторных и т. д.);
- использование греческого алфавита как в уравнениях, так и в тексте;
- выполнение вычислений в символьном режиме;
- выполнение операций с векторами и матрицами;
- символьное решение систем уравнений;

- аппроксимация кривых;
- выполнение подпрограмм;
- поиск корней многочленов и функций;
- проведение статистических расчётов и работа с распределением вероятностей;
- поиск собственных чисел и векторов;
- вычисления с единицами измерения;
- интеграция с САПР системами, использование результатов вычислений в качестве управляющих параметров;

С помощью *Mathcad* инженеры могут документировать все вычисления в процессе их проведения.

Из достоинств программы можно выделить следующие:

- автоматизация процесса – возможно одновременно осуществлять процесс вычислений и их документирование; изменения в вычислениях отображаются в результатах («живые» расчеты);

- удобный интерфейс: уравнения, тексты, графики, исходные и промежуточные данные размещаются на одном рабочем листе;

- интеграция численных и аналитических (символьных) математических инструментов позволяет эффективно вести как обоснование проекта, так и его расчет;

- поддержка интеллектуального автоматического управления единицами физических величин.

Для решения специализированных задач возможности программы расширяются следующими пакетами, имеющими дополнительные функции и константы:

- пакет для анализа данных (англ. *Data Analysis Extension Pack*) – содержит инструменты для анализа данных;

- пакет для обработки сигналов (англ. *Signal Processing Extension Pack*) – содержит около 70 встроенных функций для аналоговой и цифровой обработки сигналов, для анализа и представления результатов в графическом виде;

- пакет для обработки изображений (*Image Processing Extension Pack*) – включает инструменты для обработки изображений, анализа и визуализации;

- пакет для работы с функциями волнового преобразования (англ. *Wavelets Extension Pack*) – содержит набор дополнительных вейвлет-функций, которые можно добавить в библиотеку встроенных функций базового модуля *Mathcad Professional*, а также содержит обширную диалоговую документацию по основным принципам вейвлетов, приложения, примеры и таблицы ссылок;

- библиотека строительства (англ. *Civil Engineering Library*) – включает справочник формул Роарка для расчета напряжений и деформаций, настраиваемые шаблоны для строительного проектирования и примеры тепловых расчётов;

- электротехническая библиотека (англ. *Electrical Engineering Library*) – содержит стандартные вычислительные процедуры, формулы и справочные таблицы, используемые в электротехнике. Текстовые пояснения и примеры облегчают работу с библиотекой — каждый заголовок имеет гиперссылку на оглавление и указатель, и его можно найти в системе поиска;

- библиотека машиностроения (англ. *Mechanical Engineering Library*) – включает справочник формул Роарка для расчета напряжений и деформаций, содержащий пять тысяч формул, вычислительные процедуры из справочника *McGraw-Hil*;

- библиотека электротехнического проектирования *Mathcad* - содержит стандартные расчетные процедуры, формулы и справочные таблицы, используемые инженерами-электриками.

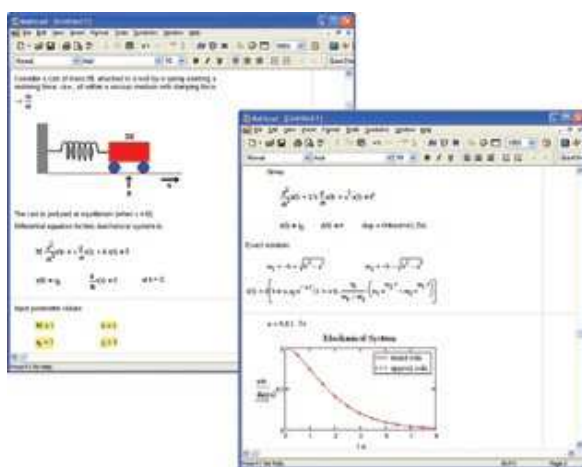


Рис. 7. Интерфейс программы *MathCad*

*MathCad* позволяет работать с данными следующих программ:

- *Microsoft Excel* (и другие приложения *MS Office*);
- *MatLab*;
- *Bentley Microstation*;
- *ANSYS Workbench*;
- результаты вычислений сохраняются в рабочих листах, которые могут использоваться повторно и преобразуются в целый ряд форматов, включая *MS Word*, *PDF*, *HTML* и *XML*.

Так как этот программный продукт разрабатывался как своего рода математический калькулятор, он не имеет библиотек для визуального программирования процессов.

*MathCad* имеет возможность создания исполняемых файлов с расширением \*.dll, а так же компилировать программы в С-код, с помощью встроенного компилятора можно создавать пользовательские функции и добавлять их в палитру стандартных операторов.

При работе с программой выявлены следующие недостатки:

- при работе с графиками: невозможность построения поверхностей, заданных

параметрически, с прямоугольной областью определения двух параметров; создание и форматирование графиков только через меню, что ограничивает возможности программного управления параметрами графики;

- нет библиотек для визуального моделирования;

- реализованы некоторые особые нестандартные функции клавиатуры (например, *Ctrl-P* порождает символ "пи");

- трудности при нумерации осей X и Y на трехмерных графиках

- длинные уравнения автоматически не переносятся на краях страницы.

Сформулируем область применения данного продукта. Данный продукт наиболее удобно использовать для решения задач инженерного характера и создания учебных интерактивных документов, визуализации вычислений.

## 6. ПЕРСПЕКТИВЫ ИСПОЛЬЗОВАНИЯ ПРОГРАММЫ SCILAB

До недавнего времени он разрабатывался исследовательскими институтами *INRIA* и *ENPC* (оба находятся во Франции), а с мая 2003 г. поддержку продукта взял на себя специально созданный для этой цели *Scilab Consortium* [13].

С 1994 года распространяется вместе с исходным кодом через Интернет. В 2003 году для поддержки *Scilab* был создан консорциум *Scilab Consortium*. Сейчас в него входят 25 участников, в том числе *Mandriva*, *INRIA* и *ENPC* (Франция).

Хотя *Scilab* является бесплатным продуктом с открытым исходным кодом, его вычислительные возможности, обеспеченные приблизительно тысячей встроенных функций, вполне соответствуют системам компьютерной математики (СКМ) профессионального уровня.

Является некоммерческим аналогом инженерного пакета *MatLab*. Имеет схожий с *MatLab* интерфейс и принципы взаимодействия с пользователем через командную строку до синтаксиса языка. Таким образом, *Scilab* можно рассматривать как облегченный вариант *MatLab*, который сохраняет основные возможности [13].

*Scilab* является типичным командным интерпретатором и структурно состоит из интерпретирующей системы, принимающей команды пользователя и возвращающей результаты, и двух библиотек: собственных функций и дополнительных – на языках *C* и *Fortran*.

Пакет поддерживает основные элементарные и множество специальных функций, применяемых в математике, в том числе – для различного вида сглаживаний и аппроксимаций, эллиптические интегралы, функции Бесселя. *Scilab* содержит также мощный набор средств для работы с полиномами – как обычными, так и матричными. Например, имеются операторы для создания полинома с заданными корнями или



коэффициентами, вычисления корней полинома (до сотой степени), деления двух полиномов, нахождения наибольшего общего делителя и наименьшего общего кратного нескольких полиномов и выполнения десятков других важных операций над полиномами.

Особенностью пакета является то, что он предназначен почти исключительно для реализации численных методов и по умолчанию оперирует с любыми значениями как с числами с плавающей точкой.

Графика *Scilab* выполнена на вполне профессиональном уровне. Основой ее реализации является концепция устройства, в которое выводится поток, создаваемый графическими функциями, – это может быть экран или внешние файлы в форматах *PostScript*, *Xfig*, *GIF* или *PMM* [13].

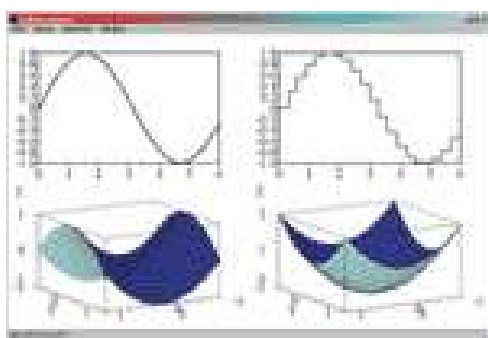


Рис. 8. Построение графиков в программе *SciLab*

*Scilab* строит одномерные, дву- и трехмерные поверхности, а так же предусмотрена широкая настройка свойств.

*Scilab* содержит полноценную справочную систему, в дополнение к которой имеется полный справочник по функциям в *HTML*-формате (на английском и французском языке). А так же в поставку входит коллекция примеров, состоящая из двух частей: первая демонстрирует способы использования основных функций пакета и доступна через меню *Demos*, вторая иллюстрирует различные программно-интерфейсные возможности (файлы находятся в отдельной папке).

Хотя *Scilab* является типичным интерпретатором, его стандартные средства включают некоторые интерактивные возможности, делающие процесс общения с системой более простым и наглядным. Так, есть функции для ввода матриц вручную и выполнения файловых операций, создания простейших диалоговых окон и даже изменения меню основного командного окна [13].

*SciLab* обеспечивает широкий набор дополнительных инструментов. Приведем в качестве примера некоторые из них.

- *RTAI-Lab* – набор инструментов, позволяющий моделировать системы в режиме

реального времени; требует модифицированного ядра операционной системы *Linux*, проект *RTAI*;

- *Lapack*, *LINPACK*, *ODEPACK*, *Atlas* – библиотеки для численных расчетов;

- *the Robotics Toolbox for Scilab/Scicos* – набор инструментов, содержащий множество полезных функций в области робототехники для изучения таких областей, как кинематики, динамики и траектории поколения, предназначен для программирования и моделирования роботов манипуляторов, а также для анализа результатов, полученных в экспериментах с реальными роботами;

- *Mavsim Scicoslab Toolbox* - средства моделирования для управления, навигации и разработки алгоритмов для микро летательных аппаратов;

- *Matlab-like plotting library* – набор инструментов для построения графиков как в *MatLab*;

- *.NET component and COM server* – инструменты для *NET* компонента и *COM* сервера;

- *Assert* – Коллекция корректировочных функций для поддержки численных модульных тестов;

- *Mathieu functions* – Инструменты для работы с функциями Матье;

- *Documentation: Floating Point Numbers* – инструменты для вычислений с плавающей запятой;

- имеются узкоспециализированные пакеты для цифровой обработки сигналов, анализа динамических систем, оптимизации со специальными ограничениями.

Это далеко не весь список инструментов, который пополняется не только от официальных разработчиков, но и от обычных пользователей.

Инженерный пакет *SciLab* работает с данными, полученными из программ:

- *Maple*;
- *MatLab*;
- *LabVIEW*;

В состав *Scilab* входит модуль *Scicos* - система компьютерного моделирования, аналогичная *Simulink* – инструмент для редактирования блочных диаграмм и симуляции. Рассмотрим возможности программирования, предоставляемые пакетом *SciLab*

Инженерный пакет позволяет создавать пользовательские объекты, функции, типы данных. Для этих целей *Scilab* располагает мощным встроенным языком, обладающим широким набором конструкций для организации циклов, условных переходов, операций ввода/вывода. Естественно, что с помощью этого языка можно получить доступ ко всем внутренним возможностям приложения.

В данной СКМ реализована исключительно гибкая концепция процедурного программирования – процедурам (называемым в *Scilab*

функциями) разрешается возвращать целые массивы результатов произвольного типа, а список аргументов даже не обязан быть фиксированным. Процедуры можно объединить в библиотеки и сохранять в скомпилированном бинарном формате (расширение \*.bin) – именно так хранятся стандартные функции этой СКМ.

Scilab имеет полноценные средства отладки, которые дают возможность устанавливать контрольные точки, получать информацию о переменных по ходу выполнения процедуры [13].

Scilab поддерживает взаимодействие с кодом на языках C и Fortran.

Есть возможность компиляции программ в двоичный код, компиляции в Си-код, создание исполняемых файлов.

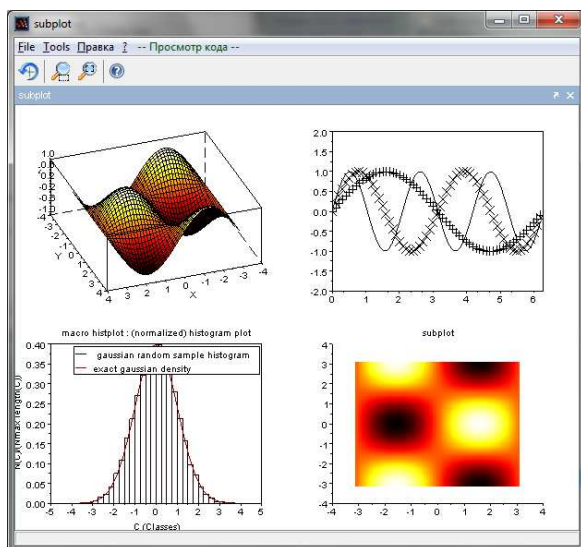


Рис. 9. Визуализация данных в SciLab

Обобщим недостатки, присущие пакету SciLab

- Интерфейс не очень удобен для пользователя;
- слабая поддержка аналитических операций и использование ограниченной разрядной сетки при вычислениях;
- количество символьно-аналитических операции ограничено;
- отладчик Scilab не интерактивен – контрольные точки необходимо разместить с помощью специальных операторов в теле программы, а управление процессом происходит через командную строку;

• если диаграмма, созданная пользователем в приложении SciCos, имеет логические ошибки, при запуске процессов программа изредка может зависнуть или закрыться, не сохранив изменения.

Таким образом, можно сделать вывод, что пакет SciLab позволяет создавать недорогие, но вполне конкурентоспособные решения в области обработки данных, численной реализации алгоритмов и визуализации. Прекрасно подходит для ведения научных расчетов и анализа

полученных данных. Огромное количество дополнительных наборов инструментов, созданных пользователями, расширяет, как сферы применения (физика, энергетика, химия), так возможности программы (узкоспециализированные функции, алгоритмы, и т.д.). Scicos используется для обработки сигналов систем автоматического управления, систем массового обслуживания, а также для изучения физических и биологических систем.

## 7. ПЕРСПЕКТИВЫ ИСПОЛЬЗОВАНИЯ ПРОГРАММЫ LABVIEW

Сформулируем требования к системе, которые необходимо обеспечить для комфортной работы в LabVIEW.

Минимальные требования

- Процессор *Pentium III/Celeron* 866 МГц или аналогичный *AMD*
- Оперативная память 256 MB
- Монитор с разрешением: 1024 x 768
- ОС: *Windows 7/Vista/XP/Windows Server 2003 R2 /Windows Server 2008 R2*
- Место на жестком диске 1.6 GB (включая драйверы)

Рекомендуемые:

- Процессор *Pentium 4* или аналогичный *AMD*
- Оперативная память 1 Гб
- Монитор с разрешением: 1024 x 768
- Операционная система *Windows 7/Vista/XP/Windows Server 2003 R2 / Windows Server 2008 R2*
- Место на диске: 1.6 GB (включая драйверы)

LabVIEW это коммерческий продукт. На официальном сайте разработчика стоимость лицензии для коммерческих организаций *LabVIEW Full Development System* составляет \$4000, стоимость лицензии для университета составляет порядка \$ 15 000.

Программа LabVIEW появилась на рынке математического программного обеспечения в 1986 г. Разработчиком является известная во всем мире американская компания *National Instruments*. Пакет быстро завоевал популярность среди инженеров и ученых, которые использовали этот продукт на разных стадиях проектирования, создания комплексных приложений в задачах измерения, тестирования, управления, автоматизации научного эксперимента и образования.

LabVIEW (*Laboratory Virtual Instrument Engineering Workbench*) – это среда графического программирования, выпущенная для технических специалистов, инженеров, преподавателей и ученых, позволяет разрабатывать прикладное программное обеспечение для организации взаимодействия с измерительной и управляющей аппаратурой, сбора, обработки и отображения информации и результатов расчетов, а также моделирования как отдельных объектов, так и

автоматизированных систем в целом.

В *LabVIEW* программы создаются в виде графических диаграмм, подобных обычным блок-схемам. Иногда можно создать приложение, вообще не прикасаясь к клавиатуре компьютера.

Система программирования *LabVIEW* является открытой и имеет встроенную поддержку применяемых в настоящее время программных интерфейсов, таких как *Win32 DLL*, *COM*, *.NET*, *DDE*, сетевых протоколов на базе *IP*, *DataSocket* и др. В состав *LabVIEW* входят библиотеки управления различными аппаратными средствами и интерфейсами, такими как *PCI*, *CompactPCI/PXI*, *VME*, *VXI*, *GPIB* (КОП), *PLC*, *VISA*, системами технического зрения и др. Программные продукты, созданные с использованием *LabVIEW*, могут быть дополнены фрагментами, разработанными на *C/C++*, *Pascal*, *Basic*, *FORTRAN*. А так же можно использовать модули, разработанные в *LabVIEW*, в проектах, создаваемых в других системах программирования. Таким образом, *LabVIEW* позволяет разрабатывать приложения, взаимодействующие со многими видами аппаратных средств, поддерживаемых операционной системой компьютера.

Программа позволяет создать разработчику панель виртуальных приборов, что превращает персональный компьютер и набор произвольного контрольно-измерительного оборудования в многофункциональный измерительно-вычислительный комплекс.

Несомненным достоинством *LabVIEW* является то, что разработчику и пользователю доступны функционально идентичные системы программирования для различных операционных систем, таких как *Microsoft Windows 95/98/NT/2000/XP*, *Linux*, *MacOS*. Например, программа разработанная под *Windows* будет почти без изменений работать на компьютере с *Linux*.

*LabVIEW* позволяет работать с широким спектром оборудования различных производителей, функциональность расширяется благодаря встроенным, либо добавленным к базовому пакету, многочисленным библиотекам компонентов.

- набор инструментов для подключения внешнего оборудования по интерфейсам и протоколам *RS-232*, *GPIB-488*, *TCP/IP*;
- библиотеки удалённого управления экспериментом;
- набор инструментов управления роботами и системами машинного зрения;
- набор инструментов для генерации сигналов, цифровой обработки ;
- набор инструментов различных математических методов обработки данных;
- набор инструментов визуализации данных, обработки (включая *3D*-модели);
- моделирования сложных систем;
- базы данных для хранения информации и инструменты для генерации отчетов;

- набор инструментов для взаимодействия с приложениями в рамках концепции *COM/DCOM/OLE* [14].

Для выполнения на тех компьютерах, на которых не установлена полная версия программы существует специальный компонент *LabVIEW Application Builder*. Для работы таких программ требуется бесплатно распространяемый компонент «*LabVIEW Runtime Engine*» и, при необходимости, драйверы используемых внешних устройств.

Пакет может работать с программами и данными, созданными в программах:

- *MatLab*;
- *SciCos*;
- *MathCad*.

Программа *LabVIEW* является средой графического программирования, т.е. работа с пакетом сводится к созданию и настройке блок-диаграмм.

Приложения и программы, разработанные при помощи данного программного продукта, могут быть скомпилированы в бинарный исполняемый код.

Инженерный пакет поддерживает импорт, редактирование сценариев *\*.m* файлов, традиционно используемые в области математического моделирования и анализа, обработки сигналов и сложных математических вычислений с помощью модуля *LabVIEW MathScript RT*. Позволяет использовать их совместно с кодом на языке *G* и создавать самостоятельные приложения для настольных ПК или оборудования, работающего в режиме реального времени.

Отметим недостатки системы *LabVIEW*.

Программа *LabVIEW* является хорошо проработанной системой визуального программирования, однако, можно отметить следующие недостатки:

- сложность в освоении. Разработчики программы для получения базовых знаний предлагают пройти специальный обучающий курс;
- стоимость лицензии для учебного заведения с дополнительными модулями составляет около \$ 15000.

Пакет первоначально был разработан для использования в исследовательских лабораториях, и на данный момент является наиболее популярным программным пакетом для проведения исследований в лабораториях фундаментальной науки (например, *Lawrence Livermore*, *Argonne*, *Batelle*, *Sandia*, *Jet Propulsion Laboratory*, *White Sands* и *Oak Ridge* в США, *CERN* в Европе), а так же в отраслевых промышленных лабораториях.

*LabVIEW* начинают использовать в сфере образования, в университетских лабораторных практикумах – особенно по предметам электротехники, механики и физики.

Пакет *LabVIEW* получил широкое распространение: применяется на борту

космических аппаратов, на подводных лодках, от буровых установок в Северном море до промышленных предприятий в Новой Зеландии. В связи с ростом возможностей Internet сфера применения *LabVIEW* стала расширяться не только в географическом, но и в виртуальном пространстве (*cyberspace*). Все большее число разработчиков создает виртуальные приборы, допускающие удаленное управление и наблюдение через *Internet*. Измерительные системы на основе виртуальных приборов отличаются своей многофункциональностью, гибкостью и низкой стоимостью как с точки зрения оборудования, так и с точки зрения затрат времени на разработку.

В настоящее время *LabVIEW* широко применяется в следующих сферах:

- автомобильная промышленность;
- телекоммуникации;
- аэрокосмическая промышленность;
- полупроводниковая промышленность;
- разработка и производство электроники;

- управление технологическими процессам;

- биомедицина.

Благодаря своей гибкости и масштабируемости, *LabVIEW* может использоваться на всех этапах технологического процесса: от моделирования и разработки прототипов продуктов до широкомасштабных производственных испытаний.

## 8. ВЫБОР ПРОГРАММЫ ДЛЯ РАЗРАБОТКИ БЛОКА УПРАВЛЕНИЯ

На основе проведенного обзора математического ПО составлена сравнительная таблица характеристик рассмотренных программ.

Для решения поставленных задач был выбран пакет *SciLab*. По данным таблицы 1. видно, что этот продукт является наиболее подходящим по системным требованиям и возможностям. Пакет *SciLab* имеет открытый исходный код, выпускается для операционных систем *Windows*, *Linux*, *Mac OS*.

Таблица 1.

### Сравнение характеристик рассмотренных программ

Наименование	<i>MatLab</i>	<i>Octave</i>	<i>Macysma</i>	<i>MathCad</i>	<i>SciLab</i>	<i>LabVIEW</i>
Удовлетворяет ли системным требованиям одноплатного компьютера	превышает требуемое объем памяти на диске	да	да	превышает требуемое объем памяти на диске	да	Превышает требования к процессору, рекомендуемой ОЗУ, объем памяти на диске
Стоимость лицензии	свыше \$7000 включая Simulink	бесплатное ПО	от \$1 000	от \$ 800	Бесплатное ПО	от \$ 4 000
Наличие библиотек визуального моделирования	да	нет	нет	нет	да	да
Открытость программного кода	нет	да	нет	нет	да	нет
Возможность создавать исполняемые файлы	да	да	да	нет	да	да

Программа *SciLab* выпускается под лицензией *GPL* (англ. *General Public License* – Открытое лицензионное соглашение).

Преимуществом этой программы является бесплатность, невысокие системные требования. По работе с этой программой имеется достаточное количество литературы, в сети Интернет можно найти большое количество форумов и сообществ пользователей.

Можно сделать вывод, что ПО *SciLab* является

более подходящим для реализации поставленных в рамках данной работы задач.

Пакет *SciLab* использовался для создания системы управления в режиме реального времени вакуумным переключателем типа *EX-BB 6-20/1000 УЗ-1* в выпускной работе «Имитационное моделирование системы управления с использованием пакета *SciLab*». Система управления разработана с использованием библиотеки визуального моделирования *SciCos*.



Модель была скомпилирована в бинарный исполняемый файл, который использовался для управления включением и отключением вакуумного переключателя. Тестирование системы управления на реальном объекте прошло успешно: система обеспечивала требуемые временные характеристики переключения.

## ЗАКЛЮЧЕНИЕ

Результаты, полученные в рамках данной работы, показывают, что современные свободные математические и инженерные пакеты могут быть успешно использованы для создания систем управления в робототехнике. Наиболее перспективным является применение свободно распространяемого открытого пакета *SciLab*, который позволяет проводить как моделирование, так и выгружать СИ код модели, открытый к модификации и компиляции. Исходный код хорошо анализируется, обладает интуитивно понятной структурой и легко модифицируется. В данной работе продемонстрирован пример создания аппаратно-программного комплекса управления автономным гусеничным шасси с использованием одноплатного компьютера и встраиваемой операционной системы Линукс. Полученные результаты позволяют рекомендовать данный метод создания систем управления к использованию при синтезе методов и проектировании сложных систем автоматизации и робототехники.

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

Авторы благодарят за постановку задачи и общее руководство работами д.т.н. В.А. Жмудя.

## ЛИТЕРАТУРА

- [1] Компьютер-пресс: науч. электрон. журн. URL: <http://www.compress.ru/Archive/CP/2006/7/18/#MatLab>
- [2] Официальный сайт разработчика MatLab. URL: [www.mathworks.com](http://www.mathworks.com)
- [3] Статьи по автоматическому трейдингу и оптимизации стратегий: электрон. журн. URL: <http://www.mql5.com/ru/articles/44>
- [4] Тюменский государственный университет Информационная служба: URL: <http://info.tsogu.ru/support/download/matlab>
- [5] Свободные математические программы: URL: <http://openmath.ucoz.ru/>
- [6] Официальный сайт разработчика Octave: URL: <http://octave.sourceforge.net/odepkg/overview.html>
- [7] The 8th International conference of developers and users of free / open source software. URL: [http://summer.lvee.org/en/reports/LVEE\\_2011\\_23](http://summer.lvee.org/en/reports/LVEE_2011_23)
- [8] Экспонента.ру: науч. электрон. журн. URL: <http://www.exponenta.ru/soft/Others/macsyma/macsyma.asp>
- [9] Компьютер-пресс: науч. электрон. журн. URL: <http://www.compress.ru/article.aspx?id=12530&iid=474>
- [10] Мир ПК [Электронный ресурс]: URL: <http://www.osp.ru/pcworld/1999/04/160253/>
- [11] Экспонента.ру: науч. журн. – URL: <http://www.exponenta.ru/soft/Mathcad/Mathcad.asp>
- [12] Средства автоматизации: науч. электрон. журн. – URL: <http://mathlab2011.h12.ru/index.php?p=2>
- [13] IT для бизнеса: URL: [http://ko.com.ua/scilab\\_sereznaya\\_matematika\\_dostupnaya\\_vsem\\_1\\_6168](http://ko.com.ua/scilab_sereznaya_matematika_dostupnaya_vsem_1_6168)
- [14] Официальный сайт разработчика LabVIEW: URL: <http://www.labview.ru/>

*Колкер Алексей Борисович* – доцент кафедры Автоматики НГТУ, к.т.н.,

**e-mail:** [fiery77@yandex.ru](mailto:fiery77@yandex.ru)

*Ливенец Дмитрий Александрович* – магистр, окончил кафедру Автоматики НГТУ,

**e-mail:** [dmitry.livenets@gmail.com](mailto:dmitry.livenets@gmail.com)

*Кошелева Алёна Игоревна* – магистр, окончила кафедру Автоматики НГТУ,

**e-mail:** [kosheleva.alyona@mail.ru](mailto:kosheleva.alyona@mail.ru)



# Разработка системы управления стенда для исследования переходных процессов вакуумного высоковольтного выключателя

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

**Аннотация:** Описаны результаты разработки системы управления учебным стендом для исследования переходных процессов в вакуумном высоковольтном переключателе типа *EX-BB 6-20/1000 УЗ-1*.

**Ключевые слова:** Программное обеспечение, вакуумный переключатель.

## 1. ВВЕДЕНИЕ

В работе исследуется возможность использования пакета *SciLab* в режиме моделирования и компиляции исполняемого модуля на примере разработки стенда для исследования переходных процессов вакуумного высоковольтного переключателя. Требуется разработать систему автоматического управления в режиме реального времени с использованием программного пакета *SciLab*, оценить параметры быстродействия [1]. Система реального времени (СРВ) представляет собой аппаратно-программный комплекс, реагирующий в предсказуемые времена на непредсказуемый поток внешних событий: система должна успеть отреагировать на событие, произошедшее на объекте, система должна успевать реагировать на одновременно происходящие события. Даже если два или больше внешних событий происходят одновременно, система должна успеть среагировать на каждое из них в течение интервалов времени, критического для этих событий (*meet deadline*).

Любая система реального масштаба времени может быть описана как состоящая из трех основных подсистем:

- объект управления диктует требования в реальном масштабе времени;
- подсистема контроля и управления обеспечивает выполнение требуемых вычислений и связь с оборудованием (представлена процессорами, управляющими местными ресурсами: память и устройства хранения, доступ к локальной сети в реальном масштабе времени);
- подсистема оператора контролирует полную деятельность системы.

## 2. РАЗРАБОТКА МОДЕЛИ УПРАВЛЕНИЯ

Разрабатываемая модель управления состоит из объекта управления, согласующего устройства, подсистемы контроля и управления. Объектом управления является вакуумный выключатель

типа *EX-BB 6-20/1000 УЗ-1*.

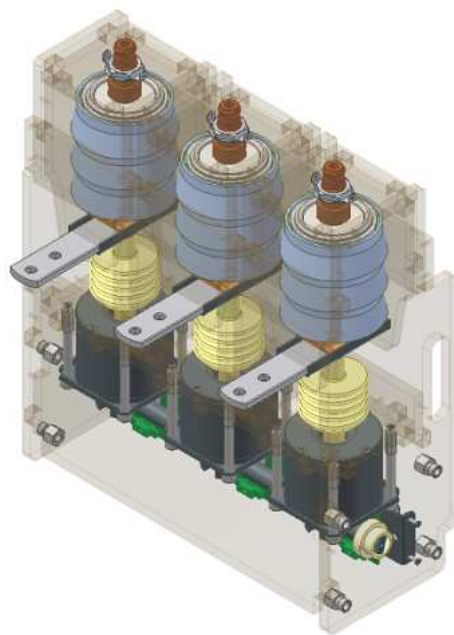


Рис. 1 Вакуумный переключатель типа *EX-BB 6-20/1000 УЗ-1*

Вакуумный выключатель типа *EX-BB (BB)* предназначен для коммутации электрических цепей в промышленных и сетевых установках, в сетях трехфазного переменного тока с изолированной или заземленной через дугогасительный реактор нейтралью частотой 50 и 60 Гц с номинальным напряжением до 10 кВ, а также для электрических сетей общего назначения. Полное время отключения не более 40 мс, собственное время отключения не более 20 мс, собственное время включения 100 мс. Поэтому разрабатываемая система реального времени с применением такого переключателя жесткая.

Для исследования переходных процессов и свойств объекта управления создается стенд, состоящий из подсистемы управления и контроля в режиме реального времени, основанной на пакете *SciLab* [2–5].

В *Scicos* имеются блоки трех типов: базовые постоянные блоки (*Regular Basic Blocks*), базовые блоки пересечения нуля (*Zero Crossing Basic Blocks*) и блоки синхронизации (*Synchro Basic Blocks*). Эти блоки могут иметь два типа входов и выходов: постоянные и активационные. Постоянные выходы соединяются связями с постоянными входами, активационные

выходы соединяются с активационными входами.

*RTAI-Lab* представляет собой палитру блоч-схем, которые могут быть скомпилированы и запущены в режиме реального времени под операционной системой *Linux*. В данном проекте используется операционная система *Ubuntu 9.10*, поддерживающая режим реального времени.

При использовании *RTAI-Lab* системы управляющие сигналы передаются на исполнительный механизм через порт цифрового ввода-вывода, допускающий работу в режиме реального. В качестве такого порта может выступать специализированная плата или *LPT (Line Print Terminal)* порт. Порт используется для подключения периферийных устройств, внешних устройств хранения данных, для организации связи между двумя компьютерами, подключения каких-либо механизмов телесигнализации и телеуправления.

Применение виртуального осциллографа и мониторинга позволяет взаимодействовать с исполнительным механизмом в режиме реального времени.

Требуется установить, можно ли реализовать систему мягкого реального времени используя в качестве устройства ввода-вывода звуковую карту персонального компьютера.

Модель представляет собой систему управления, которая запускает одиночный импульс, когда входной сигнал достигает заданного уровня на указанном такте.

Входной сигнал представляет собой синусоидальное напряжение, частотой 50 Гц, амплитуда 5 В.

### 3. РАЗРАБОТКА СТРУКТУРНОЙ СХЕМЫ МОДЕЛИ

Требуемый уровень входного сигнала и номер такта задаются константами *Уровень* и *Такт*. Из входного сигнала на сумматоре вычитается константа *Уровень*, таким образом, когда входной сигнал достигает искомого уровня, на выходе блока *Сумматор* появится ноль. Выход сумматора подается на вход компаратора, сравнивающего значения с нулем. При смене знака, с положительного на отрицательный на инкрементный счетчик тактов подается импульс. Выход счетчика сравнивается с константой *Такт*. Когда на выходе компаратора появится единица, запускается формирователь одиночного импульса (ФОИ). Структурная схема системы изображена на *рис. 2*.

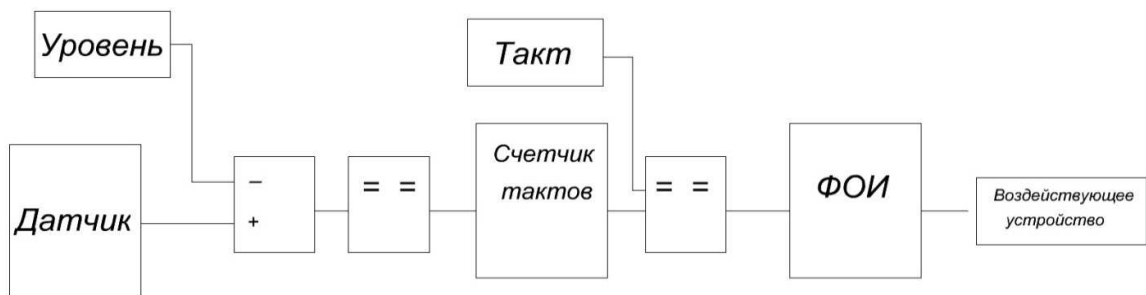


Рис.2. Структурная схема модели

Для считывания информации, поступающей от внешних устройств, в системе *SciLab* используется блок *Sensor* (датчик входа). Необходимо обеспечить считывание информации со звуковой карты и представление сигнала в виде, который может обработать программа.

Принято решение модифицировать блок *SENSOR* (датчик входа), принадлежащий палитре блоков *RTAI-lib*.

Параметры блока, задаваемые пользователем:

- количество выходов;
- открытый программный код – возможность редактирования для решения конкретных задач.

В качестве компаратора, регистрирующего пересечение нуля, использован блок *Edge Trigger*, принадлежащий палитре *Branching*.

Параметры блока. Пользователь указывает, на каком событии выдавать импульс: пересечение нуля при переходе входного сигнала из отрицательной области в положительную, из

положительной в отрицательную, или любое.

Счетчик тактов представляет собой встроенный блок *Counter* принадлежащий палитре *Sources*.

Параметры блока, задаваемые пользователем:

- максимальное и минимальное значения счетчика;
- тип счетчика: инкрементный или декрементный.

Блок *Relational* (палитра *Others*) играет роль компаратора.

Параметры блока:

- режим компаратора: на выходе компаратора появится единица либо при равенстве двух сигналов на входе, либо если сигналы приблизительно равны, либо первый строго меньше второго, меньше или равен, строго больше, больше либо равен

- использовать пересечение нуля;
- тип данных на выходе блока.

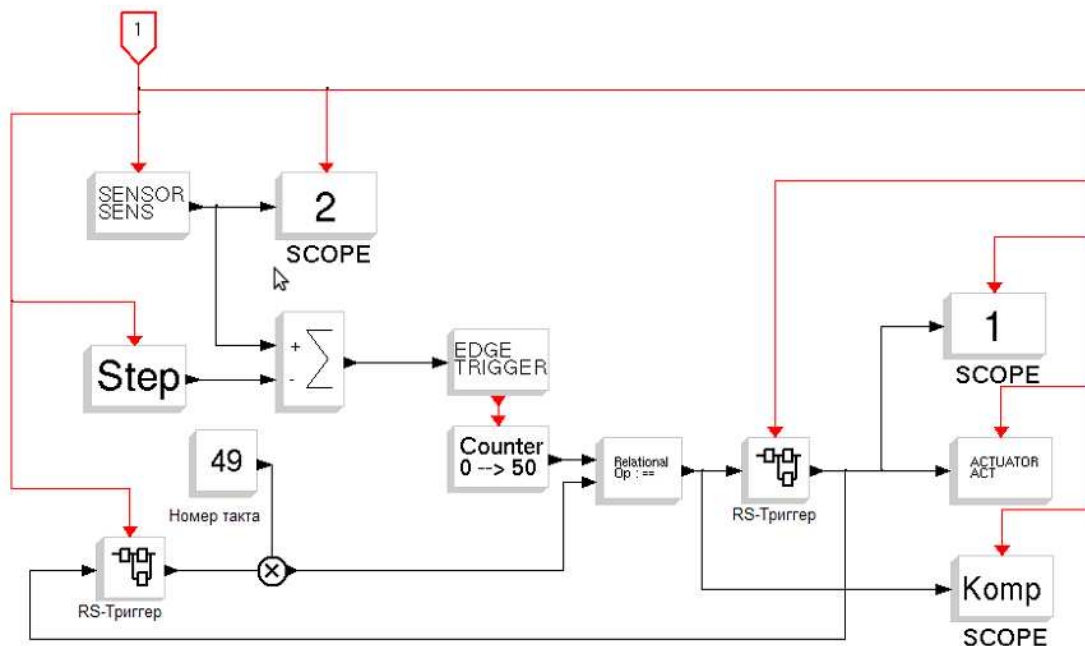


Рис. 3. Параметры блока Sensor

#### 4. РАЗРАБОТКА ФОРМИРОВАТЕЛЯ ОДИНОЧНОГО ИМПУЛЬСА

ФОИ состоит из двух RS-триггеров.

Таблица 2

Таблица состояний RS-триггера

$S$	$R$	$Q(t)$	$!Q(t)$
0	0	$Q(t-1)$	$!Q(t-1)$
0	1	0	1
1	0	1	0
1	1	0	1

Пришедшая с выхода компаратора единица поступает на вход  $S$  триггера и одновременно поступает на вход блока задержки, который в

свою очередь последовательно соединен с входом  $R$ . Таким образом, формируется прямоугольный импульс, длительность которого равна длительности времени задержки. Длительность времени задержки регулируется блоком *Freq\_div* – делитель частоты, в качестве параметра передается коэффициент деления.

С выхода RS-триггера с задержкой, равной шагу дискретизации, единица поступает на вход  $S$  триггера «защелки». На инверсном выходе триггера «защелки» появится единица, умножается на константу номер такта и сравнивается на компараторе с данными счетчика тактов. Таким образом, обеспечивается единственность импульса.

Программные блоки системы *SciLab* принадлежат разным палитрам, поэтому необходимо обеспечить соответствие типов выходов и входов. Соответствие размерности портов блоков обеспечивается блоком *Convert to*.

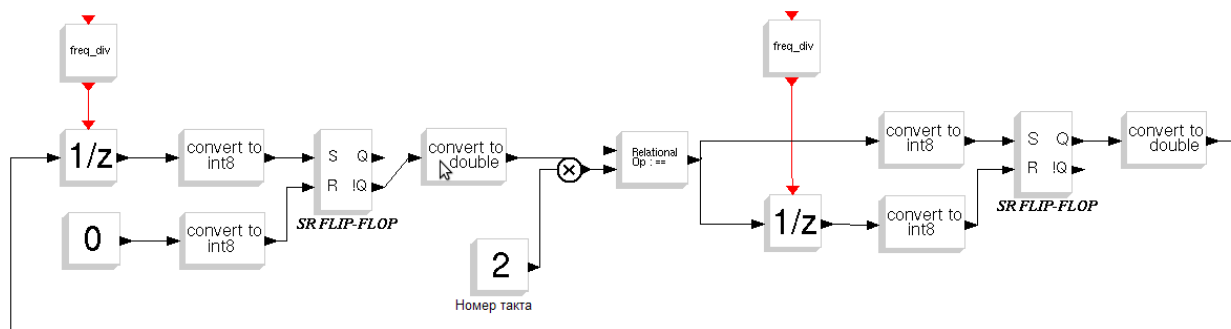


Рис. 4. Схема формирователя одиночного импульса

Выход формирователя одиночно импульса подается на блок *Actuator* – воздействующее устройство. Данный блок обеспечивает подачу

управляющего сигнала на выход LPT порта компьютера.

Блоки *Sensor* и *Actuator* имеют четыре

состояния:

- инициализация – на всех пинах LPT порта устанавливаются нулевой уровень;
- при включении на всех пинах устанавливается уровень единицы
- вход системы приходит 0 – все биты LPT обнуляются, кроме программируемых;
- при завершении работы программы – все биты LPT обнуляются.

Параметры блока, задаваемые пользователем:

- количество входов.

Блок *Actuator* имеет открытый программный код на языке Си.

## 5. ОТЛАДКА И КОМПИЛИЦИЯ МОДЕЛИ

Таким образом, модель системы управления примет вид (см. рис. 6).

Модель системы объединяется в подсистему «*Super Block*» для дальнейшей компиляции.

Встроенная команда *RTAI*-системы *RTAI CodeGen* создает исполняемый файл, который представляет собой программу на языке Си. Система *RTAI-Lab* имеет виртуальный осциллограф, который позволяет наблюдать

результат выполнения созданной системы управления.

Для тестирования используется генератор импульсов Г5-54. Синхронизация выполняется по переднему фронту генератора.

Шаг дискретизации системы выбран 1 мкс. Длительность импульса генератора 1мс. В результате эксперимента на экране осциллографа получили эпюры, показаны на рис. 7.

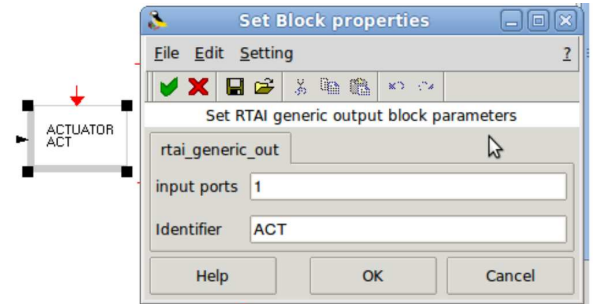


Рис. 5. Параметры блока *Actuator*

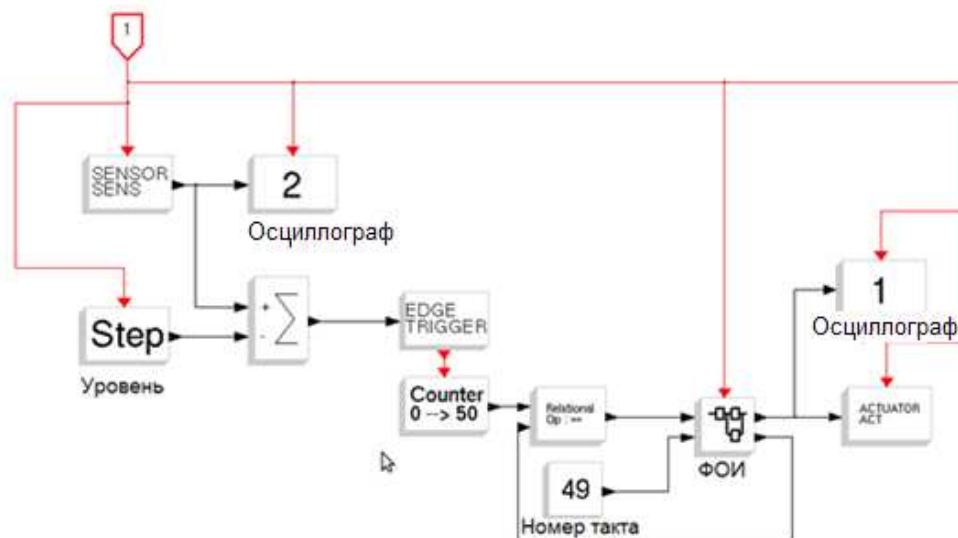


Рис. 6. Функциональная схема модели управления

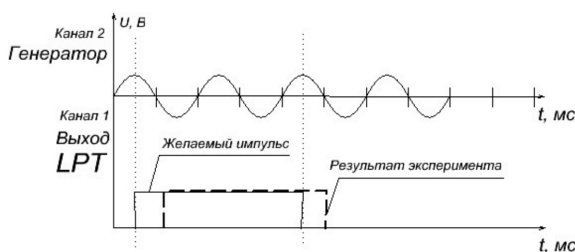


Рис. 7. Результаты тестирования

В результате моделирования можно сделать следующие выводы.

Функциональные блоки программы, не имеющие активационные входы, при создании моделей реального времени использовать не рекомендуется, т.к. они вносят погрешность

превосходящую шаг дискретизации системы.

Использование базового блока делителя частоты *Freq\_div* вносит существенную временную ошибку в формирование импульса: момент формирования переднего фронта импульса отклоняется от заданного значения на величину, превышающую величину шага дискретизации, длительность импульса изменяется больше, чем на величину шага дискретизации.

Исходя из вышесказанного, звуковую карту в качестве системы ввода-вывода в режиме реального времени использовать нельзя, так как это устройство имеет внутреннюю буферную память, способом заполнения которой невозможно управлять. Принято решение, использовать *LPT* порт как систему ввода-вывода

– как аналог портов `gpio`. Таким образом, необходимо разработать модель управления, которая формирует одиночный импульс длительностью 40 мс с задержкой 5 мс при переходе входного сигнала через уровень ноля, обеспечить запуск и останов при нажатии соответствующих кнопок.

При включении системы поступают прямоугольные импульсы на датчик. По нажатию кнопки Пуск формируется одиночный импульс. При нажатии кнопки Сброс на выходе воздействующего устройства нулевой сигнал – сброс напряжения удержания.

Структурная схема модели представлена на рис. 8.



Рис. 8. Структурная схема модели

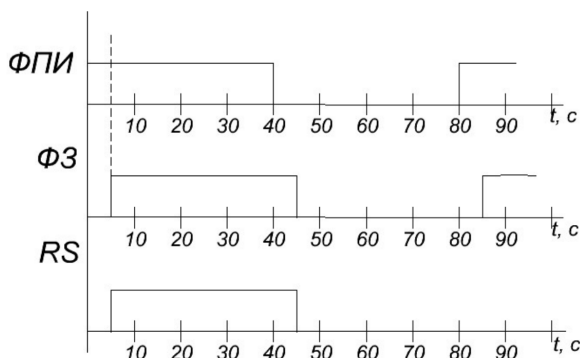


Рис. 9. Временная диаграмма ФОИ

Формирователь прямоугольных импульсов (ФПИ) формирует прямоугольные сигналы заданной длительности. Сигналы поступают на формирователь задержки (ФЗ) и сдвигаются во времени на заданную величину. С выхода ФЗ сигнал поступает на вход «S» RS-триггера.

В результате требуется получить следующую временную диаграмму (см. рис. 9).

На выходе формирователя одиночного импульса появляется импульс заданной длительности и с заданной задержкой.

Недопустимо одновременного нажатия кнопок Пуск и Сброс, а также недопустимо, чтобы на входы R и S триггера одновременно подавались ноли. 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. Логика управления может быть реализована следующим образом (см. рис. 10).

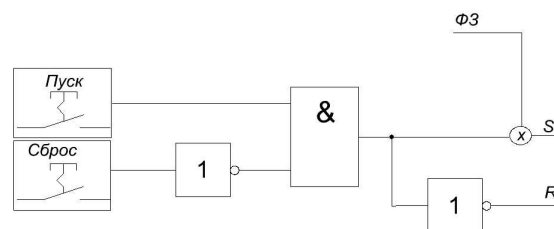


Рис.10. Структурная схема логики управления

Для считывания информации, поступающей от внешних устройств (входной сигнал, нажатие кнопок Пуск и Сброс), выбран блок `Sensor` (датчик входа). Необходимо обеспечить считывание информации с порта `LTP` и представление сигнала в виде, который может обработать программа.

Принято решение модифицировать стандартный блок `SENSOR` (датчик входа), принадлежащий палитре блоков `RTAI-lib`.

Выход формирователя одиночного импульса подается на блок `Actuator` – воздействующее устройство. Данный блок обеспечивает подачу управляющего сигнала на выход `LPT` порта компьютера.

В качестве формирователя прямоугольных импульсов выбран блок `RTAI-square` из палитры блоков `RTAI`.

Параметры блока, устанавливаемые пользователем:

- амплитуда импульса;
- период, с;
- длительность импульса, с;
- наклон, градус;
- время задержки, с.

Установив длительность импульса 40 мс, система сформирует импульс желаемой длительности. Выход блока последовательно соединен с ФЗ, который в свою очередь передает сигнал на установочный вход S RS-триггера. На рис. 11 приведена функциональная схема ФОИ.



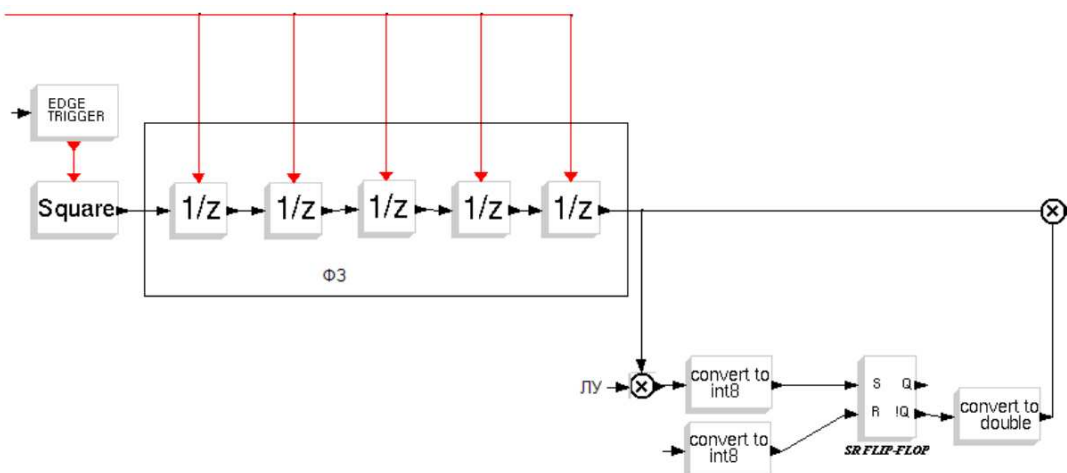


Рис. 11. Функциональная схема ФДИ

Тестирование блока делителя частоты *Freq\_div* показало, что этот блок вносит существенную ошибку. По этой причине использовать схему формирователя одиночного импульса, предложенную в разделе 2, не рекомендуется. Принято решение использовать блок *Delay operator* (Оператор задержки), который осуществляет задержку на величину одного шага дискретизации. Так как шаг дискретизации системы выбран 1 мс, потребуется последовательно соединить пять блоков *Delay operator*.

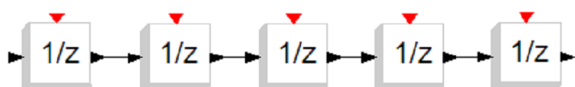


Рис. 12. Формирователь задержки

Сигналы с кнопок принимаются системой через блоки *Sensor*. Для реализации двух кнопок используются два блока *Sensor*.



Рис. 13. Блок *IFTHEL\_f* и *ANDBLK*

В качестве логических элементов «И» и

инверторов используются логические блоки пакета из библиотеки *Events IFTHEL\_f* и *ANDBLK* соответственно.

Блок *IFTHEL\_f* имеет один активационный вход для тактирования от общего генератора, вход последовательный, а так же два активационных выхода *Then* и *Else*. В случае если условие выполняется, на выходе *Then* устанавливается уровень логической единицы, а на выходе *Else* – логического нуля. В случае не выполнения условия – наоборот.

Блок *ANDBLK* имеет два активационных входа и один выход.

Таблица 5

Таблица значений входов и выхода блока *ANDBLK*

Вход 1	Вход 2	Выход
0	0	0
0	1	0
1	0	0
1	1	1

Таким образом, пуск системы происходит при включенной кнопке Пуск и отжатой кнопке Сброс.

Функциональная модель блока управления изображена на рис. 14.



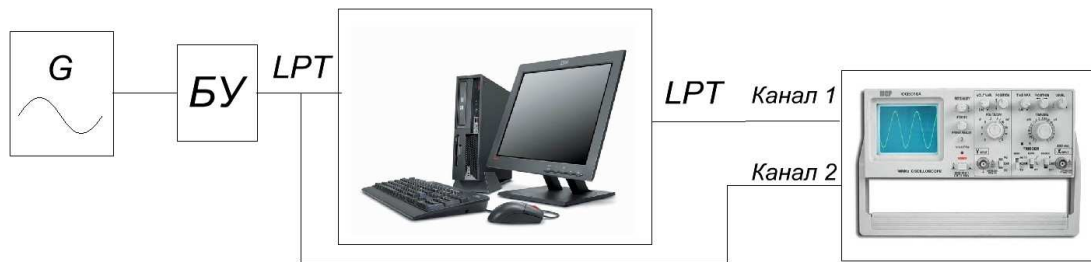


Рис. 16. Схема для тестирования

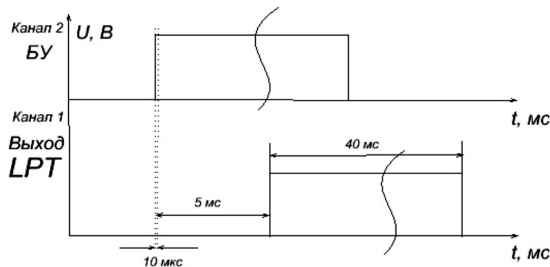


Рис.17. Результаты эксперимента

### ЗАКЛЮЧЕНИЕ

В работе продемонстрирована возможность использования пакета *SciLab* в режиме моделирования и компиляции исполняемого модуля на примере разработки стенда для исследования переходных процессов вакуумного высоковольтного переключателя. Результаты будут использоваться для развития робототехнических стендов кафедры Автоматики НГТУ.

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

Авторы благодарят за постановку задачи и

общее руководство работами д.т.н. В.А. Жмудя.

### ЛИТЕРАТУРА

- [1] 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.iciworld.com/files/file\\_pool/0B329000195966240272/file/WAFER-945GSE2\\_UMN\\_v2.00.pdf](http://www.iciworld.com/files/file_pool/0B329000195966240272/file/WAFER-945GSE2_UMN_v2.00.pdf)
- [2] RTAI - Real Time Application Interface Official Website: URL: <https://www.rtai.org/>
- [3] Scicos: Block diagram modeler/simulator: URL: <http://scicos.org/>
- [4] The Free Platform for Numerical Computation: URL: <http://www.scilab.org/support/documentation>
- [5] *SciLab* refence manual. Scilab Group INRIA Meta2 Project/ENPC Cergrene. - Domaine de Voluceau – Rocquencourt: - BP 105 - 78153 Le Chesnay Cedex (France). 2001 – 700 с.

**Колкер Алексей Борисович** – доцент кафедры Автоматики НГТУ, к.т.н.,

**e-mail:** [fiery77@yandex.ru](mailto:fiery77@yandex.ru)

**Ливенец Дмитрий Александрович** – магистр, окончил кафедру Автоматики НГТУ,

**e-mail:** [dmitry.livenets@gmail.com](mailto:dmitry.livenets@gmail.com)

**Кошелева Алёна Игоревна** – магистр, окончила кафедру Автоматики НГТУ,

**e-mail:** [kosheleva.alyona@mail.ru](mailto:kosheleva.alyona@mail.ru)

# 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

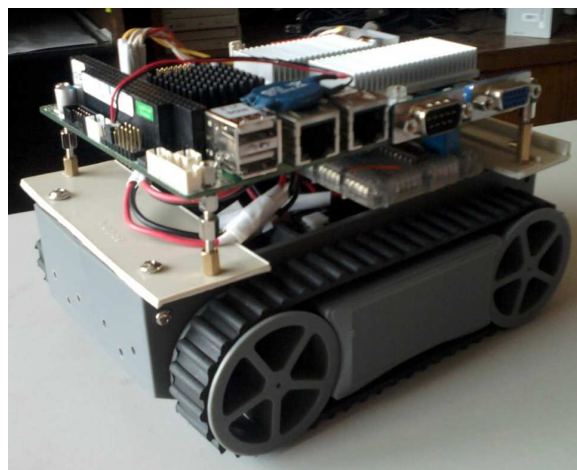


Рис. 2. Объект управления

## 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.

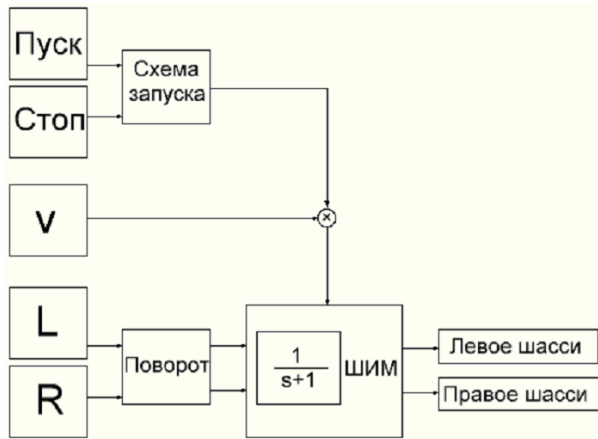


Рис. 3. Структурная схема общая

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.

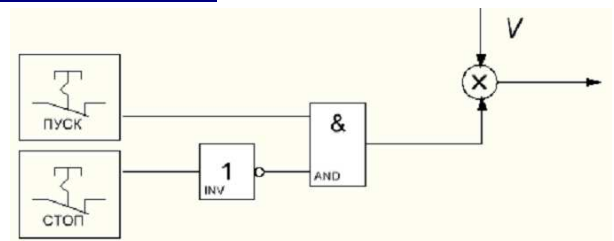


Рис. 4. Схема запуска

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.

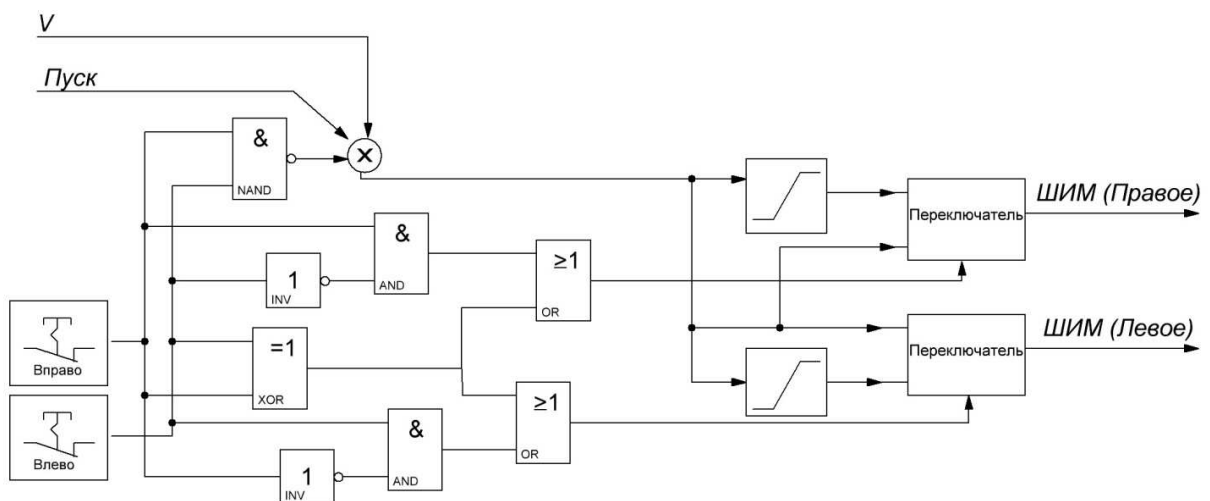


Рис. 5 . Схема переключения направления

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.

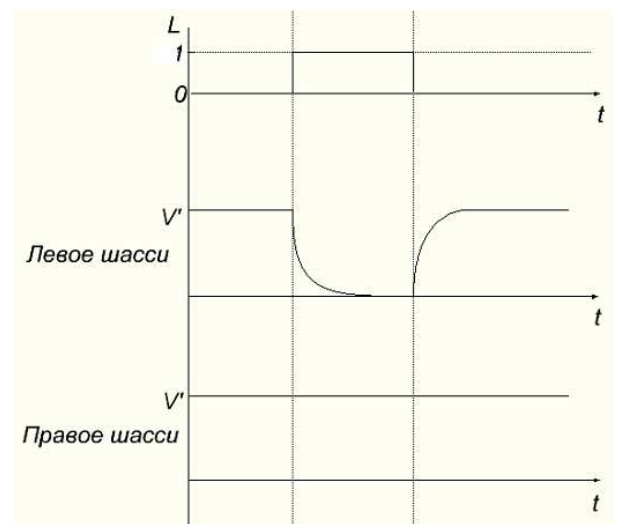


Рис. 6. Временная диаграмма шасси при



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».

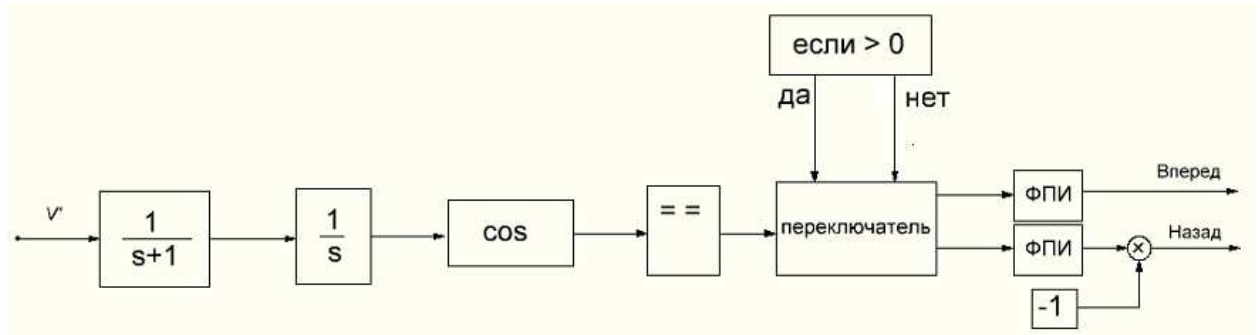


Рис. 7. Широтно-импульсный модулятор

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

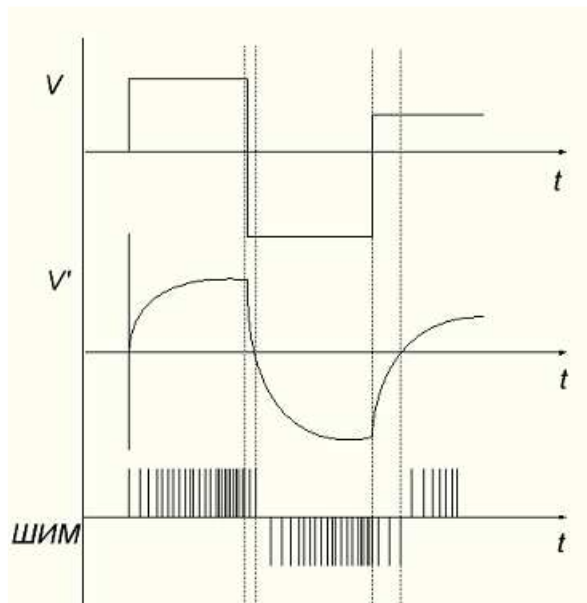


Рис. 8. Временная диаграмма блока управления

#### 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).

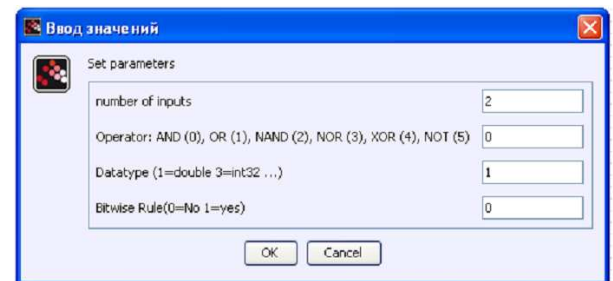


Рис. 9. Блок *Logical op*

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*.

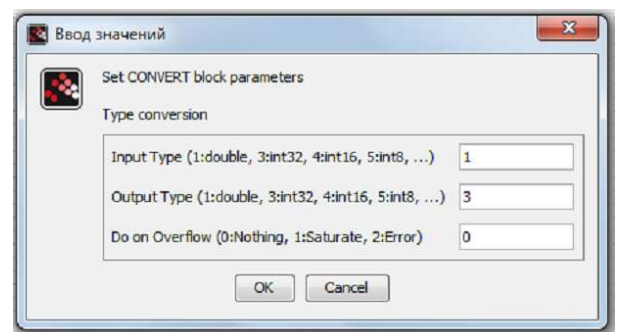


Рис.10. Блок *Convert to*

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.*

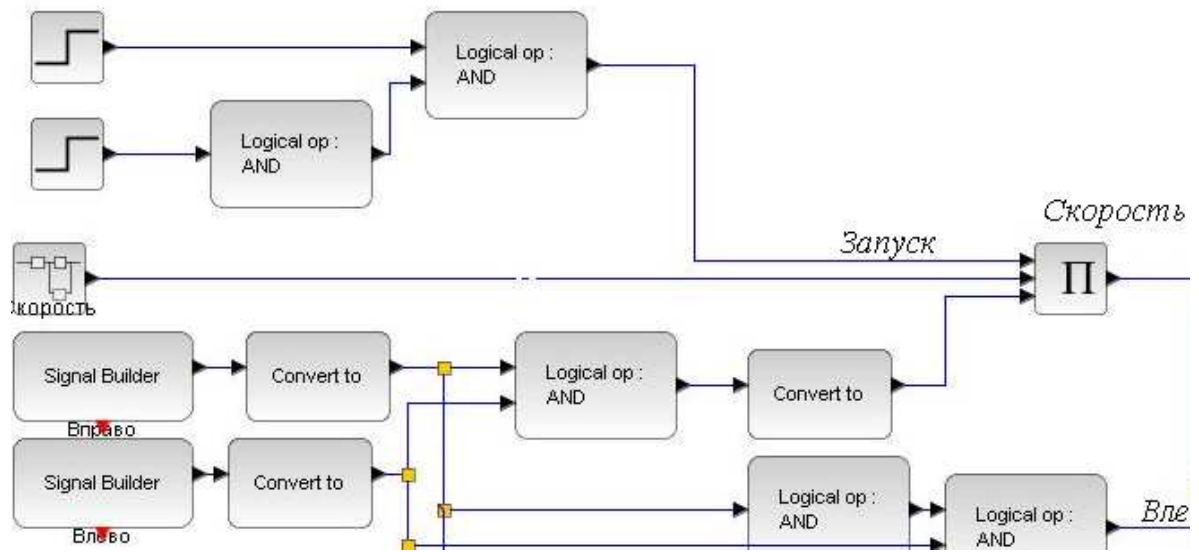


Рис. 11. Схема запуска

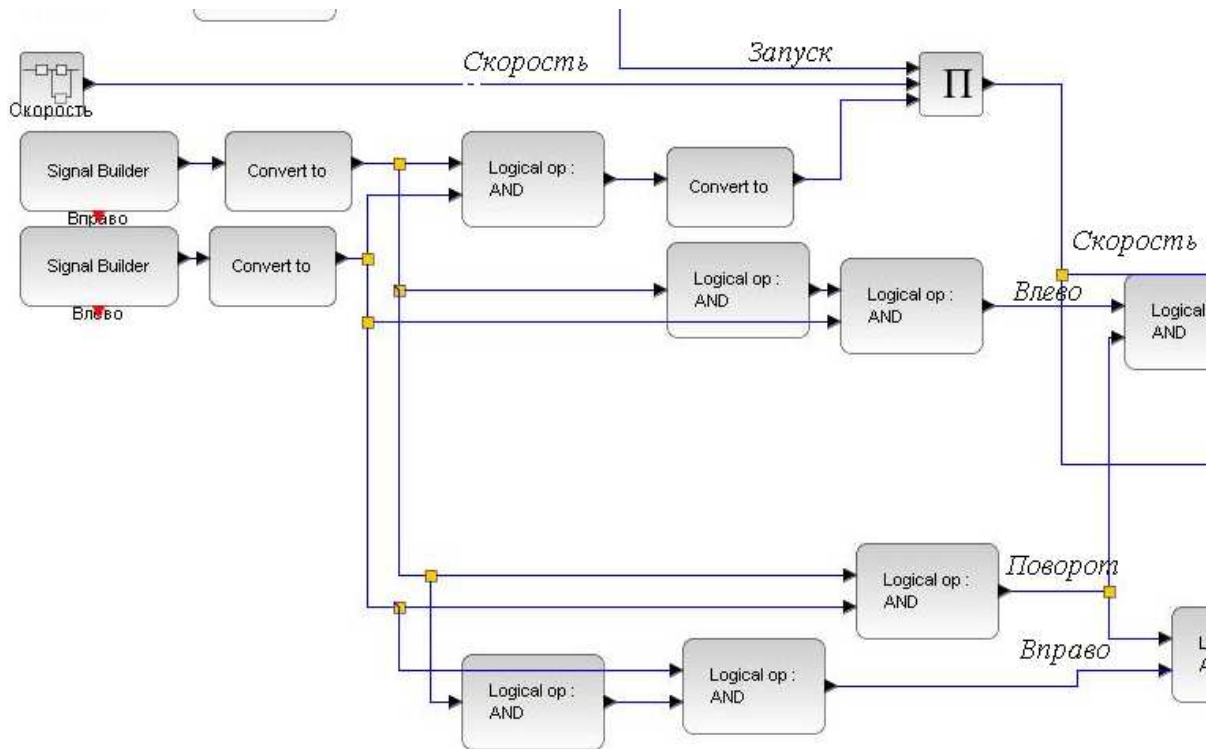


Рис.12. Схема управления поворотом

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.

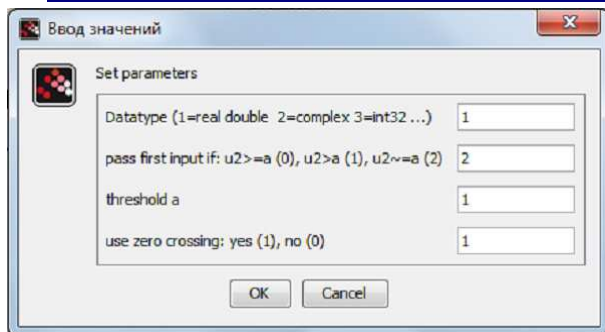


Рис. 13. Блок *Switch*

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.

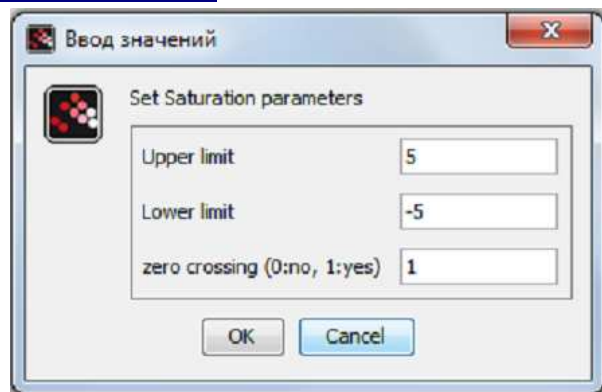


Рис.14. Блок *Saturation*

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.

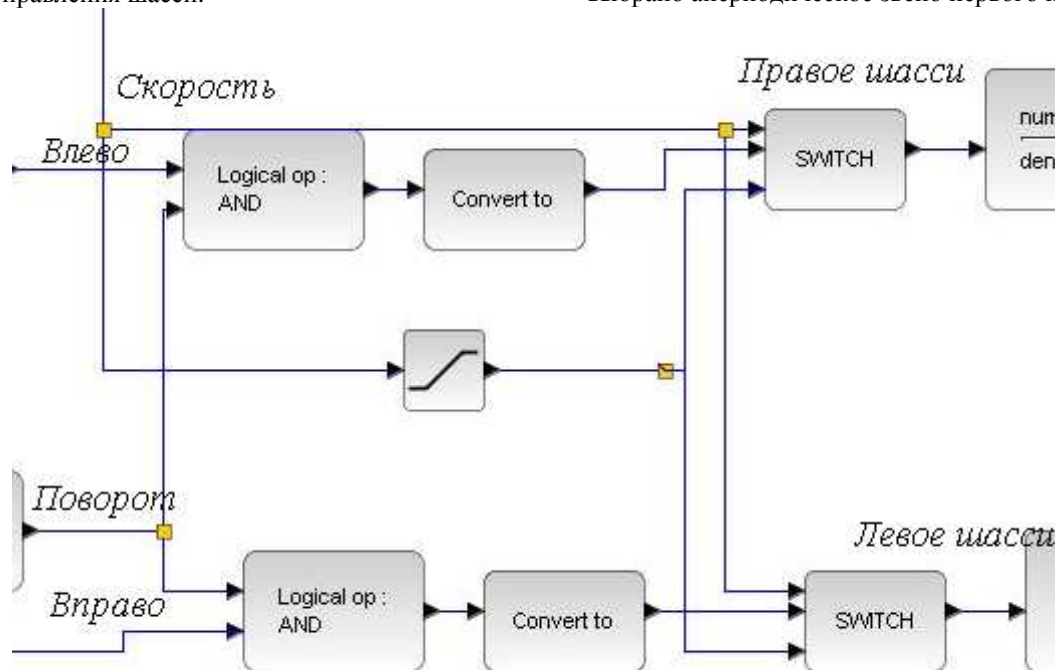


Рис.15. Схема управления правым (левым) шасси

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.

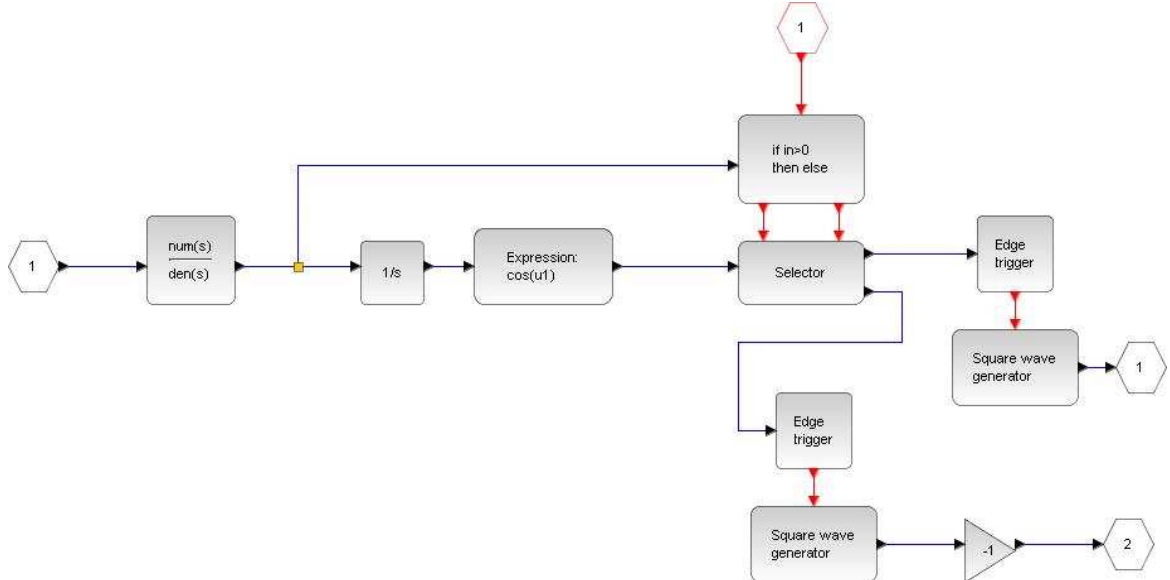


Рис.16. Широтно-импульсный модулятор

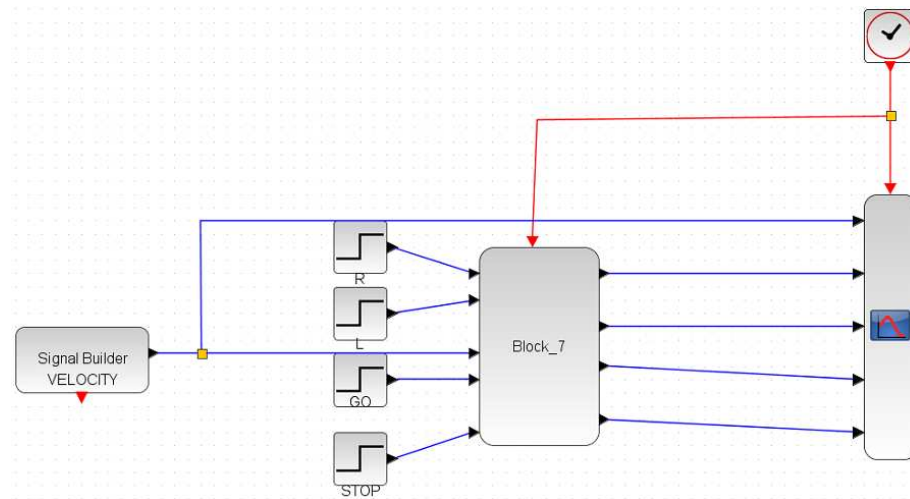


Рис.17. Схема управления

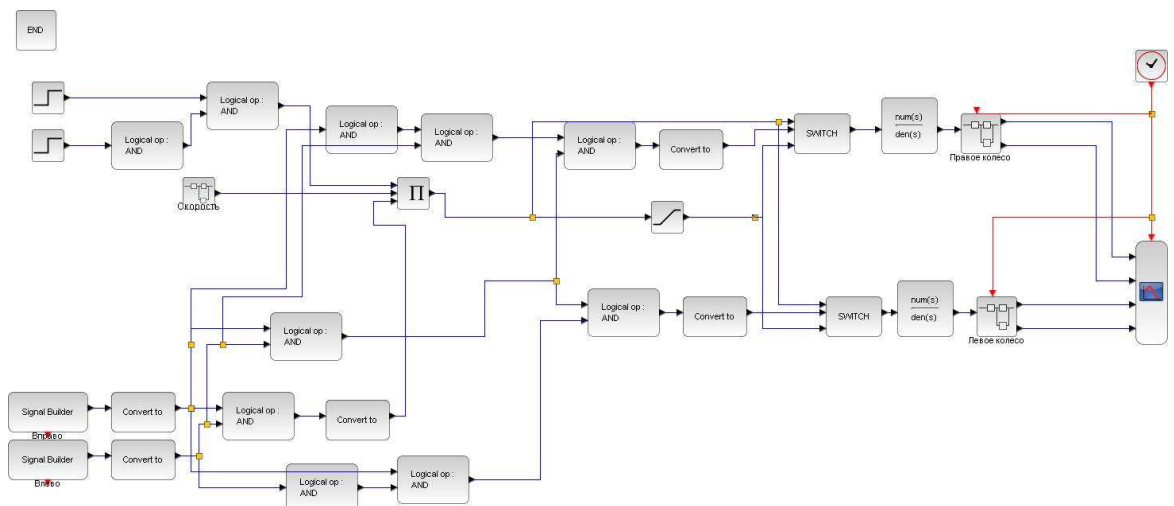


Рис. 18. Функциональная схема модели управления

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.

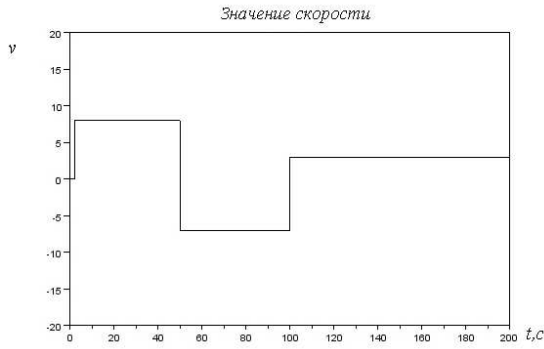


Рис. 19. Значение скорости с блока логического программирования сигнала для отладки системы управления

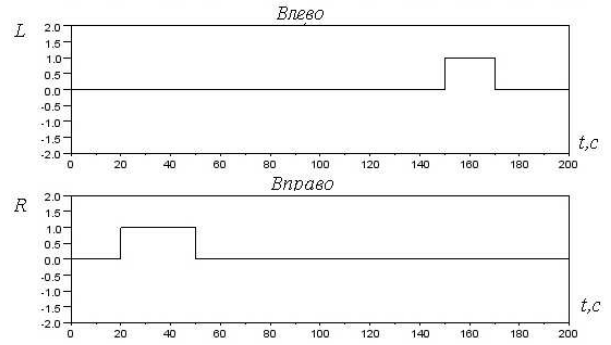


Рис. 2. Изменения направления движения

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

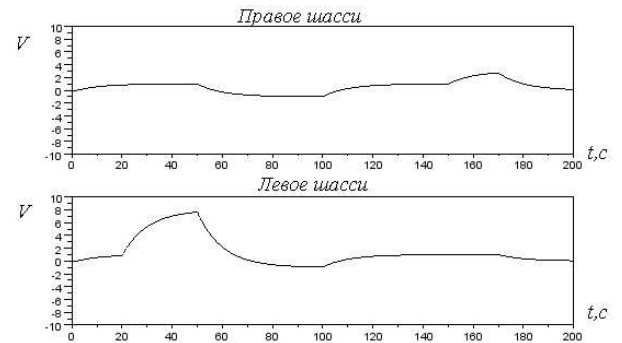


Рис. 21. Значение скорости на входе ШИМ

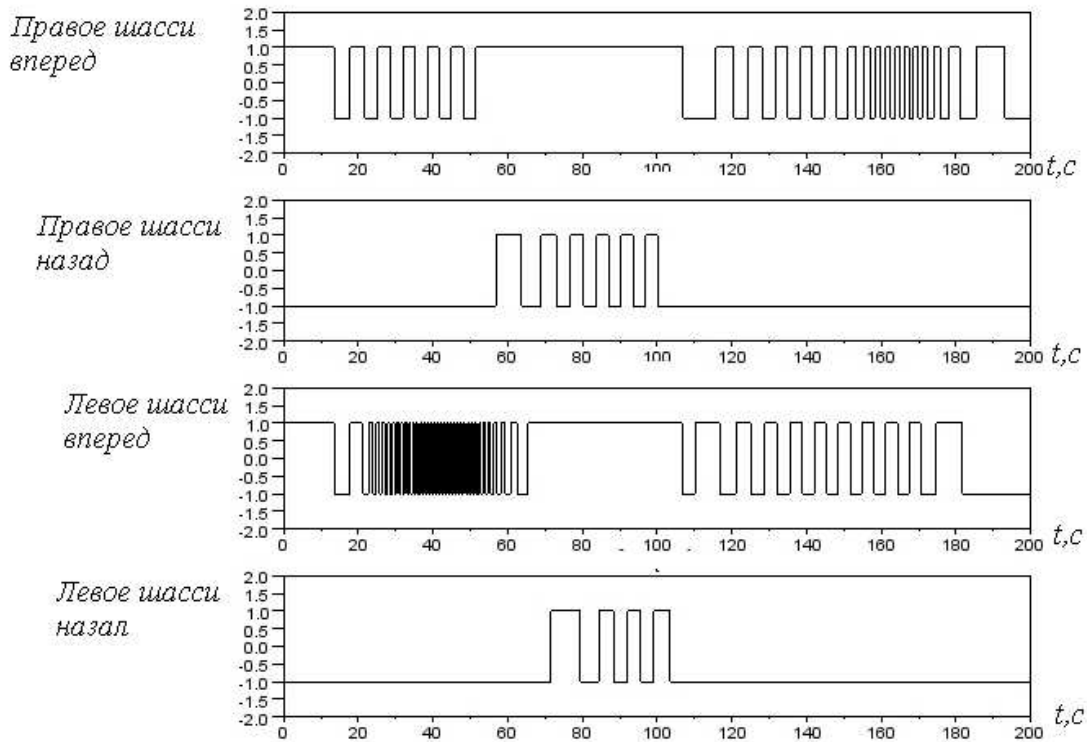


Рис. 22. Выходные сигналы

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.

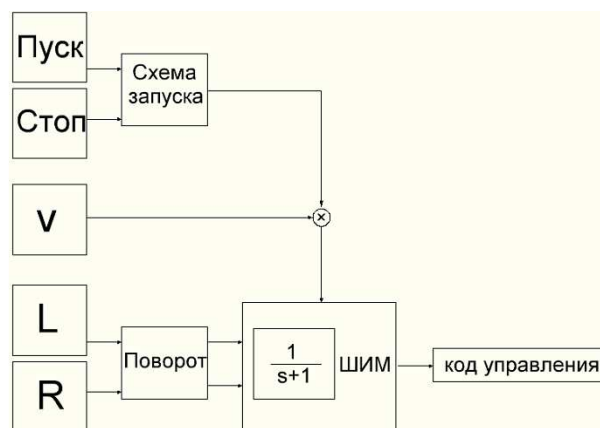


Рис. 23. Структурная схема

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.

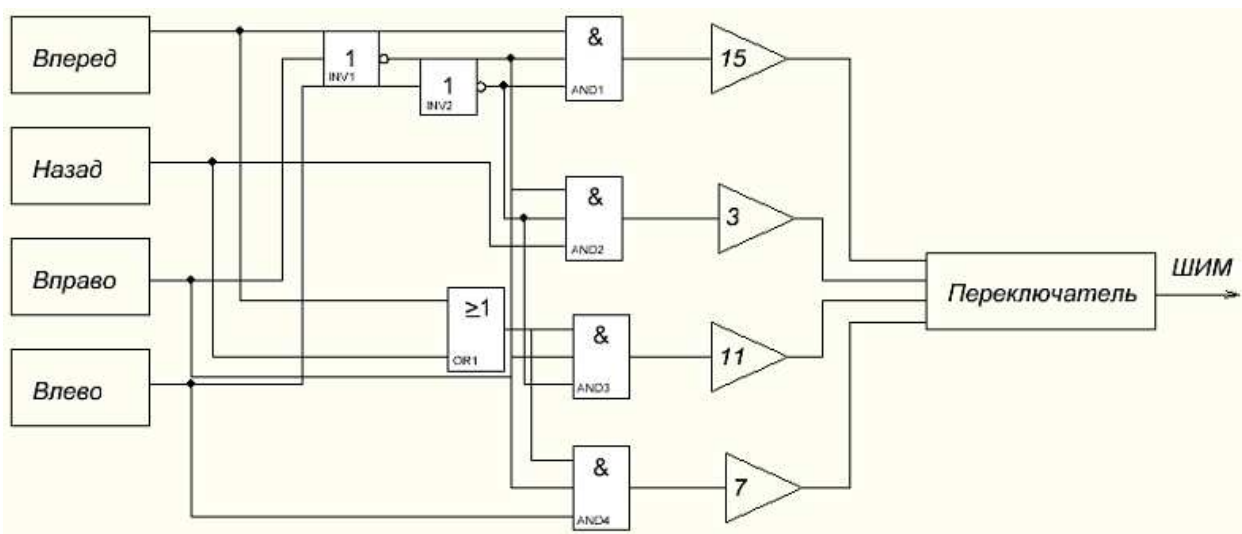


Рис. 24. Схема управления направлением

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.

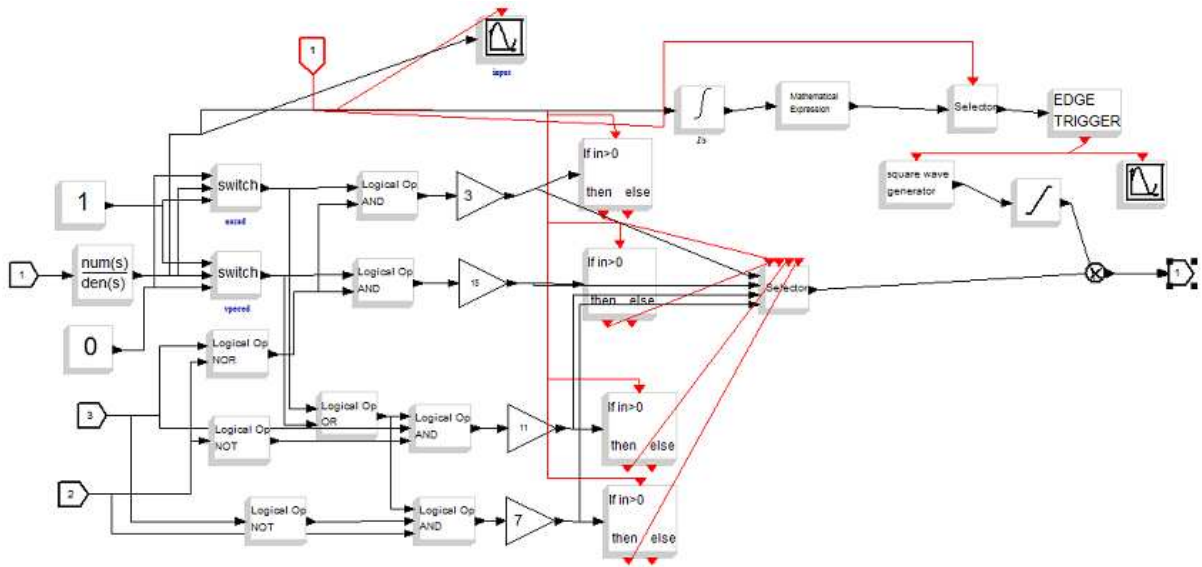


Рис. 25. Функциональная схема

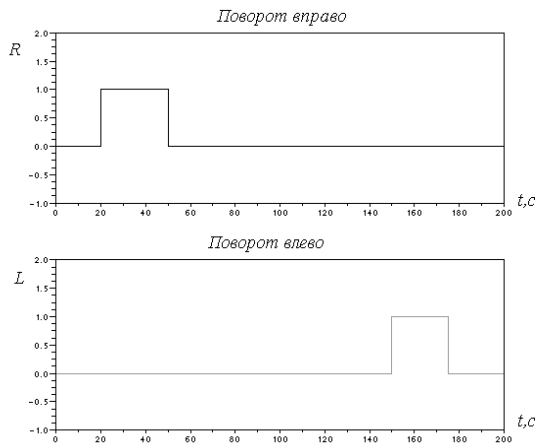


Рис. 26. Входные сигналы

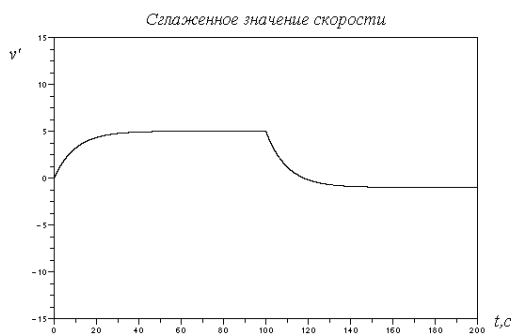


Рис. 27. Сглаженное значение скорости

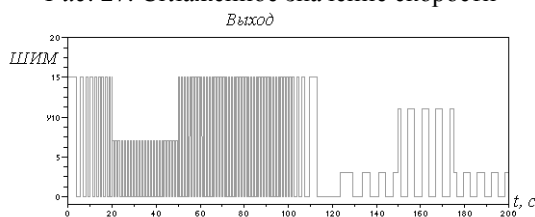


Рис. 28. Выход ШИМ

## 7. ОТЛАДКА И КОМПИЛЯЦИЯ

Для компиляции и запуска программы на объекте управления используется схема,

показанная на рис. 29.

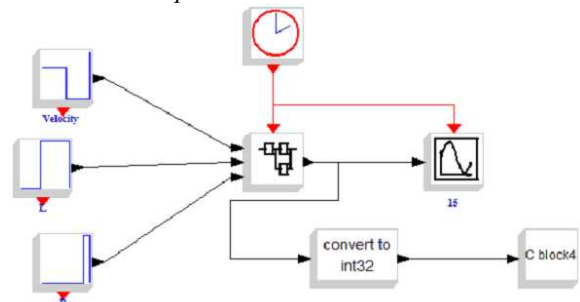


Рис. 29. Схема для тестирования

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](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](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](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: oao\_nips@bk.ru**

# Исследование вариантов создания интеллектуальных систем робототехники на базе одноплатных компьютеров и свободных операционных систем

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

**Аннотация:** Обсуждаются варианты создания простейших роботов на основе конструктора «Лего» и одноплатных компьютеров с использованием свободного (открытого) программного обеспечения. Приведен пример решение указанной задачи.

**Ключевые слова:** робототехника, свободное программное обеспечение

## 1. ВВЕДЕНИЕ

В предыдущих статьях был выполнен анализ современных математических и инженерных пакетов на предмет их применимости для создания систем управления методом визуального программирования. Как показывает исследование, проведенное ранее [1], наилучшими характеристиками обладает пакет *SciLab*, способный создавать исходный код модели на языке Си (попадающий под термины лицензии *GPL*). В рамках второй главы рассмотрим задачи создания встраиваемого варианта операционной системы под управлением ядра *Linux* для запуска *SciLab* приложений, а также для осуществления процессов мониторинга и телеуправления.

Таким образом, основными задачами данной главы являются:

- исследование и выбор операционной системы для одноплатного компьютера;
- разработка программной системы удаленной передачи данных устройству на малом расстоянии;
- исследование и тестирование работоспособности разработанной системы в различных условиях;
- использование совместно системы управления и системы передачи данных в программной реализации.

## 2 ИНТЕЛЛЕКТУАЛЬНАЯ РОБОТОТЕХНИЧЕСКАЯ СИСТЕМА

Интеллектуальная робототехническая система включает объект управления совместно со средой, в которой она работает. Объект управления представляет непосредственно механизмы перемещения инструмента и изделия. В состав манипуляторов входят исполнительные двигатели, которые осуществляют их перемещение по заданным законам  $R_d$  и  $R_i$ .

Информация о положении выходных звеньев манипуляторов определяется датчиками, расположенными в шарнирах звеньев манипуляторов, которые получают информации о выходных координатах механизмов перемещения, их скоростях, ускорениях и силах. Основная функция системы управления манипуляторами состоит в формировании законов перемещения исполнительными механизмами манипуляторов в реальном времени  $U_n(t)$  и  $U_d(t)$ . Данные системы обычно работают в следящем режиме, обеспечивающем выполнение каждой степенью подвижности манипуляторов заданной траектории перемещения с требуемыми точностью, скоростью и усилием. Выходными координатами манипуляторов являются  $R_d$  и  $R_i$ . В результате взаимодействия инструмента с деталью создается усилие  $P(t)$ , которое воздействует на исполнительные органы манипуляторов. Применительно к рассматриваемой системе в качестве объекта управления и внешней среды следует рассматривать манипуляторы перемещения изделия, инструмента и непосредственно сам технологический процесс.

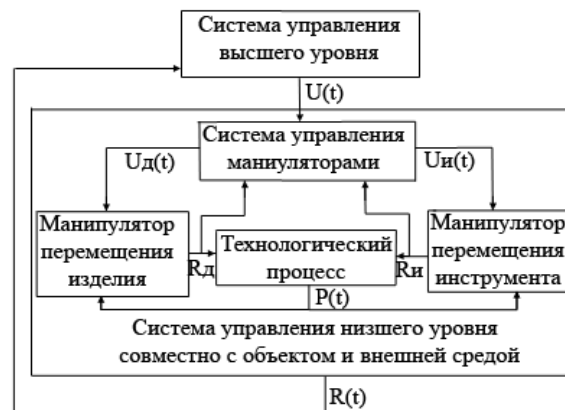


Рис. 1. Структура интеллектуальной робототехнической системы

На рис. 1 представлена обобщенная структура интеллектуальной робототехнической системы. В качестве системы управления высшего уровня выступает система, основанная на методах управления знаниями и принятия решений, использующая алгоритмы, рассмотренные в предыдущем разделе. Общим информационным управляющим каналом на систему управления



низшего уровня является канал передачи управляющих сигналов  $U(t)$  и обратной связи от системы низшего уровня - сигнал  $R(t)$ . Управляющее воздействие  $U(t)$  представляет выбранную программу действия из некоторого множества  $U$  и соответствующую заданной детали, либо обработке заданной поверхности детали. Какую из программ следует выбрать, решается системой высшего уровня как на основе информации от системы распознавания поверхности, так и на основе указаний оператора, управляющего робототехнической системой. Выбранная программа  $U(t)$  задается непрерывно в реальном масштабе времени. Обратная связь  $R(t)$  может нести полную информацию о работе системы управления низшего уровня в виде логических сигналов о ее состоянии, непрерывную информацию о геометрических размерах, качестве обработки поверхности детали и информацию о состоянии внешней среды, например, о температуре окружающей среды или двигателей, о состоянии сопутствующих обработке других устройств.

### 3. ОБЗОР СУЩЕСТВУЮЩИХ СРЕДСТВ ДЛЯ РЕШЕНИЯ ПОСТАВЛЕННЫХ ЗАДАЧ

#### 3.1 Обзор, сравнение и выбор одноплатных встраиваемых компьютеров

Для реализации нашего проекта, необходим встраиваемый одноплатный компьютер [2–24].

##### Преимущества одноплатного компьютера:

- компактность;
- оптимизация за счет правильного подбора встроенных компонентов;
- возможность расширения за счет горизонтального масштабирования (при наличии поддержки соответствующей шины, например, PC/104).

##### Недостатки:

- невозможность улучшения конфигурации;
- низкая скорость шины PC/104 (ISA).

Таблица 1

Сравнение характеристик одноплатных компьютеров

Характеристика	IEI Wafer-945gse2-n270-r10	Advantech PCM-4153
Процессор	Intel® Atom™ N270 1.6GHz	AMD Geode® LX800
Оперативная память	DDR2 SDRAM 1024MB	DDR333 SDRAM 512 MB
Чипсет	Intel® 945GSE +ICH7M	AMD Geode LX800
BIOS	AMI BIOS, SPI 8Mbit Flash ROM	AWARD 4Mbit Flash BIOS
Ethernet	2 × RTL8111E PCIe Interface GbE Controllers	Поддерживаемый чипсетом: 2 x 10/100Mbps - Intel 82551ER
Интерфейс ввода/вывода	2 × SATA; 6 × USB 2.0 (2 на поверхности, 4 поддерживаемых чипсетом); 1 × CF Type II; 1 × RS-232; 1 × RS-232/422/485 (поддерживаемые чипсетом); Клавиатура/мышь (поддерживаемые чипсетом); Цифровые вх/вых 8-бит , 4-бит вход/ 4-бит выход; Super I/O: ITE IT8718F	IDE интерфейс: Поддерживаемый чипсетом; Последовательный порт: 3×RS-232, Поддерживаемые чипсетом: 1xRS-232/422/485 Параллельный порт: 1 Клавиатура/мышь; 4×USB; Цифровые вх/вых: 8бит
Дополнительно	Звук: Realtek ALC662	Звук: Realtek ACL203
Питание	AT/ATX, +5V, +12V для LCD панели. Потребление: 5V×2A	AT, +5 V , +12 V для PC104 LCD инвертора. Потребление: +5 V × 1.46 A, +12 V × 0.02 A
Рабочая температура, °C	0 ~ 60	0 ~ 60
Размеры и вес	146 мм x 102 мм, 230г	96 мм x 115 мм, 162г
Поддержка ОС	Windows (все) Linux	Win XP, Win CE and Win XPe, Linux
Расширения	PC/104	PC/104
Стоимость, долл.	432	553

Сравнивая имеющиеся одноплатные компьютеры фирмы IEI модели Wafer-945gse2-n270-r10 и фирмы Advantech модели PCM-4153, можно выявить следующие сравнительные

характеристики, представленные в Таблице 1. Как видно из Таблицы 1, главное отличие данных одноплатных компьютеров состоит в центральном процессоре и оперативной памяти. Основным

требованием для работы в нашем проекте является быстродействие, стоимость, а также совместимость по размерам с другим оборудованием, используемым в исследовании, и поддержка производителем операционной системы *Linux*. В результате был сделан выбор компьютера фирмы *IEI* (приобретен за счет средств проекта).

### 3.2 Обзор, сравнение и выбор операционных систем для одноплатного компьютера

В настоящее время среди высокопроизводительных компьютеров все еще актуальны легковесные дистрибутивы, которые бы весили немного и загружались менее чем за минуту, и были бы просты в установке. Был произведен обзор компактных дистрибутивов систем, и выбраны наиболее подходящие из них для сравнения.

*Damn Small Linux (DSL)* – один из первых из всех мини-дистрибутивов на основе *Debian* для архитектуры x86. Со временем *Damn Small Linux* вырос в общественный проект, в который было вовлечены сотни человеко-часов точной настройки, включая полностью автоматическую систему установки, как локальных, так и сетевых программ. Кроме того, появилась система сохранения данных, которые можно было сохранять на любой носитель, включая жесткий диск, гибкий диск или устройство *USB*. Использует программу определения устройств от *Knoppix*, которая довольно долго грузится (для маленького дистрибутива), однако, тем не менее, правильно определяет большинство оборудования. Включает свою систему пакетов *.dsl*, но и позволяет использовать репозитории пакетов *Debian*. Оконный менеджер по умолчанию – *Fluxbox*. Создавался для использования на старых компьютерах, поэтому имеет низкие требования к процессору и оперативной памяти и маленький размер (50 мегабайт). Минимальные системные требования: ЦП – 486DX, ОЗУ – 16 МБ.

*Slax* основан на программном обеспечении *Slackware Linux*. Очень легко расширяется/перестраивается дополнительными модулями. *SLAX* загружается в память и освобождает носитель, с которого был загружен. Использует в качестве рабочей графической среды *KDE* (также в дистрибутив включён *Fluxbox*). Одно из преимуществ *SLAX* – модульная структура, что позволяет легко модифицировать его под конкретные нужды пользователя. Дистрибутив можно собрать самому (модули имеют расширение *\*.lzm*), включив в состав диска нужные приложения. Также есть скрипты, осуществляющие конвентирование в формат *\*.lzm* пакетов других дистрибутивов, в частности, *tgz2lzm* для конвертирования пакетов *Slackware* и *deb2lzm* для конвертирования пакетов *Debian/Ubuntu*. На сайте программы есть возможность в онлайн сформировать свою сборку дистрибутива с

произвольным выбором модулей, и скачать её впоследствии.

*Tiny Core Linux* – легковесный дистрибутив с графической оболочкой, цель которого обеспечение базовой системы с использованием *BusyBox*, *FLTK* и другого легковесного программного обеспечения. Маленький размер достигается в том числе за счет отсутствия дополнительного ПО: в нём нет ни браузера, ни офисных программ. Но все необходимые приложения могут быть легко загружены из интернета после запуска системы. Размер образа 10Мб, либо 6Мб без графической оболочки (*MicroCore*). Минимальные требования: ОЗУ 48Мб (36Мб для *MicroCore*), процессор i486DX. В *TCL* используется *Tiny X* вместо *X*-сервера, и *uClibc* вместо *glibc* для экономии памяти и уменьшения размера дистрибутива.

*Puppy Linux* – дистрибутив операционной системы *GNU/Linux*. Это дистрибутив, похожий в использовании на *Windows*, который включает все необходимые приложения, но при этом имеет достаточно малый размер (около 70 мегабайт) и скромные системные требования (процессор *Pentium I*, оперативная память — 32 мегабайта). Может полностью загружаться в оперативную память, не используя носитель. Отличается от других мини-дистрибутивов по скорости (загружается в 2-3 раза быстрее любого другого мини-дистрибутива) и системным требованиям (для нормальной работы дистрибутива достаточно 32 мегабайта оперативной памяти). Содержит минимальный набор самых нетребовательных к ресурсам программ. Использует оконный менеджер *JWM*. *Puppy Linux* работает практически на любом оборудовании, легок в освоении и использовании, а также легко настраивается под личные нужды конкретного пользователя.

*Gentoo* – дистрибутив, основной особенностью которого является сборка пакетов из исходных кодов. Это достаточно популярный дистрибутив *Linux* с мощной и гибкой технологией *Portage*, которая совмещает в себе возможности конфигурирования, настройки, а также автоматизированную систему управления пакетами. Отличительной особенностью *Gentoo* является возможность качественной оптимизации под конкретное аппаратное обеспечение.

В Таблице 2 представлены сравнительные характеристики операционных систем. Исходя из Таблицы 2, был выбран дистрибутив *Tiny Core Linux*, благодаря таким преимуществам, как легкость, быстрота и огромный функционал. С помощью данной системы были достигнуты поставленные задачи: управление двигателями на близком расстоянии.

Особенностью *Tiny Core Linux* является не её компактность как таковая, а как раз наоборот – расширяемость. Посредством специального ПО можно собрать систему, в которой не было бы ни единой лишней программы, причём сделать это куда быстрее и эффективнее, нежели при использовании дистрибутивов типа *Gentoo*.

Сравнение дистрибутивов *Linux*

Название	Достоинства	Недостатки
<i>Tiny Core Linux</i>	Легкая в установке, быстрая загрузка, никаких лишних пакетов, только минимальный пакет для работы, много устанавливаемых дополнений, размер дистрибутива 8-64мб. Очень хороший функционал	Не очень удобный интерфейс. Ориентирован на опытных пользователей.
<i>Slax</i>	Легкая в установке, быстрая загрузка. Простая в настройке система. Удобный интерфейс. Размер около 200Мб	Много лишних пакетов, которые нам для работы не нужны. Требует минимум 256Мб RAM
<i>Damn Small Linux</i>	Легкая в установке, быстрая загрузка. Компактная, всего 50Мб. Нетребовательная система, хватает всего 24Мб RAM	Многих модулей не хватает, необходимо доустанавливать.
<i>Purpy Linux</i>	Легкая в установке, быстрая загрузка. Удобный интерфейс. Если RAM более 256мб – система может работать в памяти, позволяя вытащить накопитель. Простая установка дополнений. Размер дистрибутива около 100Мб.	Много лишних пакетов, которые нам для работы не нужны.
<i>Gentoo</i>	Сборка с нуля, установка всего, что необходимо самому.	Сборка с нуля – сложная и долгая процедура.

## 3.3 Обзор шасси

Гусеничное шасси *RP5 Tank* является идеальным и недорогим для того чтобы построить на его базе мобильного робота. Основной характеристикой данного шасси является наличие двух электромоторов в комплекте с платформой (по 1 мотору на гусеницу), которые могут работать от 6 батареек AA, что является несомненным плюсом. Шасси маневренно и легко в управлении, способно преодолевать небольшие препятствия. В механизме шасси есть технологическое отверстие в центре корпуса, которое занимает 6 батареек типа AA для питания двигателей. Кроме того, в устройстве предусмотрены крепления для компьютера или какой-либо платы.

Характеристики шасси:

- напряжение двигателей: (7.2-12В) ;
- рабочий ток: 160, 180 мА;
- максимальная скорость: 30 см/с;
- коэффициент торможения: 80:1;
- преодолеваемый подъем: > 30 °;
- эффективная нагрузка:> 7.5 Кг;
- размеры:190мм x 130мм x 60мм;
- вес 700г.

На данный момент существует много проектов на данном шасси, есть проекты *RP5 Tank* и *Arduino*. Существует много различных модулей к плате, однако все эти проекты основаны на микроконтроллере *Arduino*. До текущего момента не было реализовано программных и аппаратных модулей для встраиваемых компьютеров. В данной работе

разрабатывается и исследуется система управления на одноплатном компьютере с полноценной операционной системой и сетевыми интерфейсами, так как данный набор программных и аппаратных средств предоставляет производительность и гибкость в управлении, в отличие от 8-битного контроллера *Arduino*. К тому же, одноплатный компьютер предоставляет в полное распоряжение сетевые интерфейсы *Wi-Fi* и *Ethernet*, в то время как на микроконтроллере *Arduino* эти интерфейсы доступны только через специальные платы расширения. В рамках первого этапа проекта была реализована система управления на базе одноплатного компьютера и шасси *RP5 Tank*. Другим перспективным шасси является более мощный вариант *DAGU6wd* (приобретена также в рамках гранта). Данная платформа представляет собой би колесную платформу с возможностью размещения до 5 кг полезной нагрузки, однако требует более мощной силовой части (рабочий ток до 6А 7.2в на двигатель). Работы по созданию робота на базе *DAGU 6WD* будут выполнены в рамках второго этапа проекта.

## 4. РАЗРАБОТКА СИСТЕМЫ УПРАВЛЕНИЯ

## 4.1 Установка ОС на носитель. Установка дополнений

Для того, чтобы установить данную систему на носитель необходимо выполнить несколько шагов:

- Скачать дистрибутив. *Tiny Core Linux* с официального сайта;

о С помощью программы *UnetBootIn* записать скачанный ISO образ на носитель.

Также данная программа записывает загрузчик в загрузочную область (MBR) на данный носитель. После этих простых шагов можно использовать данный носитель в одноплатном компьютере.

Для поставленных задач нам необходимо дополнительно установить пакеты программ.

Нужно использовать менеджер управления пакетами. Для начала необходимо соединиться с репозиторием, после чего появляется список доступных пакетов. При выборе какого-либо из них выводится его описание, список зависимостей, которые будут дополнительно установлены. Для установки достаточно нажать кнопку *Install Selected*, после чего пакет скачивается и устанавливается. Разрешение зависимостей происходит автоматически.

Дополнительные пакеты, необходимые для работы.

- о *OpenSSH* - для удаленного подключения к одноплатному компьютеру;
- о *mc* - для более удобной работы с файлами;
- о *ffserver* - для захвата и трансляции видео по сети. Нужно написать конфигурацию *ffserver* (в приложении) и его необходимо положить в папку *etc*;
- о *gcc* для удобства компиляции на одноплатном компьютере, необязателен;
- о *gfortran* - для запуска суперблока *Scicoslab*.

Кроме того в загрузочном файле необходимо прописать автозапуск всех необходимых используемых нами программ (*wi-fi*, *openssh*, *ffserver*). Ниже представлено содержимое файла *bootlocal.sh*.

```
#запуск openssh
sudo /usr/local/etc/init.d/openssh start
#запуск и начальные установки wi-fi
sudo ifconfig wlan0 down
sudo iwconfig wlan0 mode ad-hoc
sudo iwconfig wlan0 channel 1
sudo iwconfig wlan0 Bit 54Mb/s
sudo iwconfig wlan0 essid livliv
sudo ifconfig wlan0 192.168.1.3 netmask
255.255.255.0 up
sudo ifconfig wlan0 up
#запуск ffserver и ffmpeg для трансляции
камеры по сети
sudo ffserver
sudo ffmpeg -s 320x240 -f video4linux2 -i
/dev/video0 http://localhost:54345/webcam ffm
```

Также необходимо модифицировать файл *filetool.lst*, содержимое которого представлено ниже.

```
opt
home
#для доступа к ssh
usr/local/etc/ssh
#для сохранения пароля пользователя
etc/passwd
etc/shadow
```

## 4.2 Работа с сокетами в Linux. Программа передачи данных по сокету

Сокет (*socket*) - это конечная точка сетевых коммуникаций. Приложение просто пишет данные в сокет; их дальнейшая буферизация, отправка и транспортировка осуществляется используемым стеком протоколов и сетевой аппаратурой. Чтение данных из сокета происходит аналогичным образом. В программе сокет идентифицируется дескриптором типа *int*. Программа получает дескриптор от операционной системы при создании сокета, а затем передаёт его сервисам *socket API* для указания сокета, над которым необходимо выполнить то или иное действие.

Тип сокета определяет способ передачи данных по сети. Существуют следующие типы сокетов:

- о потоковые сокеты - требуют установки соединения. Передача потока данных с предварительной установкой соединения. Обеспечивается надёжный канал передачи данных, при котором фрагменты отправленного блока не теряются, не переупорядочиваются и не дублируются;

- о дейтаграмные сокеты, не требуют установки соединения. Передача данных в виде отдельных сообщений (датаграмм). Предварительная установка соединения не требуется. Обмен данными происходит быстрее, но является ненадёжным: сообщения могут теряться в пути, дублироваться и переупорядочиваться. Допускается передача сообщения нескольким получателям (*multicasting*) и широковещательная передача (*broadcasting*);

- о доменные сокеты - средства для обеспечения межпроцессного взаимодействия (*IPC*);

- о низкоуровневые сокеты (*raw sockets*). Их отличие от обычных сокетов состоит в том, что с их помощью программа может взять на себя формирование некоторых заголовков, добавляемых к сообщению.

Рассмотрим пользовательский интерфейс работы с сокетами.

```
#include <sys/types.h>
#include <sys/socket.h>

int socket(int domain, int type, int protocol);
int bind(int sockfd, struct sockaddr *addr, int
addrLen);
int listen(int sockfd, int backlog);
int accept(int sockfd, void *addr, int
*addrLen);
int connect(int sockfd, struct sockaddr
*serv_addr, int addrLen);
int send(int sockfd, const void *msg, int len,
int flags);
int sendto(int sockfd, const void *msg, size_t
len, int flags, const struct sockaddr *to, socklen_t
toLen);
int recv(int sockfd, void *buf, int len, int
flags);
```



```
int recvfrom(int sockfd, void *buf, size_t len,
int flags, struct sockaddr *from, socklen_t *fromlen);
```

```
int close(int fd);
```

Рассмотрим атрибуты сокета.

С каждым сокетом связываются три атрибута: домен, тип и протокол. Эти атрибуты задаются при создании сокета и остаются неизменными на протяжении всего времени его существования. Для создания сокета используется функция `socket`, имеющая следующий прототип.

```
#include <sys/types.h>
#include <sys/socket.h>
```

```
int socket(int domain, int type, int protocol);
```

Домен определяет пространство адресов, в котором располагается сокет, и множество протоколов, которые используются для передачи данных. Чаще других используются домены *Unix* и *Internet*, задаваемые константами `AF_UNIX` и `AF_INET` соответственно. При задании `AF_UNIX` для передачи данных используется файловая система ввода/вывода Unix. В этом случае сокеты используются для межпроцессного взаимодействия на одном компьютере и не годятся для работы по сети. Константа `AF_INET` соответствует *Internet*-домену. Сокеты, размещённые в этом домене, могут использоваться для работы в любой IP-сети.

Последний атрибут определяет протокол, используемый для передачи данных. Как было отмечено ранее, часто протокол однозначно определяется по домену и типу сокета. В этом случае в качестве третьего параметра функции `socket` можно передать 0, что соответствует протоколу по умолчанию. Тем не менее, иногда (например, при работе с низкоуровневыми сокетами) требуется задать протокол явно. Числовые идентификаторы протоколов зависят от выбранного домена.

Прежде чем передавать данные через сокет, его необходимо связать с адресом в выбранном домене (эту процедуру называют именованием сокета). Иногда связывание осуществляется неявно (внутри функций `connect` и `accept`), но выполнять его необходимо во всех случаях. Вид адреса зависит от выбранного вами домена. В Unix-домене это текстовая строка - имя файла, через который происходит обмен данными. В Internet-домене адрес задаётся комбинацией IP-адреса и 16-битного номера порта. IP-адрес определяет хост в сети, а порт - конкретный сокет на этом хосте. Протоколы *TCP* и *UDP* используют различные пространства портов.

Для явного связывания сокета с некоторым адресом используется функция `bind`. Её прототип имеет вид:

```
int bind(int sockfd, struct sockaddr *addr, int
addrlen);
```

В качестве первого параметра передаётся дескриптор сокета, который мы хотим привязать

к заданному адресу. Второй параметр, `addr`, содержит указатель на структуру с адресом, а третий - длину этой структуры.

```
struct sockaddr_in {
short int sin_family; // Семейство адресов
unsigned short int sin_port; // Номер порта
struct in_addr sin_addr; // IP-адрес
unsigned char sin_zero[8]; // "Дополнение"
до размера структуры sockaddr
};
```

Здесь поле `sin_family` соответствует полю `sa_family` в `sockaddr`, в `sin_port` записывается номер порта, а в `sin_addr` - IP-адрес хоста. Поле `sin_addr` само является структурой, которая имеет вид:

```
struct in_addr {
unsigned long s_addr;
};
```

Существует два порядка хранения байтов в слове и двойном слове. Один из них называется порядком хоста (host byte order), другой - сетевым порядком (network byte order) хранения байтов. При указании IP-адреса и номера порта необходимо преобразовать число из порядка хоста в сетевой. Для этого используются функции `htons` (Host TO Network Short) и `htonl` (Host TO Network Long). Обратное преобразование выполняют функции `ntohs` и `ntohl`.

Для TCP-сокеты процесс взаимодействия рассмотрен в *Таблице 3*.

Установка соединения на стороне сервера состоит из четырех этапов:

- o Создание сокета (функция `socket`);
- o Привязка сокета к локальному адресу (функция `bind`). Если компьютер имеет несколько сетевых интерфейсов с различными IP-адресами, вы можете принимать соединения только с одного из них, передав его адрес функции `bind`. Для соединения с клиентами через любой интерфейс необходимо задать в качестве адреса константу `INADDR_ANY`. Что касается номера порта, можно задать конкретный номер или 0;
- o На следующем шаге создаётся очередь запросов на соединение (функция `listen`);

```
int listen(int sockfd, int backlog);
```

Первый параметр - дескриптор сокета, а второй задаёт размер очереди запросов. Каждый раз, когда очередной клиент пытается соединиться с сервером, его запрос ставится в очередь, так как сервер может быть занят обработкой других запросов. Если очередь заполнена, все последующие запросы будут игнорироваться.

- o При наступлении готовности сервера к обслуживанию запроса вызывается функция `accept`.

```
int accept(int sockfd, void *addr, int
*addrlen);
```



Схема взаимодействия серверного и клиентского сокета

Сервер	Клиент
1. Создание сокета, ожидающего запросы на соединения от клиентов.	2. Создание клиентского сокета и попытка соединения с сервером
3. Разрешение попытки соединения от клиента	-
4. Передача и прием данных.	4. Передача и прием данных.
5. Закрытие соединения.	5. Закрытие соединения.

Функция *accept* создаёт для общения с клиентом новый сокет и возвращает его дескриптор. Параметр *sockfd* задаёт слушающий сокет. После вызова он остаётся в слушающем состоянии и может принимать другие соединения. В структуру, на которую ссылается *addr*, записывается адрес сокета клиента, который установил соединение с сервером. В переменную, адресуемую указателем *addrlen*, изначально записывается размер структуры; функция *accept* записывает туда длину, которая реально была использована. Если вас не интересует адрес клиента, вы можете просто передать *NULL* в качестве второго и третьего параметров.

Следует обратить внимание, что полученный от *accept* новый сокет связан с тем же самым адресом, что и слушающий сокет, но дело в том, что адрес *TCP*-сокета не обязан быть уникальным в *Internet*-домене. Уникальными должны быть только соединения, для идентификации которых используются два адреса сокетов, между которыми происходит обмен данными.

На стороне клиента для установления соединения используется функция *connect*, которая имеет следующий прототип.

```
int connect(int sockfd, struct sockaddr
*serv_addr, int addrlen);
```

Здесь *sockfd* - сокет, который будет использоваться для обмена данными с сервером, *serv\_addr* содержит указатель на структуру с адресом сервера, а *addrlen* - длину этой структуры.

После того как соединение установлено, можно начинать обмен данными. Для этого используются функции *send* и *recv*. Функция *send* используется для отправки данных и имеет следующий прототип.

```
int send(int sockfd, const void *msg, int len,
int flags);
```

Здесь *sockfd* - это, как всегда, дескриптор сокета, через который мы отправляем данные, *msg* - указатель на буфер с данными, *len* - длина буфера в байтах, а *flags* - набор битовых флагов, управляющих работой функции.

Для чтения данных из сокета используется функция *recv*.

```
int recv(int sockfd, void *buf, int len, int
flags);
```

В целом её использование аналогично *send*. Она точно так же принимает дескриптор сокета, указатель на буфер и набор флагов. Флаг *MSG\_OOB* используется для приёма срочных данных, а *MSG\_PEEK* позволяет «подсмотреть» данные, полученные от удалённого хоста, не удаляя их из системного буфера (это означает, что при следующем обращении к *recv* вы получите те же самые данные). Полный список флагов можно найти в документации. По аналогии с *send* функция *recv* возвращает количество прочитанных байтов, которое может быть меньше размера буфера.

Закончив обмен данными, нужно закрыть сокет с помощью функции *close*. Это приведёт к разрыву соединения.

```
int close(int fd);
```

Для дейтаграмных сокетов (*SOCK\_DGRAM*) установка соединения не требуется. Вместо этого после открытия сокета можно использовать функции *recvfrom* и *sendto*

```
int recvfrom(int sockfd, void *buf, size_t len,
int flags, struct sockaddr *from, socklen_t *fromlen);
int sendto(int sockfd, const void *msg, size_t
len, int flags, const struct sockaddr *to, socklen_t
toLen);
```

Однако можно все равно использовать соединение для дейтаграмных сокетов. На системном уровне установления соединения не произойдет, но вместо функций *recvfrom* и *sendto* можно будет использовать функции *send* и *recv*.

В данной работе для удаленного управления используются дейтаграмные сокет (протокол *UDP*) по следующим причинам:

- отсутствие необходимости установки соединения
- высокая скорость передачи
- Отсутствие установки соединения перекладывает ответственность по обработке сбоев при передаче пакетов на разработчика системы. В данной системе нет необходимости

использовать протокол *TCP* с установлением соединения ввиду того, что потерянные данные при повторной передаче могут быть неактуальны и кроме того, повторная отправка вносит дополнительные задержки в процесс коммуникации. Протокол *UDP* избавлен от проблем с установкой соединения, однако ненадежность передачи данных может привести к неправильному порядку принятия пакетов. Для устранения ошибок принятия устаревших пакетов дополнительно передается отметка времени (*timestamp*), и все пакеты с отметкой времени меньше предыдущих пакетов отбрасываются.

Так как для принятия пакетов используется функция *recv(recvfrom)*, блокирующая вывод, то при отсутствии пакетов функция заблокирует поток. Существует несколько способов решения данной проблемы:

- работа с сокетами в отдельном потоке;
- неблокирующие сокеты;
- установка таймаута для выхода из функции принятия пакетов.

Самым простым способом является установка таймаута чтения.

Для установки таймаута чтения пакетов используется функция *setsockopt*.

```
#include <sys/types.h>
#include <sys/socket.h>
int setsockopt(int s, int level, int optname,
const void *optval, socklen_t optlen);
```

Функция *setsockopt* позволяет манипулировать флагами, установленными на сокете. Флаги могут существовать на нескольких уровнях протоколов; они всегда присутствуют на самом верхнем из них.

При манипулировании флагами сокета должен быть указан уровень, на котором находится этот флаг, и имя этого флага. Для манипуляции флагами на уровне сокета *level* задается как *SOL\_SOCKET*. Для манипуляции флагами на любом другом уровне этим функциям передается номер соответствующего протокола, управляющего флагами. Например, для указания, что флаг должен интерпретироваться протоколом *TCP*, в параметре *level* должен передаваться номер протокола *TCP*.

Флаг *optname SO\_RCVTIMEO* устанавливает значение тайм-аута, который определяет максимальное количество времени, в течение которого функция ожидает его завершения. Он принимает значение *timeval* структуры с указанием предела в секундах и микросекундах.

Существует множество других флагов, документацию по ним можно прочитать в интернете.

Параметры *optval* и *optlen* используются в функции *setsockopt* для доступа к значениям флагов. Если значение флага не используется, то параметр *optval* может быть *NULL*.

В случае успеха возвращается ноль. При ошибке возвращается -1, а значение *errno* устанавливается должным образом.

### 4.3 Создание блока в ScicosLab . Разработка программного решения для работы с устройством

Для передачи данных по беспроводной сети нам необходимо написать небольшую программу на *C*. Для этого в *ScicosLab* есть специальный блок.

Существующий блок *C block* содержит в себе следующий код:

```
#include <scicos/scicos_block4.h>

void socket_client(scicos_block *block,int
flag)
{
/* init */
if (flag == 4) {
/* output computation */
} else if(flag == 1) {
/* ending */
} else if (flag == 5) {
}
}
```

Функция *socket\_client* имеет на входе следующие параметры: *block* – указатель на текущий блок и *flag* – состояние работы модели. Значение флага 4 – инициализация схемы – открытие доступа к устройствам, открытие файлов

Значение флага 1 – вход в блок при моделировании – в данном случае при выполнении этого условия нужно считывать данные и отправлять их удаленному компьютеру.

Значение флага 5 – завершение моделирования, закрытие файлов и портов.

При создании блока в нашем случае надо указать, что выходных портов нет, то есть, убрать значения *Output port size*, подробнее как это сделать показано на *рис. 2*.

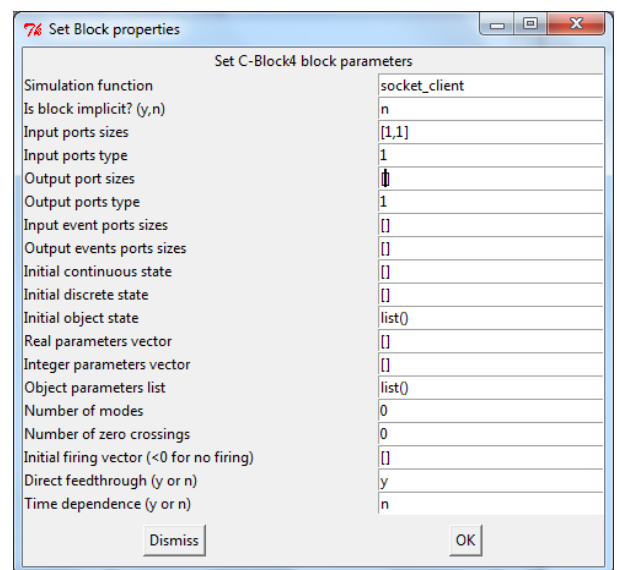


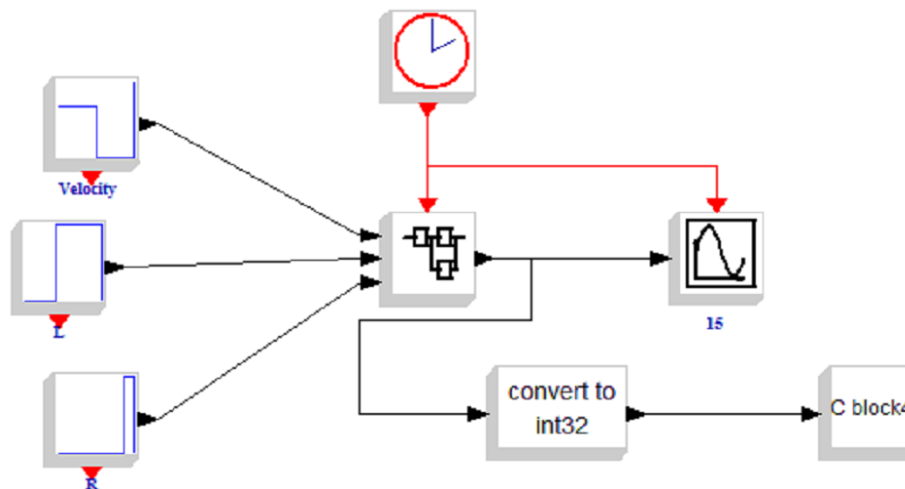
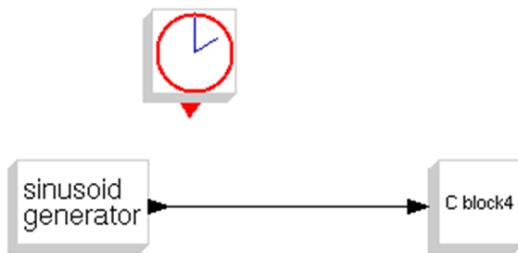
Рис. 2. Создание блока *Cblock*

Для использования входных сигналов в *C block* используется функция *GetRealInPortPtrs(block,1)*. Далее, в код внедряется работа с сокетами в *C block*. Простейшая схема передачи синусоидального сигнала для тестирования представлена на *рис. 3*.

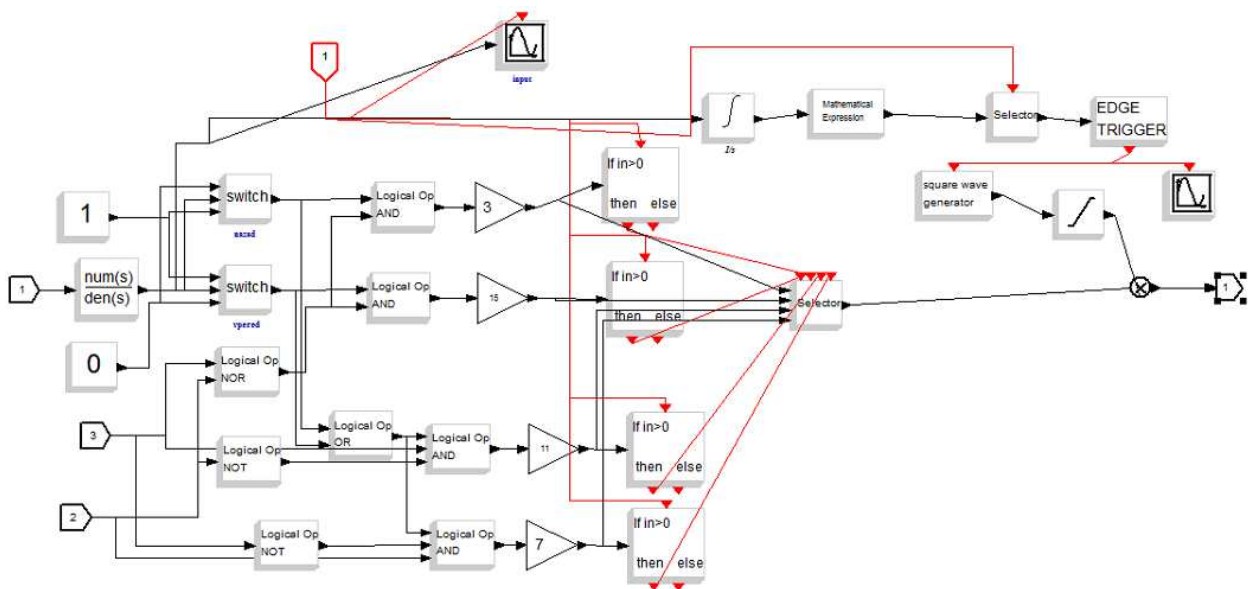
*Рис. 3* Схема передачи синусоидального сигнала

Чтобы создать программу для управления шасси необходимо собрать схему в один суперблок. С помощью *scicos* и функции *Code Generation* можно получить код проекта, созданного в *ScicosLab*, и скомпилировать его в единую программу, которую можно запустить на одноплатном компьютере.

На *рис. 4-5* представлена схема ШИМ для управления двигателями шасси в *ScicosLab*, которая передает данные по беспроводной связи на шасси на одноплатном компьютере под управлением ОС *Linux*.



*Рис. 4* Схема ШИМ для управления двигателями шасси



*Рис. 5* Схема ШИМ для управления двигателями шасси

#### 4.4 Работа с портами gpio

*GPIO (General Purpose Input-Output)* – это портв общего назначения, входы / выходы микроконтроллера, доступные для прямого

управления.

*GPIO* являются двунаправленными выводами (регистрами) ввода/вывода, которые нужны для реализации функций предусмотренных производителем микросхемы (по умолчанию),

или функций задаваемых (описываемых программно) пользователем.

Управление *GPIO* можно осуществлять несколькими способами:

- из пользовательского пространства:
- использование *DebugFS* (со специально скомпилированным ядром);
- использование функций *inb* и *outb* (при наличии отображения портов в *BIOS*).
- Из пространства ядра:
- использование драйвера для работы с *GPIO*.

Самый простой способ – это использование команд чтения и запись в порт *inb* и *outb*. Однако, такой способ доступен только при наличии отображенных портов, что запрограммировано в прошивке *BIOS*.

Использование *DebugFS* для работы с *GPIO* подразумевает наличие драйвера для установленного микроконтроллера *SuperIO* и скомпилированное ядро с поддержкой *DebugFS* и *GPIO*. В случае работы через *DebugFS* порты *GPIO* будут доступны в виде файлов из пользовательского пространства.

При работе в пространстве ядра необходимо писать свой драйвер, а также запускать его на ядре, скомпилированном с поддержкой *GPIO*.

На плате одноплатного компьютера установлен контроллер ввода-вывода *SuperIO IT8718F*, для которого в *BIOS* была поддержка портов ввода-вывода.

Для работы с функциями *inb* и *outb* необходимо из-под суперпользователя разрешить ввод/вывод в порты. Для этого используются функции *ioperm* и *iopl*

```
#include <sys/io.h>
int ioperm(unsigned long from, unsigned long
num, int turn_on);
int iopl(int level);
```

Функции *ioperm* и *iopl* изменяет уровень привилегий ввода/вывода текущего процесса в соответствии с *level*. Этот вызов необходим для того, чтобы 8514-совместимые X-серверы могли работать под управлением Linux. Этим X-серверам необходим доступ ко всем 65536-и портам ввода/вывода, соответственно, им недостаточно вызова *ioperm*. В дополнение к тому, что на высоком уровне привилегий процессу разрешен неограниченный доступ к портам ввода/вывода, он может также запретить системные прерывания. Стандартный уровень привилегий ввода/вывода обычного процесса равен 0.

Функция *ioperm()* может дать доступ только к портам с 0x000 по 0x3ff; для других портов нужно использовать *iopl()*, который дает доступ ко всем портам сразу. Для предоставления доступа программе ко всем портам ввода/вывода нужно использовать системный вызов *iopl(3)* от суперпользователя.

- *outb, outw, outl, outsb, outsw, outsl* - функции вывода данных в порт;

- *inb, inw, inl, insb, insw, insl* - функции чтения данных порта

Это семейство функций используется для низкоуровневой работы с портами ввода/вывода. В основном, они предназначены для использования внутри ядра, но могут быть вызваны и пользовательской программой.

Для управления портами ввода-вывода, в том числе и *GPIO*, на одноплатном компьютере установлен микроконтроллер *ITE IT8718F*, у которого контакты 20-23 и 30-33 выведены на порты ввода и вывода соответственно. *BIOS* запрограммировано обращение к порту по адресу 0xA21.

Для работы с портами *GPIO* в нашей программе необходимо использовать функции *inb* и *outb*:

- *outb*(сигнал, 0xA21) – вывод сигнала по адресу 0xA21;
- *inb*(0xA22) – захват сигнала с адреса 0xA22.

#### 4.5. Схема сопряжения одноплатного компьютера и шасси

Для сопряжения шасси и одноплатного компьютера необходима схема, которая бы принимала сигнал с выходов компьютера и передавала на двигатели шасси.

Для управления двигателями шасси необходимо устройство, которое бы преобразовывало управляющие сигналы малой мощности в токи, достаточные для управления моторами. Такое устройство называют драйвером двигателей.

Существует достаточно много самых различных схем для управления электродвигателями. Они различаются как мощностью, так и элементной базой, на основе которой они выполнены.

Для данного проекта наиболее подходящим является простой драйвер управления двигателями, выполненный в виде микросхемы L293D.

L293D содержит сразу два драйвера для управления электродвигателями небольшой мощности (четыре независимых канала, объединенных в две пары). Имеет две пары входов для управляющих сигналов и две пары выходов для подключения электромоторов. Кроме того, у L293D есть два входа для включения каждого из драйверов. Эти входы используются для управления скоростью вращения электромоторов с помощью широтно-модулированного сигнала (ШИМ).

L293D обеспечивает разделение электропитания для микросхемы и для управляемых ею двигателей, что позволяет подключить электродвигатели с большим напряжением питания, чем у микросхемы. Разделение электропитания микросхем и электродвигателей может быть также необходимо для уменьшения помех, вызванных бросками напряжения, связанными с работой моторов.

Принцип работы каждого из драйверов, входящих в состав микросхемы, идентичен, поэтому на рис. 2.6 приведена функциональная схема одного мотора.

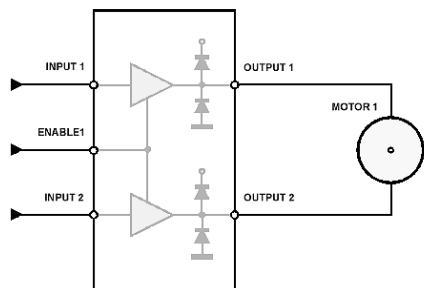


Рис. 6. Схема подключения драйвера

К выходам OUTPUT1 и OUTPUT2

подключен электромотор MOTOR1. На вход ENABLE1, включающий драйвер, подается сигнал (+5V). Если при этом на входы INPUT1 и INPUT2 не подаются сигналы, то мотор вращаться не будет.

Если вход INPUT1 соединить с положительным полюсом источника питания, а вход INPUT2 - с отрицательным, то мотор начнет вращаться. Если поменять полярность - соответственно будет вращаться в другую сторону. Если на вход ENABLE1 ничего не подается, то при любых вариантах наличия сигналов на входах INPUT1 и INPUT2 мотор вращаться не будет.

Представить принцип работы драйвера двигателя можно, рассмотрев следующую Таблицу 4:

Таблица 4

Набор состояний на входах и выходах двигателя

ENABLE1	INPUT1	INPUT2	INPUT2	OUTPUT1	OUTPUT2
1	0	0	0	0	0
1	1	0	0	1	0
1	0	1	1	0	1
1	1	1	1	1	1

На рис. 7 представлены назначения выводов микросхемы L293D.

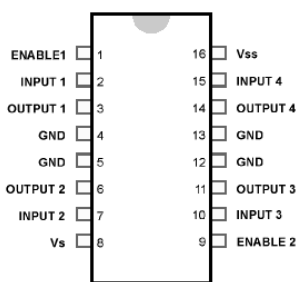


Рис. 7. Микросхема драйвера L293D

Из рис. 7 видно следующее:

- входы ENABLE1 и ENABLE2 отвечают за включение каждого из драйверов, входящих в состав микросхемы;
- входы INPUT1 и INPUT2 управляют двигателем, подключенным к выходам OUTPUT1 и OUTPUT2;
- входы INPUT3 и INPUT4 управляют двигателем, подключенным к выходам OUTPUT3 и OUTPUT4;
- контакт Vs отвечает за питание электродвигателей;
- контакт Vss обеспечивает питание самой микросхемы;

○ четыре контакта GND соединяют с "землей" (общим проводом или отрицательным полюсом источника питания).

Рассмотрим характеристики микросхемы L293D

- напряжение питания двигателей (Vs) - 4,5...36В;
- напряжение питания микросхемы (Vss) - 5В;
- допустимый ток нагрузки - 600mA (на каждый канал);
- пиковый (максимальный) ток на выходе - 1,2A (на каждый канал);
- логический "0" входного напряжения - до 1,5В;
- логическая "1" входного напряжения - 2,3...7В;
- скорость переключений до 5 кГц;
- защита от перегрева.
- Кроме того, для сопряжения шасси и одноплатного компьютера необходим инвертор, чтобы задавать направление движения. Для этого используется микросхема К155ЛН1(шесть логических элементов НЕ). На рис. 8 представлено условно-графическое изображение К155ЛН1.
- 1,3,5,9,11,13 - входы;
- 2,4,6,8,10,12 - выходы;
- 7 - общий;



- 14 - напряжение питания.

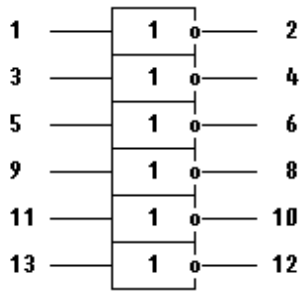


Рис. 8 Условно-графическое изображение микросхемы К155ЛН1.

На рис. 9 представлена структурная схема управления двигателями шасси с портов GPIO одноплатного компьютера.

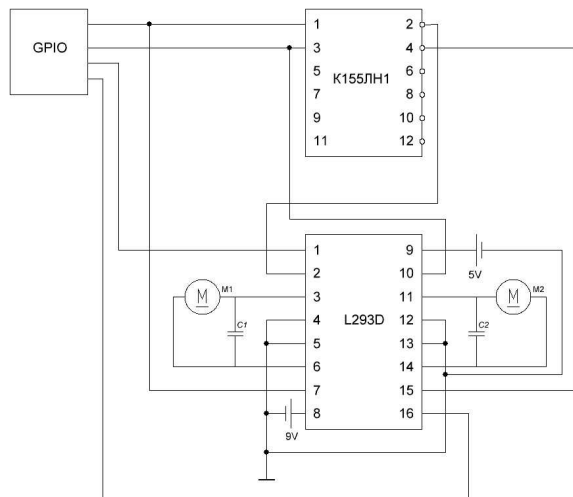


Рис. 9 Структурная схема управления двигателями с GPIO

Принцип работы схемы следующий:

На GPIO выходы одноплатного компьютера приходит сигнал в виде числа. Число кодируется в двоичную систему и на цифровые выходы устанавливаются последние 4 бита этого числа. Схема выводов представлена в Таблице 5.

Описание:

- Выход 0 – ШИМ двигатель 1;
- Выход 1 - ШИМ двигатель 2;
- Выход 2 – направление двигателя 1;

Выход 3 – направление двигателя 2.

К драйверу L293D подключено +5В с цифровых выходов одноплатного компьютера для питания микросхемы и +9В от батареек типа АА для питания двигателей. Питание одноплатного компьютера от внешнего аккумулятора 5В. Параллельно к каждому двигателю подключен керамический конденсатор номиналом ~0,1 мкФ, который помогает справиться с наводками от работающего электродвигателя

Таблица 5

Схема выводов цифровых выводов

Контакт	Описание	Контакт	Описание
1	Земля	2	+5В
3	Выход 3	4	Выход 2
5	Выход 1	6	Выход 0
7	Вход 3	8	Вход 2
9	Вход 1	10	Вход 0

#### 4.6 Сборка системы

Структурная схема системы представлена на рис. 10. На рис. 11 представлен собранный макет системы управления шасси на одноплатном компьютере. Управление движением моторизованного компьютера происходит с помощью внешнего ПК, с которого на программный сервер в одноплатном компьютере посылаются управляющие сигналы; далее, эти сигналы обрабатываются на встроенном компьютере, а данные с камеры передаются обратно на стационарный ПК. Передача данных производится по сокету по беспроводной связи, в данном случае *Wi-Fi*. На ПК может использоваться как отдельная программа передачи данных с клавиатуры, либо программа в блоке *Scilab*.

Для начала на ПК подключиться к сети *Ad-hoc livliv*. Для доступа к *Web* камере, подключенной к одноплатному компьютеру необходимо набрать в браузере в адресной строке <http://192.168.1.3:54345> и открыть просмотр в удобном для пользователя формате: *flv* или *swf*. (рис. 12)

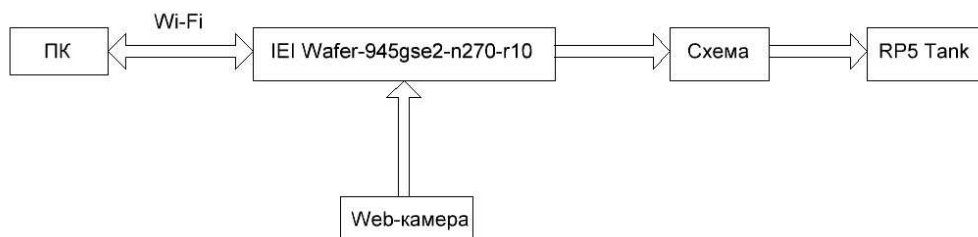


Рис. 10 Структурная схема системы управления шасси

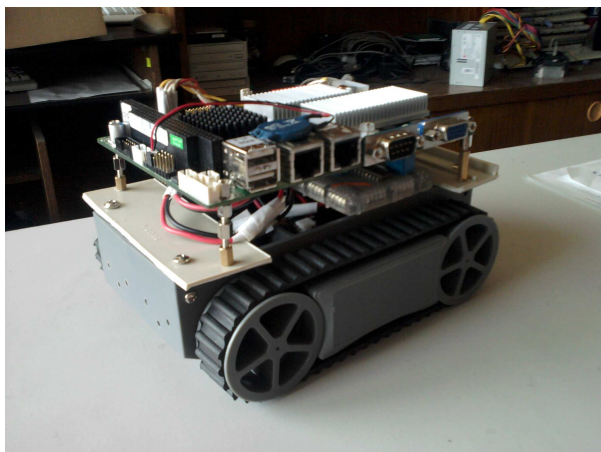


Рис. 11 Макет системы управления шасси со смонтированным одноплатным компьютером и установленной аккумуляторной батареей

Для запуска передачи сигнала необходимо

подключиться по *ssh* к одноплатному компьютеру и запустить приемную часть:

```
ssh -l tc 192.168.1.3
sudo sh
cd /mnt/sda1/
./udpserver
```

А также запустить на ПК программу передачи сигнала из *ScicosLab*, либо простую программу передачи данных с клавиатуры написанную на C.

Для запуска программы управления с помощью клавиатуры на ПК:

```
./udpclient 192.168.1.3
```

Управление происходит с помощью клавиш *W*(вперед), *S*(назад), *A*(влево), *D*(вправо), пробел (стоп).

**ffserver Status**

Available Streams

Path	Served Conns	bytes	Format	Bit rate kbits/s	Video kbits/s	Audio kbits/s	Feed
<a href="#">webcam.flv</a>	0	0	flv	200	200	flv	webcam.ffm
<a href="#">webcam.swf</a>	0	0	swf	200	200	flv	webcam.ffm
<a href="#">index.html</a>	3	3553	-	-	-	-	-

**Feed webcam.ffm**

Stream type kbits/s codec Parameters  
 0 video 200 flv 320x240, q=10-15, fps=30

**Connection Status**

Number of connections: 3 / 1000  
 Bandwidth in use: 0k / 1000k

#	File	IP	Proto	State	Target bits/sec	Actual bits/sec	Bytes transferred
1		192.168.1.3		HTTP_WAIT_REQUEST	0	0	0
2	index.html	192.168.1.3	HTTP/1.1	HTTP_WAIT_REQUEST	0	0	0
3	webcam.ffm(input)	127.0.0.1	HTTP/1.1	RECEIVE_DATA	200k	388k	561k

Generated at Wed May 23 05:20:31 2012

Рис. 12 Веб-интерфейс для доступа к Web-камере

#### 4.7 Исследование работоспособности системы

Сети *Wi-Fi* (*IEEE 802.11*) могут работать в двух основных режимах:

- режим точки доступа;
- режим *Ad-hoc*.

В режиме маршрутизации вся информация передается через точку доступа-маршрутизатор. В результате топология сети приобретает

конфигурацию типа «звезда».

*IEEE 802.11*— набор стандартов связи, для коммуникации в беспроводной локальной сетевой зоне частотных диапазонов 2,4; 3,6 и 5 ГГц.

В настоящее время существует два основных стандарта *Wi-Fi*:

- *802.11n* – 150Мбит/с, 2,4—2,5 или 5,0 ГГц;

© **АВТОМАТИКА И ПРОГРАММНАЯ ИНЖЕНЕРИЯ. 2012, №1(1)**

○ 802.11g — 54 Мбит/с, 2,4 ГГц стандарт (обратная совместимость с b);

○ 802.11b — Улучшения к 802.11 для поддержки 5,5 и 11 Мбит/с.

Среди перечисленных стандартов сейчас наиболее распространен стандарт 802.11g ввиду массовой доступности сетевых карт и чипсетов, поддерживающих его. Стандарт 802.11n только приобретает свою популярность, но у него есть очень много преимуществ, среди которых:

○ высокая скорость работы (150 Мбит/с по одной антенне, 300Мбит/с по двум антеннам);

○ два диапазона работы (2,4МГц и 5МГц) позволяют уйти от помех.

Сети *Ad-hoc* – децентрализованные беспроводные самоорганизующиеся сети, не имеющие постоянной структуры. Клиентские устройства соединяются «на лету», образуя собой неравномерную сеть. Каждый узел сети пытается переслать данные, предназначенные другим

узлам. При этом определение того, какому узлу пересылать данные, производится динамически, на основании связности сети. Это является отличием от проводных сетей и управляемых беспроводных сетей, в которых задачу управления потоками данных выполняют маршрутизаторы (в проводных сетях) или точки доступа (в управляемых беспроводных сетях).

В *Таблицах 6 и 7* представлены данные по тестированию функционирования работы системы управления шасси при свободном и загруженном канале.

Как видно из *Таблиц 6 и 7*, скорость работы в режиме *ad-hoc* немного выше, чем в централизованном виде связи. Количество потерянных пакетов приблизительно одинаковое в обоих режимах. Потери при средней загруженности канала, около 5% пакетов. Задержки при передаче пакетов незначительны и не вносят помех в процесс управления.

*Таблица 6*

**Тестирование без загрузки канала**

Вид беспроводной связи	<i>Ad-hoc</i>		<i>IEEE 802.11 g</i>		<i>IEEE 802.11 b</i>	
	Общее количество посланных пакетов	Количество потерянных пакетов	Общее количество посланных пакетов	Количество потерянных пакетов	Общее количество посланных пакетов	Количество потерянных пакетов
Общее количество посланных пакетов	200	400	200	400	200	400
Количество потерянных пакетов	2-10	10-20	2-10	10-20	30-40	60-80

*Таблица 7*

**Тестирование при средней загруженности канала**

Вид беспроводной связи	<i>Ad-hoc</i>		<i>IEEE 802.11g</i>		<i>IEEE 802.11 b</i>	
	Общее количество посланных пакетов	Количество потерянных пакетов	Общее количество посланных пакетов	Количество потерянных пакетов	Общее количество посланных пакетов	Количество потерянных пакетов
Общее количество посланных пакетов	200	400	200	400	200	400
Количество потерянных пакетов	5-15	15-30	5-15	15-30	40-60	70-90

**ЗАКЛЮЧЕНИЕ**

В данной статье описан процесс сборки встраиваемого варианта операционной системы *Linux* для запуска *SciLab* приложений. Решены задачи обмена данными между приложением *SciLab*, выполняющимся на хост – системе и на объекте. Методы создания встраиваемых операционных систем и приложений для них могут быть рекомендованы для разработки систем управления реальными объектами.

Полученные результаты показывают, что современные свободные математические и инженерные пакеты могут быть успешно использованы для создания систем управления в робототехнике. Наиболее перспективным является применение свободно распространяемого открытого пакета *SciLab*,

который позволяет проводить как моделирование, так и выгружать СИ код модели, открытый к модификации и компиляции. Исходный код хорошо анализируется, обладает интуитивно понятной структурой и легко модифицируется. В данной работе продемонстрирован пример создания аппаратно-программного комплекса управления автономным гусеничным шасси с использованием одноплатного компьютера и встраиваемой операционной системы *Linux*. Полученные результаты позволяют рекомендовать данный метод создания систем управления к использованию при синтезе методов и проектировании сложных систем автоматизации и робототехники.

Работа выполнена по заданию Министерства образования и науки РФ, проекты

## ЛИТЕРАТУРА

[1] А.Б. Колкер, Д.А. Ливенец, Кошелева А.И. Обоснование выбора программного обеспечения для робототехники. Настоящий журнал «Автоматика и программная инженерия».

[2] User manual [Электронный ресурс] - URL: [http://www.ieiworld.com/files/file\\_pool/0B329000195966240272/file/WAFER-945GSE2\\_UMN\\_v2.00.pdf](http://www.ieiworld.com/files/file_pool/0B329000195966240272/file/WAFER-945GSE2_UMN_v2.00.pdf)

[3] RTAI - Real Time Application Interface Official Website [Электронный ресурс] - URL: <https://www.rtai.org/>

[4] Scicos: Block diagram modeler/simulator [Электронный ресурс] – URL: <http://scicos.org/>

[5] The Free Platform for Numerical Computation [Электронный ресурс] - 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 – 700 с.

[7] Nikoukhah R., Steer S. SCICOS - A Dynamic System Builder and Simulator. 1998 – 15 с.

[8] Modeling and Simulation in Scilab/Scicos with ScicosLab 4.4 Справочное пособие / Stephen L. Campbell и др. – Springer, 2010, 332 с.

[9] Campbell S. L. Modeling and Simulation in Scilab/Scicos /S. L. Campbell, J.P. Chancelier, R. Nikoukhah – NY 10013, USA: Springer Science , 2006. – 309с.

[10] Галынский В.М. Свободно распространяемые системы компьютерной алгебры и возможности их применения в образовании / В.М. Галынский // Белорусский Государственный Университет Минск, Беларусь. – 2006. – 4с.

[11] W. R. Stevens, S. A. Rago. Advanced Programming in the UNIX® Environment: Second Edition / Addison Wesley Professional, 2005 – 960 с.

[12] Стивенс У. UNIX: Разработка сетевых приложений / СПб.: Питер, 2007 – 1040 с.

[13] Марк Дж. Рочкинд. Программирование для UNIX. Наиболее полное руководство Advanced UNIX Programming / БХВ-Петербург, Русская Редакция, 2005 г. – 704 с.

[14] Н. Мэтью, Р. Стоунс. Основы программирования в Linux / БХВ-Петербург, 2009 г.

[15] SciLab refence manual. Scilab Group INRIA Meta2 Project/ENPC Cergrene. - Domaine de Voluceau – Rocquencourt: - BP 105 - 78153 Le Chesnay Cedex (France).

2001 – 700 с.

[16] Nikoukhah R., Steer S. SCICOS. A Dynamic System Builder and Simulator, 1998 15 с.

[17] Modeling and Simulation in Scilab/Scicos with ScicosLab 4.4 Справочное пособие / Stephen L. Campbell и др. – Springer, 2010, 332 с.

[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.[Электронный ресурс] URL: <http://www.chipinfo.ru/dsheets/ic/155/ln1.html>

[21] DatasheetCatalog.com – свободный онлайн-справочник электронных компонентов и полупроводников [Электронный ресурс] 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.[Электронный ресурс] URL: [http://www.ieiworld.com/product\\_groups/industrial/content.aspx?gid=00001000010000000001&cid=09050665109937740140&id=08323535591322447606](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.[Электронный ресурс] URL: [http://www.advantech.com/products/PCM-4153/mod\\_A8209479-19F4-47AD-8446-84FBD131D005.aspx](http://www.advantech.com/products/PCM-4153/mod_A8209479-19F4-47AD-8446-84FBD131D005.aspx)

[24] DFRobot-An Online Opensource Robot [Электронный ресурс] 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](mailto:fiery77@yandex.ru)

*Ливенец Дмитрий Александрович* – магистр, окончил кафедру Автоматики НГТУ,

**e-mail:** [dmitry.livenets@gmail.com](mailto:dmitry.livenets@gmail.com)

*Кошелева Алёна Игоревна* – магистр, окончила кафедру Автоматики НГТУ,

**e-mail:** [kosheleva.alyna@mail.ru](mailto:kosheleva.alyna@mail.ru)

*Жмудь Вадим Аркадьевич* – заведующий кафедрой Автоматики НГТУ, д.т.н.,

**e-mail:** [oao\\_nips@bk.ru](mailto:oao_nips@bk.ru)

**III**  
**Инновационные технологии**  
**Перспективные технологии**  
**Моделирование систем**  
**Облачные технологии**  
**Национальная программная**  
**платформа**



**Содержание раздела:**

- А.В. Ляпидевский, В.А. Жмудь, А.М. Подолец.  
Преимущества свободного программного обеспечения для стратегических технологий. С. 101–106.

# Преимущества свободного программного обеспечения для стратегических технологий

А.В. Ляпидевский, В.А. Жмудь, А.М. Подолец

**Аннотация:** Обсуждаются преимущества перехода на свободное и открытое программное обеспечение для стратегических технологий на примере задач робототехники и задач управления организацией.

**Ключевые слова:** свободное программное обеспечение, открытое программное обеспечение, центр компетенции, автоматика, робототехника, стратегические технологии

## ВВЕДЕНИЕ

Переход на свободное и открытое программное обеспечение связан не только с желанием экономии затрат на программные средства. Этот вопрос является ключевым для решения важнейших стратегических задач, поскольку только открытость кода может гарантировать отсутствие вредоносных «закладок» в программе. Вместе с тем отсутствие необходимости оплаты за использование программных средств также может служить существенным основанием для отказа от использования дорогостоящей программной продукции. Термины «Открытое программное обеспечение» и «Свободное программное обеспечение» (СПО) не являются синонимами, однако, существенная часть программного обеспечения является одновременно и свободным и открытым, что является дополнительным аргументом для его использования. В статье обсуждаются перспективы развития СПО для ключевых стратегических технологий, а также для учебных целей и целей управления организациями.

## 1. ЦЕНТРЫ КОМПЕТЕНЦИИ «СВОБОДНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ»

24 декабря 2010 года в ОАО «НИПС» был открыт первый отечественный Центра компетенции «Свободного программного обеспечения» (ЦК «СПО»), в связи с чем прошла пресс-конференция с выступлением координатора Национальной программной платформы, доктора технических наук, генерального директора ОАО «Концерн «Сириус» Леонида Михайловича Ухлинова [1]. В пресс-конференции также приняли участие: Ольга Вячеславовна Иванова, заместитель генерального директора ОАО «Концерн «Сириус» по маркетингу и сбыту; Александр Валерьевич Ляпидевский, директор ОАО «НИПС», кандидат экономических наук; Вадим Аркадьевич Жмудь, доктор технических наук, профессор (см. *Фото 1 - 2*).



Фото 1. Л.М. Ухлинов на открытии ЦК «СПО»



Фото 2. Открытие ЦК «СПО» при ОАО «НИПС»

В своей речи Л.М. Ухлинов отметил, что более 80% программного обеспечения, используемого в России, – импортного производства и охарактеризовал эту ситуацию как ненормальную во многих отношениях. «Во-первых, российские предприятия и госструктуры ежегодно отчисляют десятки миллиардов рублей лицензионных платежей за пользование ПО, во-вторых, это создает угрозу в плане информационной безопасности, в-третьих, ведёт к «утечке» за рубеж наших высококвалифицированных программистов».

Понятно, что такая ситуация нетерпима для России. В связи с этим возникает задача – создание Национальной программной платформы на основе СПО и отечественного проприетарного ПО, что приведёт к повышению уровня грамотности в области использования ИТ, стимулированию внедрения решений, входящих в НПП, в госсекторе и бизнесе, развитию и созданию новых

отечественных коллективов и научных школ в сфере ИТ. Координатором технологической платформы «Национальная программная платформа» выступает Государственная Корпорация «Ростехнологии» в лице Концерна «Сириус». В настоящее время порядка 90 организаций подали заявки на участие в НПП, это – НИИ, конструкторские организации, производственные, сервисные компании, вузы. Для успешной реализации Национальной программной платформы необходимо создание федеральной сети Центров компетенции СПО. В ближайшее время Концерн «Сириус» предполагает открытие таких центров в Санкт-Петербурге, Самаре, Пензе и других городах России, а первый наш Центр компетенции будет открыт на базе Новосибирского института программных систем.

ОАО «НИПС» рассматривает развитие свободного программного обеспечения в качестве одного из приоритетов своей деятельности. По этому направлению осуществляется сотрудничество со стратегическим партнёром – компанией «АльтЛинукс». Тематический семинар в очередной раз выявил, с одной стороны, большой интерес к продуктам и сервисам СПО у научных, учебных организаций, а также инфраструктур региона, а с другой – отсутствие в Сибирском регионе достаточно серьёзной организационной, инфраструктурной и методической поддержки продуктов СПО. Это подтверждает актуальность проекта «Национальная программная платформа» и конкретно такого его аспекта, как формирование федеральной сети Центров Компетенции СПО.

Впоследствии были открыты центры компетенций аналогичного направления на базе других предприятий, входящих в ОАО «Концерн «Сириус», в том числе в Санкт-Петербурге на базе ОАО «Научно-исследовательский институт программных средств» [2, 3], в Кирове, Самаре, Пензе и других городах России. Целью открытия этих центров компетенции является развитие информационных технологий России путем реализации Национальной программной платформы.

## 2. ПРЕИМУЩЕСТВА СПО ДЛЯ УЧЕБНЫХ ЦЕЛЕЙ

В статьях [4–7] описаны результаты работ, выполненных с целью создания учебных стендов по робототехнике и автоматике. В статье [4] дан детальный анализ популярных пакетов программного обеспечения, которые потенциально могут быть использованы для управления робототехническими стендами. Выбор свободного программного обеспечения обоснован не только по экономическим соображениям, но на основе комплексного анализа возможностей и удобства использования различных пакетов, хотя стоимостные показатели, разумеется, также занимают существенное место в анализе. Авторами проанализированы следующие продукты: *MatLab*, *Octave*, *Macsyma*, *MathCad*, *SciLab*, *LabVIEW*, сделан

выбор в пользу *SciLab*. Указанный продукт позволяет проводить моделирование, выгружать СИ код модели, открытый к модификации и компиляции. Авторы отмечают, что «исходный код хорошо анализируется, обладает интуитивно понятной структурой и легко модифицируется».

Положительные результаты исследований, полученные на основании этого выбора [7], позволяют признать его правильность.

## 3. ПРЕИМУЩЕСТВА СПО ДЛЯ ЦЕЛЕЙ УПРАВЛЕНИЯ ПРЕДПРИЯТИЕМ

Основная проблема, связанная с использованием информационных технологий на предприятии – это высокая стоимость комплексных решений на базе проприетарных решений и низкий уровень квалификации ИТ-специалистов и пользователей. Среди СПО-решений существуют достаточно проработанные и надежные системы и приложения, которые в комплексе решают первую задачу. Но чаще всего внедрение СПО на предприятии оказывается не успешным из-за отсутствия необходимых знаний специалистов и пользователей при работе с компьютерами, операционными системами и программами. Для решения этой задачи появляется необходимость использовать в составе комплексного решения такое СПО, которое включает возможность работать пользователю в привычной для него среде, например, использовать ПО внешне похожее на проприетарные продукты компании Майкрософт. Также можно предложить использовать методы быстрого обучения внутри программ, которые пользователям совершенно не знакомы, и использовать современные возможности интерфейсов и предоставления ПО.

Стандартизированные сервисы, такие как: электронная почта, система мгновенных сообщений, файловое хранилище, электронный документооборот, системы безопасности, уже давно используют полностью или в своем составе СПО для повышения надежности и качества работы. Способ предоставления этих сервисов в последние годы изменился и прошел путь от устанавливаемых приложений до «облачных» решений. Тенденции современности – предоставление любого типа программного обеспечения через веб-браузер ОС, причем самой массовой ОС становится СПО *Android* на базе *GNU/Linux*.

Сообщество разработчиков СПО уже учитывает эти тенденции, и каждый день создает новые и новые программы, позволяющие по-новому взглянуть на управление ИТ-инфраструктурой и на работу за компьютером. От того, как построена ИТ-структура предприятия зависит эффективность работы её бизнес-процессов и коммуникаций между сотрудниками и клиентами. Современные ИТ-инфраструктуры, кроме стандартизированных функций, имеют в наличии дополнительные сервисы, позволяющие расширить возможности

компании, используя новые принципы коммуникаций и обмена информацией. Системы, построенные на веб-технологиях и «облачных» вычислениях, предоставляют руководителям предприятий возможность использовать все имеющиеся мобильные устройства для работы в приложениях, корпоративного общения и эффективного управления своими задачами или задачами своих сотрудников. Широкое развитие социальных сетей привело крупные корпорации [8] к выводу: а) в корпоративной среде эффективно использовать социальные принципы общения (*social work*); б) информационные системы должны быть построены максимально приближенными к популярным социальным сетям, в которых большая часть сотрудников проводит в повседневности большую часть своего рабочего времени.

В настоящее время в СПО существуют решения обладающими возможностями социальных сетей, включающие в себя набор стандартизированных сервисов и позволяющие быстро создать для предприятия эффективную среду для управления и коммуникаций. Качество СПО повышается с каждым днем, поскольку разработкой занимаются сотни тысяч людей по всему миру. Такую возможность им предоставляет открытый код и свободная лицензия, основанная на **четырёх степенях свободы** [9]. Свобода номер ноль – это свобода запускать программу так, как вам этого хочется. Свобода номер один – это свобода изучать исходный код программы и изменять его так, чтобы программа работала в соответствии с вашими пожеланиями. Свобода номер два – это свобода помогать остальным, свобода распространять точные копии программы среди других пользователей, свобода номер три – это свобода вносить вклад в сообщество. Это свобода распространять модифицированные копии программ, если вам этого хочется. Если у пользователя программы есть все четыре свободы, программу можно отнести к свободному программному обеспечению. Свободное ПО – это общественный строй, который определяет, как распространять и использовать программы. Если общество использует свободное ПО – это путь вперед, так как свободная программа – это знание. Пользователи могут постигнуть это знание, сохранить его, изменить, дополнить, а также применять любыми иными способами. Если общество использует проприетарное ПО, движение вперед зависит от его платежеспособности в будущем. Проприетарное ПО – это также зависимость от того или иного лица или компании, это социальная проблема, и наша цель – ее устранить. Написать свободную программу означает принести пользу обществу [9].

Очевидным преимуществом СПО для использования на отечественных предприятиях – возможность быстро получить квалифицированную поддержку и изменить программу для своих

требований, задействовав специалистов из числа российских разработчиков. Повсеместное использование СПО на предприятиях, популяризация СПО среди отечественных разработчиков позволит повысить уровень квалификации специалистов, создаст тысячи новых рабочих мест и приведет к укреплению позиции Сообщества СПО и создаст массовое движение, направленное на создание новейших систем, программ, комплексов и приложений для управления и коммуникации на предприятиях любого типа. Это позволит быстро заменить устаревшие системы и сократит издержки связанные с их использованием.

#### 4. ПРИМЕР РЕАЛИЗАЦИИ ОФИСНЫХ ФУНКЦИЙ В СПО

Вниманию читателей предлагается ознакомление с возможностями СПО для реализации функций офиса.

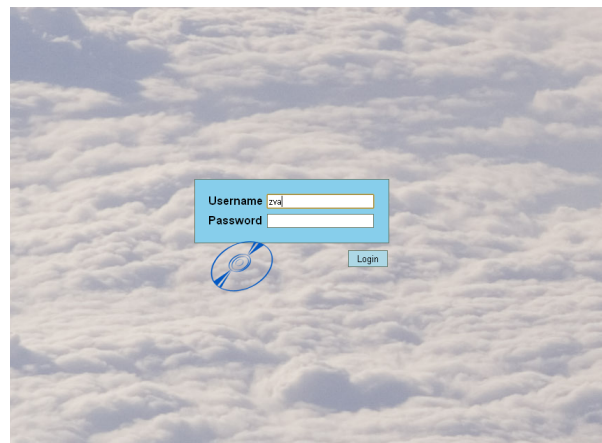


Рис. 1. Авторизированный вход в свободное офис-приложение в концепции, предлагаемой ЦК СПО при ОАО «НИПС»

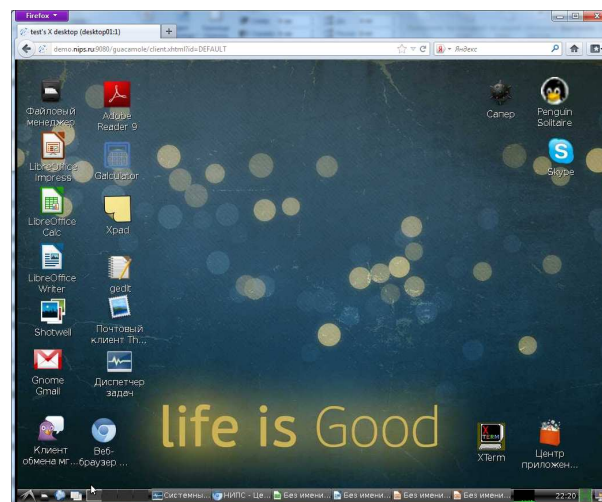


Рис. 2. Фрагмент экрана свободного офис-приложения в концепции, предлагаемой ЦК СПО при ОАО «НИПС»



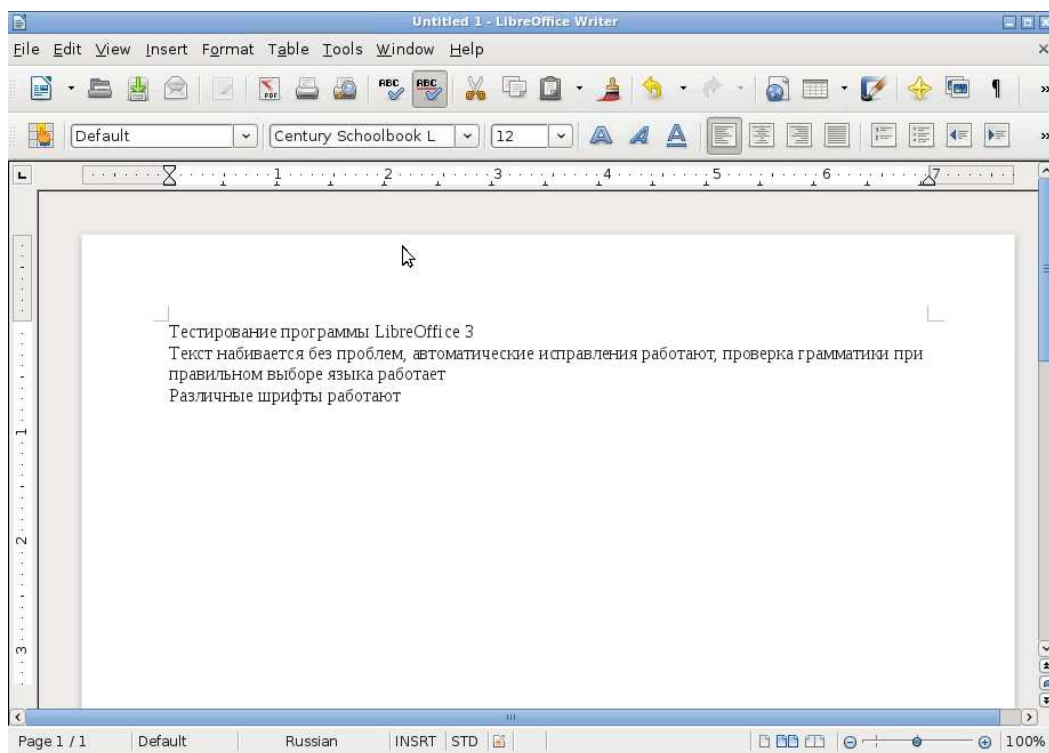


Рис. 3. Подготовка текстового документа программе *LibreOffice 3* в концепции ЦК СПО

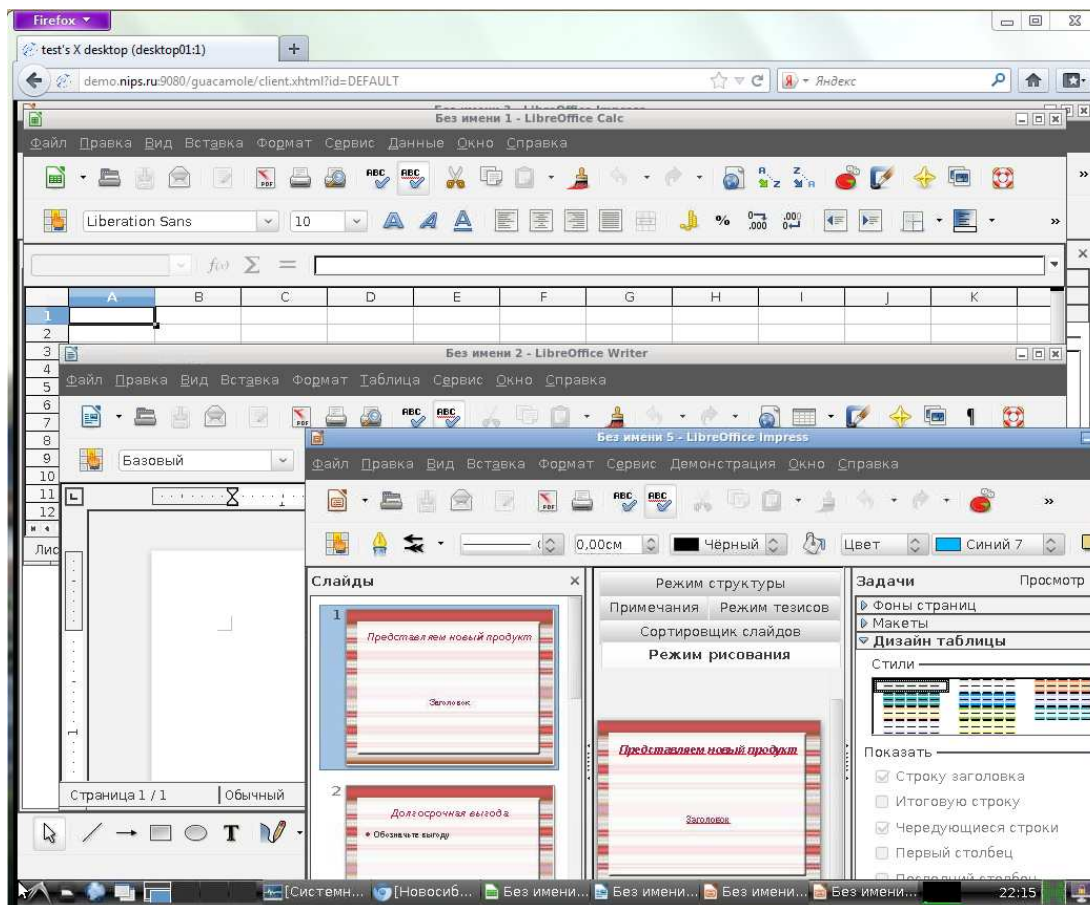


Рис. 4. Подготовка презентации



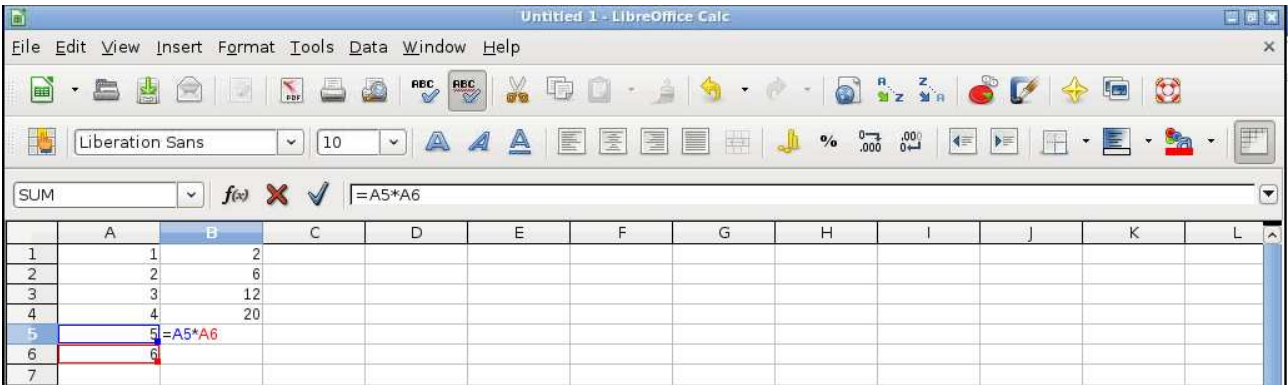


Рис. 5. Электронные таблицы в концепции ЦК СПО

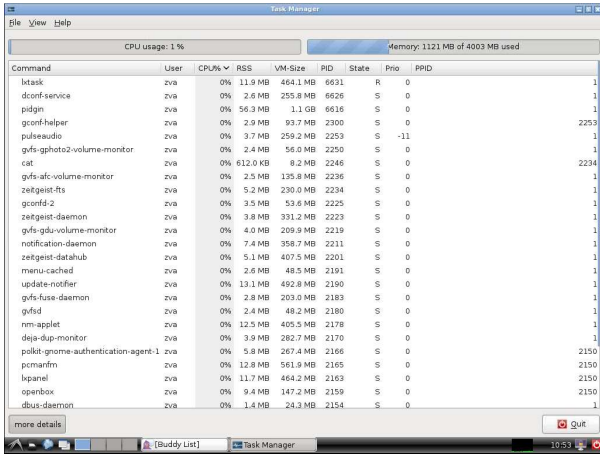


Рис. 6. Мониторинг загрузки процессора

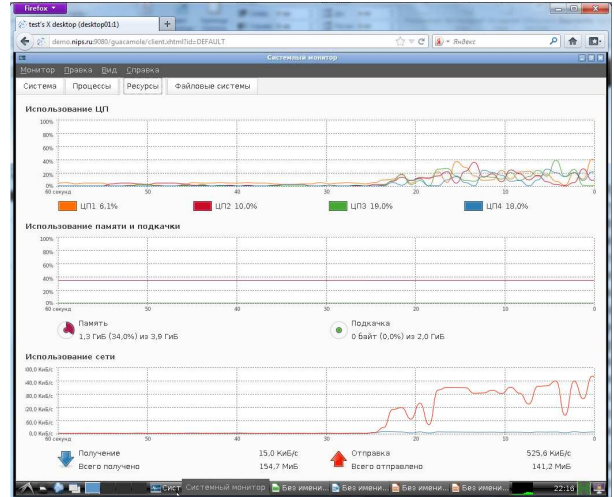


Рис. 9. Мониторинг загрузки системы



Рис. 7. Калькулятор в концепции ЦК СПО

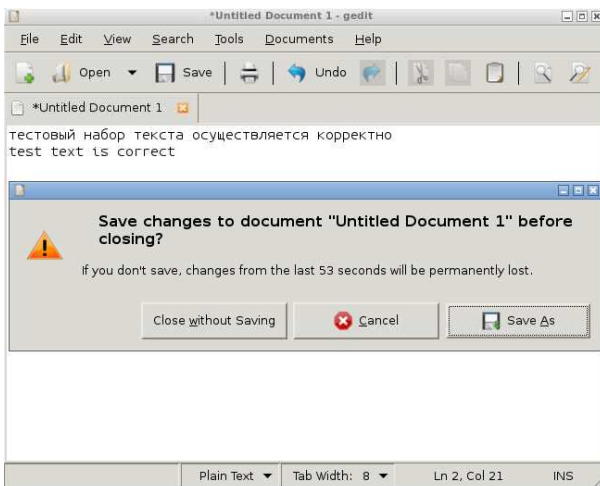


Рис. 8. Текстовый редактор

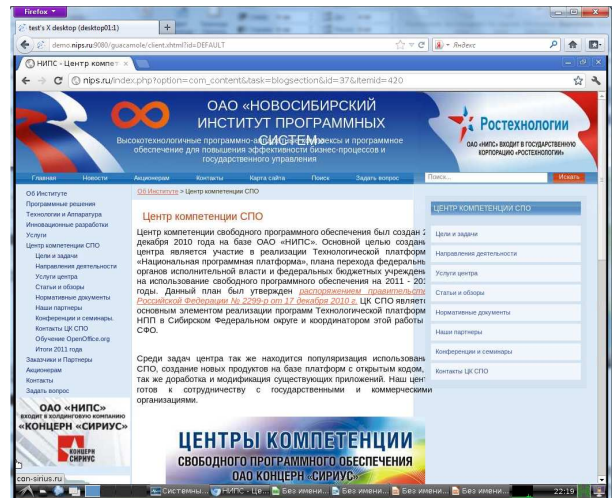


Рис. 10. Выход в интернет

## ЗАКЛЮЧЕНИЕ

Переход на СПО в стратегических направлениях развития для предприятий, отрасли и государства в целом означает освобождение от зависимости от разработчика и собственника. Но это не главное. Исключение возможностей скрытых «закладок», которые могут оказаться вредоносными не обязательно по умыслу, но и по недосмотру разработчика, позволяет не просто выполнять какие-либо функции, но и гарантировать их выполнение. Вредоносные программы, вирусы, сбой

работы и другие проблемы проприетарного ПО в открытых кодах являются меньшей проблемой, поскольку СПО чаще модифицируется, поэтому разработчикам вредоносных программ приходится чаще модифицировать свои «продукты»

Использование ПО, имеющего закрытые фрагменты, можно сравнить с приобретением мебели, в которой часть ящиков закрыты и недоступны владельцу – там может быть скрыто все, что угодно; именно в таких закрытых фрагментах кодов заводятся вирусы, черви и иные вредоносные программы, часть из которых может не быть обнаружено, вследствие их пассивности, фрагменты не препятствуют работе программ, но способствуют дополнительной загрузке памяти, что существенно замедляет работу компьютера.

К сожалению, имеются и существенные проблемы при переходе на СПО, самая очевидная из которых – не полная совместимость форматов документов и специализированный файлов свободного и проприетарного ПО, а также трудности адаптации пользователей к иному принципу взаимодействия с персональным компьютером. Требуется набор таких решений, чтобы обмен был по всем файлам и в обе стороны, иначе одна из сторон может оказаться отсеченной от важных для нее процессов, например, редакция журнала может не принять файл в редакторе, не используемом в данной организации. Совершенно очевидно, что эти проблемы будут обязательно решены.

Первый Центр Компетенции Свободного программного Обеспечения ОАО «Новосибирский Институт Программных Систем» призывает более активно концентрировать и консолидировать усилия на решении этих и других проблем, связанных с более широким внедрением СПО.

## СПИСОК ЛИТЕРАТУРЫ

[1] [http://www.nips.ru/index.php?option=com\\_content&task=view&id=322&Itemid=106](http://www.nips.ru/index.php?option=com_content&task=view&id=322&Itemid=106)

[2] <http://niisvt.ru/component/content/article/news/centre.html>

[3] <http://www.rg.ru/2011/10/21/matrica.html>

[4] А.Б. Колкер, Д.А. Ливенец, Кошелева А.И. Обоснование выбора программного обеспечения для робототехники. // Автоматика и программная инженерия. 2012. № 1(1). С. 48–61.

[5] А.Б. Колкер, Д.А. Ливенец, Кошелева А.И. Разработка системы управления стенда для исследования переходных процессов вакуумного высоковольтного выключателя. // Автоматика и программная инженерия. 2012. № 1(1). С. 62–69.

[6] А.Б. Колкер, Д.А. Ливенец, Кошелева А.И., В.А. Жмудь. Разработка блока системы управления роботом с применением инженерного пакета SciLab. // Автоматика и программная инженерия. 2012. № 1(1). С. 70–80.

[7] А.Б. Колкер, Д.А. Ливенец, Кошелева А.И., В.А. Жмудь. Исследование вариантов создания интеллектуальных систем робототехники на базе одноплатных компьютеров и свободных операционных систем. // Автоматика и программная инженерия. 2012. № 1(1). С. 81–96.

[8] <http://www-01.ibm.com/software/ru/mag/5/article.html>

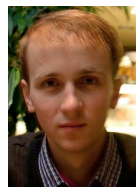
[9] <http://lenta.ru/articles/2011/12/03/stallman/>



**Александр Валерьевич Ляпидевский** – директор ОАО «Новосибирский институт программных систем», кандидат экономических наук, автор более 30 научных статей, включая 2 патента. Область научных интересов: программные средства и системы, информационные технологии, Национальная программная платформа, экономика, инновации.  
E-mail: [nips@nips.ru](mailto:nips@nips.ru)



**Вадим Аркадьевич Жмудь** – заведующий кафедрой Автоматики в НГТУ, профессор, доктор технических наук, автор более 200 научных статей. Область научных интересов и компетенций – теория автоматического управления, электроника, лазерные системы, оптимизация, измерительная техника.  
E-mail: [oao\\_nips@bk.ru](mailto:oao_nips@bk.ru)



**Андрей Михайлович Подолец** – Руководитель первого Центра Компетенции СПО ОАО «НИПС». Область научных интересов и компетенции – информационные системы, СПО, биомедицина, робототехника.  
E-mail: [a.podolets@spo.nips.ru](mailto:a.podolets@spo.nips.ru)

**IV**  
**Популярные статьи**  
**Введение в специальность**  
**К сведению авторов**

### Содержание раздела:

- В.А. Жмудь. Апология теории автоматического управления. С. 109–133.
- В.А. Жмудь. Будущее программы двойных магистерских дипломов по автоматике: взаимодействие НГТУ с вузами России, Украины, Болгарии, Чехии и Франции. С. 134–137.
- Требования к публикациям в научном электронном журнале «Автоматика и программная инженерия» С. 138.

# Апология Теории Автоматического Управления

В.А. Жмудь

**Аннотация:** Статья предназначена студентам и абитуриентам, а также всем интересующимся теорией автоматического управления вне рамок основной специальности.

**Ключевые слова:** теория автоматического регулирования, кибернетика

«Рассматривать предметы с разных точек зрения так же хорошо, как и рассматривать их под одним углом зрения, или даже еще лучше, ибо такое рассмотрение шире и полезнее».

*М. Монтень. Опыты.*

«...Так выпьем же за кибернетику!»

*Из к/ф «Кавказская пленница»*

## 1. ПЕРВЫЕ АВТОМАТЫ

«Твои уши всегда указывают тебе правильное направление, но ты всегда должен знать, зачем тебе туда надо».

*Антон Ван Вебер*

С древних времен люди для облегчения своей жизни придумывали не только орудия труда, но и **автоматические приспособления** – такие устройства, которые меняют свое состояние в различных условиях.

Примитивным примером **автомата** может служить захлопывающиеся ловушки, которые

дошли и до наших времен почти в первозданном виде и используются для отлова диких животных. Капкан и мышеловка – миниатюрные аналоги этих устройств. В этих устройствах в зачаточной форме есть все элементы **автоматической системы**. В них есть «**объект**» (скобы капкана или собственно ловушка), «**датчик**», который воспринимает изменения ситуации, в данном случае появление животного (это педаль капкана или крючок с наживкой), и «**регулятор**», который изменяет состояние устройства (в данном случае взведенная пружина).

В древних храмах использовались **автоматические приспособления**, скрытые от глаз прихожан, они помогали жрецам делать предсказания или просто наглядно «подтверждали» контакты жреца с божеством. Например, в одном из античных храмов после зажигания жертвенного огня распахивались двери храма. Секрет дверей состоял в том, что огонь заставлял испаряться воду, которая конденсировалась в другом сосуде, нарушенное равновесие **приводило в действие механизм** открывания дверей, работающий под действием маховиков, раскручивающихся от тросов, на которые подвешивался груз.

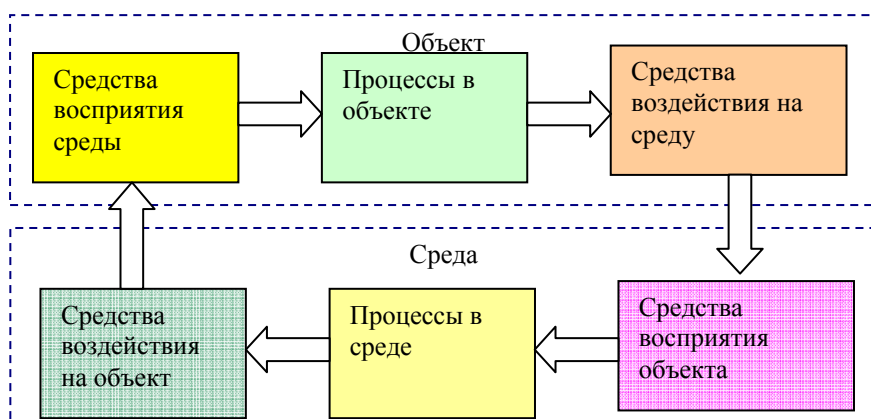


Рис. 1. Типичная обратная связь взаимодействия объекта и среды



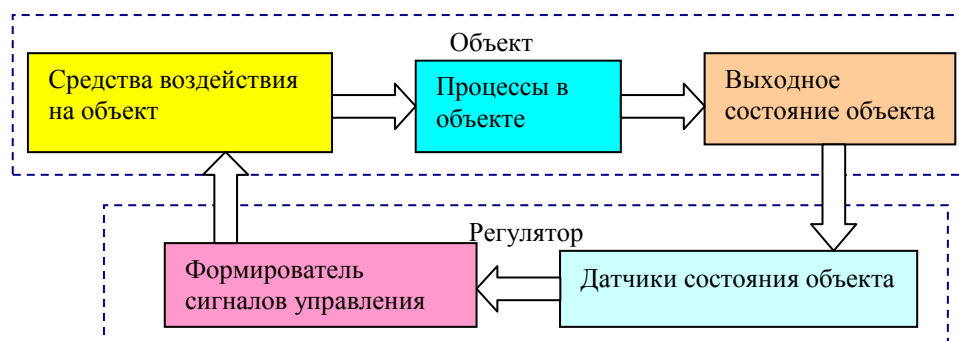


Рис. 2. Типичная структура с обратной связью в автоматических устройствах

В другом храме символ плодородия поднимался в случае влажной погоды и опускался в засуху, чему способствовало изменение веса гигроскопичного вещества на чаше, **приводившей его в движение**: в сухую погоду это вещество высыхало, и вес его уменьшался. Всевозможные плачущие иконы и завывающие по ночам трубы – это тоже **пример устройств, самостоятельно (без участия человека), но вовсе не самопроизвольно изменяющих свое состояние при определенных внешних условиях.**

Благодаря развитию технических средств и орудий труда, в том числе и **автоматических устройств**, человечество вошло в эпоху **развитых технологий**. При этом значение **автоматических устройств** для жизни человека столь сильно возросло, что сегодня мы уже просто не мыслим своей жизни без таких устройств, которые кажутся нам обыденными. Все они используют **основной принцип автоматике**: использование информации о состоянии объекта для **управления** им, то есть так называемый **контур обратной связи с выхода объекта на его вход.**

## 2. АВТОМАТИКА КАК ТАКОВАЯ

«Нет ничего более практичного, чем хорошая теория».

*Густав Роберт Кирхгоф*

Отличительной чертой **автоматических устройств** является использование **отрицательной обратной связи**, то есть такой связи, когда ответное воздействие управляющего устройства на объект противоположно по знаку причине, его вызвавшей. Устройства с однократным действием обратной связи, такие как капканы, ловушки, охранные системы и т.п., собственно к автоматике как к науке относятся весьма отдаленно, поскольку для их разработки не требуется специальный математический аппарат. Более сложные устройства, регулирующие или стабилизирующие одну или несколько величин с высокой точностью, требуют для своей разработки применения

специальных областей прикладной математики, которые и составляют предмет науки **«Автоматика»**, или, как ее раньше называли **«Кибернетика»**.

**Теория автоматического управления** (ТАУ) или регулирования имеет приложения практически во всех отраслях науки и техники, но этого мало: она применима и к анализу экономических, биологических, экологических объектов, живых организмов и их объединений (сообществ, роев, экосистем, биоценозов). Она же необходима для понимания и описания многих химических реакций, физических процессов, климатических, геологических, океанографических и многих других явлений. Многие непонятные (и даже «невозможные») на первый взгляд явления становятся понятными и прогнозируемыми с применением ТАУ.

Ядерный реактор и лазер, система кровообращения и одна-единственная клетка, устойчивость экосистемы и совместная эволюция (коэволюция) животного мира – все эти процессы таят в себе множество загадок. **И разгадать эти загадки без ТАУ невозможно.**

Математический аппарат ТАУ универсален, как универсален в природе **принцип динамической обратной связи** (то есть обратной связи, действующей не мгновенно, а с некоторым **реальным темпом**).

**«Неустойчивость»** некоторых процессов движения к равновесному состоянию – это вместе с тем и замечательная **«устойчивость» периодических колебаний** около этого равновесного состояния. Именно ей мы обязаны тому, что мир не превратился в одну большую «черную дыру», все атомы не слетелись воедино, электроны не упали на ядра своих атомов, мир движется, развивается и существует, а энергия в процессах, его составляющих, переходит из одной формы в другую. **Если бы этого не было**, все процессы завершили бы в своих окончательных равновесных состояниях, **всякому движению пришел бы конец.**

Но этого не происходит: электроны не прекращают свой бег вокруг ядер, расстояние между атомами в веществе неизменно

вследствие равновесия динамических и потенциальных сил притяжения и отталкивания, все живые и неживые системы вовлечены в круговорот. Если в одном месте вода испаряется, то в другом месте она выпадает в виде дождя или снега, если где-то размываются берега, то где-то они намываются, если где-то горы разрушаются, то где-то растут. Все находится в движении, отдельные компоненты этого движения уравнивают друг друга, и в этом – причина относительной стабильности мира. **Понять и описать эти движения помогает ТАУ.** И все же основная область применения ТАУ – **проектирование замкнутых систем управления** для науки, техники, производства, быта и иных применений. Без этой теории даже правильно рассчитанные и изготовленные элементы соединенные воедино не смогут образовать систему с требуемым качеством. Применение же ТАУ позволяет даже с неустойчивыми объектами, даже с недостаточно качественными датчиками и даже с недостаточно надежными регуляторами разработать и изготовить надежные, точные и одновременно быстродействующие системы.

Есть еще такое прикладное научное направление, как «Проектирование надежных систем из ненадежных элементов». Одно только это название – и то уже завораживает, не так ли? Это направление многими разделами пересекается с ТАУ, то есть с Кибернетикой. И оно не могло бы существовать без ТАУ.

### **3. СИСТЕМНЫЙ ПОДХОД К АНАЛИЗУ**

«Там где заканчивается телескоп, начинается микроскоп. Какой из этих приборов имеет больший обзор?»

*Виктор Гюго*

**Системный подход** зачастую определяют просто как «учет всех связей объекта с внешним миром». Это определение не достаточно. В противоположность «классическому подходу», который состоит в мысленном разделении исследуемого объекта на составляющие компоненты для понимания объекта в целом, «системный подход» предполагает применение аппарата теории замкнутых динамических систем, то есть ТАУ.

Дело в том, что просто «учет» внешних связей предполагает всего лишь принятие к сведению совокупность входных и выходных сигналов объекта. Под «сигналами» мы тут понимаем все, что передается от одного элемента к другому и изменяется во времени. Эти сигналы передаются не мгновенно, а за какое-то время. Но даже учет быстродействия прохождения этих сигналов только в пределах рассматриваемого объекта еще не дает системного подхода. При таком подходе

интуитивно можно предположить, что чем выше такое быстродействие, тем, вроде бы, лучше. При системном же подходе зачастую быстродействие какого-то элемента целесообразно снижать (хотя бы частично в области каких-то частот), чтобы получить в итоге лучшую систему – даже и более быстродействующую! Например, если рассмотреть взаимодействие растения с внешней средой, то под учетом внешних условий мы бы вправе были понимать учет климатических условий, в которых данное растение произрастает, а под влиянием его на среду – то, что данное растение производит. Если действие растения на среду столь несущественно, что среда может рассматриваться вне зависимости от деятельности растения, то системный подход не требуется. Совсем иная ситуация возникла бы, если, например, рассматривать такие объекты, как тайга, савана, тундра. В данном случае не только климат влияет на состояние такого объекта, но и объект влияет на климат. При этом важно, насколько быстро объект откликается на изменения в климате, и насколько быстро новые изменения в климате возникнут в результате изменения объекта. При определенных соотношениях этих быстродействий система является устойчивой, саморегулирующейся. При других соотношениях система может стать неустойчивой, саморазрушающейся. Поэтому **системный подход предполагает отыскание в системах петель взаимного влияния, определение динамических свойств элементов в этих петлях и с помощью ТАУ анализ устойчивости и качества переходных процессов в этих системах**, и чаще всего корректирование динамических свойств одного из элементов для получения требуемых свойств системы в целом. Элемент, динамические свойства которого мы можем легко корректировать в любую сторону, как раз и называют «регулятором».

### **4. ЗАДАЧИ ПСИХОЛОГИИ**

«Иногда стоит узнать человека поближе, как хочется послать его подальше».

*По К.С. Мелихану*

«И не следует ставить мне в укор...».

*М. Монтень. Опыты*

Не так уж много в мире объектов, которые бы не имели никакого отношения к замкнутым динамическим системам. Честно говоря, таких объектов вовсе нет. Просто некоторые объекты в некоторых условиях при решении некоторых задач можно рассматривать без использования ТАУ. Но чем сложнее задача, условия или объект, тем острее потребность в аппарате ТАУ, и тем пагубнее оказываются ошибки анализа

или вычислений, выполненных без использования аппарата ТАУ.

Природа, техника, наша жизнь во всех ее аспектах просто наполнена замкнутыми динамическими системами различной степени сложности. Одной из таких систем являемся мы сами, но наряду с этим наш организм содержит множество подсистем, работающих на этом же принципе. При общении с внешним миром мы являемся элементами таких же систем, и зачастую наше поведение определяется именно этой нашей ролью в системе значительно сильнее, чем нашей собственной волей. Так два человека могут образовать систему, свойства которой будут определяться не свойствами каждого человека в отдельности, а именно сочетанием этих свойств этих людей как элементов замкнутой динамической системы.

У Станислава Лема в романе «Голос неба» есть такая фраза: «...По существу никто из специалистов не мог примириться с тем, что **великую Тайну Человека, необъяснимые черты его природы можно вывести из общей теории автоматического регулирования**». Слова великого писателя-фантаста оказались пророческими. Конечно, не в том смысле, что всякое поведение человека может быть выведено однозначно из ТАУ. Но поведение человека, когда он является частью замкнутой динамической системы, а также в том случае, когда речь идет о его функциях, которые реализуются внутри такой системы, эти слова полностью подтверждает.

Рассмотрим общение двух человек. Все произносимые нами фразы и выполняемые (или не выполняемые) нами действия можно по степени воздействия на настроение другого человека (партнера) разделить на три группы: отрицательные, нейтральные и положительные. Если хотя бы один партнер не может ответно воздействовать на настроение другого, то контура с обратной связью не возникает. Препятствия могут быть дистанционные или иерархические. Например, в случае восприятия другого человека по телевизору, или в качестве непрерывающего руководителя контура обратной связи нет, а есть только причинно-следственная связь в единственном направлении.

Если же партнеры оба могут воздействовать на настроение друг друга, то это воздействие, разумеется, зависит от полученного со стороны партнера предшествующего воздействия, и оно же повлияет на последующее воздействие. Налицо **контур обратной связи «действие» - «реакция» - «последствие»**. Этот **замкнутый контур** может передавать как положительное, так и отрицательное воздействие. Термин «положительная обратная связь» или «отрицательная обратная связь» указывает отнюдь не на характер воздействия

(т.е. не на улучшение или ухудшение настроения) а на совпадение или несовпадение знаков приращений действия и последствия.

**Отрицательная обратная связь** означает, что любая помеха, возникшая в контуре, порождает в нем такой отклик, который **возвращается в точку ее возникновения с противоположным знаком**. Следовательно, в контуре с отрицательной обратной связью действие контура стремится ослабить результат действия помехи. Это справедливо лишь в случае устойчивости этого контура.

**Положительная обратная связь** означает, что любая помеха, возникшая в контуре, порождает в нем такой отклик, который **возвращается в точку ее возникновения с тем же знаком**. Следовательно, в контуре с положительной обратной связью действие контура стремится усилить результат действия помехи. Но, поскольку результат усиления также усиливается, и так далее, и если усиление достаточно большое (больше единицы), то получается, что любая сколь угодно малая помеха в контуре с положительной обратной связью может привести к сколь угодно большим последствиям. Нам знакомы (хотя бы теоретически) такие системы на примере цепных реакций, лазеров и вообще большинства генераторов.

В межличностных отношениях обратная связь чаще всего имеет положительный знак, то есть характер воздействия совпадает с характером реакции. Если нам настроение подняли, то и мы стремимся его ответно поднять, если же нам его испортили, то испортим и мы. Пример отрицательной обратной связи в межличностных отношениях дан в миниатюре М.М. Жванецкого, бичующей некоторый сорт людей: «Когда мы с ними по-людски, они с нами по-хамски, а когда мы с ними по-хамски, они с нами по-людски».

Из ТАУ известно, что системы с положительной обратной связью устойчивы только в том случае, если общий коэффициент усиления воздействия по всему контуру меньше единицы. То есть результат от воздействия, внесенного в контур, слабее, чем это само воздействие. Именно поэтому мы чаще всего не ссоримся с малознакомыми или малоинтересными для нас людьми. А если и ссоримся, то не сильно: «Гав!» - «Гав!» - и разошлись. Если же коэффициент будет больше единицы, то система с положительной обратной связью всегда неустойчива. Действие такой системы состоит в лавинообразно нарастающей приязни или неприязни. Коль любить – так до могилы, коль воевать, так уж насмерть! Кстати, само слово «лавинообразно» также указывает на одно из неустойчивых систем с положительной обратной связью – на снежную лавину. В снежной лавине также сдвиг небольшого

количества снега вызывает изменения условий, которые приводят к еще большему сдвигу еще большей массы снега. Коэффициент больше единицы, обратная связь положительна, такая система неустойчива.

Неустойчивость психологической системы из двух человек не всегда вызывает ссору этих людей. Неустойчивость означает лишь движение в одну сторону в эмоциональном восприятии личностями друг друга. Такое движение может реализоваться не только как возрастание неприязни, но и как нарастание положительных эмоций, в частности, возникновении дружбы или любви. Психологические условия в заданном сочетании двух людей одинаково неблагоприятны или благоприятны как для дружбы или любви, так и для ненависти. Одна и та же пара может достичь обожания или ненависти по тем же самым причинам. Недаром говорят: «От любви до ненависти – один шаг». Так Кукушка и Петух из одноименной басни И.А. Крылова мгновенно достигают насыщения поощрениями, так как в ответ на похвалу стараются похвалить друг друга еще сильнее. Гоголевские Иван Иванович и Иван Никифорович, напротив, от пустяка доходят до смертельной ненависти, поскольку каждый раз в ответ на обиду каждый из них стремится нанести обиду еще более сильную. Если же реакция двух партнеров различна, то система может оставаться устойчивой. Поэтому пара холерика и флегматика во взаимоотношениях не подвергается таким стрессам, в такой паре не возникает ни обожания, ни ненависти. Недаром известно, что иногда противоположные черты характера являются причиной длительных отношений. Из двоих ссорящихся кто-то один должен быть спокойней, выдержанней, умнее. Тогда есть шанс, что ссора затихнет скорее. Если же выдержанным окажутся оба в равной степени, то просто ссора растянется во времени надолго, а по сути это также будет неустойчивая система, как и в том случае, когда оба в равной степени невыдержанны.

## **5. ЗАДАЧИ БИОЛОГИЧЕСКИХ ПРОЦЕССОВ**

«Если вы все время будете двигаться только в одном направлении, мы всегда сможем отгеснить вас туда, где весь ваш здравый смысл не будет иметь никакого значения».

*Из речи Юлия Цезаря к своим сенаторам*

«...Где можно жертвовать другими, Зачем нам жертвовать собой?»

*В. Соловьев. Победителей судят*

В биологических процессах теория автоматического управления проявляется очень

часто. Известно утверждение, что вся эволюция – это решение вопроса, кто кого съест. Добавлю, что этот вопрос никогда не решается в чью-либо пользу окончательно. Всем известен пример, для чего «нужны» хищники. Утверждается, что если хищники перестанут существовать, то начнут размножаться травоядные животные, которые съедят всю траву и, наконец, сами вымрут. Но в таком примитивном понимании в природе процессы идут очень редко – это катастрофа. **Стабилизирующие обратные связи действуют в природе постоянно**, а не время от времени. В данном примере действие их проявляется в том, что небольшое увеличение количества хищников приводит к небольшому снижению количества травоядных животных, в результате возрастает конкуренция за пищу, и количество хищников стабилизируется на том уровне, который необходим для стабильного существования на данном ареале данного биоценоза. Аналогично небольшое уменьшение количества хищников приведет к увеличению травоядных, что будет способствовать росту количества хищников.

По отношению к траве и другим кормам травоядные задействованы в аналогичной петле авторегулирования, и играют роль хищников. Трава играет сходную роль по отношению к первичным ресурсам почвы. А почва, со своей стороны, поглощает и траву, и травоядных, и хищников. Таким образом, **имеют место множество петель, множество контуров, в которых отдельные параметры стабилизируются**. Неустойчивость этих контуров приводит к экологическим катастрофам. Здесь слово «катастрофа» принято не в качестве показателя масштаба бедствия, а в качестве граничного состояния системы автоматического регулирования, в которое она попадает при нарушении устойчивости.

Но и сами катастрофы могут являться частью устойчивого контура в более глобальном смысле. Так, например, лесной пожар, несомненно, является катастрофой, но с позиции общей эволюции планеты пожар леса может явиться необходимой частью преобразования лиственного леса в хвойную тайгу, поскольку хвойные деревья меньше страдают от пожара, более приспособлены к преодолению его последствий.

Неожиданно резкие изменения животного мира происходят на исторических отрезках так «быстро», что ученые постоянно ищут внешнюю причину. Традиционно наука движителем эволюции считает случайную изменчивость (мутации) у отдельных особей. Тогда приходится признать глобальное несоответствие между количеством закрепившихся мутаций и вероятностью возникновения таких изменчивостей в природе. Количество видов на Земле поражает



воображение. Каждый вид отличается неисчислимым видом признаков. Эти признаки зафиксированы в генных наборах. Все наборы генов, соответствующие реально существующим видам, целесообразны. Но далеко не все варианты наборов генов целесообразны. Для формирования генных наборов тех видов, которые живут или жили на планете Земля методом «случайного перебора» потребовалось бы чрезмерно много таких «пробных реализаций», шансы на успех такой алгоритм едва ли имел. За время перебора ситуация могла измениться кардинально, нужное решение эволюции пришло бы с опозданием. Поэтому идея случайной изменчивости и естественного (природного) отбора не достаточна для объяснения имеющегося разнообразия видов. У непосвященных в ТАУ исследователей создается мистическое ощущение целенаправленного отбора генов для видообразования. Это – одна из причин успешности религиозного мировоззрения. Необъяснимое несоответствие успешности эволюционных процессов определяется, вероятнее всего, включением механизма положительной обратной связи, который ускоряет мутации (изменчивость) в сложные для вида периоды. В этом случае можно предположить, что небольшое полезное изменение в случае выявления его полезности включает механизм, провоцирующий изменения в этом же направлении, ускоряет движение в этом направлении многократно. Включение механизма отрицательной обратной связи, наоборот, закрепляет найденный генный набор данного вида: отклонения от успешного набора генов снижает жизнеспособность особи, и это обеспечивает стабильность данного генного набора. Ускоренное движение от имеющегося состояния при положительной обратной связи и автоматическая стабилизация имеющегося состояния при отрицательной обратной связи – это те свойства любой части живой природы, которые следует учитывать для более глубокого понимания происходящих в ней процессов.

Американские ученые широко рекламируют теорию катастрофического вымирания динозавров вследствие падения метеорита. Но при ближайшем рассмотрении выясняется, что отрезок времени, на протяжении которого вымирали динозавры, хотя и кажется мгновением на интервале в 150 миллионов лет, все же слишком длинен для того, чтобы быть объясненным падением единственного метеорита – это многие тысячи лет. Да и вымирали динозавры не все сразу, а вид за видом, последовательно. Некоторые животные, являющиеся современниками динозавров, не вымерли вовсе. Следует ли это объяснять целой серией падений метеоритов, с

завидной регулярностью падающих на Землю именно в те моменты, когда подходит пора для вымирания тех или иных видов динозавров, поскольку уже подготовилась почва для новых видов животных? Выходит, не упали эти метеориты в наиболее подходящее время, и не будь они в точности требуемой величины, чтобы с одной стороны не уничтожить все живое, а с другой стороны – устранить динозавров и расчистить путь млекопитающим, так и не было бы эволюции вовсе? От такого предположения недалеко и до идеи о Боге, который очень своевременно устраивает на Земле катастрофы, чтобы развитие жизни на ней шло правильным путем.

**Причина таких резких перемен**, видимо, тоже кроется вовсе не в метеоритах, упавших в нужном месте и в нужное время, а **в положительной обратной связи** при протекании этих процессов. Для этого надо рассматривать всю пищевую цепочку и всю цепочку утилизации туш, а ведь от этих цепочек сохранились только отдельные фрагменты отдельных звеньев. Остальное надо достраивать мысленно, чтобы понять эти процессы. Разумеется, крупные хищники питались более мелкими животными (а многие ученые достаточно основательно полагают, что самые крупные динозавры вообще были не хищниками, а питались падалью). Более мелкие животные питались еще более мелкими, а те в свою очередь – растительной пищей и насекомыми. В основании травоядных животных тоже лежит растительная пища. Следовательно, с одной стороны, история динозавров существенно зависит от истории растительного покрова и насекомых. Но ведь и растения и насекомые тоже развивались не на пустом месте. Для растений условия формируются жизнью животных и насекомых, для насекомых – жизнью растений, животных, птиц. Есть еще морские животные, микроорганизмы и прочее, прочее – все, что влияет на почвообразование, на состав атмосферы, и так далее. Имеет место и конкуренция отдельных видов, занимающих сходное место в пищевых цепях. У млекопитающих и птиц было преимущество: теплокровные животные лучше охотятся ночью и в холодное время года. Появление цветковых растений должно было резко преобразовать мир насекомых, следовательно, это должно было вызвать резкую перестановку сил и во всех остальных пищевых цепочках. То же самое относится и к изменениям в «подстилке» - слою опавших листьев и хвои, из которого насекомые извлекают пищу, превращая его в почву, то есть пищу для растений. Все так сложно, хотя каждый элемент интуитивно понятен. **Без учета множества контуров влияния, обратных связей наряду с прямым воздействием,**



**невозможно верно понять процессы в природе.**

Привлечение ТАУ дает основания утверждать: в животном мире имеют место не только плавные эволюционные изменения, но и резкие, революционные. Эволюция идет устойчиво вследствие действия отрицательных обратных связей, стабилизирующих положение видов и биоценозов. Революция идет лавинообразно вследствие включения положительных обратных связей или вследствие нарушения устойчивости систем с отрицательными связями. **Выбор между эволюцией и революцией осуществляется** не волей высшего существа, а **условиями устойчивости замкнутых динамических систем.** Именно в периоды революций, видимо, происходит видообразование. Разумеется, упавший метеорит может ускорить или стимулировать какой-то процесс, но не более того. Считать метеорит причиной скачка эволюции – столь же наивно, как считать крик животного или человека или падение какого-то камня причиной лавины, селя, обвала. Метеорит или крик – это лишь спусковой крючок, случайная помеха в неустойчивой системе, а причина неустойчивости – накопление в виде потенциала положительных обратных связей, которые реализуются в виде резкого изменения динамического состояния системы.

## **6. ФИЛОСОФСКИЙ АСПЕКТ**

«Непризнание не есть незнание»

*Жан Лакан (фр).*

«Эти методы пригодны лишь для решения относительно простых задач, которые часто могут быть сформулированы лишь благодаря далеко идущей идеализации, иногда настолько далеко, что фактически вместо поставленной задачи решается совсем иная».

*Я. З. Цыпкин. Адаптация и обучение в автоматических системах*

Материалисты учили, что материальное является причиной, а духовное – следствием. В этом они противопоставляли свою позицию идеализму, утверждавшему обратное, считающему духовное начало причиной, а материальные явления – следствием. Философам следовало бы обратиться к примерам из практики, из жизни и из науки. Этот спор именно потому и затянулся, что подтверждающие примеры для своих взглядов находили без труда и материалисты, и идеалисты. И при этом ни те, ни другие не могли свою теорию подвести под все известные явления: слишком узкой оказывалось любое из этих односторонних утверждений. На самом же деле **имеют место оба влияния**, как

материального начала на духовное, так и наоборот. Не только бытие определяет сознание, но и сознание определяет бытие. Не только история творит личности, но и личности творят историю. Не только «в здоровом теле здоровый дух» (точнее, у человека с больным телом меньше шансов на здоровые эмоции), но и здоровый дух может оздоровить тело (как в случае с А.В. Суворовым), а слабый духом человек доведет себя до состояния немощи простой праздностью. В вопросах о божественном или эволюционном происхождении видов споры не умолкают до сих пор. Как бы ни была привлекательна теория эволюции Ч. Дарвина материалистам, слишком много нестыковок имеет место. Невозможно простым перебором и действием природы методом проб и ошибок объяснить столь быстрое видообразование и практически полное отсутствие побочных, неудачных ветвей. Ведь если бы природа действовала перебором, то наряду с существами, приспособленными к изменившимся условиям существования, возникало бы во столько же раз больше неприспособленных существ, во сколько раз больше неудачных комбинаций генов по сравнению с удачными комбинациями. Этого не происходило, видимо, потому, что не только на стадии выбраковки природные условия участвовали в видообразовании, но и на всех стадиях эволюции живых организмов, и сами организмы эволюционировали не изолированно, каждый отдельно, а все вместе как единый биоценоз. Только так можно объяснить существование совершенно далеких видов, не могущих обходиться друг без друга, как, например, некоторые виды муравьев не могут обходиться без некоторых растений, а эти растения не могут обходиться без этих видов муравьев. Примеры такого удивительного содружества находим и в лишайниках, и во многих других группах живых организмов и целых экосистемах. По-видимому, **практически ни один вид не может успешно существовать изолированно от того экологического и биологического окружения, которое вокруг него сложилось**, но и ни один вид не существует без обратного влияния на это окружение. **Все взаимосвязано, и именно взаимно, а не в одном направлении.** Поэтому и в основном вопросе философии давно следует отказаться как от идеализма, так и от материализма, признавая, что **как материальная основа влияет на духовную сущность любого предмета и любой данности, так и обратное влияние обязательно имеет место.** Даже на стадии возникновения нет необходимости выделять что-то одно как причину, а другое – как следствие. Ведь материя никогда не возникала и нигде не исчезнет (вопреки мнению некоторых

ортодоксальных поклонников теории большого взрыва), но материя не могла существовать ни мгновения в отрыве от законов ее существования и взаимодействия, поэтому законы материального мира столь же фундаментальны, как и сама материя. Без материи законы – ничто, но и материя без законов ее существования невозможна. Если же говорить о предметах, возникших когда-то, скажем, в результате эволюции, то и тут имело по всей вероятности место именно одновременное проявление влияния материальной и духовной основы (под духовной основой не имеется в виду ничего божественного, и лишь законы существования и поведения материальной субстанции).

Бессмысленно обсуждать, что произошло раньше – человек стал разумным или его мозг увеличился. Эти **процессы происходили одновременно и влияли друг на друга**. Поэтому на символический (шуточный) вопрос: «Что было раньше, курица или [куриное] яйцо?» следует, видимо, ответить: «Они возникли практически одновременно в сравнении с интервалом движения к этому состоянию. Пока яйцо сносила не курица, он было не куриное, курица только тогда стала курицей в полном смысле, когда начала нести куриные яйца».

**Понятие «первичный» и «вторичный» в философии – не достаточны. Следует дополнительно ввести понятия «элемент замкнутой системы» и «замкнутая система взаимовлияния».** Иными словами, **в философию следует привнести системный подход, используемый в теории автоматического управления.**

## **7. ФИЗИЧЕСКИЕ ЯВЛЕНИЯ**

«Я больше ниспровергал старых истин, чем утверждал новых».

*Станислав Лем. Глас неба.*

«Они, кажется, не столько заботились о достоверности, сколько хотели поупражнять свой ум на трудном предмете».

*М. Монтень. Опыты*

**В отсутствии ТАУ многие вопросы физики были для самих физиков камнем преткновения, и многие из них остаются и до сих пор таковыми, поскольку физики не изучают ТАУ.** Эта теория способна объяснить, почему электрон в атоме, который испытывает только силу электрического притяжения и, казалось бы, должен неминуемо упасть на ядро, все же не только в некоторых случаях не падает, а не падает вообще практически никогда. То есть движение, которое должно закончиться раз и навсегда падением электрона, вместо этого происходит бесконечно, причем без потери энергии. Эта проблема поставила в тупик

абсолютно всех физиков, размышляющих над этим вопросом, самые известные из них – Нильс Бор и Макс Планк. Нильс Бор так и не нашел удовлетворительной модели, описывающий траекторию движения электрона под действием сил притяжения ядра даже для атома водорода, то есть когда электрон всего один. Макс Планк оригинально поступил – он и не стал находить такую модель, а по совету Эйнштейна попросту принял как один из постулатов утверждение, что электрон в атоме не падает на его ядро, а совершает движения по устойчивой стационарной орбите. Таким образом, вместо ответа на вопрос «Почему?» был дан тезис «Так есть!». И хотя все понимают, что не может электрон перестать быть электроном вблизи ядра атома, **все его поведение должно описываться на основе элементарных законов электродинамики**, тем не менее, эта задача теперь не решается, и, видимо, не будет решаться никогда и никем. Однако **с позиции ТАУ эта задача элементарна, и решение ее, соответствующее реальной действительности, практически лежит на поверхности.** Изюминка в том, что при движении электрона к ядру скорость его растет по мере приближения неимоверно, и достигает скорости света (или скажем для скептиков мягче – становится соизмеримой с ней). А в этом случае **запаздывание взаимодействия двух частиц становится существенным.** А при **существенном запаздывании устойчивость движения неминуемо нарушается.** Поэтому если электрон способен разогнаться до релятивистских скоростей, то он неминуемо не упадет на ядро, а будет колебаться около этой конечной цели с некоторой постоянной амплитудой и постоянной частотой – именно с постоянной, поскольку система не линейна. **Это называется автоколебаниями.** Специалисту по ТАУ не надо разъяснять, чем отличаются линейные колебания, амплитуда и частота которых зависит от начальных условий, от нелинейных, которые от начальных условий не зависят. Любой физик, рассуждая о планетарной модели атома, которую предлагал Нильс Бор, скажет, что **эта модель не адекватна действительности**, поскольку при такой модели размеры атома должны были бы зависеть от начальной скорости электрона, и не были бы одинаковыми для одинаковых атомов. Нелинейная же модель, учитывающая квадратичную зависимость силы от расстояния и зависимость силы от скорости (что тоже вполне естественно для взаимодействия, распространяющегося с конечной скоростью) **решение лежит на поверхности, оно очевидно**, и оно именно таково, каково в реальности – **стационарная колебательная система.** Укажем еще на одно важнейшее отличие круговой орбиты от автоколебательной

траектории. В круговой орбите потенциальная энергия объекта приблизительно остается постоянной, и кинетическая – тоже. Действительно, любой естественный или искусственный спутник летит примерно с постоянной скоростью и на примерно одном и том же расстоянии. При автоколебаниях объект «на полной скорости проходит равновесное состояние» и вследствие инерции не остается там, а летит дальше. В момент прохождения этого равновесного состояния потенциальная энергия равна нулю, ведь в этом состоянии все силы уравновесили бы друг друга. Скорость же в этот самый момент максимальна, следовательно, кинетическая энергия тоже максимальна. Далее объект удаляется от равновесного состояния и теряет скорость, поскольку возникают силы, возвращающие его в это состояние. Следовательно, потенциальная энергия растет, а кинетическая – падает. В момент максимального удаления от равновесного состояния потенциальная энергия максимальна, кинетическая же равна нулю, так как скорость объекта падает на мгновение до нуля. Итак, потенциальная энергия при автоколебаниях регулярно переходит в кинетическую, а кинетическая затем обратно в потенциальную. Поскольку потери энергии в таких колебаниях отсутствуют (можно указать и на причины этого и на следствия – отсутствие излучения), то такое колебание может происходить вечно. Любой атом отличается от «вечного двигателя» только тем, что он не производит никакой работы, а, следовательно, двигателем не является.

**Очень многие физические явления стали бы понятны, если бы были привлечены весьма строгие и математически обоснованные результаты ТАУ к некоторым моделям физических процессов [1–6]. Достаточным основанием для привлечения ТАУ к анализу любой системы, и даже необходимым условием применения этого аппарата является обратная связь в динамических системах. То есть, нужны два факта: обратная связь и существенность ее динамических свойств.** Обратная связь – это взаимное влияние одной величины на другую (и обратно – «другой» на «одну»). Обратная связь имеет место: ведь положение электрона влияет на силу, сила влияет на ускорение, ускорение влияет на скорость, а скорость влияет на положение. То есть: какую величину не измени – изменятся и все другие в едином контуре взаимовлияния. Наличие существенной зависимости от динамических свойств тоже не подлежит сомнению. Ведь возрастание скорости электрона почти ничем не ограничено, так как возрастание силы притяжения происходит почти до «бесконечности» с уменьшением расстояния «до нуля». А если бы даже и

принимать в расчет ограничение скорости, скажем, постулатом теории относительности (есть основания сомневаться, что он справедлив), то даже и в этом случае скорость, пусть она составляет всего лишь часть от скорости света, она начнет весьма сильно влиять на силы взаимодействия и на результат этого взаимодействия. Вообще говоря, **для того, чтобы назвать систему динамической, достаточно признания, что ее быстроедействие не бесконечно. Поэтому, строго говоря, все системы – динамические.** Исключением можно считать лишь те системы, в которых изменение одной группы величин во времени столь медленно происходит, что их изменением на коротких интервалах времени можно пренебречь, и поэтому систему можно рассчитывать по приращениям величин в цикле.

К таким системам относятся модели движения астрономических тел. Но если и туда привести учет динамики, то получим в результате уравнения прецессий, которые имеют место, так что, строго говоря, и эти системы – динамические.

Среди явлений, которые физики до сих пор не могут объяснить, особое место занимает шаровая молния. Привлечение ТАУ позволяет утверждать, что если бы сгусток вещества в плазменном состоянии мог бы быть преобразован в диполь, и одна из его компонент под действием электромагнитных сил притяжения была бы разогнана до релятивистских скоростей, то этот сгусток мог образовать квазиустойчивое образование, аналогичное макромодели атома водорода. То есть можно указать условия, при которых, например, капля парафина может превратиться в подобие светящегося шара. Если это образование будет квазиустойчивым, то есть не долговечным, но существующим на некотором конечном интервале времени, то в момент образования, жизни и (особенно) распада оно будет излучать высокочастотные электромагнитное излучение в очень широком спектре частот. Приблизительно это и происходит при наблюдении шаровой молнии. То есть имеется возможность не только объяснить природу шаровой молнии, но и указать, каким образом ее можно воспроизвести. Для этого необходимо осуществить сильный разряд электрического поля через какое-то количество вещества, легко переходящего в плазменное состояние – это может быть природный или искусственный парафин. Разряд должен ионизировать отдельные компоненты вещества и достаточно сильно разнести их дистанционно, чтобы при движении к рекомбинации они могли достаточно сильно разогнаться. Кто знает, может быть тут скрыт потенциал для новых научных открытий?

## 8. ОСНОВЫ ТЕОРИИ

«Интересы науки так же мало для меня значат, как для науки – мои интересы».

*Амброз Бирс. Гипнотизер.*

Давайте обсудим несколько вопросов: **«Что называют замкнутой динамической системой? Что называют неустойчивостью (устойчивостью) системы? Каковы основные причины неустойчивости замкнутых динамических систем?»** Прежде всего, **динамическая система** – это система из динамических элементов. **Динамическим элементом** мы назовем любой элемент, имеющий отклик на входное воздействие, который зависит от времени. Практически, под это понятие подпадает любой элемент, способный давать какой-либо **отклик на входное воздействие**. В частности, утверждение Козьмы Пруткова: «Щелкни кобылу по носу – она махнет хвостом» - это тоже одна из форм описания динамического элемента. Удобнее всего описывать их именно **по соотношению между выходным сигналом и входным воздействием**, которое этот выходной сигнал вызвало. Когда структура элемента не известна, такой элемент порой называют «черным ящиком», если же известно соотношение между входным и выходным воздействием, то есть математическая модель элемента, то этот элемент рассматривают в теории управления как **динамическое звено**.

Как правило, наряду с зависимостью выходного сигнала от входного сигнала, он зависит еще от некоторых условий, которые не известны и не измеримы. Поэтому уже не важно, известна эта дополнительная зависимость, или нет – в любом случае рассчитать ее нельзя, коль скоро сама величина не измеряется. Поэтому все отличие выходного сигнала от идеального, т.е. от такого выходного сигнала, который был бы в отсутствии этих дополнительных воздействий, сводят воедино и называют **помехой**.

Именно для **противодействия помехе организуют систему автоматического управления**. Если бы помехи не было, для управления объектом контура обратной связи не требовалось бы, но **при наличии помехи управление без обратной связи немыслимо**. Динамические элементы могут быть **линейными** или **нелинейными**. Идеальных линейных динамических элементов в природе не существует, но большей класс объектов можно свести к таким элементам, если осуществить **линеаризацию** отношения выходного сигнала к входному воздействию вблизи точки равновесного состояния. **Линеаризация – это приближенное**

**описание нелинейного объекта с помощью линейной модели** в малой области.

**Равновесное состояние** – это такое состояние покоя элемента, при котором входное воздействие принято равным нулю (или все они равны нулю, если их несколько). В ТАУ говорят лишь об изменениях воздействий, то есть о сигналах на входе и выходе в отклонениях от равновесного состояния. Поэтому, скажем, равновесное состояние системы терморегуляции живого организма при нулевых условиях не означает температуру, равную абсолютному нулю, а означает лишь, что температура такова, при которой в системе устанавливается равновесие, допустим, 36,6 градусов Цельсия. Отклонение от этой температуры – есть сигнал для системы терморегуляции, а отклонение концентрации каких-то биологических веществ или ферментов – это отклик системы, управляющее воздействие, осуществляющее стабилизацию.

Любое входное воздействие можно представить как **приращение некоторого воздействия**. Любое изменение выходного состояния – **приращение выходного сигнала** или просто «выходной сигнал». Приращение выходного сигнала, отнесенное к приращению входного сигнала, может описываться всего лишь коэффициентом, но чаще и правильнее описывать его функциональной зависимостью, содержащей также зависимость от времени.

Если это был бы всего лишь коэффициент, то не было бы необходимости употреблять слово «динамический». Поскольку все реальные объекты характеризуются конечным быстродействием, строго говоря, объектов, которые не были бы динамическими, в природе не существует.

Следовательно, описание кобылы надо уточнить: «Щелкни кобылу по носу – **через четверть секунды** она махнет хвостом». А если бы еще было сказано «А не щелкнешь – так никогда и не махнет», то из этого мы могли бы заключить, что кобыла – **полностью управляемый объект**: количество взмахов хвоста может строго контролироваться через входные воздействия – щелчки по носу, то есть **выходное состояние может быть полностью под контролем входных воздействий**.

Но на любой объект управления, как сказано выше, всегда **действуют еще и неизвестные возмущающие величины**, которые невозможно измерить непосредственно, результат действия которых, однако, накладывается на общий результат. В примере Козьмы Пруткова таким возмущающим воздействием могло бы быть воздействие слепней и иные побудительные мотивы, в связи с которыми кобыла бы махала хвостом и в том случае, когда никто ее не щелкал по носу, или не махнуть хвостом даже в



случае, когда ее щелкнули по носу. Итак, **динамический элемент – это объект или часть объекта, который может быть описан зависимостью выходного сигнала от входного сигнала и времени, а также от некоторых неизвестных факторов, называемых помехой.**

**Системой называется цепь из нескольких динамических элементов, при условии, что в этой цепи имеется хотя бы одна петля. Замкнутая динамическая система – это такая система, в которой замкнута имеющаяся петля обратной связи.** Разомкнутой системой называют ту же самую систему при условии, что связь с одним из элементов **искусственно мысленно прерывается**, для того, чтобы осуществить анализ (а в реальности она все равно замкнута). Если в системе обратная связь прерывается не мысленно, но и фактически, то это уже не система, а цепь динамических элементов.

Замкнутость петли используется для управления. Объект должен быть управляемым и наблюдаемым, и тогда даже в том случае, если на объект действует помеха и заставляет его изменять свое состояние в нежелательном направлении, контур обратной связи позволяет достигать предписанного состояния объекта.

Если вследствие укуса слепнями кобыла произвольно может махнуть хвостом от сорока до ста раз в час, то, подсчитывая количество таких взмахов (наблюдая) и добавляя требуемое количество щелчков (управляя), мы можем вручную стабилизировать эту величину, скажем, на уровне 200 взмахов в час. Это будет автоматизированная система управления, то есть система, действующая с участием человека в контуре обратной связи. Автоматическая система управления сделала бы это без нашего участия, если бы нам в голову пришло сконструировать такое «дощелкиватель кобыл». При этом если бы мы вручную устанавливали требуемое количество взмахов в час, система была бы все равно автоматической, а не автоматизированной.

**В контуре автоматического управления человек не присутствует, а его участие сводится только к формированию предписанной величины.** Быстродействие реакции человека не влияет на динамические свойства системы – в этом особенность любой автоматической системы и в этом ее отличие от автоматизированной системы.

Сама кобыла в этом отношении также является системой, которая устанавливает оптимальное равновесие по самочувствию - она не столь рьяно трясет хвостом, чтобы вконец утомиться, но и не столь редко, чтобы ее закусали слепни.

Итак, мы ответили на вопрос, что называют замкнутой динамической системой.

В динамической системе (ДС) обратная связь, как отмечалось, может быть как **отрицательной**, так и **положительной**. Отрицательная обратная связь (ОС) – это такая связь, при которой всякое отклонение от равновесного состояния вызывает возникновение сигнала, приводящего к уменьшению этого отклонения.

Если **на всех частотах в системе действует отрицательная ОС**, то в отсутствии внешних воздействий **система стремится возвратиться в исходное состояние.**

Положительная обратная связь – это такая связь, при которой всякое отклонение от равновесного состояния вызывает возникновение сигнала, приводящего к увеличению этого отклонения. **Если хотя бы на некоторых частотах в системе действует положительная ОС, то помеха на этих частотах усиливается в сравнении с помехой на других частотах.** Если же при этом коэффициент усиления всего контура, условно разомкнутого, на этих частотах больше единицы, то система не может возвратиться в исходное состояние, поскольку это её стремление к равновесному состоянию становится неустойчивым.

**Причина неустойчивости лежит в том, что на некоторых частотах отрицательная ОС переходит в положительную ОС.**

Системы с положительной ОС могут оставаться устойчивыми только в том случае, если эта ОС имеет коэффициент по всему контуру меньше единицы. Это логично: добавление к случайному возмущению какой-то доли от этого возмущения еще не приводит к лавинообразному нарастанию итогового отклонения от равновесного состояния. Если же случайное возмущение порождает добавку с этим же знаком и с усилением (т.е. с коэффициентом больше единицы), то произойдет лавинообразное отклонение от равновесного состояния. Итак, положительная обратная связь не нарушит устойчивости только тогда, когда она возникает для частот, на которых коэффициент усиления контура меньше единицы.

Откуда же возникает положительная ОС, если в целом сигнал обратной связи подается с таким расчетом, чтобы ее знак был отрицательный? Если сигналы, воздействующие на объект, разложить на гармонические компоненты и рассмотреть их отдельно, то каждая компонента получает индивидуальный сдвиг фазы, проходя через объект. Чем выше частота, тем, как правило, больше сдвиг фазы. Сдвиг фазы на 180 градусов равнозначен инвертированию сигнала, и именно **в этом случае отрицательная ОС превращается в положительную.** Следовательно, те частоты, которые задерживаются на 180 градусов и



больше, не должны усиливаться в контуре с коэффициентом, большим единицы.

## 9. ИНЕРЦИОННОСТЬ ОБЪЕКТА

«Тот, кто полагает, что нельзя ничего знать, не знает и того, можно ли знать, почему он утверждает, что он ничего не знает»

*М. Монтень. Опыты*

**Ничто не откликается на воздействие мгновенно.** Любой объект, который может быть описан откликом на входное воздействие, обязательно характеризуется некоторыми динамическими свойствами.

Поэтому же любой физический закон в его математическом выражении представляет лишь асимптотическое **приближение к действительности**. Так, например, закон о том, что действие равно противодействию, справедлив лишь в среднем, либо в предположении о бесконечной скорости распространения сил, которое, конечно же, если говорить строго, не соответствует действительности. Простейшая модель инерции – фильтр (звено) первого порядка.

Фильтром первого порядка называют звено, затухание частотного отклика которого обратно пропорционально частоте в первой степени. Такой фильтр в области затухания вносит сдвиг фаз только 90 градусов. Следовательно, если контур замкнутой системы описывается звеном первого порядка, такая система устойчива при любых коэффициентах усиления. Но звенья первого порядка существуют лишь теоретически или приближенно – в сравнении с пренебрежимо малыми звеньями более высоких порядков, которые входят в систему, но не влияют на ее устойчивость в силу малого коэффициента усиления системы на тех частотах, на которых эти высшие порядки объекта проявляются. На практике ни одна система не может сохранять устойчивость при сколь угодно большом коэффициенте усиления, в теории же такие системы существуют – это системы первого порядка. Если в системе имеется два фильтра, то сдвиг фаз постепенно переходит от 90 градусов к 180 градусов (при этом в области пропускания фильтров сдвиг фаз равен нулю). Такая система может быть устойчивой или неустойчивой, в зависимости от коэффициента усиления контура. Если единичное усиление соответствует суммарному сдвигу частот менее 180 градусов, система будет оставаться устойчивой. Повышением коэффициента усиления такую систему можно довести до возбуждения.

Отметим почти парадоксальное единство (при всей противоположности) устойчивости и неустойчивости: устойчивость всех движений

породило бы неустойчивость всего сущего, а неустойчивость движений к равновесию порождает устойчивость всеобщего движения, стабильность состояния вечно движущейся системы «Вселенная». Устойчивое движение к равновесному состоянию каждой подсистемы породило схлопывание этих подсистем, коллапс Вселенной как таковой, движение в ней прекратилось бы. И наоборот: неустойчивое движение к равновесному состоянию порождает извечную и нескончаемую устойчивость множества колебательных движений подсистем около равновесного состояния.

И в итоге достигается устойчивость вселенной, как вечно движущейся системы. Она движется во всех своих проявлениях и сущностях. Двигутся галактики и метagalактики, звезды и планеты. Двигается всё в вечных кругах, спиралах и эллиптических орбитах с прецессией, с круговоротом веществ и организмов, биоценозов и биосистем, биологических содружеств и противоборств, с движением в молекулах, атомах, элементарных частиц и того, из чего они составлены. Сама материя – это лишь движение её точечных различий. И все эти движения порой устойчивы благодаря лишь неустойчивому стремлению к покою.

## 10. СИСТЕМА ИЗ ДВУХ ИДЕНТИЧНЫХ ОБЪЕКТОВ ВСЕГДА НЕУСТОЙЧИВА

«Если с вами во всем соглашаются, проверьте, слушают ли вас».

*По К.С. Мелихану*

Если в контуре задействованы два идентичных объекта, то общий сдвиг фаз будет удваиваться. Следовательно, даже если эти объекты описываются фильтрами первого порядка (в первом приближении), то в итоге сдвиг фаз будет равен 180 градусов или более (за счет дополнительных звеньев).

Итак, из ТАУ следует, что система будет неустойчивой не только при положительной обратной связи с коэффициентом больше единицы, а также при отрицательной обратной связи и большом коэффициенте в случае, если темпы реагирования двух объектов будут приблизительно одинаковы. **Система из двух идентичных объектов почти всегда не устойчива.** Слово «почти» относится к требованию достаточного коэффициента усиления, а именно: он должен быть больше единицы. Такая система возникает, например, когда два человека идут навстречу друг другу по узкой тропинке. Каждый стремится уступить дорогу другому, но в результате зачастую возникают метания из стороны в сторону. Если один из прохожих отягощен коляской, санками или большой ношей, или если он более

медлителен вследствие возраста, то метаний не возникнет, и это также можно вывести из общих положений ТАУ.

Казалось бы, рассмотренный пример несерьезен, однако те же самые причины вызывают не только метания прохожих на узкой тропинке, но и метания испуганного пешехода перед движущимся транспортом (что намного опаснее), а также могут привести к столкновению самолетов и такие примеры, к сожалению, были. **Если бы проектировщики автопилотов и специалисты, разрабатывающие должностные инструкции авиадиспетчеров, знали эту проблему более детально, в некоторых случаях трагедии удалось бы избежать.**

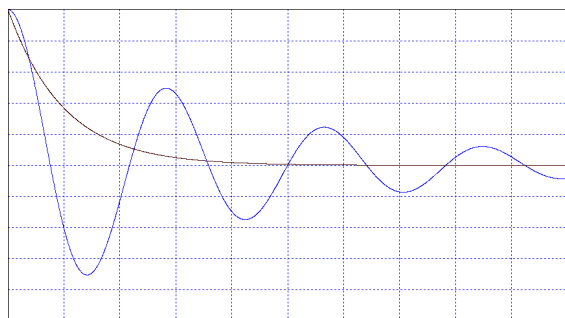


Рис. 3. Примеры устойчивых переходных процессов

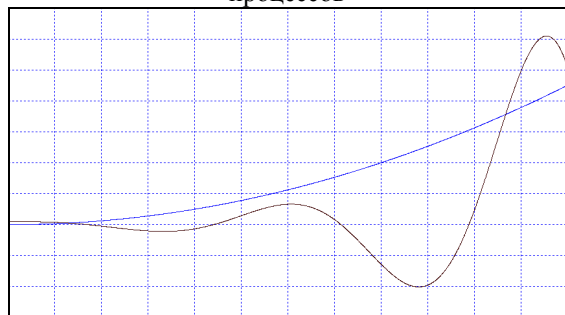


Рис. 4. Примеры неустойчивых переходных процессов

## 11. ЭКОНОМИКА СОЦИАЛИЗМА И РЫНОЧНАЯ ЭКОНОМИКА

«Злоупотреблять можно только хорошими вещами».

М. Монтень. Опыты

Всем известны кризисы, сокрушающие рыночную и плановую экономику. Эти кризисы называют «кризисом перепроизводства», но в реальности имеются колебания в обе стороны – как в область перепроизводства продукта, так и в область его же дефицита.

**Объяснение этого явления простое.** Быстродействие отклика производителя на ситуацию приблизительно совпадает с быстродействием отклика потребителя.

Производитель стремится стабилизировать ситуацию, то есть **по структуре в системе имеется отрицательная обратная связь**: чем больше произведено товаров, тем легче удовлетворяется спрос, следовательно, тем ниже цена и меньше потребность в дополнительном производстве; чем меньше товаров, тем они дефицитнее, тем больше в них потребность, и тем больше их производят. Следовательно, количество любых товаров должно в принципе уменьшаться при его перепроизводстве, и увеличиваться при его недостатке. Откуда же берутся кризисы? Население тоже реагирует на количество товаров. Чем больше их предложено, тем меньше население торопится их купить, следовательно, тем больше их остается на прилавках. Чем меньше их предложено, тем активнее население их покупает, опасаясь, что они совсем исчезнут, следовательно, тем меньше их будет оставаться.

Итак, потребитель имеет положительный коэффициент передачи по приращению количества товаров, производитель – отрицательный, общий коэффициент передачи в контуре отрицателен и больше единицы.

Помните, как директриса в фильме «Служебный роман» распекала своего подчиненного: «Откуда у нас в стране еще возникают перебои с некоторыми отдельными товарами? Это потому что некоторые нерадивые сотрудники в нашем статистическом бюро вовремя не планируют производство этих самых товаров!». Она ошибалась. Дефицит товара нельзя не заметить, когда он возникает, но когда он возникает производство этого товара запускать уже поздно. Дефицит невозможно запланировать. Пока дефицитный товар будет запланирован и пока наладится дополнительный выпуск, пока товар дойдет до производства, пройдет много времени. В результате, вероятнее всего, на рынок выплеснется такое количество этого товара, что возникнет его затаривание на складах. После дефицита возникнет перепроизводство.

Та же самая **всесильная отрицательная обратная связь**, которая в природе, чаще всего, стабилизирует ситуацию, в экономике может приводить к кризисам, поскольку не обеспечена устойчивость этой обратной связи. **Неустойчивая отрицательная обратная связь не стабилизирует, а дестабилизирует ситуацию.**

Элемент «промышленность - торговая сеть» вносит одну задержку, элемент «торговая сеть – потребитель» вносит другую задержку, в итоге для некоторой очень низкой частоты эта задержка становится равной 180 градусам, а коэффициент усиления все еще больше единицы. Вот на этой-то частоте и возникнут колебания: в одно время товара нет, и выпуск его снижается, в другое время товар имеется, а

выпуск его необоснованно наращивается. За счет неудачного стечения запаздываний получается так, что наращивание выпуска товара происходит тогда, когда товара на рынке мало, но на рынок поступает товар с задержкой, именно тогда, когда товара уже достаточно (и было бы достаточно даже без этого наращивания).

Система идет в разнос, то есть становится неустойчивой. Если один из этих элементов искусственно ускорить или замедлить, систему можно сделать устойчивой. Замедлить систему можно, например, разясняя, что рост потребности на данный товар – ажиотажный, не имеющий никакого отношения к реальной потребности. Действительно, иногда мы покупаем мыло, соль, спички, крупу и так далее не потому, что они у нас кончились, а потому, что опасаемся, что они закончатся в торговых точках. Способ ускорить один из рассмотренных элементов – замкнуть цепь обратной связи не через инерционную систему государственного контроля производства продуктов массового потребления (включающую в себя постановления партии и правительства, строительство новых заводов, организации программ по решению вопроса), а через мелкого товарного производителя. Мелкие производители реагируют на требования рынка более оперативно, чем сам рынок меняет свои требования, и поэтому система становится устойчивой. Отсюда у наших бывших партийно-хозяйственных лидеров так неожиданно возникла любовь к «рыночной экономике». Устойчивость также можно было бы обеспечить, внося регулирование не только по состоянию (то есть по отсутствию товаров), но и по скорости изменения состояния. Это называется **введением в управление производной**. Товар надо начинать выпускать не тогда, когда его стало мало, а тогда, когда его количество начало уменьшаться.

Но поскольку товары товарам рознь, то и рыночная экономика может спасать в области производства не самых значимых товаров массового производства, но она может оказаться несостоятельной в области других товаров и привести к инфляции – другому неустойчивому процессу, порожденному также положительной обратной связью в области спроса на особый «товар» - на деньги.

## 12. ПОЛИТИЧЕСКИЕ ПРОЦЕССЫ

«Началось всё с раздоров между министрами: уж если собирались вместе три министра, то лишь для того, чтобы сговориться, как половчее изгнать четвертого».

*А. Решильё. Мемуары.*

«О, на что только люди не отваживаются! Что только они могут сделать! А что люди делают ежедневно, не осознавая, что они творят!»

*В. Шекспир. Много шума из ничего*

«В своем «Государстве» он [Платон] прямо заявляет, что для пользы людей часто бывает необходимо их обманывать».

*М. Монтень. Опыты.*

Политические процессы порой приводят к катастрофическим последствиям, которых никто не ждал и не планировал – к кризисам, включая военные конфликты. Здесь тоже может причиной быть **неустойчивость системы дипломатических уступок и нажимов**. Обе стороны могут одновременно начать **терять контроль ситуации**, и это тем более вероятно, чем более похожи страны, вступившие в дипломатический контакт. Если одна из стран реагирует более медленно, чем другая, вероятность устойчивой политики выше. Поскольку личность лидера страны также играет большую роль в темпах и силе ответных дипломатических шагов, равновесие между противостоящими державами держится не только на объективном раскладе сил, но и на личных качествах первых лиц в стране.

При диктатуре обратная связь между властью и народом минимальна. Поэтому диктатура редко разваливается по причине чисто субъективной, для ее краха нужны объективные причины – катастрофы в экономике или внешней политике. Демократия же может развалиться и без всяких объективных экономических или политических причин, поскольку демократия предполагает не только влияние правительства на народ, но и **обратное влияние** народа на правительство. Если этот контур взаимного управления, который должен строиться как устойчивый, вдруг в силу особых динамических свойств, то есть в силу соотношения быстроедействия отдельных элементов, вдруг станет неустойчивым, то произойдет революция. Причем, эта революция не будет обусловлена ни войной, ни неурожаем, ни внешней агрессией, никакими иными объективными факторами, вызвавшими неустойчивость, либо такие факторы могут всего лишь сыграть роль внешнего возмущения, спускового крючка, который привел в действие систему с положительной обратной связью, обрушил лавину, накопившую неустойчивый потенциал.

Общеизвестно, что ни в одной из

революций ни одна из сторон, инициировавших ее, никогда не достигала поставленных целей. И это естественно, поскольку революционные процессы никогда не контролируются полностью ни одной стороной, они развиваются по своим законам – о катастрофах, то есть **по законам теории систем с положительными усиливающими обратными связями** в контуре. Если бы преобразование управлялось со стороны тех, кто его затеял, это были бы уже не революции, а реформы. Реформы могут перейти в революции, если система утрачивает свойства, которые делали ее устойчивой. Для того, чтобы революция вернулась на стадию реформ, необходимо менять так много элементов в системе, что эти процессы всегда болезненны и длительны. К тому же эта ситуация осложняется отсутствием какой-либо силы, которая могла бы целеустремленно исправлять связи в общественной системе. Да это и естественно: всякая сила, руководящая общественной системой, не стремится сделать ее устойчивой и самоорганизующейся, а стремится максимально подчинить ее себе. В этом кроется ловушка и парадокс: **максимальное воздействие отнюдь не является наилучшим управлением!** Так же, как объект, «наилучшим образом» откликающийся на управляющие сигналы – не самый лучший объект для управления. В таких случаях простая борьба с пьянством оборачивается тотальным уничтожением виноградников, дисциплина понимается лишь как полное повиновение вплоть до муштры и безынициативности, а свобода трактуется как вседозволенность – «перегибы на местах».

В системах с такими объектами регулятор выполняет функции демпфирующего устройства, сглаживает управляющий сигнал. На этих примерах мы видим, что для того, чтобы успешно управлять общественной системой, не достаточно иметь просто хороший руководящий орган, его качества должны соответствовать качествам управляемого объекта, и это соответствие должно осуществляться с позиции системного подхода. Простое копирование «регулятора», успешно действующего в совершенно других условиях, никогда не даст желаемого результата, ведь объект-то другой.

### 13. СТОЙКОСТЬ СУЕВЕРИЙ

«Именно эта его беспощадность в высказывании мыслей, в равной степени неприемлемых и для науки, и для религии, его безграничный еретизм, эта его универсальная кощунственность, проистекающая из интеллектуальной отваги, должно быть, и оттолкнули от него всех читателей».

*С. Лем. Новая космогония.*

«Мозг здорового бодрствующего человека – предельно неустойчивая хаотическая система».

*Пригожин И.Р., Стенгерс И. Время, хаос, квант*

Различные суеверия (куда я отношу и религию) очень стойко держатся в сознании людей. Одной из причин этого я усматриваю **положительную обратную связь** в процессе усвоения этих суеверий. Чем больше мы в них верим, тем больше внимания обращаем на различные приметы и признаки грядущих несчастий, тем дольше держим их в памяти. Неприятности случаются достаточно регулярно, у всех и всегда, независимо ни от возраста, ни от пола, ни от погоды, ведь не будь неприятностей, мы бы стали называть неприятностями даже такие мелкие события, на которые в нормальной жизни не обратили бы внимания. Адаптация человеческой психики к фактическим событиям жизни приводит к тому, что почти каждый человек считает, что в его жизни имеются и радостные события, и неприятности, хотя если бы имелся стандарт на неприятности и на радости, то оказалось бы, что кто-то лишен неприятностей, а кто-то радостей. Это – действие **отрицательной обратной связи**, которая подстраивает наше ощущение того, что является неприятностью, а что – удачей к тому, что в действительности с нами происходит.

Другой контур обратной связи, на этот раз **положительной**, таков: если отмечена примета, то следует ждать неприятностей, а кто ждет неприятностей, тот всегда их дождется (аксиома). Кто дождался неприятности и сопоставил ее с ложной причиной – приметой – тот еще более уверовал во взаимосвязь этой приметы и впоследствии неприятности.

Существует две взаимоисключающие жизненные позиции: «Все люди – волки» и «Все люди – братья». Как отметил Николай Козлов (современный психолог), обе они обладают организующей силой. Эти мировоззрения заставляют нас относиться к людям именно так, что они в ответ будут к нам относиться в точности так, как мы от них ожидаем, и тем самым будут утверждать нас в изначальной точке зрения. Относитесь к людям по-братски, и они будут своим поведением убеждать вас в том, что такое отношение обоснованно.



Относитесь к ним по-скотски, и они докажут вам, что они этого заслуживают. Налицо положительная обратная связь, причем с коэффициентом намного больше единицы.

#### 14. ГИПОТЕЗЫ

«То, как мы пытаемся описать Природу, обычно является непонятным для нас самих».

*Ричард П. Фейнман.*

Странная теория света и вещества  
«Если, рассуждая о природе богов и происхождении вселенной, мы не достигнем желанной нашему уму цели, то в этом нет ничего удивительного. Ведь следует помнить, что и я, говорящий, и вы, судьи – всего лишь люди; так что, если наши соображения будут правдоподобны, не следует стремиться ни к чему большему».

*М. Монтень. Опыты.*

«Гипотез не измышляю»

*Исаак Ньютон*

В природе мы наблюдаем множество отрицательных обратных связей – они стабилизируют всевозможные процессы. Встречаются и положительные обратные связи, дестабилизирующие другие процессы.

Лавина, срывающаяся с гор, ядерный взрыв, землетрясение, эпидемии и вспышки численности отдельных видов – все это примеры действия положительных обратных связей. Примеров действия отрицательных связей всех не перечислить! Это и круговорот воды в природе, и стабильность численности животных и растений в биоценозе, сотрудничество различных видов и паразитизм – тоже! Мы видели, что некоторые явления в физике принципиально не понятны, если не привлекать аппарат ТАУ, но они же оказываются вполне объяснимыми с этих позиций.

В биологии есть тоже много загадок, которые кажутся неразрешимыми.

Одна из таких загадок – эволюция. Нет, я не ставлю под сомнение эволюционную теорию Дарвина, но кое-какие цифры никак не сходятся.

Согласно представлению Ч. Дарвина природа действовала методом проб и ошибок, но при этом пробы были абсолютно случайны, а качество этих проб определял лишь естественный отбор. То есть все неудачные пробы природы были отвергнуты неудачными итогами введения их в жизнь, а удачные – закреплялись. Чисто теоретически эта теория очень красива. На практике она полностью опровергается. Если бы природа реализовывала все свои «пробы», то мир был бы завален трупами от этих неудач. Ведь к неудачам следует относить не только нежизнеспособные

особи, но и такие, которые способны не только родиться, но и вырасти до взрослого состояния, хотя и не участвовать в размножении вида, и не только по причинам неудачности данной «конструкции», а также и по причинам ее неуместности в данном климатическом и биологическом окружении. Ведь для того чтобы **чисто случайно** наткнуться на единственно правильное решение, надо перепробовать чрезвычайно много вариантов, что природа просто не могла бы себе позволить. Среди них – даже и удачные в смысле отрыва от новой среды, от новых внешних условий, а неудачные они лишь по отношению к изменившейся окружающей среде. То есть, допустим, если в сложившейся ситуации животному нужны оперенные крылья, то природа должна перепробовать и рога, и лапы, крылья, покрытые шерстью, и чешуей, и панцирь, и прочие новые приспособления, а среди них также и крылья. И где же эти самые «пробы»? Не говоря о том, что между бескрылым животным и крылатым стоит очень много промежуточных форм (как крылья не того размера или силы), мы все же приходим к выводу, что поиск должен носить признаки осмысленности, направленности, далеко не все тупиковые варианты проходят апробацию. И вообще летающее животное от нелетающего отличает отнюдь не один новый признак, а **целый набор** из сотни-другой отличительных свойств, включая и форму тела, и особую структуру абсолютно всех других органов (даже глаз). Между этими отличиями «дистанции огромного размера». В принципе они определены разными наборами генов, но и сами гены – это величайшая загадка природы. Ведь они хранят не только ту информацию, каким должен быть организм взрослой особи (как раз эту-то информацию они, видимо, не хранят), а хранят они информацию о том, **как должен развиваться организм**, начиная с оплодотворенной клетки, как должен формироваться каждый орган и **на каждой ступени своего развития**. Причем, не в большом объеме этой информации тут самая большая загадка, хотя и в этом тоже – ведь самая первая оплодотворенная клетка является не только хранилищем этой информации, как ей предстоит развиваться и в какую особь следует развиваться, она еще содержит и **считыватель этой информации**, и **завод по воплощению этой информации в жизнь**. Достаточно убрать хотя бы одно звено, и организм не вырастет. И как тут говорить о случайном переборе всех вариантов? Все эти случайные варианты попросту нежизнеспособны, и они вообще практически никогда не реализуются. В тех редких случаях, когда в геномном наборе происходят мутации, никогда не возникает нового вида или животного с хотя бы одним



новым признаком, который бы мог давать какое-то новое свойство (не обязательно положительное), закрепляться наследственно и не препятствовать живучести организма. Но даже если бы это и было (а этого нет), то надо было бы признать, что природа ставит целенаправленные эксперименты, а не случайные. И вот тут-то объяснение может крыться **в наличии положительных обратных связей**. То есть на стадии видообразования природа не просто столь же случайно, как и в обычных ситуациях, осуществляет изменчивость во всех направлениях сразу (авось, повезет), а делает эту изменчивость в нужном направлении, **с прогнозируемыми результатами** (именно так!). Следовательно, в генетической информации видов содержится также и информация о том, как должны мутировать виды в критической ситуации, подобно тому, как в механизме роста дерева содержится информация о том, как должен искривляться ствол, чтобы обойти неожиданно возникшее на пути его роста препятствие. Дерево, встречая препятствие, не расщепляется и не прекращает рост, а начинает изгибать ствол в единственном направлении – как будто кто-то мудрый руководит этим процессом огибания. Точно также и птицы, развиваясь изолированно в отсутствие естественных врагов, теряют способность летать и приобретают новые качества, идентичные в идентичных условиях. Так страусы эму и нанду весьма схожи по строению и по образу жизни, и при этом совершенно не являются родственниками. Следовательно, имеется некоторая генетическая информация на уровне птиц вообще о том, как приспособливаться к этим новым условиям. Другой вопрос, примыкающий к этому – как эта информация могла возникнуть и как она передается. Многие решают его привлечением Бога. Наука по этому пути не должна идти.

Сейчас уже достоверно известны два акта. Первой: вирусы способны жить в космическом пространстве, вирусы могут находиться в льдистых структурах, например, комет, то есть кометы могут переносить вирусы от одних небесных тел к другим. Второй: на Земле практически везде, где есть вода, там есть и жизнь, даже в подземных пещерах, где, казалось бы, бактериям и вирусам нечем питаться, тем не менее, есть жизнь в зародышевой форме, и ей для жизни достаточно лишь воды и природных материалов из недр планеты.

Из этого можно предположить, что вовсе не земная жизнь, а жизнь вселенская – итог эволюции в космическом масштабе. В соответствии с теорией панспермии, планета Земля заражена жизнью тотально, и космос, вероятно, тоже.

С другой стороны, время существования Земли столь мало с позиции необходимого

времени для того, чтобы создать то богатейшее разнообразие животного мира, которое существует, простым методом перебора и отбраковки. Любой специалист по информатике с этим согласится при наличии правильных исходных данных. Этими правильными исходными данными являются, во-первых, количество отличий одного вида от другого – так называемое кодовое расстояние. То есть если бы всю генетическую информацию, скажем, о человеке, закодировать двоичным кодом, и также поступить с кодовой информацией о ближайшем родственнике, не о горилле, о предке человека, то количество несовпадающих бит (элементарных единиц информации) будет кодовым расстоянием между человеком и этим предком. При этом какая-то часть бит будет не столь важной, ибо она отличается и у различных людей, а другая часть бит будет принципиально важной, ибо она отвечает за отличия человека от гориллы. Любопытно, что те биты, которые отличают одного человека от другого, меняются всегда – нет двух одинаковых людей. Те же биты, которые отличают человека от всех остальных животных, изменяются чрезвычайно редко. Это происходит в силу различных природных процессов, приводящих к изменениям тех и других особенностей наследственности. Итак, если принять возможным случайное изменение одного бита за некоторую константу, которую можно оценить частотой появления уродцев – но оценить лишь приближенно, ибо нам не известен ни один факт появления нового вида на протяжении истории человека – то можно представить лишь нижнюю границу времени, которое необходимо, чтобы от предка человека возник человек.

А ведь это – лишь крохотная ступенька в эволюции! Вспомним теперь, чем отличается человек, скажем, от черепахи, от рыбы, от медузы – да что там говорить, от амебы, от вируса! И где было природе на Земле получить достаточный запас времени для экспериментов с вирусами и бактериями, пока не возникнет человек? Проблема тут же перестает быть проблемой, если мы допускаем, что эволюция происходила не только и не столько на Земле, сколько во Вселенной вообще и во времени – бесконечно.

Поскольку далеко не все ученые современности являются сторонниками «теории большого взрыва» (таковым не был и Эйнштейн) и возникновения Вселенной из ничего, все эти чисто теоретические и не проверяемые гипотезы не соответствуют критериям научности. Причиной этих теория явилась ошибка при трактовке эффекта Хаббла. Поэтому мы можем оставить разговоры о сроке существования Вселенной. Со строго научной позиции она существует вечно, и будет

существовать вечно. В этих простых и естественных предположениях любая даже самая немыслимая эволюция имела достаточно времени, чтобы осуществиться до конца. На Земле же попросту развернулась ее частная реализация для данных частных условий.

В этом случае мы можем (или должны) предположить, что даже самый ничтожный вирус несет в себе информацию не только о себе самом и о том, как он должен существовать и развиваться. Он может также нести в себе информацию и о том, как он способен мутировать, чтобы развиваться до чего-то большего, чем вирус, до одноклеточного существа, и далее до многоклеточного, и далее, далее.

Фантастика? В каком-то смысле да, но достаточно обоснованная фантастика на базе понимания, насколько сильно отличаются по сути системы без обратных связей от систем с таковыми связями. Эта фантастика – гораздо более научная гипотеза, чем теория о возможности путешествий во времени, о расширяющейся Вселенной и об ее возникновении в результате взрыва.

В природе все охвачено обратными связями – это факт, не нуждающийся в доказательствах. И если мы где-то не замечаем этой связи, то это характеризует не природу, а нашу недостаточную наблюдательность или праздность ума. И вот если на нас, на людей, влияют другие живые организмы, вызывающие у нас болезни (кстати, некоторые живые организмы нам просто необходимы и болезни у нас возникают при отсутствии этих организмов), то и мы тоже на эти организмы влияем. И вирусные болезни не только губят отдельные особи, они к тому же и считывают генетическую информацию у этих самых особей. Этот факт доказан тем, что вирусы приспосабливаются к лекарствам, предназначенным для борьбы с ними, что и снижает действенность противовирусных препаратов.

Чем более распространена та или иная вирусная болезнь, тем, видимо, сильнее эта обратная связь, ведь тем быстрее происходят мутации этих вирусов, их приспособляемость к новым достижениям медицины.

Случайные и неслучайные мутации происходят в организме человека, мутации происходят в вирусах, по-видимому, при заражении организма вирусами эти процессы могут взаимодействовать или взаимно «считываться». Вирусами, видимо, заражаются все живые существа в той или иной форме. Это – еще неизученная генетическая связь не только между особями одного вида, но даже между особями разных видов, родов, классов.

Итак, вирус гриппа – возможно, один из основных факторов эволюции видов. Если бы человечество перестало болеть вирусными

болезнями, может быть, оно бы перестало обмениваться генетической информацией со всем остальным биоценозом, и, нельзя исключать, что в этом случае оно вымерло бы. А может быть, оно и не возникло бы.

Вирус уходит в космос и несет в другие миры информацию, и не только о гриппе, но и о человеке. Так что когда в другом мире где-то зародится новая жизнь, на ее качество, возможно, окажет влияние не только вирус, выросший на земле, но и все остальные виды, передавшие этому вирусу свою генетическую информацию.

Как знать, может быть, чем больше гениальных людей болеет гриппом, тем больше генов гениальности будет передано на другие планеты? И лучше будет та новая жизнь, которая возникнет где-нибудь в далеком созвездии? Конечно, последний абзац данного раздела – исключительно гипотетический.

Однако и это «предсказание» уже сбылось. По сообщению журнала «Мир науки» [7], не только в некоторых вирусах, но и во многих высших организмах происходит не только прямой синтез ДНК по РНК, но и обратный необычный синтез РНК по ДНК. Таким образом, найдено место замыкания обратной связи в петле саморазвития и эволюции всего живого на Земле, а само предсказание сделано было совершенно без использований фактических биологических сведений.

## **15. К ТЕОРИИ СОЦИАЛЬНЫХ КАТАСТРОФ И КОНФЛИКТОВ**

«Для понимания действительности нужна фантазия».

*Г.Малкин*

### **15.1. Предмет исследования**

Социальными катастрофами будем называть **неустойчивое поведение социальных систем**, доводящее переходные процессы до крайности. Это: революции, войны мировые и войны приблизительно равных по силе противников, экономические и иные кризисы, революции, бунты. Сюда же надо отнести забастовки (кроме особой формы почти плановых забастовок), самоубийства, разводы, бегство детей или взрослых из семьи, преступления под действием стресса, супружеские измены, расколы общественных объединений и тому подобное.

Все эти ситуации характеризуются крайним поведением особей, выходящим за рамки обычного поведения социального индивидуума или группы индивидуумов.

По этой причине не будем относить к социальным катастрофам захватнические войны более сильной державы или группы союзников

по отношению к более слабым государствам, развод с целью более выгодного брачного союза, плановое банкротство с целью извлечения дополнительных прибылей, плановые забастовки, которые определяются скорее традициями, чем состоянием дел, и тому подобные события.

В данном случае слово «катастрофа» применяется не в смысле большого негативного результата, а в смысле **нелогичности поведения людей, потери управляемости ситуацией, незапланированный конфликт**. Это не означает, что мы не считаем бедствием захватническую войну, убийство с целью грабежа и тому подобные несчастья. Просто эти явления могут быть объяснены определенным замыслом той или иной действующей стороны. Социальная же катастрофа в рассматриваемом нами значении – это бедствие, которое не было спланировано и реализовано теми силами, которые могли бы его осуществить.

Волк может съесть барана, это будет обычным явлением. Аналогично сильная держава может поработить слабую. Данный конфликт антагонистический, непримиримый. Однако, он не катастрофический. То есть нельзя сказать, что в данном случае обе стороны не ведают, что творят, и получают отнюдь не тот результат, к которому стремились: про сильную сторону этого не скажешь. Сильная сторона является ведущей, управляющей, довлеющей, она добивается того, к чему стремится, не считаясь со слабой стороной. Такое поведение и его результаты вполне предсказуемы из рассмотрения обычной причинно-следственной цепи. Вот если бараны накинута на волка, это будет не обычным явлением. Схватка льва с крокодилом со смертельным исходом для обеих сторон (см. «Приключения Барона Мюнхгаузена») – вот пример «социальной» катастрофы. Война стран при приблизительном паритете сил, не выгодная, в конечном счете, ни для одной из сторон – это кризис дипломатии, нарушение устойчивости, которую в обычном состоянии обеспечивают взаимные консультации и переговоры. Это является катастрофой потому, что ни одна из сторон к этому не стремилась, и, тем не менее, обе стороны вели себя так, чтобы это произошло.

Рассмотрим пример революции. В социалистических учебниках истории дело изображается так, будто большевики спланировали и осуществили революцию. На самом деле, конечно, большевикам это было бы не под силу. Революция произошла как результат групповых действий (**неконтролируемых** в целом никем) противоборствующих сил, действительно вносящих управляющий вклад в ход истории в этот момент. Управляющий вклад, но **не управление**. Управление отличается именно тем, что управляющее воздействие вводится

ровно таким, чтобы получить желаемый результат. Если результат воздействия на систему может быть наблюден, то ввод управляющего воздействия осуществляется не вслепую, а с попыткой достижения цели. Однако если наблюдение результата сильно запаздывает от времени введения управляющего воздействия, то **система может стать неустойчивой** настолько, что лучше бы тем, кто пытается «осознанно» управлять, делать это неосознанно и без наблюдения результатов. В этом случае управление не было бы настолько катастрофическим. Группки заговорщиков – подпольщиков могли, конечно, ускорить революцию, но не могли бы полностью организовать этот процесс. Не снежинка ломает леса, а огромный снежный ком, который налипает на эту снежинку. Но для возникновения лавины необходимы соответствующие условия. Условия эти таковы, что **изначальное слабое воздействие в системе многократно усиливается, приводя к сильному воздействию**. Механизмы этого явления одинаковы в радиочастотном генераторе, в лазере, в снежной лавине, в революции и в иных процессах неустойчивого управления (устойчивой генерации).

Война может оказаться неизбежной, если одна или несколько держав развиваются стремительно по сравнению с другими, и если уровень цивилизации еще не дошел до такого состояния, когда стабильность в мире может гарантироваться международными организациями. Эти державы не считаются со своими жертвами, здесь возникает простое порабощение, **управление** с помощью силы. Если же несколько держав имеют соизмеримые силы, то, в конечном счете, никто из их лидеров не заинтересован в войне. Есть примеры в истории, когда обе стороны, развязавшие войну, твердо уверены, что они стремились только к миру, а война явилась следствием их опасения, что в дальнейшем предотвратить агрессию противника уже не будет возможным. **Незапланированная ни одной из сторон** война против примерно равного по силе соперника в этом смысле рассматривается нами как «катастрофа». Если же одна из сторон сознательно стремится развязать войну, по-видимому, в этом случае теория катастроф не поможет избежать конфликта.

Возвращаясь к революции, отметим, что, как правило, никогда ни одна из тех сил, которые способствуют возникновению революции, не способна сохранить свою власть в ходе ее развития. Поэтому даже те силы, которые замышляют и осуществляют революцию, перестают влиять на ее ход. Следовательно, в какой-то момент **система перестает быть управляемой**. В конечном счете, в уже развалившейся системе приходится

заново создавать аппарат управления.

В разводах со скандалом также обе стороны теряют больше, чем они бы потеряли при мирном разрешении конфликтов. То же можно сказать про преступление, как крайнюю меру решения проблем, если, конечно, это не результат чьего-то преступного плана – в этом случае имеет место отношения «злодей и жертва», без обратной связи. Каким бы социальным чутьем жертва не обладала, она не может преодолеть изощренный план, в котором психология жертвы не имеет значения. Если же преступление произошло спонтанно, в результате разгоревшейся ссоры, то имеет место именно **социальная катастрофа**.

### 15.2. Метод исследования

Метод анализа устойчивости социальных систем основан на принципах анализа устойчивости замкнутых динамических систем, разработанных Ляпуновым и другими классиками теории автоматического управления, которые широко известны.

Основные структуры и краткие результаты рассмотрены выше, в разделах 8 и 9.

### 15.3. Некоторые результаты

Катастрофой можно назвать и ту ситуацию, когда отклонение от равновесия станет недопустимо большим, превышая обычный прогнозируемый результат, как бы «следствие неадекватно причине». В природе большая часть механизмов регулирования устойчива. Одна из причин устойчивости может состоять в различном быстродействии элементов этой системы. Другая причина может состоять в локальных стабилизирующих обратных связях, которые, в конечном счете, также приводят к различному быстродействию разных элементов. Различие быстродействия обеспечивает в качестве следствия единственную причину затухания амплитудно-частотной характеристики в области единичного коэффициента.

При каких же условиях такое неустойчивое движение может происходить? Если предельно упростить эти условия, то они, чаще всего, состоят в том, что в системе задействованы два или более элемента с соизмеримым быстродействием, и при этом общий коэффициент усиления контура больше единицы.

Из теории автоматического управления мы знаем, что большая часть простейших элементов системы описывается простым апериодическим звеном, характеризующимся своим коэффициентом усиления и своим характерным временем запаздывания. Переходный процесс такой системы на ступенчатое воздействие описывается экспоненциальным движением. Запаздывание на очень низких

частотах у таких систем пренебрежимо мало, а на очень высоких частотах соответствует 90-градусному сдвигу. Последовательное соединение двух систем с приблизительно одинаковым быстродействием при условии общего коэффициента усиления большего, чем единица, дает ту самую неустойчивую систему. В такой системе задержка воздействия в контуре на некоторых частотах соответствует 180 градусам, то есть отрицательная связь становится положительной. Именно на этой частоте система начинает колебаться с все увеличивающейся амплитудой. При некоторых дополнительных условиях одного колебания достаточно, чтобы довести систему до катастрофы, то есть до крайнего состояния всех задействованных величин.

Чтобы движение было устойчивым, оказывается, достаточно, чтобы быстродействие двух самых медленных апериодических звеньев существенно различались. Тогда в значительном диапазоне коэффициентов усиления система будет устойчивой. Именно в том случае, когда народ колеблется от нерешительности к отчаянию, а правительство также мечется от уступчивости к репрессиям, происходит та самая раскачка, которая приводит к катастрофе.

Таков же механизм конфликтов разных общественных групп. Сначала представители какой-то группы просто заботятся о сохранении своих традиций, особенностей. Это всегда связано с проведением мысли о том, как хороша данная группа. Но слово «хороший» – это сравнительная характеристика. Из этого следует, что представители других групп «хуже». Отсюда обиды. За мелкие обиды – мечь, то есть нанесение обидчику ущерба большего, чем нанес он. Коэффициент усиления больше единицы. Если обидчик уничтожается (такое было в истории), то конфликт заканчивается. Это, конечно, трагедия. Это – Макиавеллевский метод достижения победы. Если обидчик не уничтожен, то он будет мстить. И его мечь вызовет желание еще большей мести. Возникает опять раскачка системы, колебания, доходящие до катастрофы.

Таким образом, известные фразы: «Война – есть продолжение политики иным путем» и «Война – это сконцентрированная политика» – не проясняют сути дела. Они лишь скрывают истинные причины этих, безусловно, катастрофических явлений. Кроме тех случаев, когда «злодей поедает жертву», война – это крах дипломатии. А крах происходит вследствие приблизительно одинаковых способов ведения переговоров, идентичных взглядов на «допустимость» тех или иных силовых мер, приблизительно одинаковых военных сил и иных неудачных соотношений. Таким примером может служить политический кризис, который ставит мир на грань войны. Положительным примером по выходу из такого кризиса может



служить неординарное действие одного из политиков (именно только одной стороны). Это кажется странным, но именно различия методов ведения переговоров в кризисной ситуации могут служить выходом из этого кризиса.

В работе В.И. Ленина «Государство и революция» утверждается, что можно построить бесклассовое, следовательно, бескризисное общество. Автор верил, что только существование капиталистических стран является причиной войн, что войны между социалистическими государствами невозможны. Однако был период, когда именно отношения между СССР и Китаем вызвали наибольшее беспокойство. И это было как раз в тот момент, когда военные силы вступили в такую стадию, что нельзя было с достаточной уверенностью сказать, какая из стран способна нанести другой больший ущерб, зато можно было точно предсказать, что ни одна из этих стран не выиграет от такой войны. Ни одна война не может считаться выигрышной, с моральной точки зрения. Но, в данном случае речь идет о гарантированной невозможности извлечения каких-либо преимуществ.

Каждый народ предпочел бы, чтобы им управляли без социальных катастроф. Лучше было бы, чтобы революций не было. В конечном счете, только мирное развитие дает человечеству больше свободы. Никакая политическая система не сделает неандертальцев свободными. Только развитие науки, техники, технологии и иных производительных сил способно дать человеку истинную свободу. Те, кто рушат дворцы, не несут свободу. Те, кто строят дома, её приближают. Даже если эти дома построены не совсем свободными людьми.

Вспомним героев П.Мериме, Н.В. Гоголя, А.М. Горького убивающих собственного дитя. Для того ли они заводили детей, чтобы их убивать? Убийство собственного дитя – это крайняя степень конфликта. Любой родитель старается воспитывать потомство. Применяя поощрения и наказания, родитель добивается послушания от ребёнка. Ребенок, в свою очередь, хочет быть независимым. Для того чтобы заслужить поощрения и избежать наказания ребенок изобретает уловки. Сообразительный ребенок быстро постигает, что ложь, капризы, неадекватная реакция и иные уловки помогают достигать этой цели. Но ложь рано или поздно раскрывается, на капризы родитель перестает обращать внимание, и для усиления воздействия на ребенка родитель усиливает степень наказания и степень поощрения. Как правило, одни и те же родители используют телесные наказания и денежные поощрения для своих детей. И то и другое – крайности, весьма нежелательные в семейных отношениях. Применение крайних мер –

свидетельство катастрофической раскочки системы. Каков выход? Родители не могут увеличить быстродействие своих реакций, значит, необходимо уменьшить быстродействие, ввести интегрирование, накапливание обратного воздействия – воздействие должно быть не откликом на сегодняшнюю провинность или заслугу, а подведением итогов за неделю, месяц, полугодие. Воздействие должно нарастать намного медленнее, чем в той системе, которая дошла до неустойчивого управления. Там, где ремень оказался бессилем, более эффективным было бы предупреждение: для начала получи просто замечание, но чем больше накопится таких замечаний, тем сильнее будет итоговое наказание. Соответственно, после успеха не следует на радостях бежать покупать велосипед, а это сильное поощрение должно явиться результатом накопления многих успехов. При таком интегрировании провинности и успехи компенсируют друг друга, и если они чередуются и приблизительно соизмеримы по «величине», то и реагировать, оказывается, никак не надо. Если же вы хотите, чтобы успехи преобладали, то необходимо лишь постепенно повышать планку требований.

Бесконфликтным управлением можно более успешно вести ребенка к намеченной цели, тогда как конфликтное реагирование лишь разрушает систему «ребенок – родители». Упомянутые литературные герои – вовсе и не герои, а неудачники, а их дети – жертвы неумелого воспитания.

Николо Макиавелли учил государей, как править, избегая кризисов. Но в его трудах изложена тактика управления людьми. Стратегия же этого дела состоит, прежде всего, в недопущении социальных кризисов. Лучше отдать провинцию, чем потерять страну. Но тот, кто ради мира раздаёт провинции направо и налево, тот рано или поздно станет настолько слабым, что также лишится страны. Умный государь должен чутко реагировать на внешнеполитические события, включая расклад сил. Умных государей в истории было не так много. Мудрых государей, пожалуй, перечислить ещё труднее.

Наука управления коллективами и государствами без кризисов только еще требует своего развития, но без теории автоматического управления она не сможет быть верной и эффективной.



## 16. СИСТЕМНОЕ УПРАВЛЕНИЕ КОЛЛЕКТИВОМ

«Быстрее реформируется то, что было хорошо».

*Г.Малкин*

Один из аспектов функционирования поощрений и наказаний в коллективе состоит в положительной обратной связи по контуру «действие» - «реакция» - «последствие». Это замкнутый контур, который может передавать как положительное, так и отрицательное воздействие, а термин «положительная обратная связь» указывает на совпадение знаков приращения действия и последствия.

### Пример 1.

**Конструктивная инициатива сотрудника.** Сотрудник проявил полезную инициативу (решил особо сложную задачу, подал ценное предложение, проявил особую старательность и т.п.). Ответное действие администрации состоит в том, что она поощряет сотрудника в финансовой или нефинансовой форме. Если поощрение много ниже ожидаемого по результату, то оно никак не влияет на ситуацию, или даже может сыграть негативную роль. В этом случае замкнутого контура не возникает. Если поощрение намного выше того, на что рассчитывал сотрудник, то это вызывает у него большой положительный подъем сил и эмоций, он старается впредь и далее быть как можно полезнее фирме. Здесь мы имеем контур с положительной обратной связью и большим коэффициентом усиления. Такой контур не может оставаться в положении равновесия, поэтому движения в нем направлены в сторону повышения активности сотрудника и величины поощрения. В результате ситуация быстро «зашкаливает»: либо сотрудник достигает своей предельной производительности и фирма больше не имеет оснований для повышения его зарплаты, либо фирма достигает предельной возможности материального и морального стимулирования и сотрудник более не имеет основания для повышения активности.

**Конструктивная инициатива предприятия.** Предприятие решает повысить зарплату сотруднику в одностороннем порядке. В ответ сотрудник начинает более старательно относиться к порученному заданию, подавая повод к дальнейшему увеличению зарплаты. Если коэффициент передачи больше единицы, мы приходим к той же конструкции, что была рассмотрена выше. Конструкция отношений в обоих случаях одинаковая, а отличаются только места зарождения начального сигнала.

**Резюме: положительная связь при положительном начальном движении движется в позитив до насыщения.** Это – неустойчивая система, управление

утрачивается, поскольку при насыщении невозможна положительная динамика поощрений, следовательно, сотрудник рано или поздно начнет воспринимать отсутствие поощрений как наказание, и его действия будут далее все менее и менее эффективными. Но **положительная связь при отрицательном начальном движении движется в негатив до насыщения.** То есть работник и работодатель будут максимально недовольны друг другом, эффективность труда и его оплата будут двигаться к минимуму.

**При малом коэффициенте замкнутого контура нарушения устойчивости даже при положительной обратной связи не возникает:** повышение зарплаты столь незначительно, что сотрудник не считает своим долгом отреагировать на него повышением производительности.

### Пример 2.

**Промашка сотрудника.** Допустим, сотрудник совершает промашку, и фирма наказывает его. Один из таких примеров – возмещение убытков за счет сотрудника. **Если коэффициент усиления по контуру маленький,** то есть наказание много меньше вины, **то опять в этом случае замкнутого контура не возникает.** Если же наказание велико, то сотрудник озлобляется, и начинает активно разрушать устои существования фирмы. Эта ситуация крайне опасная, поскольку озлобленный сотрудник может вынашивать план мщения долгое время и нанести ответный удар в тот момент, когда он будет наиболее чувствительный для фирмы. Отметим, что по определению, раз наказание превысило вину сотрудника, то и месть должна превысить величину наказания, с позиции конструкции взаимоотношений

**Промашка фирмы.** Временное снижение заработной платы может быть вызвано материальными трудностями. В условиях инфляции, роста цен, отсутствие повышения зарплаты в течение длительного времени может рассматриваться сотрудником как отрицательная инициатива предприятия. Тот же эффект возникает, когда зарплата остается неизменной или мало растет в сравнении с ростом зарплаты других сотрудников этой же фирмы или людей этой же профессии на других предприятиях. Мы опять имеем ту же конструкцию: если коэффициент усиления мал, то циркулирующие в системе сигналы не имеют особого значения, и систему можно рассматривать как разомкнутую. Небольшая обида сотрудника может не выразиться ни в какой форме протеста. Если же обида велика, то сотрудник открыто идет на деструктивные отношения: гласно заявляет о своем несогласии с линией фирмы, либо подбивает коллег к бунту, а в ряде случаев грозит увольнением или

даже увольняется. Поскольку наиболее квалифицированные кадры могут легче устроиться в другой фирме, то в первую очередь увольняются именно они. Для коммерческого предприятия тяжесть ухода ценного специалиста может многократно усугубиться ещё и тем, что он переходит к конкурентам, унося с собой не только личный опыт, но и знание секретов и конфиденциальных сведений фирмы.

**Резюме:** положительная связь с большим коэффициентом разрушает устойчивость в системе. Не следует в значительной мере поощрять за каждый успех и наказывать на каждое поражение. Маленький коэффициент обратной связи характерен для предприятий социалистического периода, когда сотрудник был уверен, что ему не грозит ни большая премия, ни серьёзное наказание, и не находил для себя особых мотивов ни для увеличения усилий, ни для того, чтобы всерьёз опасаться совершать какие-либо ошибки.

Как бы ни казалось парадоксальным, оказывается, что **для устойчивого существования фирмы необходима не положительная, а отрицательная обратная связь**. Конечно, это не следует понимать таким образом, что провинившегося сотрудника следует поощрять, а отличившего наказывать, хотя в условиях полного хаоса и анархии даже такая методика, как ни странно, скорее приводит к положительному результату, чем система с положительной обратной связью и большим коэффициентом. Но здесь дело в том, что система с большим положительным коэффициентом в принципе не может быть устойчивой никогда, она всегда движется либо вверх, либо вниз, либо раскачивается с увеличивающейся амплитудой, а в условиях «ограниченной величины сигналов» быстро приходит к «насыщению». То есть производственные отношения либо быстро переходят в разряд торговли услугами «сначала деньги – потом стулья», либо распадаются с накопленными обидами. В межличностных отношениях эта модель проявляется в явлениях лавинообразно нарастающей приязни или неприязни, причем и тот и другой путь могут проходить одни и те же личности. Не даром говорят: «от любви до ненависти – один шаг». Понятно, что когда арсенал средств похвал и премий будет исчерпан, наступит некоторый период упоения, после чего охлаждение, а затем – период раздражения друг другом.

Первый и основной вывод, который следует сделать, состоит в том, что система, безусловно, будет устойчивой, если она описывается отрицательной обратной связью с небольшим коэффициентом усиления. **Но устойчивость системы не тождественна устойчивости фирме.** То есть фирма наряду с

сохранением коллектива заинтересована в росте производительности индивидов, а каждый индивид заинтересован в повышении личных доходов.

Поскольку даже слабая отрицательная связь душит инициативу, то есть уменьшает коэффициент усиления объекта в сравнении с объектом, не охваченным обратной связью, такую связь следует признать неоптимальной. Вместе с тем, отсутствие отрицательной обратной связи опасно для устойчивости коллектива.

**Отрицательную обратную связь можно было бы допустить при условии положительного внешнего воздействия, и, вероятно, это одна из целесообразных моделей системы.**

Речь идет о том, что фирма планомерно повышает зарплату своим сотрудникам. Но с целью того, чтобы не выходить на насыщения по величине платежеспособности, сотрудникам, особо выдающимся, наряду с полагающимся им сверхвысоким поощрением одновременно за непроизводительную вину (а таковая всегда находится) назначается наказание. А сотрудников, давно ничем не блеснувших, поощряют авансом за какие-то личные качества непроизводительного плана (или из жалости к его социальному положению). Здесь мы наблюдаем макиавеллевский принцип награждения не причастных и наказания невинных, который успешно применялся многими тиранами, начиная от древнейших времен до самых новейших (Александр Македонский, Юлий Цезарь, Октавиан Август, Петр Первый, Иван Грозный, Наполеон, Иосиф Сталин и многие другие). Поднятие к престолу ничтожеств и свержение знати чрезвычайно способствовало укреплению личной власти монарха именно в силу отрицательной обратной связи.

Однако следует признать, что наиболее действенная модель отношений должна строиться на иных принципах. В силу этого, целесообразно рассмотреть модель с положительным коэффициентом меньше единицы и с большим отрицательным коэффициентом по производной (приращению). Такая система устойчива и позволяет вносить внешнее положительное управляющее воздействие с целью планового роста мощности предприятия и его доходов.

Суть модели состоит в том, что доля прибыли, принесенная личным вкладом сотрудника, передается ему в виде премии. **Необходимо и достаточно для устойчивости, чтобы коэффициент был меньше единицы (доля меньше целого).** То есть не вся прибыль, полученная от положительной инициативы должна передаваться в премию. Точно также не весь убыток, нанесенный халатностью, должен

покрываться за счет виновного. Процент того и другого целесообразно устанавливать равный. **Отрицательная связь по производной** состоит в том, что при большом темпе роста опережении уровня доходов какой-либо части сотрудников над средним уровнем темпов его роста в силу большой зависимости их вклада в сумму прибыли необходимо **введение прогрессивного налога** на эту часть прибыли. В противном случае система также может, оставаясь устойчивой, двигаться к линии «зашкаливания». Так обстоят дела с окладами управленческих работников на фоне общего довольно низкого уровня. Это приводит к нежелательному росту напряженности в обществе. Ограничение на скорость роста прибыли никоим образом не ограничивают её потенциальной величины, что хорошо согласуется и с историческими примерами. Массы, как правило, не возражают против капитала как такового, а недовольство вызывает только слишком быстрое сосредоточение крупных капиталов в одних руках. Так автор песни «Бесса ме мучо» составила себе многомиллионный капитал на авторских правах на эту песню, и это не вызывает неприязни к ней, хотя можно с уверенностью предположить, что если бы она попыталась продать эту песню хотя бы даже и за сто тысяч, едва ли бы это предложение было встречено кем-либо с пониманием. Другой поучительный пример демонстрирует случай с пареньком, в обязанность которого входило переключать клапан в паровом котле Ватта. Он изобрел рычаг, который делает это автоматически, восторженный Ватт назначил его руководителем специально созданного отдела изобретений и установил ему достаточный высокий оклад, однако этот человек за всю свою жизнь больше ничего не изобрел. Причиной этого было, вероятно, полное отсутствие стимулов.

Таким образом, первые шаги на пути оздоровления производственно - материальных отношений на малом предприятии состоят в следующем:

1. Установление и активное применение нормативов поощрения за полезную инициативу по результатам реального роста прибыли в виде некоторой доли.
2. Установление и применение при необходимости нормативов долевых вычетов за ущерб от халатных действий или бездействия.
3. Контроль за темпом роста прибыли ключевых сотрудников и коррекция коэффициента по пунктам 1 и 2 для установления желательных пропорций между темпами роста благосостояния отдельных сотрудников и темпом роста процветания фирмы.

4. Внесение инициативы (управляющего воздействия) по росту среднего уровня заработной платы сотрудников, исходя из конъюнктуры и с учетом инфляции.
5. Использование слабостей конкурентов для переманивания ключевых сотрудников.
6. Установление информационных заслонов о величине заработной платы между отдельными сотрудниками (поскольку отсутствие её роста при информации о наличии роста у коллеги рассматривается как спад).
7. Оповещение о положительных примерах поощрения: относительно полезности дополнительных успехов и личной инициативы сотрудников.
8. Широко использование нематериальных методов поощрения (и порицания). В этом случае целесообразно следить, чтобы поощрение исходило от представителя администрации, олицетворяющего фирму в глазах сотрудника, а порицание замыкать через коллектив.
9. Постоянно поддерживать в сотрудниках уверенность в том, что фирма имеет дополнительные резервы для поощрения особо ценных сотрудников, но эти резервы ограничены, и возможность их увеличения определяется личным вкладом каждого сотрудника.
10. Привлечение в фирму близких родственников ключевых сотрудников на вспомогательные должности, создание династий, что препятствует перетеканию ценных кадров к конкурентам под влиянием сиюминутных интересов или эмоций.
11. Создание дополнительных коллективных образований с общими интересами, руководство торжественными мероприятиями с привлечением в них небольшой доли прибыли (не более 1-1,5% фонда заработной платы одного месяца). Препятствие созданию непроизводительных коллективных образований, противопоставляющих себя руководству.

## **ЗАКЛЮЧЕНИЕ**

Теория автоматического регулирования содержит огромный потенциал для новых подходов во многих отраслях науки. Это касается и фундаментальной науки, и прикладной, технических и гуманитарных наук, исторических и философских.

В силу этого получение высшего образования в области автоматике полезно в любом случае, не зависимо от сферы дальнейшей профессиональной деятельности.

**ЛИТЕРАТУРА**

[1] Патент на полезную модель № 107346. Интерферометр. Правообладатель: Учреждение Российской академии наук Сибирское отделение РАН Институт лазерной физики (Ru), Автор: Жмудь Вадим Аркадьевич (Ru), Заявка № 2011100243, приоритет полезной модели от 11 января 2011 г., зарегистрировано в Государственном реестре полезных моделей Российской Федерации 10 августа 2011 г., срок действия истекает 11 января 2021 г. Опубликовано в бюллетене №22, 10.08.2011. Класс МПК G01 В 9/02 (2006.01).

[2] Zhmud V.A., Bugrov S.V. The modeling of the electron movements inside the atom on the base of the non-quantum physics. // Proceedings of the 18th IASTED International Conference "Applied Simulation and Modeling" (ASM 2009). Sept. 7-9, 2009. Palma de Mallorca, Spain. P.17 – 23.

[3] V.A. Zhmud. On the Relativist Corrections of Processed Data from the Global Cosmic Systems GPS and GLONASS // Proceedings of DST-RFBR-Sponsored Second Indo-Russian Joint Workshop on Computational Intelligence and Modern Heuristics in Automation and Robotics. NSTU, Novosibirsk, Russia. 9th – 12th September 2011. Новосибирск, НГТУ. pp. 251–255.

[4] V.A. Zhmud. About the essence of the Relativist concept to the Corrections of Processed Data from the Global

Cosmic Systems GPS and GLONASS// Proceedings of RFBR and DST Sponsored "The 2-nd Russian-Indian Joint Workshop on Computational Intelligence and Modern Heuristics in Automation and Robotics", 10 – 13 September, 2011, Additional volume, pp.74–101.

[5] Бугров С.В., Жмудь В.А. Моделирование нелинейных движений в динамических задачах физики // Сборник научных трудов НГТУ. Новосибирск. 2009. 1(55). С. 121 – 126.

[6] Жмудь В.А. Обоснование нерелятивистского некантового подхода к моделированию движения электрона в атоме водорода // Сборник научных трудов НГТУ. Новосибирск. 2009. 3(57). С. 141 – 156.

[7] Харольд Вармус. Обратная транскрипция. В мире науки (Scientific American). 1987. № 11. с. 22–29.



Вадим Жмудь – заведующий кафедрой Автоматики в НГТУ, профессор, доктор технических наук, автор 200 научных статей. Область научных интересов и компетенций – теория автоматического управления, электроника, лазерные системы, оптимизация, измерительная техника.  
E-mail: [оao\\_nips@bk.ru](mailto:оao_nips@bk.ru)

# Будущее программы двойных магистерских дипломов по автоматике: взаимодействие НГТУ с вузами России, Украины, Болгарии, Чехии и Франции

В.А. Жмудь

**Аннотация:** Обсуждаются перспективы сближения программ подготовки по направлениям подготовки в области автоматике, робототехники и мехатроники в российских, украинских, болгарских и чешских вузах для реализации программы двойных дипломов.

**Ключевые слова:** автоматика, мехатроника, робототехника, высшее профессиональное образование, программа двойных дипломов, магистратура.

## ВВЕДЕНИЕ

Новосибирским государственным техническим университетом (Кафедра Автоматики, Международный отдел, Факультет Автоматики и Вычислительной техники и Факультет Мехатроники и Автоматизации) осуществляется взаимодействие с университетами России, Украины, Болгарии, Чехии, Франции с целью реализации программ двойных дипломов в рамках гранта TEMPUS (MPAM).

Суть программы состоит в том, чтобы предоставить студентам возможность обучения параллельно в двух вузах, один из которых – из дальнего зарубежья (Болгария, Чехия, Франция). Программа подготовки магистра включает четыре семестра (два года). Каждый из семестров должен дать студенту подготовку, оцениваемую в 30 зачетных единиц (то есть единиц, признанных в европейской и американской системах для оценки объема пройденных учебных курсов). Указанные единицы называются иногда «кредитами», что является фонетической калькой англоязычного термина. Один «кредит» соответствует приблизительно 36 часам, что при 18-недельном семестре можно приравнять к двум часам занятий в неделю. Зачастую имеется в виду один аудиторный час в неделю и один час самостоятельной подготовки. Не всегда совпадают понятия «час», поскольку где-то используется «академический час», то есть 45 минут, где-то – обычный. Таким образом, само понятие «кредит» варьируется в разных системах, точно определить объем этой меры не возможно.

Тем не менее, при достижении признания условного равенства понятий «кредит» и достаточно близкой программы подготовки в разных вузах возникает возможность условного приравнивания различных дисциплин. В этом случае разные вузы не только в пределах страны, но и в пределах Европейского сообщества могут зачитывать образование, полученное в других вузах в качестве выполненной части учебной программы. Поэтому студент, обучающийся параллельно в двух вузах может получить дипломы обоих вузов по окончании обучения, то есть после того, как наберет требуемое количество «кредитов» в области профессиональной подготовки.

Идеальная схема предполагает следующий алгоритм обучения. Студент обучается три семестра в одном вузе, а один семестр – за рубежом в техническом вузе, реализующем близкую по направлению программу подготовки. По окончании он защищает дипломную работу в обоих вузах и получает два диплома.

Автором предприняты попытки предварительного анализа учебных планов для уточнения перспектив сближения программ подготовки по направлениям подготовки в области автоматике, робототехники и мехатроники в вузах перечисленных стран для реализации программы двойных дипломов. Актуальность решения задачи, обозначенной в названии статьи, обосновывается победой программы TEMPUS (MPAM), ориентированной на эти цели.

## 1. ОСНОВНЫЕ ПРОБЛЕМЫ

Первые выводы по результатам анализа и совещания могут быть сформулированы в следующих тезисах.

1. Количество «кредитов» в программах разных стран не совпадает.
2. Структура Учебных планов не совпадает (количество технических и гуманитарных дисциплин).
3. Типовое количество «кредитов» даже для близких по содержанию дисциплин, а также



структура этих дисциплин (лекции, практические занятия, лабораторные работы, самостоятельная работа) также не совпадают.

4. В целом содержание каждого учебного плана направлено на получение близких компетенций (знаний и умений) в близкой области, поэтому различия носят в большинстве случаев формальный характер и связаны с методологией преподавания больше, чем с содержанием направления подготовки.

Поэтому целесообразно сближение учебных планов подготовки и рабочих программ отдельных дисциплин, что позволило бы предоставить возможность студентам получать двойные дипломы.

Если бы учебный план двух университетов разных стран, например, России и Чехии, был бы совершенно идентичен, то любая часть обучения программы могла бы быть пройдена в любом из этих университетов по выбору студента, при этом полный курс обучения соответствовал бы одновременно полному курсу в любом из этих университетов. При условии соглашений и иных правовых документов оба университета могли бы выдавать по окончании учебы диплом о высшем образовании, то есть выпускник сразу по окончании или с незначительной задержкой получал бы два диплома о высшем образовании в разных странах. Это позволило бы ему в дальнейшем более свободно осуществлять свою профессиональную деятельность почти в любой из европейских стран, предъявляя тот диплом, который в данной стране имеет больший вес. В идеале этот процесс мог бы способствовать выравниванию значения дипломов каждой из этих стран, что составляет суть так называемого «Болонского процесса».

Таким образом, наиболее простым решением было бы принятие полностью одинаковых учебных планов. Естественный путь для этого – сопоставление всех программ, анализ достоинств и недостатков, совместная разработка на этой основе наилучшей программы, соединяющей достоинства и исключающей недостатки каждой из программ.

К сожалению, основным препятствием такого пути являются:

1. Фактически достигнутые положительные изменения учебных планов тех стран, которые более продвинулись в Болонском процессе. С их позиции изменять учебные планы в сторону сближения с учебными планами стран, находящихся в арьергарде этого процесса означает ухудшение ситуации, на что они, естественно, не согласятся.

2. Стандарты учебных планов, утвержденные на уровне министерств или иных вышестоящих организаций, в частности, перечни обязательных дисциплин, стандарты на количество часов и их распределение по

семестрам и дисциплинам, количество лекционных часов, лабораторных работ, практических занятий и самостоятельной работы, и т. д.

Однако даже в самой неблагоприятной ситуации могут быть найдены пути сближения учебных планов, хотя следует признать, что для окончательного сближения будут все же необходимы решения на уровне министерств и ведомств, а возможно и на уровне договоренностей между государствами (например, о взаимном признании документов о высшем профессиональном образовании).

## **2. ПРЕДЛАГАЕМЫЕ РЕШЕНИЯ**

В данной статье мы не будем касаться правовой стороны вопроса, лежащей вне сферы нашей компетенции, а уделим основное внимание фактической стороне дела сближения учебных планов и рабочих программ.

Первый путь, который можно реализовать, состоит в изменении структуры необязательных дисциплин в сторону их соответствия обязательным дисциплинам университетов, с которыми предполагается реализация программы двойных дипломов. К сожалению, такой путь приемлем лишь для сближения с одним конкретным университетом или, в лучшем случае, с университетами одной страны.

Например, при рассмотрении учебных программ НГТУ и Чешского государственного технического университета, мы можем выявить следующие особенности:

1. Количество семестров в обоих университетах для магистров (master) одинаковое – четыре семестра, 120 кредитов, по 30 кредитов в семестр.

2. В Чешской программе каждый семестр содержит 30 кредитов, состоящих из шести дисциплин, как правило, каждая по пять кредитов. Все дисциплины – технические, отсутствуют философия, иностранный язык и иные дисциплины гуманитарно-философского направления. Имеются обязательные дисциплины. В первом семестре обязательных дисциплин шесть, во втором – четыре, в третьем – в третьем – три, в четвертом – две. Дисциплин по выбору студента в первом семестре нет, во втором – две из шести, в третьем – три из девяти, в четвертом – три из шести.

3. В программе НГТУ, в соответствии со стандартом ФГОС-3, содержится пять обязательных дисциплин. В первом семестре обязательных дисциплин четыре, во втором – три, в третьем – ни одной, в четвертом – две. По выбору студента дисциплин в первом семестре нет, во втором – две из четырех, в третьем – две из четырех, в четвертом – две из четырех.

Часть дисциплин в плане НГТУ преподаются два семестра, кроме того, к обязательным

© **АВТОМАТИКА И ПРОГРАММНАЯ ИНЖЕНЕРИЯ. 2012, №1(1)**

дисциплинам причислены «Дипломное проектирование» и «Дипломная работа» поэтому общее количество обязательных дисциплин – только пять.

Таким образом, можно увидеть, что в Чешской программе обязательных дисциплин пятнадцать, и даже если считать совпадающими указанные две дисциплины, регламентирующие работу над дипломом, то остается тринадцать обязательных дисциплин.

Если пойти по пути включения всех обязательных дисциплин Чешской программы в дисциплины по выбору университета и в дисциплины по выбору студента, то и это не будет решением задачи. Действительно, маловероятно обеспечить ответную меру со стороны Чешского университета, то есть убедить руководство факультетом включить обязательные дисциплины Учебного плана НГТУ в перечень дисциплин по выбору Чешской программы.

Вследствие указанных проблем предложение Чешской стороны состояло в замене программы двойных дипломов программой

последовательного получения сначала дипломов российских или украинских университетов, затем – после дополнительного семестра (или более) обучения в Чешском университете – получение диплома Чешского университета. Такое предложение следует признать не соответствующим сути программы двойных дипломов. При реализации этой схемы студент по окончании НГТУ фактически должен быть отчислен с выдачей диплома. То есть студент должен утратить связь с НГТУ. Дальнейшее его обучение в Чехии или в ином университете с целью получения дополнительного высшего образования на базе имеющегося диплома не является реализацией программы двойных дипломов. Кроме того, тут могут возникнуть дополнительные проблемы, в частности, с призывом в ряды ВС РФ, а также с получением финансовой поддержки для получения образования. Таблица 1 позволяет сопоставить дисциплины программ НГТУ и Чешского университета в их распределении по семестрам. В Таблицах 2–4 сопоставляются дисциплины программ НГТУ по содержанию.

**Таблица 1. Программы первого семестра Чешского университета и НГТУ**

Основные индикаторы	Программа Чешского университета				Программа НГТУ			
	1	2	3	4	1	2	3	4
Семестр	6	6	6	5	8	8	6	3
Дисциплин	6	6	6	5	6	7	6	3
Технических дисциплин	30	30	30	30	30	30	30	30
Кредитов	14	14	14	14	19	18	10	0
Неделя	6	4	3	2	4	3	0	2
Обязательных дисциплин	0	2	3	3	4	5	6	1
Дисциплин по выбору кафедры или студента								

**Таблица 2. Аналоги дисциплин и курсов НГТУ и Чешского университета**

НГТУ	Чешский университет
Математическое моделирование объектов и систем управления / Системы автоматической оптимизации	Applied Mathematics / Simulation of Electromechanical Systems / Applications of Automatic Control Methods
Случайные процессы в системах автоматического управления	Probability and Statistics
История и методология науки в области управления / Современные проблемы теории управления	Automatic Control / Algebraic Methods for Control Design / State Space Control Systems / System Identification / Automatic Control of Production Machines / Non-linear systems
Электромеханические системы управления	Drives and Servomechanisms
Компьютерные и сетевые технологии	Computer based Signal Processing
Автоматизированное проектирование средств и систем управления	Grammars and Automata
Научно-методологический семинар	Diploma Work Seminary
Компьютерные технологии управления в технических системах	Computer Control Systems / Digital Control / Database Systems
Многоканальные управляющие системы	Multivariable Control Systems
Адаптация в технических системах	Adaptive Control
Автоматизированное проектирование информационных систем	Design of Automation Systems
Промышленные сети автоматизации / Методы дистанционной обработки данных / Методы и алгоритмы обработки сигналов и изображений / Беспроводные технологии передачи данных	Coding and Cryptography
Программные и аппаратные средства автоматизи	Hardware software co-design
Защита выпускной работы	Term Project / Diploma thesis

**Таблица 3. Дисциплины и курсы Чешского университета без аналогов**

<b>НГТУ (Комментарий)</b>	<b>Чешский университет</b>
Физика для бакалавров	Electromagnetic field and Optics
Физика для бакалавров	Continuum mechanics and thermodynamics
Электроника для бакалавров	Electronics Circuits and Components
Выбор студентов	Technical Diagnostics / Robotics / Image Processing Photonics / Machine Vision / Non-Contact Measurement Methods / Intelligent Robotics / Control systems of Vehicles

**Таблица 4. Дисциплины НГТУ без аналогов**

<b>НГТУ</b>	<b>Комментарий</b>
Научно-исследовательская практика	Без комментариев
Научно-исследовательская работа	
Педагогическая практика	
Государственный экзамен по курсу	Выбор студентов
Системы с распределенными параметрами	Требуется в НГТУ для аспирантуры
Философия	Требуется в НГТУ для аспирантуры
Иностранный язык	Требуется в НГТУ для аспирантуры

Анализ программы Болгарского университета показывает, что получение двойного диплома достижимо более простым путем в этом случае. Программа Болгарского университета соответствует только девяносто кредитам, реализуется в три семестра и содержит только шесть обязательных дисциплин. Все двенадцать дисциплин реализуются в первый и второй семестры, в третьем семестре осуществляется только работа над дипломом.

Таким образом, если Болгарский университет включает в перечень своих дисциплин по выбору обязательные дисциплины программы НГТУ, а НГТУ включает в перечень своих дисциплин по выбору обязательные дисциплины Болгарского университета, то задача решена. Студент, выбравший эти дисциплины при обучении в НГТУ, фактически проходит за период обучения в НГТУ все необходимые и все выборные дисциплины, которые содержатся в программе Болгарского университета. То есть у студента появляется шанс в итоге реализации программы НГТУ претендовать на получение двух дипломов при условии прохождения преддипломной практики в Болгарском университете и выполнения двух защит – по одной в каждом из университетов. При этом у студента из Болгарии не возникает возможности по результатам обучения только в Болгарском университете получить оба диплома, поскольку программа Болгарского университета меньше на 30 кредитов программы НГТУ. Таким образом, приближения к взаимному признанию дипломов обеих стран все-таки не достигается, или оно не существенно, так как одностороннее признание дипломов едва ли может быть осуществлено.

Другим эффективным решением является «индивидуальная образовательная траектория», то есть утверждение учебного плана подготовки конкретного студента в каждом конкретном случае.

Уже в 2014 году кафедра Автоматики НГТУ планирует отправить за счет программы TEMPUS (MPAM) одного или двух студентов на обучение в Чехию, Болгарию или Францию. Поедут лучшие студенты, владеющие английским языком или языком принимающей стороны в достаточной степени, чтобы проходить обучение в течение семестра. Будет оплачен проезд, проживание, обучение. Программа TEMPUS (MPAM) фактически охватит магистрантов 2013 и 2014 годов поступления. Это является достаточно неплохим стимулом для выпускников бакалавриата НГТУ и других вузов технического направления для поступления в магистратуру по направлению 220400 «Управление в технических системах» на кафедру Автоматики (количество бюджетных мест – 21, количество собственных выпускников бакалавриата – 16, в связи с чем поступление на бюджетные места для выпускников других кафедр и университетом вполне реально).

#### **ЗАКЛЮЧЕНИЕ**

Таким образом, перспективы сближения программ подготовки достаточно привлекательны; возможные пути решения этой проблемы обозначены в данной статье.

Автор выражает благодарность международной службе НГТУ и лично проректору по международным связям профессору Е.Б. Цюю, а также М.В. Михайловой, А.М. Гушиной, Ю.Н. Долгодворову, В.В. Некрасову и сотрудникам кафедры Автоматики НГТУ.

**Вадим Жмудь** – заведующий кафедрой Автоматики в НГТУ, профессор, доктор технических наук, автор 200 научных статей. Область научных интересов и компетенций – теория автоматического управления, электроника, лазерные системы, оптимизация, измерительная техника.  
E-mail: [oa0\\_nips@bk.ru](mailto:oa0_nips@bk.ru)

## Требования к публикациям в научном электронном журнале «Автоматика и программная инженерия»

**Тематика журнала:** автоматика, робототехника, автоматизированные системы, программные системы и средства (программная инженерия), защита информации, сетевые технологии, Национальная программная платформа (НПП), облачные технологии, инновации, информационные технологии, моделирование систем, свободное и открытое программное обеспечение.

**Целевая аудитория журнала:** научные, педагогические и инженерные кадры, аспиранты, докторанты и студенты, представители бизнеса в области информационных технологий и автоматике.

### Основные требования.

К опубликованию принимаются оригинальные научные статьи, ранее не опубликованные ни в каких средствах информации, включая научные и иные журналы и электронные сайты.

Не допускается дословное цитирование без кавычек из чужих источников (плагиат) и из своих источников (автоплагиат), допускается опубликование ранее опубликованных иллюстраций и уравнений, если это требуется для понимания текста. Иллюстрации из чужих источников должны быть снабжены ссылками на эти источники.

Статья должна содержать новые научные результаты (соответствовать требованию научности) и соответствовать тематике журнала.

Научный журнал Автоматика и программная инженерия (АиПИ) издается на двух языках: русский и английский.

Авторам предлагается присылать статьи на адрес заместителя главного редактора [oao\\_nips@bk.ru](mailto:oao_nips@bk.ru) свои публикации в формате Word-2003 на двух языках. Допускается присылать статьи для первого рецензирования на одном языке (любом) с последующим обязательным предоставлением окончательной версии статьи на двух языках.

Иллюстрации – черно-белые или цветные (цветные предпочтительно).

Подписи на иллюстрациях также должны делаться на двух языках, или только на

английском (в этом случае в русскоязычном варианте статьи в подрисуночных подписях следует давать перевод этих подписей).

Формат страницы: А4, текст в две колонки, шрифт Times New Roman, размер 10 пт через один интервал, заголовок – в одну колонку шрифт 24 пт, литература – 8 пт. Информация об авторах – в конце статьи, желательна фотография, электронная почта. Перечень авторов – на первой странице (только фамилии, И.О. и место работы).

Статья обязательно должна содержать аннотацию и ключевые слова, литературные ссылки, необходимые для понимания статьи и/или с указанием источников цитирования.

Присланные статьи подвергаются внутреннему анонимному рецензированию. Редакционная коллегия вправе отклонить статью от опубликования с указанием причин этого отклонения или предложить доработать статью по замечаниям рецензента.

Ответственность за отсутствие плагиата, за научное содержание статей и за качество перевода на английский язык лежит на авторе (авторах) публикации.

Российские авторы должны присылать перед окончательным опубликованием статьи скан-файл экспертного заключения о возможности открытого опубликования статьи в ее окончательном виде.

Все вопросы по опубликованию можно уточнить по электронной почте.

В дальнейшем все формальные требования к материалам будут опубликованы на сайте журнала и/или в одном из номеров журнала.

Статьи, написанные в соавторстве действительных членов РАН или РАЕН, а также статьи, сопровождающиеся положительной рецензией действительных членов РАН или РАЕН (твердая копия с оригинальной подписью, заверенной по месту работы, пересылается по почте), могут быть опубликованы без дополнительного рецензирования. При этом не исключается предложение редакции о редакторской правке и/или о лучшем качестве иллюстративного материала.

## Automatics and Program Engineering 2012. N 1 (1) Content

### I. Automatics / Automated Systems / Measuring Means and Systems

1. V. Zhmud, A. Zavorin, A. Polishchuk, O. Yadrishnikov. Analysis of the Design Method of Robust Regulator by Means of Double Iterative parallel numerical Optimization. P. 7–16.
2. V. Zhmud, D. Tereshkin, A. Liapidevskiy, A. Zakharov. Radio Frequency Method for the Measuring of Supersmall Displacements and Vibrations. P. 17–34.
3. E. Yu. Kutenkova, B.N. Rakhimov, T.V. Larina, Sh. I. Madumarov. Research of optoelectronic method and development of devices for liquid media controlling and monitoring. P. 35–42.
4. T.V. Larina, E. Yu. Kutenkova, V.A. Zhmud, D.D. Alijanov. Optoelectronic method of the control of physical and chemical parameters of hard metals surfaces. P. 43–48.

### II. Robotics / Program Means and Systems / Free Software / Open Software

5. A. Kolker, D. Livenets, A. Kosheleva. Justification of the Choice of Software for Robotics. P. 51–64.
6. A. Kolker, D. Livenets, A. Kosheleva. Development of Control Systems of Stand for Learning of the Transient Process in the High Vacuum Switch. P. 65–72.
7. A. Kolker, D. Livenets, A. Kosheleva, V. Zhmud. The Development of the Unit for the Robot Controlling System with the using of Engineering Software SciLab. P. 73–83.
8. A. Kolker, D. Livenets, A. Kosheleva, V. Zhmud. The Researching of the Variants for the Creation of Intellectual Robotic Systems on the Base of Single-Plate Computers and Free Operation Systems. P. 84–98.

### III. Innovation Technologies / Future Technologies / Modeling of Systems / Cloud Technologies / National Program Platform

9. V. Zhmud, A. Liapidevskiy, A. Podolets. The Advantages of the Free Software for the Strategy Technologies. P. 101–106.

### IV. Popular Papers / Professional Orientation / Information for Authors

10. V. Zhmud. Apology of the Theory of Automatic Control. P. 109–133.
11. V. Zhmud. Future of the double diploma program on Automatics: the collaboration with Universities of Russia, Ukraine, Bulgaria, Czech, France. P. 134–137.
12. Requirements for electronic publications in the scientific journal “Automatics and Program Engineering” P. 138.



## Automatics and Program Engineering: Editing Cometeec

**Head of the Editing Cometeec – Professor Vladimir Guzhov,**  
Novosibirsk State Technical University (NSTU)

**Vice-Head of the Editing Cometeec – Professor Vadim Zhmud,**  
NSTU, OAO “Novosibirsk Institute of Program Systems” (NIPS)

**Responcible Secretary - Professor Galina Frantsuzove, NSTU**

### Editing Comiteec:

**Anatoliy Vostrikov** Professor, Department of Automation, NSTU

**S.S. Mahapatra** Professor in the Department of Mechanical Engineering, NIT  
Rourkela, **India**

**Hubert Roth** Head of the Department of Automatic Control Engineering of  
University if Siegen, **Germany** , professor

**Vasiliy Gubarev** Professor, Department of Computer Techniques, NSTU

**Alexander Voevoda** Professor, Department of Automation, NSTU

**Evgeniy Rabinovich** Professor, Department of Computer Techniques, NSTU

**V. J. Savsani** Associate Professor at B. H. Gardi college of engineering and  
technology, Surat, India

**Boris Borisov** Head of Laboratory in Institute of Laser Physocs of Siberean  
Branch of Russian Academy of Sciences, Professor

**Boris Poller** Head of Laboratory in Institute of Laser Physocs of Siberean  
Branch of Russian Academy of Sciences, Professor

**Alexander Liapidevsliy** Director of NIPS, deputy-professor

**Vladimir Gololobov** Head of Laboratory in NIPS, deputy-professor

**Mikhail Kalinin** Head of Department in NIPS

**Nimatzhon Rakhimov** Head of the Laboratory of Optoelectronic methof of non-  
distructing control, Siberean State Geodesy Academy, Professor