#### Федеральное государственное учреждение «Федеральный научный центр Научно-исследовательский институт системных исследований

Российской академии наук»

(ФГУ ФНЦ НИИСИ РАН)

### ТРУДЫ НИИСИ РАН

#### TOM 11 № 3

## МАТЕМАТИЧЕСКОЕ И КОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ СЛОЖНЫХ СИСТЕМ:

ТЕОРЕТИЧЕСКИЕ И ПРИКЛАДНЫЕ АСПЕКТЫ

MOCKBA 2021

#### Редакционный совет ФГУ ФНЦ НИИСИ РАН:

В.Б. Бетелин (председатель), Е.П. Велихов, С.Е. Власов, В.А. Галатенко, В.Б. Демидович (отв. секретарь), Ю.В.Кузнецов (отв. секретарь), Б.В. Крыжановский, А.Г. Кушниренко, А.Г. Мадера , М.В. Михайлюк, В.Я. Панченко, В.П. Платонов

#### Главный редактор журнала:

В.Б. Бетелин

#### Научный редактор номера:

А.Г. Мадера

#### Тематика номера:

Моделирование физических процессов в микро- и наноэлектронике, математические исследования, многопроцессорные вычислительные системы, математическое моделирование и визуализация систем виртуальных изображений, информационные технологии в учебной информатике

Журнал публикует оригинальные статьи по следующим областям исследований: математика, математическое и компьютерное моделирование, обработка изображений, визуализация, системный анализ, методы обработки сигналов, информационная безопасность, информационные технологии, высокопроизводительные вычисления, оптико-нейронные технологии, микро- и наноэлектроника, вопросы численного анализа, история науки и техники.

#### The topic of the issue:

Modeling of physical processes in micro- and nanoelectronics, mathematical issues, multiprocessor computer systems, mathematical modeling and visualization of virtual image systems, information technology in educational informatics

The Journal publishes novel articles on the following research arias: mathematics, mathematical and computer modeling, image processing, visualization, system analysis, signal processing, information security, information technologies, high-performance computing, optical-neural technologies, micro- and nanoelectronics, problems of numerical analysis, history of science and of technique.

Заведующий редакцией: В.Е.Текунов

Издатель: ФГУ ФНЦ НИИСИ РАН, 117218, Москва, Нахимовский проспект 36, к. 1

© Федеральное государственное учреждение «Федеральный научный центр Научно- исследовательский институт системных исследований Российской академии наук», 2021 г.

### СОДЕРЖАНИЕ

1. МОДЕЛИРОВАНИЕ ФИЗИЧЕСКИХ ПРОЦЕССОВ В МИКРО- И
НАНОЭЛЕКТРОНИКЕ
А.Г. Мадера. Корреляционный анализ стохастической нестационарной темпера-
туры микросхем5
А.Г. Мадера, Е.К. Гребенникова. Тепловая обратная связь в электронных системах
и её математическое моделирование
Е.К. Гребенникова. Алгоритм компьютерного моделирования тепловой обратной
связи в электронных системах
Н.В. Масальский. Влияние физических параметров на ВАХ цилиндрических КНИ
КМОП нанотранзисторов с полностью охватывающим затвором
ІІ. МАТЕМАТИЧЕСКИЕ ИССЛЕДОВАНИЯ
М. М. Петрунин. О некоторых особенностях вычислений в проблеме периодично-
сти непрерывных дробей элементов эллиптических полей
Ю.Н. Штейников. Множество частных и Сидоново свойство
ІІІ.МНОГОПРОЦЕССОРНЫЕ ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ
А.Н. Годунов, И.И. Хоменков, В.Г. Щепков. Синхронизация системных часов в мно-
гопроцессорных системах 29
IV. МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ И ВИЗУАЛИЗАЦИЯ СИ-
СТЕМ ВИРТУАЛЬНОГО ОКРУЖЕНИЯ
А.В. Мальцев. Синтез изображений трехмерных сцен с моделированием капель до-
ждя на объективе виртуальной камеры
Е.В. Страшнов. Инверсная кинематика для управления движением двуногих ша-
гающих роботов в системах виртуального окружения41
V. ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В УЧЕБНОЙ
ИНФОРМАТИКЕ
М.С.Дьяченко, А.Г. Леонов, М.А. Матюшин. Исследование и разработка методов ма-
шинного обучения и архитектур нейронных сетей для применения в области про-
верки
М.С.Дьяченко, А.Г. Леонов, Д.А. Сокунов. Элементы искусственного интеллекта при
автоматизации тестирования веб-интерфейсов на примере ЦОП Мирера54
А.Г. Леонов, А.Е. Орловский. Методы интеграции цифровых образовательных сред
в цифровую образовательную платформу Мирера59

#### **CONTENT**

#### I. MODELING OF PHYSICAL PROCESSES IN MICRO- AND NANO-ELECTRONICS

A.G. Madera. Correlation analysis of stochastic non-stationary temperature of microcir-
cuits5
A.G. Madera, E.K. Grebennikova. Thermal feedback in electronic systems and its mathematical modeling
E.K. Grebennikova. Algorithm of computer simulation of thermal feedback in electronic
systems
N.V. Masalsky. Effects of the physical parameters on the I-V data of cylindrical surround-
ing gate SOI CMOS nanotransistor
VI.MATHEMATICAL ISSUES
M.M. Petrunin. On some features of calculations on the problem of periodicity of contin-
ued fractions of elements of elliptic fields20
Y. N. Shteinikov. Quotient set and Sidon property27
VII. MULTIPROCESSOR COMPUTER SYSTEMS
A.N. Godunov, I.I. Khomenkov, V.G. Shchepkov. System clock synchronization for multi-
viii. MATHEMATICAL MODELING AND VISUALIZATION OF VIR-
TUAL IMAGE SYSTEMS
A.V. Maltsev. Image Synthesis of 3D Scenes with Simulation of Raindrops on Virtual
Camera's Lens
E.V. Strashnov. Inverse Kinematics for Motion Control of Bipedal Walking Robots in
Virtual Environment Systems
IX. INFORMATION TECHNOLOGY IN EDUCATIONAL INFORMATICS
M.S. Dyachenko, A.G. Leonov, M.A. Matushin. Research and development of machine
learning methods and neural network architectures for usage in the field of verifica-
tion48
M.S. Dyachenko, A.G. Leonov, D.A. Sokunov. Elements of artificial intelligence in the
automation of testing web interfaces on the example of the DEP Mirera54
A.G. Leonov, A.E. Orlovskii. Methods of integration of digital educational environments
into digital educational platform Mirera59

## Корреляционный анализ стохастической нестационарной температуры микросхем

#### А.Г. Мадера<sup>1</sup>

<sup>1</sup>ФГУ НИИСИ РАН, Москва, РФ, alexmadera@mail.ru;

Аннотация. В статье проведен корреляционный анализ стохастической нестационарной температуры кристалла интегральной микросхемы и получены уравнения для математического моделирования ее математического ожидания, дисперсии и корреляционных моментов. Анализ проведен при стохастическом характере полного теплового сопротивления МС в корпусе, температуры среды или охлаждающей жидкости, начальной температуры МС и мощности ее потребления. Сравнение расчетных и экспериментальных данных для установившейся температуры кристалла МС показывает, что наибольшая температура рассчитанная вероятностными методами хорошо согласуется с данными эксперимента, в то время как пренебрежение ее стохастическим характером приводит к существенно большим погрешностям достигая 70%.

**Ключевые слова:** стохастический, нестационарный, температура, микросхема, математическое ожидание, дисперсия, корреляция

#### 1. Введение

При проектировании электронной аппаратуры с применением интегральных микросхем (МС) необходимо оценивать нестационарную температуру кристалла МС, а также время, за которое кристалл МС выйдет на заданный тепловой режим.

При этом, как правило полагают, что рассеиваемая мощность или мощность потребления МС и полное тепловое сопротивление МС в корпусе принимают свои наибольшие или номинальные (типовые, средние) значения. Получаемые при таком подходе температуры могут сильно отличаться от своих реальных значений, встречаемых на практике.

Это обусловливается тем, что параметры, определяющие тепловой режим МС (мощность, тепловые сопротивления МС, температура среды и т.д.) обладают существенным статистическим разбросом, вызываемым как статистическим разбросом технологии изготовления и монтажа МП в системе, так и статистическим разбросом и флуктуациями параметров окружающей среды. Следовательно температура МС является стохастической и ее необходимо анализировать вероятностными методами [1 – 4].

В настоящей работе проводится корреляционный анализ стохастической нестационарной температуры кристалла МС  $T_j(t,\omega)$  при стохастических значениях полного теплового сопротивления МС в корпусе  $R_{jc}(\omega)$ , температуры среды или охлаждающей жидкости  $T_a(\omega)$ , начальной температуры  $T_0(\omega)$  и мощности потребления  $P(t,\omega)$ , где  $\omega \in \Omega$  – элементарные события  $\omega$  из пространства элементарных событий  $\Omega$ .

#### 2. Математическая модель стохастической нестационарной температуры МС

Стохастическое дифференциальное уравнение, описывающее стохастическую нестационарную избыточную температуру кристалла МС  $\theta(t) = T_j(t) - T_a$  (превышение над температурой среды или охлаждающей жидкости) имеет вил

$$\rho c V \dot{\theta}(t, \omega) + \theta(t, \omega) / R_{jc}(\omega) = P(t, \omega), (1)$$
  
$$\theta(0, \omega) = \theta_0(\omega),$$

где  $\dot{\theta}(t) = d\theta/dt$ ;  $\rho cV$  — объемная теплоемкость МС в корпусе,  $\rho$ , c, V — плотность, удельная теплоемкость и объем корпуса МС.

В уравнении (1) случайная величина  $R_{jc}(\omega)$  распределена по логарифмически-нормальному закону [5] с математическим ожиданием  $m_{R_{jc}}$  и дисперсией  $\sigma_{R_{jc}}^2$ ; случайная функция  $P(t,\omega)=P_0(\omega)h(t)$ , где  $P_0(\omega)$  – случайная величина с математическим ожиданием  $m_{P0}$  и дисперсией  $\sigma_{P0}^2$ , функция h(t)=1 при  $t\geq 0$  и h(t)=0 при t<0; случайная величина  $\theta_0(\omega)$  имеет математическое ожидание  $m_{\theta_0}$  и дисперсию  $\sigma_{\theta_0}^2$ .

# 3. Уравнения для математического ожидания, дисперсии и корреляций 11стохастической нестационарной температуры МС

Уравнение для математического ожидания  $m_{\theta}(t)$  стохастической температуры  $\theta(t,\omega)$  по-

лучим применив к уравнению (1) оператор математического ожидания при фиксированном значении  $R_{jc}$ , а затем усреднив полученное условное математическое ожидание  $M(\theta(t,\omega)|R_{jc})$  по случайным значениям  $R_{jc}(\omega)$  в соответствии с логарифмически-нормальным законом распределения, получим

$$\begin{split} m_{\theta}(t) &= m_{P0} m_{R_{jc}} \big( 1 - f(a) \big) + m_{\theta_0} f(a_2), \\ \text{где} \quad a &= a_2 \exp(-\sigma_x^2), \quad a_2 = (t/\rho c V) \exp(-m_x), \\ m_x &= \ln \left( m_{R_{jc}} \right) - \sigma_x^2 / 2, \, \sigma_x^2 = \ln(\sigma_{R_{jc}}^2 / m_{R_{jc}} + 1), \\ f(a) &= \frac{1}{\sqrt{\pi}} \cdot \\ &= \int_{-\infty}^{\infty} exp \big( -a \cdot \exp\left(-\sqrt{2}\sigma_x z\right) \big) \exp(-z^2) \, dz. \end{split}$$
 (3)

Дисперсия стохастической температуры определяется как [6]

$$\sigma_{\theta}^{2}(t) = \overline{\sigma}^{2}(\theta | R_{jc}) + \sigma^{2}(M(\theta | R_{jc})) \tag{4}$$
 где  $\overline{\sigma}^{2}(\theta | R_{jc}) = M_{R_{jc}}(\sigma^{2}(\theta | R_{jc})),$  
$$\sigma^{2}(M(\theta | R_{jc})) = M(M^{2}(\theta | R_{jc}) - m_{\theta}^{2}(t),$$

 $M_{R_{jc}}(\cdot)$  — оператор математического ожидания по R.

Условная дисперсия  $\sigma^2(\theta|R_{jc})$  в выражении (4) определяется из следующей системы уравнений [7] для условных корреляционных моментов  $k_{ij}(t) = M(Y_i^0 Y_j^0 | R_{jc}), i, j = 1,2$ :

$$\dot{k}_{11} = -\frac{2k_{11}}{\rho c V R_{jc}} + \frac{2k_{12}}{\rho c V} \\
\dot{k}_{12} = -\frac{k_{12}}{\rho c V R_{jc}} + \frac{k_{22}}{\rho c V} \\
\dot{k}_{22} = 0$$
(5)

Ввиду независимости случайных величин  $\theta_0(\omega)$  и  $P(t,\omega),\,k_{11}(0)=\sigma_{\theta_0}^2.$ 

Решая систему (5) относительно  $\dot{k}_{11} = \sigma^2(\theta | R_{jc})$ , проводя усреднение по  $R_{jc}$  и подставляя в (4) получаем

$$\begin{split} \sigma_{\theta}^{2}(t) &= M(P^{2})M(R_{jc}^{2})[1 - 2f(a_{1}) + f(2a_{1})] + \\ &+ M(\theta_{0}^{2})f(2a_{2}) + 2m_{P}m_{R_{jc}}m_{\theta_{0}}[f(a) + f(2a)] - \\ &- m_{\theta}^{2}(t), \end{split}$$
 (6)

где  $a_1 = aexp(-\sigma_x^2)$ , f(a) рассчитывается по стандартной программе на компьютере.

Сравнение расчетных и экспериментальных данных для установившейся температуры показывает, что наибольшая температура рассчитанная вероятностными методами хорошо согласуется с данными эксперимента [8], в то время как детерминированная температуры отличается от наибольшей экспериментальной температуры на 35%, ожидаемой на практике – на 70%.

Таким образом, случайный разброс параметров теплового режима МС оказывает существенное влияние на температуру МС и при проектировании МС необходимо учитывать ее стохастичность.

### Correlation Analysis of Stochastic Non-Stationary Temperature of Microcircuits

#### Alexander Madera

**Abstract.** The article presents a correlation analysis of the stochastic unsteady temperature of an integrated microcircuit crystal and obtains equations for mathematical modeling of its mathematical expectation, variance and correlation moments. The analysis was carried out with the stochastic nature of the total thermal resistance of the MS in the housing, the temperature of the medium or coolant, the initial temperature of the IC and the power of its consumption. Comparison of the calculated and experimental data for the steady-state temperature of the IC crystal shows that the highest temperature calculated by probabilistic methods is in good agreement with the experimental data, while neglecting its stochastic nature leads to significantly larger errors reaching 70%..

**Keywords:** stochastic, non-stationary, temperature, microcircuit, mathematical expectation, variance, correlation

#### Литература

- 1. А.Г. Мадера, П.И. Кандалов. Анализ интервально стохастических температурных полей технических систем. «Программные продукты и системы», 2014, № 4, 41–45.
- 2. А.Г. Мадера, П.И. Кандалов. Моделирование температурных полей технических систем в условиях интервальной неопределенности. «Тепловые процессы в технике», Т. 6 (2014), № 5, 225–229.
  - 3. А.Г. Мадера, П.И. Кандалов. Компьютерное моделирование температурных полей техниче-

ских систем при интервально стохастической неопределенности параметров. «Прикладная информатика», Т. 10 (2015), N 1(55), 106–113.

- 4. А.Г. Мадера, П.И. Кандалов. Математическое моделирование интервально стохастических тепловых процессов в технических системах при интервальной неопределенности определяющих параметров. «Компьютерные исследования и моделирование», Т. 8 (2016), № 3, 501–520.
- 5. Н. Хастингс, Дж. Пикок. Справочник по статистическим распределениям. М., Статистика, 1980.
- 6. Г.Я. Мирский. Характеристики стохастической взаимосвязи и их измерения. М., Энергоиздат, 1982.
- 7. A. Madera, P. Kandalov. Thermal Processes in Electronic Equipment at Uncertainty. "Journal of Engineering Thermophysics", V. 29 (2020), N. 1, 170–180.
- 8. А.Г. Мадера, Г.В. Резников. Прогнозирование температуры кристалла по температуре на крышке корпуса ИС. «Электронная техника. Сер. 3: Микроэлектроника», 1987, № 1, 85–88.

# Тепловая обратная связь в электронных системах и ее математическое моделирование

А.Г. Мадера<sup>1</sup>, Е.К. Гребенникова<sup>2</sup>

 $^1$  ФГУ ФНЦ НИИСИ РАН, Москва, Россия, alexmadera@mail.ru  $^2$  ФГУ ФНЦ НИИСИ РАН, Москва, Россия, ekaterina.grebennikova@inbox.ru

Аннотация. В статье рассматривается новый подход к математическому моделированию тепловых процессов в электронных системах с тепловой обратной связью, обусловленной температурной зависимостью мощностей, потребляемых электронными компонентами. Рассматриваемый подход основан на описании тепловой обратной связи матричными нелинейными уравнениями с полиномиальной правой частью. Предлагаемый в статье метод позволяет свести систему матричных нелинейных степенных уравнений к системе независимых несвязанных алгебраических уравнений. Сравнение результатов, полученных разработанным и численным методами, показывает их полное совпадение.

**Ключевые слова:** математическое моделирование, тепловые процессы, тепловая обратная связь, полиномиальное матричное уравнение, спектральное разложение, матричное преобразование подобия.

#### 1. Введение

Математическая модель тепловых процессов в электронных системах (ЭС) в условиях воздействия тепловой обратной связи (ТОС) [1, 2], описывается полиномиальным матричным уравнением HT = P(T), с невырожденной  $N \times N$ -матрицей графа тепловой модели ЭС [3, 4, 5] с элементами  $h_{lm}$ , l, m = 1, 2, ..., N, искомым N-векторм температур  $T = (T_1, T_2, ..., T_N)^T$  в N узлах графа и вектор-функцией мощностей потребления  $P(T) = (P_1(T_1), P_2(T_2), ..., P_N(T_N))^T$  электронных элементов нелинейно зависящих от температуры элементов.

Мощности потребления  $P_i(T_i)$ , i=1,2,...,N, электронных элементов можно аппроксимировать степенными рядами Тейлора или полиномами k-й степени. В рассматриваемом методе мощности потребления  $P_i(T_i)$  представляются в виде полиномом k-й степени , а именно  $P_i(T_i) = p_{0i} + p_{1i}T_i + p_{2i}T_i^2 + \cdots + p_{ki}T_i^k$ , где  $p_{li}$ , l=1,2,...,k, i=1,2,...,N — постоянные коэффициенты

Тогда уравнение HT = P(T) запишется в виде:

$$HT = \begin{pmatrix} P_1(T_1) \\ P_2(T_2) \\ \vdots \\ P_n(T_n) \end{pmatrix} =$$

$$= \begin{pmatrix} p_{01} + p_{11}T_1 + p_{21}T_1^2 + \dots + p_{k1}T_1^k \\ p_{02} + p_{12}T_2 + p_{22}T_2^2 + \dots + p_{k2}T_2^k \\ \vdots \\ p_{0n} + p_{1n}T_n + p_{2n}T_n^2 + \dots + p_{kn}T_n^k \end{pmatrix} (1)$$

В существующей научной литературе отсутствуют аналитические матричные методы получения решения нелинейных матричных уравнений (1). Как правило, для решения подобных математических систем прибегают к различным численным методам, требующим больших вычислений и обладающих рядом существенных ограничений, таких как выбор подходящего шага, скорость сходимости используемого метода, проблема сходимости метода, точности и ошибок вычислений. Зачастую для разрешения подобных уравнений вводят различные допущения, позволяющие сводить исходное нелинейное матричное уравнение (1) к линейному. В то же время, однако, практика показывает, что допущение линейности приводит к существенному несовпадению между аналитическим решением и результатами, полученными численными методами на компьютере.

В данной статье описывается аналитический метод решения нелинейных матричных уравнений с полиномиальной правой частью (1). Метод апробирован при математическом моделировании тепловых процессов с тепловой обратной связью в реальных электронных системах. Сравнительный анализ решений, полученных разработанным методом и известными численными

методами, показывает их полное совпадение.

#### 2. Метод получения решения матричных уравнений с полиномиальной правой частью

Исходное матрично-векторное уравнение (1) с неизвестным вектором Т преобразуем к матричному уравнению относительно неизвестной матрицы T [6, 7, 8]. Для этого представим вектор  $T = (T_1, T_2, ..., T_N)^T$  в виде произведения  $T = \mathcal{T}$ . диагональной  $N \times N$ матрицы T = $diag\{T_1, T_2, ..., T_N\}$  и единичного N-вектора j = $(1 \ 1 \dots 1)^T$ , то есть в виде  $T = (T_1, T_2, \dots, T_N)^T =$  $\mathcal{T}\cdot j$ . Причем для степеней матрицы  $\mathcal{T}$  справедливо  $\mathcal{T}^m = diag\{T_1^m, T_2^m, ..., T_N^m\}, \ m = 1, 2, ... k,$  так что  $(T_1^m, T_2^m, ..., T_N^m)^T = diag\{T_1^m, T_2^m, ..., T_N^m\}j = \mathcal{T}^mj.$ 

Введя векторы  $\mathcal{P}_0 = (p_{01}, p_{02}, ..., p_{0N})^T$ ,  $\mathcal{P}_1 = \sum_{n=0}^{\infty} p_{nN}$  $(p_{11}, p_{12}, ..., p_{1N})^T, ..., \mathcal{P}_k = (p_{k1}, p_{k2}, ..., p_{kN})^T,$ и подставляя их в матрично-векторное уравнение (1), получим матричное уравнение относительно неизвестной диагональной  $N \times N$ -матрицы  $\mathcal{T}$ 

$$H\mathcal{T}j = \mathcal{P}_0 + \mathcal{T}\mathcal{P}_1 + \mathcal{T}^2\mathcal{P}_2 + \dots + \mathcal{T}^k\mathcal{P}_k. \tag{2}$$

Производя преобразование подобия T = $XDX^{-1}$  с неособенной трансформирующей  $N \times$ N-матрицей [8] X, и учитывая, что для степеней  $\mathcal{T}^m$  матрицы  $\mathcal{T}$  справедливы соотношения  $\mathcal{T}=$  $XDX^{-1}$ ,  $\mathcal{T}^2 = XD^2X^{-1}$ , ...,  $\mathcal{T}^k = XD^kX^{-1}$ , ypasнение (2) приведется к следующему виду:

$$HXDX^{-1}j = P_0 + XDX^{-1}P_1 + XD^2X^{-1}P_2 + \cdots + XD^kX^{-1}P_k.$$
 (3)

Для неособенной матрицы Н существует спектральное разложение  $U^{-1}HU = \Lambda_H$  с трансформирующей  $N \times N$ -матрицей подобия U и диагональной  $N \times N$ -матрицей ее собственных значений  $\Lambda_H$  [7, 8].

Представив трансформирующую матрицу подобия X как  $X = U \Lambda_H^{\frac{-1}{2}} V^{-1}$  с неизвестной, трансформирующей  $N \times N$ -матрицей подобия Vи умножив уравнение (3) слева на матрицу

$$V\Lambda_{H}^{\frac{-1}{2}}U^{-1}, \text{получим}$$

$$V\Lambda_{H}^{\frac{-1}{2}}U^{-1} \cdot H \cdot U\Lambda_{H}^{\frac{-1}{2}}V^{-1}DX^{-1}j =$$

$$= V\Lambda_{H}^{\frac{-1}{2}}U^{-1}P_{0} + V\Lambda_{H}^{-1}V^{-1}DX^{-1}P_{1} +$$

$$+V\Lambda_{H}^{-1}V^{-1}D^{2}X^{-1}P_{2} + \dots +$$

$$+V\Lambda_{H}^{-1}V^{-1}D^{k}X^{-1}P_{k}.$$

Поскольку 
$$V\Lambda_H^{\frac{-1}{2}}\underbrace{U^{-1}\cdot H\cdot U}_{\Lambda_H}\Lambda_H^{\frac{-1}{2}}V^{-1} =$$

 $VV^{-1} = I$ , то матрица H оказывается приведен-

ной к диагональному виду с единичной матрицей  $I = diag\{1,1,...,1\}.$ 

Умножая теперь уравнение (4) слева на матрицу  $V\Lambda_HV^{-1}$ , получим

$$V\Lambda_{H}V^{-1}DX^{-1}j = V\Lambda_{H}^{\frac{1}{2}}U^{-1}P_{0} + DX^{-1}P_{1} + D^{2}X^{-1}P_{2} + \dots + D^{k}X^{-1}P_{k}.$$
 (5)

Введем векторы 
$$U^{-1}j=\alpha=\text{ и }U^{-1}P_l=\beta_l=\text{, }l=0,1,2,...\text{, }k,$$

и примем во внимание, что  $X^{-1} = V \Lambda_H^{\frac{1}{2}} U^{-1}$ . Тогда можно записать следующие соотношения:

$$X^{-1}j = V\Lambda_{H}^{\frac{1}{2}}U^{-1}j = V\Lambda_{H}^{\frac{1}{2}}\alpha,$$
  

$$X^{-1}P_{1} = V\Lambda_{H}^{\frac{1}{2}}U^{-1}P_{l} = V\Lambda_{H}^{\frac{1}{2}}\beta_{l},$$
  

$$l = 0,1,2,...,k.$$

Подставляя последние выражения в уравнение (5) получим

$$V\Lambda_{H}V^{-1}DV\Lambda_{H}^{\frac{1}{2}}\alpha = V\Lambda_{H}^{\frac{1}{2}}\beta_{0} + DV\Lambda_{H}^{\frac{1}{2}}\beta_{1} + D^{2}V\Lambda_{H}^{\frac{1}{2}}\beta_{2} + \dots + D^{k}V\Lambda_{H}^{\frac{1}{2}}\beta_{k}.$$
(6)

Примем, чтобы спектральное разложение матрицы V имело вид  $V^{-1}DV = \Lambda_D$ , где  $\Lambda_D =$  $diag\{\lambda_{D1},\lambda_{D2},...,\lambda_{DN}\}$  – диагональная матрица собственных значений матрицы D. Тогда умножая уравнение (6) слева на матрицу  $V^{-1}$  получим окончательное матрично-полиномиальное уравнение

$$\beta_0 + \Lambda_D(\beta_1 - \Lambda_H \alpha) + \Lambda_D^2 \beta_2 + \dots + \Lambda_D^k \beta_k = 0.$$
 (7) относительно неизвестных диагональных элементов матрицы  $\Lambda_D$ .

Ввиду диагональности матриц  $\Lambda_H$  и  $\Lambda_D$  матричное уравнение (7) распадается на N независимых несвязанных между собой полиномиальных уравнений k-го порядка относительно неизвестных собственных значений  $\lambda_{Di}$ , 1,2,...,N, матрицы D, а именно,

$$\beta_{0i} + (\beta_1 - \Lambda_{Hi}\alpha_i)\lambda_{Di} + \beta_{2i}\lambda_{Di}^2 + \dots + \\ + \beta_{ki}\lambda_{Di}^k = 0,$$

$$i = 1, 2, \dots, N.$$
(8)

Полученные уравнения являются условиями, которым должны подчиняться собственные значения  $\lambda_{Di}$ , i = 1, 2, ..., N, матрицы D.

Таким образом искомый вектор температур Т исходного матричного уравнения (1) с полиномиальной правой частью определяются соотно-

$$T = Tj = XDX^{-1}j = U\Lambda_{H}^{-\frac{1}{2}} \underbrace{V^{-1}DV}_{\Lambda_{D}} \Lambda_{H}^{\frac{1}{2}} U^{-1} ==$$
(9)

Отметим, что в выражении (9) трансформирующая матрица подобия *U* известна априори из спектрального разложения матрицы Н исходной системы уравнений (1), а элементы  $\lambda_{Di}$ , i = 1,2, ..., n, диагональной матрицы  $\Lambda_D$ , определяются из решения алгебраических полиномиальных уравнений k-го порядка (9).

Описанный метод позволяет найти искомый вектор температур T матричного нелинейного уравнения (1) с полиномиальной правой частью. Сравнение решений, полученных разработанным и численным методами, показал их полное совпаление.

#### 3. Заключение

В данной статье представлен метод математического моделирования стационарного (установившегося) теплового процесса в ЭС с тепловой обратной связью, описываемый нелинейным матричным уравнением с полиномиальной правой частью (1). В отличии от известных на сегодняшний день методов, разработанный метод является общим аналитическим методом решения, не содержащим ограничений по виду и степени полинома и позволяющим найти за-

мкнутое аналитическое решение матричного нелинейного уравнения (1) в матричном виде. Применение метода к реальным ЭС показывает его высокую точность, адекватность и относительную простоту, что в сравнении с численными методами, обладающими, традиционными и существенными недостатками (проблема выбора шага, проблема сходимости и его скорости, точность, ошибки округления, и пр.), выгодно отличается от последних.

Публикация выполнена в рамках государственного задания ФГУ ФНЦ НИИСИ РАН «Проведение фундаментальных научных исследований (47 ГП)» по теме № FNEF-2021-0001 «Математическое обеспечение и инструментальные средства для моделирования, проектирования и разработки элементов сложных технических систем, программных комплексов и телекоммуникационных сетей в различных проблемно-ориентированных областях. 0580-2021-0001», рег. № 121031300047-6.

## Thermal Feedback in Electronic Systems and Its Mathematical Modeling

#### Alexander G. Madera, Ekaterina K. Grebennikova

**Abstract.** The article considers a new approach to mathematical modeling of thermal processes in electronic systems with thermal feedback due to the temperature dependence of powers consumed by electronic components. The approach under consideration is based on the description of thermal feedback by matrix nonlinear equations with a polynomial right-hand side. The method proposed in the article allows us to reduce the system of matrix nonlinear power equations to a system of independent unrelated algebraic equations. A comparison of the results obtained by the developed and numerical methods shows their complete coincidence.

**Keywords**: mathematical modeling, thermal processes, thermal feedback, polynomial matrix equation, spectral decomposition, matrix similarity transformation

#### Литература

- 1. Madera A.G. Modeling thermal feedback effect on thermal processes in electronic systems // Computer Research and Modeling, 2018, vol. 10, no. 4, pp. 483–494. DOI: 10.20537/2076-7633-2018-10-4-483-494
- 2. Madera A.G. The effect of thermal feedback and statistical technological dispersion of microcircuits parameters on their thermal mode // IOP Conference Series: Materials Science and Engineering, Vol. 862, Issue 32, 27 May 2020; Number of paper 032106. 8 pages. DOI: 10.1088/1757-899X/862/3/032106
- 3. Мадера А.Г., Кандалов П.И. Анализ интервально-стохастических температурных полей технических систем // Программные продукты и системы. 2014. № 4. С. 41-45
- 4. Мадера А.Г., Кандалов П.И. Моделирование температурных полей технических систем в условиях интервальной неопределенности // Тепловые процессы в технике. 2014. Т. 6. № 5. С. 225 229
- 5. Мадера А.Г. Моделирование теплообмена в технических системах. М.: Научный Фонд «ПИЛ им. акад. В.А. Мельникова», 2005, 208 с.
- 6. Madera A.G. A method for explicitly solving matrix differential equations in second-order ordinary derivatives based on diagonalizing the matrices of the equation using spectral decomposition and Kronecker matrix algebra // Differential equation and control processes. 2021, N. 3, P.
  - 7. Ланкастер П. Теория матриц. М.: Наука, 1973. 280 с.
  - 8. Хорн Р., Джонсон Ч. Матричный анализ. М.: Мир, 1989. 656 с.

# Алгоритм компьютерного моделирования тепловой обратной связи в электронных системах

#### Е.К. Гребенникова<sup>1</sup>

<sup>1</sup>ФГУ ФНЦ НИИСИ РАН, Москва, Россия, e-mail: ekaterina.grebennikova@inbox.ru

Аннотация. Представлен алгоритм компьютерного моделирования стационарного стохастического теплового процесса в электронных системах с тепловой обратной связью с применением метода Монте-Карло. Алгоритм основан на стохастической математической модели процесса, представляющей собой нелинейное стохастическое полиномиальное матричное уравнение, и методе аналитического представления ее решения. Разработанный алгоритм компьютерного моделирования представлен в виде блок-схемы с пошаговым описанием необходимых вычислений на каждом шаге. В результате работы алгоритма рассчитываются статистические меры (математические ожидания, дисперсии, среднеквадратические отклонения) стохастических температур элементов электронной системы, на основании которых вычисляются интервалы возможных значений реальных температур, имеющих место на практике.

**Ключевые слова:** компьютерное моделирование, математическое моделирование, матричные вычисления, блок-схема, тепловая обратная связь, электронные системы, тепловое моделирование

#### 1. Введение

Представлен алгоритм компьютерного моделирования стационарного стохастического теплового процесса в электронных системах (ЭС) в условиях воздействия тепловой обратной связи. Математической моделью описываемого процесса является нелинейное матричное полиномиальное уравнение  $AG(\omega)A^TT(\omega) =$  $P(T,\omega)$ , правая часть  $P(T,\omega)$  которого представляет собой вектор с элементами, изменяющимися от температуры в каждом элементе по квадратичному закону, а именно,  $P_i(T_i, \omega) = P_{0i}(T_0, \omega) + a_i(\omega)T_i + b_i(\omega)T_i^2, \quad i =$ 1,2,...,n,где  $T_i(\omega)$ стохастическая температура i-го элемента (узла i графа);  $P_{0i}(T_0,\omega)$ стохастическая мощность потребления при температуре  $T_0$ ;  $a_i(\omega)$ ,  $b_i(\omega)$  – стохастические коэффициенты зависимости мощности  $P_i(T_i, \omega)$  от температуры [1 – 6]. Здесь A — матрица инциденций графа тепловой модели стохастическая квадратная диагональная  $m \times m$ -матрица проводимостей ветвей графа;  $\omega \in \Omega$  – элементарные события из пространства элементарных событий  $\Omega$ .

В представленной математической тепловой модели стохастическими параметрами являются элементы  $g_k(\omega)$ , k=1,2,...,m стохастической матрицы  $G(\omega)$ , мощность  $P_{0i}(T_0,\omega)$ , коэффициенты  $a_i(\omega)$ ,  $b_i(\omega)$ . Все стохастические величины в математической модели являются интервально стохастическими

статистически независимыми величинами, возможные значения которых заключены внутри интервалов своего изменения. Иначе говоря,  $g_k(\omega) \in [\underline{g}_k, \overline{g}_k], \ P_{0i}(T_0, \omega) \in [\underline{P}_{0i}(T_0), \overline{P}_{0i}(T_0)],$   $a_i(\omega) \in [\underline{a}_i, \overline{a}_i], \ b_i(\omega) \in [\underline{b}_i, \overline{b}_i].$ 

Таким образом, исходными данными для определения вектора стохастических температур в узлах графа тепловой модели являются:

- матрица инциденции графа А;
- интервальные значения стохастических величин  $g_k(\omega) \in [\underline{g}_k, \overline{g}_k], \quad P_{0i}(T_0, \omega) \in [\underline{P}_{0i}(T_0), \overline{P}_{0i}(T_0)], \quad a_i(\omega) \in [\underline{a}_i, \overline{a}_i], \quad b_i(\omega) \in [\underline{b}_i, \overline{b}_i]$  и векторы-столбцы их нижних и верхних границ.

Алгоритм компьютерного моделирования основан на методе решения нелинейного матричного полиномиального уравнения [7] для каждого  $\omega \in \Omega$ , а также методе Монте-Карло, вычисления множества реализации решений уравнения математической модели для различных элементарных событий  $\omega \in \Omega$  [8].

## 2. Описание алгоритма компьютерной реализации

Первым этапом любого компьютерного моделирования является выделение особенностей реализуемой задачи. Исследуемая математическая модель строится на матрицах, и их частных случаях — векторах. Данные математические объекты в компьютерном представлении удобно хранить в виде xls-файлов. Для реализации компьютерного моделирования математической модели тепловых процессов был выбран высокоуровневый язык программирования Java с использованием библиотеки матричных вычислений JAMA (Java Matrix Library). Для поддержки интеграции данных с xls-файлами используется библиотека Apache POI.

Входные и выходные данные системы хранятся в виде xls-файлов, состоящих из нескольких xls-страниц. Входной файл содержит информацию о числе электронных элементов, узлов и

ветвей графа, матрице инциденции графа A, интервальные значения элементов  $g_k(\omega) \in [\underline{g}_k, \overline{g}_k], \ P_{0i}(T_0, \omega) \in [\underline{P}_{0i}(T_0), \overline{P}_{0i}(T_0)], \ a_i(\omega) \in [\underline{a}_i, \overline{a}_i], \ b_i(\omega) \in [\underline{b}_i, \overline{b}_i].$  Исходные интервальные значения хранятся в виде наборов двух столбцов, элементы которых соответствуют нижним и верхним границам интервала. Выходной файл содержит данные о рассчитанных векторах температур T и статистические меры: математические ожидания, дисперсии, среднеквадратические отклонения.

Алгоритм компьютерного моделирования представлен в виде блок-схемы (Рис. 1).

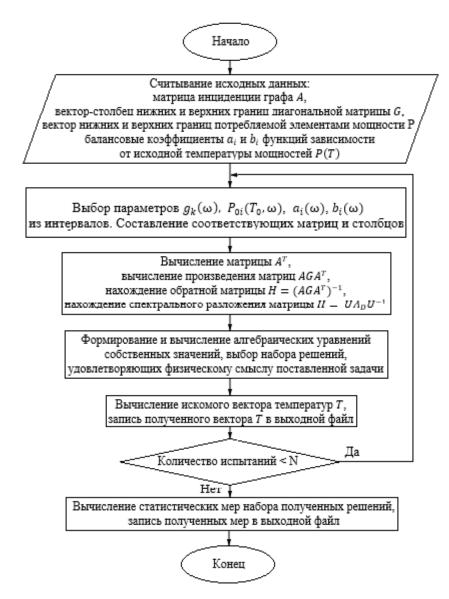


Рис. 1. Алгоритм компьютерного моделирования тепловых процессов в технических системах в условиях воздействия тепловой обратной связи

Первым этапом моделирования является наполнение проекта исходными данными. Создание на основе данных xls файла целочисленных переменных, содержащих информацию о числе электронных элементов, числе узлов и ветвей графа, и переменной двойной точности, содержащей информацию о температуре окружающей среды  $T_0$ . Создание экземпляра класса Matrix библиотеки JAMA для матрицы A. Создание двумерных массивов верхних и нижних границ  $g_k(\omega) \in [\underline{g}_k, \overline{g}_k]$  матрицы G, потребляемой элементами мощности  $P_{0i}(T_0, \omega) \in [\underline{P}_{0i}(T_0), \overline{P}_{0i}(T_0)]$ , и коэффициентов зависимости мощности от температуры  $a_i(\omega) \in [\underline{a}_i, \overline{a}_i], b_i(\omega) \in [\underline{b}_i, \overline{b}_i]$ 

Для расчета стохастических параметров исходного аналитического уравнения используется метод Монте-Карло. Таким образом, вторым этапом компьютерного моделирования является выборка значений элементов  $g_k(\omega)$  матрицы G и элементов  $P_{0i}(T_0, \omega)$ ,  $a_i(\omega)$ ,  $b_i(\omega)$  из соответствующих интервалов.

Далее поэтапно вычисляются матрицы  $A^T$ , произведение матриц AG,  $AGA^T$ , находится обратная матрица  $H = (AGA^T)^{-1}$ , и спектральное разложение  $H = U\Lambda_D U^{-1}$ .

Формируются и вычисляются алгебраические уравнения собственных значений. Производится выбор набора решений, удовлетворяющих физическому смыслу поставленной задачи по определенной в системе логике.

С учетом полученного спектрального разложения и выбранных собственных значений определяется вектор искомых температур T. Полученный вектор температур T записывается в виде строки в xls-файл выходных данных. Сохраняются промежуточные результаты для статистических мер: математических ожиданий, дисперсий, средних квадратических отклонений, нижних и верхний границ решений.

Этап выбора стохастических параметров элементов  $g_k(\omega)$ ,  $P_{0i}(T_0,\omega)$ ,  $a_i(\omega)$ ,  $b_i(\omega)$  и последующие вычисления для определения искомого вектора температур T в узлах графа и запись полученных данных в xls файл повторяются заданное число N раз.

На основе сформированных векторов температур вычисляются и записываются в выходной xls-файл основные статистические меры.

#### 3. Заключение

В статье представлен алгоритм компьютерной реализации метода расчета стационарного теплового процесса в электронных системах с тепловой обратной связью, описываемый нелинейным матричным уравнением с полиномиальной правой частью [2 articles in Lectures], разработанного на основе графа тепловой схемы [5] исходной математической модели.

Выходными данными разработанного алгоритма являются: набор искомых температур  $T_N$ , математические ожидания, дисперсия, средние квадратические отклонения, нижние и верхние границы решений, записанные в выходной xls-файл. Получаемый набор данных позволяет проводить полноценное исследование адекватности найденных решений.

В качестве усовершенствования представленного алгоритма рассматривается применение технологий распараллеливания. Предполагается доработка исходного алгоритма до автоматического составления матрицы инциденции графа A и вычисление матрицы G на основе количества заданных электронных элементов и их характеристик. Добавление редактируемой базы данных электронных элементов и их характеристик.

Также предполагается создание полноценного приложения на основе описанного алгоритма, включающего в себя графический интерфейс пользователя для создания проектов, ввода исходных данных и отображения полученных результатов.

Публикация выполнена в рамках государственного задания ФГУ ФНЦ НИИСИ РАН «Проведение фундаментальных научных исследований (47 ГП)» по теме № FNEF-2021-0001 «Математическое обеспечение и инструментальные средства для моделирования, проектирования и разработки элементов сложных технических систем, программных комплексов и телекоммуникационных сетей в различных проблемно-ориентированных областях. 0580-2021-0001», рег. № 121031300047-6.

## Algorithm of Computer Simulation of Thermal Feedback in Electronic Systems

#### Ekaterina K. Grebennikova

**Abstract.** An algorithm for computer simulation of a stationary stochastic thermal process in electronic systems with thermal feedback using the Monte Carlo method is presented. The algorithm is based on a stochastic mathematical model of the process which is a nonlinear stochastic polynomial matrix equation and a method of analytical representation of its solution. The developed algorithm of computer modeling is presented in the form of a flowchart with a step-by-step description of the necessary calculations at each step. As a result of the algorithm statistical measures (mathematical expectations, variances, standard deviations) of electronic system element stochastic temperatures are calculated, on the which basis the intervals of possible values of real temperatures that occur in practice are calculated.

**Keywords:** computer modeling, mathematical modeling, matrix calculations, block diagram, matrix, thermal feedback, electronic systems, thermal modeling

#### Литература

- 1. Madera A.G. Modeling thermal feedback effect on thermal processes in electronic systems // Computer Research and Modeling, 2018, vol. 10, no. 4, pp. 483–494. DOI: 10.20537/2076-7633-2018-10-4-483-494.
- 2. Madera A.G. The effect of thermal feedback and statistical technological dispersion of microcircuits parameters on their thermal mode // IOP Conference Series: Materials Science and Engineering, Vol. 862, Issue 32, 27 May 2020; Number of paper 032106. 8 pages. DOI: 10.1088/1757-899X/862/3/032106.
- 3. Мадера А.Г., Кандалов П.И. Анализ интервально-стохастических температурных полей технических систем // Программные продукты и системы. 2014. № 4. С. 41 45.
- 4. Мадера А.Г., Кандалов П.И. Моделирование температурных полей технических систем в условиях интервальной неопределенности // Тепловые процессы в технике. 2014. Т. 6. № 5. С. 225 229.
- 5. Мадера А.Г. Моделирование теплообмена в технических системах. М.: Научный Фонд «ПИЛ им. акад. В.А. Мельникова», 2005, 208 с.
- 6. Madera A.G. A method for explicitly solving matrix differential equations in second-order ordinary derivatives based on diagonalizing the matrices of the equation using spectral decomposition and Kronecker matrix algebra // Differential equation and control processes. 2021, N. 3, P.
- 7. Madera A.G. Mathematical modeling of various psysical processes in thechnical systems described by polynomial matrix equations // Lecture Notes in Networks and Systems, 2021.
  - 8. Ермаков С.М., Михайлов Г.А. Статистическое моделирование. М.: Наука, 1982.
- 9. Шилдт Герберт, Java 8. Полное руководство; 9-е изд.: Пер. с англ. -М.: ООО "И.Д. Вильяме", 2015. 1376 с.
- 10. Питер Уэйнрайт, Арасһе для профессионалов [Пер. И.Дранишников]. М.: Лори, 2001. XXII, 473 с.
  - 11. Ланкастер П. Теория матриц. М.: Наука, 1973. 280 с.
  - 12. Хорн Р., Джонсон Ч. Матричный анализ. М.: Мир, 1989. 656 с.

# Влияние физических параметров на ВАХ цилиндрических КНИ КМОП нанотранзисторов с полностью охватывающим затвором

#### **Н.В.** Масальский<sup>1</sup>

<sup>1</sup>ФГУ ФНЦ НИИСИ РАН, Москва, Россия, volkov@niisi.ras.ru

Аннотация. При масштабировании цилиндрических КНИ КМОП нанотранзисторов с полностью охватывающим затвором по мимо проявления коротко-канальных эффектов возникают и другие механизмы, которые ограничивают потенциал масштабирования. Эти дополнительные ограничения обусловлены и квантовомеханическими эффектами и наличием емкостных связей, влияние которых в предыдущих транзисторных архитектурах было ничтожно малым. При помощи экспериментально протестированной квазианалитической модели цилиндрического КНИ КМОП нанотранзистора с полностью охватывающим затвором численно исследованы воздействия этих механизмов на ВАХ транзисторов и предложены способы их компенсации с учетом технологических требований. Параметры прототипов транзисторов варьировались в диапазоне: длина и радиус рабочей области 11...16 нм и 1.5 ... 5 нм, соответственно, управляющие напряжения 0 ... 0.4 В.

**Ключевые слова:** нанотранзистор, кремний на изоляторе, цилиндрическая рабочая область, вольт-амперные характеристики, моделирование

#### 1. Введение

Маіпѕtrеат - "Больше Мура и больше, чем Мур" охватывает создание нанотранзисторных архитектур, среди которых доминируют те, канал которых полностью окружен затвором, в частности, с цилиндрической геометрией, которые находят применение при разработке высокоэффективных СБИС [1-5]. Важно, что возможность масштабирования данной архитектуры сохраняется для топологических норм 3 нм [6]. Это делает изучение тонкостей физики работы кремниевых цилиндрических транзисторов с полностью охватывающим затвором важными и с научной, и с практической точек зрения.

В [7] рассмотрена квазианалитическая модель для комплементарного кремниевого нанотранзистора с цилиндрической геометрией и полностью охватывающим затвором для тонких каналов. Предложенный подход моделирования поддерживает целостность транзистора с помощью набора самосогласованных выражений, которые способны с высокой точностью воспроизводить его поведение. Его отличают точность, простота настройки параметров, экономия вычислительных ресурсов и надежность. Все внутренние узловые уравнения модели решаются в самой модели, тем самым сокращая время вычислений. Такая модель имеет явное преимущество перед TCAD моделями [8] за счет лучшей сходимости и меньшего количества параметров. Следует отметить, что любая модель будет полезна только в том случае, если она может моделировать достоверные характеристики транзистора с учетов технологических ограничений [9]. Для достижения этой цели модель должна учитывать любые вторичные эффекты, которые могут повлиять на характеристики устройства в реальных режимах работы.

Целью настоящей работы исследование физических параметров, связанных со вторичными эффектами в ультра тонких КНИ КМОП наноразмерных транзисторах с полностью охватывающим затвором.

#### 2. Методика моделирования

Рисунок 1 иллюстрирует представление анализируемой нанотранзисторной структуры, где на толстой кремниевой пластине (Si) расположен изолирующий слой оксида кремния (SiO2), на верхней грани которого помещена транзисторная структура. Она включает исток (поз. 1), сток (поз. 2) и рабочую область (поз. 3) протяженностью Lg, которая также является и длиной затвора транзистора.

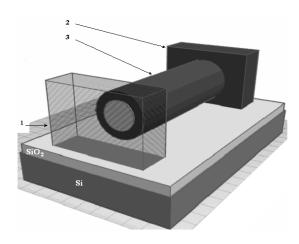


Рис. 1. Структурная схема МОП транзистора с полностью охватывающим затвором, где 1 – исток, 2 – сток, 3 –рабочая область

Компьютерное моделирование выполнено для транзисторных структур с различными геометрическими размерами рабочей области. В расчетах длина рабочей области Lg изменяется в диапазоне 11...16 нм, радиус R от 1.5 до 5 нм. Фиксированы размеры стока/истока 50х50х50 нм. Области стока/истока равномерно легированы с разными концентрациями Nds от 0.5x1016 до 0.5x1021 см-3. Кремниевая рабочая область полагается нелегированной с собственной концентрацией носителей 1х1013 см-3. Границы примыкания рабочей области к истоку/стоку являются резкими. Зазор(спейсер) между затвором и областями истока и стока имеет конечную величину. Толщина затвора фиксирована и составляет 32 нм. Толщина подзатворного диэлектрика (tox) варьируется в пределах 1.0...1.3 нм из-за чрезмерного роста туннельного тока затвора [10, 11]. В расчетах для зоны проводимости рассмотрены три пары вырожденных долин с анизотропной эффективной массой носителей, подзатворный диэлектрик обеспечивает высоту потенциального барьера 3.15 эВ при условии изотропной эффективной массы носителей. Температура решетки постоянна и равна 300 К. Омические контакты формируются на стоке и истоке с противоположной стороны активной области.

На рис. 2 представлена зависимость характеристической длины l от R, tox, из которой можно определить диапазон параметров R и tox для заданного значения Lg при которых выполняется условие полного подавления коротко-канальных эффектов (ККЭ): Lg>2l(R, tox).

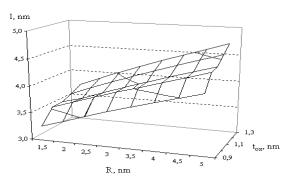


Рис. 2. Распределение характеристической длины в плоскости  $(R,\,t_{ox})$ 

#### 3. Расчеты и обсуждение

Для минимизации емкости затвор-исток (затвор-сток) и сток-исток области стока и истока выполнены в виде фигуры переменной толщины [12], например, с параболическим контуром (см. рис. 3). Тогда зазор между затвором и истоком (стоком) будет увеличиваться квадратично в ортогональном направлении от рабочей области. Отношение Lgds/lgds, где Lgds и lgdsмаксимальное и минимальное расстояние между затвором и стоком(истоком) является граничным условием для уровня емкостной связи [6, 13]. Принимая в расчетах lgds = 1.5 нм и с учетом того, что максимальный размер стока/истока 50 нм получаем, что наибольшее значение отношения Lgds/lgds примерно 15, что является показателем очень слабой емкостной нагрузки [6, 13]. Это говорит о хорошей компенсации паразитных емкостей влияющих на временные, в частности, на время переключения транзистора, характеристики.

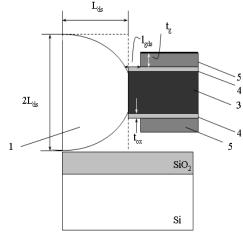


Рис. 3. Структурная схема узла сток(исток)затвор, где 1 - исток, 3 - рабочая область, 4 – подзатворный диэлектрик, 5 — поликремниевый затвор, lgds - размер зазора между затвором и истоком(стоком), tox - толщина подзатворного диэлектрика, tg - толщина затвора.

На рисунке 4 приведены экстрагированные из расчетов зависимости максимального тока транзистора Ids\_max при Uds=Ugs=0.4 В для разных концентраций легирования стока/истока и с различными Lg.

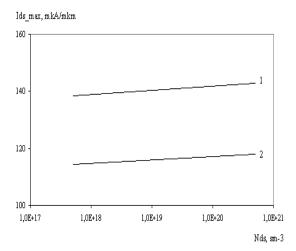


Рис. 4. BAX Ids(Uds) при Ug=0.4 B, 1 – прототип I - Lg=11 нм, R=1.5 нм, 2 – прототип II - Lg=16 нм, R=1.5 нм и во всех структурах tox=1 нм

С изменением уровня легирования соответствующим образом изменяется встроенный потенциал (Ubi) [14]. Это вызывает изменение поверхностного потенциала и, соответственно, изменяется ток транзистора Ids. В исследуемом диапазоне концентраций изменение поверхностного потенциала оставляет около 3%. Поэтому ток Ids практически пропорционально зависит от уровня легирования. При этом для разных Lg наблюдается существенное различие в тока из-за различия значений поверхностного потенциала. Так для Lg=11 и 16 нм отношение потенциалов составляет 0.81, отношение токов – 1.21.

Хорошо известно, что с ростом R увеличивается ток Ids [7, 15, 16]. Однако, в силу ограничения по ККЭ значение параметров Lg, R, tox жестко связаны между собой. Применение концепции переменной толщины (в нашем случае диаметра) рабочей области позволяет ослабить это условие. В настоящей работе мы рассматриваем кремниевую рабочую область в виде усеченного конуса. Меньший диаметр примыкает к истоку больший к стоку. При этом толщина tox остается постоянной. Следует отметить, что такая конструкция является технологически реализуемой. В нашем случае для Lg = 11 нм и tox = 1 нм, опираясь на рис. 2, мы выбрали Rmin=1.5 нм и Rmax=3.3 нм и получили более чем двукратное увеличение ток Ids. Однако, с увеличением Lg потенциал данного подхода снижается. Для Lg=16 нм можно добиться только 50 % увеличения тока, для Lg=22 нм - менее 20% . Все

из-за усиления ККЭ и роста туннельного тока затвора. По мере того как толщина становится меньше, ток утечки увеличивается из-за туннелирования, что приводит к снижению надежности транзистора.

На рисунке 5 приведены BAX Ids(Uds) при Ugs=0.4 В для разных толщин tox при различных Lg. Параметр tox является одним из главных, которые определяют характеристики транзистора, из-за емкостной связи данной пленки со всеми конструктивными компонентами транзистора. По мере увеличения толщины подзатворого окисла увеличивается емкость затвора, что приводит к усилению емкостной связи и соответствующему снижению тока транзистора [8, 17]. Кроме того, при изменении длины канала аналогично с предыдущим случаем уменьшается ток транзистора, примерно в той же пропорции.

Ids, mkA/mkm

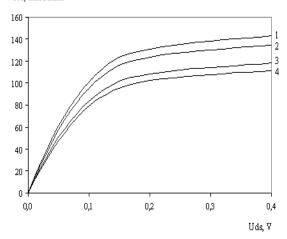


Рис. 5. BAX Ids(Uds) при Ug=0.4 B, 1 – прототип I - Lg=11 нм, R=1.5 нм, tox=1 нм , 2 - Lg=11 нм, R=1.5 нм, tox=1 нм , 3 – прототип II - Lg=16 нм, R=1.5 нм, tox=1 нм, 4 – прототип II - Lg=16 нм, R=1.5 нм, tox=1 нм

На рис 6. приведены экстрагированные из численно рассчитанных BAX Ids(Ug) во всем рассматриваемом диапазоне длин и радиусов рабочей области транзистора крутизны подпороговой характеристики SS.

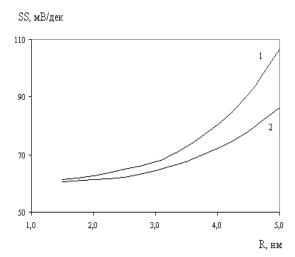


Рис. 6. Зависимость SS( R)при Uds=Ugs=0.4 B, где 1-Lg=11 нм, 2 - Lg=16 нм

Здесь параметр SS улучшается по мере уменьшения радиуса рабочей области. Однако, и пороговое напряжение синхронно увеличивается с уменьшением радиуса [9]. При этом оно становится более чувствительным к изменению радиуса при его малых значениях [16, 17]. Такой быстрый рост порогового напряжения обусловлен квантово-механическими эффектами (квантовыми эффектами ограничения) [11, 18]. Однако, рассеяние на шероховатости поверхности может также существенно влиять на ток транзистора. Эти два эффекта могут стать важными факторами в ограничении масштабирования транзисторов с полностью охватывающем затвором. Данная проблема характерна для всех транзисторных устройств с ультратонкими рабочими областями [3, 4].

Важно, что радиус лучше характеризует свойства масштабирования, чем характеристическая длина, что может быть связано с квантовыми эффектами. Поэтому при масштабировании следует опираться на радиус. Например, чтобы транзистор был электростатически устойчивым (то есть с SS < 80 мВ/дек) [7, 18], радиус рабочей области должен не больше 1/3 длины

затвора:  $R \le 1/3 L_g$ 

#### 4. Заключение

Численно проанализированы прототипы цилиндрических КНИ КМОП нанотранзисторов с полностью охватывающим затвором с топологическими параметрами, которые соответствуют 3 нм техпроцессу. Радиус активной области варьировался от 1.5 нм до 5 нм, толщина подзатворного окисла от 1 нм до 1.3 нм, уровень легирования истока/стока от 0.5х1016 до 0.5х1022 см-3, управляющие напряжения от 0 до 0,4 В.

При помощи экспериментально протестированной квазианалитической модели цилиндрического КНИ КМОП нанотранзистора с полностью охватывающим затвором исследованы вторичные механизмы, связанные с применением концепции переменной толщины и для активной зоны транзистора и для областей стока и истока. Предложенные с учетом технологических требований конструкции позволяют существенно увеличить ток транзистора и нивелировать паразитные емкости сток(исток)-затвор и сток-исток. Результаты моделирования показывают, что электро-физические свойства исследованных транзисторных структур отвечают требованиям к элементной базе высокоэффективных нантранзисторных СБИС с низким напряжением пита-

Публикация выполнена в рамках государственного задания ФГУ ФНЦ НИИСИ РАН «Проведение фундаментальных научных исследований (47 ГП)» по теме № FNEF-2021-0001 «Математическое обеспечение и инструментальные средства для моделирования, проектирования и разработки элементов сложных технических систем, программных комплексов и телекоммуникационных сетей в различных проблемно-ориентированных областях 0580-2021-0001», рег. № 121031300047-6.

### Effects of the Physical Parameters on the I-V Data of Cylindrical Surrounding Gate SOI CMOS Nanotransistor

#### N.V. Masalsky

**Abstract.** When scaling cylindrical surrounding gate SOI CMOS nanotransistors, in addition to the manifestation of short-channel effects, other mechanisms appear that limit the scaling potential. These additional

limitations are due to both quantum mechanical effects and the presence of capacitive bonds, the manifestation of which in previous transistor architectures was negligible. Using an experimentally tested quasi-analytical model of a cylindrical surrounding gate SOI CMOS nanotransistor, the influence of these mechanisms on the I-V data of transistors is numerically investigated and methods for their compensation are proposed, taking into account technological requirements. The parameters of the prototype transistors varied in the range: the length and radius of the working area of 11 ... 16 nm and 1.5 ... 5 nm, respectively, the control voltages of 0 ... 0.4 V.ext of Abstract.

**Keywords:** nanotransistors, silicon on insulator, cylindrical working area, I-V data, simulation.

#### Литература

- 1. D. Hsieh, J. Lin, P. Kuo, T. Chao. Comprehensive analysis on electrical characteristics of Pi-Gate poly-Si junctionless FETs. «IEEE Trans. Electron Dev.», (2017), 64(8), 2992–2998.
- 2. Y. Guerfi, G. Larrieu G. Vertical silicon nanowire field effect transistors with nano scale gate-all-around. «Nanoscale Res. Lett». (2016), 11(1), 210-217.
  - 3. Nanoelectronics: Devices, Circuits and Systems. Editor by Brajesh Kumar Kaushik. Elsevier, 2018.
- 4. S. Sahay, M. Kumar. Junctionless Field-Effect Transistors: Design, Modeling, and Simulation. Wiley-IEEE Press, 2019.
- 5. H.H. Radamson, H. Zhu, Z. Wu, X. He, H. Lin, J. Liu, J. Xiang, Z. Kong, W. Xiong, J. Li, H. Cui, J. Gao, H. Yang, Y. Du, B. Xu, B. Li, X. Zhao, J. Yu, Y. Dong, G. Wa. State of the art and future perspectives in advanced CMOS technology. «Nanomaterials», (2020), 10(8), 1555-1641.
- 6. J.S. Yoon, E.Y. Jeong, C.K. Baek, Y.R. Kim, J.H. Hong, . J.S. Lee, R.H. Baek, Y.H. Jeong. Junction design strategy for Si bulk FinFETs for system-on-chip applications down to the 7-nm node. «IEEE Trans. Electron Dev. Lett.», (2015), 36(10), 994-996.
- 7. Н.В. Масальский. Моделирование ВАХ ультра тонких КНИ КМОП нанотранзисторов с полностью охватывающим затвором. «Микроэлектроника» (2021), Т. 49(5)б 1-8.
- 8. N. Gupta, A. Vohra, R. Chaujar. Linearity performance of Gate Metal Engineered (GME) Omega gate-silicon nanowire MOSFET: a TCAD study. «IEEE International Conference on Electron Devices and Solid-State Circuits», (2016), 208–211.
- 9. V.R. Samoju, K. Mahapatra, P.K. Tiwari. Analytical modeling of subthreshold characteristics by considering quantum confinement effects in ultrathin dual-metal quadruple gate (DMQG) MOSFETs. «Superlattices Microstruct.», (2017), V.111, 704–713.
- 10. A. Orlikovsky, V. Vyurkov, V. Lukichev, I. Semenikhin, A. Khomyakov. All quantum simulation of ultrathin SOI MOSFET in nanoscaled semiconductor-on-insulator structures and devices. Springer. 2007.
- 11. S. Jin, Y.J. Park, H.S. Min. A three-dimensional simulation of quantum transport in silicon nanowire transistor in the presence of electron-phonon interactions. «J. Appl. Phys.», (2006), vol. 99, no. 12, 123719-1–123719-10.
- 12. E. Memisevic, J. Svensson, E. Lind, L.Wernersson. Vertical nanowire TFETs with channel diameter down to 10 nm and point SMIN of 35 mV/decade. «IEEE Electron Device Lett.», (2018), V. 39, No. 5, 1089-1091
- 13. S. Majzoub, M. Taouil, S. Hamdioui. System-level sub-20 nm planar and FinFET CMOS delay modelling for supply and threshold voltage scaling under process variation. «J. of Low Power Electron.», (2019), V. 15, No. 1, 1-10.
- 14. S.H. Oh, D. Monroe, J.M. Hergenrother. Analytic description of short-channel effects in fully-depleted double gate and cylindrical, surrounding-gate MOSFETs. «IEEE Electron Device Lett.», (2000), V. 21, 445–447
- 15. M.K. Pandian, N.B. Balamurugan. Analytical threshold voltage modeling of surrounding gate silicon nanowire transistors with different geometries. «J. Electr. Eng. Technol.», (2014), V. 9, 2079–2088
- 16. M. Kumar, S. Haldar, M. Gupta, R.S. Gupta. Physics based analytical model for surface potential and subthreshold current of cylindrical Schottky Barrier gate all around MOSFET with high-k gate stack. «Superlattices Microstruct.», (2016), V.90, 215–226
- 17. X. Chen, C.M. Tan. Modeling and analysis of gate all-around silicon nanowire FET. «Microelectr. Reliab.», (2014), V.54, 1103–1108.
- 18. W. Lu. Nanowire transistor performance limits and applications. «IEEE Trans. on Electron Dev.», (2008), V. 55(11), 2859-2876.

# О некоторых особенностях вычислений в проблеме периодичности непрерывных дробей элементов эллиптических полей

#### М. М. Петрунин<sup>1</sup>

**Аннотация.** В работе В. П. Платонов, М. М. Петрунин, Доклады РАН. Математика, информатика, процессы управления, 2020, Т. 495, с. 48-54 были получены важные результаты, касающиеся проблемы периодичности разложения  $\sqrt{f}$  в непрерывную дробь для кубических многочленов f над полями алгебраических чисел. В настоящей работе впервые приводится полное доказательство этих результатов для случая N=21, которое в вышеуказанной статье в силу ограничений на объем приводилось частично.

**Ключевые слова:** эллиптическое поле, S-единицы, непрерывные дроби, периодичность, точки конечного порядка.

#### 1. Введение

Рассмотрим свободный от квадратов многочлен  $f(x) \in \mathbb{K}[x]$  степени 2g+1 над полем алгебраических чисел К. Предположим, что f(x) не делится на x и, таким образом, нормирование  $V_x$ , соответствующее линейному многочлену  $\chi$ , имеет два продолжения:  $V_{r}^{+}$ ,  $V_{r}^{-}$  в поле  $\mathbb{K}(x)(\sqrt{f(x)})$ . Тогда существует вложение  $\sqrt{f(x)}$  и тем самым поля  $\mathbb{K}(x)(\sqrt{f(x)})$  в поле формальных рядов Лорана  $\mathbb{K}((x))$ , что позволяет рассмотреть разложение этого элемента или любого другого элемента поля  $\mathbb{K}(x)(\sqrt{f(x)})$  в непрерывную дробь (подробнее, см. [1]). Пусть  ${\cal C}$  — гладкая компактификация гиперэллиптической кривой  $y^2 = f(x)$ . Рассмотрим вложение точки  $P = (0, \sqrt{f(0)})$  в якобиан  $\mathcal C$  , переводящее P в класс  $P\!-\!\infty$  . В случае, когда класс  $P-\infty$  имеет конечный порядок в якобиане, существует класс элементов поля  $\mathbb{K}(x)(\sqrt{f(x)})$ , разложение которых в непрерывную дробь периодично. Эти разложения обладают интересными свойствами, о которых можно узнать из работ [1],[2],[3].

Отметим, что некоторые элементы при указанных предположениях на пару  $(\mathcal{C}, P)$ 

заведомо периодичны: например,  $\sqrt{f(x)}/x^g$  и  $\sqrt{f(x)}/x^{g+1}$ . В свою очередь, сам элемент  $\sqrt{f(x)}$  периодичен не всегда. В связи с этим в работе [3] была поставлена проблема описания всех многочленов  $f(x) \in \mathbb{K}[x]$  степени 2g+1 для различных классов полей алгебраических чисел  $\mathbb{K}$  с квазипериодическим разложением  $\sqrt{f(x)}$  в непрерывную дробь (квазипериодичность  $\sqrt{f}$  равносильна периодичности, см. [2]).

В работе [7] были получены важные результаты, позволяющие получить полное описание кубических многочленов fполями алгебраических чисел К степени 3 над которых разложение  $\mathbb{Q}$ , для непрерывную дробь В поле формальных степенных рядов  $\mathbb{K}((x))$  периодично. Также была доказана теорема о конечности кубических многочленов  $f \in \mathbb{K}[x]$ периодическим разложением для расширений () степени не превосходящей б, и дано полное описание таких многочленов fнад произвольным полем, соответствующих эллиптическим полям с точкой кручения порядка  $N \leq 30$ .

Одним из основных результатов работы [7] была следующая теорема, полученная с использованием параметризаций пар

<sup>&</sup>lt;sup>1</sup> Федеральное государственное учреждение "Федеральный научный центр Научно-исследовательский институт системных исследований Российской академии наук", Москва, Россия, petrushkin@yandex.ru;

эллиптическая кривая и точка кручения (см. [5, 6]).

Теорема 1. Число классов эквивалентности свободных квадратов кубических многочленов  $f \in \mathbb{K}[x]$ , отличных от вида  $cx^3+1$ , над полем алгебраических чисел  $\mathbb{K}$ степени d = 3 $\mathbb{O}$ , имеющих f(x)периодическое разложение непрерывную дробь над  $\mathbb{K}$ , — конечно, и определяется следующими представителями:  $12x^3 - 8x^2 + 4x + 1$ ,

$$12x^{3} - 5x^{2} + 2x + 1,$$

$$-120x^{3} + 25x^{2} + 2x + 1,$$

$$x^{3} + (-6z^{2} - 6)x^{2} +$$

$$+(249z^{2} + 105z + 360)x +$$

$$+\frac{2397}{2}z^{2} + \frac{2055}{4}z + \frac{3495}{2},$$

где z корень многочлена  $t^3 - t^2 + \frac{1}{2}t - \frac{1}{12}$ ;

$$(x+9z^2-6z+2)\cdot(x^2+$$
  
+ $(-48z^2+36z-14)x+216z^2-156z+65),$   
где z корень многочлена  $t^3-3t^2-5$ ;

$$x^{3} + \frac{1}{84}(64z^{2} - 40z - 123)x^{2} +$$

$$+ \frac{1}{49}(-32z^{2} + 32z + 39)x + \frac{1}{343}(216z - 369),$$

где 
$$z$$
 корень многочлена  $t^3 + t^2 - 2t - \frac{9}{2}$ .

Доказательство теоремы 1 сводилось к рассмотрению случаев существования специальной точки кручения порядка  $N \le 21$ ,  $N \neq 17.19$  на соответствующей эллиптической кривой. Существенную роль в рассмотрении каждого случая играли высокопроизводительные символьные вычисления, но в силу ограничений на объём работы [7] привести полные выражения результатов вычислений для всех случаев не представлялось возможным. Были приведено доказательство только случаев N = 11 и N = 20. В настоящей статье мы даём полное изложение случая N = 21.

Следуя схеме доказательства теоремы 1 из работы [7] рассмотрим кривую  $X_1(21)$ , заданную соотношением

$$g_{21}(u,t)=u^4+(3t^2+1)u^3+\\+(t^5+t^4+2t^2+2t)u^2+(2t^4+t^3+t)u+t^3.$$
 Тогда формулы  $r(t,u)=((-t^8+t^7-4t^6+4t^5-4t^4+5t^3--2t^2+3t-1)/(t^3-1))u^3+\\+((-3t^{10}+3t^9-13t^8+14t^7-17t^6+22t^5--13t^4+16t^3-8t^2+4t-2)/(t^3-1))u^2+\\+((-t^{13}-3t^{11}-2t^{10}+3t^9-8t^8+13t^7--10t^6+14t^5-9t^4+9t^3-3t^2+t--1)/(t^3-1))u+(-t^{12}+t^{11}-5t^{10}+5t^9-7t^8+8t^7-4t^6+6t^5-3t^4+2t^3-1)/(t^3-1))u^3+((3t^3+t-1)/(t^3-1))u^2+((t^6+t^5+2t^3-1)/(t^3-1))u^2+((t^6+t^5+2t^3-1)/(t^3-1))u+(t^5+t^3-1)/(t^3-1)$ 

в соответствии с рассуждением из работы [7] определяют соответствующую эллиптическую кривую:

$$f_{21} = x^3 - (-\frac{1}{4}(t-1)^{-1}(t^2+t+1)^{-2}(t^{25}-t^{24}+6t^{23}-9t^{22}-t^{21}+13t^{20}-69t^{19}+t^{116t^{18}}-148t^{17}+168t^{16}-33t^{15}-107t^{14}+t^{142t^{13}}-818t^{12}+1162t^{11}-1487t^{10}+t^{1534t^9}-1538t^8+1322t^7-1021t^6+t^{168t^5}-400t^4+228t^3-77t^2+23t-t^{164t^2}-69t^$$

#### Порядок кручения 21.

$$-160377t^{12} + 125873t^{11} - 90468t^{10} + 59261t^9 - \\ -35077t^8 + 18532t^7 - 8586t^6 + 3381t^5 - \\ -1099t^4 + 259t^3 - 40t^2 + t + 1)u - \\ -\frac{1}{2}t^3(t^2 + t + 1)^{-2}(t^{28} - \\ -9t^{27} + 49t^{26} - 202t^{25} + 668t^{24} - \\ -1863t^{23} + 4493t^{22} - 9519t^{21} + 17982t^{20} - \\ -30557t^{19} + 47105t^{18} - 66347t^{17} + 85791t^{16} - \\ -102337t^{15} + 112876t^{14} - 115288t^{13} + 109081t^{12} - \\ -95474t^{11} + 77207t^{10} - 57426t^9 + 39109t^8 - \\ -24154t^7 + 13391t^6 - 6559t^5 + 2756t^4 - \\ -958t^3 + 249t^2 - 43t - 1))x - \\ -(-\frac{1}{4}(t - 1)t(t^2 + t + 1)^{-2}(t^{38} - \\ -10t^{37} + 60t^{36} - 269t^{35} + 960t^{34} - \\ -2853t^{33} + 7193t^{32} - 15473t^{31} + 28223t^{30} - \\ -42145t^2 + 45809t^{28} - 14961t^{27} - 89865t^{26} + \\ +319571t^{25} - 725550t^{24} + 1341192t^{23} - 2162273t^{22} + \\ +3133625t^{21} - 4147987t^{20} + 5063965t^{19} - 5735701t^{18} + \\ +6050306t^{17} - 5957421t^{16} + 5481161t^{15} - 4712809t^{14} + \\ +3783158t^{13} - 2830332t^{12} + 1967861t^{11} - 1266519t^{10} + \\ +750506t^9 - 406470t^8 + 199344t^7 - 87390t^6 + 33651t^5 - \\ -11086t^4 + 3019t^3 - 637t^2 + 93t - 7)u - \\ -\frac{1}{4}(t - 1)t^3(t^2 + t + 1)^{-2} \cdot \\ \cdot (t^{35} - 11t^{34} + 72t^{33} - 353t^{32} + \\ + 1395t^{31} - 4662t^{30} + 13533t^{29} - 34732t^{28} + \\ +79895t^{27} - 166304t^{26} + 315709t^{25} - 550099t^{24} + \\ +884262t^{23} - 1317276t^{22} + 1825183t^{21} - 2359343t^{20} + \\ +2852285t^{19} - 3230265t^{18} + 3431242t^{17} - 3420138t^{16} + \\ +3198870t^{15} - 2805763t^{14} + 2304619t^{13} - 1769234t^{12} + \\ +1265568t^{11} - 840238t^{10} + 515073t^9 - 289477t^8 + \\ +147783t^7 - 67650t^6 + 27306t^5 - 9475t^4 + \\ +2718t^3 - 601t^2 + 93t - 7)).$$

Рассмотрим разложение квадратичной иррациональности  $\sqrt{f_{21}}/x^2$  в непрерывную дробь в  $\mathbb{K}(t,u)((x))$ . В этом случае

$$L_9 = (\frac{1}{t}(-t^{18} - 6t^{17} - 5t^{16} + 31t^{15} + 58t^{14} - 36t^{13} - 45t^{6}u - 375t^{5}u^2 + 26t^{4}u^3 - 132t^6 - 45t^{6}u - 375t^{12} + 26t^{4}u^3 - 132t^6 - 45t^{6}u - 375t^{12}u^2 + 26t^{4}u^3 - 132t^6 - 45t^{6}u - 375t^{12}u^2 + 26t^{4}u^3 - 132t^6 - 45t^{6}u - 375t^{12}u^2 + 26t^{4}u^3 - 132t^6 - 45t^{6}u - 375t^{5}u^2 + 26t^{4}u^3 - 132t^6 - 45t^{6}u - 375t^{6}u - 375t^{6}u - 45t^{6}u - 45t^$$

 $-112t^{15} + 56t^{14} + 73t^{13} - 32t^{12} + 63t^{11} + 178t^{10} -t^9 - 221t^8 - 131t^7 + 79t^6 + 111t^5 + 5t^4 -24t^3 - 12t^2 + t + 2)u - t^{21} - 7t^{20} - 10t^{19} +$  $+31t^{18} + 90t^{17} - 5t^{16} - 218t^{15} - 169t^{14} +$  $+190t^{13} + 345t^{12} + 21t^{11} - 271t^{10} - 154t^9 +$  $+70t^8 + 113t^7 + 3t^6 - 17t^5 - 16t^4 + 4t^3 + t^2)x^3$ причем  $L_n$  не пропорционален  $x^k$  при  $0 \le n < 9$ . Степень S -единицы гиперэллиптического поля, заданного многочленом  $f_{21}$ , совпадает с порядком точки кручения с x = 0 и равна 21.

 $\sqrt{f_{21}/x^2}$ Квазипериод разложения непрерывную дробь совпадает с периодом и равен 20. По критерию периодичности квадратного корня из [4], примененной в случае нечётной степени, S -единицы периодичен, если и только если коэффициент  $P_n$  при  $x^{-1}$  обращается в нуль. Запишем это условие:

$$p_1(t,u) = \frac{1}{2(t^3+t^2+t)}(t^{18}u+t^{17}+5t^{16}u+\\ +3t^{15}u^2-t^{16}-7t^{15}u-3t^{14}u^2+t^{13}u^3+7t^{15}-\\ -63t^{14}u+19t^{13}u^2-t^{12}u^3-16t^{14}-109t^{13}u-\\ -47t^{12}u^2+6t^{11}u^3-51t^{13}-99t^{12}u-128t^{11}u^2-\\ -15t^{10}u^3-124t^{12}+88t^{11}u-251t^{10}u^2-45t^9u^3-\\ -15t^{11}+139t^{10}u+5t^9u^2-77t^8u^3+209t^{10}+\\ +163t^9u+417t^8u^2+12t^7u^3+243t^9+\\ +188t^8u+423t^7u^2+152t^6u^3+111t^8+\\ +61t^7u+210t^6u^2+123t^5u^3-228t^7+\\ -289t^5u-187t^4u^2-131t^3u^3-86t^5-\\ -134t^4u-175t^3u^2-55t^2u^3+59t^4-\\ -82t^3u+68t^2u^2-15tu^3+6t^3+70t^2u+\\ +2tu^2+19u^3+17t^2+23tu+19u^2)=0.$$
 Условия приведённые выше, задают нульмерный идеал, базис Грёбнера которого

состоит из 5 уравнений, последнее из которых

зависит только от t:

```
129836167174024286623281087096035199812039976123
                                                         359656972781230710867814645695388365130304643
   369036649288058763453907869035941314872979893000\\
                                                         89293795623657162816420806115827726530206219775
    18548023882003469517611583870862171401719996589\\
                                                         2277827494281127835496159422737459645825262739
                                                          2998758185985704384553178383737442649128099652420_{ \tt c}{}_{26}
    12039766229768730121663259094558697094141294997685
    129836167174024286623281087096035199812039976123
                                                           18548023882003469517611583870862171401719996589
   8348139091244212372688302563514078461961617178330
                                                          93499221798779797501745415335450371635577817885
    129836167174024286623281087096035199812039976123
                                                          3329132491641648374955925310154748713129230157
   11042312136342380127139142406085858788723497533468
                                                          11364301416787558325196308019152591315014884342905_{22}
    129836167174024286623281087096035199812039976123
                                                            18548023882003469517611583870862171401719996589
   13126973411007187401386294138096929230130209115228
                                                          204634767159746410325676814487865490117045554958019
     9987397474924945124867775930464246139387690471
                                                            129836167174024286623281087096035199812039976123
    3130266089548388276461668265080048523209029320489
                                                          232698633435021969344998222279289421310483438197811
     2060891542444829946401287096762463489079999621
                                                           129836167174024286623281087096035199812039976123
   31437565050480396965396983316712664368612833798782
                                                           73339097121469585707320399558660479990195473702155
     18548023882003469517611583870862171401719996589
                                                           129836167174024286623281087096035199812039976123
                                                          440084510177271710936674353592218231808866421085518\\
   15681288121506884250109481542183923380835662959912
     6182674627334489839203861290287390467239998863
                                                            129836167174024286623281087096035199812039976123
   807262767757467243554191954882564389548618367806540
                                                           151650411628413956591837068109597709244231116682117
     129836167174024286623281087096035199812039976123\\
                                                             18548023882003469517611583870862171401719996589\\
   21728245737628069620772743636260850790789540308932
     129836167174024286623281087096035199812039976123\\
                                                              2277827494281127835496159422737459645825262739\\
  + 1053916471923696514440615930714098801940688257442657
                                                            123351219106387159100151593928297422706625732159516
      129836167174024286623281087096035199812039976123
                                                              18548023882003469517611583870862171401719996589
842183981909349623028337920697239714112224538843561
                                                       55666791992132733153817976472974907579712174255900\\
                                                        14426240797113809624809009677337244423559997347
 129836167174024286623281087096035199812039976123
4719190262605166659658296542906142956131883445535
                                                     17178189287160771486991478131889481509286994422323
 2649717697429067073944511981551738771674285227
                                                       9987397474924945124867775930464246139387690471
7859956487571625805800930149303055477516738728910
                                                    -u^4 + 3(t+1)u^2 + 4t^3u + 4u^3 + \frac{51}{19}t
 9987397474924945124867775930464246139387690471
51508280031885236063743906402170157203853506200
                                                    392837623749885699448194831904526415215885125
129836167174024286623281087096035199812039976123
                                                    359656972781230710867814645695388365130304643
41563351774637225321609004983123498512347035225\\
                                                    37301648542967577740270230694651970507900070850
18548023882003469517611583870862171401719996589
                                                    2277827494281127835496159422737459645825262739
6916497583248286440072516375249659851692808148215
                                                      2539631420213339789145829270165383571917061052590
129836167174024286623281087096035199812039976123
                                                       18548023882003469517611583870862171401719996589
34308221446442419988358078501679136433583830960220
                                                       456651751239461018496987341728791712632002944370
 129836167174024286623281087096035199812039976123
                                                        1109710830547216124985308436718249571043076719
77443066369237941663729853099599264128323704028718
                                                       14101124471434256621723551809394737002815961416192
 129836167174024286623281087096035199812039976123
                                                        18548023882003469517611583870862171401719996589
                                                      55492635273505828397419917091771057425708472558723
7252343045859346119800266899973993848477921338111
 9987397474924945124867775930464246139387690471
                                                       129836167174024286623281087096035199812039976123\\
527863260941865904764015654057346254461675802139\\
                                                     116200280440173520108367401706659194449657536762933
 6182674627334489839203861290287390467239998863\\
                                                      129836167174024286623281087096035199812039976123\\
40314769732006820933939423818767805671777479964949_{\star^{17}}
                                                      473712918219510123323850769522782970600298294065261
 18548023882003469517611583870862171401719996589
                                                        129836167174024286623281087096035199812039976123
30082586559152735325154768184538019066228080158734
                                                       776739518597987436716719241138252604690601120377713
  6182674627334489839203861290287390467239998863\\
                                                        129836167174024286623281087096035199812039976123
895686005812950715523295282595179673557588331634713
                                                        131898235745464029005195828288690586050764434410477
 129836167174024286623281087096035199812039976123
                                                          18548023882003469517611583870862171401719996589
882743771665181371592533680118098052549139186748454
                                                        56626095032803837201266858084113278693702870511
 129836167174024286623281087096035199812039976123
                                                         9373775696630155701630285690277611711215073
635282268100674829312507364955365555942442467222744
                                                       71281728331191603082601150070678549189527631313193
 129836167174024286623281087096035199812039976123
                                                         18548023882003469517611583870862171401719996589
351504572430393645535644969467503734066559144398149
                                                       66100297391571927168238642151382743242728617487316\\
 129836167174024286623281087096035199812039976123
                                                         43278722391341428874427029032011733270679992041\\
16463154101246631750195859413088366816052882570263
                                                      5149379574769223882921213250709928880243077756812
 18548023882003469517611583870862171401719996589\\
                                                       9987397474924945124867775930464246139387690471
       1437134896679368390732761136764465672293588855735
                                                                     19
         9987397474924945124867775930464246139387690471
71108839520037949300128954479192355705241100
                                                 137554961375679338567156721246260951885600
359656972781230710867814645695388365130304643
                                                 996279702995098922071508713837640900637963
```

```
51379567540175815838259235099341195018614949
                                                                        2103257150767431057706518395879464123569033
  2121412774519876193044670795632562353186632220\\
                                                                          351314303322913339617582472195446142744906285
   359656972781230710867814645695388365130304643
                                                                          51379567540175815838259235099341195018614949
                                                                          4204003092142348768670227633592540943718820650
  4905304360111362968914800404761634962351148525
   359656972781230710867814645695388365130304643
                                                                          119885657593743570289271548565129455043434881
  19609134428712848244325161109104080724773600071
                                                                          6296813203277460913368038200185653942016453071
                                                                            51379567540175815838259235099341195018614949
   359656972781230710867814645695388365130304643
  70072818860000012749683097572847064493826154452
                                                                           5467375811625552672808751202535688260709619377
   359656972781230710867814645695388365130304643
                                                                            27665920983171593143678049668876028086946511
                                                                          41606924624544209107948417724467888859983242508
   2539453170883714353397483172448972682153699973
   17126522513391938612753078366447065006204983
                                                                           359656972781230710867814645695388365130304643
  601843738857714038318290531787965840322636860
                                                                        122359305413706252161001419379346708477267027484
   51379567540175815838259235099341195018614949
                                                                          359656972781230710867814645695388365130304643
  3662311508442733509703260953415822305406984707
                                                                          289816993200546881649420480210964226843549093776
    5708840837797312870917692788815688335401661
                                                                            359656972781230710867814645695388365130304643
  418969551553425663226633791862718999820876657176
                                                                            71328895790921388772679064539768985206732989330
    359656972781230710867814645695388365130304643
                                                                              51379567540175815838259235099341195018614949\\
  505010691117184438290982037073792280512762983246
                                                                            9111154381637368006537000800806388230199422359\\
    359656972781230710867814645695388365130304643
                                                                              6309771452302293173119555187638392370707099
  33472548360882290889855646086924946807089260668
                                                                          49979599941456165909655846660626551677552016131
    27665920983171593143678049668876028086946511
                                                                            51379567540175815838259235099341195018614949
  317118205784020193440141633733429939659106780773
                                                                           62522740422641740113996629768517944432771991853
    359656972781230710867814645695388365130304643
                                                                             119885657593743570289271548565129455043434881
  259913545112496972741764442765652638373284674
                                                                       6111114991153149193223352464005698442911994268\\
   1048562602860730935474678267333493775890101\\
                                                                         27665920983171593143678049668876028086946511\\
  2703348044566129320468029712519222694685176897
                                                                      +(t^2-t)u^2-(t^3-t^2)u
    27665920983171593143678049668876028086946511
   31135684291411367690893594872990648116414225
                                                                          867462949918497355236180259084168564682875
  6833482482843383506488478268212378937475788217
                                                                         18929314356906879519358665562915177112121297
   17310528625476398864546362184668498822713175
                                                                        4068038655203691451487946984114478838330725
  976211783263340500926925466887482705353684031
                                                                        9221973661057197714559349889625342695648837
  11026876537060243626993719550315300861216113320
                                                                           4430377287256231692695140982153244643474609570
                                                                           976211783263340500926925466887482705353684031
   6833482482843383506488478268212378937475788217
                                                                       38290157220421790082700729684741965229079816275<sub>,r24</sub>_
  69622189720038606151266619999156412755504388950
  6833482482843383506488478268212378937475788217\\
                                                                        2277827494281127835496159422737459645825262739\\
                                                                        35119550765331492665961504337577110871741463586_{22}
  171375306251770433724158158576473707761425043004
   6833482482843383506488478268212378937475788217\\
                                                                          976211783263340500926925466887482705353684031
  275605264179328924179712440525214477263870768677
                                                                         214147485966882236042032225794309674876264585818 20
   6833482482843383506488478268212378937475788217
                                                                          6833482482843383506488478268212378937475788217
  1373829140298192757921695731761796722363012519
                                                                      118063598285705762997948223717976069788861338724
  108467975918148944547436162987498078372631559
                                                                       6833482482843383506488478268212378937475788217
                                                                       102783590658127019571919752025277588\underline{3629973106916}_{t^{16} \ \bot}
  70408682022127612070388631769659777305447369943
   976211783263340500926925466887482705353684031
                                                                         6833482482843383506488478268212378937475788217
  71140243523790534669943335442061513776899321029
                                                                       1899558892655231017043138511647425920455227777130
   325403927754446833642308488962494235117894677
                                                                         6833482482843383506488478268212378937475788217
  2363022365981606296545546853211242231166920994335
                                                                          370223101216599930870955436675249878151125547234
    6833482482843383506488478268212378937475788217
                                                                            976211783263340500926925466887482705353684031
  2551727040915171226384761632658532289608931883502
    6833482482843383506488478268212378937475788217
  42350143186980960623878036610000702597363054677
                                                                       2041886612340518690426506774087955023540814755193
                                                                          6833482482843383506488478268212378937475788217
   119885657593743570289271548565129455043434881
                                                                        1277721154962542872468912894430803969889059395752
  233823424212370449822096871692259590525972467402
    976211783263340500926925466887482705353684031
                                                                          6833482482843383506488478268212378937475788217
                                                                       59689864663639413708791811402698709182721539116<sub>,</sub>,
  27925143734608586280880630026035442588677676058
   253091943809014203944017713637495516202806971
                                                                        976211783263340500926925466887482705353684031
                                                                       22435264217789020248373437091709404513957593499
   525652498680260269729882943708644533651983709
                                                                       525652498680260269729882943708644533651983709\\
(\frac{1}{275}) \cdot t^3 \cdot (t-1)^3 \cdot (t^2+t+1)^2 \cdot (t^6-t^5+t^4-t^3+t^2-t+1) (275t^{16}+550t^{15}-1650t^{14}+1925t^{13}-8855t^{12}+6825t^{11}-1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^{12}+1650t^
-595t^{10} + 13360t^9 - 531t^8 - 18644t^7 + 32795t^6 - 21399t^5 + 17605t^4 - 96733t^3 + 34008t^2 + 20332t + 79781) = 0.
```

Проверим реализуемость периодичного  $\sqrt{f_{21}}$  для корня каждого неконстантного множителя этого выражения.

Корни t=1 или t=z, где z является корнем неприводимого над  $\mathbb Q$  многочлена  $t^2+t+1$ , не отвечают  $f_{21}(t,x)$  с периодическим разложением  $\sqrt{f_{21}(t,x)}$ , поскольку t=z и t=1 также являются корнями знаменателя одного из коэффициентов  $f_{21}$ .

Корни t=0 или t=z, где z — корень неприводимого над  $\mathbb Q$  многочлена  $t^6-t^5+t^4-t^3+t^2-t+1$ , не отвечают  $f_{21}(z,x)$  с периодическим разложением  $\sqrt{f_{21}(t,x)}$  , поскольку t=z или t=0 влечет  $f_{21}(t,0)=0$  , тем самым P=(0,0) — точка второго порядка.

Пусть теперь z является корнем неприводимого над  $\mathbb{Q}$  многочлена  $t^{16} + 2t^{15} - 6t^{14} + 7t^{13} - \frac{161}{5}t^{12} + \frac{273}{11}t^{11} - \frac{119}{55}t^{10} + \frac{2672}{55}t^9 - \frac{531}{275}t^8 - \frac{18644}{275}t^7 + \frac{6559}{55}t^6 - \frac{21399}{275}t^5 + \frac{3521}{55}t^4 - \frac{96733}{275}t^3 + \frac{34008}{275}t^2 + \frac{20332}{275}t + \frac{79781}{275}.$ 

 $u = -\frac{14112310954083467275269}{14112310954083467275269} z - \frac{522678183484572862047}{14112310954083467275269} z$  $684169\underline{58155776966175}_{7}13\underline{\phantom{0}}\underline{\phantom{0}}266488314790817373650}_{7}12\underline{\phantom{0}}$ 2016044422011923896467 672014807337307965489 30813797880918881125 \_11 162047327068892969380 \_10 672014807337307965489 74668311926367551721 108357755252845689650 9 145373324600145517880 8 288006346001703413781 522678183484572862047  $\frac{775991671135868405185}{1568034550453718586141}z^7 + \frac{2845585396863249185492}{2016044422011923896467}z^6 \frac{168228345703206648401}{7} _{7} _{1} _{1} \\ 183786946684190889311} _{7} _{4} _{+}$ 74668311926367551721 672014807337307965489  $\frac{3320823838233130080011}{2016044422011923896467}z^3 + \frac{5909302025955067888}{10666901703766793103}z^2 -$ 1965363201017833498682 57316492418988534704065  $\frac{}{4704103651361155758423}z - \frac{}{14112310954083467275269}$ 

которому отвечает

 $f_{21}(x,z) = \frac{3}{x^3} + \frac{1}{42336932862250401825807} (4446344652149592108400z^{15} + \frac{1}{x^2} +$  $+33094719199130685204900z^{14} - 74853546936804125940450z^{13} -167500139418059153875400z^{12} + 779007612837539546808270z^{11} -2371870492387302970174020z^{10} + 5291311456831607515547680z^{9} -7502817642769305221320800z^8 + 8870037267279904803170694z^7 -11375925986540129719059418z^{6} + 14734888188297061933073184z^{5} -14076285609796798635615480z^4 + 8492832351748536553097336z^3 -7751377238629731081603774z^2 + 8399312046009075559096716z 2495554409400640732172020)x^2 + \frac{1}{2016044422011923896467}$ •  $(9882139628077577595381625z^{15} + 6572157671428438542098250z^{14} -67739423871843368280901650z^{13} + 160002892765602610188468850z^{12} -534633710716695254557203300z^{11}+962964393989567349589263525z^{10} -1315723702614001640624438120z^9 + 2243062611842254767639956070z^8 -2999239996678088596485441435z^7 +3314535342023900759053743965z^6 -3232085082427773894311818059z^5 + 3516027168507995057941638408z^4 -3988156674494609221790249272z^3 + 1769922423142084666246669293z^2 -$ -1128127018996943005262709864z + 2200701557824434125080271561)x + $+\frac{1}{896019743116410620652}(498833309739538935673939625z^{15}+$  $+334620735512878996438686000z^{14} -3435535697138044103427237150z^{13} +$  $+8058712986015483453854767850z^{12} - 26792608802357852856017983950z^{11} +$  $+48039153423868574010736370925z^{10} -65051918713379859697247753020z^9 +$  $+110901605096805433685493959295z^{8} - 148567799391932286854189609445z^{7} +$  $+163881819593115084450066707950z^{6} -158669799745765815720637856619z^{5} +$  $+172449045226786216238115775413z^4 -197461607410080370931227658030z^3 +$  $+87004266326606104235658650778z^2 - 54163356918466195836957819714z +$ +108998585493874523184874393309).

Разложение  $\sqrt{f_{21}}$  над числовым полем степени 16 периодично с квазипериодом 19 , и периодом 38 .

ИСТОЧНИК ФИНАНСИРОВАНИЯ: Работа выполнена в рамках государственного задания по проведению фундаментальных научных исследований по проекту № 0580-2021-0011.

### On Some Features of Calculations on the Problem of Periodicity of Continued Fractions of Elements of Elliptic Fields

#### M. M. Petrunin

**Abstract.** In the paper V.P. Platonov, M.M. Petrunin, Doklady Mathematics, **102**:3 (2020), 487–492, the important results were obtained concerning the problem of the periodicity of the continued fraction expansion of cubic polynomials over algebraic number fields. In this paper, for the first time, a complete proof of these results is given for the case N=21, which was only partially given in the original paper due to volume limitations.

Keywords: elliptic field, S-units, continued fractions, periodicity, torsion points

#### Литература

- 1. Платонов В. П. Теоретико-числовые свойства гиперэллиптических полей и проблема кручения в якобианах гиперэллиптических кривых над полем рациональных чисел // УМН. 2014. Т. 69:1, № 415. С. 3–38.
- 2. Платонов В. П., Петрунин М. М. Группы S-единиц и проблема периодичности непрерывных дробей в гиперэллиптических полях // Тр. МИАН. 2018. Т. 302, С. 354–376.
- 3. Платонов В. П., Федоров Г. В. О проблеме периодичности непрерывных дробей в гиперэллиптических полях // Математический сборник. 2018. Т. 209, № 4. С. 54–94.
- 4. Platonov V. P., Petrunin M. M., Zhgoon V. S. On the problem of periodicity of continued fraction expansions of  $\sqrt{f}$  for cubic polynomials over number fields // Doklady Mathematics. 2020. Vol. 102, no 1. P. 288-292.
- 5. Kubert D.S. Universal bounds on the torsion of elliptic curves // Proceedings of the London Mathematical Society. 1976. Vol. 3, no. 2. P. 193–237.
- 6. Sutherland A. Constructing elliptic curves over finite fields with prescribed torsion // Mathematics of Computation. 2012. Vol. 81, no. 278. P. 1131–1147.
- 7. В. П. Платонов, М. М. Петрунин О конечности числа периодических разложений в непрерывную дробь  $\sqrt{f}$  для кубических многочленов над полями алгебраических чисел // Доклады Российской академии наук. Математика, информатика, процессы управления. 2020. Т. 495 С. 48-54.

### Множество частных и Сидоново свойство

#### Ю.Н. Штейников<sup>1</sup>

<sup>1</sup>ФГУ ФНЦ НИИСИ РАН, Москва, Россия, yuriisht@gmail.com

**Аннотация.** В данной статье мы строим пример , показывающий количественные оценки сверху на размер множества частных двух множеств натуральных чисел с ассимптотической плотностью меньшей единицы. Этот пример показывает в частности, что размер множества дробей не обладает свойством множеств Сидона.

Ключевые слова: дроби, делимость, множество.

#### 1. Введение

Рассмотрим следующую конструкцию. Пусть имеется достаточно большое положительное число X и два множества А,В с асимптотическими плотностями a,b Стандартным соответственно. образом обозначим через А/В множество рациональных дробей вида р/q, где числа р,q принадлежат соответственно множествам А, В. Аналогичным образом через АВ обозначим множество чисел вида рф , где снова р,ф принадлежат соответственно множествам А, В. В цикле работ [1-3] было показано, что при фиксированных а, b у нас имеется бесконечно много пар неравных чисел (u,w) из AB, что |u-w| ограничено сверху и зависит от значений а,b. Этот результат неоднократно уточнялся. Последние улучшения были достигнуты и выведены из следующего утверждения о множестве частных этих же двух множеств А,В целых чисел. Цитируем теорему из [7].

**Теорема 1.** Для любого положительного e>0 существует такая константа C(e)>0 такая, что для произвольных множеств A, B с плотностями a,b соответственно u ограниченных на отрезке [1,X] справедливо следующее соотношение  $|A/B|>C(e)|A||B|(ab)^e$ .

В дальнейшей работе было показано, что скрывающийся множитель  $C(e)(ab)^e$  в этой теореме можно заменить на вполне конкретную выписываемую убывающую к 0 функция при ав стремящемся к 0. Это утверждение является неулучшаемым в том смысле, что этот множитель должен обладать этим свойством. В данной статье мы построим пример, показывающий какую сверху оценку можно выписать в качестве этого множителя  $C(e)(ab)^e$ . Данный пример основан на конструкции предложенной в статье [3].

#### 2. Конструкция примера

Возьмем какое-нибудь натуральное число m>1. Интервал [T, T+T/m] содержит не менее 2m

простых чисел если T достаточно велико. Пусть P обозначает какое-то множество этих 2m простых чисел,  $d(I) = \prod p$ , где произведение берется по всем простым числам из множества простых I, |I|=m. A(P) пусть обозначает множество чисел, делящихся на какой-нибудь d(I), где I произвольное подмножество размера m из множества P. Оценим ассимптотическую плотность множества A(P).

Теперь будем брать параметры : m — единственный параметр,  $T=m^{8+e}$ , где е — сколь угодно малое число. Подставляя это мы видим, что при некоторой абсолютной константе c>0 выполнено  $|A|^2 / |A/A| > \exp$  (c log a/log log a), где а- плотность множества A. Отсюда мы заключаем, что в качетстве  $C(e)(ab)^e$  не может быть величина ограниченная снизу для всевозможных параметров a,b > 0.

Отсюда мы заключаем, что множество дробей положительной плотности свойством Сидона не обладает.

#### 3. Заключение

ИСТОЧНИК ФИНАНСИРОВАНИЯ: Работа выполнена в рамках государственного задания по проведению фундаментальных научных исследований по проекту № 0580-2021-0011.

### **Quotient Set and Sidon Property**

#### Yurii Shteinikov

**Abstract.** In this article, we construct an example showing quantitative upper bounds for the size of the set of quotients of two sets of natural numbers with an asymptotic density of less than one. This example shows, in particular, that the size of the set of fractions does not have the property of Sidon sets.

Keywords: fractions, divisivility, set.

#### Литература

- 1. C. Sandor. On the minimal gaps between products of members of sequence with positive density. "Ann. Univ. Sci. Budapest Eotvos Sect. Math.", 28(2005), 3-7.
- 2. J. Cilleruelo, T.H. Le. Gaps in product sequence. "Israel Journal of Mathematics", 179 (2010), p. 285-295.
- 3. J. Cilleruelo, D. S. Ramana, O. Ramare. The number of rational numbers determined by large sets of integers. "Bull. London Math. Soc.", 42 (3) (2010), 517-526.

## **Синхронизация системных часов в многопроцессорных системах**

А.Н. Годунов <sup>1</sup>, И.И. Хоменков <sup>2</sup>, В.Г. Щепков <sup>3</sup>

<sup>1</sup>ФГУ ФНЦ НИИСИ РАН, г. Москва, Россия, nkag@niisi.ras.ru; <sup>2</sup>ФГУ ФНЦ НИИСИ РАН, г. Москва, Россия, nkigor@niisi.ras.ru; <sup>3</sup>ФГУ ФНЦ НИИСИ РАН, г. Москва, Россия, nkvs@niisi.ras.ru

**Аннотация.** В статье описан способ синхронизации системных часов на отдельных процессорных модулях в многопроцессорных системах, работающих под управлением операционной системы реального времени.

**Ключевые слова:** синхронизация часов, многопроцессорные системы, часы реального времени, операционная система реального времени, OCPB, POSIX, ARINC-653

#### 1. Введение

Важным аспектом в работе многопроцессорных систем является правильное и эффективное взаимодействие процессорных модулей между собой. Системные часы, используемыми всеми программными процессами и потоками на процессорном модуле, являются важнейшим средством синхронизации работы программного обеспечения на различных процессорах в системе и межпроцессорного обмена данными. Только слаженная работа всех процессорных модулей может обеспечить эффективную работы всей системы в целом.

Системные часы в любой аппаратуре зависят от частоты кварцевого генератора. Все кварцевые генераторы имеют погрешность частоты, которая получается в процессе изготовления. Другим существенным фактором, влияющим на частоту кварцевого генератора, является нестабильность температуры самой аппаратуры и температуры окружающей среды. Погрешность в частоте кварцевого генератора приводит к погрешности в показаниях системных часов. Эта погрешность накапливается с течением времени, в результате системные часы на разных процессорах в системе показывают разное время.

При старте многопроцессорной системы применяются различные программные способы начальной синхронизации системных часов на разных модулях, например, с помощью запроса по сети текущего времени от одного сервера времени. В какой-то момент можно добиться состояния, когда показания всех часов в системе оказывается одинаковым, но в ходе работы они будут расходиться все больше и больше.

В системах, использующих ОСРВ предъяв-

ляются высокие требования к временным характеристикам работы модулей. Идеальных кварцевых генераторов не существует и поэтому требуется средства синхронизации системных часов в процессе работы. Они могут быть как программными, так и аппаратными. Аппаратные методы синхронизации позволяют добиться более точной работы часов, чем программные методы. На практике обычно используются некоторая комбинация этих методов и программно-аппаратная реализация системы синхронизации часов позволяет обычно добиться необходимой точности часов.

Возможным решением является использование системных часов одного процессора в качестве эталонных и корректировка часов на остальных модулях в соответствии с ними. При этом показания часов могут расходиться с астрономическим временем, но в рамках одной системы все часы будут показывать одинаковое время, и вся система в целом будет работать синхронно, что и является нашей целью.

Внутренняя синхронизация часов процессоров вовсе не исключает возможности периодической подстройки показаний часов под точное астрономическое время от внешних источников. Практическим источником точного времени могут являться навигационные системы GPS [3], ГЛОНАСС и т. д. Навигационные приёмники могут принимать сигналы со спутников и выдавать, например, сигнал PPS (Puls Per Second) — это высокоточные метки времени, которые синхронизированы с эталонными атомными часами, и выдаются с частотой 1 раз в секунду. Другим распространенным источником точного астрономического времени являются серверы точного времени в сети интернет. Текущее время можно запросить по некоторому сетевому протоколу. Сеть неизбежно вносит временные задержки на

передачу данных и по точности сетевые часы сильно уступают спутниковым.

# 2. Реализация часов реального времени в соответствии с POSIX

Стандарт программирования POSIX в настоящее время широко используется при разработке прикладного программного обеспечения (ПО) для ОСРВ. В соответствии со стандартом POSIX прикладные программы используют время в секундах и наносекундах. Это не означает, что системные часы работают с наносекундной точностью, а это сделано из соображений мобильности ПО и позволяет использовать один и тот же программный код в системах с различной разрешающей способностью аппаратных часов.

В стандарте POSIX предусмотрены следующие часы:

- CLOCK\_REALTIME часы реального времени:
  - CLOCK\_MONOTONIC монотонные часы;
- CLOCK\_PROCESS\_CPUTIME\_ID часы процессорного времени конкретного процесса;
- CLOCK\_THREAD\_CPUTIME\_ID часы процессорного времени конкретного потока управления.

Значение часов CLOCK\_REALTIME интерпретируется как время (в секундах и наносекундах), истекшее с 0 часов 1 января 1970 года, UTC.

Монотонные часы CLOCK\_MONOTONIC указывают время, прошедшее с момента пуска системы в наносекундах. Здесь уместно упомянуть о другом стандарте для ОСРВ широко используемом в авионике - ARINC 653. Например, в семействе ОСРВ Багет 3.0, разработанной НИИСИ, используются одновременно два стандарта POSIX и ARINC 653. И показания этих часов в подсистеме ARINC 653 совпадают с показаниями часов CLOCK\_MONOTONIC в подсистеме POSIX.

ОСРВ может обслуживать одни или несколько аппаратных часов (таймеров). Для каждых из используемых аппаратных часов операционная система создаёт программные часы. Для указания программных часов используется соответствующий идентификатор (номер).

Часы имеют очереди заявок на выполнения каких-либо действий в какое-либо время. В очередях находятся программные объекты процесса (таймеры, потоки управления и др.), ожидающие наступления заданного момента времени. Для объектов в очереди абсолютного времени используется абсолютное время для указания ожидаемого момента времени. В очереди относи-

тельного времени (для задания временных интервалов) используется время с момента пуска часов. Функция стандарта POSIX для установки показаний часов *clock\_settime()* изменяет только абсолютное время с момента старта часов и влияет только на объекты, стоящие в очереди абсолютного времени и не влияет на очередь относительного времени.

Процесс разработки ПО для ОСРВ во встраиваемых системах схематично выглядит так. Дистрибутив ОСРВ устанавливается на жёсткий диск инструментальной ЭВМ (ИЭВМ) на которой работает программист. Он создаёт исходный код прикладного ПО. Затем начинается процесс сборки исполняемого образа ОСРВ. Сначала запускается конфигуратора ОСРВ. Конфигуратор позволяет включить тексты программ в создаваемый исполняемый образ ОСРВ, а также задать конфигурационные параметры ОСРВ (размер памяти, сетевые адреса, и т.д.). По завершению конфигурирования производится компиляция всех исходных текстов проекта и создание одного исполняемого файла, который можно загрузить в целевую ЭВМ (ЦЭВМ) и запустить на исполнение.

Часы в ОСРВ являются объектом конфигурирования и важно правильно задать их параметры, так как от них зависят временные характеристики всей системы. При конфигурировании ОСРВ указывается номер вектора прерываний аппаратных часов, используемых для программных часов CLOCK\_REALTIME. Также можно указать номера векторов дополнительных аппаратных часов и соответствующие им идентификаторы (номера) программных часов.

Многие современные процессоры имеют многоядерную архитектуру и каждое ядро может иметь свои собственные часы. Для многоядерных процессоров существует два подхода к реализации системных часов. Первый подход реализует системных часы только на одном ядре, другие ядра получают извещения от этих часов, что наступил такой момент времени и нужно сделать такое действие. Второй подход предусматривает собственные системные часы на каждом ядре и каждое ядро работает по своему собственному расписанию, но при этом часы должны работать синхронно для всего процессора в целом.

Рассмотрим наиболее типичную ситуацию использования одних аппаратных часов для процессора. В этом случае потребуется установка времени прерывания от часов в соответствии с заявками от разных процессов (выполняемых, возможно, на разных ядрах). При наступлении указанного времени, возможно, потребуется проинформировать (посылкой события и/или

прерыванием) сразу несколько процессов, возможно, выполняющихся на разных ядрах.

Одна из возможных реализаций – создание специального высокоприоритетного управления часами. Заявки на оповещение о наступлении какого-то момента времени посылаются в виде уведомления о наступлении события. Это уведомление содержит абсолютное или относительное время и идентификатор процесса, пославшего заявку. Поток управления часами поддерживает упорядоченную очередь (очереди) заявок и в соответствии с имеющимися заявками программирует часы (устанавливает время прерывания от часов). При получении прерывания от часов поток становится работоспособным и извещает о наступлении данного момента заинтересованные процессы.

## 3. Описание алгоритма синхронизации системных часов ОСРВ

Для синхронизации системных часов в ОСРВ используется механизм прерываний процессора. Эталонные часы выдают периодический сигнал, который вызывает одновременное прерывание на всех процессорах, через жёстко фиксированный интервал времени (например, раз в секунду).

Обработчик прерываний сигнала синхронизации в ОСРВ проверяет показания системных часов модуля и определяет величину расхождения с внешними эталонными часами, а затем вызывает функцию коррекции (синхронизации) времени, если это необходимо.

Сама процедура коррекции времени содержит некоторый алгоритм изменения показаний часов, который учитывает ограничения, наложенные на возможность изменения показаний системных часов, предыдущую историю синхронизации, и прогноз на будущее. Поэтому коррекция времени происходит не скачком (моментально), а в течении некоторого интервала времени (плавно), чтобы не вызвать нарушений в работе модуля. Если ошибка времени не превышает заданный порог, то коррекция системных часов вообще не производиться. Рассмотрим этот механизм подробнее.

Для синхронизации эталонные часы через фиксированный интервал времени D (например, раз в секунду) тем или иным образом подают сигнал, который вызывает прерывание. При этом значение эталонных часов не сообщается. Описываемые ниже средства позволяют плавно изменять показания системных часов с тем, чтобы они стали кратны D в моменты прерывания от эталонных часов.

Если предположить, что погрешность часов не превышает D, то по получению сигнала от эталонных часов можно вычислить значение погрешности. Если дополнительно предположить, что часы спешат, то (новое) значение погрешности Ef равно остатку от деления значения часов на D (оба числа - целые значения в наносекундах). Если же предположить, что часы отстают, то значение погрешности равно

Es = Ef - D.

Обозначим через E значение погрешности равное Ef, если часы спешат, и равное Es, если часы отстают.

Для измерения времени аппаратные часы (наплатные или процессорные) используют счетчик, к которому с большой частотой через равные интервалы времени вычитается (или прибавляется) 1. Когда счетчик становится равным 0 (некоторой величине N), часы запрашивают прерывание процессора. Если t - время в наносекундах, обозначим через C(t) количество единиц счётчика, соответствующее времени t.

Для регулировки часов можно увеличивать или уменьшать число единиц счётчика за интервал времени D. Если часы спешат, то следует увеличить число единиц счётчика (часы будут идти медленнее). Если часы отстают, то следует уменьшить число единиц счётчика (часы будут идти быстрее).

При незначительных значениях погрешности никаких действий не производится. Это позволяет сократить накладные расходы при небольших отклонениях в показаниях часов. С этой целью пользователь определяет пороговое значение Etsh (в наносекундах). Если |E| <= Etsh, то никаких действий по регулировке часов не производится.

Теперь рассмотрим случай |E| > Etsh. В этом случае количество единиц счётчика за интервал времени D будет изменено до получения нового сигнала от эталонных часов. После получения нового сигнала количество единиц счётчика за интервал времени D будет вновь пересчитано (номинальная величина интервала D в наносекундах при этом остается неизменной).

Так как большое изменение фактической (не номинальной) длины интервала времени D нежелательно (особенно нежелательно большое сокращение его величины), то величина такого изменения ограничена. Пользователь определяет две константы Efmax и Esmax. Первая из них определяет максимальную величину (в наносекундах), на которую показания часов могут уменьшиться за секунду. Вторая - максимальную величину, на которую показания часов могут увеличиться за секунду.

Вначале определяется, спешат часы или отстают. Считаем, что часы спешат, если

|Ef|/Efmax < |Es|/Esmax,

В противном случае считаем, что часы отстают.

Функция опроса часов должна учитывать изменение числа единиц счётчика в 1 секунде. Для опроса хранится время последнего прерывания от часов (в наносекундах и единицах счётчика), а также текущее количество единиц счётчика в одной секунде. При опросе часов берётся время последнего прерывания в наносекундах, опрашивается значение счётчика, вычисляется время сначала в единицах счётчика, а затем в наносекундах и, наконец, вычисляется текущее время в наносекундах.

Отметим, что всегда хранится текущее и расчётное значение тика. После получения сигнала записывается новое расчётное значения тика. После первого прерывания от часов после получения сигнала от эталонных часов расчётное значение становится текущим.

Отметим, что всегда хранится текущее и новое (расчётное) значение числа единиц в 1 секунде. После получения сигнала от эталонных часов вычисляется новое (расчётное) значение. При очередном прерывании от системных часов текущее значение замещается расчётным.

# 4. Программный интерфейс для синхронизации системных часов

Текущая реализация программного обеспечение для синхронизации системных часов ОСРВ [1] состоит из трёх функций, одного обработчика прерываний и одной вспомогательной утилиты:

Текущая реализация программного обеспечение для синхронизации системных часов ОСРВ состоит из трёх функций, одного обработчика прерываний и одной вспомогательной утилиты:

- функция ОСРВ: *clockAdjust()*. Эта функция постепенно (не за один вызов) подстраивает показания системных часов с тем, чтобы не нарушить работу системы в целом;
- функции пакета поддержки модуля (ППМ): boardClockGetTime() и boardClockAdjust(). Первая функция предназначена для опроса точного времени (по показаниям системных часов), вторая для настройки системных часов (ускорения или замедления);
- обработчик прерываний от эталонного таймера (написанная прикладным программистом). В простейшем случае обработчик лишь вызывает функцию OCPB *clockAdjust()* с нужными параметрами;
- команда OCPB: css (clock\_sync\_show). Это вспомогательная утилита, которая отображает

историю синхронизации системных часов на экране для наглядной проверки работы системы синхронизации в ходе отладки ПО. В окончательной версии ПО использование этой утилиты не обязательно.

Функция обработки прерываний от эталонных часов вызывает функцию корректировки часов операционной системы *clockAdjust()* со следующими аргументами:

D - период (время между двумя последовательными сигналами точного времени в наносекундах);

*Efmax* - максимальная величина, на которую показания часов можно уменьшить за секунду (в наносекундах);

*Esmax* - максимальная величина, на которую показания часов можно увеличить за секунду (в наносекундах);

Etsh - пороговое значение, при котором коррекция не производится (в наносекундах).

В результате работы функции *clockAdjust()* показания системных часов постепенно изменяются с тем, чтобы их показания стали кратны D в момент прерывания от эталонных часов.

Для работы функции clockAdjust() требуется поддержка со стороны ППМ, а именно, функция clockAdjust() вызывает следующие функции ППМ:

boardClockGetTime() — функция опроса точного времени (по показаниям системных часов); boardClockAdjust() — функция корректировки часов.

Функция boardClockGetTime() возвращает значение часов с максимальной точностью, которую может обеспечить используемая аппаратура. Возвращаемое время вычисляется по формуле

t = t1 + t2, где

t1 - время, прошедшее до предыдущего прерывания от часов (в нсек);

t2 - время, прошедшее с момента предыдущего прерывания от часов до настоящего момента (в нсек).

Величина t2 вычисляется по формуле

t2 = ((currcount \* 1000000000)/ count1sec), где currcount — число единиц счётчика, прошедшее с момента предыдущего прерывания от часов;

count1sec – число единиц счётчика в 1 секунде.

Аргументы функции регулировки (коррекции) системных часов boardClockAdjust(), входящей в состав ППМ:

D - время, за которое погрешность должна быть устранена (обычно длина периода, с каким эталонный таймер посылает прерывания), в неск:

diff - погрешность часов (положительная,

если часы спешат, и отрицательная, если часы отстают), в нсек.

Функция увеличивает или уменьшает скорость хода часов с тем, чтобы за указанный период устранить имеющуюся погрешность работы часов (отставание или опережение). С этой целью функция вычисляет новое значение числа единиц счётчика в 1 секунде по формуле

count1sec + (count1sec\*diff)/D, где

*count1sec* - номинальное (обычно используемое) число единиц счётчика в 1 секунде.

Новое (расчётное) значение числа единиц в 1 секунде будет использоваться после очередного прерывания от часов (для программирования часов и вычисления точного времени). До этого момента используется старое (текущее) значение (для вычисления точного времени).

Таким образом, следует хранить текущее и новое (расчётное) значение числа единиц в 1 секунде. При очередном прерывании от часов текущее значение замещается расчётным.

Функция может использоваться только для основных (системных) часов.

#### 5. Команда css (clock\_sync\_show)

Для того чтобы увидеть на экране как происходит реальная синхронизация системных часов нужно воспользоваться командой *css* или *clock\_sync\_show* операционной системы, набрав ее в командном интерпретаторе.

На экране появиться таблица из нескольких колонок с основными параметрами синхронизации системных часов (формат выдачи можно посмотреть в приложениях к статье):

- 1. TIME[sec] текущее время системных часов [sec];
- 2. ERROR[ns] текущая ошибка системных часов, относительно эталонных часов [ns];
- 3. CORRECTION[ns] текущая величина коррекции системных часов [ns];

С помощью опций утилиты можно вывевести на экран дополнительные статистические данные:

- 4. TICK\_1SEC число тиков процессора за 1 секунду;
- 5. TICK\_1SEC\_SHFT изменение числа тиков процессора по сравнению с предыдущим вызовом функции;
- 6. LIN\_ERR[ns] текущая линейная (систематическая) погрешность [ns];
- 7. LIN\_ERR\_AVR[ns] текущая средняя линейная (систематическая) погрешность (скользящее среднее) [ns];
- 8. TIME\_INTR[sec] период эталонных часов (по системным часам) [sec];
- 9. TIME\_SHIFT[ns] колебания периода эталонных часов (по системным часам) [ns].

По умолчанию только выводятся на экран только 3 первых колонки. Количество колонок можно изменять с помощью опции -v LEVEL.

Третья колонка "CORRECTION [ns]" показывает на сколько наносекунд была произведена коррекция системных часов при вызове функции boardClockAdjust(), если эта колонка содержит число +0, то коррекция системных часов не производилась.

Буфер для записи истории коррекции времени может вмещать в себя 120 записей (120 секунд), после этого запись в буфер прекращается. Чтобы не терять информацию об истории нужно запустить команду clock\_sync\_show в периодическом режиме css -i10 (смотри описание опций команды ниже).

Команда css (clock\_sync\_show) имеет опции, меняющие формат выдачи на экран. Опции, набранные маленькими буквами, задают какойто параметр, а опции, набранные большими буквами, его отменяют.

-> css -h

Usage: css [-i INTERVAL|I] [-a|-A] [-t|-T] [-f|-F] [-b|-B] [-n NUM|-N] [-v LEVEL|-V] [-1|-L] [-h|-?]

Option:

- -i INTERVAL print periodical statistics with interval INTRVL in seconds
- -I disable to print periodical statistics
- -a print all possible clock correction information
- -A disable to print clock correction information (cancel showing thread)
- -t print all records from clock
  correction history table (default)
  -T disable to print records from
  clock correction history table
  -f print last clock correction
- function call with parameters
  -F disable to print last clock
  correction function call with parameters
- -b print table header for each
  output
- -B disable to print table header for each output (continues table) -n NUM - print only NUM times
- -N print always
- -v LEVEL verbose print (LEVEL count columns in table: max LEVEL
  = 9)
- -V standart print (default LEVEL

```
= 3 )
-1 - enable linear error correc-
tion (default)
-L - disable linear error correc-
tion
-h - print help
Example:
    css -i10
```

Следующий раздел выдачи команды css содержит параметры последнего вызова функции clockAdjust(system\_time\_t D,int Efmax,int Esmax, int Etsh) (смотри описание функции выше) и статистику работы этой функции: сколько было вызовов функции, сколько раз была произведена коррекция часов:

```
-> css -f
```

#### CLOCK CORRECTION STATICTICS:

Clock correction function calls : 162 times.

Clock correction was realy made : 159 times.

```
Clock correction function last call: 163,999'998'250[sec]
```

O[ns].

# 6. Приложение 1. Пример коррекции системных часов, когда системные часы спешат (время «сжиматься»)

Etsh =

->css

clockAdjust(D, Efmax, Esmax, Etsh) = +20'000'000[ns]

```
D = 1'000'000'000[ns]

Efmax = 20'000'000[ns]

Esmax = 80'000'000[ns]

Etsh = 1'000[ns]
```

#### TIME[sec] ERROR[ns] CORRECTION[ns]

```
42,072'059'722 +72'059'722 +20'000'000

43,052'469'397 +52'469'397 +20'000'000

44,032'861'544 +32'861'544 +20'000'000

45,013'253'756 +13'253'756 +13'253'756
```

46,000'168'355	+168'355	+168'355
46,999'989'150	-10'850	-10'850
48,000'000'105	+105	+0
49,000'000'012	+12	+0
49,999'999'975	-25	+0
50,999'999'975	-25	+0

# 7. Приложение 2. Пример коррекции системных часов, когда системные часы отстают (время «растягивается»)

->css

clockAdjust(D, Efmax, Esmax, Etsh) = -80'000'000[ns]

```
D = 1'000'000'000[ns]

Efmax = 20'000'000[ns]

Esmax = 80'000'000[ns]

Etsh = 1'000[ns]
```

#### TIME[sec] ERROR[ns] CORRECTION[ns]

```
319,577'111'195 -422'888'805
                               -80'000'000
320,663'989'436 -336'010'564
                               -80'000'000
321,750'946'059 -249'053'941
                               -80'000'000
322,837'902'478 -162'097'522
                               -80'000'000
323,924'859'083
                 -75'140'917
                               -75'140'917
325,006'105'526
                  +6'105'526
                               +6'105'526
                   +109'252
                                +109'252
326,000'109'252
                    -5'264
326,999'994'736
                               -5'264
327,999'999'987
                      -13
                                 +0
329,000'000'027
                      +27
                                 +0
                      +35
                                 +0
330.000'000'035
330,999'999'997
                       -3
                                +0.
```

#### 8. Заключение

Описанный программно-аппаратный метод синхронизация системных часов ОСРВ позволяет получить одинаковое время на всех процессорных модулях с заданной точностью, используя как внешние, так и внутренние источники точного времени. Это позволяет организовать слаженную и эффективную работу всей многопроцессорной системы за счёт точного исполнения циклограммы работы каждого процессорного модуля и минимизации времени ожидания при межмодульном взаимодействии. Временные затраты на синхронизацию часов при этом минимальны, фактически время на обработку одного прерывания раз в секунду. Алгоритм обеспечивающий плавное изменение показаний системных часов позволяет избежать сбоев в работе модулей, что в свою очередь повышает отказоустойчивость всей многопроцессорной системы.

### System Clock Synchronization for Multiprocessor Systems

#### Alexander Godunov<sup>1</sup>, Igor Khomenkov<sup>2</sup>, Vladimir Shchepkov<sup>3</sup>

- <sup>1</sup> Scientific Research Institute of System Analisys, Moscow, Russia, nkag@niisi.ras.ru;
- <sup>2</sup> Scientific Research Institute of System Analisys, Moscow, Russia, nkigor@niisi.ras.ru;
  - <sup>3</sup> Scientific Research Institute of System Analisys, Moscow, Russia, nkvs@niisi.ras.ru

**Abstract.** This article describes the method of system clock synchronization for multiprocessor systems working under control of real-time operating system.

**Keywords:** system clock synchronization, multiprocessor systems, real-time, RTOS, POSIX, ARINC-653

#### Литература

- 1 А.Н. Годунов, В.А. Солдатов, Операционные системы семейства Багет (сходства, отличия и перспективы). «Программирование», Москва, 2014, №5, с. 68-76.
- 2. О.Н Пушкарев, Беспроводная система синхронизации времени на основе сигналов навигационных спутников ГЛОНАСС и GPS. Беспроводные технологии, № 2, 2010, с. 40.
- 3. B. Hellstrom, GPS-free synchronization of Digital Terrestrial TV and Mobile TV distribution networks. Net Insight AB, Sweden, 2007, p. 9
- 4. Kang Wang, Yong Hui Hu, Zai Min He, Hong Jiao Ma, Design and Implementation of IEEE 1588 High Precision Clock Synchronization Based on GPS Time Service. Advanced Materials Research, 2012, pp.532-533

# Синтез изображений трехмерных сцен с моделированием капель дождя на объективе виртуальной камеры

#### А.В. Мальцев<sup>1</sup>

<sup>1</sup>ФГУ ФНЦ НИИСИ РАН, Москва, Россия, avmaltcev@mail.ru

**Аннотация.** В работе предлагаются решения для визуализации трехмерной виртуальной среды с распределенным моделированием эффекта попадания дождевых капель на линзу объектива виртуального устройства наблюдения. Разработанные методы и подходы основаны на использовании параллельных вычислений с помощью современных многоядерных графических процессоров и обеспечивают рендеринг виртуальных сцен в масштабе реального времени.

**Ключевые слова:** трехмерная сцена, виртуальная камера, рендеринг, капли дождя, моделирование, шейдеры, графический процессор, постобработка

#### 1. Введение

На сегодняшний день во многих областях человеческой деятельности используются технологии и методы трехмерного виртуального моделирования. Особую актуальность такое моделирование имеет в области систем виртуального окружения и имитационно-тренажерных комплексов. Можно выделить несколько ключевых преимуществ, которые достигаются в результате замены реальных объектов на их виртуальные модели. Одно из них заключается в возможности обучения операторов управлять сложными и зачастую дорогими техническими средствами без угрозы их поломки ввиду недостатка опыта. После получения и отработки базовых навыков на виртуальной модели вероятность испортить настоящее оборудование существенно снижается. Кроме того, привлечение технологий виртуального моделирования в разы уменьшает финансовые затраты на закупку и обслуживание оборудования. Одна виртуальная модель может достаточно легко быть масштабирована на большое число обучающихся, а также скорректирована при изменении своего реального прототипа. Расширяется и спектр возможностей тренировки. Например, заметно проще и безопаснее имитировать многие нештатные ситуации.

Правильность и качество навыков, привитых оператору путем обучения на виртуальных моделях, во многом зависит от того, насколько синтезируемая на компьютере трехмерная виртуальная среда близка к окружающей нас действительности. Важным элементом такой среды являются модели устройств наблюдения. Например, дистанционно управляемая реальная техника в большинстве случаев снабжена видеока-

мерами, позволяющими оператору контролировать окружающее пространство, оценивать ситуацию, принимать решения и получать обратную визуальную связь в ответ на свои управляющие воздействия. Реализация таких устройств в виртуальной среде обычно производится на базе виртуальной камеры с имитацией для нее физических параметров реальных объективов [1, 2]. Если речь идет о моделировании техники с устройствами наблюдения, находящейся на открытых территориях вне помещений, то важное значение имеет имитация сложных метеорологических условий [3, 4]. Попадание атмосферных осадков на камеру может существенно ухудшать получаемую с нее картинку и затруднять принятие оператором решений. Поэтому моделирование таких ситуаций требует особого внимания при выполнении тренировок.

В данной работе предлагаются оригинальные решения для распределенного рендеринга на графическом процессоре (GPU) в масштабе реального времени изображений трехмерной виртуальной среды, видимых с помощью моделей реальных устройств наблюдения, с учетом воздействия атмосферных осадков в виде дождя на стекло объектива виртуальной камеры.



Рис. 1. Шумовая RGB-текстура

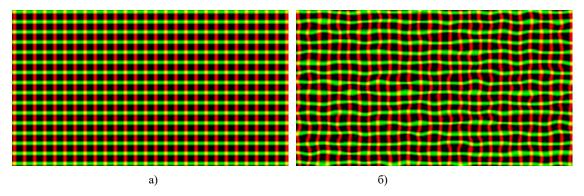


Рис. 2. Прямая (а) и искаженная (б) сетки для рендеринга капель

# 2. Моделирование капель

Предлагаемый подход для имитации капель дождя на объективе виртуального устройства наблюдения заключается в распределенной постобработке изображений трехмерной сцены, видимых виртуальной камерой и синтезируемых посредством имеющихся в системе визуализации механизмов рендеринга. Постобработка выполняется с помощью специального шейдера на современном многоядерном GPU параллельно во множестве потоков, где каждый поток отвечает за свой пиксел кадра. На вход такого шейдера загружается исходное изображение, а также шумовая RGB-текстура  $T_{\rm n}$  (рис. 1), пикселы которой окрашены случайным образом. На выходе имеем результирующее изображение с требуемой имитацией капель. Далее рассмотрим подробнее предлагаемую постобработку.

В процессе своего выполнения шейдер производит построение q невидимых сеток в области обрабатываемого изображения, в узлах которых моделируются водяные капли. Каждая сетка имеет свой размер S ячейки, определяющий величину капли в момент ее появления. Пусть W и H — ширина и высота изображения (в пикселах),  $P = (i, j), i \in [0, W], j \in [0, H]$  — координаты обрабатываемого потоком GPU пиксела (с началом отсчета от левого нижнего угла изображения),  $S_{\text{max}}$  — максимальный размер ячейки сетки (в пикселах), соответствующий самым крупным каплям. Введем также целочисленный коэффициент  $k \in [1, q]$ , задающий размер ячейки k-ой сетки

$$S_k = \frac{S_{\text{max}}}{k} \, .$$

Таким образом, чем больше k, тем мельче сетка и моделируемые в ее узлах дождевые капли.

Построение сетки выполняется на основе синусоидальной функции

$$G = \left(\sin F_x, \sin F_y\right), F = \frac{2\pi \cdot P}{S_b}.$$
 (1)

В качестве ее узлов будем рассматривать те пикселы, для которых оба компонента G максимальны и равны 1. Это возможно в том случае,

когда 
$$m{F} = \left(m rac{\pi}{2}, n rac{\pi}{2}
ight)$$
 , где  $m, \, n \, \in \mathbb{Z}$  . Рисунок

2а иллюстрирует пример описанной сетки. На нем красные линии соответствуют максимальным значениям  $G_x$ , зеленые —  $G_y$ , а точки их пересечения являются узлами. Однако генерируемые на регулярной сетке капли будут выглядеть одинаково. Чтобы придать им различную форму, синтезируемая сетка искажается (рис. 26) путем внесения в выражение для F из формулы (1) псевдослучайного вектора N:

$$F = \frac{2\pi \cdot P}{S_k} + N .$$

В качестве вектора N берутся значения  $C_{n,r}$  красной и  $C_{n,g}$  зеленой компонент цвета  $C_n$  тексела шумовой текстуры  $T_n$ , приведенные к диапазону [-1.0, 1.0]:

$$N = \begin{pmatrix} 2 & 0 & -1 \\ 0 & 2 & -1 \end{pmatrix} \begin{pmatrix} C_{n,r} \\ C_{n,g} \\ 1 \end{pmatrix},$$

$$C_{n} = \left(C_{n,r}, C_{n,g}, C_{n,b}\right) = T_{n} \left(0.1 \frac{P_{x}}{W}, 0.1 \frac{P_{y}}{H}\right),$$

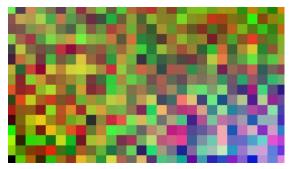


Рис. 3. Наложение шумовой текстуры на область вывода изображения сцены

где  $T_n(x, y)$  — цвет, получаемый из  $T_n$  по текстурным координатам (x, y); W, H — ширина и высота обрабатываемого изображения (в пикселах).

Кроме получения вектора N для искажения сетки, шумовая текстура используется шейдером с целью задания псевдослучайных векторов D, влияющих на параметры моделируемых капель (форму, количество и т.д.). Каждый вектор D отвечает целиком за одну каплю. Следовательно, он должен применяться при обработке всех пикселов из области изображения, ассоциированной с этой каплей. Для *k*-ой сетки такой областью является квадрат со стороной  $S_k$  и центром в узле. Поэтому при получении вектора D с помощью текстуры  $T_{\rm n}$  необходимо осуществить такую выборку, чтобы каждый тексел  $T_{\rm n}$  покрывал площадь вокруг соответствующего узла сетки, равную площади ячейки (рис. 3). Учитывая определение G и F (см. (1)), легко заметить, что левый нижний узел сетки смещен относительно левого нижнего угла изображения на 1/4 размера  $S_k$  ячейки по обеим осям (вправо и вверх). Таким образом, для совмещения центров текселов шумовой текстуры с узлами сетки, при расчете текстурных координат необходимо скорректировать положение P обрабатываемого пиксела на  $0.25S_k$ . Тогда

$$\begin{split} &D = \left(D_{r}, D_{g}, D_{b}\right) = \\ &= T_{n} \left(\frac{\left[\left(P_{x} + 0.25S_{k}\right) / S_{k}\right]}{W_{n}}, \frac{\left[\left(P_{y} + 0.25S_{k}\right) / S_{k}\right]}{H_{n}}\right) = \\ &= T_{n} \left(\frac{\left[P_{x} / S_{k} + 0.25\right]}{W_{n}}, \frac{\left[P_{y} / S_{k} + 0.25\right]}{H_{n}}\right), \end{split}$$

где  $W_{\rm n}, H_{\rm n}$  – ширина и высота текстуры  $T_{\rm n}$  (в пикселах), а квадратные скобки обозначают взятие целой части числа.

На основе значений компонент векторов G и D, а также времени T, прошедшего с момента запуска приложения, шейдер вычисляет для каждого пиксела специальный параметр t по следующим формулам:

$$l = 1 - v \left\{ T \cdot D_b + D_g \right\},$$

$$t = \begin{cases} \left( G_x + G_y \right) l, \ l > 0 \\ 0, \ l \le 0 \end{cases},$$

где  $v \ge 0$  – коэффициент, задающий скорость высыхания капель, а фигурные скобки обозначают взятие дробной части вещественного числа. Далее выполняется сравнение t с некоторым пороговым значением  $\lambda \in [0.0, 2.0]$  (по проведенным в данной работе экспериментам оптимальное  $\lambda = 0.5$ ). Если  $t < \lambda$ , будем считать, что рассматриваемый пиксел не попадает в область моделирования капли, поэтому его цвет остается таким же, как в исходном изображении, а обработка прекращается. Применение порогового значения по параметру t выделяет контур моделируемой капли вокруг соответствующего ей узла сетки и обеспечивает динамику размера капли (имитация высыхания) с течением времени.

Необходимо отметить, что моделирование капель во всех узлах сетки приведет к получению нереалистичного изображения. Чтобы решить данную проблему, шейдер, используя псевдослучайный параметр из шумовой текстуры  $T_{\rm n}$ , выбирает лишь часть узлов, в которых и будут



Рис. 4. Моделирование капель дождя на объективе виртуальной камеры

рисоваться капли. При этом коэффициент k размера ячейки сетки учитывается таким образом, что доля узлов с меньшими по размеру каплями превосходит долю узлов с большими. Выборка осуществляется путем проверки для рассматриваемого пиксела неравенства

$$(1+q-k)\cdot 0.08 > D_r$$
, (2)

где q — общее количество формируемых шейдером сеток с разными размерами ячеек,  $D_r$  — r-компонент вектора D. Если оно не соблюдается, обработка пиксела прекращается, а его цвет остается таким же, как в исходном изображении.

При выполнении условия  $t \ge \lambda$  и соблюдении неравенства (2) вычисляется вектор V смещения текстурных координат для обрабатываемого пиксела, имитирующий рефракцию в дождевой капле:

$$V = 0.3 \frac{v}{|v|}, v = (\cos F_x, \cos F_y, t).$$

Тогда цвет  $C_{\text{res}}$  рассматриваемого пиксела изображения для k-ой сетки определяется шейдером по формуле:

$$C_{\text{res}} = C_{\text{imo}} \left( P_{x} - V_{x}, P_{y} - V_{y} \right),$$

где  $C_{\text{img}}(x, y)$  – цвет пиксела (x, y) исходного изображения.

Для моделирования капель разных размеров проводится построение q сеток с коэффициентами k размеров ячеек от 1 до q. Итоговый цвет пиксела, в который попадают одновременно несколько капель разных размеров, определяется как  $C_{\rm res}$ , вычисленный для меньшего значения k.

# 3. Результаты

На основе полученных в работе решений были созданы программные модули для системы визуализации, обеспечивающие распределенный синтез изображения трехмерных сцен в мас-

штабе реального времени с моделированием попадания дождевых капель на объектив виртуального устройства наблюдения. Программные компоненты были реализованы с использованием графической библиотеки OpenGL, шейдерного языка GLSL последнего поколения и принципов распределенных вычислений на современных многоядерных GPU.

Данные модули прошли успешную апробацию в составе системы визуализации «GLView» [5] и программного комплекса виртуального окружения VirSim [6], созданных в ФГУ ФНЦ НИИСИ РАН. Результаты апробации показали возможность применения разработанных подходов при реализации программного обеспечения для систем виртуального окружения и имитационно-тренажерных комплексов.

На рисунке 4 представлен пример визуализации трехмерной сцены виртуального полигона с моделированием капель дождя на виртуальной камере.

#### 4. Заключение

В данной статье рассмотрена задача моделирования работы виртуальных устройств наблюдения в сложных метеорологических условиях, когда объектив камеры подвержен воздействию атмосферных осадков в виде дождя. Для ее решения предложены оригинальные методы и подходы, основанные на распределенной постобработке изображений с помощью современных GPU в масштабе реального времени. Полученные результаты могут быть использованы в имитационно-тренажерных комплексах, системах виртуального окружения, анимации, компьютерных играх и т.д.

Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 19-07-00393.

# Image Synthesis of 3D Scenes with Simulation of Raindrops on Virtual Camera's Lens

# **Andrey Maltsev**

**Abstract.** The paper proposes solutions for visualization of three-dimensional virtual environment with distributed simulation of raindrops on virtual surveillance device's lens. Developed methods and approaches are based on using parallel computing by means of modern multicore graphics processors and provide real-time visualization of virtual scenes.

**Keywords:** three-dimensional scene, virtual camera, rendering, raindrops, simulation, shaders, graphics processor, post-processing

# Литература

- 1. А.В. Мальцев, Л.А. Финагин. Методы моделирования параметров и фокусировки камеры в виртуальном окружении. «Труды НИИСИ РАН», Том 10 (2020), № 3, 29-33.
- 2. А.В. Мальцев. Методы моделирования световых эффектов и искажений видеосигнала в виртуальных средствах наблюдения. «Информационные технологии и вычислительные системы», 2019, № 2, 54-61.
- 3. P. Fearing. Computer modelling of fallen snow. In Proceedings of the 27th annual conference on Computer graphics and interactive techniques, 2000, 37-46.
- 4. А.В. Мальцев. Моделирование атмосферных осадков в трехмерных сценах с использованием CUDA. «Информационные технологии и вычислительные системы», 2015, № 2, 31-39.
- 5. М.В. Михайлюк, М.А. Торгашев. Система визуализации «GLView» для имитационно-тренажерных комплексов и систем виртуального окружения. Труды Международной научной конференции Графикон-2015, 2015, 96-101.
- 6. М.В. Михайлюк, А.В. Мальцев, П.Ю. Тимохин, Е.В. Страшнов, Б.И. Крючков, В.М. Усов. Система виртуального окружения Virsim для имитационно-тренажерных комплексов подготовки космонавтов. «Пилотируемые полеты в космос», Т. 37 (2020), № 4, 72-95.

# Инверсная кинематика для управления движением двуногих шагающих роботов в системах виртуального окружения

**Е.В.** Страшнов<sup>1</sup>

<sup>1</sup>ФГУ ФНЦ НИИСИ РАН, Москва, Россия, strashnov\_evg@mail.ru

Аннотация. В работе рассматриваются методы и алгоритмы решения задачи инверсной кинематики для управления двуногими шагающими роботами в системах виртуального окружения. Предлагается подход, в котором генерация шаблона ходьбы осуществляется путем вычисления углов в сочленениях ног робота для заданных траекторий движения его центра масс и стоп. Для реализации инверсной кинематики задействован итерационный метод Левенберга-Марквардта с применением коэффициентов затухания, что обеспечивает его численную устойчивость и сходимость. Апробация предлагаемых в статье решений проводилась в программном комплексе виртуального окружения на примере управления двуногим шагающим роботом в полуавтоматическом режиме и показала их адекватность и эффективность для реализации движения робота с сохранением равновесия.

**Ключевые слова:** двуногий шагающий робот, инверсная кинематика, генерация шаблона ходьбы, метод Левенберга-Марквардта, система виртуального окружения

# 1. Введение

Одним из перспективных направлений исследований является антропоморфная робототехника, изучающая методы построения и управления человекоподобными роботами. В последние годы во многих странах ведется активная разработка двуногих шагающих роботов (см. [1], [2], [3]), которые по своему строению подобны человеку. Это связано с тем, что антропоморфные роботы наиболее универсальны для их возможного применения в различных областях, куда входит образование, медицина, строительство и т.д. Такие роботы могут заменить человека по выполнению рутинных и повторяющихся работ (например, уборка помещений, сварка, проведение хирургических операций и т.п.). Кроме того, в настоящее время ведутся исследования по применению антропоморфных роботов в экстремальных условиях [4], опасных для здоровья и жизни человека. Примером служит ликвидация последствий техногенных аварий и катастроф, тушение пожаров, работы в условиях космоса и на поверхности других планет. Для успешного выполнения этих задач антропоморфный робот должен быть способен передвигаться внутри рабочего пространства с помощью ходьбы. В связи с этим создание систем управления для реализации ходьбы антропоморфных роботов с сохранением равновесия является важной и актуальной задачей.

Задача управления движением двуногого шагающего робота приводит к необходимости построения траекторий для его центра масс и стоп.

Для генерации шаблона ходьбы требуется решать задачу инверсной кинематики, которая заключается в вычислении углов в сочленениях педипуляторов (ног) робота в зависимости от его текущей конфигурации. Обзор существующих решений задачи инверсной кинематики приведен в работах [5] и [6] на примере методов, активно используемых в компьютерной графике и робототехнике. Среди них выделяются численные итерационные методы, которые условно подразделяются на три группы: методы Якоби, эвристические методы и методы Ньютона. В методах Якоби для нахождения решения используется транспонированная или псевдообратная матрица Якоби. Эти методы обладают слабой сходимостью, плохой устойчивостью и приводят к неестественным конфигурациям робота. Примером эвристического подхода для расчета инверсной кинематики является метод покоординатного спуска [7], который прост в реализации и требует мало вычислений. Однако, этот метод не пригоден для шагающих роботов и генерации шаблона ходьбы. В методах Ньютона задача формулируется, как поиск минимума функции относительно невязки. Проблема в том, что эти методы не гарантируют глобальную сходимость и зависят от начального значения. Альтернативой является метод Левенберга-Марквардта [8], который наиболее эффективен для решения задачи минимизации. В этом методе можно регулировать численную устойчивость путем выбора коэффициентов затухания.

В данной статье рассматриваются методы и подходы реализации инверсной кинематики для

управления двуногим шагающим роботом в системах виртуального окружения. Предлагается подход, в котором генерация шаблона ходьбы для реализации прямолинейного движения робота по ровной поверхности осуществляется путем построения траекторий для его центра масс и стоп. Для этих траекторий с помощью инверсной кинематики вычисляются углы в сочленениях ног робота. При этом, для обеспечения динамической устойчивости движения робота задействована математическая модель обратного маятника с виртуальной высотой [9]. Предлагаемые в статье решения были апробированы в программном комплексе VirSim [10] на примере моделирования созданной виртуальной модели двуногого шагающего робота [11].

### 2. Постановка задачи

Двуногий шагающий робот (ДШР) представляет собой механическое устройство, которое по своему строению подобно человеку. Такие роботы, как правило, имеют манипуляторы для захвата предметов и педипуляторы (ноги) для ходьбы. Управление движением ДШР заключается в том, чтобы обеспечить его перемещение с сохранением равновесия.

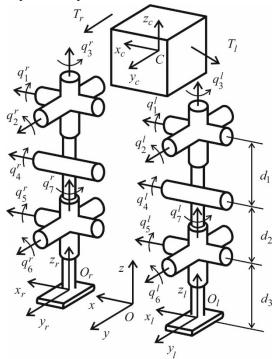


Рис. 1. Кинематическая структура ног двуногого шагающего робота

Рассмотрим ДШР, кинематическая структура ног которого показана на рис. 1. Движение робота описывается с помощью обобщенных координат  $q_i^l$  и  $q_i^r$ ,  $i = \overline{1, N}$  (углов в сочленениях ног

робота), где N — число степеней свободы ноги робота. В схеме на рис. 1 каждая нога робота имеет по семь степеней свободы (три для бедра, одну для колена и три для стопы). Для описания движения робота рассмотрим систему координат  $Cx_cy_cz_c$  его торса, где точка C является центром масс робота, и системы координат  $O_lx_ly_lz_l$  и  $O_rx_ry_rz_r$  его стоп. Все координаты задаются относительно мировой системы координат (МСК)  $Oxyz_r$ , которая для удобства находится посередине между стопами ног робота в его изначальном положении.

Координаты торса ДШР задаются с помощью матрицы трансформации  $\mathbf{T}_c$  (4 на 4), описывающей переход из МСК в систему координат торса. Эта матрица имеет следующий вид

$$\mathbf{T}_c = \begin{pmatrix} \mathbf{R}_c & \mathbf{p}_c \\ 0 & 1 \end{pmatrix}, \tag{1}$$

где  $\mathbf{R}_c$  и  $\mathbf{p}_c$  — матрица поворота и вектор положения торса робота.

В свою очередь, матрицы трансформации  $\mathbf{T}_{f,l}$  и  $\mathbf{T}_{f,r}$  для стоп робота имеют вид

$$\mathbf{T}_{f,l} = \begin{pmatrix} \mathbf{R}_{f,l} & \mathbf{p}_{f,l} \\ 0 & 1 \end{pmatrix}, \ \mathbf{T}_{f,r} = \begin{pmatrix} \mathbf{R}_{f,r} & \mathbf{p}_{f,r} \\ 0 & 1 \end{pmatrix}, \ (2)$$

где  $\mathbf{R}_{f,l}$  и  $\mathbf{R}_{f,r}$  – матрицы поворота стоп робота,  $\mathbf{p}_{f,l}$  и  $\mathbf{p}_{f,r}$  – векторы положения стоп робота.

Задача инверсной кинематики для ДШР формулируется следующим образом: требуется найти углы  $q_i^I$  и  $q_i^r$ ,  $i=\overline{1,N}$  в сочленениях ног робота для заданных координат его торса и стоп, которые задаются с помощью матриц  $\mathbf{T}_c$ ,  $\mathbf{T}_{f,I}$  и  $\mathbf{T}_{f,I}$ .

Далее опишем предлагаемые решения этой задачи, включая прямую кинематику и генерацию шаблона ходьбы для ДШР.

# 3. Прямая кинематика

Задача прямой кинематики для ДШР заключается в том, что, зная координаты торса робота (положение и ориентация) и углы в сочленениях ног робота, необходимо найти координаты стоп робота. Другими словами, требуется выразить  $\mathbf{T}_{t,l}$  и  $\mathbf{T}_{t,r}$  через  $\mathbf{T}_c$ ,  $q_i^l$  и  $q_i^r$ ,  $i=\overline{1,N}$ .

На рис. 1 показано, что бедра ног робота задаются относительно его торса посредством матриц трансформации  $T_t$  и  $T_r$ , которые имеют вид

$$\mathbf{T}_{l} = \begin{pmatrix} 1 & 0 & 0 & d_{lx} \\ 0 & 1 & 0 & d_{ly} \\ 0 & 0 & 1 & d_{lz} \\ 0 & 0 & 0 & 1 \end{pmatrix}, \ \mathbf{T}_{r} = \begin{pmatrix} 1 & 0 & 0 & d_{rx} \\ 0 & 1 & 0 & d_{ry} \\ 0 & 0 & 1 & d_{rz} \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

где  $d_{lx}$ ,  $d_{ly}$ ,  $d_{lz}$  – компоненты вектора между центром масс робота и бедром его левой ноги;  $d_{rx}$ ,  $d_{ry}$ ,  $d_{rz}$  – компоненты вектора между центром масс робота и бедром его правой ноги.

Пусть величины  $d_k$ ,  $k=1,\overline{M}$  задают длины звеньев ног робота, где M – число звеньев. Тогда матрицы трансформации  $\mathbf{T}_{f,l}$  и  $\mathbf{T}_{f,r}$  вычисляются в виде произведения матриц:

$$\mathbf{T}_{f,l} = \mathbf{T}_{l} \prod_{k=1}^{M} \mathbf{T}_{\varphi,k}^{x} \mathbf{T}_{\psi,k}^{y} \mathbf{T}_{\theta,k}^{z} \mathbf{T}_{dk} ;$$

$$\mathbf{T}_{f,r} = \mathbf{T}_{r} \prod_{k=1}^{M} \mathbf{T}_{\varphi,k}^{x} \mathbf{T}_{\psi,k}^{y} \mathbf{T}_{\theta,k}^{z} \mathbf{T}_{dk} ,$$
(3)

где  $\phi$  ,  $\psi$  и  $\theta$  соответствуют углам поворота  $q_i^I$  и  $q_i^r$  ,  $i=\overline{1,N}$  .

В соотношениях (3) матрицы вида  $\mathbf{T}_{\phi}^{x}$ ,  $\mathbf{T}_{\psi}^{y}$  и  $\mathbf{T}_{\theta}^{z}$  задают повороты вокруг осей x, y и z и определяются как

$$\begin{split} \mathbf{T}_{\phi}^{x} &= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi & 0 \\ 0 & \sin \phi & \cos \phi & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}; \\ \mathbf{T}_{\psi}^{y} &= \begin{pmatrix} \cos \psi & 0 & \sin \psi & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \psi & 0 & \cos \psi & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}; \\ \mathbf{T}_{\theta}^{z} &= \begin{pmatrix} \cos \theta & -\sin \theta & 0 & 0 \\ \sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \end{split}$$

В свою очередь матрицы  $\mathbf{T}_{dk}$  вычисляются следующим образом

$$\mathbf{T}_{dk} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_k \\ 0 & 0 & 0 & 1 \end{pmatrix}, \ k = \overline{1, M} \ .$$

Таким образом, с использованием (2) и (3) можно получить координаты (положения и ориентации) стоп робота.

# 4. Инверсная кинематика

Рассмотрим решение задачи инверсной кинематики на примере левой ноги ДШР. Требуется найти углы поворота  $q_i^l$ ,  $i=\overline{1,N}$ , для которых должно быть выполнено

$$\mathbf{p}_{f,l}(\mathbf{q}^l) = \mathbf{p}_{f,l}^d, \ \mathbf{R}_{f,l}(\mathbf{q}^l) = \mathbf{R}_{f,l}^d,$$

где  $\mathbf{p}_{f,l}(\mathbf{q}^l)$  и  $\mathbf{R}_{f,l}(\mathbf{q}^l)$  – положение и ориентация стопы левой ноги робота, которые вычисляются с использованием выражений (2) и (3);  $\mathbf{p}_{f,l}^d$  и  $\mathbf{R}_{f,l}^d$  – целевое положение и целевая ориентация стопы левой ноги робота.

В этом случае вектор невязки по положению стопы определяется как

$$\mathbf{e}_{p}(\mathbf{q}^{l}) = \mathbf{p}_{f,l}^{d} - \mathbf{p}_{f,l}(\mathbf{q}^{l}). \tag{4}$$

Вектор невязки относительно ориентации задается следующим образом

$$\mathbf{e}_{o}(\mathbf{q}^{l}) = \mathbf{a} \left( \mathbf{R}_{f,l}^{d} \mathbf{R}_{f,l}(\mathbf{q}^{l}) \right). \tag{5}$$

Здесь вектор  $\mathbf{a}(\mathbf{R})$  задает эквивалентное представление ориентации в виде мгновенной оси поворота. Если матрица  $\mathbf{R}$  не является диагональной, то этот вектор вычисляется как

$$\mathbf{a}(\mathbf{R}) = \frac{\arctan 2(\|\mathbf{v}\|, r_{11} + r_{22} + r_{33} - 1)}{\|\mathbf{v}\|} \mathbf{v}, \quad (6)$$

где  $\mathbf{R} = \{r_{ii}\}, i, j = \overline{1,3}, \text{ а вектор } \mathbf{v} \text{ находится как}$ 

$$\mathbf{v} = (r_{32} - r_{23}, r_{13} - r_{31}, r_{21} - r_{12})^{\mathrm{T}}.$$

В противном случае возможны следующие варианты. Если матрица единичная, то будет  $\mathbf{a} = \mathbf{0}$ . Иначе возможны еще три варианта, где матрица  $\mathbf{R}$  задает поворот вокруг одной из трех осей на угол  $\pi$ , а вектор  $\mathbf{a}(\mathbf{R})$  определяется как

$$\mathbf{a}(\mathbf{R}) = \frac{\pi}{2} (r_{11} + 1, r_{22} + 1, r_{33} + 1)^{\mathrm{T}}.$$
 (7)

Выражения (4) и (5) используются для вычисления общей невязки, которая объединяется в вектор  $\mathbf{e}(\mathbf{q}^l) = \left(\mathbf{e}_p(\mathbf{q}^l), \mathbf{e}_o(\mathbf{q}^l)\right)^{\mathrm{T}}$ .

Для вывода основных соотношений инверсной кинематики ДШР сначала рассмотрим матрицу Якоби  $\mathbf{J}(\mathbf{q}^{l})$  для координат стопы робота, которая вычисляется путем дифференцирования невязки  $\mathbf{e}(\mathbf{q}^{l})$  по  $\mathbf{q}^{l}$ :

$$\mathbf{J} = -\left[\frac{\partial \mathbf{e}(\mathbf{q}^l)}{\partial \mathbf{q}^l}\right] = -\left[\frac{\partial \mathbf{e}_p(\mathbf{q}^l)}{\partial \mathbf{q}^l}, \frac{\partial \mathbf{e}_r(\mathbf{q}^l)}{\partial \mathbf{q}^l}\right]. \quad (8)$$

Из соотношения (4) получим, что

$$\frac{\partial \mathbf{e}_{p}(\mathbf{q}^{l})}{\partial \mathbf{q}^{l}} = -\frac{\partial \mathbf{p}_{f,l}(\mathbf{q}^{l})}{\partial \mathbf{q}^{l}}.$$
 (9)

В свою очередь, дифференцирование невязки  $\mathbf{e}_{o}(\mathbf{q}^{t})$  с использованием (5) приводит

$$\frac{\partial \mathbf{e}_{o}(\mathbf{q}^{l})}{\partial \mathbf{q}^{l}} = \frac{\partial \mathbf{a} \left( \mathbf{R}_{f,l}^{d} \mathbf{R}_{f,l}(\mathbf{q}^{l}) \right)}{\partial \mathbf{R}_{f,l}} \frac{\partial \mathbf{R}_{f,l}(\mathbf{q}^{l})}{\partial \mathbf{q}^{l}}. (10)$$

Вычисление частных производных по  $\mathbf{q}^{t}$  в (9) и (10) выполняется путем дифференцирования матрицы трансформации  $\mathbf{T}_{t,t}$  из (2):

$$\frac{\partial \mathbf{T}_{f,l}}{\partial \mathbf{q}^{l}} = \begin{pmatrix} \frac{\partial \mathbf{R}_{f,l}}{\partial \mathbf{q}^{l}} & \frac{\partial \mathbf{p}_{f,l}}{\partial \mathbf{q}^{l}} \\ 0 & 0 \end{pmatrix}. \tag{11}$$

При дифференцировании используется соотношение (3), где частные производные для матриц вида  $\mathbf{T}_{\sigma}^{x}$ ,  $\mathbf{T}_{\psi}^{y}$  и  $\mathbf{T}_{\theta}^{z}$  вычисляются как

$$\begin{split} \frac{\partial \mathbf{T}_{\varphi}^{x}}{\partial \varphi} &= \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & -\sin \varphi & -\cos \varphi & 0 \\ 0 & \cos \varphi & -\sin \varphi & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}; \\ \frac{\partial \mathbf{T}_{\psi}^{y}}{\partial \psi} &= \begin{pmatrix} -\sin \psi & 0 & \cos \psi & 0 \\ 0 & 0 & 0 & 0 \\ -\cos \psi & 0 & -\sin \psi & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}; \\ \frac{\partial \mathbf{T}_{\theta}^{z}}{\partial \theta} &= \begin{pmatrix} -\sin \theta & -\cos \theta & 0 & 0 \\ \cos \theta & -\sin \theta & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}. \end{split}$$

Вычисление частной производной вектора **a** мгновенной оси вращения из (10) осуществляется путем дифференцирования (6) и (7).

Для решения задачи инверсной кинематики воспользуемся итерационным методом Левенберга-Марквардта [8]. В этом методе на каждом шаге итерации искомая задача сводится к поиску минимума функции:

$$F(\Delta \mathbf{q}_{k}^{l}) = \frac{1}{2} \mathbf{r}_{k}^{\mathrm{T}} \mathbf{W}_{e} \mathbf{r}_{k} + \frac{1}{2} \Delta \mathbf{q}_{k}^{l,\mathrm{T}} \mathbf{W}_{n} \Delta \mathbf{q}_{k}^{l} \rightarrow \min, (12)$$
  
$$\Delta \mathbf{q}_{k}^{l} = \mathbf{q}_{k+1}^{l} - \mathbf{q}_{k}^{l}, \ \mathbf{r}_{k} = \mathbf{e}_{k} - \mathbf{J}_{k} \Delta \mathbf{q}_{k}^{l},$$

где k — номер итерации,  $\mathbf{q}_k^l$  — вектор обобщенных координат на k-ой итерации,  $\mathbf{e}_k = \mathbf{e}(\mathbf{q}_k^l)$ ,  $\mathbf{J}_k = \mathbf{J}(\mathbf{q}_k^l)$ ,  $\mathbf{W}_e = diag\left\{w_{e,j}\right\}$ ,  $w_{e,j} > 0$  — весовая матрица,  $\mathbf{W}_n = diag\left\{w_{n,i}\right\}$ ,  $w_{n,i} > 0$  — матрица коэффициентов затухания,  $i = \overline{1, N}$ ,  $j = \overline{1, 6}$ .

Тогда решение задачи (12) осуществляется по следующему правилу:

$$\mathbf{q}_{k+1}^{l} = \mathbf{q}_{k}^{l} + \mathbf{H}_{k}^{-1} \mathbf{g}_{k}, \qquad (13)$$

где  $\mathbf{g}_k = \mathbf{J}_k^{\mathrm{T}} \mathbf{W}_e$ ,  $\mathbf{H}_k = \mathbf{J}_k^{\mathrm{T}} \mathbf{W}_e \mathbf{J}_k + \mathbf{W}_n$  – матрица Гессе (размерностью N на N).

Итерации в (13) с начальным значением  $\mathbf{q}_0^l = \mathbf{0}$  выполняются до тех пор, пока не будет выполнено  $\left|\Delta q_{k,i}^l\right| \leq \varepsilon$ ,  $i = \overline{1,N}$  или превышено максимальное число итераций  $K_{\max}$ , где  $\varepsilon$  — заданная точность.

Чтобы обеспечить сходимость метода, матрица  $W_{n}$  выбирается в следующем виде

$$\mathbf{W}_{n} = (\mathbf{e}_{k}^{\mathrm{T}} \mathbf{W}_{e} \mathbf{e}_{k} + \tilde{w}_{n}) \mathbf{I}_{N}, \qquad (14)$$

где  $\tilde{w}_n$  – коэффициент поправки,  $\mathbf{I}_N$  – единичная матрица (N на N).

Для реализации инверсной кинематики предлагается следующий алгоритм:

- 1. Инициализация  $\mathbf{q}_{0}^{l} = \mathbf{0}$ .
- 2. Цикл по  $k = 1, ..., K_{\text{max}}$ :
- 2.1. Вычисляем невязку  $\mathbf{e}_k = \mathbf{e}(\mathbf{q}_k^l)$  по формулам (4) и (5).
  - 2.2. Находим  $\mathbf{W}_{n}$  по формуле (14).
- 2.3. Дифференцируем  $\mathbf{T}_{f,l}$  по  $\mathbf{q}_k^l$  с использованием (3) и (11).
  - 2.4. Вычисляем  $J_k$  по формулам (8)-(10).
  - 2.5. Находим  $\mathbf{H}_{k}$  и  $\mathbf{g}_{k}$  из (13).
  - 2.6. Вычисляем обратную матрицу  $\mathbf{H}_{k}^{-1}$ .
  - 2.7. Находим  $\mathbf{q}_{k+1}^{l}$  по формуле (13).
- 2.8. Если  $\left|\Delta q_{k,i}^l\right| \leq \varepsilon$ ,  $i=\overline{1,N}$ , то досрочный выход из цикла Конец алгоритма.

Решение инверсной кинематики для вычисления углов правой ноги ДШР осуществляется аналогичным образом.

# 5. Генерация шаблона ходьбы

Прямолинейное движение ДШР по ровной поверхности представляет собой цикличное повторение движений правой и левой ноги робота. Задача управления движением ДШР заключается в обеспечении его перемещения с сохранением равновесия. При этом, предполагается, что робот движется на постоянной высоте, его торс сохраняет вертикальное положение, а стопы всегда параллельны поверхности и их движение выполняется в сагиттальной плоскости.

Генерация шаблона ходьбы ДШР включает задание траекторий движения центра масс робота и его стоп, для которых с применением инверсной кинематики вычисляются изменения углов в сочленениях ног робота. Рассмотрим эту задачу на примере выполнения роботом шага правой ногой. Воспользуемся критерием точки нулевого момента (ТНМ) [12] динамической устойчивости ДШР. Согласно этому критерию, если в процессе движения робота ТНМ находится в пределах выпуклой оболочки точек контакта опорной поверхности ног, то равновесие робота сохраняется. Чтобы использовать критерий ТНМ для управления ДШР воспользуемся моделью обратного маятника с виртуальной высотой [9]. В этой модели координаты ТНМ  $(p_{x}, p_{y})$  связаны с координатами центра масс

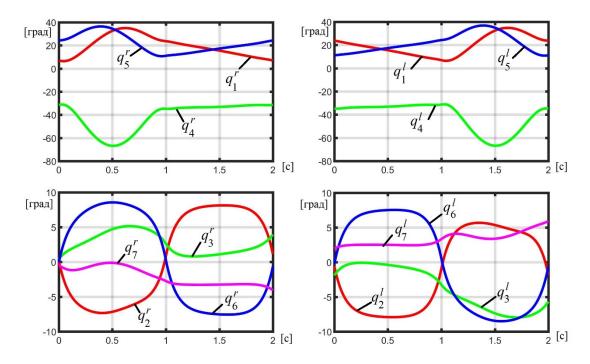


Рис. 2. Углы в шарнирах ног робота

робота  $(x_c, y_c)$  с помощью уравнений

$$p_x = x_c - \frac{\alpha z_0}{g} \ddot{x}_c; \ p_y = y_c - \frac{\beta z_0}{g} \ddot{y}_c,$$
 (15)

где g — ускорение свободного падения,  $z_0$  — высота маятника,  $\alpha$  и  $\beta$  — константы, с помощью которых регулируется высота маятника.

Шаг выполняется с фазой опоры на левую ногу. Поэтому запишем уравнения (15) в системе координат стопы левой ноги робота. Для этого рассмотрим преобразование

$$x_c^l = x_0 + x_c \; ; \; y_c^l = -L + y_c \; ,$$
 (16)

где  $(x_c^l,y_c^l)$  — координаты центра масс робота в системе координат левой стопы,  $x_0$  — половина расстояния между стопами ног робота, L — длина шага.

Тогда в системе координат стопы левой ноги робота будет  $p_x=0$  и  $p_y=0$ , а уравнения (15) преобразуются к виду

$$\ddot{x}_{c}^{l} - \frac{g}{\alpha z_{0}} x_{c}^{l} = 0 \; ; \; \ddot{y}_{c}^{l} - \frac{g}{\beta z_{0}} y_{c}^{l} = 0 \; .$$

Решая эти уравнения и выполняя обратное преобразование к (16), получим траекторию движения центра масс на отрезке [0,T]:

$$x_{c}(t) = -x_{0} + C_{1} \exp(k_{x}t) + C_{2} \exp(-k_{x}t);$$
  

$$y_{c}(t) = L + C_{3} \exp(k_{y}t) + C_{4} \exp(-k_{y}t); (17)$$
  

$$z_{c}(t) = z_{0} = const,$$

где  $k_x=\sqrt{g/\alpha z_0}$  ,  $k_y=\sqrt{g/\beta z_0}$  , T — время выполнения шага,  $C_i$  ,  $i=\overline{1,4}$  — константы, которые вычисляются из граничных условий

$$x_c(0) = x_c(T) = 0$$
;  $y_c(0) = -L/2$ ;  $y_c(T) = L/2$ .

Движение стопы правой ноги задается в параметрическом виде:

$$x_{r}(t) = x_{0} = const;$$

$$y_{r}(t) = L\left(1 - \cos\frac{\pi t}{T}\right);$$

$$z_{r}(t) = \frac{H}{2}\left(1 - \cos\frac{2\pi t}{T}\right),$$
(18)

где H – высота подъема стопы.

Аналогично (17) и (18) задаются траектории при выполнении шага левой ноги робота.

# 6. Результаты моделирования

Предложенные в статье методы, алгоритмы и подходы для управления движением ДШР с применением инверсной кинематики были реализованы в программном комплексе виртуального окружения VirSim [10], разработанном в ФГУ ФНЦ НИИСИ РАН. Моделирование проводилось на примере созданной виртуальной модели робота [11], которая соответствует реальному роботу Skybot F-850. Генерация шаблона ходьбы ДШР осуществляется по траекториям (17) и (18) с параметрами T = 1 с, L = 20 см, H = 10 см,  $\alpha = 10^{-3}$ ,  $\beta = 1$ . Эти траектории участвуют для

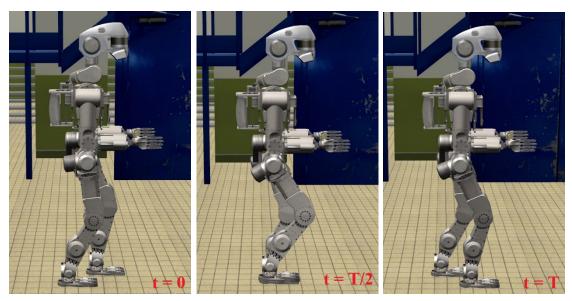


Рис. 3. Моделирование движения модели ДШР в виртуальной сцене

вычисления углов в сочленениях ног робота путем решения задачи инверсной кинематики с помощью метода Левенберга-Марквардта. Для этого были выбраны следующие параметры:  $K_{\max}=10000$ ,  $\epsilon=10^{-12}$ ,  $\tilde{w}_n=10^{-3}$ ,  $w_{e,j}=0.01$ ,  $w_{e,k}=1$ ,  $j=\overline{1,3}$ ,  $k=\overline{4,6}$ . На рис. 2 показаны траектории изменения углов  $q_i^r(t)$  и  $q_i^l(t)$ ,  $i=\overline{1,N}$  ног робота при выполнении шагового цикла, где сначала выполняется шаг правой ногой, а затем левой. Эти траектории являются целевыми для ПД-регуляторов, с помощью которых вычисляются напряжения, подаваемые на электроприводы в сочленениях ног робота.

Апробация предлагаемых в статье решений была проведена в виртуальной сцене помещения, где ДШР может свободно перемещаться. Управление роботом осуществляется в полуавтоматическом режиме путем задания команд для основных движений (вперед, остановка, поворот вправо, поворот влево). На рис. 3 показаны положения робота в саггитальной плоскости (в моменты времени t=0, t=T/2 и t=T) при вы-

полнении шага правой ногой. Результаты апробации в программном комплексе VirSim показали, что предложенные методы и подходы для управления ДШР с применением инверсной кинематики позволяют реализовать движение робота с сохранением равновесия.

# 6. Заключение

В данной работе рассмотрены решения для реализация инверсной кинематики в задаче управления движением двуногого шагающего робота. Предложенный подход на основе генерации шаблона ходьбы позволяет обеспечить динамическую устойчивость робота в процессе его движения по ровной поверхности. Полученные в работе результаты могут быть использованы для решения различных практических задач в системах виртуального окружения, имитационно-тренажерных комплексах, образовательных приложениях, анимации и т.д.

Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 20-07-00371.

# Inverse Kinematics for Motion Control of Bipedal Walking Robots in Virtual Environment Systems

### **Evgeny Strashnov**

**Abstract.** This paper considers methods and approaches for inverse kinematics solution to control of bipedal walking robots in virtual environment systems. An approach is proposed, in which walking pattern generation is realized by joint angles computation of robot legs for given center of mass and feet trajectories. To implement inverse

kinematics, the iterative Levenberg-Marquardt method with damping factors is used that ensure its numerical stability and convergence. Approbation of solutions proposed in this paper was carried out in software complex of virtual environment on the example of bipedal walking robot control in semi-automatic mode and showed their adequacy and efficiency for implementing the robot motion while keeping its balance.

**Keywords:** bipedal walking robot, inverse kinematics, walking pattern generation, Levenberg-Marquardt method, virtual environment system

# Литература

- 1. И.А. Бочков. Обзор способов реализации ходьбы двуногого шагающего робота. «Политехнический молодежный журнал МГТУ им Н.Э. Баумана», Т. 36 (2019), № 7, 7 с.
- 2. S. Saeedvand, M. Jafari, H.S. Aghdasi, and J. Baltes. A comprehensive survey on humanoid robot development. «The Knowledge Engineering Review», vol. 34 (2019), no. 20, 1-18.
- 3. С.П. Хурс, А.А. Верейкин. Состояние и тенденции развития антропоморфной робототехники. «Машины и установки: проектирование, разработка и эксплуатация», 2016, 25-52.
- 4. Е.И. Юревич, Основы робототехники: учеб. пособие, 4-е изд., перераб. и доп. СПб: БХВ-Петербург, 2017, 304 с.
- 5. S.R. Buss. Introduction to inverse kinematics with Jacobian transpose, pseudoinverse and damped least squares methods. Technical report, University of California, 2009.
- 6. A. Aristidou, J. Lasenby, Y. Chrysanthou, and A. Shamir. Inverse kinematics techniques in computer graphics: A survey. «Computer Graphics Forum», vol. 37 (2018), no. 6, 35–58.
- 7. B. Kenwright. Inverse kinematics Cyclic coordinate descent (CCD). «Journal of Graphics Tools», vol. 16 (2012), no. 4, 177-217.
- 8. T. Sugihara. Solvability-unconcerned inverse kinematics based on Levenberg-Marquardt method with robust damping. «9th IEEE-RAS International Conference on Humanoid Robots», Paris, France, 2009, 555-560.
- 9. C.-S. Park, T. Ha, J. Kim, and C.-H. Choi. Trajectory generation and control for a biped robot walking upstairs. «International Journal of Control, Automations and Systems», vol. 8 (2010), no. 2, 339-351.
- 10. М.В. Михайлюк, А.В. Мальцев, П.Ю. Тимохин, Е.В. Страшнов, Б.И. Крючков, В.М. Усов. Система виртуального окружения VirSim для имитационно-тренажерных комплексов подготовки космонавтов. «Пилотируемые полеты в космос», Т. 37 (2020), № 4, 72-95.
- 11. Е.В. Страшнов, И.Н. Мироненко, Л.А. Финагин. Командный режим управления виртуальным двуногим шагающим роботом. «Труды НИИСИ РАН», Т. 10 (2020), № 4, 33-39.
- 12. M. Vukobratovic, and B. Borovac. Zero-moment point thirty-five years of its life. «International Journal of Humanoid Robotics», 1 (2004), 157-173.

# Исследование и разработка методов машинного обучения и архитектур нейронных сетей для применения в области проверки

М.С.Дьяченко<sup>1</sup>, А.Г. Леонов<sup>2</sup>, Матюшин М.А.<sup>3</sup>

<sup>1</sup> ФГУ ФНЦ НИИСИ РАН, Москва, Россия, mdyachenko@niisi.ru

<sup>2</sup> ФГУ ФНЦ НИИСИ РАН, Москва, Россия, МГУ им. М. В. Ломоносова, Москва, Россия, МПГУ, Москва, Россия, Государственный университет управления, Москва, Россия, dr.1@vip.niisi.ru;

<sup>3</sup> ФГУ ФНЦ НИИСИ РАН, Москва, Россия, itsaprank@yandex.ru

Аннотация. Мировые тенденции в области цифровизации образования требуют переноса существенной доли образовательного процесса в цифровые образовательные среды с автоматической проверкой выполненных слушателями заданий. Обратная сторона автоматической верификации — это использование обучающимися чужих решений в своих ответах, выдавая их за свои. В данной статье рассматривается задача автоматической и полуавтоматической детекции заимствований на примере сдачи решений в практикумах по программированию. В качестве подхода к исследованию был выбран путь построения предиктивных моделей вероятности заимствования. Были проведены исследования применимости методов глубинного машинного обучения к решению задачи выявления заимствования, а также рассмотрен ряд эвристических методик, призванных улучшить точность и расширить рамки использования построенных глубоких моделей.

**Ключевые слова:** антиплагиат, машинное обучение, глубокое обучение, нейронные сети, распознавание программных текстов, детектирование заимствований

#### 1. Введение

Задача детектирования плагиата в программных кодах в последнее время нередко упоминается в литературе [1], [2]. Наиболее общая постановка рассматриваемой проблемы звучит следующим образом. Пусть имеется практикум по программированию из нескольких заданий, решениями которых являются программные коды. В ситуации, когда несколько студентов одновременно проходят данный практикум, нередки случаи заимствования, или, по-другому, плагиата кодов решений. Связано это зачастую с тем, что набор вариантов заданий ограничен, при чем их количество заведомо меньше, чем суммарный поток студентов, решающих практикум за все время его существования. Не является редкой и ситуация, когда вариантов заданий меньше, чем студентов, проходящих практикум одновременно.

В этих ситуациях становится неизбежной практика выдачи разным студентам одинаковых заданий, что в свою очередь приводит к возможности их сдачи путем простого копирования чужого успешного решения, при необходимости, с минимальными несущественными изменениями

Очевидным подходом к решению данной

проблемы является автоматическая генерация индивидуальных заданий, при которой каждый студент получает уникальное задание. В этом случае решение заведомо не может быть получено копированием решения любой другой задачи. У данного подхода имеются свои преимущества и свои недостатки. К последним относится, например, повышенная нагрузка на преподавателей, при которой педагогу придется решать нетривильную проблему - метапрограммирование заданий, фактически вводя параметризацию условий задач в каждый практикум.

Авторами был выбран метод построения предиктивной модели вероятности заимствования сданного программного текста при условии сданных до настоящего момента программных текстов и проведено исследования его применимости к решению задачи заимствования. Для технической реализации моделей были выбраны методы глубинного машинного обучения

# 2. Методы решения

Для решения проблемы нечестного заимствования исходного кода программ возможно применение нескольких основных методик или их комбинаций:

- Сравнение исполняемых файлов побайтово или построчно.
- Анализ стиля написания программного кода [1].
  - Структурный анализ программы [3].
- Анализ аномальной активности при прохождении обучаемым заданий [4].

Одним из начальных базовых алгоритмов может служить точное сравнение исполняемых файлов. В то время как явным преимуществом является простота реализации и высокая скорость работы (при большом потоке студентов), очевидными недостатками является, во-первых, узкая область применимости, а во-вторых, высокая сложность, возникающая при покрытии краевых случаев. Первый недостаток напрямую связан с отсутствием гладкости предсказания, под которым подразумевается невозможность оценить степень заимствования, ведь предсказание данного алгоритма зачастую бинарно. Следствием этого является техническая сложность анализа слабых мест реализованного алгоритма, то есть, сложность, его дальнейшего улучшения. Вторым недостатком является необходимость построения дополнительной, зачастую, сложной логики для покрытия случаев заимствования, отличающихся от точного посимвольного копирования программного текста. Данный метод, тем не менее, как показало наше исследование, может показывать лучшие результаты на небольших выборках студентов, в связи с возможностью сравнительно легкой адаптации под конкретные кейсы заимствования.

Структурный анализ программы, в противовес построчному сравнению исполняемых файлов, представляется математически сложной и малоизученной областью исследований. Основной сложностью здесь является тот факт, что извлечение структурного графа программы из исходного кода напрямую решается только компиляцией последнего. В связи с этим в задачу детектирования заимствования программного кода также вносится дополнительная трудность, связанная с потенциальным использованием разных компиляторов, а также с, возможно, непредсказуемым изменением структурного графа программы при несущественном изменения программного кода в угоду оптимизации процесса исполнения. В то время как для одних компиляторов (например, дсс) последняя особенность является отключаемой опцией, для других она вживлена в сам код компилятора и поддерживается уже на уровне написания кода (например, интерпретаторы языка Python).

Что до анализа аномальной активности обучаемых при прохождении заданий, с одной сто-

роны, методы данного класса выглядят перспективно. Особенно учитывая общий успех поведенческого анализа в смежных областях, таких, как поиск и рекомендательные системы. С другой стороны, для полноценной утилизации машинерии анализа аномальной активности необходимо наличие систем сбора существенного объема поведенческой информации студентов, систем ее хранения и быстрой обработки. Для многих цифровых платформ, предоставляющих функционал прохождения практикумов по программированию, перечисленные представляют серьезную сложность в реализации и являются крупной надстройкой над основным предметом исследования - непосредственно программными кодами.

Учитывая вышесказанное, методы анализа стиля написания программного кода выглядят наиболее перспективно для применения в реальных приложениях. Связано это с тем, что для их утилизации достаточно хранения лишь исходных текстов программ, получающиеся предиктивные модели не являются требовательными с точки зрения ресурсов, а их обучение, как показывает настоящее исследование, возможно без предварительной ручной разметки данных, так как само понятие стиля написание может быть выведено из косвенного признака принадлежности программ одному и тому же автору.

# 2.1 Анализ стиля написания программного кода

Анализ стиля написания программного кода предполагает, что автор решения обладает в каком-то смысле уникальным стилем написания программного кода. Это позволяет связать заимствование решения с близостью стиля сданного решения к стилю одного из уже сданных решений.

Формально для определения близости стилей необходимо сформировать вектор-представление программного текста. Это можно сделать, например, заранее определив набор метрик кода, на основании которых будет выполняться сопоставление. Примерами таких метрик являются тип отступов, постановки операторов и элементов блоков, характер именования и длина имен параметров, функций и классов и др. Сформированный набор метрик затем подается на вход полносвязной нейронной сети [5], которая формирует итоговый вектор представления программного кода.

Дальнейшее использование последнего для обучения модели полносвязной нейрон-

ной сети может быть схематически представлено на рисунке 1.

Здесь под эмбеддером понимается совокупность системы извлечения метрик кода и полносвязной нейронной сети. Под расстоянием понимается любая функция близости между векторами, допускающая гладкую оптимизацию. Запечатленный на рисунке выше процесс оптимизации - настройки весов полносвязной нейронной сети, известен как минимизация трипплет лосса [6].

Формально, последний выражается формулой (1)

Здесь под a, p, n понимаются векторапредставления программных кодов, под d понимается выбранный способ измерения расстояния между векторами. Предполагается, что исходные коды, соответствующие a, p принадлежат одному автору, исходные коды a, n принадлежат разным авторам. Для улучшения результатов сходимости могут быть использованы техники семплинга сложных примеров [7].

После обучения полученная модель эмбеддера позволяет получать вектор-представления программных кодов для новых сдаваемых решений. Считая, что модель доставляет минимум триплет лоссу, можно считать, что векторы решений одного автора будут близки по выбранной метрике d,в то время как векторы решений разных авторов будут далеки по той же метрике. Это позволяет использовать для вероятности заимствования формулу (2)

где минимум берется по всем b - решениям автора B. Параметр масштаба  $\sigma$  является гиперпараметром формулы, настраиваемым вручную на валидационной выборке.

# 2.2 Экспериментальная проверка

Для экспериментальной проверки описанного выше метода был отобран набор из 20 авторов, представивших программные коды в открытый доступ на сайте https://github.com. Для обучения, валидации и тестирования были выбраны программные коды, гарантированно написанные одним рассматриваемым автором. Суммарное число элементов всех выборок составило 217 примеров. Пропорции разделения на тренировочную, валидационную и тестовую выборки составили 50/20/30. Разделение на выборки происходило по авторам, то есть, представленный в одной выборке автор гарантированно не был представлен ни в одной из двух других. Мо-

дель эмбеддера представляла из себя 4-слойную полносвязную нейронную сеть [5].

Обученная модель затем была применена в реальных условиях и показала качество ниже ожидаемого. Вероятнее всего, неустойчивые результаты связаны с различием обучающего и реального множеств исходных кодов, а также с особенностями последнего, такими как использование шаблонов для решений. В этой связи, для реального использования обученной модели были применены эвристические правила, ограничивающие область применения модели. В частности, предварительно (перед применением стилистической модели) было реализовано построчное сравнение исходных кодов после удаления комментариев и пробельных символов.

# 2.3. Рисунки, формулы и таблицы

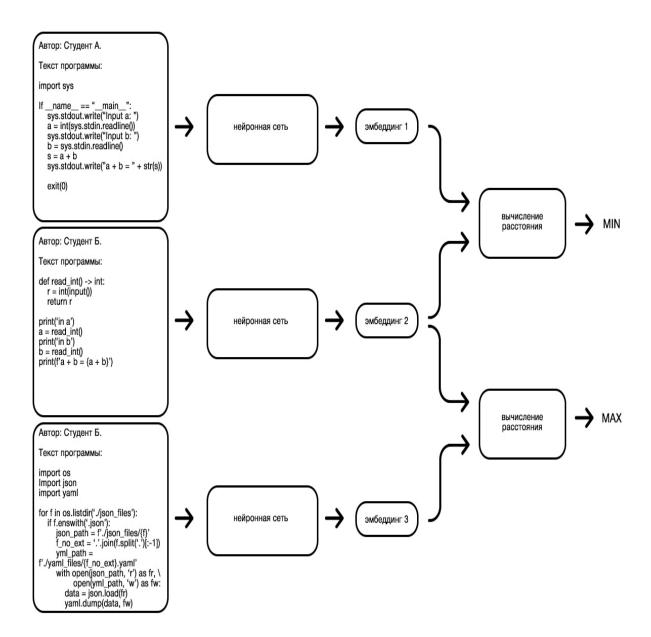


Рис. 1 Процесс обучения модели

triplet 
$$loss(a, p, n) = d(a, p) - d(a, n) \rightarrow min$$
 (1)

$$P$$
(решение  $a$  заимствовано у автора  $B$ ) =  $max(1 - \frac{min_b(d(a,b))}{\sigma^2},0)$  (2)

#### 3. Заключение

В данной работе был исследован ряд возможных решений для проверки программных кодов на заимствование, изучен накопленный опыт построения подобных систем, выбран оптимальный способ реализации, создающий минимальные технические издержки. Выбранный способ реализации включает построение глубокой нейросетевой модели. Для обучения последней был собран обучающий набор данных, содержащий программные коды из открытых источников.

После обучения полученная модель была протестирована на предмет применимости в реальных условиях. Различия в реальной и

тренировочной выборке, однако, снизили качество предсказания. Для улучшения качества рамки применения стилистической модели были искусственно ограничены введением простых для вычисления и разумных эвристик.

Дальнейшие исследования по усовершенствованию модели предсказания заимствования включают в себя анализ более интеллектуальных способ построения векторов исходных текстов. В частности, перспективными выглядят способы построения векторов с использованием Transformer моделей [8], а также моделей на базе токенизации и построения отпечатков кода [9].

Работа выполнена при поддержке гранта РФФИ № 19-29-14057.

# Research and Development of Machine Learning Methods and Neural Network Architectures for Usage in the Field of Verification

# Michael Dyachenko, Alexander Leonov, Maxim Matushin

Abstract. Global trends in the digitalization of education require the transfer of a significant share of the educational process to digital educational environments with automatic verification of the tasks completed by students. The downside of automatic verification is that students use other people's solutions in their answers, passing them off as their own. This article discusses the problem of automatic and semi-automatic detection of borrowings on the example of passing solutions in programming workshops. As an approach to the study, the path of constructing predictive models of the probability of borrowing was chosen. We studied the applicability of deep machine learning methods to solving the problem of detecting borrowing, and also considered a number of heuristic techniques designed to improve the accuracy and expand the scope of using the constructed deep models.

**Keywords:** anti-plagiarism, machine learning, deep learning, neural networks, program text recognition, borrowing detection

# Литература

- 1. Kimberly Redmond Lannan Luo Qiang Zeng. A Cross-Architecture Instruction Embedding Model for Natural Language Processing-Inspired Binary Code Analysis. https://arxiv.org/pdf/1812.09652.pdf.
- 2. Okutan Ahmet. Use of Source Code Similarity Metrics in Software Defect Prediction. https://arxiv.org/pdf/1808.10033.pdf.
- 3. Погребной В.К. Задача определения оценок сходства структур двух графов на основе выделения общих частей. // Известия Томского политехнического университета. Т. 322. №5, 2013.
- 4. Rose Yu Huida Qiu Zhen Wen Ching-Yung Lin. Current and Future Challenges in Mining Large Networks: Report on the Second SDM Workshop on Mining Networks and Graphs. 2016. https://www.semanticscholar.org/paper/Current-and-Future-Challenges-in-Mining-Large-
- 5. ROSENBLATT F. THE PERCEPTRON: A PROBABILISTIC MODEL FOR INFORMATION STORAGE AND ORGANIZATION IN THE BRAIN. // Psychological Review Vol. 65, No. 6, 1958.
- 6. Schultz M.; Joachims T. Learning a distance metric from relative comparisons. // Advances in Neural Information Processing Systems. 16: 41-48, 2004.
- 7. Joshua Robinson Ching-Yao Chuang Suvrit Sra-Stefanie Jegelka. Contrastive Learning with Hard Negative Samples. 2020. https://arxiv.org/abs/2010.04592.

- 8. Ashish Vaswani Noam Shazeer Niki Parmar Jakob Uszkoreit Llion Jones Aidan N. Gomez Lukasz Kaiser Illia Polosukhin. Attention Is All You Need. 2017. https://arxiv.org/abs/1706.03762.
  - 9. И. А. Посов В. Е. Допира. Методы поиска плагиата в кодах программ. // Известия СПбГЭТУ 10. <ЛЭТИ> №6, 2019.

# Элементы искусственного интеллекта при автоматизации тестирования веб-интерфейсов на примере ЦОП Мирера

М.С.Дьяченко<sup>1</sup>, А.Г. Леонов<sup>2</sup>, Д.А. Сокунов<sup>3</sup>

 $^1$  ФГУ ФНЦ НИИСИ РАН, Москва, Россия, МГУ им. М.В.Ломоносова, Москва, Россия, МПГУ, Москва, Россия, Государственный университет управления, Москва, Россия, dr.l@vip.niisi.ru;  $^2$  ФГУ ФНЦ НИИСИ РАН, Москва, Россия, mdyachenko@niisi.ru

<sup>3</sup> ФГУ ФНЦ НИИСИ РАН, Москва, Россия, МГУ им. М.В.Ломоносова, Москва, Россия, sokunov@ niisi.ru;

Аннотация. Тестирование программного обеспечения является одним из трудозатратных, времяёмких и дорогостоящих этапов разработки практически любого цифрового продукта. Поиск и выявление возможных и имеющихся уязвимостей в системе, их анализ, разработка качественных тестовых сценариев для полного покрытия системы тестами, а также многократное повторение однотипных процедур в рамках тестовых циклов требуют большого объема человеческих ресурсов и времени. Использование технологий искусственного интеллекта и интеллектуальных систем выглядит достаточно перспективным решением для данных проблем. Ценность искусственного интеллекта заключается в сокращении прямого участия разработчика и(или) тестировщика в задачах, не поддающихся автоматизации традиционными средствами. В данной статье проводится анализ некоторых основных методов применения искусственного интеллекта в автоматизированном тестировании веб-интерфейсов, а также обзор существующего на сегодняшний день программного обеспечения, способного реализовывать данные методы. В качестве практического приложения рассмотрены вопросы тестирования веб-интерфейса цифровой образовательной платформы Мирера.

**Ключевые слова:** Веб-интерфейс, web-API, искусственный интеллект, интеллектуальная система, автоматизированное тестирование, функциональное тестирование, ЦОП Мирера

# 1. Введение

Процессы цифровизация экономики, протекающие сейчас повсеместно в мире, требуют, в том числе, трансформации всего образовательного процесса, использование смешанной формы обучения, когда самостоятельная работа слушателя в цифровой образовательной среде (ЦОС), ориентированной на изучаемый предмет, поддержана непрерывным контактом с преподавателем, при котором образовательный процесс фактически становится непрерываемым единым потоком освоения желаемой компетенции. Возможности, предоставляемые различными ЦОС для получения адекватной оценки достижения учащихся объединяются в единые цифровые образовательную платформы (ЦОП), например, Мирера[1], курсы в которой, изначально были ориентированы на освоение программирования, но, в последствии, расширены на предметы естественно-научного и гуманитарного содержания.

При этом тестирование подобного программного продукта, как и любого другого цифрового продукта, обязательная часть цикла разработки, которая необходима для достижения высокого уровня качества выпускаемых про-

грамм. Практика показывает, что чем позже будет замечена ошибка и чем серьезнее она будет, тем больше временных, человеческих и материальных ресурсов потребуется на её устранение. Поэтому применение технологий тестирования начинается как можно раньше, часто даже на этапе описания концепции и формирования требований к системе [2][3].

Автоматизированное тестирование программного обеспечения — существенная, и потенциально, самая крупная часть процесса тестирования на этапе контроля качества в процессе разработки программного обеспечения, при котором используют программные средства и системы для проведения тестов и проверки результатов выполнения, что помогает сократить время тестирования и упростить его процесс.

### 2. Статический анализ

На «нулевой» стадии тестирования, при поиске ошибок разработчиком в собственном программном коде, ручной поиск может стать довольно долгим и трудозатратным процессом. Это касается как мелких ошибок, например, приводящих к обращению/модификации памяти за границами выделенного блока памяти, так и более существенных возникающих, например, при вычислении квадратного корня из отрицательного числа, деление на ноль, softlock, то есть состояние программы при котором она продолжает выполняться, однако перестаёт функционировать должным образом и не может быть завершена корректно (пример - зацикливается). Такие ошибки могут проявляться лишь на особых наборах тестовых данных.

При поиске подобных проблем часто используют статический анализ кода программы, производимый без реального исполнения исследуемых программ. Термин обычно применяют к анализу, производимому при помощи специальных программ, тогда как ручной анализ называют «program understanding», «program comprehension», что можно перевести как «понимание кода».

Данный особый вид тестирующего программного обеспечения способен автоматически отслеживать опечатки и находить потенциально слабые места. Также подобные инструменты можно настроить и на проверку стиля написания кода — размера табуляции, способа именования переменных, расстановки скобок границ блоков кода и прочие элементы форматирования исходного кода, облегчающие чтение (сопровождение) кода. Это позволит группе разработчиков в рамках одного проекта придерживаться единого стиля кодирования (и форматирования текста программы), сделав весь код более однородным и читаемым.

Одним из основных преимуществ использования статического анализа в тестировании гибкость и относительная простота настройки параметров анализа. Это позволяет варьировать степень "строгости" проверки кода, позволяя настроить и подключить инструменты статического анализа в любых проектах. Другим немаловажным достоинством систем статического анализа является возможность полной автоматизации. Данный инструмент можно «запрограммировать» как на обычное составление отчётов с найденными недочетами, так и на автоматическое редактирование проверяемых файлов, позволяя дополнять и изменять исходный код, приводя его к единому стилю форматирования, попутно исправляя опечатки.

Формально данный этап не относится к сфере тестирования ПО, однако применение продвинутых алгоритмов статического поиска ошибок поможет заметно ускорить общий процесс цикла разработки.

В качестве инструмента статического анализа в разработке ЦОП Мирера предлагается использовать ESLint [4]. ESLint работает с кодом, написанным на языке программирования JavaScript, и разработан так, что все правила и настройки проверки работают и подключаются,

как отдельные плагины. По умолчанию в нём уже имеются базовые настройки и элементы искусственного интеллекта (ИИ). Благодаря открытому исходному коду ESLint и поддержке динамического подключения пользовательских правил данный инструмент можно настраивать и совершенствовать под любые проекты, позволяя добиться более интеллектуальных и эффективных проверок. Также ESLint может самостоятельно проводить редактирование файлов согласно всем обнаруженным замечаниям. Данный фреймворк написан с использованием Node.js, что позволит проще интегрировать его в цикл разработки.

Помимо ESLint, можно также упомянуть JSHint [5] и JSLint [6]. Данные инструменты похожи на ESLint, однако они имеют более ограниченный функционал и меньшую гибкость для настроек по сравнению с ним.

# 3. Написание и запуск автотестов

Говоря об автоматизации процесса тестирования, естественным будет обратиться к инструментам для написания и выполнения автотестов. Если ручное регрессионное тестирование — довольно однообразное, отнимающее большое количество ресурсов, занятие, то, напротив, его автоматизация путём составления автотестов способна заметно ускорить весь процесс тестирования и снизить количество допускаемых человеком-тестировщиком ошибок, также увеличить объем выполняемых тестов. Тестировщику достаточно будет один раз записать скрипт (код алгоритма) проведения теста и затем запускать его в любое время по желанию, перекладывая всю рутину на компьютер [7].

В качестве такого инструмента авторы остановились на Cypress [8]. Это набор инструментов с открытым исходным кодом, предназначенный для функционального тестирования web-API. Он предоставляет достаточно мощный инструментарий для отладки и контроля выполнения автотестов. По ходу проведения теста Cypress журналирует все выполняемые в процессе тестирования действия и посылаемые или получаемые запросы API, а каждый шаг сопровождается созданием снимков страницы - до и после выполнения шага. Cypress поддерживает самые распространенные веб-браузеры - Google Chrome, Opera, Mozilla Firefox, Microsoft Edge, Internet Explorer и прочие.

Благодаря открытому исходному коду, Сургез можно интегрировать во множество интеллектуальных систем и инструментов, а собираемая им информация может стать основой для дальнейшего анализа и обучения этих систем.

# 4. Визуальное тестирование веб-страниц

Большинство фреймворков, выполняющих и контролирующих проведение автотестов, проверяют корректность выполнения тестов на уровне бэк-энда: наличие или отсутствие на странице каких-либо элементов, их свойства (количество, текст внутри, цвет и т.д.). При разработке веб-интерфейсов важной составляющей процесса тестирования является проверка того, что при взаимодействии с системой не нарушается строение и визуальная составляющая вебстраниц: каждый элемент стоит на своем месте, ничего не исчезло, никуда не перемещено и не выходит за границы областей отображения. Задача такого рода тестирования не является тривиальной, так как требует огромного количества времени на проверку отображения визуального контента для всех использумых платформ, браузеров и разрешений экрана.

Перспективным решением данной проблемы выглядит использование специализированных интеллектуальных систем, способных отслеживать отображение веб-страниц в реальном времени во время проведения функционального тестирования. В этом случае, данные инструменты можно настроить на более детальное отслеживание свойств элементов страницы и реагирование на их аномальные изменения (например, линейные размеры). Другой идеей может являеться сбор снимков экрана с дальнейшем обучением тестирующей системы визуальному анализу полученных снимков, выявляя по находящимся на них отличиях визуальные аномалии.

Фреймворк Applitools [9] по мнению авторов является одним из лучших в своем классе для проведения визуального тестирования. Applitools обеспечивает визуальное управление приложениями, а также тестирование и мониторинг пользовательского интерфейса при помощи встроенного искусственного интеллекта Visual AI. В отличие от традиционных инструментов функционального тестирования, которые требуют массы строк кода для проверки макета, Applitools проверяет весь экран всего одним вызовом АРІ. Из явных достоинств данного инструмента можно выделить широкий список сторонних программ и сервисов, которые поддерживают Applitools и в которые его легко интегрировать.

Данный фреймворк является довольно интересным, однако ввиду политики коммерческого распространения и отсутствия доступа к исходному коду, подробная информация об алгоритмах и принципах работы Visual AI в открытых источниках отсутствует.

# 5. Оптимизация запусков больших объемов автотестов

С ростом числа автотестов увеличивается и время, которое необходимо потратить на их, пусть и автоматизированное, выполнение. На финальных стадиях развития проекта полный прогон всех тестов может занимать непозволительно много времени, что ставит вопрос об ускорении процесса тестирования.

Одним из способов добиться этого является оптимизация последовательности выполнения автотестов. Выделяя общие фрагменты, можно группировать тесты таким образом, чтобы при запуске добиться минимизации количества уже протестированных повторений действий и тем самым добиться заметного ускорения. Анализ того, какие тесты стоит группировать между собой и каким именно образом, является сложной проблемой, решение которой предлагается проводить с использованием систем с искусственным интеллектом.

К таким готовым решением можно отнести Testim Automate [10], который позиционируется создателями как аналог Cypress с некоторыми важными изменениями и доработками. Главной отличительной особенностью Testim является наличие встроенного искусственного интеллекта. ИИ анализирует запускаемый набор автотестов, выделяя повторяющиеся блоки действий, самостоятельно перестраивает алгоритм выполнения данных автотестов (при этом не редактируя сами скрипты), позволяя заметно уменьшить время их выполнения. Testim Automate также помогает в написании автотестов, например, позволяя пользователю составлять алгоритм автотеста, используя готовые опции из списка. Данные действия в GUI Testim Automate самостоятельно преобразует в код, ускоряя таким образом ещё и процесс написания автотестов. При этом Testim Automate лишён некоторых недостатков, присущих Cypress, например, позволяет одновременно поддерживать нескольких ресурсов или вкладок в одном тесте. К недостаткам можно отнести то, что, как и Cypress, Testim Automate является проприетарным программным обеспечением.

# 6. Автоматический анализ статистики запусков и сегрегация автотестов

Еще одним способом ускорить процесс регрессионного тестирования является выбор из всего множества автотестов определённого подмножества, состоящего лишь из тех автотестов, которые с большей вероятностью найдут новые

ошибки. В таком случае возникает вопрос, какие тесты стоит запускать в первую очередь, а какие можно отложить?

Одним из методов отбора автотестов для запуска является анализ изменений в новой версии цифрового продукта. Используя системы отслеживания версий и изменений в коде программ (например, GitHub, GitLab SourceTree), можно явно идентифицировать участки кода, содержащие изменения. Различным областям программы можно сопоставить свои наборы автотестов, то есть группы автотестов, обязательных для повторной проверки. Автоматическая сегрегация и выборка таких наборов по установленным флагам и правилам является не сложной задачей для систем искусственного интеллекта, позволяя и сэкономить время на отбор автотестов, и оптимизировав очередную итерацию регрессионного тестирования.

Другим вариантом является отбор автотестов на основе статистики их запусков и количества и частоты обнаруживаемых ошибок [11][12]. Ведение статистики всех проведенных тестов, а также логирование всех выявленных ошибок является важной составляющей в анализе качества продукта. Однако ручной учёт всех запусков и результатов трудоемким процессом, который, тем не менее, можно автоматизировать. Интегрировав систему сбора статистики в приложение запуска автотестов, позволит без участия тестировщика сортировать тесты по категориям вместе с результатами запусков.

Удобным инструментом ведения статистики является Allure [13]. Это фреймворк с открытым исходным кодом, разработанный компанией Яндекс в 2014 году и предназначенный для автоматической генерации и сортировки отчетов о запущенных автотестах. Allure основан на стандарте вывода результатов xUnit. Любой отчет создается в два этапа. Во время выполнения теста (первый шаг) небольшая библиотека, подключенная к платформе тестирования, сохраняет информацию о выполненных тестах в файлы XML. Во время создания отчета (второй шаг) файлы XML преобразуются в отчет HTML-формата. Данный инструмент имеет модульную структуру, позволяя легче интегрировать её в проект, а также совместим со многими другими фреймворками, работающими с С#, Python, Java, JavaScript, Ruby, PHP и прочими языками программирования, а также с Jira, GitLab и GitHub [14]. В качестве инструмента ведения статистики автотестов ЦОП Мирера также использовалась Allure.

Allure способен автоматически собирать всю необходимую информацию, поступающую от других инструментов тестирования, например,

журналов запусков автотестов из Cypress, и составлять на её основе статистику обнаруживаемых ошибок. При этом по-прежнему стоит задача автоматизированного анализа собранной статистики. Так, анализируя полученные данные, с использованием систем искусственного интеллекта возможно в автоматическом режиме определять, какие участки программы являются наиболее или наименее уязвимыми и, соответственно, выстраивать приоритеты тестам для дальнейших запусков. Построение такой модели и обучение её анализу кода программы и статистики запусков автотестов является, пожалуй, самой сложной проблемой из всех перечисленных в данной статье, требующей отдельного изучения.

Готовых практических решений, удовлетворяющих задачам авторов в области анализа статистики или изменений кода в новых версиях тестируемого программного обеспечения (с использованием элементов искусственного интеллекта), на момент написания статьи не обнаружено.

# 7. Выводы

Изучив (в том числе на практике) основные фреймворки, способные помочь в достижении высокого уровня автоматизации тестирования, авторы предлагают список задач и методик развития инструментария автоматизированного тестирования и интеграции его в цифровые продукты на примере ЦОП Мирера:

- 1. Использовать ESLint в качестве базового фреймворка для статического анализа кода. Разработать собственную систему формирования стиля написания кода, а также механизм обнаружения некоторых видов уязвимостей.
- 2. Использовать Cypress в качестве основного фреймворка для написания и запуска автотестов. Интегрировать в его среду разработки усовершенствованный анализатор кода на базе ESLint.
- 3. Использовать Allure для сбора статистики запусков автотестов. Интегрировать его в Сургез для автоматического сбора результатов, а также в Jira для более удобного контроля тестов всеми разработчиками.

Фреймворки Testim Automate и Applitools под сформулированные задачи не подходят ввиду своей политики коммерческого распространения и отсутствия открытого исходного кола.

# 8. Заключение

В данной статье был приведён обзор некоторых направлений использования искусственного интеллекта в автоматизированном тестировании на примере ЦОП Мирера. Был выделен список фреймворков, на основе которых можно проводить поиск ошибок кода, написание и запуск автотестов, а также проводить автоматический сбор статистики. Проанализировав доступные ореп source решения, авторам стало очевидно, что готовых решений по анализу требуемой информации для выявления ошибок и выделению множества автотестов на запуск искусственным интеллектом на данный момент нет.

Решение данной проблемы будет одним из направлений развития авторами систем тестирования веб-интерфейсов.

Исследование выполнено в рамках государственного задания по проведению фундаментальных исследований по теме «Разработка, реализация и внедрение семейства интегрированных многоязыковых сред программирования с автоматизированной проверкой заданий для учащихся образовательных организаций, ДОО, младшей, основной и старшей школы и студентов педагогических университетов.» (№ 0580-2021-0010).

# Elements of Artificial Intelligence in the Automation of Testing Web Interfaces on the Example of the DEP Mirera

### Michael Dyachenko, Alexander Leonov, Dmitry Sokunov

Abstract. Software testing is one of the time-consuming and expensive stages of the development of almost any digital product. The search and identification of possible and existing vulnerabilities in the system, their analysis, the development of high-quality test scenarios to fully cover the system with tests, as well as the repeated repetition of similar procedures within test cycles require a large amount of human resources and time. The use of artificial intelligence technologies and intelligent systems looks like a promising enough solution for these problems. The value of artificial intelligence is to reduce the direct involvement of the developer and/or tester in tasks that cannot be automated by traditional means. This article analyzes some of the main methods of using artificial intelligence in automated testing of web interfaces, as well as an overview of the software currently available that can implement these methods. As a practical application, the issues of testing the web interface of the digital educational platform Mirera are considered.

**Keywords:** Web interface, web API, artificial intelligence, intelligent system, automated testing, functional testing, DEP Mirera

# Литература

- 1. Платформа Мирера [Электронный ресурс] URL: https://www.mirera.ru (дата обращения: 07.09.2021)
  - 2. C. Kaner, J. Falk, H.Q. Nguyen. Testing Computer Software 1999.
  - 3. E. Dusting, J. Rashka, J. Paul. Automated Software Testing 1999.
  - 4. ESLint [Электронный ресурс] URL: https://eslint.org/ (дата обращения: 07.09.2021)
  - JSHint [Электронный ресурс] URL: https://jshint.com/docs/ (дата обращения: 07.09.2021)
  - 6. JSLint [Электронный ресурс] URL: https://jslint.com/help.html (дата обращения: 07.09.2021)
- 7. "Integrating Artificial Intelligence in Software Testing Benefits and Strategy" [Электронный ресурс] URL: https://medium.com/xenonstack-ai/integrating-artificial-intelligence-in-software-testing-benefits-and-strategy-b22deaa9cf31 (дата обращения: 07.09.2021)
  - 8. Cypress [Электронный ресурс] URL: https://www.cypress.io/ (дата обращения: 07.09.2021)
  - 9. Applitools [Электронный ресурс] URL: https://applitools.com/ (дата обращения: 07.09.2021)
- 10. Testim Automate [Электронный ресурс] URL: https://www.testim.io/ (дата обращения: 27.09.2021)
  - 11. A. Martin-Lopes. AI-Driven WEB API Testing 2019.
  - 12. J. Gao, C. Tao. What is AI Software Testing? and Why 2019.
  - 13. Allure [Электронный ресурс] URL: http://allure.qatools.ru/ (дата обращения: 07.09.2021)
- 14. "Allure фреймворк от Яндекса для создания простых и понятных отчётов автотестов" [Электронный ресурс] URL: https://habr.com/ru/company/yandex/blog/232697/ (дата обращения: 07.09.2021)

# Методы интеграции цифровых образовательных сред в цифровую образовательную платформу Мирера

# А.Г. Леонов<sup>1</sup>, А.Е. Орловский<sup>2</sup>

<sup>1</sup>ФГУ ФНЦ НИИСИ РАН, Москва, Россия, МГУ им. М.В.Ломоносова, Москва, Россия, МПГУ, Москва, Россия, Государственный университет управления, Москва, Россия, dr.l@vip.niisi.ru; <sup>2</sup>ФГУ ФНЦ НИИСИ РАН, Москва, Россия, МГУ им. М.В.Ломоносова, Москва, Россия, orlovskiy@vip.niisi.ru;

Аннотация. Цифровая экономика ставит широкий круг задач перед производством, наукой, образованием и другими областями деятельности человека. Особое внимание уделяется цифровой трансформации образования, при которой происходят глубинные преобразования учебного процесса, внедряются гибридные формы обучения, при которой упор делается на использование предметных цифровых образовательных сред. Цифровое управление университетом не может полностью интегрировать цифровые образовательные среды в образовательный процесс. Для этого требуется создавать отдельные образовательные платформы, которые включают межпредметные связи различных цифровых образовательных сред в единый образовательный процесс.

В статье рассматриваются вопросы и методы интеграции различных цифровых образовательных сред в образовательную платформу Мирера с сохранением и возможным расширением функционала, доступного для студентов и преподавателей. Приводятся и разбираются примеры подключения образовательных систем с их особенностями и функциональными возможностями.

**Ключевые слова:** цифровая образовательная платформа, цифровая образовательная среда, Мирера, ЭВМ-практикум, КуМир, ПиктоМир

# 1. Введение

Цифровизация образования сейчас является одним из доминирующих процессов в цифровой экономике. Создание авторских цифровых образовательных платформ, подключение всевозможных программно-аппаратных модулей для формирования обстановки обучения, близкой к реальному общению с преподавателем, интеграция видеоконференцсвязи, подготовка собственных курсов и электронных пособий к ним, оформление заданий, автоматизация проверки решений — все это только малая часть функционала идеальной системы для педагогического программного обеспечения.

Особенно остро проблема отсутствия технологичных и готовых к серьезной нагрузке систем цифрового образования проявилась во время пандемии COVID-19. Методы борьбы с инфекцией предполагают социальное дистанцирование, что переносит образовательный процесс любого уровня в дистанционную форму.

Большинство платформ массовых открытых онлайн курсов (МООК), предоставляющих услуги организации курсов преподавателям, не ставят своей целью систематическое освоение программы даже по одному предмету [1]. Зачастую данные системы предлагают короткие

курсы, состоящие из некоторой выжимки материала по теме и из предложенных задач к решению. Взаимодействие преподавателя со студентом при этом или отсутствует, или сильно ограничено, когда преподаватель выступает в роли автора курса и никак не связан с непосредственным обучением слушателей.

Выходом из данной ситуации является использование так называемой гибридной формы обучения, когда самостоятельная работа студента в цифровой образовательной среде, ориентированной на изучаемый предмет, поддержана непрерывным контактом с преподавателем, при котором образовательный процесс фактически становится непрерывным потоком освоения желаемой компетенции [2].

В этом случае процесс обучения максимально похож на аудиторное занятие: перед глазами обучаемого преподаватель ведет рассказ о конкретной теме своего предмета, демонстрирует связанные с темой занятия материалы, использует виртуальную доску, отвечает на возникшие вопросы в реальном времени [3]. А поскольку в распоряжении учителя и ученика весь функционал цифровой среды, всегда есть доступ к набору инструментов, который отсутствует и невозможно обеспечить в режиме классического аудиторного образования.

# 2. ЦОП Мирера

Цифровая образовательная платформа (ЦОП) Мирера предоставляет возможность создания и ведения курсов по естественно-научным специальностям, в частности по Программированию, Информатике и ИКТ, Робототехнике, Теории формальных грамматик и так далее, а также по Педагогике, Психологии, Истории, Налоговому праву и другим предметам, используя различные цифровые образовательные среды (ПиктоМир, КуМир, ЭВМ-практикум), системы программирования с открытым кодом, и публикации образовательных материалов в виде онлайн-курсов [4][5][6].

Одной из основных целей при разработке компонентов для данной платформы является создание простого, удобного и технологичного интерфейса для максимально продуктивного сотрудничества преподавателя со студентом. Промежуточные результаты использования ЦОП Мирера говорят о том, что методически продуманное наполнение образовательным контентом позволяет обучаемым за семестр самостоятельно решать до 500 и более заданий разного рода и уровня сложности.

Стоит отметить, что продуманный интерфейс далеко не единственное преимущество Миреры: весь доступный функционал направлен на облегчение образовательного процесса путем автоматизации каждого этапа. Автоматизация включает в себя следующие функциональные элементы: сдачу и верификацию задач, проверку решений на плагиат, подключение сторонних систем, рассылку уведомлений и новостей, прохождение рубежных контролей и зачетов, управление сложностью заданий (индивидуальные образовательные траектории), обеспечение обратной связи, а также подключение к видеоконференциям и другие особенности платформы.

# 3. Интеграция

Разработка образовательных компонентов не должна ограничиваться независимым, отдельно существующим набором инструментов. Важная часть платформы — интеграция других систем и сервисов. При этом процесс интеграции одной системы в другую необязательно связан с односторонним расширением функциональных возможностей системы. Интеграция позволяет добавить функционал в обе системы, при этом сохранив целостность и функциональность каждой цифровой образовательной среды.

#### 3.1 ЭВМ-практикум

Одним из примеров описываемого процесса является интеграция программной системы

ЭВМ-практикум в ЦОП Мирера [7] [8]. ЭВМпрактикум – это цифровая образовательная среда, в которой в качестве объектов выступают оперативная память и регистры центрального процессора некой виртуальной ЭВМ (модифицированный и сокращенный вариант Intel x86). С помощью ЦОС ЭВМ-практикум можно не только непосредственно редактировать эти объекты, но и создавать программу на ассемблере этой ЭВМ, выполнять ее непрерывно или по шагам и наблюдать, как меняются объекты - память, регистры, стек и пр. Используя ЦОС ЭВМпрактикум можно организовывать курсы по устройству процессора продолжительностью, например, 18 академических часов, при этом обучаемый научится составлять простейшие программы на ассемблере (в кодах) ЭВМ по вычислению индуктивных функций, больших чисел, обработке матриц и, тем самым, освоит общие принципы работы и устройство процессоров современных ЭВМ.

Таким образом, с точки зрения ЦОП Мирера ЦОС ЭВМ-практикум предоставляет интерфейс для написания программного кода, загрузки тестирующего файла и проверки решения на загруженных тестах.

Идеи интеграции и ее непосредственная реализация заключаются в следующем: ЦОП Мирера позволяет создавать задачи, выбрав в качестве внешней тестирующей системы ЭВМ-практикум, который в свою очередь может запрашивать задачи нужного типа из Миреры и использовать полученные данные. Достаточно авторизоваться в ЦОП Мирера, и пользователь моментально получает доступ ко всем доступным ему в курсе задачам.

Для обеспечения возможности использования ресурса на любой платформе, а также для реализации веб-версии ЭВМ-практикум использует средства Qt [9].

Для взаимодействия с АРІ ЦОП Мирера, в частности, получить список задач, доступных пользователю, необходимо провести процедуру авторизации, используя АРІ и веб-интерфейс сервиса аутентификации пользователей Инфомир (используются модули Qt WebEngine и Qt Network [10]). Перед тем, как пользователь ЦОС ЭВМ-практикум начинает взаимодействие с ЦОП Мирера, появляется диалоговое окно, перенаправляющее на страницу входа. После успешного ввода данных, запускается стандартный протокол авторизации OAuth 2.0, позволяющий сервису Инфомир верифицировать пользователя и направить в ответном запросе ключ доступа access\_token. ЦОС ЭВМ-практикум использует в дальнейших обращениях к ЦОП Мирера данный ключ, чтобы запросы проходили аутентификацию и возвращали нужный результат.



Рис. 1 Схема OAuth 2.0

Описанный механизм реализован с помощью Qt классов QDialog и QWebEngineView, которые осуществляют отображение страницы входа Инфомир, а также QwebEngineUrlScheme-Handler и QNetworkAccessManager, с помощью которых происходит передача, получение и обработка верификационных данных.

ОАuth 2.0 авторизация подразумевает использование специального кода authorization\_code, который в случае успешного входа в конечном итоге обменивается на access\_token. Также передаются параметры, включающие в себя адрес перенаправления после авторизации и специальный идентификатор для предотвращения CSRF-атак. Данные параметры позволяют проверить подлинность идентификатора при ответном запросе из системы Инфомир, чтобы оценить соответствие пользователя-отправителя и пользователя-адресата. Генерация идентифика-тора и параметров запроса происходит в классе AuthDialog.

Для перехвата и обработки перенаправляющего запроса со сгенерированными ключами доступа используется класс AuthSchemeHandler. Предва-рительно необходимо зарегистрировать пользовательскую схему URL, чтобы получить данные после перенаправления запроса.

```
void AuthSchemeHandler::re-
questStarted(QWebEngineUrlRequestJob
*job)
  QUrl\ requestUrl = job->requestUrl();
  QUrlQuery\ query = QUrlQuery(reques-
tUrl):
  QString\ stateParsed = query.query-
ItemValue("state");
  QString\ code = query.query-
ItemValue("code");
  OSettings settings;
  QJsonObject body;
  body.insert("client_id", set-
tings.value("auth/client").toString());
  body.insert("code_verifier",
QString(codeVerifier));
  body.insert("code", code);
  body.insert("redirect_uri", "ewm");
  RequestHandler* requestHandler = EW-
MApp->getRequestHandler();
  int loginStatus = requestHandler
->login(QJsonDocument(body).toJson());
```

Класс RequestHandler осуществляет непосредственное отправление запросов авторизации и последующих запросов, обращенных к АРІ ЦОП Мирера. В последних добавляется специальный заголовочный параметр, содержащий ключ access\_token.

В результате такого взаимодействия пользователь ЭВМ-практикума может продолжать его

использовать в привычном варианте, при этом отправляя свои решения в ЦОП Мирера напрямую из него, а ЦОП Мирера будет получать информацию о попытках решений пользователя и предоставлять статистику внутреннему пользователю в полном объеме.

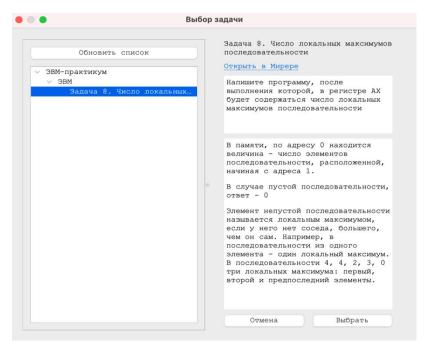


Рис. 2 ПОС ЭВМ-плактикум: выбол залачи

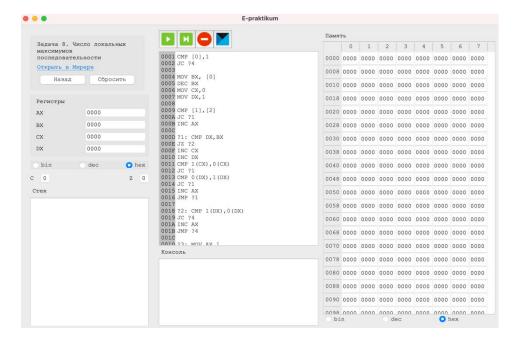


Рис. 3 ЦОС ЭВМ-практикум: отправление решения

# 3.2 КуМир

Под интеграцией сторонних сервисов подразумевается не только взаимодействие двух систем по определенному протоколу с оптимальным обменом сообщений, но и непосредственное внедрение внешней тестирующей системы в ЦОП Мирера. Примером данного взаимодействия может служить система программирования КуМир [11].

Цель данной интеграции – обеспечить функциональную возможность создавать, редактировать и запускать задачи на школьном алгоритмическом языке, прямо находясь на платформе Мирера, проверять решение, используя компилятор, который является частью ЦОС КуМир. В

результате пользователь может не переключаться между различным программным обеспечением, а осваивать различные предметы в знакомом интерфейсе единой универсальной цифровой образовательной платформы.

В ЦОП Мирера используется встроенный, свободно распространяемый редактор Асе Editor для создания и редактирования решений, в том числе и кода программ, в ЦОП Мирера. Этот редактор, в том числе, позволяет создавать и использовать собственные стили текста, подсветку и конфигурации препроцессора для максимального погружения в привычную среду разработки [12].

```
Кумир
  1
    алг лог четырехугольник (цел a,b,c,d)
  2
        дано І а,b,c,d положительны
  3
         надо І выяснить существует ли четырехугольник
              I с данными длинами сторон
  5
  6
        знач:=(a+b+c+d)>2*imax4(a,b,c,d)
  7
    кон
  8
    алг цел imax4(цел a,b,c,d)
  9
        надо I знач = максимум из чисел a,b,c,d
 10
    нач
 11
        знач:= imax(imax(a,b), imax(c,d))
 12
    кон
 13
```

Рис. 4 КуМир в ЦОП Мирера

Асе Editor предоставляет средства для достаточно гибкого создания правил подсветки и набора управляющих выражений, определяющих синтаксис языка написания, в том числе, естественно и языка программирования. При внедрении созданы поведенческие шаблоны, структуры, описывающие грамматику и механизмы, присущие школьному алгоритмическому языку (КуМир). Также определены ключевые слова, типы данных и команды, доступные для взаимодействия с языком программирования. Более того, создан специальный файл, описывающий CSS стиль для каждой конструкции

языка.

Данные правила получили названия Kumir-Mode, KumirHighlightRules, KumirTheme и дополнили библиотеку с аналогичными структурами для других языков программирования.

При этом Ace Editor автоматически использует описанные «регулировщики» синтаксиса и подсветки, когда слушатель находится в редакторе программы, написанной на школьном алгоритмическом языке (КуМир). Также автоматически применяется тема, отвечающая за стиль языка.

```
ace.define(
    "ace/mode/kumir highlight rules".
   ["require", "exports", "ace/lib/oop", "ace/mode/text_highlight_rules"],
   function (acequire, exports, module) {
    "use strict":
   const oop = acequire("ace/lib/oop");
   const TextHighlightRules = acequire("ace/mode/text_highlight_rules").TextHighlightRules;
   const KumirHighlightRules = function () {
       const keywords = (
            "или|и|не|да|нет|" +
            "нач|кон|нц|кц|алг|" +
            "ввод|вывод|если|то|иначе|" +
            "при|пока|все|использовать|" +
            "для от до шаг"
       );
       const types = ("цел|вещ|лог|сим|лит|таб|целтаб|вещтаб|логтаб|симтаб|литтаб");
            "вправо влево вверх вниз сверху снизу слева справа " +
            "свободно|стена|закрасить|клетка|чистая|закрашена
       );
       const robots = ("Кузнечик|Водолей|Черепаха|Чертежник|Робот");
       let algorithms = ("");
       let keywordMap = {
            "keyword": keywords,
            "entity.name.type": types,
            "keyword.operator": commands,
            "support.class": robots
       };
```

Рис. 5 KumirHighlightRules

Интерфейс для подключения в ЦОП Мирера достаточно прозрачен и систематизирован, что позволяет связать практически любой сервис, следуя по пути полной интеграции или выбрав внедрение среды на платформу. Система КуМир на данный момент ограничивается вторым вариантом интеграции в ЦОП Миреру, однако взаимодействие двух систем с сохранением использования исходного программного обеспечения также возможно. Более того, создание и внедрение данного механизма находится в стадии разработки, что в скором времени приведет к использованию системы программирования КуМир с возможностью общения с ЦОП Мирера на базе текущего приложения КуМир.

#### 4. Заключение

В результате проделанной работы цифровая образовательная платформа Мирера предоставляет возможность использовать свои функциональные компоненты во внешних сервисах, а также внедрять сторонние системы непосредственно на платформу. Механизм интеграции с другими независимыми средами позволяет использовать их на платформе Мирера, делиться результатами образовательной деятельности и

наблюдать за статистикой обучения, находясь при этом в комфортной цифровой среде.

Исследование выполнено в рамках государственного задания по проведению фундаментальных исследований по теме «Разработка, реализация и внедрение семейства интегрированных многоязыковых сред программирования с автоматизированной проверкой заданий для учащихся образовательных организаций, ДОО, младшей, основной и старшей школы и студентов педагогических университетов.» (№ 0580-2021-0010).

# Methods of Integration of Digital Educational Environments into Digital Educational Platform Mirera

### Alexander Leonov, Anton Orlovskii

**Abstract.** The digital economy poses a wide range of tasks to product manufacturers, science, education and other areas of human activity. Special attention is paid to the digital transformation of education, in which deep transformations of the educational process, hybrid forms of education are introduced, in which emphasis is placed on the use of discipline-based digital educational environments. Digital university management cannot fully integrate digital educational environments into the educational process. To do this, it is necessary to create separate educational platforms that include interdisciplinary connections of various digital educational environments in a single educational process.

The article discusses the issues and methods of integrating various digital educational environments into the Mirera educational platform with the preservation and possible expansion of the functionality available to students and teachers. Examples of connecting educational systems with their features and functionality are given and analyzed.

**Keywords:** digital educational platform, digital educational environment, Mirera, E-praktikum, KuMir, PiktoMir

# Литература

- 1. A. McAuley, B. Stewart, G. Siemens, D. Cormier. The MOOC Model for Digital Practic
- 2. Андреева Н.В., Рождественская Л.В., Ярмахов Б.Б. Шаг школы в смешанное обучение. // Москва-2016
- John Estson. Blended Learning: The Evolution of Online and Face-to-Face Education from 2008-2015.
- 4. Платформа Мирера [Электронный ресурс] URL: https://www.mirera.ru (дата обращения: 01.09.2021)
- 5. Методическое пособие по системе ПиктоМир [Электронный ресурс] URL: https://www.niisi.ru/piktomir/m2019.pdf (дата обращения: 01.09.2021)
- 6. Кушниренко А.Г., Леонов А.Г., Ройтберг М.А. Знакомим дошкольников, младших школьников с азами алгоритмики с помощью систем ПиктоМир и КуМир // Труды НИИСИ РАН. 2015. Т. 5, № 1. С. 134-137.
- 7. Леонов А. Г., Прилипко А. А. Автоматизированный практикум по машинным языкам основа изучения языков программирования // Прорывные научные исследования как двигатель науки нового времени. НОУ САНКТ-ПЕТЕРБУРГСКИЙ ИНСТИТУТ ПРОЕКТНОГО МЕНЕДЖМЕНТА Санкт-Петербург, Россия, 2016. С. 223–230.
- 8. Система ЭВМ-практикум [Электронный ресурс] URL: http://e.kumir.su (дата обращения: 01.09.2021)
  - 9. Qt [Электронный ресурс] URL: https://qt.io/ (дата обращения: 01.09.2021)
- 10. Qt modules [Электронный ресурс] URL: https://doc.qt.io/qt-5/qtmodules.html (дата обращения: 01.09.2021)
- 11. Кушниренко А.Г., Леонов А.Г. КуМир учебное программное обеспечение базового курса информатики // Информатика в школе. 2017. № 8.
  - 12. Ace Editor [Электронный ресурс] URL: https://ace.c9.io/ (дата обращения: 01.09.2021)

Подписано в печать 28.10.2021 г. Формат 60x90/8 Печать цифровая. Печатных листов 8 Тираж 100 экз. Заказ № 884

Отпечатано в ППП «Типография «Наука» 121099, Москва, Шубинский пер., 6