

## МЕТОДЫ ГРАММАТИЧЕСКОЙ ЭВОЛЮЦИИ И СЕТЕВОГО ОПЕРАТОРА ДЛЯ СИНТЕЗА СИСТЕМЫ УПРАВЛЕНИЯ ДИНАМИЧЕСКИМ ОБЪЕКТОМ

Дивеев А.И.<sup>1</sup>, Казарян Д.Э.<sup>2</sup>

<sup>1</sup>Вычислительный центр им. А.А. Дородницына РАН, Москва, Россия (119333, г. Москва, ул. Вавилова, 40), e-mail: aidiveev@mail.ru

<sup>2</sup>Кафедра кибернетики и мехатроники, Российский университет дружбы народов, Москва, Россия (117923, г. Москва, ул. Орджоникидзе, 3), e-mail: kazaryan.david@gmail.com

---

Рассмотрена задача синтеза системы управления для нелинейного динамического объекта. Задача синтеза ставится как задача поиска управляющей функции от состояния объекта. Синтезированная система должна достигать заданной цели для некоторого множества начальных условий. Цель задана в виде нескольких функционалов качества. Для решения поставленной задачи используются методы грамматической эволюции и сетевого оператора. Грамматическая эволюция — подкласс генетического программирования, использующий для построения математического выражения систему продукционных правил. В методе сетевого оператора математическое выражение представлено графом. В качестве поискового алгоритма использовался стационарный генетический алгоритм. Выбор множества удовлетворительных управляющих функций осуществлялся построением множества Парето. Проведён вычислительный эксперимент, в результате которого каждым из рассматриваемых методов были получены управляющие функции для нелинейной пружины Дуффинга.

---

Ключевые слова: синтез системы управления, грамматическая эволюция, сетевой оператор, генетический алгоритм.

## GRAMMATICAL EVOLUTION AND NETWORK OPERATOR METHODS FOR SYNTHESIS OF THE CONTROL SYSTEM FOR A DYNAMIC OBJECT

Diveev A.I.<sup>1</sup>, Kazaryan D.E.<sup>2</sup>

<sup>1</sup>Institution of Russian Academy of Sciences Dorodnicyn Computing Centre of RAS, Moscow, Russia (119333, Moscow, Vavilov str. 40) e-mail: aidiveev@mail.ru

<sup>2</sup>Cybernetics and Mechatronics dept., People's Friendship University of Russia, Moscow, Russia (117198, Moscow, Miklukho-Maklaya str. 6), e-mail: kazaryan.david@gmail.com

---

We considered the control system synthesis problem for the nonlinear dynamic object. The control synthesis problem is a problem of the search for the function of the object state. Synthesized system should reach the chosen goal for the given set of the initial conditions. We stated the goal as a set of a quality functionals. We used grammatical evolution method and network operator method to solve the given problem. Grammatical evolution is a subclass of the genetic programming field, that uses production rules system to build the mathematical expression. In the network operator method mathematical expression is represented as a graph. We chose the steady-state genetic algorithm as a search engine for both methods. The solution was presented in a form of the Pareto set, that contained a set of the satisfactory control functions. We chose the nonlinear Duffing oscillator as a dynamic object. We conducted the computational experiment. Both methods were proved to be able to solve the control system synthesis problem.

---

Key words: control system synthesis, grammatical evolution, network operator, genetic algorithm.

### Введение

Необходимость поиска символьных выражений возникает при решении разнообразных задачи синтеза управления. Синтез системы управления состоит в построении управления в виде функции от координат пространства состояний объекта. Аналитические методы синтеза систем управления применимы только к узкому классу систем. Одна из основных причин сложности аналитического поиска многомерной функции управления состоит в том, что она, как правило, имеет разрывы первого рода по значениям или производным.

В данной работе рассматривается приложение двух методов из области генетического программирования к задаче синтеза системы управления. Генетическое программирование (ГП) было предложено Дж. Козой в [7] в качестве метода поиска программ, отвечающих заданному критерию. ГП применялось для решения задачи синтеза системы управления роботом-тележкой [6].

В последнее время с целью усовершенствования ГП появились методы грамматической эволюции [8; 9], аналитического программирования [10] и сетевого оператора [1–5]. В работе рассмотрены методы грамматической эволюции и сетевого оператора как наиболее подходящие для решения задачи синтеза управления.

Одной из главных проблем в области ГП является выбор представления решений, подходящего как для задачи, так и для выбранного метода поиска. Для представления решения в методе сетевого оператора используется направленный ациклический граф. Такой граф называется сетевым оператором. Он может быть представлен в виде верхней треугольной матрицы специального вида, построенной на основе матрицы смежности. Такое представление особенно удобно для проведения вычислительных экспериментов, так как матричное представление является естественным для большинства современных языков программирования. Однако метод сетевого оператора не лишён и недостатков, в числе которых следует назвать отсутствие возможности использования при поиске функций арности выше двух и невозможность включения в символьную запись оператора условия.

В грамматической эволюции для построения символьного выражения используются принципы формальной грамматики. Грамматика языка выражения задаётся в виде системы продукционных правил в форме Бэкуса-Наура (БНФ). Грамматическая эволюция лишена недостатков, присущих методу сетевого оператора, однако символьные выражения, получаемые в процессе работы метода, требуют либо поддержки функции *eval* со стороны языка, либо написания интерпретатора под каждую новую задачу.

В данной работе сравниваются описанные методы применительно к задаче синтеза управления нелинейной системой. В качестве объекта управления выбрана пружина Дуффинга.

### **Постановка задачи синтеза системы управления**

Задана модель объекта управления в виде системы обыкновенных дифференциальных уравнений:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}), \quad (1)$$

где, в общем случае,  $\mathbf{f}(\mathbf{x}, \mathbf{u})$  — нелинейная векторная функция векторных аргументов  $\mathbf{x} \in \mathbf{R}^n$ ,  $\mathbf{u} \in U^m \subseteq \mathbf{R}^n$ , где множество  $U$  замкнуто и ограничено,  $m < n$ .

Задано множество начальных условий

$$X_0 \subseteq \mathbf{R}^n. \quad (2)$$

Задано множество целевых значений

$$X_f \subseteq \mathbf{R}^n, X_f \cap X_0 = \emptyset. \quad (3)$$

Задан функционал качества управления

$$J = \int_0^{t_f} f_0(\mathbf{x}(t), \mathbf{u}(t)) dt, \quad (4)$$

где

$$t_f = \begin{cases} t, & \text{если } \mathbf{x}(t) \in X_f, \\ t^+ & \text{иначе,} \end{cases} \quad (5)$$

и  $t^+$  — заданная верхняя граница приемлемого времени управления.

Требуется найти функцию управления

$$\mathbf{h}(\mathbf{x}) : \mathbf{R}^n \rightarrow \mathbf{R}^m, \quad (6)$$

обеспечивающую минимум функционалу (4) и удовлетворяющую терминальным условиям

$$\mathbf{x}(t_f) \in X_f, t_f \leq t^+, \quad (7)$$

и ограничениям на управление

$$\mathbf{u} = \mathbf{h}(\mathbf{x}(t)) \in U, \quad (8)$$

для  $\forall \mathbf{x}(0) \in X_0$  системы (1).

В общем случае искомая функция управления (6) нелинейна и может иметь разрывы, что затрудняет или делает невозможным построение системы управления аналитическими методами.

### **Грамматическая эволюция**

Грамматическая эволюция (ГЭ), предложенная в [9], объединяет принципы генетического программирования и формальной грамматики. Первой особенностью ГЭ, отличающей её от ГП, состоит в том, что она обрабатывает популяцию целочисленных хромосом, тогда как ГП работает с символьными S-выражениями. Хромосомы могут быть как бинарными, так и содержать числа в десятичной системе счисления. Хромосомы могут иметь произвольную или фиксированную длину. ГЭ — модульный алгоритм. Для поиска решения можно использовать любой алгоритм, способный работать с популяциями. В данной работе использовался стационарный генетический алгоритм.

Для преобразования хромосомы в символьное выражение используется система особым образом пронумерованных продукционных правил, заданных в форме Бэкуса-Наура. Эти

правила описывают язык математического выражения. Для построения такой системы требуется ввести несколько множеств:  $N$  — множество нетерминальных символов,  $T$  — множество терминальных символов,  $D$  — множество нетерминальных символов, один из которых выбирается в качестве стартовой символьной строки.

Рассмотрим следующий пример. Пусть задано простейшее алгебраическое выражение

$$\sin x + 2 \times x. \quad (9)$$

Тогда для построения системы продукционных правил, в соответствии с которыми можно вывести выражение (9), можно использовать следующие множества:

$$\begin{aligned} N &= \{ \langle \text{expr} \rangle, \langle \text{op} \rangle, \langle \text{val} \rangle, \langle \text{function} \rangle \}, \\ T &= \{ 2, x, +, \times, \sin \}, \\ D &= \{ \langle \text{expr} \rangle \}. \end{aligned} \quad (10)$$

На основе множеств (10) строим систему, состоящую из  $r = 4$  продукционных правил, число которых соответствует количеству элементов множества  $N$  (таблица 1). Здесь символ “|” является обозначением оператора выбора между вариантами.

Таблица 1. Система продукционных правил для построения выражения (9).

№ правила	Выражение	Вариант	№ варианта
1	$\langle \text{expr} \rangle$	$\langle \text{expr} \rangle \langle \text{op} \rangle \langle \text{expr} \rangle \mid \langle \text{function} \rangle (\langle \text{expr} \rangle) \mid \langle \text{val} \rangle$	0   1   2
2	$\langle \text{op} \rangle$	+   $\times$	0   1
3	$\langle \text{val} \rangle$	2   $x$	0   1
4	$\langle \text{function} \rangle$	sin	0

В столбце «Выражение» находится выражение, подлежащее замене, в столбце «Вариант» расположены возможные варианты замены для заданного выражения. Пронумеруем варианты в каждом правиле числами от 0 до  $k_i - 1$ ,  $i = \overline{1, r}$ , где  $k_i$  — число вариантов замены для  $i$ -го правила.

Пусть задана целочисленная хромосома  $\mathbf{p} = [p_1 p_2 \dots p_n]^T$ . Реконструировать символьное выражение возможно по следующему алгоритму.

1. Пусть  $H$  — строка, содержащая символ из множества  $D$ . Зададим  $j := 1$ .
2. Выберем самый левый нетерминальный символ  $h \in N$  в строке  $H$  и определим номер правила  $i$  в соответствии с этим символом.
3. Заменяем  $h$  на вариант замены с номером  $q = p_j \bmod k_i$ .
4. Если в  $H$  нет нетерминальных символов и  $j \neq n$ , то  $j := j + 1$  и переходим к п. 2; если в  $H$  есть нетерминальные символы и  $j = n$ , то применяем оператор обёртывания [9]; иначе построение символьного выражения завершено.

Одной из возможных хромосом для выражения (9) будет  $\mathbf{p} = [0 \ 1 \ 0 \ 1 \ 0 \ 0 \ 0 \ 1 \ 1]^T$ .

### Метод сетевого оператора

Сетевой оператор [1–5] представляет собой ациклический ориентированный граф, который описывает некоторое математическое выражение. Для построения сетевого оператора вводятся четыре конечных упорядоченных множества: множество переменных  $X$ , множество параметров  $Q$ , множество унарных операций  $O_1$  и множество бинарных операций  $O_2$ , причём бинарные операции должны обладать свойствами коммутативности, ассоциативности и иметь единичные элементы. Данный граф обладает следующими характеристиками: каждому узлу-источнику сопоставлен элемент из множеств  $X$  или  $Q$ , каждой дуге графа сопоставлен элемент из множества  $O_1$ , каждому узлу, не являющемуся источником, сопоставлен элемент из множества  $O_2$ .

Сетевой оператор описывает несколько математических выражений, число которых равно числу узлов, не являющихся источниками. Обычно лишь часть таких узлов считаются «выходами» сетевого оператора.

Сетевой оператор может быть представлен в виде матрицы, построенной на основе матрицы смежности исходного графа. Матрица сетевого оператора  $\Psi$  имеет верхний треугольный вид.

Метод сетевого оператора успешно применялся для синтеза систем управления [1–5].

### Вычислительный эксперимент

Рассмотрим задачу синтеза управления пружины Дуффинга. Модель объекта управления описывается следующей системой уравнений:

$$\begin{cases} \dot{x}_1 = x_2, \\ \dot{x}_2 = -x_1 - x_1^3 + u(\mathbf{x}). \end{cases} \quad (11)$$

Функция управления ограничена:

$$u(\mathbf{x}) \in [-1; 1]. \quad (12)$$

Для выполнения ограничений (12) принимаем условия, что если в момент  $t_k$  найденная функция  $\mathbf{h}(\mathbf{x}(t_k)) > 1$ , то  $u(\mathbf{x}(t_k)) = 1$ , а если  $\mathbf{h}(\mathbf{x}(t_k)) < -1$ , то  $u(\mathbf{x}(t_k)) = -1$ .

Таблица 2. Система продукционных правил, применявшаяся в методе ГЭ.

№ правила	Выражение	Вариант	№ варианта
1	<expr>	(<expr> <op> (<expr>)   <f1 arg>(<expr>)   <val>	0   1   2
2	<op>	+   -   ×   /	0   1   2   3
3	<val>	$x_1$   $x_2$   $q_1$   $q_2$   $q_3$   $q_4$	0   1   2   3   4   5

4	<f1arg>	step   abs   sign   logist   absqrt   atan	0   1   2   3   4   5
---	---------	-----------------------------------------------	--------------------------

Пусть начальными условиями для данной системы будет множество

$$X_0 = \{ \mathbf{x}^{(0,1)}, \mathbf{x}^{(0,2)}, \mathbf{x}^{(0,3)}, \mathbf{x}^{(0,4)} \}, \quad (13)$$

где  $\mathbf{x}^{(0,1)} = [-1 \ -1]^T$ ,  $\mathbf{x}^{(0,2)} = [1 \ -1]^T$ ,  $\mathbf{x}^{(0,3)} = [-1 \ 1]^T$ ,  $\mathbf{x}^{(0,4)} = [1 \ 1]^T$ . Для всех начальных условий целевым множеством будет

$$X_f = \{ \mathbf{x}^{(f)} \}, \quad (14)$$

где  $\mathbf{x}^{(f)} = [0 \ 0]^T$ .

Будем искать функцию управления в соответствии с функционалами

$$J_1 = \sum_{i=1}^4 \left( \sqrt{\sum_{j=1}^2 \left( x_j(t_f^{(i)}) - x_j^{(f)} \right)^2} \right)_{\mathbf{x}(0)=\mathbf{x}^{(0,i)}} \rightarrow \min, \quad (15)$$

$$J_2 = \sum_{i=1}^4 t_f^{(i)} \rightarrow \min, \quad (16)$$

где индекс  $\mathbf{x}(0)=\mathbf{x}^{(0,i)}$  означает, что система (11) интегрируется при начальном условии  $\mathbf{x}^{(0,i)}$ ;  $t_f$  задаётся в соответствии с (5);  $t_f^{(i)}$  — время окончания моделирования для  $\mathbf{x}^{(i)}$ , а максимально приемлемое время  $t^+ = 15$  с.

Алгоритм ГЭ был модифицирован. Основная модификация заключается в том, что в генетическом алгоритме использовался принцип коэволюции: помимо популяции, хранящей закодированные выражения языка, для которого заданы множества  $N, T, D$  и система продукционных правил (таблица 2), генетический алгоритм обрабатывает также и популяцию числовых параметров, причём эти параметры входят в множество  $T$ . Метод сетевого оператора использует аналогичный принцип: вместе с популяцией вариаций базисного решения обработке с помощью генетических операторов скрещивания и мутации подвергается и популяция параметров, относящихся к множеству  $Q$ .

Методом грамматической эволюции была найдена следующая функция управления:

$$h(\mathbf{x}, \mathbf{q}) = \left( \sigma(\arctg q_4 + \operatorname{sgn} x_1) (\theta(\theta(\operatorname{sgn} x_1)) + \operatorname{sgn} x_2) - q_2 \right) x_1 - x_2 \Big| q_4^{-1} q_1$$

где  $\mathbf{q} = [14.0325 \ 15.6875 \ 27.1875 \ 3.625]^T$ ,  $\theta(z)$  — функция Хевисайда,  $\sigma(z) = 1/(1 + e^{-z})$ .

Методом сетевого оператора была найдена следующая функция управления:

$$h(\mathbf{x}, \mathbf{q}) = \begin{cases} u, & \text{если } |u| < 1, \\ 1, & \text{если } u \geq 1, \\ 0, & \text{если } u \leq -1, \end{cases}$$

где

$$\begin{aligned} u &= c + \sqrt[3]{a+b} + \operatorname{sgn}(c) \ln(|c|+1) + \operatorname{sgn}(a+b) \sqrt{|2(a+b)|}, \\ c &= 2(a+b) + \operatorname{sgn}(a+b) \sqrt{|a+b|} + \operatorname{sgn} a \sqrt{|a|}, \\ a &= q_1 x_1 + q_2 x_2 + \sqrt{q_1 x_1}, \\ b &= \sqrt[3]{x_1} + x_1 - x_1^3. \end{aligned}$$

Здесь вектор параметров  $\mathbf{q} = [0.36523 \quad 3.30566]^T$ .

Функции управления в методах ГЭ, и сетевого оператора выбирались из множества Парето оптимальных решений.

Синтезированное рассматриваемыми методами управление затем проверялось на новом множестве начальных условий, которые не использовались для процесса поиска,

$$X_N = \left\{ [2 \quad 0]^T, [0 \quad 2]^T, [-2 \quad 0]^T, [0 \quad -2]^T \right\}$$

Результаты моделирования системы, полученной методом грамматической эволюции, приведены на рис. 1; результаты моделирования системы, полученной методом сетевого оператора, приведены на рис. 2.

Результаты моделирования показывают, что оба метода успешно синтезировали функцию управления, решающую задачу приведения системы в точку, то есть синтезированные управляющие функции обеспечивают решение задачи синтеза не только для тех точек, которые использовались при поиске управления, но и для других точек из определенной области.

Рис. 1. Фазовые траектории системы с управлением, полученным методом ГЭ, для множеств начальных условий  $X_0$  и  $X_N$ .

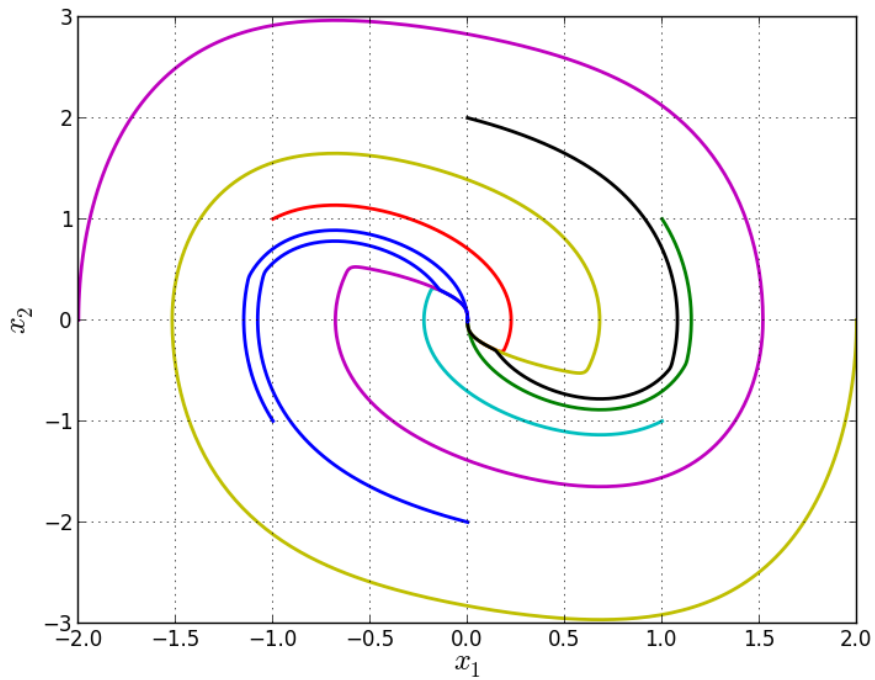
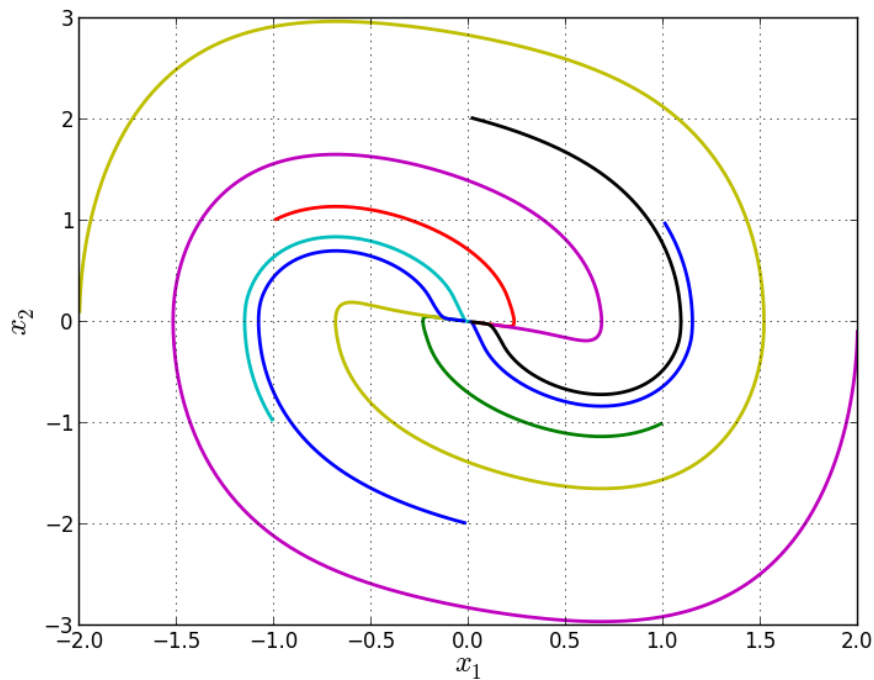


Рис. 2. Фазовые траектории системы с управлением, полученным методом сетевого оператора, для множеств начальных условий  $X_0$  и  $X_N$ .





## Список литературы

1. Дивеев А.И., Софронова Е.А. Метод генетического программирования для автоматического подбора формул в задаче структурного синтеза системы управления // Тр. Института системного анализа РАН. Динамика неоднородных систем / под ред. Ю.С. Попкова. - М. : ИСА РАН, КомКнига, 2006. - Вып. 10(1). - С. 14–26.
2. Дивеев А.И., Северцев Н.А. Метод сетевого оператора для синтеза системы управления спуском космического аппарата при неопределенных начальных условиях // Проблемы машиностроения и надежности машин. – 2009. - № 3. - С. 85–91.
3. Дивеев А.И. Синтез адаптивной системы управления методом сетевого оператора // Вопросы теории безопасности и устойчивости систем : сб. статей. - М. : ВЦ РАН, 2010. - Вып. 12. - С. 41–55.
4. Дивеев А.И. Численный метод сетевого оператора для синтеза системы управления с неопределенными начальными значениями // Известия РАН. Теория и системы управления. – 2012. - № 2. - С. 63–78.
5. Atencia V.J.M., Diveev A.I., Sofronova E.A. The Network Operator Method for Synthesis of Intelligent Control System, in Proc. of 7th IEEE Conference on Industrial Electronics and Applications, Singapore, 2012, pp. 169–174.
6. Bourmistrova A., Khantsis S. Control System Design Optimization via Genetic Programming, in Proceedings of IEEE Congress on Evolutionary Computation, Singapore, 2007, pp. 1993–2000.
7. Koza J.R. Genetic Programming: On the Programming of Computers by Means of Natural Selection. Cambridge: MIT Press, 1992, 840 p.
8. O'Neill M., Ryan C., Keijzer M., Cattolico M. Crossover in Grammatical Evolution. Genetic Programming and Evolvable Machines 4(1), 2003, pp. 67–93.
9. O'Neill M., Ryan C., Grammatical Evolution, IEEE Transactions on Evolutionary Computation 5(4), 2001, pp. 349–358.
10. Zelinka I., Analytic programming by Means of Soma Algorithm in Proceedings of the 8th International Conference on Soft Computing (MENDEL-02), Brno, Czech Republic: VUT v Brno, 2002, pp. 93–101.

### Рецензенты:

Дикусар В.В., д. ф. м. н., профессор, главный научный сотрудник ФГБУ «Вычислительный центр им. А.А. Дородницына» Российской академии наук, г. Москва.

Мочалов И.А., д. т. н, профессор, кафедра «Системы автоматического управления» (ИУ-1), Московский государственный технический университет им. Н.Э. Баумана, г. Москва.