

## РЕШЕНИЕ ЗАДАЧИ СИНТЕЗА СИСТЕМЫ УПРАВЛЕНИЯ МЕТОДОМ ВАРИАЦИОННОГО ГЕНЕТИЧЕСКОГО ПРОГРАММИРОВАНИЯ

Ибадулла С.И.<sup>2</sup>, Дивеев А.И.<sup>1</sup>, Софронова Е.А.<sup>2</sup>

<sup>1</sup>Федеральное государственное бюджетное учреждение науки Вычислительный центр им. А.А. Дородницына Российской академии наук, Москва, Россия (119333, Москва, ул. Вавилова, 40)

<sup>2</sup>Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования Российский университет дружбы народов, (117198, Москва, Миклухо-Маклая, 6)

Рассматривается задача синтеза системы управления, в которой необходимо найти управление как функцию от координат пространства состояний объекта. Для решения задачи предложено использовать новый метод вариационного генетического программирования. Приведено описание метода вариационного генетического программирования. В отличие от классического метода генетического программирования в новом вариационном методе генетического программирования все генетические операции выполняются на множествах векторов, описывающих малые вариации возможного решения. Определены малые вариации генетического программирования и предложена структура данных в виде целочисленного вектора для описания малой вариации. Для описания возможного решения предложено использовать упорядоченное множество векторов из двух компонент, первая из которых указывает на количество аргументов функции, а вторая на номер функции. Для описания малой вариации используется вектор из трех компонент: первая компонента указывает на номер вариации, вторая компонента устанавливает точки вариации, а третья компонента указывает на номер функции, если она необходима при выполнении вариации. Представлен численный пример синтеза системы управления мобильным роботом в условиях пространственных ограничений.

Ключевые слова: синтез системы управления, метод вариаций базисного решения, генетическое программирование

## CONTROL SYSTEM PROBLEM SOLUTION BY VARIATIONAL GENETIC PROGRAMMING METHOD

Ibadulla S.I.<sup>2</sup>, Diveev A.I.<sup>1</sup>, Sofronova E.A.<sup>2</sup>

<sup>1</sup>Institution of Russian Academy of Science Dorodnicyn Computing Centre of RAS, Moscow, Russia (119333, Moscow, Vavilova str., 40)

<sup>2</sup>Peoples' Friendship University of Russia (117198, Moscow, Miklukho-Maklaya str., 6)

We examine the problem of synthesis of control systems, where we need to find the control as a function of the space coordinates of the object's state. To solve the problem it is proposed to use a new method of variational genetic programming. A description of the method of variational genetic programming is given. In contrast to the classical method of genetic programming in a new variational method for genetic programming all genetic operations are performed on sets of vectors describing small variations of possible solutions. Small variations in genetic programming are defined and a data structure as an integer vector to describe a small variation is proposed. To describe a possible solution there proposed to use an ordered set of vectors of the two components, the first one of which indicates the number of arguments to a function, and the second one indicates the function index. To describe a small variation a three components' vector there used. The first component indicates the index of variation, the second part sets the points of variation, and the third component indicates the function index, if it is necessary while realizing the variation. There is a numerical example of the synthesis of mobile robot controlling system under spatial constraints conditions.

Keywords: control systems synthesis, the method of variations of the basis solutions, genetic programming

### Введение

Задача синтеза системы управления требует для своего решения нахождения многомерной функции, описывающей зависимость управления от координат пространства состояний объекта. Одним из эффективных численных методов ее решения является метод сетевого оператора [1-3]. По сравнению с другими численными методами символьной

регрессии, методом генетического программирования [4, 5], методом грамматической эволюции [6] или методом аналитического программирования [7], использует при поиске принцип малых вариаций базисного решения. Это позволит ввести метрику на пространстве математических выражений, ограничить область поиска в этом пространстве и определить направление поиска, задав успешное базисное решение близкое к оптимальному решению. Чем лучше разработчик на основе опыта и результатов исследования задачи задаст базисное решение, тем эффективнее алгоритм поиска решения.

Метод сетевого оператора ищет математические выражения, построенные на множестве операций с одним или двумя аргументами. Это ограничение не позволяет, например, синтезировать математические выражения, включающие оператор условия. Данный оператор необходим при синтезе систем с блоком принятия решений, но его реализация требует наличия не менее трех аргументов. Метод сетевого оператора при синтезе логико-функционального управления использует функцию дискретизации с применением функции Хэвисайда и поиск решения на множестве логических операций методом логического сетевого оператора [1, 2].

В настоящей работе рассматривается метод вариационного генетического программирования, который по аналогии с методом сетевого оператора использует принцип вариаций базисного решения, но обладает возможностью искать решения в форме математических выражений, построенных на множестве функций с любым количеством аргументов. Приведен пример синтеза системы управления мобильным роботом разработанным методом вариационного генетического программирования.

### **Постановка задачи**

Задана математическая модель объекта управления

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

где  $\mathbf{x} \in \mathbb{R}^n$ ,  $\mathbf{u} \in U \subseteq \mathbb{R}^m$ ,  $m \leq n$ ,  $U$  - ограниченное замкнутое множество.

Задано множество начальных состояний

$$X_0 = \{ \mathbf{x}^{0,1}, \dots, \mathbf{x}^{0,M} \}. \quad (2)$$

Задано терминальное состояние в форме  $n - r$  мерного многообразия

$$\varphi_i(\mathbf{x}(t_f)) = 0, \quad i = \overline{1, r}, \quad (3)$$

где  $t_f$  - время управления, определяемое выполнением условия (3)

Задан критерий качества управления

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

Необходимо найти управление в виде

$$\mathbf{u} = \mathbf{h}(\mathbf{x}), \quad (5)$$

где функция

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

после подстановке в систему (1) позволяет получить решение  $\mathbf{x}(t, \mathbf{x}^{0,i})$  системы  $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{h}(\mathbf{x}))$ , которое для любого начального состояния из (2)  $\mathbf{x}^{0,i} \in X_0$  попадает в терминальное состояние (3) при выполнении ограничений

$$\mathbf{h}(\mathbf{x}(t, \mathbf{x}^{0,i})) \in U \quad (7)$$

и минимизирует функционал (4).

При численном решении задачи заменим функционал (4) суммой функционалов, вычисленных для каждого начального состояния из (2)

$$\tilde{J} = \sum_{i=1}^k \left( \int_0^{t_f} f_0(\mathbf{x}(t), \mathbf{u}(t)) dt \right)_{\mathbf{x}^{0,i}} \rightarrow \min. \quad (8)$$

### Метод вариационного генетического программирования

Метод генетического программирования использует генетический алгоритм для строк символов. Символьная запись математического выражения в генетическом программировании представляет собой бескомматную строку символов. Каждый символ соответствует некоторой операции или функции. Каждая функция характеризуется определенным количеством аргументов. Функции без аргументов являются переменными или параметрами. Порядок символов в строке определяет соответствие между функциями и их аргументами. В генетическом программировании чаще используют префиксную запись символов. В данной записи символ функции стоит в строке раньше или левее символа аргумента.

Введём множество упорядоченных наборов функций с определённым количеством аргументов

$$F = \{F_0, \dots, F_n\}, \quad (9)$$

где

$$F_i = (f_1(z_1, \dots, z_i), \dots, f_{m_i}(z_1, \dots, z_i)) \quad , \quad i = \overline{0, n}, \quad (10)$$

$f_j(z_1, \dots, z_i)$  - функция под номером  $j$  с количеством аргументов  $i$ .

В качестве кода функции используем целочисленный вектор из двух элементов

$$\mathbf{s} = [s_1 \ s_2]^T, \quad (11)$$

где  $s_1$  - количество аргументов функции,  $s_2$  - номер функции в множестве  $F_{s_1}$ .

Согласно принятым обозначениям для функции  $f_j(z_1, \dots, z_i)$  получаем

$$\mathbf{s} = [i \ j]^T. \quad (12)$$

Математическое выражение представляет собой упорядоченное множество кодов функций

$$S = (\mathbf{s}^1, \dots, \mathbf{s}^K), \quad (13)$$

где  $\mathbf{s}^i = [s_1^i \ s_2^i]^T$ ,  $i = \overline{1, K}$ .

Рассмотрим пример. Пусть имеем математическое выражение

$$y = e^{-ax} \cos(bx + c).$$

Для данного выражения имеем следующие множества функций

$$F_0 = (a, b, c, x), \quad F_1 = (-z, e^z, \cos(z)), \quad F_2 = (z_1 + z_2, z_1 \cdot z_2).$$

Запишем функции в кодах

$$\bar{F}_0 = \left( \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ 2 \end{bmatrix}, \begin{bmatrix} 0 \\ 3 \end{bmatrix}, \begin{bmatrix} 0 \\ 4 \end{bmatrix} \right), \quad \bar{F}_1 = \left( \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \begin{bmatrix} 1 \\ 3 \end{bmatrix} \right), \quad \bar{F}_2 = \left( \begin{bmatrix} 2 \\ 1 \end{bmatrix}, \begin{bmatrix} 2 \\ 2 \end{bmatrix} \right),$$

где  $\bar{F}_i$  - множество кодов функций множества  $F_i$ .

В рассматриваемом математическом выражении используем четыре раза функции с двумя аргументами, три раза функции с одним аргументом и пять раз функции без аргументов или параметры и переменные

$$y = \left( e^{-(a \cdot x)} \right) \cdot \left( \cos((b \cdot x) + c) \right).$$

Запись математического выражения в кодах имеет вид

$$S = \left( \begin{bmatrix} 2 \\ 2 \end{bmatrix}, \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 2 \\ 2 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ 4 \end{bmatrix}, \begin{bmatrix} 1 \\ 3 \end{bmatrix}, \begin{bmatrix} 2 \\ 1 \end{bmatrix}, \begin{bmatrix} 2 \\ 2 \end{bmatrix}, \begin{bmatrix} 0 \\ 2 \end{bmatrix}, \begin{bmatrix} 0 \\ 4 \end{bmatrix}, \begin{bmatrix} 0 \\ 3 \end{bmatrix} \right).$$

При составлении кода математического выражения используем правила:

- а) код первого аргумента функции следует непосредственно сразу после кода функции;
- б) коды второго и следующих аргументов функции следуют после кода функции с нулевым количеством аргументов;
- в) каждый код аргумента относится к ближайшей слева функции с недостающим количеством аргументов.

Из установленных правил следуют определенные свойства кода записи математического выражения.

Введём понятие индекса для символа записи математического выражения. Индекс  $T(j)$  символа  $\mathbf{s}^j$  математического выражения (13) указывает на минимальное количество недостающих справа кодов символов

$$T(j) = 1 - j + \sum_{i=1}^j s_1^i, \quad (14)$$

где

$$T(j) > 0, \quad j = \overline{1, K-1}, \quad (15)$$

$$T(K) = 0, \quad (16)$$

$K$  - количество символов в математическом выражении.

Рассмотрим малые вариации записей математического выражения (13).

Основная задача малых вариаций состоит в получении с их помощью преобразование одной записи математического выражения в любую другую запись.

Любая вариация записи не должна нарушать условия (15), (16). Удовлетворяющие условиям (15), (16) записи математических выражений считаем допустимыми.

При выполнении вариации необходимо учитывать максимально возможное значение первой компоненты кода символа. Значение первой компоненты указывает на количество аргументов функции или на множество функций с определенным количеством аргументов. Пусть используем функции с количеством аргументов не больше  $m$ . С учетом максимального количества  $m$  аргументов функции значение первой компоненты кода определяем в виде остатка от деления на величину  $m + 1$

$$s_1^j = s_1^j \bmod(m + 1). \quad (17)$$

Значение второй компоненты кода элемента не должно превышать количество функций в указанном первой компонентой множестве. Для этой цели после изменения значения первой компоненты значение второй компоненты определяем в виде остатка от деления этого значения на количество функций во множестве плюс один

$$s_2^j \bmod(|F_t| + 1), \quad \text{где } t = s_1^j. \quad (18)$$

В результате получаем пять малых вариаций записей кодов в генетическом программировании:

0 – изменение значения второй компоненты кода элемента;

1 – удаление кода элемента с единичным значением первой компоненты;

2 – вставка в заданную позицию кода элемента с единичным значением первой компоненты;

3 – увеличение значения первой компоненты кода элемента на единицу и добавление в последнюю позицию кода с нулевым значением первой компоненты;

4 – уменьшение значения первой компоненты кода элемента на единицу и удаление начинающегося со следующего элемента подвыражения.

Для описания малой вариации используем вектор из трех компонент

$$\mathbf{w} = [w_1 \ w_2 \ w_3]^T, \quad (19)$$

где  $w_1$  - номер вариации,  $w_2$  - номер варьируемого элемента или позиция для вставки нового элемента,  $w_3$  - значение второй компоненты.

При решении задачи синтеза управления методом вариационного генетического программирования используем генетический алгоритм.

Для удобства вычислений определим нулевой вектор вариаций

$$\mathbf{w} = [0 \ 0 \ 0]^T.$$

Нулевой вектор вариаций не выполняет вариации с записью кода

$$S = \mathbf{w} \circ S.$$

Множество функций без аргументов разделим на подмножества переменных и параметров

$$F_0 = (x_1, \dots, x_n, q_1, \dots, q_p), \quad F_0 = F_x \cup F_q, \quad \forall \mathbf{s} = [0 \ s_2]^T, \quad (20)$$

$$f_{0,s_2} = \begin{cases} x_{s_2}, & \text{если } s_2 \leq n, \\ q_{s_2-n}, & \text{если } s_2 > n. \end{cases}$$

Значения элементов вектора параметров  $\mathbf{q} = [q_1 \ \dots \ q_p]^T$  целесообразно искать вместе с кодами математических выражений. Для поиска значений параметров используем тот же генетический алгоритм.

Приведем этапы генетического алгоритма

Задаём базисное решение

$$u_i = h_i^0(\mathbf{x}, \mathbf{q}^0), \quad i = \overline{1, m}, \quad (21)$$

где  $\mathbf{q}^0 = [q_1^0 \ \dots \ q_p^0]^T$  - базисное значение вектора параметров.

Базисное решение определяем в виде вектора из строк кодов элементов

$$\mathbf{S}^0 = [S_1^0 \ \dots \ S_m^0]^T, \quad (22)$$

где  $S_i^0 = (s^{0,i,1}, \dots, s^{0,i,K_i})$ ,  $i = \overline{1, m}$ .

Базисное значение вектора параметров кодируем в форме кода Грея

$$\mathbf{z}^0 = [z_1^0 \ \dots \ z_{p(c+d)}^0]^T, \quad (23)$$

где  $z_i^0 \in \{0,1\}$ ,  $i = \overline{1, p(c+d)}$ ,  $c$  - число бит для целой части числа,  $d$  - число бит для дробной части числа.

Каждое решение определяем с помощью набора заданной длины  $r$  из векторов вариаций.

Для определения базисного решения определяем набор из нулевых векторов вариаций

$$\mathbf{W}^0 = (\mathbf{w}^{0,1}, \dots, \mathbf{w}^{0,r}), \quad (24)$$

где  $\mathbf{w}^{0,i} = [0 \ 0 \ 0]^T$ ,  $i = \overline{1, r}$ .

Генерируем множество возможных решений в виде пар из кодов параметров и наборов векторов вариаций

$$\mathbf{G} = ((\mathbf{W}^0, \mathbf{z}^0), \dots, (\mathbf{W}^H, \mathbf{z}^H)) \quad (25)$$

Каждое возможное решение для вычисления значений функционалов преобразуем в записи кодов и векторы параметров

$$(\mathbf{W}^j, \mathbf{z}^j) \leftrightarrow (\mathbf{S}^j, \mathbf{q}^j), \quad (26)$$

где  $\mathbf{S}^j = [S_1^j \ \dots \ S_m^j]^T$ ,

$$S_i^j = \mathbf{w}^{j,r} \circ \dots \circ \mathbf{w}^{j,1} \circ S_i^j. \quad i = \overline{1, m}. \quad (27)$$

При выполнении операции скрещивания генетического алгоритма отбираем два возможных решения  $(\mathbf{W}^{i_1}, \mathbf{z}^{i_1})$ ,  $(\mathbf{W}^{i_2}, \mathbf{z}^{i_2})$ . Определяем две точки скрещивания  $k_1$ ,  $k_2$ ,  $1 \leq k_1 \leq r$ ,  $1 \leq k_2 \leq p(c+d)$ .

Строим четыре новых решения  $(\mathbf{W}^{H+i}, \mathbf{z}^{H+i})$ ,  $i = \overline{1, 4}$ . Два новых решения получаем в результате скрещивания кодов для структурной и параметрической частей. Два других новых решения получаем в результате скрещивания кодов только для кодов параметров

$$\mathbf{W}^{H+1} = (\mathbf{w}^{i_1,1}, \dots, \mathbf{w}^{i_1, k_1-1}, \mathbf{w}^{i_2, k_1}, \dots, \mathbf{w}^{i_2, r}), \quad \mathbf{W}^{H+2} = (\mathbf{w}^{i_1, 2}, \dots, \mathbf{w}^{i_2, k_1-1}, \mathbf{w}^{i_1, k_1}, \dots, \mathbf{w}^{i_1, r}),$$

$$\mathbf{W}^{H+3} = \mathbf{W}^{i_1}, \quad \mathbf{W}^{H+4} = \mathbf{W}^{i_2},$$

$$\mathbf{z}^{H+1} = [z_1^{i_1} \ \dots \ z_{k_2-1}^{i_1} \ z_{k_2}^{i_2} \ \dots \ z_{p(c+d)}^{i_2}]^T, \quad \mathbf{z}^{H+2} = [z_1^{i_2} \ \dots \ z_{k_2-1}^{i_2} \ z_{k_2}^{i_1} \ \dots \ z_{p(c+d)}^{i_1}]^T,$$

$$\mathbf{z}^{H+3} = [z_1^{i_1} \ \dots \ z_{k_2-1}^{i_1} \ z_{k_2}^{i_2} \ \dots \ z_{p(c+d)}^{i_2}]^T, \quad \mathbf{z}^{H+4} = [z_1^{i_2} \ \dots \ z_{k_2-1}^{i_2} \ z_{k_2}^{i_1} \ \dots \ z_{p(c+d)}^{i_1}]^T.$$

Выполняем операцию мутации для каждого нового возможного решения  $(\mathbf{W}^{H+i}, \mathbf{z}^{H+i})$ ,  $i = \overline{1, 4}$ . Случайно с заданной вероятностью мутации  $P_\mu$  для каждого нового решения определяем точки мутации  $m_1$ ,  $m_2$ ,  $1 \leq m_1 \leq r$ ,  $1 \leq m_2 \leq p(c+d)$ . Генерируем новое значение вектора вариаций  $\tilde{\mathbf{w}}^{H+i}$  и новое значение компоненты  $\tilde{z}^{H+i} \in \{0,1\}$ . Помещаем сгенерированные значения в позиции  $m_1$ ,  $m_2$  новых возможных решений

$$\mathbf{W}^{H+i} = (\mathbf{w}^{H+i,1}, \dots, \mathbf{w}^{H+i, m_1-1}, \tilde{\mathbf{w}}^{H+i}, \mathbf{w}^{H+i, m_1+1}, \dots, \mathbf{w}^{H+i, r}),$$

$$\mathbf{z}^{H+i} = [z_1^{H+i} \ \dots \ z_{m_2-1}^{H+i} \ \tilde{z}^{H+i} \ z_{m_2+1}^{H+i} \ \dots \ z_{p(c+d)}^{H+i}]^T, \quad i = \overline{1, 4}.$$

Проверяем для каждого нового возможного решения значение функции приспособленности. Остальные операции выполняем в соответствии с правилами выбранной для поиска решения разновидности генетического алгоритма.

Для синтеза систем управления в большинстве случаев достаточно использовать множества из аргументов и параметров

$$F_0 = (x_1, \dots, x_n, q_1, \dots, q_p), \quad (28)$$

и следующие множества из функций с одним, двумя и тремя аргументами

$$\begin{aligned} F_1 = & (f_{1,0}(z) = z, f_{1,1}(z) = \operatorname{sgn}(z)z^2, f_{1,2}(z) = z^2, f_{1,3}(z) = -z, f_{1,4}(z) = \operatorname{sgn}(z)\sqrt{|z|}, \\ & f_{1,5}(z) = z^{-1}, f_{1,6}(z) = e^z, f_{1,7}(z) = \ln|z|, f_{1,8}(z) = \frac{1-e^{-z}}{1+e^{-z}}, f_{1,9}(z) = \begin{cases} 1, & \text{if } z \geq 0 \\ 0, & \text{otherwise} \end{cases}, \\ & f_{1,10}(z) = \operatorname{sgn}(z), f_{1,11}(z) = \cos(z), f_{1,12}(z) = \sin(z), f_{1,13}(z) = \arctan(z), f_{1,14}(z) = z^3, \\ & f_{1,15}(z) = \sqrt[3]{z}, f_{1,16}(z) = \begin{cases} z, & \text{if } |z| \leq 1 \\ \operatorname{sgn}(z), & \text{otherwise} \end{cases}, f_{1,17}(z) = \operatorname{sgn}(z)\ln(|z|+1), \\ & f_{1,18}(z) = \operatorname{sgn}(z)(e^{-|z|}-1), f_{1,19}(z) = \operatorname{sgn}(z)(1-e^{-|z|}), f_{1,20}(z) = \begin{cases} 1, & \text{if } z > a \\ -1, & \text{if } z < 0 \\ 3za^{-2} - 2z^3a^{-3}, & \text{otherwise} \end{cases}, \\ & f_{1,21}(z) = \begin{cases} 1, & \text{if } z > a/2 \\ -1, & \text{if } z < -a/2 \\ 3za^{-2} - 4z^3a^{-3}, & \text{otherwise} \end{cases}, f_{1,22}(z) = z - z^3, \end{aligned} \quad (29)$$

$$\begin{aligned} F_2 = & (f_{2,0}(z_1, z_2) = z_1 + z_2, f_{2,1}(z_1, z_2) = z_1 z_2, f_{2,2}(z_1, z_2) = \max\{z_1, z_2\}, f_{2,3}(z_1, z_2) = \min\{z_1, z_2\}, \\ & f_{2,4}(z_1, z_2) = z_1 + z_2 - z_1 z_2, f_{2,5}(z_1, z_2) = \operatorname{sgn}(z_1 + z_2)\sqrt{z_1^2 + z_2^2}, \\ & f_{2,6}(z_1, z_2) = \operatorname{sgn}(z_1 + z_2)(|z_1| + |z_2|), f_{2,7}(z_1, z_2) = \operatorname{sgn}(z_1 + z_2)\max\{|z_1|, |z_2|\}), \end{aligned} \quad (30)$$

$$\begin{aligned} F_3 = & \left( f_{3,0}(z_1, z_2, z_3) = \begin{cases} z_2, & \text{if } z_1 \leq 0 \\ z_3, & \text{otherwise} \end{cases}, f_{3,1}(z_1, z_2, z_3) = \begin{cases} z_3, & \text{if } z_1 \leq z_2 \\ -z_3, & \text{otherwise} \end{cases}, \right. \\ & \left. f_{3,2}(z_1, z_2, z_3) = \begin{cases} z_2, & \text{if } z_1 < z_2 \\ z_3, & \text{if } (z_1 \geq z_2) \wedge (z_1 > z_3) \\ z_1, & \text{if } (z_1 \geq z_2) \wedge (z_1 \leq z_3) \end{cases}, f_{3,3}(z_1, z_2, z_3) = \begin{cases} z_2 - z_1, & \text{if } |z_1 - z_2| < |z_1 - z_3| \\ z_3 - z_1, & \text{otherwise} \end{cases} \right), \end{aligned} \quad (31)$$

Рассмотрим пример. На рис. 1 приведена схема управления и геометрические параметры мобильного робота.

Математическая модель объекта управления [8] имеет следующий вид:

$$\dot{x} = u_1 \cos(\theta), \quad (32)$$

$$\dot{y} = u_1 \sin(\theta), \quad (33)$$

$$\dot{\theta} = \frac{u_1}{L_b} \tan(u_2), \quad (34)$$

где  $x, y$  - координаты центра масс мобильного робота;  $\theta$  - угол направления вектора скорости,  $L_b$  - обобщенный габаритный параметр мобильного робота.



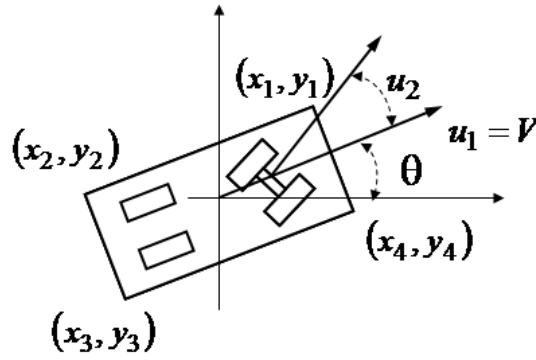


Рис. 1 Схема управления мобильным роботом

На рис. 1  $x_i, y_i, i = \overline{1,4}$ , - координаты углов робота. Положение углов робота определяются с помощью соотношений

$$x_i = x'_i \cos(\theta) - y'_i \sin(\theta), y_i = x'_i \sin(\theta) + y'_i \cos(\theta), i = \overline{1,4}, \quad (35)$$

$$x'_1 = x \cos(\theta) + y \sin(\theta) + L_b, y'_1 = -x \sin(\theta) + y \cos(\theta) + L_b / 4, \quad (36)$$

$$x'_2 = x \cos(\theta) + y \sin(\theta) - L_b, y'_2 = -x \sin(\theta) + y \cos(\theta) + L_b / 4, \quad (37)$$

$$x'_3 = x \cos(\theta) + y \sin(\theta) - L_b, y'_3 = -x \sin(\theta) + y \cos(\theta) - L_b / 4, \quad (38)$$

$$x'_4 = x \cos(\theta) + y \sin(\theta) + L_b, y'_4 = -x \sin(\theta) + y \cos(\theta) - L_b / 4, \quad (39)$$

Значения управлений ограничены

$$V^- \leq u_1 \leq V^+, u_2^- \leq u_2 \leq u_2^+, \quad (40)$$

Задано множество начальных условий

$$X_0 = \{(x_{0,1}, y_{0,1}, \theta_{0,1}), \dots, (x_{0,k}, y_{0,k}, \theta_{0,k})\}. \quad (41)$$

Задано терминальное многообразие

$$x - x^f = 0, y - y^f = 0, \theta - \theta^f = 0. \quad (42)$$

где  $x^f, y^f, \theta^f$  - координаты терминального положения центра масс робота.

Заданы пространственные ограничения в виде логических функций

$$\left( (x_j - x^{\alpha_i}) < 0 \right) \vee \left( (x_j - x^{\beta_i}) > 0 \right) \wedge \left( (y_j - y^{\alpha_i}) < 0 \right) \vee \left( (y_j - y^{\beta_i}) > 0 \right), j = \overline{1,4} \quad (43)$$

где  $x^{\alpha_i} < x^{\beta_i}, y^{\alpha_i} < y^{\beta_i}, i = \overline{1, k_p}, k_p$  - число препятствий.

Необходимо найти управление в форме функций от координат пространства состояний

$$u_1 = h_1(x, y, \theta), u_2 = h_2(x, y, \theta) \quad (44)$$

Управление (44) должно обеспечивать перемещение робота из любого заданного в (41) начального состояния в терминальное положение (42), удовлетворяя ограничениям (43) для углов робота за минимальное время

$$t_f \rightarrow \min. \quad (45)$$

Для решения задачи используем вариационный генетический алгоритм. В численном эксперименте использовали следующие значения параметров:  $L_b = 4$ ,  $V^- = -5$ ,  $V^+ = 5$ ,  $u^- = -1$ ,  $u^+ = 1$ ,  $k_p = 2$ ,  $x^{\alpha_1} = -20$ ,  $y^{\alpha_1} = -2$ ,  $x^{\beta_1} = -5$ ,  $y^{\alpha_2} = 2$ ,  $x^{\alpha_2} = 5$ ,  $y^{\alpha_2} = -2$ ,  $x^{\beta_2} = 20$ ,  $y^{\beta_2} = 2$ .

Множество начальных значений имело следующий вид:  $X_0 = \{(-8, -4, 0), (8, -4, 0), (-8, 4, 0), (8, 4, 0)\}$ . Терминальные условия имели значения:  $x^f = 0$ ,  $y^f = 0$ ,  $\theta^f = 0$ . Множество аргументов включало следующие элементы  $F_0 = (x^f - x, y^f - y, \theta^f - \theta, q_1, q_2, q_3)$ , где  $q_1, q_2, q_3$  - числовые параметры, которые также ищутся вместе со структурами функций (44).

Задаем базисное решение в виде

$$u_i = \begin{cases} u_i^-, & \text{если } \tilde{u}_i \leq u_i^- \\ u_i^+, & \text{если } \tilde{u}_i \geq u_i^+ \\ \tilde{u}_i, & \text{иначе} \end{cases}, i = 1, 2,$$

где  $\tilde{u}_1 = q_1(x^f - x) + q_2(y^f - y)$ ,  $\tilde{u}_2 = q_3(\theta^f - \theta)$ ,  $q_1 = 1$ ,  $q_2 = 1$ ,  $q_3 = 1$ .

С учетом множеств функций (28)-(31) запись базисного решения имеет вид

$$S_1 = \left( \begin{bmatrix} 2 \\ 0 \end{bmatrix}, \begin{bmatrix} 2 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 3 \end{bmatrix}, \begin{bmatrix} 2 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ 4 \end{bmatrix} \right), S_2 = \left( \begin{bmatrix} 2 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 2 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 5 \end{bmatrix} \right).$$

Для учета ограничения использовали функцию штрафа, которая добавлялась к значению функционала (45)

$$p_i = \begin{cases} 0, & \text{если } \Delta x \Delta y < 0 \\ \int_0^{t_f} \left( \sqrt{(\Delta x)^2 + (\Delta y)^2} \right) dt - \text{иначе}, & i = \overline{1, 4}, \end{cases}$$

где  $\Delta x = \min_j \left\{ \left( x^{\alpha_j} - x^i \right), \left( x^{\beta_j} - x^i \right) \right\}$ ,  $\Delta y = \min_j \left\{ \left( y^{\alpha_j} - y^i \right), \left( y^{\beta_j} - y^i \right) \right\}$ .

При расчетах были использованы следующие параметры генетического алгоритма: количество возможных решений в начальной популяции  $H = 1024$ ; число поколений 128; число попыток скрещиваний в одном поколении 512; число вариаций одного возможного решения 8; число бит для кода целой части параметра 4; число бит для кода дробной части параметра 12; число поколений между сменой базисного решения 24.

В результате синтеза было получено следующее решение

$$S_1 = \left( \begin{bmatrix} 2 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 2 \\ 6 \end{bmatrix}, \begin{bmatrix} 1 \\ 21 \end{bmatrix}, \begin{bmatrix} 1 \\ 7 \end{bmatrix}, \begin{bmatrix} 1 \\ 17 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \begin{bmatrix} 0 \\ 3 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \end{bmatrix} \right),$$

$$S_2 = \left( \begin{bmatrix} 3 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 7 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 10 \end{bmatrix}, \begin{bmatrix} 1 \\ 18 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 14 \end{bmatrix}, \begin{bmatrix} 0 \\ 5 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 17 \end{bmatrix}, \begin{bmatrix} 1 \\ 12 \end{bmatrix}, \right.$$

$$\left. \begin{bmatrix} 1 \\ 19 \end{bmatrix}, \begin{bmatrix} 1 \\ 21 \end{bmatrix}, \begin{bmatrix} 1 \\ 5 \end{bmatrix}, \begin{bmatrix} 1 \\ 20 \end{bmatrix}, \begin{bmatrix} 1 \\ 7 \end{bmatrix}, \begin{bmatrix} 1 \\ 15 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 7 \end{bmatrix}, \begin{bmatrix} 1 \\ 15 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right).$$

$$q_1 = 12.58398, \quad q_2 = 3.95142, \quad q_3 = 7.75732.$$

На рис. 2-4 приведены результаты моделирования синтезированной системы управления.

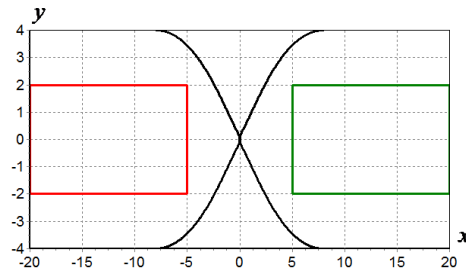


Рис. 2 Траектории движения центра масс объекта для различных начальных состояний

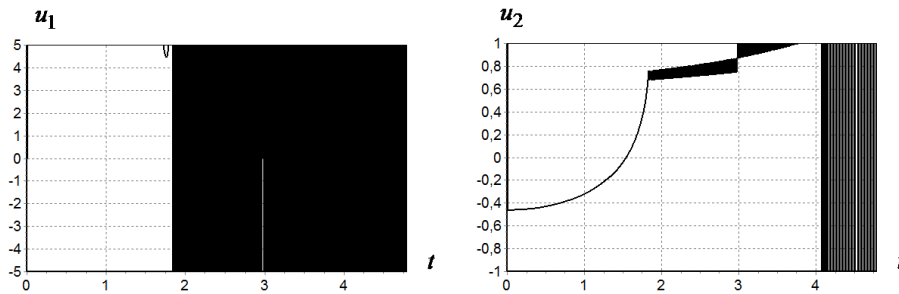


Рис. 3 Графики управлений для начального состояния  $x(0) = -8$ ,  $y(0) = 4$ ,  $\theta(0) = 0$

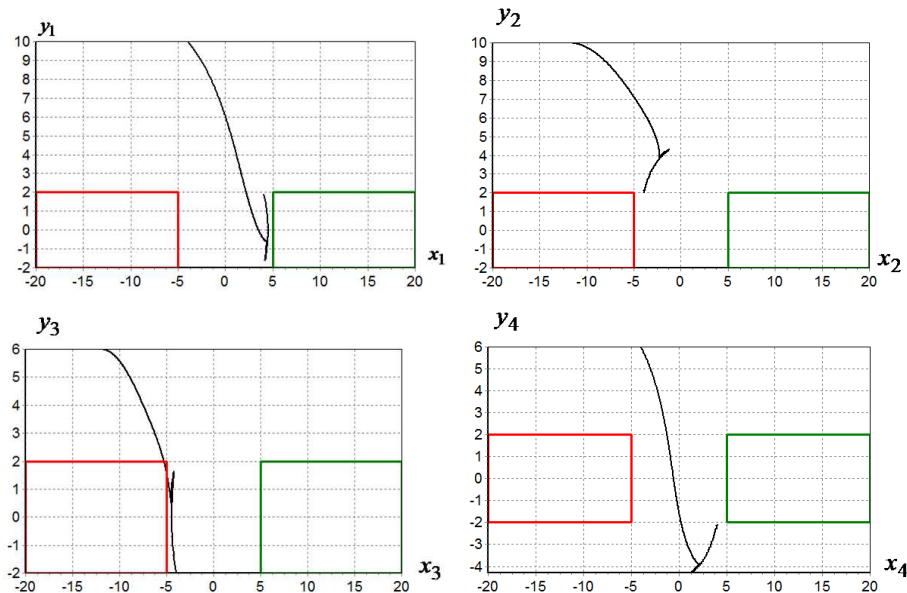


Рис. 4 Траектории движения углов объекта для начального состояния  $x(0) = -8$ ,  $y(0) = 4$ ,  $\theta(0) = 0$

## Список литературы

1. Дивеев А.И. Метод сетевого оператора. М.: ВЦ РАН, 2010. 178 с.
2. Дивеев А.И., Софронова Е.А. Метод сетевого оператора и его применение в задачах управления. М.: Изд-во РУДН, 2012. 182 с.
3. 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.
4. Koza J.R. Genetic Programming: On the Programming of Computers by Means of Natural Selection. Cambridge, Massachusetts, London, MA: MIT Press, 1992, 819 p.
5. Bourmistrova A., Khantsis S., Genetic Programming in Application to Flight Control System Design Optimization. Chapter in the book New Achievements in Evolutionary Computation, Book edited by: Peter Korosec, 318 p, February 2010, INTECH, Croatia, P. 195-226.
6. O'Neill M., Automatic Programming in an Arbitrary Language: Evolving Programs in Grammatical Evolution: PhD thesis. University of Limerick, 2001 – 163 p. Kristinsson, K., and Dumont G. A. System identification and control using genetic algorithms, IEEE Transactions on Systems, Man, and Cybernetics, Vol. 22(5), 1992, – P. 1033 – 1046.
7. Zelinka I., Analytic programming by Means of Soma Algorithm // Mendel '02 In: Proc. 8th International Conference on Soft Computing Mendel'02, Brno, Czech Republic, 2002. P. 93-101.
8. Kentaro Oyama and Kenichiro Nonaka Model Predictive Parking Control for Nonholonomic Vehicles using Time-State Control Form// 2013 European Control Conference (ECC) July 17-19, 2013, Zürich, Switzerland. P. 458-465.

### Рецензенты:

Воронин Е.А., д.т.н., профессор, заведующий кафедрой Вычислительной техники и прикладной математики Московского государственного агроинженерного университета им. В.П. Горячкина, г.Москва.

Никутьчев Е.В., д.т.н., профессор, проректор по научной работе НОУ ВПО Московский технологический институт "ВТУ", г.Москва.