

# МОДИФИЦИРОВАННЫЙ АЛГОРИТМ ПОСТРОЕНИЯ ОБЛАСТЕЙ ДОСТИЖИМОСТИ ЛИНЕЙНЫХ ДИСКРЕТНЫХ ДИНАМИЧЕСКИХ СИСТЕМ

**В.В. Булаев<sup>1, 2</sup>, И.О. Осипов<sup>3</sup>**

<sup>1</sup> Уральский федеральный университет им. первого Президента России Б.Н. Ельцина, г. Екатеринбург, Россия,

<sup>2</sup> АО «НПО Автоматики им. академика Н.А. Семихатова», г. Екатеринбург, Россия,

<sup>3</sup> Южно-Уральский государственный университет, г. Челябинск, Россия

Рассматривается модифицированный общий рекуррентный алгебраический метод построения областей достижимости линейных дискретных динамических систем. Приводятся алгоритмы прикладных программ для построения областей достижимости. На численном примере производится сравнительный анализ оригинального (Шориков А.Ф. Минимаксное оценивание и управление в дискретных динамических системах. Екатеринбург: Изд-во Урал. гос. ун-та, 1997. 242 с.) и предлагаемого алгоритмов.

*Ключевые слова:* области достижимости, дискретная динамическая система, выпуклые множества.

## Введение

В ряде задач современной теории управления возникает потребность в построении и анализе областей достижимости исследуемой динамической системы. В рамках данной статьи изложен модифицированный алгоритм построения точных областей достижимости линейных дискретных динамических систем (ЛДДС).

В основе предлагаемой модификации лежит общий рекуррентный алгебраический метод построения областей достижимости [1, 2], разработанный А.Ф. Шориковым и В.А. Тюлюкиным (Институт математики и механики имени Н.Н. Красовского, г. Екатеринбург). Детальное описание модификаций оригинального метода изложено в работе [3]. Целью настоящей статьи является изложение алгоритмов пакета программ для построения областей достижимости ЛДДС, которые могут быть без особых затруднений реализованы на современных ЭВМ.

## 1. Теоретические основания алгоритма

Для построения дальнейших рассуждений примем ряд обозначений:  $\mathbb{R}^n$  –  $n$ -мерное евклидово пространство векторов столбцов;  $\mathbb{N}$  – множество всех натуральных чисел;  $\emptyset$  – пустое множество;  $X^{[i]}$  –  $i$ -я строка матрицы  $X$ ;  $X_{[j]}$  –  $j$ -й столбец матрицы  $X$ ; символом  $[\setminus j]$  обозначим все множество элементов, за исключением  $j$ -го; множество всех вершин выпуклого компактного и ограниченного многогранника  $X$  будем считать матрицей, обозначаемой с «крышкой» –  $\widehat{X}$ , в которой координаты каждой вершины представлены векторами-столбцами;  $I_n$  – единичная матрица  $n \times n$ ;  $E_{n \times m}$  – матрица  $n \times m$ , состоящая из единиц;  $O_{n \times m}$  – нулевая матрица  $n \times m$ ;  $NaN$  – неопределенность, необходимая при описании алгоритма.

Рассмотрим класс ЛДДС на целочисленном промежутке времени  $t \in \overline{0, T}$ ,  $T \in \mathbb{N}$ :

$$x(t+1) = A(t)x(t) + B(t)u(t), \quad (1)$$

где  $x(t)$  – фазовый вектор системы,  $x(t) \in \mathbb{R}^n$ ;  $u(t)$  – вектор управления,  $u(t) \in \mathbb{R}^p$ ;  $A(t)$  – матрица состояния системы,  $A(t) \in \mathbb{R}^{n \times n}$ ;  $B(t)$  – матрица управления,  $B(t) \in \mathbb{R}^{n \times p}$ .

В дальнейшем примем ряд предположений относительно динамики исследуемой системы. Будем считать, что множество начальных состояний системы (1) является выпуклым, замкнутым и ограниченным многогранником с конечным числом вершин:

$$x(0) \in \mathbf{X}(0) \subset \mathbb{R}^n. \quad (2)$$

Также положим, что вектор управления удовлетворяет геометрическим ограничениям

$$u(t) \in \mathbf{P} \subset \mathbb{R}^p \quad \forall t \in \overline{0, T-1}, \quad (3)$$

где  $\mathbf{P}$  является выпуклым, замкнутым и ограниченным многогранником с конечным числом вершин.

При условии выполнения указанных предположений введем определение области достижимости для дискретной управляемой системы (1)–(3) на финальный момент времени  $T$  [2, 4].

**Определение 1.** Областью достижимости фазовых состояний управляемой системы (1)–(3) на финальный момент времени  $T$ , соответствующей паре  $(0, \mathbf{X}(0))$ , называется множество, определяемое соотношением

$$\mathbf{G}(0, \mathbf{X}(0); T) = \{x(T) \mid x(T) \in \mathbb{R}^n, x(t+1) = A(t)x(t) + B(t)u(t), t \in \overline{0, T-1}, x(0) \in \mathbf{X}(0), u(t) \in \mathbf{P}\}.$$

В работах А.Ф. Шорикова [1, 2] показано, что область достижимости  $\mathbf{G}(0, \mathbf{X}(0); T)$ , соответствующая определению 1, представляет собой выпуклый, замкнутый и ограниченный многогранник с конечным числом вершин. Для такой области достижимости справедливо следующее рекуррентное (полугрупповое) свойство:

$$\mathbf{G}(0, \mathbf{X}(0); T) = \mathbf{G}(t, \mathbf{G}_+(t); T) \quad \forall t \in \overline{1, T-1},$$

где  $\mathbf{G}_+(t) = \mathbf{G}(0, \mathbf{X}(0); t)$  – область достижимости на момент времени  $t$ , соответствующая паре  $(0, \mathbf{X}(0))$ , которая является выпуклым, замкнутым и ограниченным многогранником с конечным числом вершин.

Тогда задача построения области достижимости  $\mathbf{G}(0, \mathbf{X}(0); T)$  ЛДДС (1)–(3) сводится к рекуррентной процедуре построения одношаговых областей достижимости:

$$\mathbf{G}(t, \mathbf{G}_+(t); t+1), t \in \overline{0, T-1}, \mathbf{G}_+(0) = \mathbf{X}(0). \quad (4)$$

Поскольку указанные области достижимости являются выпуклыми многогранниками, то удобно воспользоваться вершинным описанием этих множеств.

## 2. Алгоритм построения областей достижимости

Далее изложим алгоритм построения областей достижимости (4) ЛДДС в форме псевдокода. В пакет программ построения областей достижимости входит три программы: ReachSets, ConvSM и SimplexHull.

### 2.1. Описание программы ReachSets

Программа ReachSets является основной программой для вычисления вершин областей достижимости ЛДДС (1)–(3).

Входные данные:  $n$  – порядок системы;  $m$  – количество вершин многогранника начальных фазовых состояний;  $k$  – количество вершин многогранника управлений;  $T$  – количество шагов алгоритма;  $A(t) \in \mathbb{R}^{n \times n} \times \overline{0, T-1}$  – матрица системы (задается как трехмерный массив);  $B(t) \in \mathbb{R}^{n \times p} \times \overline{0, T-1}$  – матрица управления (задается как трехмерный массив);  $\widehat{X}(0) \in \mathbb{R}^{n \times m}$  – множество всех вершин многогранника начальных фазовых состояний;  $\widehat{P} \in \mathbb{R}^{p \times k}$  – множество всех вершин многогранника допустимых управлений.

Выходная информация:  $\widehat{G}(t) \subset \mathbb{R}^n \times \overline{0, T}$  – набор матриц, столбцы которых являются вершинами областей достижимости для соответствующих моментов времени.

$$\text{Синтаксис: } \widehat{G} := \text{ReachSets}(n, m, k, T, A(t), B(t), \widehat{X}(0), \widehat{P}).$$

Алгоритм:

```

 $\widehat{G}(0) := \widehat{X}(0);$                                 % присваиваем начальное множество
                                                    % вершин области достижимости
for all  $t$  from 0 to  $T-1$  do

```

```

 $\tilde{G} := \emptyset;$ 
for all  $j$  from 1 to  $m$  do % формируем набор точек в  $\mathbb{R}^n$ 
 $\tilde{G} := \tilde{G} \cup \left[ A(t) \cdot \text{ repmat}(\hat{G}_{[j]}(t), k) + B(t)\hat{P} \right];$ 
end
 $[\hat{G}(t+1), m] := \text{ConvSM}(\tilde{G}, n, k \cdot m);$  % ищем вершины конечного
% множества точек  $\tilde{G}$ 
end

```

Примечание: операция  $\text{ repmat}(x, k)$  превращает вектор  $x$  в матрицу путем его повторения  $k$  раз. Здесь и далее операция объединения « $\cup$ » подразумевает «наращивание» матрицы столбцами справа.

### 2.2. Описание программы ConvSM

ConvSM – основная вспомогательная программа, предназначенная для поиска всех вершин выпуклой оболочки множества достижимости из множества точек  $\tilde{G} \in \mathbb{R}^{n \times m}$ .

Входные данные:  $\tilde{G} \in \mathbb{R}^{n \times m}$  – набор точек, сформированный путем перебора крайних допустимых управлений;  $n$  – порядок системы;  $m$  – количество точек в  $\tilde{G}$ .

Выходная информация:  $\hat{G}$  – набор всех вершин конечного множества точек  $\tilde{G}$ ;  $m$  – количество вершин.

Синтаксис:  $[\hat{G}, m] := \text{ConvSM}(\tilde{G}, n, m)$ .

Алгоритм:

```

 $col := 1;$  % присваиваем переменную столбца
while  $col \leq m$  do
 $[ver, \tilde{G}, l, m] := \text{SimplexHull}(\tilde{G}, col, n, m);$  % проверяем точку
% с индексом  $col$ 
if  $ver == 1$  % проверяем признак вершины
 $col := col + 1;$ 
else
 $col := col + l;$ 
end
end
 $\hat{G} := \tilde{G};$ 

```

### 2.3. Описание программы SimplexHull

Программа SimplexHull является адаптацией модифицированного симплекс-метода (СМ) для решения задачи поиска выпуклой оболочки конечного множества точек в  $\mathbb{R}^n$  [3]. В качестве дополнительных функций в ней фигурирует поиск внутренних точек, которые могут быть охвачены симплексом, опирающимся на базисные векторы, а также поиск вершин, которым соответствуют базисные векторы.

Входные данные:  $\tilde{G}$  – набор точек, сформированный путем перебора крайних допустимых управлений;  $col$  – номер столбца, соответствующий проверяемой точке;  $n$  – порядок системы;  $m$  – количество точек в  $\tilde{G}$ .

Выходная информация:  $\tilde{G}$  – набор точек, с учетом проверки точки с индексом  $col$ ;  $ver$  – признак вершины, если  $\tilde{G}_{[col]}$  вершина, то  $ver = 1$ , иначе –  $ver = 0$ ;  $l$  – количество обнаруженных вершин;  $m$  – количество точек в  $\tilde{G}$ .

Синтаксис:  $[ver, \tilde{G}, l, m] := \text{SimplexHull}(\tilde{G}, col, n, m)$ .

Алгоритм:

$\varepsilon :=$  «установить требуемую точность (например,  $10^{-9}$ )»;

$M := \tilde{G}$ ; % создаем симплекс-таблицу

$M :=$  «все строки матрицы  $M$ , в которых элементы столбца  $M_{[col]}$  отрицательны, умножить на  $-1$ »;

$M^{[n+1]} := E_{1 \times m}$ ;

$M^{[n+2]} :=$  «дополнить матрицу  $M$  строкой, элементы которой равны суммам элементов соответствующих столбцов»;

$A := M$ ; % создаем вспомогательную матрицу

$B := I_{n+2}$ ; % создаем матрицу базисных векторов

$basis := O_{1 \times n+1}$ ; % присваиваем список базисных векторов

**while** 1 **do** % запускаем «бесконечный» цикл

**if**  $A_{[col]}^{[n+2]} \leq \varepsilon$  % проверяем условие остановки СМ

**if**  $A_{[col]}^{[n+2]} \leq \varepsilon$  % проверяем условие вершины

$ver := 0$ ;

$b :=$  «найти ненулевые элементы списка  $basis$ »;

$A :=$  «удалить последнюю строку матрицы  $A$ »;

$A_{[b]} := NaN$ ; % защищаем базисные векторы от удаления

$b :=$  «оставить только элементы, которые меньше  $col$ »;

$X_b := \tilde{G}_{[b]}$ ;

$s :=$  «составить вектор-строку, равную суммам абсолютных значений элементов столбцов матрицы  $A$ »;

$n_s :=$  «найти номера элементов массива  $s$ , которые меньше чем  $1 + \varepsilon$ »;

$\tilde{G} :=$  «удалить из  $\tilde{G}$  столбцы с номерами  $b$  и  $n_s$ »;

$\tilde{G} := X_b \cup \tilde{G}$ ;

$l := \dim(b)$ ;

$m := m - \dim(n_s)$ ;

**else**

$ver := 1$ ;

$\tilde{G} := \tilde{G}$ ;

$l := 0$ ;

$m := m$ ;

**end**

**break**;

**end**

$\tilde{A} := A$ ;

$\tilde{A}_{[col]} := NaN$ ;

$j :=$  «найти номер элемента последней строки матрицы  $\tilde{A}$  с наибольшим значением»;

$x_j := A_{[j]}^{[n+2]}$ ;

$i :=$  «найти номер строки из условия  $\min_{i \in \{1, n+1\}} \frac{A_{[col]}^{[i]}}{x_j^{[i]}}$ ,  $\frac{A_{[col]}^{[i]}}{x_j^{[i]}} \geq 0$ »;

```

B[i] := M[j];           % производим смену базиса
A := B-1M;           % пересчитываем вспомогательную матрицу A
basis[i] := j;       % формируем список базисных векторов

```

end

Примечание: операция  $\dim(x)$  сообщает размерность вектора  $x$ .

### 3. Пример численного моделирования

Продемонстрируем результаты работы оригинального [1] и изложенного алгоритмов на примере следующей динамической системы:

$$x(t+1) = A(t)x(t) + Bu(t), \quad t \in \overline{0, 69}, \quad x(t) \in \mathbb{R}^2, \quad u(t) \in \mathbb{R}^2,$$

$$A(t) = \begin{bmatrix} 0,8877 & -0,012 \\ 0,0258 & 0,4215 \end{bmatrix} \begin{bmatrix} \cos \frac{\pi}{18}t & \sin \frac{\pi}{36}t \\ -\sin \frac{\pi}{36}t & \cos \frac{\pi}{18}t \end{bmatrix}, \quad B = \begin{bmatrix} 1 & 0,5 \\ 0 & 1 \end{bmatrix}, \quad (5)$$

$$|x_1(0)| \leq 5, \quad |x_2(0)| \leq 5, \quad |u_1| \leq 1, \quad |u_2| \leq 1,5.$$

Области достижимости динамической системы (5) изображены на рис. 1.

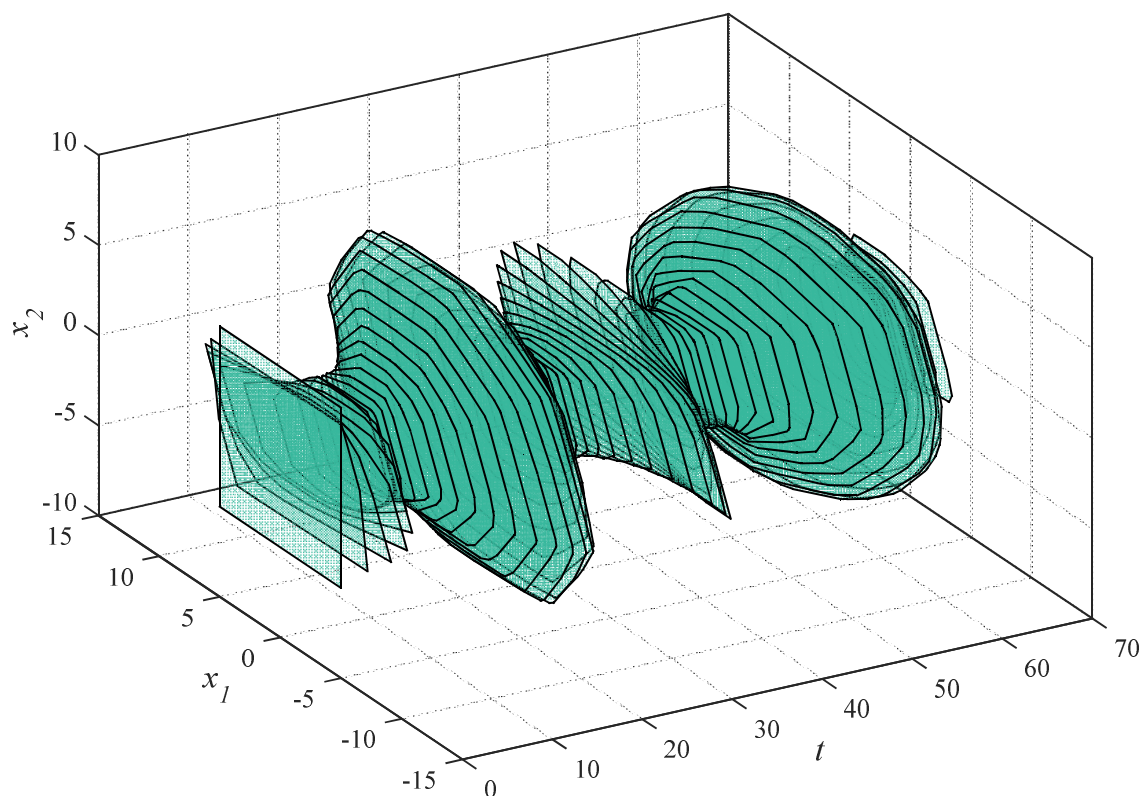


Рис. 1. Области достижимости системы (5)

Моделирование производилось на персональном компьютере Intel(R) Core i7-3770, 3.40 GHz, 8 GB RAM в программном пакете MATLAB 8.4 (R2014b). На рис. 2 приведены графики времени вычислений ( $T$ ) области достижимости и количества вызываемых итераций СМ ( $N_{iter}$ ) на каждом шаге для оригинального (Original) и модифицированного (Modified) алгоритмов.

Построение указанных областей достижимости оригинальным алгоритмом заняло 7,60 с, общее число итераций СМ составило 109 726. Время вычислений модифицированным алгоритмом составило 2,95 с, общее число итераций СМ – 44 993.

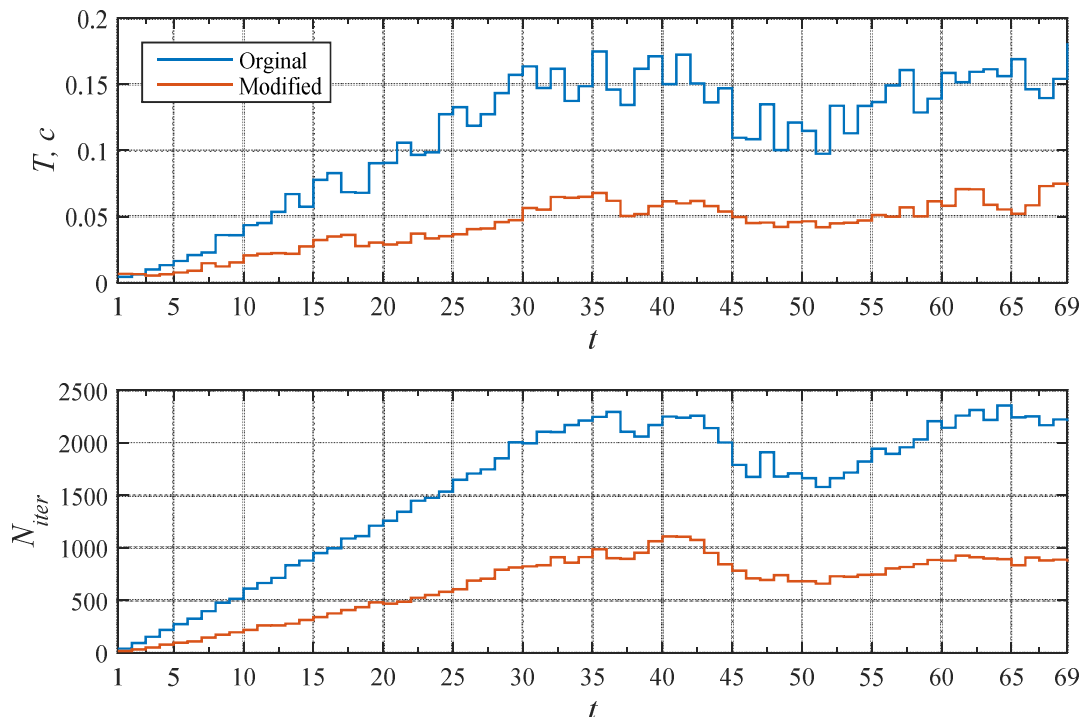


Рис. 2. Сравнительные характеристики алгоритмов

### Заключение

По результатам сравнительного анализа можно сделать вывод, что модифицированный алгоритм построения областей достижимости ЛДДС работает существенно быстрее, чем оригинальный алгоритм [1]. Ускорение быстродействия достигается за счет сокращения числа решаемых задач линейного программирования на этапе поиска выпуклой оболочки конечного множества точек.

Предлагаемый алгоритм построения областей достижимости является сравнительно простым и может играть центральную роль в задаче построения множества допустимых программных управлений ЛДДС. Очевидно, что указанный алгоритм справедлив и для построения обратных областей достижимости при условии существования обратной матрицы системы  $A^{-1}$ .

### Литература

1. Шориков, А.Ф. Минимаксное оценивание и управление в дискретных динамических системах / А.Ф. Шориков. – Екатеринбург: Изд-во Урал. гос. ун-та, 1997. – 242 с.
2. Тюлюкин, В.А. Алгоритм решения задачи терминального управления для линейной дискретной динамической системы / В.А. Тюлюкин, А.Ф. Шориков // Автоматика и телемеханика. – 1993. – № 4. – С. 115–127.
3. Булаев, В.В. Модификация общего рекуррентного алгебраического метода построения областей достижимости линейных дискретных управляемых систем / В.В. Булаев, А.Ф. Шориков. – Тр. Шестой Междунар. науч. конф. «Информационные технологии и системы». – Челябинск: Изд-во Челяб. гос. ун-та, 2017. – С. 47–52.
4. Красовский, Н.Н. Теория управления движением (Линейные системы) / Н.Н. Красовский. – М.: Наука, 1968. – 476 с.

**Булаев Владимир Владимирович**, аспирант кафедры прикладной математики, Уральский федеральный университет им. первого Президента России Б.Н. Ельцина, инженер-конструктор отдела управления движением, АО «НПО Автоматики им. академика Н.А. Семихатова»; г. Екатеринбург; bulaev1991@mail.ru.

**Осипов Иван Олегович**, студент кафедры систем автоматического управления, Южно-Уральский государственный университет, г. Челябинск; ivan\_osipov@live.ru.

*Поступила в редакцию 27 июня 2017 г.*

---

DOI: 10.14529/ctcr180106

## THE MODIFIED ALGORITHM FOR THE LINEAR DISCRETE-TIME DYNAMIC SYSTEMS REACHABLE SETS COMPUTATION

**V.V. Bulaev**<sup>1, 2</sup>, bulaev1991@mail.ru,

**I.O. Osipov**<sup>3</sup>, ivan\_osipov@live.ru

<sup>1</sup> Ural Federal University named after the first President of Russia B.N. Yeltsin, Ekaterinburg, Russian Federation,

<sup>2</sup> JSC "Scientific and Production Association of automatics named after academician N.A. Semikhatov", Ekaterinburg, Russian Federation,

<sup>3</sup> South Ural State University, Chelyabinsk, Russian Federation

The paper considered the modified general recursion algebraic method for the linear discrete-time dynamic systems reachable sets computation. The algorithms of application programs for reachable sets constructing is described. The comparative analysis of original (Shorikov A.F. [Minimax Estimation and Control in Discrete-Time Dynamic Systems]. Yekaterinburg. Ural State University Publ., 1997. 242 p. (in Russ.)) and suggest algorithms is provided by giving of numerical example.

*Keywords: reachable sets, discrete-time dynamic system, convex sets.*

### References

1. Shorikov A.F. *Minimaksnoe otsenivanie i upravlenie v diskretnykh dinamicheskikh sistemakh* [Minimax Estimation and Control in Discrete-Time Dynamic Systems]. Ekaterinburg, Ural State University Publ., 1997. 242 p.
2. Tyulyukin V.A., Shorikov A.F. [The Solution Algorithm of Terminal Control Problem for Linear Discrete-Time System]. *Automatics and Telemekhanics*, 1993, no. 4, pp. 115–127. (in Russ.)
3. Shorikov A.F., Bulaev V.V. [A Modification of the Generalized Recursion Method of the Linear Control Systems Reachable Sets Computation]. *Trudy shestoy mezhdunar. nauch. konf. "Informatsionnye tekhnologii i sistemy"*. [Proc. of 6 International Scientific Conference "Information Technologies and systems" (MMIT 2016)]. Chelyabinsk, Chelyabinsk State Univ. Publ., 2017, pp. 47–52. (in Russ.)
4. Krasovskiy N.N. *Teoriya upravleniya dvizheniem (Lineynie sistemi)* [The Theory of Control of Motion (Linear Systems)]. Moscow, Nauka Publ., 1968. 476 p.

*Received 27 June 2017*

---

### ОБРАЗЕЦ ЦИТИРОВАНИЯ

Булаев, В.В. Модифицированный алгоритм построения областей достижимости линейных дискретных динамических систем / В.В. Булаев, И.О. Осипов // Вестник ЮУрГУ. Серия «Компьютерные технологии, управление, радиоэлектроника». – 2018. – Т. 18, № 1. – С. 56–62. DOI: 10.14529/ctcr180106

### FOR CITATION

Bulaev V.V., Osipov I.O. The Modified Algorithm for the Linear Discrete-Time Dynamic Systems Reachable Sets Computation. *Bulletin of the South Ural State University. Ser. Computer Technologies, Automatic Control, Radio Electronics*, 2018, vol. 18, no. 1, pp. 56–62. (in Russ.) DOI: 10.14529/ctcr180106