

УДК 004.021

**АЛГОРИТМ ОПРЕДЕЛЕНИЯ ПОЛНЫХ ОБЛАСТЕЙ
УСТОЙЧИВОСТИ НЕСТАЦИОНАРНЫХ
НЕЛИНЕЙНЫХ СИСТЕМ**

В.П. Бердников

*Московский технологический университет (МИРЭА), Москва 119454, Россия
@Автор для переписки, e-mail: berdnikov_vp@mail.ru*

В статье предлагается численный алгоритм построения кусочно-линейных функций Ляпунова для исследования абсолютной устойчивости нелинейных нестационарных систем. Такие функции определяют необходимые и достаточные условия устойчивости нелинейных нестационарных систем, удовлетворяющих секторным ограничениям. В случае асимптотической устойчивости системы выполнение алгоритма приведет к построению поверхности функции Ляпунова в виде многогранника размерности, равной размерности исходной системы. Такой многогранник может быть использован для построения кусочно-линейной функции Ляпунова. Количество граней многогранника возрастает по мере приближения системы к границе устойчивости в пространстве параметров, что может приводить к неприемлемым временным затратам на вычисления. В качестве примеров приводятся анализ конкретных систем 2-го и 3-го порядка и результаты сравнения с классическими методами. Даны конкретные рекомендации по выбору начальных условий работы алгоритма.

Ключевые слова: дифференциальные включения, нелинейные нестационарные системы, абсолютная устойчивость, функции Ляпунова, области устойчивости, политопа, радиус кривизны

**ALGORITHM OF DETERMINATION OF NON-STATIONARY
NONLINEAR SYSTEMS FULL STABILITY AREAS**

V.P. Berdnikov

*Moscow Technological University (MIREA), Moscow 119454, Russia
@Corresponding author e-mail: berdnikov_vp@mail.ru*

The paper proposes a numerical algorithm for constructing piecewise linear Lyapunov functions for investigating the absolute stability of nonlinear nonstationary systems. Such functions define necessary and sufficient conditions for the stability of nonlinear

nonstationary systems satisfying sector constraints. In the case of asymptotic stability of the system, the implementation of the algorithm will lead to the construction of the Lyapunov function level set in the form of a polyhedron of dimension equal to the dimension of the original system. Such a polyhedron can be used for constructing a piecewise linear Lyapunov function. The number of faces of the polyhedron increases as the system approaches to the stability boundary in the parameter space, which can lead to unacceptable time costs for calculations. The analysis of specific systems of the 2nd and 3rd order and the results of comparison with classical methods are given. Specific recommendations on the algorithm initial conditions choice are given.

Keywords: differential inclusions, nonlinear nonstationary systems, absolute stability, Lyapunov functions, stability areas, polytope, radius of curvature.

Традиционно для анализа систем с нестационарными нелинейными элементами и систем с параметрической неопределенностью применяют аппарат теории абсолютной устойчивости, например, частотные методы (круговой критерий) [1]. Данные методы являются лишь достаточными, следовательно, они не способны выделить истинную область устойчивости системы в пространстве параметров. Вместе с тем в работах ряда авторов [2, 3] отмечается, что выполнение частотных критериев эквивалентно существованию у системы квадратичной функции Ляпунова. При наличии нескольких нелинейных нестационарных элементов противоположное утверждение неверно, т.е. у системы может существовать квадратичная функция Ляпунова, а частотные критерии могут не выполняться [3]. Таким образом, аппарат функций Ляпунова является более универсальным средством анализа устойчивости систем с секторными нестационарными нелинейными элементами и систем с параметрической неопределенностью.

В связи с этим были предложены различные типы функций Ляпунова, определяющих необходимые и достаточные условия устойчивости подобных систем. К таким функциям относятся кусочно-линейные функции Ляпунова и гладкие выпуклые функции Ляпунова. Вопросы построения кусочно-линейных функций Ляпунова рассматривались в работах [7, 8]. В них вычисляются верхние и нижние оценки показателя Ляпунова, позволяющие сделать заключение об устойчивости системы. Чтобы получить указанные оценки, авторы вышеуказанных публикаций переходят от исходной системы к ее дискретному аналогу.

Цель настоящей работы – разработать метод определения полных областей устойчивости нестационарных нелинейных систем, не использующий показатель Ляпунова, а также не требующий дискретизации исходной системы.

1. Постановка задачи

В работе исследуется устойчивость системы управления, в структуре которой есть несколько (либо один) нелинейных нестационарных элементов. Уравнение такой системы запишем в виде:

$$\frac{dx}{dt} = Ax + \sum_{j=1}^m b^j \varphi_j(\sigma_j, t), \quad (1)$$
$$\sigma_j = (c^j, x) = \sum_{i=1}^d c_i^j x_i, \quad \varphi_j(0, t) \equiv 0$$

где $x = (x_1, x_2, \dots, x_d)^T \in \mathbb{R}^d$ – d -мерный вектор-столбец переменных состояния;

A – постоянная $(d \times d)$ матрица;

b^j и c^j ($j = 1, \dots, m$) – постоянные d -мерные вектор-столбцы;

m – число нестационарных нелинейных элементов.

Предполагается, что нелинейные нестационарные элементы $\varphi_j(\sigma_j, t)$ удовлетворяют секторным ограничениям:

$$\delta_j^1 \leq \frac{\varphi_j(\sigma_j, t)}{\sigma_j} \leq \delta_j^2$$

$$(-\infty < \delta_j^1 \leq \delta_j^2 < \infty, \quad j = 1, \dots, m)$$

при всех σ_j и t . Частным случаем (1) является система, изображенная на рис. 1, где $W(p)$ – передаточная функция.

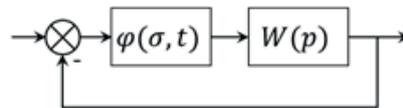


Рис. 1. Структурная схема системы управления.

Как было показано ранее [4–6], проблему устойчивости системы (1) можно свести к вопросу об устойчивости эквивалентного (эквивалентность понимается в смысле совпадения множеств решений при одинаковых начальных условиях) дифференциального включения

$$\frac{dx}{dt} \in F(x), \quad F(x) = \text{conv} \bigcup_{k=1}^N A_k x, \quad (2)$$

где conv – выпуклая оболочка множества;

$\bigcup_{k=1}^N$ – знак объединения;

A_k – квадратные матрицы размера $(d \times d)$.

Дифференциальные включения описывают класс объектов, в которых в каждый момент времени вектор направления движения в фазовом пространстве не определяется некоторым уравнением (в отличие от обычных дифференциальных уравнений), а принадлежит множеству $F(x)$, определяемому правой частью дифференциального включения.

Матрицы A_k для представления системы (1) в виде (2) формируются следующим образом:

$$A_k = A + \sum_{j=1}^m \lambda_j b^j (c^j)^T, \quad k = 1, \dots, 2^m \quad (3)$$

где числа λ_j принимают значения δ_j^1 или δ_j^2 .

Таким образом, общее количество матриц A_k равно 2^m , что соответствует количеству всех возможных комбинаций чисел λ_j . Множество A_k обозначим \mathcal{A} .

В [4–6] для исследования устойчивости дифференциального включения (2) вводится единая кусочно-линейная функция Ляпунова, которую описывают следующей формулой:

$$\mathcal{A}(l, x) = \max_{1 \leq v \leq M} \{ \langle l_v, x \rangle \}, \quad (4)$$

здесь: l – множество векторов l_v , при этом в l входят и сами вектора l_v , и противоположно направленные им вектора $-l_v$;

M – общее количество этих векторов.

Пример такой функции приведен на рис. 2.

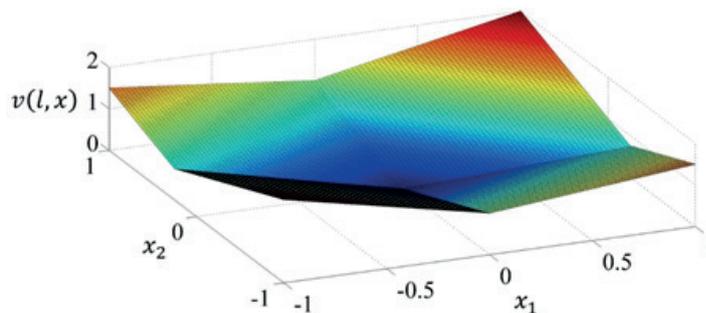


Рис. 2. Кусочно-линейная функция Ляпунова $\square(l, x)$, число граней $M = 4$.

Доказано [4–6], что кусочно-линейные функции определяют необходимые и достаточные условия устойчивости нелинейных нестационарных систем, удовлетворяющих секторным ограничениям. Если система асимптотически устойчива, то можно найти такую функцию (4), что она будет убывать на всех решениях (2). Задачу определения устойчивости дифференциального включения (2), а, значит, и дифференциального уравнения (1), можно рассматривать как задачу построения выпуклого центрально-симметричного многогранника, у которого все вектора $A_k p$ направлены внутрь многогранника, p – любая точка на границе многогранника, $A_k \in \mathcal{A}$.

Действительно, поверхности уровня функции (4) суть выпуклые центрально-симметричные многогранники, l_v являются нормальными к граням этого многогранника, а M определяет число его граней, при условии, что не существует всюду «неактивных» векторов l_v (рис. 3). Отсюда для каждого вектора $l_v \in l$ существует некая точка x : в ней и некоторой ее окрестности $\mathcal{A}(l, x) = \langle l_v, x \rangle$.

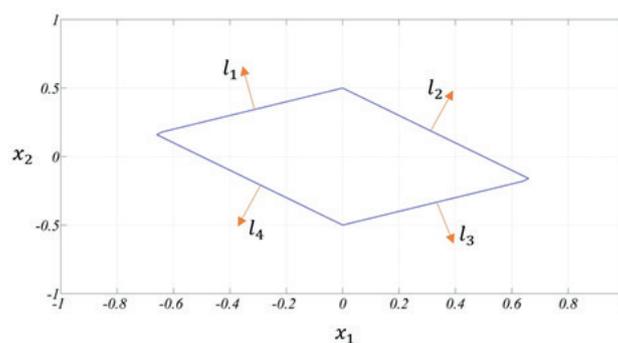


Рис. 3. Поверхность уровня кусочно-линейной функции Ляпунова.

Следует подчеркнуть, что функция (4), в свою очередь, является положительно однородной первой степени, т.е. выполняется следующее условие:

$$\mathcal{A}(l, \tau x) = \tau \mathcal{A}(l, x), \quad (5)$$

где τ – любое положительное действительное число.

Используя свойства скалярного произведения векторов, функции максимума

$$(l, ax) = a(l, x), \max_v \{af_v(x)\} = a \max_v \{f_v(x)\},$$

и (4), непосредственно приходим к уравнению (5).

Таким образом, поверхности уровня кусочно-линейной функции отличаются лишь масштабом и не отличаются по форме. Дифференциальное включение (2) также является однородным. Поэтому если на поверхности уровня кусочно-линейной функции все вектора дифференциального включения (2) направлены внутрь, то они будут направлены внутрь и на всех других поверхностях уровня.

2. Алгоритм «под-над» («beneath-beyond») построения выпуклых многомерных многогранников

Поскольку известный метод построения кусочно-линейных функций Ляпунова основан на конструировании многогранной поверхности уровня, то ниже будут даны некоторые определения, касающиеся многомерных выпуклых многогранников, а также алгоритм их итерационного построения.

Выпуклым многогранником (выпуклым *политопом*) называется выпуклая оболочка конечного числа точек. Выпуклый политоп задается описанием его границы, состоящей из граней. Каждая грань выпуклого политопа является выпуклым множеством (т.е. выпуклым политопом более низкой размерности); k -грань обозначает k -мерную грань (т.е. грань, аффинная оболочка которой имеет размерность k). Если политоп P имеет размерность d , то его $(d-1)$ -границы называются гипергранями, $(d-2)$ -границы – подгранями, 2-границы – ребрами, а 1-границы – вершинами. У трехмерного многогранника (3-политопа) гипергрань – это обычный треугольник, а подгрань одновременно является ребром. Политоп P размерности d называется d -симплексом (или просто *симплексом*), если он является выпуклой оболочкой $(d+1)$ аффинно независимых точек. Для $d = 0, 1, 2,$ и 3 соответствующий симплекс представляет собой точку, отрезок, треугольник и тетраэдр. При этом любая k -грань симплекса – это тоже симплекс размерности k .

Симплициальным политопом называется d -политоп, у которого каждая гипергрань является симплексом. Каждая гипергрань симплициального d -политопа, являющаяся $(d-1)$ -симплексом, определяется в точности d вершинами и содержит d подграней. Ввиду того, что политоп – это ограниченный полиэдр, он может быть задан системой неравенств следующего вида [9]:

$$(l_v, x) + b_v \leq 0, \quad 1 \leq v \leq M. \quad (6)$$

В этом случае уравнение $(l_v, x) + b_v = 0$ является аффинной оболочкой соответствующей гиперграней, гиперплоскостью, в которой лежит гипергрань, а l_v – нормаль к этой гиперграней, направленная в сторону от многогранника (рис. 4). Выпуклый политоп всегда полностью лежит в одном полупространстве, определяемом аффинной оболочкой его гиперграней.

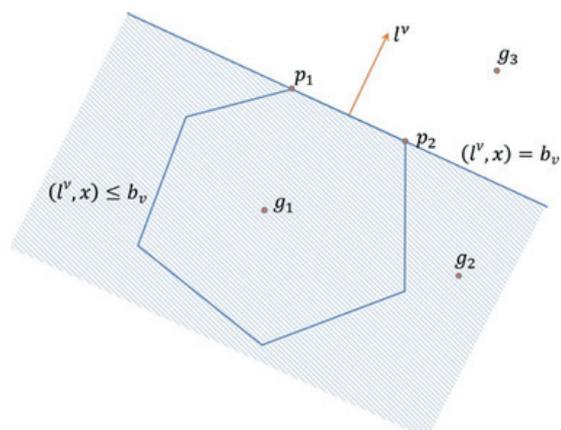


Рис. 4. Аффинная оболочка грани многогранника.

Известно, что через $k+1$ аффинно независимых точки p_1, p_2, \dots, p_{k+1} проходит единственная гиперплоскость размерности k . Так, в d -мерном пространстве значение $(l_v, x) + b_v$, где $(l_v, x) + b_v = 0$ – уравнение гиперплоскости, проходящей через точки p_1, p_2, \dots, p_{k+1} , вектор l_v – нормаль к гиперплоскости может быть вычислена по следующей формуле:

$$f_v(x) = \det \begin{pmatrix} x^1 & x^2 & \dots & x^d & 1 \\ p_1^1 & p_1^2 & \dots & p_1^d & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ p_d^1 & p_d^2 & \dots & p_d^d & 1 \end{pmatrix}, \quad p_k = (p_k^1, p_k^2, \dots, p_k^d). \quad (7)$$

Для вычисления i -ой компоненты вектора l_v следует в матрице (7) в первой строке i -ый элемент заменить на единицу, а все остальные – на ноль. Для вычисления b_v необходимо все элементы первой строки, кроме последнего, заменить на ноль. Следовательно, чтобы найти вектор l_v и число b_v по заданным точкам, требуется вычислить $(d+1)$ определитель следующего вида

$$l_v^1 = \det \begin{pmatrix} 1 & 0 & \dots & 0 & 0 \\ p_1^1 & p_1^2 & \dots & p_1^d & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ p_d^1 & p_d^2 & \dots & p_d^d & 1 \end{pmatrix}, \quad l_v^2 = \det \begin{pmatrix} 0 & 1 & \dots & 0 & 0 \\ p_1^1 & p_1^2 & \dots & p_1^d & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ p_d^1 & p_d^2 & \dots & p_d^d & 1 \end{pmatrix}, \quad (8)$$

$$l_v^d = \det \begin{pmatrix} 0 & 0 & \dots & 1 & 0 \\ p_1^1 & p_1^2 & \dots & p_1^d & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ p_d^1 & p_d^2 & \dots & p_d^d & 1 \end{pmatrix}, \quad b_v = \det \begin{pmatrix} 0 & 0 & \dots & 0 & 1 \\ p_1^1 & p_1^2 & \dots & p_1^d & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ p_d^1 & p_d^2 & \dots & p_d^d & 1 \end{pmatrix},$$

$$l_v = (l_v^1, l_v^2, \dots, l_v^d).$$

Вычисленная таким образом нормаль к гиперграни многогранника может быть направлена как внутрь многогранника, так и вовне. Если при подстановке в $(l_v, x) + b_v$, вместо

x некоторой внутренней точки многогранника (g_1 на рис. 4) получается отрицательное число, то грань направлена во внешнюю сторону, если положительное – то во внутреннюю. В последнем случае следует l_v и b_v домножить на -1 , чтобы многогранник мог быть представлен в виде (6).

Ниже приведена основная процедура итерационного алгоритма «под-над» («beneath-beyond») построения выпуклой оболочки множества точек [10]. Будем утверждать, что точка p лежит «под» гранью F_v , если $(l_v, x) + b_v < 0$, т.е. p лежит в том же полупространстве, что и многогранник P (точки g_1 и g_2 на рис. 4), а также, что точка p лежит «над» гранью F_v , если $(l_v, x) + b_v \geq 0$. Для описания алгоритма потребуется следующая теорема.

Теорема [11]. В симплицальном политопе некоторая подгрань является общей точно для двух гиперграней, и две гиперграни F_1 и F_2 имеют общую подгрань e тогда и только тогда, когда e определяется общим подмножеством из $(d-1)$ вершины для множеств вершин, определяемых F_1 и F_2 (в этом случае F_1 и F_2 называются смежными по e).

Алгоритм «под-над» («beneath-beyond»). Добавление точки p к многограннику P .

Шаг 1. Определить множество S гиперграней F_v многогранника P , «над» которыми находится точка p . Если точка p лежит «под» всеми гипергранями многогранника (множество S пусто), то она является внутренней точкой, и перестраивать многогранник не требуется. Такие гиперграни называются «видимыми» из точки p (рис 5; на рис. 5а множество S состоит из F_1 и F_2).

Шаг 2. Определить границу множества S , т.е. определить множество T подграней e_j , которые не являются общими для любых пар гиперграней из S . Это означает, что для видимой гиперграней F_1 смежная с ней по $e_1 \in T$ грань F_2 не является видимой. На рис. 5б $T = \{e_1, e_2, e_3, e_4\}$, а e_5 является общей подгранью двух видимых смежных гиперграней.

Шаг 3. Построить конус новых гиперграней из точки p и подграней множества T (рис. 5в). Так как в симплицальном многограннике подгрань – это симплекс размерности $d-2$, составленный из $d-1$ точек, то добавляя к ней одну точку, можно построить симплекс размерности $d-1$, т.е. гипергрань многогранника.

Шаг 4. Удалить множество S старых гиперграней F_v . После добавления внешней точки p гиперграни из S оказываются внутри многогранника (рис. 5г).

Алгоритм «под-над», позволяющий итерационно строить многогранники произвольной размерности путем последовательного добавления к ним точек, создает базу, на которой могут быть созданы итерационные алгоритмы построения кусочно-линейных функций Ляпунова (4). Заметим также, что самым затратным – с вычислительной точки зрения – является шаг 1, так как, возможно, потребуется перебрать все грани многогранника прежде, чем будет сформировано множество S .

3. Алгоритм построения поверхности уровня функции Ляпунова

Как уже отмечалось выше, условие устойчивости системы (1) с геометрической точки зрения означает существование такого выпуклого центрально-симметричного многогранника P , у которого из каждой точки v его границы векторы $A_k v$ направлены во внутреннюю часть. Аналитически это условие можно записать следующим способом:

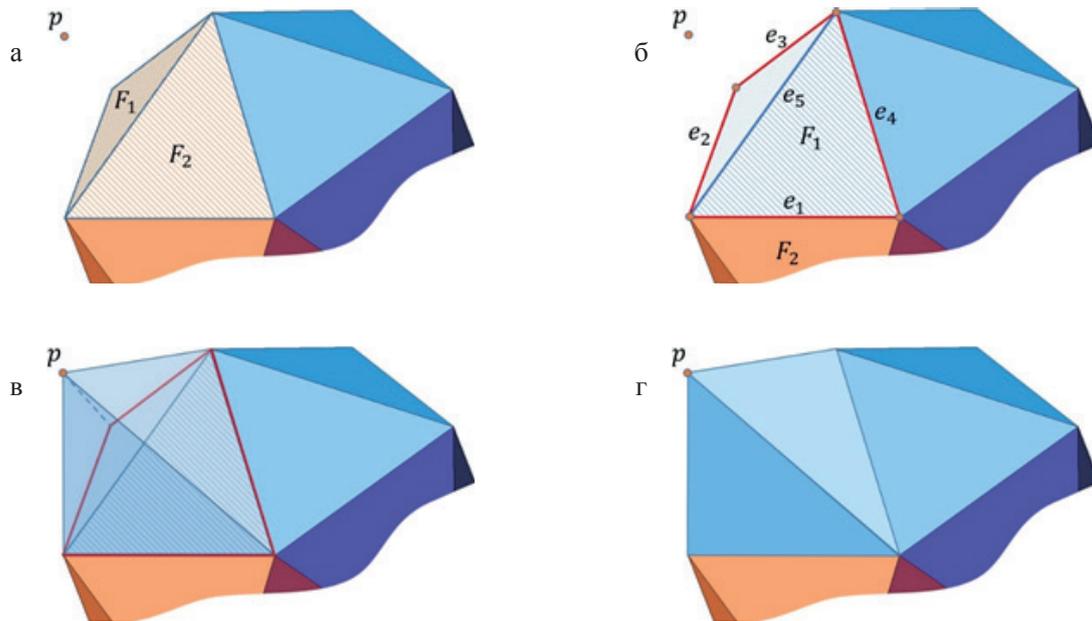


Рис. 5. Иллюстрация работы алгоритма добавления точки к политопу.

$$(A_k v, n) = \|A_k v\| \|v\| \cos \varphi < 0. \quad (9)$$

Здесь $\| \cdot \|$ обозначает норму вектора. Тогда для выполнения неравенства (9) необходимо, чтобы угол между векторами $A_k v$ и n был больше $\pi/2$. Важно заметить, что гиперграницы выпуклого d -политопа представляют собой выпуклые многогранники размерности $(d-2)$ [12]. Так как $(A_k v, n)$ – линейный по v функционал, а линейный функционал на выпуклом многогранном множестве достигает максимума в его вершинах [13], то проверку (8) на устойчивость системы (1) можно проводить только в вершинах многогранника P .

Учитывая вышесказанное, новые точки в предлагаемом итерационном алгоритме построения многогранной поверхности уровня кусочно-линейной функции Ляпунова нужно добавлять в направлении $A_k v_1 / \|A_k v_1\|$ на расстоянии h . В этом случае новая гипергрань будет содержать отрезок, соединяющий точку v_1 и точку $v_1 + h A_k v_1 / \|A_k v_1\|$, а это значит, что угол между нормалью n гиперграницы и вектором $A_k v_1$ в вершине v_1 будет равен $\pi/2$ (рис. 6).

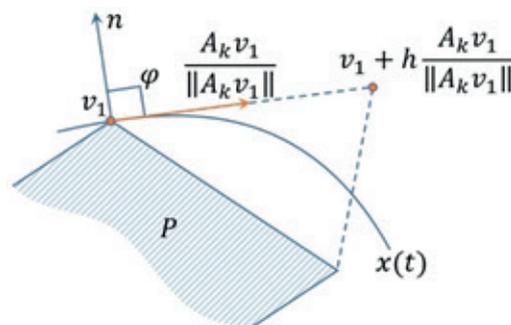


Рис. 6. Построение многогранника с использованием направления $A_k v_1 / \|A_k v_1\|$.

Действительно, так как компоненты нормали определяются формулами (8), а в состав гиперграницы входят точки v_1 и $v_1 + h A_k v_1 / \|A_k v_1\|$, то условие (9) можно переписать так:

$$(A_k v_1, n) = \det \begin{pmatrix} (A_k v_1)^T & 0 \\ (v_1)^T & 1 \\ (v_1 + h A_k v_1 / \|A_k v_1\|)^T & 1 \\ \vdots & \vdots \\ (v_d)^T & 1 \end{pmatrix}, \quad (v_i)^T = (v_i^1, \dots, v_i^d).$$

Очевидно, что третья строка является линейной комбинацией первых двух, то есть определитель равен нулю, что, в свою очередь, дает $\varphi = \pi/2$.

Следовательно, остается вопрос выбора шага h . В данной статье величину шага предлагается вычислять пропорционально кривизне решения $x(t) = e^{At} x_0$ (A – матричная экспонента, матрица, зависящая от t , которая используется для универсального решения линейного дифференциального уравнения). Из дифференциальной геометрии [14] известно, что радиус кривизны в многомерном пространстве определяется соотношением:

$$r = \frac{|\gamma(t)|^3}{\sqrt{|\dot{\gamma}(t)|^2 |\ddot{\gamma}(t)|^2 - (\dot{\gamma}(t), \ddot{\gamma}(t))^2}},$$

где $\gamma(t)$ – параметрическая кривая.

В случае $\gamma(t) = x(t)$, т.е. когда надо найти радиус кривизны фазовой траектории линейной системы, $\dot{x} = A_k x$

$$\begin{aligned} \dot{\gamma}(t) &= \dot{x}(t) = A_k x(t), \\ \ddot{\gamma}(t) &= \ddot{x}(t) = A_k^2 x(t). \end{aligned}$$

Таким образом, в каждой точке фазового пространства оказывается возможным найти радиус кривизны соответствующей траектории $x(t) = e^{A_k t} x_0$. Найдя среди всех вершин некоторого многогранника такую вершину v_1 , у которой значение $(A_k v_1, n)$ максимально для всех $A_k \in \mathbb{A}$ и n (под n понимается нормали к граням, содержащим x_0), можно получить новую точку:

$$v_2 = v_1 + \frac{A_k v_1}{\|A_k v_1\|} h, \quad h = \varepsilon r,$$

где ε – заранее заданное положительное число, параметр алгоритма, выбираемый до начала его работы. Окончательно получаем формулу расчета новых точек для многогранной поверхности уровня функции Ляпунова:

$$v_2 = v_1 + \varepsilon \frac{A_k v_1 (A_k v_1, A_k v_1)}{\sqrt{(A_k v_1, A_k v_1) (A_k^2 v_1, A_k^2 v_1) - (A_k v_1, A_k^2 v_1)^2}}. \tag{10}$$

На основании приведенных уравнений предлагается следующий алгоритм построения поверхности уровня функции Ляпунова (4):

Шаг 1. Задается выпуклый центрально-симметричный многогранник P_0 с непустой внутренностью, а также положительное число ε .

Шаг 2. Для вершины v_1 многогранника P_i (i – номер итерации, алгоритм начинается с $i = 0$), у которой значение $(A_k v_1, n)$ максимально, вычисляется точка v_2 по формуле (10).

Шаг 3. С помощью алгоритма «под-над» строится выпуклая оболочка P_{i+1} многогранника P_i и точки v_2 . Если $P_{i+1} \neq P_i$, то начинается новая итерация и шаги 2–4 повторяются. Если на какой-то итерации $P_{i+1} = P_i$, то процедура прекращается, и полученный многогранник является поверхностью уровня функции Ляпунова (4).

Шаг 4. Если количество итераций i превышает некоторый заранее установленный предел N , то это свидетельствует либо о том, что система неустойчива, либо о том, что она находится вблизи границы устойчивости. Тогда следует уменьшить ε и повторить алгоритм сначала. Если алгоритм не останавливается на шаге 3 при любом ε , то система (1) неустойчива.

Подчеркнем, что при уменьшении ε затрачивается больше вычислительных ресурсов, и растет время выполнения алгоритма. Если процедура расчета по алгоритму завершилась успешно, то по полученному многограннику можно восстановить функцию (4), для чего необходимо в качестве l_v взять нормали к граням многогранника, деленные на b_v . На шаге 3, когда требуется построить выпуклую оболочку новой точки v_2 и многогранника P_i , используется алгоритм «под-над».

Как отмечалось выше, самым трудоемким (с вычислительной точки зрения) в этом алгоритме является процесс поиска первой «видимой» грани [10, 11]. Однако в нашем случае для поиска первой «видимой» грани достаточно проверить только грани, содержащие в своем составе v_1 из (10). Если же окажется, что v_2 «под» всеми этими гранями, то добавлять точку v_2 не нужно. Вопросы применения алгоритма построения поверхности уровня функции Ляпунова (4) к конкретным системам рассмотрены ниже.

Пример 1. На рис. 1 дана структурная схема системы управления. Передаточная функция линейной части системы

$$W(p) = \frac{1}{p^2 + 0.5p + 1}.$$

Нелинейный нестационарный элемент $\varphi(\sigma, t)$ удовлетворяет секторным ограничениям.

$$0 \leq \frac{\varphi(\sigma, t)}{\sigma} \leq \delta.$$

Определим граничный коэффициент δ_{sp} такой, чтобы система была устойчива при любом $\delta < \delta_{sp}$ и неустойчива при $\delta \geq \delta_{sp}$.

Чтобы решить эту задачу, преобразуем систему на рис. 1 к виду (2). Линейную часть $W(p)$ представим в виде:

$$\dot{x} = Ax + bu,$$

$$A = \begin{pmatrix} 0 & 1 \\ -1 & -0.5 \end{pmatrix}, \quad x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, \quad b = \begin{pmatrix} 0 \\ 1 \end{pmatrix}. \quad (11)$$

Переменная u является входом, подаваемым на линейный блок. Как видно из структурной схемы (рис. 1), в данном случае $u = \varphi(\sigma, t)$. Остается заметить, что вход нелинейного элемента записывается следующим образом:

$$\sigma = (c, x), \quad c = \begin{pmatrix} -1 \\ 0 \end{pmatrix}. \quad (12)$$

Следовательно, данная система описывается уравнениями вида (1). Поскольку в ней лишь один нелинейный нестационарный блок ($m = 1$), то количество входящих в (2) матриц равно двум, и λ_1 из (3) принимает значения $\delta_1^1 = 0$, $\delta_1^2 = \delta$, которые являются границами сектора нестационарной нелинейности $\varphi(\sigma, t)$.

$$A_1 = A + \lambda_1 bc^T = \begin{pmatrix} 0 & 1 \\ -1 & -0.5 \end{pmatrix}, \quad (13)$$

$$A_2 = A + \lambda_1 bc^T = \begin{pmatrix} 0 & 1 \\ -(1 + \delta) & -0.5 \end{pmatrix}.$$

Сформированные матрицы позволяют непосредственно применить алгоритм для анализа устойчивости исходной системы. Чтобы продемонстрировать эффективность предлагаемого алгоритма, найдем наибольшее δ , при котором работают классические методы. При использовании кругового частотного критерия имеем $\delta_{\text{чк}} \approx 1.25$.

Для определения $\delta_{\text{зр}}$ с помощью алгоритма начнем постепенно увеличивать δ , начиная с малых значений, до тех пор, пока алгоритм будет строить поверхность уровня функции Ляпунова. Каждому значению δ необходимо подобрать такой коэффициент ε из (10), при котором может быть успешно построена поверхность уровня функции Ляпунова (4). Результаты численных экспериментов приведены в табл. 1. На рис. 7 представлены соответствующие первому и последнему экспериментам поверхности уровня функций Ляпунова.

Увеличение значения δ требует уменьшения параметра ε , а это, в свою очередь, ведет к росту числа граней поверхности уровня Ляпунова (в данном случае многоугольника, рис. 7). Уже при значении $\delta = 2.2$ не удастся подобрать такое даже очень малое ε , позволяющее построить функцию Ляпунова, что может свидетельствовать о близости к границе устойчивости при таком значении δ . При приближении системы к границе устойчивости количество граней начинает быстро расти. Из табл. 1 видно, что при $\delta = 1.1$ количество граней равно 38, а при $\delta = 2.15$ количество граней уже составляет 1420. Понятно, что при этом требуется больше времени. В первом эксперименте (табл. 1) на построение многогранника

Таблица 1. Количество граней при различных размерах секторных ограничений для системы второго порядка

№ п/п	Количество граней	δ	ε	Затраченное время, с
1	38	1.1	1.190	0.072
2	46	1.25	0.160	0.086
3	56	1.4	0.130	0.109
4	74	1.55	0.095	0.121
5	96	1.7	0.075	0.148
6	144	1.85	0.050	0.207
7	282	2.0	0.025	0.423
8	1420	2.15	0.005	4.655

уходит всего 0.072 с, тогда как в восьмом эксперименте затраченное время составляет уже 4.655 с, но даже такое время является вполне достаточным и позволяет рассматривать возможность применения алгоритма для систем малой размерности в тех случаях, когда требуется рассчитывать устойчивость системы в режиме реального времени.

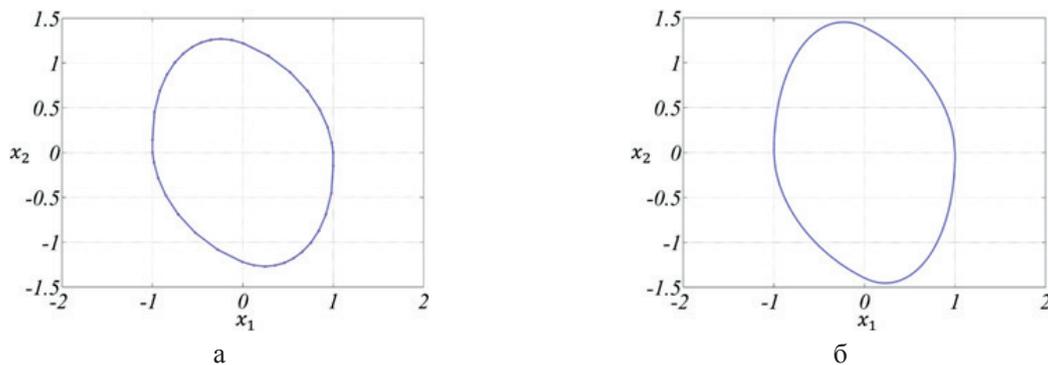


Рис. 7. Поверхности функций Ляпунова: а) $\delta = 1.1$, $\varepsilon = 1.190$; б) $\delta = 1.25$, $\varepsilon = 0.005$.

Наибольшее значение δ , для которого удается построить многогранную поверхность уровня функции Ляпунова (4), равно 2.15, т.е. $\delta_{sp} = 2.15$. Сопоставление с классическими методами, которые дают максимальное δ_{chk} , равное 1.25 [7], убедительно показывает, что предложенный алгоритм позволяет существенно (в 1.7 раза) улучшить оценку истинной границы устойчивости. Это может иметь чрезвычайно важное значение, в частности, в системах с высокой степенью управляемости, имеющих ограниченную область устойчивости. Кроме того, при проектировании нечетких регуляторов на первом этапе синтеза определяют максимальный размер сектора, в котором должна лежать нелинейная характеристика нечеткого регулятора [15]. Так как предложенный алгоритм увеличивает границы секторов по сравнению с классическими методами, то появляется возможность обоснованно синтезировать более точные и быстродействующие САУ на основе нелинейных (нечетких) законов управления.

В примере 1 описана система второго порядка. Рассмотрим систему третьего порядка ($d = 3$) и установим, как влияет увеличение порядка системы на время работы алгоритма. В этом случае поверхность уровня функции Ляпунова (4) будет уже не многоугольником, а многогранником в трехмерном пространстве.

Пример 2. Линейная часть системы описывается передаточной функцией

$$W(p) = \frac{p+1}{p^3 + 1.5p^2 + 3p + 1}.$$

В структурной схеме (рис. 1) по-прежнему считаем, что нелинейный нестационарный элемент удовлетворяет секторным ограничениям

$$0 \leq \frac{\varphi(\sigma, t)}{\sigma} \leq \delta.$$

Максимальное значение δ , для которого выполняется круговой частотный критерий, равно $k_{чк} \approx 2.28$ [16]. Воспользуемся формулой (3).

$$A_1 = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -3 & -4 & -1.5 \end{pmatrix}$$

$$A_2 = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -(3+\delta) & -(4+\delta) & -1.5 \end{pmatrix}$$

Составим таблицу численных экспериментов при различных значениях δ (табл. 2). Как и в примере 1, при увеличении δ требуется уменьшать ε , что приводит к увеличению количества граней и, соответственно, времени расчета. Если при $\delta = 1$ на построение поверхности уровня из 372 граней требуется 1.335 с, то при $\delta = 3$ затрачивается уже 27.402 с, что значительно больше, чем для системы второго порядка, также находящейся у границы устойчивости. При $\delta = 3.5$ необходимо брать ε настолько малым, что время работы алгоритма начинает измеряться в минутах.

Таблица 2. Количество граней при различных размерах секторных ограничений для системы третьего порядка

№ п/п	Количество граней	δ	ε	Затраченное время, с
1	372	1.0	0.220	1.335
2	636	1.5	0.175	2.305
3	760	2.0	0.135	8.272
4	1644	2.5	0.095	9.729
5	2936	3.0	0.065	27.402

На рис. 8 приведена поверхность уровня функции Ляпунова для случая эксперимента № 1 и эксперимента № 2.

Из приведенных расчетов видно, что разработанный алгоритм позволяет определить максимальное значение δ , равное 3, что больше $k_{чк}$, которое было определено с помощью классического метода. Анализ поверхностей, изображенных на рис. 7 и 8 позволяет сделать вывод о достаточно сложной форме поверхности функции Ляпунова системы (2),

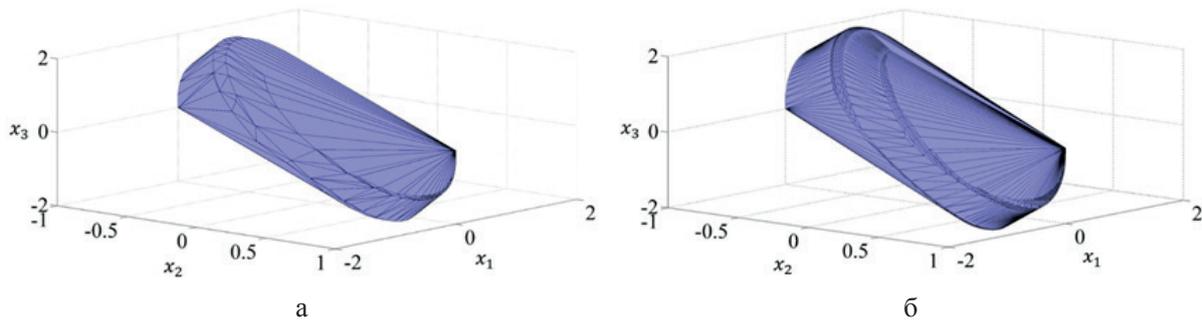


Рис. 8. Поверхность уровня функции Ляпунова: а) $\delta = 1, \varepsilon = 0.220$; б) $\delta = 3, \varepsilon = 0.065$.

находящейся близко к границе устойчивости. Очевидно, что такая форма не может быть аппроксимирована поверхностью уровня квадратичной функции Ляпунова (эллипсоидом соответствующей размерности), что в известном смысле и объясняет достаточность критериев устойчивости, основанных на квадратичных функции Ляпунова и частотных методах.

4. Анализ влияния начального многогранника на время работы разработанного алгоритма

В примерах 1 и 2 предполагалось, что на первом шаге работы в качестве начального многогранника выбирается выпуклая оболочка точек

$$\begin{aligned} p_1 &= (1, 0, \dots, 0), -p_1, \\ p_2 &= (0, 1, \dots, 0), -p_2, \\ &\dots \\ p_d &= (0, 0, \dots, 1), -p_d, \end{aligned}$$

где d – размерность пространства.

Общее количество точек равно $2d$, в случае $d = 2$ начальный многогранник – это ромб, при $d = 3$ – октаэдр. Найдем, как задание начального многогранника влияет на время работы алгоритма и количество граней в построенном многограннике на границе устойчивости. Для этого рассмотрим систему из первого примера с $\delta = 2.15$. Будем последовательно задавать различные выпуклые центрально-симметричные четырехугольники и далее применять остальные шаги алгоритма.

На рис. 9 приведены временные затраты при задании 100 вариантов многогранников.

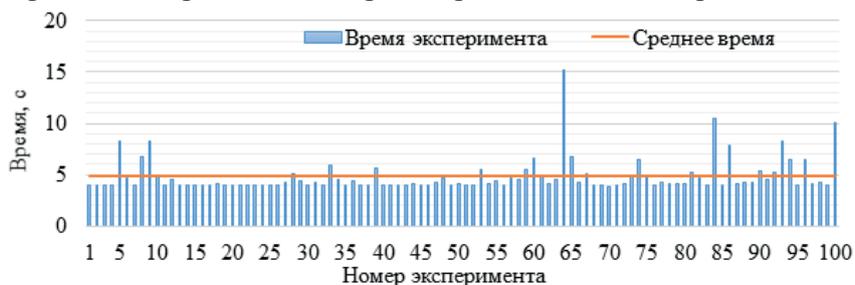


Рис. 9. Время, затрачиваемое алгоритмом, при случайном задании начального многогранника для системы второго порядка.

Из диаграммы на рис. 9 можно сделать вывод, что при определенных конфигурациях начального многогранника время работы алгоритма может существенно увеличиваться и

превышать среднее время расчета более, чем в два раза. Следовательно, корректный выбор начального многогранника может существенно сократить время работы алгоритма на границе устойчивости.

Ниже приведены данные о количестве граней построенных в этих экспериментах многогранников (рис. 10).



Рис. 10. Количество граней построенных многогранников для системы второго порядка.

Как видно, количество граней практически не зависит от конфигурации начального многогранника и колеблется в диапазоне 1342–1400. Для практического применения алгоритма предлагается задавать начальный многогранник как выпуклую оболочку собственных векторов (f_i и $-f_i$) одной из матриц A_k , входящих в (3). Так, если в качестве начального многогранника использовать собственные вектора матрицы A_1 , то затраченное время будет равняться 3.923 с, если матрицы A_2 , то 3.946 с.

Проведем такой же эксперимент для системы третьего порядка из примера 2. Диаграмма времени, затрачиваемого на работу алгоритма, приведена на рис. 11.

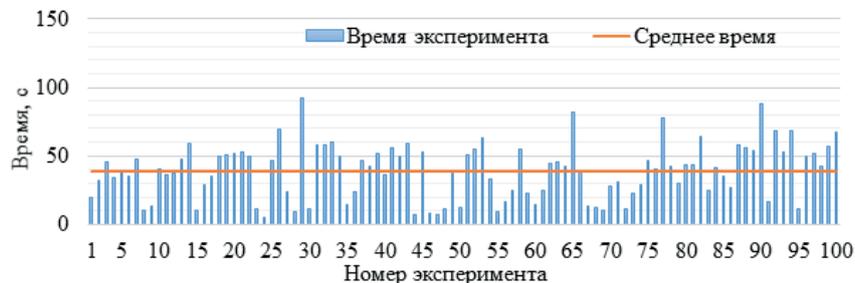


Рис. 11. Время, затрачиваемое алгоритмом при случайном задании начального многогранника, для системы третьего порядка.

Очевидно (рис. 11), что разница между минимальным и максимальным временем выполнения алгоритма достигает 90 с, т.е. корректный выбор начального многогранника имеет большое значение.

На рис. 12 показано количество граней построенных многогранников.

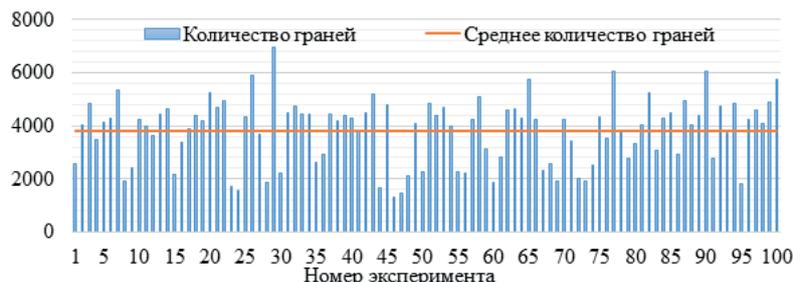


Рис. 12. Количество граней построенных многогранников для системы третьего порядка.

В отличие от системы второго порядка, у системы третьего порядка наблюдается значительный разброс количества граней в различных экспериментах.

Теперь в качестве начальных многогранников зададим выпуклые оболочки собственных векторов матриц A_1 и A_2 . Для матрицы A_1 время работы алгоритма равно 6.165 с, а для матрицы A_2 3.946 с, что значительно меньше среднего времени работы алгоритма (примерно 40 с).

Из полученных экспериментальных данных следует, что задание в качестве начального многогранника выпуклой оболочки собственных векторов (и противоположно-направленных) одной из матриц A_k существенно экономит время работы алгоритма.

Заключение

В работе получен алгоритм, обеспечивающий необходимые и достаточные условия устойчивости нелинейных нестационарных систем. В отличие от частотных методов, он позволяет выявлять полные области устойчивости в пространстве параметров. Даны рекомендации по выбору начальной конфигурации многогранника. Показано, что в качестве начального многогранника следует выбирать выпуклую оболочку обобщенных собственных векторов одной из матриц системы. Конкретные численные примеры подтверждают подобный выбор, который позволяет существенно снизить время работы предложенного нами алгоритма по сравнению с произвольным заданием начального многогранника.

Вместе с тем сложность алгоритма (и, соответственно, время расчета) возрастает по мере роста порядка объекта, поэтому необходимо искать пути повышения быстродействия алгоритма в режиме реального времени. Одним из возможных путей является разработка алгоритма построения гладких функций Ляпунова, определяющих необходимые и достаточные условия устойчивости нелинейных нестационарных систем.

Работа выполнена при поддержке гранта РФФИ (проект №16-19-00052).

Литература:

1. Барабанов А.Т., Катковник В.Я., Нелепин Р.А., Хлыпало Е.И., Якубович В.А. Методы исследования нелинейных систем автоматического управления / Под ред. Р.А. Нелепина. М.: Наука, 1975. 448 с.
2. Якубович В.А. Абсолютная неустойчивость нелинейных систем управления. II. Системы с нестационарными нелинейностями. Круговой критерий // Автоматика и телемеханика. 1971. № 6. С. 25–34.
3. Каменецкий В.А. Абсолютная устойчивость и абсолютная неустойчивость систем управления с несколькими нелинейными нестационарными элементами // Автоматика и телемеханика. 1983. № 12. С. 20–30.
4. Молчанов А.П., Пятницкий Е.С. Функции Ляпунова, определяющие необходимые и достаточные условия абсолютной устойчивости нелинейных нестационарных систем управления // Автоматика и телемеханика. 1986. № 3. С. 63–73.
5. Молчанов А.П., Пятницкий Е.С. Функции Ляпунова, определяющие необходимые и достаточные условия абсолютной устойчивости нелинейных нестационарных систем управления II // Автоматика и телемеханика. 1986. № 4. С. 5–15.
6. Молчанов А.П., Пятницкий Е.С. Функции Ляпунова, определяющие необходимые

и достаточные условия абсолютной устойчивости нелинейных нестационарных систем управления III // Автоматика и телемеханика. 1986. № 5. С. 38–49.

7. Барабанов Н.Е. Метод вычисления показателя Ляпунова дифференциального включения // Автоматика и телемеханика. 1989. № 4. С. 53–58.

8. Guglielmi N., Laglia L., Protasov V. Polytope Lyapunov functions for stable and for stabilizable LSS // Foundations of Computational Mathematics. 2017. V. 17. № 2. P. 567–623.

9. Емеличев В.А., Ковалев М.М., Кравцов М.К. Многогранники, графы, оптимизация. М.: Наука, 1981. 344 с.

10. Barber C.B., Dobkin D.P., Huhdanpaa H. The Quickhull algorithm for convex hulls // ACM Transactions on Mathematical Software. 1995. V. 22. Iss. 4. P. 469–483.

11. Препарата Ф., Шеймос М. Вычислительная геометрия: Введение. М.: Мир, 1989. 478 с.

12. Циглер Г.М. Теория многогранников. М.: МЦНМО, 2014. 568 с.

13. Еремин И.И., Астафьев Н.Н. Введение в теорию линейного и выпуклого программирования. М.: Наука, 1976. 192 с.

14. Аминов Ю.А. Дифференциальная геометрия и топология кривых. М.: Наука, 1987. 160 с.

15. Макаров И.М., Лохин В.М., Манько С.В., Романов М.П. Интеллектуальные системы автоматического управления / Под ред. Макарова И.М., Лохина В.М. М.: Физматлит, 2001. 576 с.

16. Баркин А.И. Абсолютная устойчивость систем управления. М.: Либроком, 2012. 176 с.

References:

1. Barabanov A.T., Katkovnik V.Ya., Nelepin R.A., Hlypalo E.I., Yakubovich V.A. Methods for investigating nonlinear automatic control systems / Ed. R.A. Nelepin. Moscow: Nauka Publ., 1975. 448 p. (in Russ.).

2. Yakubovich V.A. Absolute instability of nonlinear control systems. II. Systems with nonstationary nonlinearities. Circle criteria // Avtomatika i telemekhanika. (Automation and Remote Control). 1971. № 6. P. 25–34. (in Russ.).

3. Kameneckij V.A. Absolute stability and absolute instability of control systems with several nonlinear nonstationary elements // Avtomatika i telemekhanika. (Automation and Remote Control). 1983. № 12. P. 20–30. (in Russ.).

4. Molchanov A.P., Pyatnickij E.S. Lyapunov functions that determine necessary and sufficient conditions for absolute stability of nonlinear time-varying control systems. I // Avtomatika i telemekhanika. (Automation and Remote Control). 1986. № 3. P. 63–73. (in Russ.).

5. Molchanov A.P., Pyatnickij E.S. Lyapunov functions that determine necessary and sufficient conditions for absolute stability of nonlinear time-varying control systems. II // Avtomatika i telemekhanika. (Automation and Remote Control). 1986. № 4. P. 5–15. (in Russ.).

6. Molchanov A.P., Pyatnickij E.S. Lyapunov functions that determine necessary and sufficient conditions for absolute stability of nonlinear time-varying control systems. III // Avtomatika i telemekhanika. (Automation and Remote Control). 1986. № 5. P. 38–49. (in Russ.).

7. Barabanov N.E. The method of calculating the Lyapunov exponent of a differential

inclusion // *Avtomatica i telemehanica*. (Automation and Remote Control). 1989. № 4. P. 53–58. (in Russ.).

8. Guglielmi N., Laglia L., Protasov V. Polytope Lyapunov functions for stable and for stabilizable LSS // *Foundations of Computational Mathematics*. 2017. V. 17. № 2. P. 567–623.

9. Emelichev V.A., Kovalyov M.M., Kravcov M.K. Polyhedra, graphs, optimization. Moscow: Nauka Publ., 1981. 344 p. (in Russ.).

10. Barber C. B., Dobkin D. P., Huhdanpaa H. The Quickhull algorithm for convex hulls // *ACM Transactions on Mathematical Software*. 1995. V. 22. Iss. 4. P. 469–483.

11. Preparata F., Shamos M. *Computational Geometry: An Introduction*. Moscow: Mir Publ., 1989. 478 p. (in Russ.).

12. Cigler G.M. *The theory of polyhedra*. Moscow: MCNMO Publ., 2014. 568 p. (in Russ.).

13. Eryomin I.I., Astafev N.N. *Introduction to the theory of linear and convex programming*. Moscow: Nauka Publ., 1976. 192 p. (in Russ.).

14. Aminov Yu.A. *Differential geometry and topology of curves*. Moscow: Nauka Publ., 1987. 160 p. (in Russ.).

15. Makarov I.M., Lohin V.M., Manko S.V., Romanov M.P. *Intellectual automatic control systems* / Ed. I.M. Makarov, V.M. Lohin. Moscow: Fizmatlit Publ., 2001. 576 p. (in Russ.).

16. Barkin A.I. *Absolute stability of control systems*. Moscow: Librokom Publ., 2012. 176 p. (in Russ.).

Об авторе:

Бердников Василий Петрович, аспирант кафедры проблем управления Института кибернетики ФГБОУ ВО «Московский технологический университет» (119454, Россия, Москва, пр-т Вернадского, д. 78).

About author:

Vasily P. Berdnikov, Postgraduate Student, Chair of Control Problems, Institute of Cybernetics, Moscow Technological University (78, Vernadskogo Pr., Moscow 119454, Russia).