

УДК 004.021

МОДИФИЦИРОВАННЫЙ АЛГОРИТМ ОПРЕДЕЛЕНИЯ ПОЛНЫХ ОБЛАСТЕЙ УСТОЙЧИВОСТИ НЕСТАЦИОНАРНЫХ НЕЛИНЕЙНЫХ СИСТЕМ

В.П. Бердников

МИРЭА - Российский технологический университет, Москва 119454, Россия

@Автор для переписки, e-mail: berdnikov_vp@mail.ru

В статье предлагается численный алгоритм построения сплайн-функций Ляпунова для исследования абсолютной устойчивости нелинейных нестационарных систем. В случае асимптотической устойчивости системы выполнение алгоритма приведет к построению поверхности уровня функции Ляпунова в виде кусочно-гладкой (гладкой, при выполнении дополнительных условий) замкнутой поверхности размерности, равной размерности исходной системы. Показано, что данный алгоритм позволяет существенно улучшить оценки границы устойчивости, получаемые с помощью частотных методов. В отличие от алгоритма построения кусочно-линейных функций Ляпунова, время работы предлагаемого алгоритма построения сплайн-функций Ляпунова не стремится к бесконечности при приближении системы к границе устойчивости. Данное обстоятельство позволяет использовать модифицированный алгоритм для определения устойчивости систем, находящихся близко к границе устойчивости. Приведены оценки точности определения области устойчивости на примере конкретной системы 3-го порядка. Даны рекомендации по выбору начальных условий работы алгоритма.

Ключевые слова: дифференциальные включения, нелинейные нестационарные системы, абсолютная устойчивость, функции Ляпунова, области устойчивости, сплайн Безье, полиномы Бернштейна.

MODIFIED ALGORITHM FOR DETERMINATION OF FULL STABILITY AREAS IN NONSTATIONARY NONLINEAR SYSTEMS

V.P. Berdnikov

MIREA - Russian Technological University, Moscow 119454, Russia

@Corresponding author e-mail: berdnikov_vp@mail.ru

The paper proposes a numerical algorithm for constructing Lyapunov spline functions for investigating the absolute stability of nonlinear nonstationary systems. 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 piece-wise smooth (smooth, if additional conditions are met) closed surface of dimension equal to the dimension of the original system. It is shown that the modified algorithm significantly improves the stability boundary estimates obtained with frequency methods. Unlike the algorithm for constructing piecewise linear Lyapunov functions, the running time of the proposed algorithm for constructing the Lyapunov spline functions does not tend to infinity as the system approaches the stability boundary. This circumstance makes it possible to use a modified algorithm to determine the stability of systems that are close to the stability boundary. An estimate of the accuracy of determining the stability area using an example of a third-order system is shown. Specific recommendations on the algorithm initial conditions choice are given.

Keywords: differential inclusions, nonlinear nonstationary systems, absolute stability, Lyapunov functions, stability areas, Bezier splines, Bernstein polynomials.

Настоящая статья является логическим продолжением ранее опубликованной работы [1], в которой на основе теоретических исследований о функциях Ляпунова [2–4], определяющих необходимые и достаточные условия устойчивости систем с секторными нестационарными нелинейными элементами, предложен численный алгоритм построения кусочно-линейных функций Ляпунова. На конкретных примерах было показано: в общем случае при приближении к границе устойчивости системы в пространстве параметров количество граней многогранной поверхности уровня функции Ляпунова стремится к бесконечности, что, в свою очередь, влечет за собой увеличение времени численного построения функции Ляпунова. Многогранная поверхность уровня кусочно-линейной функции Ляпунова при этом становится все более «гладкой». Таким образом, результаты исследований автора [1] совпадают с литературными данными [2–4] и свидетельствуют о целесообразности построения кусочно-гладкой (гладкой) функции Ляпунова вблизи границы устойчивости системы.

Необходимо отметить, что в рамках решения задачи об абсолютной устойчивости параметрически возмущаемых систем ранее уже был разработан метод построения гладких функций Ляпунова вида форм четных степеней (однородных полиномов многих переменных четных степеней), полное описание которого, а также обширную библиографию можно найти в монографии [5]. Метод базируется на решении задачи о положительной определенности форм четных степеней. Однако формы старших степеней не имеют аналога критерия Сильвестра для квадратичных форм, и определение положительной/отрицательной определенности формы старших степеней представляет собой самостоятельную нетривиальную задачу. Именно отсюда, в частности, возникают трудности поиска функции Ляпунова непосредственно в виде полинома четной степени, большей двух.

В отличие от [5], нами предлагается ниже метод построения гладких функций Ляпунова на основе сплайн-функций. При этом особая схема построения гарантирует положительную определенность функции Ляпунова, а сама функция строится в ходе решения специальной минимаксной задачи оптимизации.

1. Постановка задачи

Рассмотрим систему управления, в структуре которой есть несколько (либо один) нелинейных нестационарных элементов. Уравнение такой системы можно записать 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 $x = (x_1, x_2, \dots, x_d)^T$ – 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;

(\cdot, \cdot) – скалярное произведение векторов.

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 .

Ранее показано [2–4], что вопрос об 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 объединения точек (рис. 1);

$\bigcup_{k=1}^N$ – 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 .

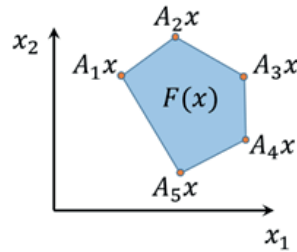


Рис. 1. Графическое представление правой части дифференциального включения (2).

Таким образом, общее количество матриц A_k равно 2^m , что соответствует количеству всех возможных комбинаций чисел λ_j .

Если существует строго выпуклая во всем пространстве \mathbb{R}^d положительно однородная степени $s \geq 1$ функция $v(x)$ ($x \in \mathbb{R}^d, x \neq 0, v(0)=0$), с отрицательно определенной производной в силу дифференциального включения $dv(x)/dt$, то дифференциальное включение (2) будет асимптотически устойчиво, а вместе с ним будет устойчива и соответствующая система с нелинейными нестационарными элементами (1). В случае гладкой функции $v(x)$ производная по направлению задается следующим образом:

$$\frac{dv(x)}{dt} = \max_{y \in F(x)} (\text{grad } v(x), y), \quad (4)$$

где $\text{grad } v(x)$ – градиент функции $v(x)$ в точке x .

В силу положительной однородности функции $v(x)$ ($v(\tau x) = \tau^s v(x)$, $\tau > 0$ при любом x) поверхности уровня $v(x)$ отличаются только масштабом и не отличаются формой, представляя собой границу выпуклого тела размерности d .

Если значение (4) отрицательно для любого x , то система асимптотически устойчива [2–4]. Дифференциальное включение (2) также является однородным, так как вектора $y \in F(x)$ меняют длину, но не направление при умножении x на $\tau > 0$. Следовательно, задача определения абсолютной устойчивости систем (1) и (2) редуцируется к задаче построения поверхности уровня функции Ляпунова. Условие устойчивости в этом случае (с геометрической точки зрения) можно трактовать как наличие тупого угла между нормалью $n = \text{grad } v(x)$ и каждым из векторов $y \in F(x)$ во всех точках x поверхности уровня $v(x)$. Если для случая кусочно-линейных функций Ляпунова поверхностью уровня был, как показано в [1], выпуклый многогранник, то в данной статье предлагается алгоритм построения кусочно-гладких и гладких поверхностей уровня функций Ляпунова для дифференциальных включений (2).

По существу, главным недостатком многогранников, как поверхностей уровня функций Ляпунова дифференциальных включений (2), является неограниченное возрастание количества граней при приближении системы к границе устойчивости. При таком подходе естественной выглядит идея «сглаживания» многогранника с небольшим количеством граней.

Предлагаемая нами схема построения поверхности уровня функции Ляпунова для систем (1) и (2) состоит из следующих этапов:

- 1) построение произвольного выпуклого многогранника с непустой внутренно-

стью в пространстве размерности d ;

2) сопоставление каждой гипергранни полученного многогранника некоторой гладкой поверхности, обеспечение непрерывной стыковки соседних поверхностей, а при необходимости – и гладкой стыковки;

3) оптимизация положения вершин многогранника и параметров поверхности для достижения отрицательного значения (4) в каждой точке поверхности, если это возможно (т.е. если система (1), (2) устойчива).

Перейдем к непосредственному описанию этапов построения поверхности уровня функции Ляпунова¹.

2. Построение замкнутой поверхности

На первом этапе необходимо построить выпуклый многогранник. Условия задачи требуют рассматривать только центрально-симметричные поверхности [2–4]. Следовательно, и многогранник, на основе которого нужно строить замкнутую поверхность, должен быть центрально-симметричным. Для построения такого многогранника достаточно произвольным образом задать m точек p_i в пространстве размерности d ($m \geq d$), далее воспользоваться алгоритмом построения выпуклых оболочек множества точек (см., например, [1, 6]). При этом во множество точек, на основе которого строится выпуклая оболочка, входят как p_i , так и $-p_i$. Непустая внутренность у многогранника возможна, когда не все точки лежат в $(d-1)$ -плоскости, проходящей через начало координат.

Для формирования непрерывной замкнутой поверхности уровня поступим следующим образом. Соотнесем каждую гипергрань f_r ($(d-1)$ -симплекс) с некоторой гладкой поверхностью s_r (рис. 2). Очевидно, что для непрерывности поверхности уровня функции Ляпунова у двух смежных гиперграней f_1 и f_2 на общей подгранице e поверхности s_1 и s_2 должны совпадать (рис. 3).

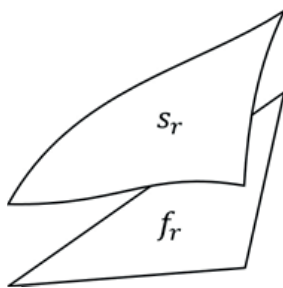


Рис. 2. Гипергрань f_r и соответствующая ей гладкая поверхность s_r в трёхмерном пространстве.

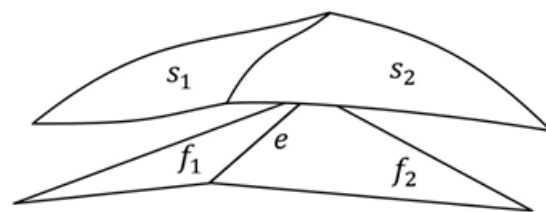


Рис. 3. Непрерывное соединение двух поверхностей s_1 и s_2 в трёхмерном пространстве.

Изложим методику построения кусочно-гладких поверхностей уровня функций Ляпунова. Для построения поверхностей s_r предлагается использовать многомерные многочлены Бернштейна-Безье (англ. *multivariate Bernstein-Bezier polynomials*) [7], в двумерном случае они полностью совпадают с обычными Безье-сплайнами [8], в трехмерном – с треугольными Безье-поверхностями (англ. *triangular Bernstein-Bézier patches*) [9]. Многомерные многочлены Бернштейна-Безье позволяют строить параметрические по-

¹В ходе описания используются свойства и определения для многогранников, приведенные в работе [1].

верхности в пространстве \mathbb{R}^d . Параметрами такой поверхности являются барицентрические координаты $u = (u_1, u_2, \dots, u_d) \in \mathbb{R}^d$, удовлетворяющие условиям

$$|u| = \sum_{i=1}^d u_i = 1, \quad u_i > 0.$$

Используя барицентрические координаты, по формуле (5) можно задать $(d+1)$ -симплекс в d -мерном пространстве, вершинами которого являются точки $p_i \in \mathbb{R}^d$. В двумерном случае симплекс эквивалентен отрезку, соединяющему точки p_1, p_2 , в трехмерном – треугольнику, вершинами которого являются p_1, p_2, p_3 .

$$p(u) = \sum_{i=1}^d p_i u_i \tag{5}$$

Многочлен Бернштейна порядка n определяется следующим образом:

$$B_\lambda^n(u) = \frac{n!}{\lambda!} u^\lambda, \quad |\lambda| = \sum_{i=1}^d \lambda_i = n, \quad \lambda_i \geq 0,$$

$$\lambda! = \lambda_1! \cdot \lambda_2! \cdot \dots \cdot \lambda_d!, \quad u^\lambda = u_1^{\lambda_1} \cdot u_2^{\lambda_2} \cdot \dots \cdot u_d^{\lambda_d},$$

где $\lambda_i \in \mathbb{Z}$,

тогда многомерная сплайн-поверхность определяется по формуле

$$s(u) = b_\lambda^n(u) = \sum_{|\lambda|=n} b_\lambda B_\lambda^n(u). \tag{6}$$

Для вычисления $s(u)$ на основе итерационной процедуры воспользуемся алгоритмом де Кастельжо (7):

$$b_\lambda^{r+1}(u) = \sum_{i=1}^d u_i b_{\lambda+e_i}^r(u), \tag{7}$$

$$|\lambda| = n - r, \quad r = 0, 1, \dots, n, \quad b_\lambda^0(u) = b_\lambda$$

здесь у векторов e_i все компоненты, кроме i -ой, равны нулю, а i -ая равна единице.

Преимущества использования алгоритма де Кастельжо по сравнению с прямым расчетом значений сплайн-поверхности по формуле (6) заключаются не только в вычислительной устойчивости при больших n , но и в том, что на предпоследнем шаге алгоритм дает d точек, через которые проходит плоскость, касательная к поверхности. Последнее обстоятельство крайне важно, так как условие устойчивости требует вычисления нормали к поверхности уровня функции Ляпунова, которую легко получить, имея d точек касательной гиперплоскости.

Графические интерпретации сплайн-поверхностей (6) и алгоритма де Кастельжо (7) для двумерного и трехмерного случаев приведены на рис. 4 и 5.

Чтобы построить такую поверхность, зададим в пространстве \mathbb{R}^d центрально-симметричный многогранник с непустой внутренностью. Все вершины этого многогранника должны

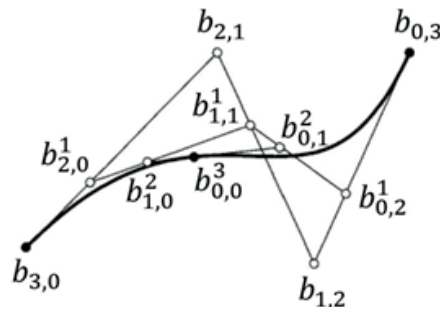


Рис. 4. Кривая Безье на плоскости, ее контрольные точки и этапы алгоритма де Кастельжо.

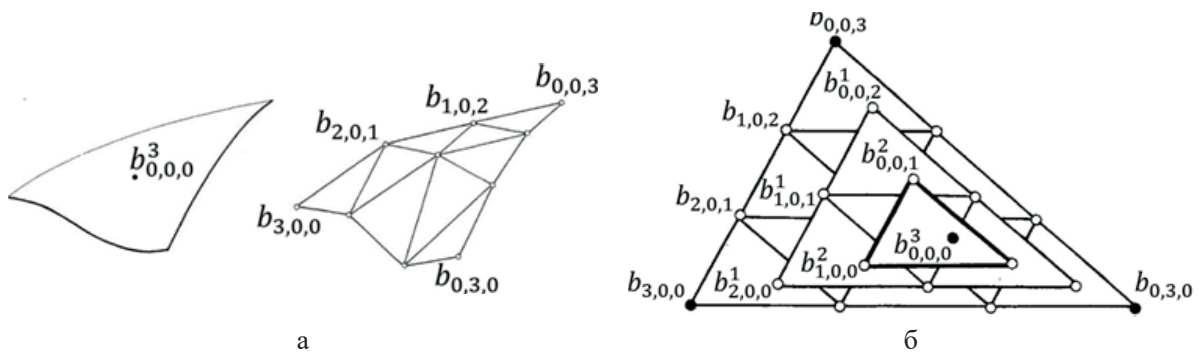


Рис. 5. Сплайн-поверхность в трехмерном пространстве и ее контрольные точки (а), алгоритм де Кастельжо итерационного вычисления значений поверхности (б).

лежать на единичной сфере. На каждой гиперграни введём сеть точек b_λ

$$b_\lambda = \frac{1}{n} \sum_{i=1}^d \lambda_i p_i, \quad |\lambda| = n, \tag{8}$$

где n определяет плотность сетки.

Пример сетки для $d = 3$ и $n = 4$ представлен на рис. 6.

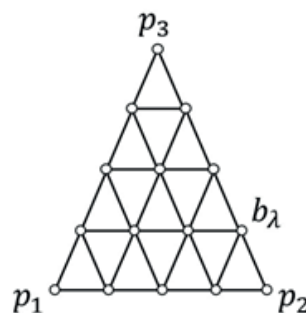


Рис. 6. Пример сетки на гиперграни в трехмерном пространстве.

Для каждой узловой точки b_λ введем некоторую величину – «высоту» $h_\lambda > 0$ и построим сплайн-поверхность s_p , используя в качестве контрольных точек $h_\lambda b_\lambda$. Вероятно, необходимо равенство высот h_λ на общем ребре двух смежных граней, тогда итоговая замкнутая поверхность будет непрерывной. На рис. 7 показана непрерывная замкнутая центрально-симметричная поверхность в трехмерном пространстве при $n = 4$, построенная на основе многогранника с восемью гранями.

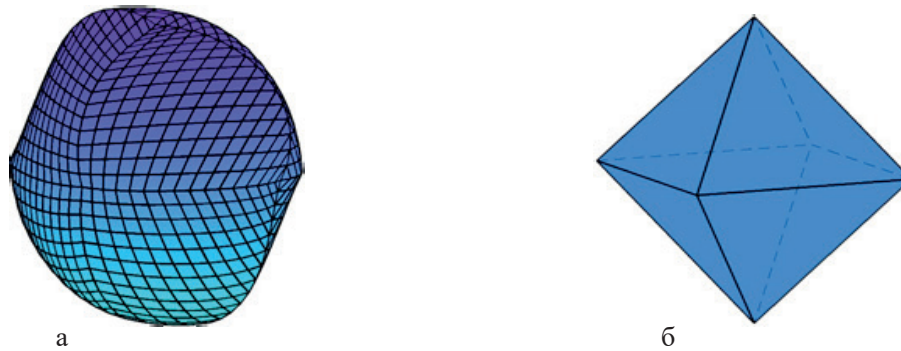


Рис. 7. Непрерывная замкнутая центрально-симметричная поверхность в трёхмерном пространстве (а), исходный многогранник (б).

Как видно, предложенная схема является гибким инструментом создания замкнутых поверхностей даже на основе очень простых многогранников. Теперь покажем, как вычислить нормаль к такой поверхности.

Выше уже отмечалось, что в процессе выполнения алгоритма де Кастельжо на предпоследнем шаге получается d точек, через которые проходит касательная плоскость. Обозначим эти точки v_1, v_2, \dots, v_d тогда нормальный вектор можно получить, используя матрицу T :

$$T = \begin{pmatrix} 1 & v_1^1 & v_2^1 & \dots & v_d^1 \\ 1 & v_1^2 & v_2^2 & \dots & v_d^2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & v_1^d & v_2^d & \dots & v_d^d \\ 0 & 1 & 1 & \dots & 1 \end{pmatrix}, \quad v_i = \begin{pmatrix} v_i^1 \\ v_i^2 \\ \vdots \\ v_i^d \end{pmatrix}. \quad (9)$$

Раскладывая определитель матрицы T по первому столбцу, получим d слагаемых, которые и являются компонентами вектора нормали. Например, для вычисления 2-ой компоненты необходимо найти

$$n^2 = (-1)^{1+2} \det \begin{pmatrix} v_1^1 & v_2^1 & \dots & v_d^1 \\ v_1^3 & v_2^3 & \dots & v_d^3 \\ \vdots & \vdots & \ddots & \vdots \\ v_1^d & v_2^d & \dots & v_d^d \\ 1 & 1 & \dots & 1 \end{pmatrix}.$$

Однако вычисленные нормали могут быть направлены как внутрь замкнутой поверхности, так и во вне ее. Чтобы нормали всегда были направлены во вне, их необходимо умножить на

$$(-1)^{d+1} \text{sign} \det \begin{pmatrix} v_1^1 & v_2^1 & \dots & v_d^1 \\ v_1^2 & v_2^2 & \dots & v_d^2 \\ \vdots & \vdots & \ddots & \vdots \\ v_1^d & v_2^d & \dots & v_d^d \end{pmatrix},$$

где $\text{sign } a$ принимает значение 1, если a положительное, и -1 , если a отрицательное.

Описанная выше процедура позволяет строить замкнутые поверхности и легко вычислять нормали к ним, что дает возможность свести задачу определения устойчивости системы (1), (2) к задаче оптимизации замкнутой сплайн-поверхности.

Перейдем к описанию последнего этапа предложенной в 1-ом разделе схемы построения поверхности уровня функции Ляпунова.

3. Формирование оптимизационной задачи

Поскольку устойчивость системы требует отрицательного значения (4) для любой точки x поверхности уровня, то оптимизационную задачу можно сформулировать как поиск такой сплайн-поверхности, для которой максимальное значение (4) является минимальным среди всех других сплайн-поверхностей такой же конфигурации. Уравнение (4) в данном случае примет вид:

$$\max_{y \in F(x)} (\text{grad } v(x), y) = \max_k (A_k s(u), n), \quad (10)$$

а учитывая (9), значение (10) можно переписать так:

$$\max_{y \in F(x)} (\text{grad } v(x), y) = \max_k \left((-1)^{d+1} \text{sign } \det(v_1 \dots v_d) \det \begin{pmatrix} A_k s(u) & v_1 & \dots & v_d \\ 0 & 1 & \dots & 1 \end{pmatrix} \right). \quad (11)$$

Отсюда очевидно, что сформулированная оптимизационная задача является задачей на минимакс и допускает решение с использованием численных методов [10, 11]. При этом на каждой итерации решения задачи на минимакс необходимо решать вспомогательную оптимизационную задачу поиска максимального значения $(A_k s(u), n)$ для всех поверхностей s_r . Если построенная в ходе выполнения процесса оптимизации поверхность уровня функции Ляпунова имеет в любой своей точке x отрицательное значение (4), то система асимптотически устойчива, если нет – то требуется выбрать другую конфигурацию многогранника с бóльшим количеством вершин и бóльшим значением n . Если не удастся построить поверхность уровня при большом количестве граней и высоком порядке сплайн-поверхности, то система неустойчива.

Переменными, которые должны меняться в ходе процесса решения минимаксной задачи оптимизации, являются вершины исходного многогранника p_i , а также высоты h_λ . Численные эксперименты показали, что для устойчивости оптимизационного процесса необходимо ввести в задачу следующие ограничения:

- $(p_i, p_i) = 1$: ограничение означает, что все вершины многогранника останутся на единичной сфере размерности d ;
- $(b_\lambda, b_\lambda) h_\lambda > 1$: предотвратит ситуацию бесконечного сжатия поверхности к началу координат;
- $\text{sign}(\det(p_1^n, \dots, p_d^n)) \cdot \det(p_1^0, \dots, p_d^0) > 0$, где p_i^n – вершины гиперграни до начала процесса оптимизации, а p_i^0 – вершины гиперграни в процессе оптимизации. Это ограничение сделает невозможным самопересечения многогранника.

Если в процессе оптимизации у сплайн-поверхности всюду значение (9) отрицательное, то процесс оптимизации нужно остановить, так как дальнейшая работа приведет к бесконечному «раздуванию» поверхности. Этот вывод следует из того, что при увеличении масштаба поверхности векторы $A_k s(u)$ и n из (9) сохраняют свое направление, но длина их увеличивается, т.е. отрицательные значения будут становиться еще меньше.

Учитывая все вышесказанное, сформулируем алгоритм построения сплайн-поверхности уровня функции Ляпунова для систем (1) и (2):

Шаг 1. По формуле (3) вычисляют матрицы A_k .

Шаг 2. На единичной сфере размерности d произвольным образом задают m точек p_i ($m \geq d$). Точки p_i не должны все лежать в $(d-1)$ -плоскости, проходящей через начало координат.

Шаг 3. Строят выпуклую оболочку множества точек, в которое входят p_i и $-p_i$, для чего можно, например, использовать алгоритм *beneath-beyond*, описанный в [1, 6].

Шаг 4. Для каждой гиперграни полученного центрально-симметричного многогранника вводят точки b_λ , вычисляемые по формуле (8). Каждой точке b_λ соответствует некоторая высота h_λ . Соответствующие высоты h_λ на общем ребре двух смежных граней должны совпадать. После этого на каждой гиперграни строят сплайн-поверхность.

Шаг 5. С помощью численных итерационных методов решается задача минимизации максимального значения $(A_k s(u), n)$, определяемого по формуле (11) при следующих ограничениях: $(p_r, p_i) = 1$, $(b_\lambda, b_\lambda)h_\lambda > 1$, $\text{sign}(\det(p_1^0, \dots, p_d^0)) \cdot \det(p_1^0, \dots, p_d^0) > 0$.

Шаг 6. На каждой итерации решают вспомогательную оптимизационную задачу поиска максимального значения $(A_k s(u), n)$ для всех поверхностей s_r .

Шаг 7. Если на какой-либо итерации максимальное значение $(A_k s(u), n) < 0$ для всех s_r при соблюденных ограничениях, описанных в шаге 5, то полученная сплайн-поверхность является поверхностью уровня однородной кусочно-гладкой функции Ляпунова, а системы (1), (2) устойчивы.

Шаг 8. Если после окончания итерационного процесса существуют s_r , для которых $(A_k s(u), n) > 0$, то следует увеличить количество вершин p_i или порядок сплайн-поверхности n и вернуться к шагу 2. Наличие $(A_k s(u), n) > 0$ при большом количестве вершин и высоком порядке поверхности свидетельствует о неустойчивости системы.

4. Анализ работы алгоритма

В заключение приведем результаты численных экспериментов на примерах расчета областей устойчивости системы 3-го порядка из [1]. Результаты сопоставления значений граничного коэффициента, полученного различными методами, даны в таблице.

Значения граничного² коэффициента при использовании различных методов анализа

Метод анализа	Частотный метод	Кусочно-линейная функция Ляпунова	Сплайн-функция Ляпунова
$\delta_{\text{гр}}$	2.15	3.0	4.06

²Под граничным понимается наибольший коэффициент, при котором метод дает положительное заключение об устойчивости.

Из таблицы видно, что граничный коэффициент, определяемый с помощью сплайн-функции Ляпунова, в 1.35 раза больше, чем для кусочно-линейной функции, и в 1.9 раза больше значения, полученного частотным методом. Здесь стоит пояснить, что с помощью кусочно-линейной функции Ляпунова можно получить более точную оценку граничного коэффициента $\delta_{гр}$, однако, уже при $\delta_{гр} = 3.0$ количество граней составляет около трех тысяч, а дальнейшее уточнение коэффициента потребует функций Ляпунова с десятками и сотнями тысяч граней, что резко отразится на времени расчета. Оно будет увеличиваться от нескольких минут до нескольких часов и более. Сплайн-функция Ляпунова с восемью гранями и $n = 4$ рассчитывается за 367 с, что составляет чуть более шести минут. Таким образом, вблизи границы устойчивости системы целесообразнее использовать сплайн-функции Ляпунова, а внутри области устойчивости – кусочно-линейные. Поверхность уровня сплайн-функции Ляпунова приведена на рис. 8.

Увеличение количества граней, а также порядка n сплайн-поверхности позволяет создавать поверхности все более сложной формы, что должно приводить к уточнению границы устойчивости. Однако численные эксперименты показали, что увеличение количества граней может приводить к неустойчивости процесса оптимизации. При этом приходится либо ограничивать максимальный шаг изменения переменных на каждом шаге оптимизационного процесса, либо вводить дополнительные ограничения (например, требование выпуклости многогранника на каждой итерации). В обоих случаях время расчета поверхности уровня становится неприемлемо высоким.

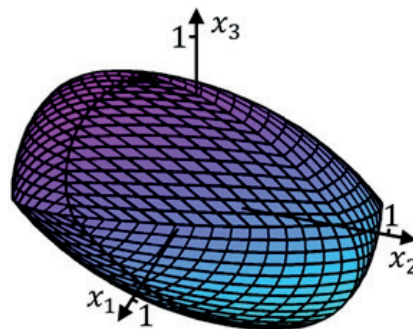


Рис. 8. Поверхность уровня сплайн-функции Ляпунова для системы третьего порядка.

Влияние увеличения порядка n сплайн-поверхности на точность определения границы устойчивости показано на рис. 9.

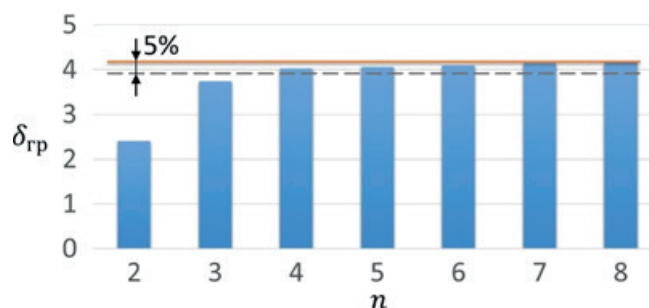


Рис. 9. Значения границы устойчивости при увеличении порядка сплайн-поверхности.

Как и следовало ожидать, наибольшее значение $\delta_{гр} = 4.152$ достигается при $n = 8$, т.е. с увеличением порядка сплайн-поверхности происходит уточнение границы устойчивости. В то же время уже при $n = 4$ погрешность определения границы устойчивости составляет менее 5% ($\delta_{гр} = 4.028$), что говорит о целесообразности использования для достаточно точной оценки областей устойчивости сплайн-поверхности невысокого порядка. Это особенно актуально при расчете областей устойчивости на плоскости (в пространстве) параметров, где необходим алгоритм построения сплайн-поверхности уровня функции Ляпунова для каждой точки плоскости (пространства), так как время работы алгоритма тем меньше, чем меньше порядок n .

При проведении исследований предполагалось, что в качестве вершин многогранника на этапе инициализации (Шаг 2 алгоритма) выбирали $p_i = e_i$, а все высоты h_λ принимали равными единице. Такие значения переменных для инициализации процесса оптимизации будем называть стандартными начальными условиями. В [1] было показано, что корректный выбор начального многогранника позволяет существенно сократить время работы алгоритма построения многогранной поверхности уровня функции Ляпунова. Аналогичное утверждение справедливо и для алгоритма построения сплайн-поверхности. Впрочем, выбор корректных начальных условий для конкретной системы (1), (2) крайне затруднителен.

При последовательном уточнении границы устойчивости возможно использование в качестве начальных условий поверхности уровня с предыдущего этапа расчета границы устойчивости. Так, в примере, рассмотренном выше, для определения максимального значения $\delta_{гр} = 4.06$ при $n = 4$ можно использовать в качестве начальной поверхность, рассчитанную для $\delta_{гр} = 3.0$. В этом случае время расчета составит 184 с, что почти в два раза меньше времени, которое затрачивается на алгоритм с использованием стандартных начальных условий (367 с). Если учесть, что время работы алгоритма для $\delta_{гр} = 3.0$ со стандартными начальными условиями составляет 32 с, то общее время расчета границы устойчивости за два этапа составит 216 с. Таким образом, вычисление поверхности уровня для $\delta_{гр} = 4.06$ в два этапа позволяет сэкономить 2.5 минуты по сравнению с одноэтапным способом.

Суммируя все вышеизложенное, можно дать следующие рекомендации по использованию предлагаемого в данной статье модифицированного алгоритма:

- количество вершин p_i задать равным d ; в качестве многогранника взять выпуклую оболочку множества точек $\{p_i, -p_i\}$;
- порядок сплайн-поверхности принять равным $n = d + 1$, имея в виду, что точность оценки границы устойчивости составит около 5%; для уточнения оценки увеличить порядок n ;
- если требуется определить устойчивость систем (1) и (2) с конкретными параметрами, то задать $p_i = e_i$, $h_\lambda = 1$;
- если требуется вычислить границу устойчивости систем (1), (2), то использовать несколько этапов, постепенно улучшая оценку границы; на каждом последующем этапе использовать поверхность с предыдущего этапа в качестве начальных условий для процесса оптимизации.

Если решение задачи требует построения полностью гладкой поверхности, то следует обеспечить дополнительные условия G^l гладкости [7, 9]. Однако целесообразно принять

во внимание, что выполнение данных условий для замкнутых поверхностей возможно не для всех конфигураций многогранника, на основе которого строится сплайн-поверхность [9]. В частности, для Безье-сплайнов целесообразно говорить о G^1 -непрерывности (геометрической, визуальной гладкости) [12]. Для визуальной гладкости поверхности уровня s_1 и s_2 на ребре e должны иметь совпадающие по направлению нормали. Из рис. 8 видно, что замкнутая поверхность имеет изломы на стыке двух Безье-сплайнов. Условия G^1 -гладкости выражаются в виде равенств, куда входят опорные точки смежных сплайнов Безье, но количество этих равенств резко возрастает с увеличением порядка n . К тому же эти условия разработаны только для размерности $d = 3$.

На практике для построения гладких поверхностей в задачу оптимизации придется вводить ограничения типа равенств, что увеличит время расчета поверхности уровня функции Ляпунова. Заметим, что, кроме схемы построения поверхностей в многомерном пространстве, изложенной в [7], существуют и другие, гарантирующие гладкость соединения двух смежных граней [9]. Однако схема, основанная на полиномах Бернштейна-Безье, является наиболее простой с точки зрения построения поверхностей, вычисления нормалей к ним и, как показано в данной работе, позволяет реализовать алгоритм определения полных областей устойчивости в пространстве параметров.

Заключение

В работе получен модифицированный алгоритм, обеспечивающий необходимые и достаточные условия устойчивости нелинейных нестационарных систем. В отличие от частотных методов, он позволяет выявлять полные области устойчивости в пространстве параметров. Вблизи границы устойчивости время работы модифицированного алгоритма существенно меньше, чем у алгоритма построения кусочно-линейных функций. Даны конкретные рекомендации по выбору параметров алгоритма. Показано, что достаточно точную оценку границу устойчивости можно получить, используя поверхности невысокого порядка.

Показано, что схема построения поверхностей уровня функции Ляпунова, основанная на полиномах Бернштейна-Безье, является наиболее простой. Использование вместо сплайнов Безье других типов поверхностей позволяет гарантировать гладкость соединения смежных поверхностей, хотя и может увеличивать время работы алгоритма. Таким образом, предложенная схема построения поверхностей функции Ляпунова является достаточно гибкой и позволяет достигать быстродействия работы алгоритма либо обеспечивать специальные свойства поверхности уровня при незначительных изменениях.

Работа выполнена при финансовой поддержке Российского научного фонда в рамках проекта № 16-19-00052 «Синтез интеллектуальных регуляторов для систем управления мобильных объектов с высокой степенью управляемости».

Литература:

1. Бердников В.П. Алгоритм определения полных областей устойчивости нестационарных нелинейных систем // Российский технологический журнал. 2017. Т. 5. № 6. С. 55–72.
2. 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.

3. 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.

4. 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.

5. Chesi G., Garulli A., Tesi A., Vicino A. Homogeneous polynomial forms for robustness analysis of uncertain systems. London: Springer-Verlag, 2009. 198 p.

6. 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.

7. Farin G. Triangular Bernstein-Bezier patches // Computer Aided Geometric Design. 1986. V. 3. № 2. P. 83–127.

8. Prautzsch H., Boehm W., Paluszny M. Bezier and B-Spline techniques. Berlin: Springer-Verlag, 2002. 304 p.

9. Farin G. Curves and surfaces for CAD: A practical guide (5th Edition). San Francisco: Morgan Kaufmann, 2001. 520 p.

10. Sun W., Yuan Y. Optimization theory and methods: Nonlinear programming. New York: Springer, 2010. 687 p.

11. Bonnans J., Gilbert J.C., Lemarechal C., Sagastizabal C.A. Numerical optimization: Theoretical and practical aspects. Berlin: Springer, 2006. 490 p.

12. DeRose T.D. Necessary and sufficient conditions for tangent plane continuity of Bezier surfaces // Computer Aided Geometric Design. 1990. № 7. P. 165–179.

References:

1. Berdnikov V.P. Algorithm of determination of non-stationary nonlinear systems full stability areas // Rossiiskiy tekhnologicheskiiy zhurnal (Russian Technological Journal). 2017. V. 5. № 6. P. 55–72. (in Russ.).

2. Molchanov A.P., Pyatnitskiy 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.).

3. Molchanov A.P., Pyatnitskiy 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.).

4. Molchanov A.P., Pyatnitskiy 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.).

5. Chesi G., Garulli A., Tesi A., Vicino A. Homogeneous polynomial forms for robustness analysis of uncertain systems. London: Springer-Verlag, 2009. 198 p.

6. 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.

7. Farin G. Triangular Bernstein-Bezier patches // Computer Aided Geometric Design. 1986. V. 3. № 2. P. 83–127.

8. Prautzsch H., Boehm W., Paluszny M. Bezier and B-Spline techniques. Berlin: Springer-Verlag, 2002. 304 p.
9. Farin G. Curves and surfaces for CAGD: A practical guide (5th Edition). San Francisco: Morgan Kaufmann, 2001. 520 p.
10. Sun W., Yuan Y. Optimization theory and methods: Nonlinear programming. New York: Springer, 2010. 687 p.
11. Bonnans J., Gilbert J.C., Lemarechal C., Sagastizabal C.A. Numerical optimization: Theoretical and practical aspects. Berlin: Springer, 2006. 490 p.
12. DeRose T.D. Necessary and sufficient conditions for tangent plane continuity of Bezier surfaces // Computer Aided Geometric Design. 1990. № 7. P. 165–179.

Об авторе:

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

About the author:

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