УДК 004.021 DOI: 10.32362/2500-316X-2018-6-5-25-44

ПОВЫШЕНИЕ ЭФФЕКТИВНОСТИ ПРОЦЕДУРЫ ПОСТРОЕНИЯ СПЛАЙН-ФУНКЦИЙ ЛЯПУНОВА ДЛЯ НЕЛИНЕЙНЫХ НЕСТАЦИОНАРНЫХ 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 функций Ляпунова для исследования абсолютной устойчивости нелинейных нестационарных систем. В случае асимптотической устойчивости выполнение алгоритма приведет к построению множества уровня функции Ляпунова в виде гладкой замкнутой поверхности размерности, равной размерности исходной системы. Для построения гладкого множества уровня функции Ляпунова разработан новый тип поверхности, что позволило свести задачу построения поверхности уровня к серии простых оптимизационных задач. Это гарантирует сходимость алгоритма. В отличие от алгоритма построения кусочно-линейных функций Ляпунова предлагаемый алгоритм обеспечивает возможность проведения анализа систем, находящихся вблизи границы устойчивости, за приемлемое время. Показана связь данного алгоритма и методов, основанных на частотных критериях и квадратичных функциях Ляпунова. Продемонстрировано значительное улучшение точности оценок границы устойчивости по сравнению с классическими методами. Выданы рекомендации по выбору начальных условий, обеспечивающие достижения баланса между точностью и скоростью работы алгоритма.

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

IMPROVING THE EFFICIENCY OF THE PROCEDURE OF LYAPUNOV SPLINE-FUNCTIONS CONSTRUCTION FOR NONLINEAR NONSTATIONARY 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 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 smooth closed surface of dimension equal to the dimension of the original system. To construct a smooth level set of the Lyapunov function, a new type of surface has been developed. Thus, the task of constructing the level set was reduced to a series of simple optimization problems, which guarantees the convergence of the algorithm. Unlike the algorithm for constructing piecewise linear Lyapunov functions, this algorithm analyses systems located near the stability boundary in an acceptable time. The relationship of this algorithm and methods based on frequency criteria and quadratic Lyapunov functions is shown. A significant improvement in the accuracy of estimates of the stability boundary was demonstrated in comparison with the classical methods. To achieve a balance between the accuracy and speed of the algorithm, recommendations on the choice of initial conditions are given.

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

Настоящая статья является логическим продолжением и развитием работ по созданию алгоритмов построения функций Ляпунова, определяющих необходимые и достаточные условия устойчивости систем с секторными нестационарными нелинейными элементами [1, 2], и опробование этих алгоритмов на практике. В статье [1] предложен алгоритм построения кусочно-линейных функций Ляпунова. В результате работы алгоритма происходит построение выпуклого многогранника, который является поверхностью уровня функции Ляпунова. В ходе численных экспериментов показано, что при приближении к границе устойчивости количество граней поверхности уровня неограниченно возрастает. Это совпадает с теоретическими выводами работ [3–5]. Таким образом, применение алгоритма, предложенного в [1], для построения полных областей устойчивости в пространстве параметров системы затруднено с практической точки зрения, так как требует чрезмерно большого времени расчета и вычислительных ресурсов.

Затем в [2] был разработан алгоритм построения кусочно-гладких поверхностей уровня, время работы которого в меньшей степени зависит от близости системы к границе устойчивости. С каждой гранью многогранника сопоставляется некоторая гладкая сплайн-поверхность Безье [6–8], благодаря чему замкнутые поверхности сложной формы удается воспроизвести на основе многогранников с малым количеством граней. Так, в [2] приведен пример системы 3-го порядка, для которой граница устойчивости определяется с помощью сплайн-поверхности функции Ляпунова на основе многогранника всего с восемью гранями. Использование алгоритма из [1] дает существенно менее точную оценку границы устойчивости для того же примера, а количество граней многогранной поверхности уровня превышает несколько тысяч. Таким образом, задача построения функции Ляпунова сведена к задаче оптимизации параметров сплайн-поверхности и вершин многогранников [2]. Эта оптимизационная задача является нелинейной, ее основные ограничения – неравенства формулируются с помощью определителей матриц. Так как в большинстве эффективных и быстрых методов оптимизации требуется вычисление градиентов и гессианов целевого функционала и ограничений [9, 10], а вычисление

производных от определителя матрицы по параметру сопряжено с рядом трудностей, то алгоритм построения поверхности уровня может в некоторых случаях не сходиться к нужному результату.

В связи с вышеизложенным в настоящей работе предлагается несколько измененная по сравнению с [2] концепция построения поверхности уровня функции Ляпунова, благодаря которой удается свести задачу определения устойчивости систем с секторными нестационарными нелинейными элементами к последовательности простых задач оптимизации. Такой подход значительно улучшает сходимость алгоритма построения поверхности уровня функции Ляпунова.

1. Постановка задачи

Рассмотрим систему управления, в структуре которой есть один или несколько нелинейных нестационарных элементов (см. также [2]). Уравнение такой системы можно записать в виде:

$$\frac{dx}{dt} = Ax + \sum_{j=1}^{m} b^j \varphi_j\left(\sigma_j, t\right)$$

$$\sigma_j = \left(c^j, x\right) = \sum_{i=1}^{d} c_i^j x_i, \qquad \varphi_j(0,t) \equiv 0 \tag{1}$$

где $x = (x_1, x_2, \ldots, x_d)^T$ – d-мерный вектор-столбец переменных состояния;

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

b^j и c^j ($j = 1, \ldots, m$) – 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

$$\delta_j^1 \le \frac{\varphi_j\left(\sigma_j, t\right)}{\sigma_j} \le \delta_j^2$$

$$\left(-\infty < \delta_j^1 \le \delta_j^2 < \infty, \;\; j = 1, \ldots, m\right)$$

при всех σ_j и t.

Ранее [3–5] было показано, 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

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

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 принадлежит множеству $F(x)$, определяемому правой частью дифференциального включения.

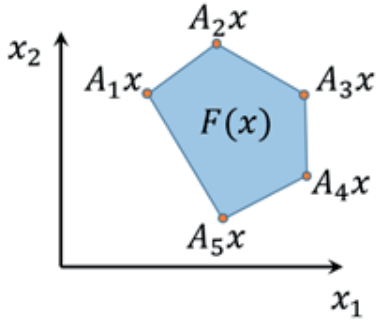


Рис. 1. Графическое представление правой части дифференциального включения (2).

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

$$A_k = A + \sum_{j=1}^{m} \lambda_j b^j \left(c^j\right)^T, \; k = 1,\ldots,2^m, \tag{3}$$

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.

Из [3–5] следует, что для асимптотической устойчивости системы (2), а вместе с ней и соответствующей системы с нелинейными нестационарными элементами (1), необходимо существование строго выпуклой во всем пространстве $\mathbb{R}^d$ однородной функции $v(x)$ ($x \in \mathbb{R}^d x \neq 0$, $v(0) = 0$), с отрицательно определенной производной в силу дифференциального включения $dv(x)/dt$. Если функция $v(x)$ гладкая, то производная функции Ляпунова в силу дифференциального включения вычисляется следующим образом:

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

где grad$v(x)$– градиент функции $v(x)$ в точке x.

В силу однородности функции Ляпунова и дифференциального включения (2) вместо построения функции Ляпунова целиком можно ограничиться построением только ее поверхности уровня, как это сделано в [1, 2]. Тогда grad$v(x)$ можно трактовать как нормаль к поверхности уровня, а x – как точку на поверхности уровня. В [2] поверхность уровня строилась на основе полиномов Бернштейна-Безье, т.е. поверхность была задана параметрически. Чтобы найти нормаль к поверхности, заданной параметрически в пространстве размерности d, первоначально необходимо найти $d-1$ касательных в точке x, а уже потом найти вектор, ортогональный к этим касательным. Построение ортогонального вектора можно выполнить с помощью вычисления определителя матрицы [11]:

$$\begin{pmatrix} \bar{e}_1 & \bar{e}_2 & \cdots & \bar{e}_d \\ \tau_1^1 & \tau_1^2 & \cdots & \tau_1^d \\ \vdots & \vdots & \ddots & \vdots \\ \tau_{d-1}^1 & \tau_{d-1}^2 & \cdots & \tau_{d-1}^d \end{pmatrix},$$

где $\bar{e}_1 = (1, 0,\ldots, 0)$, $\bar{e}_2 = (0, 1,\ldots, 0)$, …, $\bar{e}_d = (0, 0, \ldots, 1)$ – d векторов ортонормированного базиса пространства, а $\tau_i = (\tau_i^1, \tau_i^2, \ldots, \tau_i^d)$ – $d-1$ касательных в точке x. Поэтому в [2] для вычисления значения (4) используются определители.

Как отмечалось во введении, для реализации эффективных алгоритмов нелинейной оптимизации приходится вычислять градиенты и гессианы (вторые смешанные производные) от матриц, зависящих от параметра, что представляет определенную трудность. Так, если матрица T зависит от параметра s, то производная определителя этой матрицы по параметру будет иметь следующий вид [12]:

$$\frac{d}{ds}\det(T(s)) = \det(T(s))tr(T(s)^{-1}\frac{d}{ds}T(s)) = tr(adj(T(s))\frac{d}{ds}T(s)),$$

здесь det – определитель матрицы;

tr – след матрицы;

adj– присоединенная матрица.

Видно, что производная определителя по параметру выражается либо через обратную матрицу, либо через присоединенную. В первом случае при $\det(T(s)) \approx 0$ вычисление $T(s)^{-1}$ дает неопределенный результат, а во втором – требуется вычисление d^2 алгебраических дополнений $T(s)$, что существенно снижает скорость процесса оптимизации.

В представленной нами работе рассматривается проблема разработки алгоритма, сводящего задачу построения функции Ляпунова к серии простых оптимизационных задач. При этом схема построения поверхности уровня функции Ляпунова для систем (1) и (2) состоит из следующих этапов:

1) построение произвольного выпуклого многогранника с непустой внутренностью в пространстве размерности d;

2) сопоставление каждой гиперграни полученного многогранника специальной гладкой поверхности, благодаря чему обеспечивается непрерывная гладкая стыковка соседних граней;

3) определение параметров поверхности с целью достижения отрицательного значения (4) в каждой точке поверхности, если это возможно (т.е. если система (1), (2) устойчива), в ходе выполнения серии простых оптимизационных задач.

Ниже подробно описаны указанные этапы построения поверхности уровня функции Ляпунова. В ходе описания используются свойства и определения для многогранников, приведенные в работе [1], а также – для полиномов Бернштейна-Безье из работы [2].

2. Построение замкнутой поверхности

По аналогии с [2], на первом этапе строится выпуклый центрально-симметричный многогранник, для чего 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], [13]). В множество точек, на основе которого строится выпуклая оболочка, входят как p_i, так и $-p_i$. Чтобы у многогранника была непустая внутренность, необходимо иметь все точки не лежащими в одной $(d-1)$-плоскости, проходящей через начало координат. На следующем этапе происходит формирование непрерывной замкнутой поверхности уровня. С каждой гипергранью f_r ($(d-1)$-симплекс) сопоставляется некоторая гладкая поверхность s_r (рис. 2)[1]. Выше уже были отмечены трудности, возникающие при формировании оптимизационной задачи на основе таких поверхностей. Более того, при необходимости формирования полностью гладкой поверхности уровня в задачу оптимизации необходимо включать ограничения типа равенств, состоящих из определителей матриц, что также усложняет процесс оптимизации. Возможно использование других типов поверхностей (см., например, [6–8]), однако они имеют более сложную структуру, чем полиномы Бернштейна-Безье, что также негативно отразится на оптимизационном процессе.

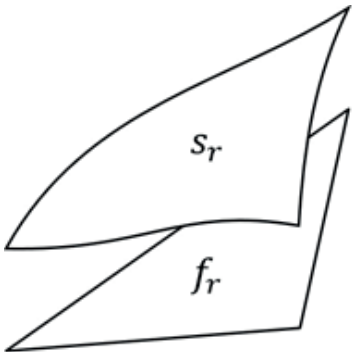


Рис. 2. Гипергрань f_r и соответствующая ей гладкая поверхность s_r в трехмерном пространстве.

Следовательно, становится очевидной необходимость разработки нового типа поверхностей в контексте решаемой задачи. Предлагаемая ниже идея позволяет решить сразу две задачи: упрощение процедуры оптимизации и разработка эффективного инструмента создания гладких замкнутых поверхностей. Последнее может иметь самостоятельный интерес и в других областях, например, в компьютерной графике. Дальнейшее изложение в данном разделе будет посвящено разработке нового типа поверхности, благодаря чему станет возможным упрощение оптимизационной процедуры из [2].

Суть идеи состоит в следующем: сначала задается уравнение нормали, а уже потом, на основе уравнения нормали, вычисляется сама поверхность. Очевидно, что для однозначности «порождаемой» таким способом поверхности на уравнения нормали должны быть наложены какие-то ограничения. Ниже дается общий вид уравнений нормали, ограничений и уравнений поверхности для произвольной размерности пространства.

Нами предлагается задавать уравнение нормали $n(u)$ в виде многомерных полиномов Бернштейна-Безье (Безье-сплайнов) порядка l:

$$n(u)=\sum_{|\lambda|=l} n_\lambda B_\lambda^l(u),\ B_\lambda^l(u)=\frac{l!}{\lambda!}u^\lambda,\ |\lambda|=\sum_{i=1}^{d}\lambda_i=l,\ \lambda_i\geq 0, \qquad (5)$$

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

[1]В [2] с этой целью используются многомерные многочлены Бернштейна-Безье [6].

где $\lambda_i \in \mathbb{Z}$, а n_λ – опорные нормали. Вектор $u = (u_1, u_2, \ldots, u_d) \in \mathbb{R}^d$ представляет собой барицентрические координаты, которые удовлетворяют условиям

$$|u| = \sum_{i=1}^{d} u_i = 1, \; u_i > 0. \tag{6}$$

Используя барицентрические координаты, получим уравнение гиперграни:

$$p(u) = \sum_{i=1}^{d} p_i u_i \; . \tag{7}$$

Уравнение поверхности будем искать в следующем виде:

$$s(u) = p(u)h(u) \; , \tag{8}$$

где $h(u)$ является неизвестной скалярной функцией. Далее $h_r(u)$ будем называть высотой гиперграни f_r или просто высотой.

Таким образом, необходимо найти функцию $h(u)$ такую, чтобы заданная $n(u)$ являлась нормалью к поверхности $s(u)$. Это возможно только в том случае, когда касательная по любому направлению $\tau(u)$ к поверхности $s(u)$ ортогональна нормали $n(u)$ для всех u, откуда получаем основное уравнение для поиска $h(u)$:

$$(\tau(u), n(u)) = 0 \; . \tag{9}$$

Первоначально рассмотрим ситуацию на плоскости, а позже обобщим результаты на произвольную размерность пространства. Для удобства вместо двух связанных соотношениями (6) барицентрических координат u_1, u_2 введем один независимый параметр t:

$$u_1 = 1 - t, \; u_2 = t, \; t \in [0,1]$$

Тогда уравнение (7) запишется как:

$$p(t) = p_1(1-t) + p_2 t \; .$$

При изменении параметра t от 0 до 1 точка p проходит по прямой от вершины p_1 до p_2 (рис. 3).

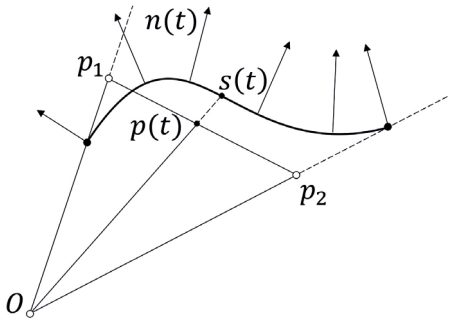


Рис. 3. Грань (отрезок $p(t)$) и соответствующая ей поверхность (кривая $s(t)$) в двумерном случае.

Тогда касательная $\tau(t)$ к кривой $s(t)$ определяется с помощью производной по параметру t:

$$\tau(\mathrm{t}) = \frac{d\,s(t)}{dt} = \frac{d\,p(t)}{dt}h(t) + p(t)\frac{d\,h(t)}{dt}$$

Подставляя уравнение $\tau(t)$ в (9), получим следующие соотношения, где для большей наглядности зависимость величин от параметра t опущена:

$$(\frac{dp}{dt}, n)h + (p, n)\frac{dh}{dt} = 0\,,$$

$$\frac{dh}{dt} = -h\frac{(\frac{dp}{dt}, n)}{(p, n)}\,.$$

Очевидно, что последнее уравнение представляет собой линейное однородное дифференциальное уравнение первого порядка с переменным коэффициентом. Общее решение его дано ниже:

$$h(t) = C\exp\left(-\int\frac{(\frac{dp}{dt}, n)}{(p, n)}dt\right) \tag{10}$$

В формуле (10) C – константа интегрирования. Отсюда поиск неизвестной функции $h(t)$ свелся к вычислению определенного интеграла. Поскольку $p(t)$, $n(t)$ являются полиномами от t, то и скалярные произведения (p, n), $(dp/dt, n)$ также являются полиномами от t, а значит, под знаком интеграла находится рациональная функция.

Чтобы проинтегрировать рациональную функцию, ее необходимо разложить на простейшие дроби, для чего придется вычислять корни знаменателя. Так как в ходе оптимизационного процесса поиска поверхности уровня функции Ляпунова параметры поверхности будут меняться, то корни знаменателя и последующее разложение на простейшие дроби придется вычислять на каждой итерации, что может оказаться чересчур затратным с точки зрения вычислений. Поэтому необходимо найти соотношения, при выполнении которых подынтегральная функция может быть преобразована к иному виду, что, в свою очередь, упростит вычисление интеграла.

Найдем производную от знаменателя по t:

$$\frac{d(p, n)}{dt} = (\frac{dp}{dt}, n) + (p, \frac{dn}{dt})\,.$$

Видно, что левое слагаемое уже содержится в числителе. Если при этом выполняется равенство

$$(p, \frac{dn}{dt}) = \alpha(\frac{dp}{dt}, n)\,, \tag{11}$$

где α – некоторое действительное число, то

$$(\frac{dp}{dt},n)=\frac{1}{(1+\alpha)}\frac{d(p,n)}{dt}.$$

Следовательно, интеграл в формуле (10) преобразуется к следующему виду:

$$\int\frac{(\frac{dp}{dt},n)}{(p,n)}dt=\frac{1}{(1+\alpha)}\int\frac{d(p,n)}{(p,n)}.$$

Он является табличным и равен

$$\frac{1}{(1+\alpha)}\int\frac{d(p,n)}{(p,n)}=\frac{1}{(1+\alpha)}\ln\left|(p,n)\right| .$$

Подставив результат решения интеграла в (10), имеем:

$$h(t)=C\exp\left(-\int\frac{(\frac{dp}{dt},n)}{(p,n)}dt\right)=C\exp\left(-\frac{1}{(1+\alpha)}\ln\left|(p,n)\right|\right)=C\frac{1}{(p,n)^{1/(1+\alpha)}}.$$

Последнее справедливо при выполнении следующего условия:

$$(p,n)>0 \qquad (12)$$

Полагая константу интегрирования $C = 1$, окончательно получаем уравнение поверхности

$$s(t)=p(t)h(t)=p(t)\frac{1}{(p(t),n(t))^{1/(1+\alpha)}},$$

которое справедливо, если выполняются условия (11) и (12).

Проверим полученный результат.

$$(\tau,n)=(\frac{ds}{dt},n)=(\frac{dp}{dt}h,n)+(p\frac{dh}{dt},n)=(\frac{dp}{dt},n)\frac{1}{(p,n)^{1/(1+\alpha)}}-\frac{1}{(1+\alpha)}(p,n)(p,n)^{(-1/(1+\alpha)-1)}\frac{d(p,n)}{dt}$$

Учитывая (11), приходим к соотношению

$$(\frac{dp}{dt},n)\frac{1}{(p,n)^{1/(1+\alpha)}}-\frac{1}{(1+\alpha)}(p,n)(p,n)^{(-1/(1+\alpha)-1)}\frac{d(p,n)}{dt}=$$

$$=\frac{1}{(1+\alpha)(p,n)^{1/(1+\alpha)}}\frac{d(p,n)}{dt}-\frac{1}{(1+\alpha)(p,n)^{1/(1+\alpha)}}\frac{d(p,n)}{dt}=0,$$

которое доказывает, что нормалью к полученной поверхности действительно является $n(t)$.

Распространим полученный результат на пространства произвольной размерности d. Поверхность, для которой $n(u)$ является нормалью, будем искать в виде

$$s(u) = p(u)h(u) = p(u)\frac{1}{\left(p(u), n(u)\right)^{1/(1+\alpha)}}, \tag{13}$$

откуда сразу получаем условие:

$$\left(p(u), n(u)\right) > 0, \tag{14}$$

гарантирующее, что при возведении $(p(u), n(u))$ в степень $1/(1+\alpha)$ получится действительное число.

Отличие от двумерного случая заключается в том, что теперь в каждой точке поверхности проходит касательная $(d-1)$-гиперплоскость. Данная гиперплоскость однозначно определяется $d-1$ линейно независимыми векторами, касательными к поверхности в данной точке. Так как барицентрические координаты не являются независимыми величинами, то для поиска касательной к поверхности, заданной с помощью барицентрических координат, необходимо использовать концепцию производных по направлению [6, 8].

Пусть ε_i барицентрические координаты d различных точек:

$$\varepsilon_1 = \left(1, 0, ..., 0\right), \varepsilon_2 = (0, 1, ..., 0), ..., \varepsilon_d = (0, 0, ..., 1)$$

С помощью данных координат зададим $d-1$ направлений σ_i:

$$\sigma_i = \varepsilon_{i+1} - \varepsilon_1 .$$

Производная по направлению $\rho = (\rho_1, \rho_2, \ldots, \rho_d)$ функции от барицентрических $f(u)$ определяется в виде [6]:

$$D_\rho f(u) = \sum_{i=1}^{d} \frac{\partial f(u)}{\partial u^i} \rho^i .$$

Тогда касательные к поверхности $s(u)$ по направлению σ_i вычисляются по формуле

$$\tau_i(u) = D_{\sigma_i} s(u) = \frac{\partial s(u)}{\partial u^{i+1}} - \frac{\partial s(u)}{\partial u^1} = \left(\frac{\partial p(u)}{\partial u^{i+1}} - \frac{\partial p(u)}{\partial u^1}\right) h(u) + p(u)\left(\frac{\partial h(u)}{\partial u^{i+1}} - \frac{\partial h(u)}{\partial u^1}\right).$$

Соотношение (9), которое должно выполняться для касательной по любому направлению, здесь можно заменить на $d-1$ уравнений для касательных по направлениям σ_i:

$$(\tau_i(u), n(u)) = 0$$

Подставляя значение $h(u)$ и опуская запись аргументов, получим:

$$(\tau_i,n)=\left(p_{i+1}-p_1,n\right)\frac{1}{\left(p,n\right)^{1/(1+\alpha)}}-\frac{1}{(1+\alpha)}(p,n)\left(\left(p,n\right)^{(-1/(1+\alpha)-1)}\frac{\partial(p,n)}{\partial u^{i+1}}-\left(p,n\right)^{(-1/(1+\alpha)-1)}\frac{\partial(p,n)}{\partial u^{1}}\right)=$$

$$=\left(p_{i+1}-p_1,n\right)\frac{1}{\left(p,n\right)^{1/(1+\alpha)}}-\frac{1}{(1+\alpha)\left(p,n\right)^{1/(1+\alpha)}}\left((p_{i+1}-p_1,n)+(p,\frac{\partial n}{\partial u^{i+1}}-\frac{\partial n}{\partial u^{1}})\right)$$

Из анализа формулы видно, что данное значение обращается в ноль, если выполняется равенство:

$$\left(p_{i+1}-p_1,n\right)-\frac{1}{(1+\alpha)}\left((p_{i+1}-p_1,n)+(p,\frac{\partial n}{\partial u^{i+1}}-\frac{\partial n}{\partial u^{1}})\right)=0,$$

или, после преобразований,

$$\left(p(u),\frac{\partial n(u)}{\partial u^{i+1}}-\frac{\partial n(u)}{\partial u^{1}}\right)=\alpha\left(p_{i+1}-p_1,n(u)\right),\ i=1,...,d-1. \tag{15}$$

Следовательно, при выполнении условий (15) Безье-сплайн $n(u)$ является нормалью к поверхности $s(u)$, заданной в виде (13). В уравнениях (15) слева и справа от знака «равно» находятся многомерные Безье-сплайны порядка l, где l – порядок Безье-сплайна $n(u)$. Хорошо известно [6, 8], что сплайны равны тогда, когда равны их опорные точки. Опорные точки сплайна, стоящего справа от знака «равно», обозначим c_λ^p, слева – c_λ^n. Приведенные ниже формулы позволяют вычислить опорные узлы сплайнов из (15):

$$c_\lambda^p=\alpha\left(p_{i+1}-p_1,n_\lambda\right),\ c_\lambda^n=\sum_{j=1}^{d}\lambda^j\left(p_j,n_{\lambda-e_j+e_i}-n_{\lambda-e_j+e_1}\right),$$

$$\lambda=(\lambda^1,\lambda^2,...,\lambda^d),\ e_1=(1,0,...,0),\ e_2=(0,1,...,0),\ e_d=(0,0,...,1),\ \left|\lambda\right|=l,\ i=1,...,d-1.$$

Теперь соотношение (15) можно переписать в виде системы уравнений, связывающих напрямую опорные узлы сплайнов $p(u)$, $n(u)$, а не сами сплайны:

$$\sum_{j=1}^{d}\lambda^j\left(p_j,n_{\lambda-e_j+e_i}-n_{\lambda-e_j+e_1}\right)-\alpha\left(p_{i+1}-p_1,n_\lambda\right)=0,\ \left|\lambda\right|=l,\ i=1,...,d-1. \tag{16}$$

Неравенство (14) также можно переписать, используя только опорные узлы $p(u)$, $n(u)$:

$$\frac{1}{l+1}\sum_{j=1}^{d}\lambda^j\left(p_j,n_{\lambda-e_j}\right)>0,\ \left|\lambda\right|=l+1. \tag{17}$$

Наконец, рассмотрим вопросы непрерывности и гладкости поверхностей, построенных на основе многогранников. Если каждой гиперграни f_r многогранника отвечает поверхность s_r, то для непрерывной и гладкой стыковки двух соседних поверхностей $s_1(u)$ и $s_2(u)$ опорные нормали сплайнов $n_1(u)$ и $n_2(u)$ должны совпадать. Более того, если функция высоты гиперграни f_r удовлетворяет соотношению $h_r(\varepsilon_i) = 1$, $i = 1,\ldots, d$, то гладкая по-

верхность будет проходить через все вершины гиперграни. Если отказаться от требования гладкости, то для непрерывной стыковки s_1 и s_2 на общей подграни соответствующие опорные точки Безье-сплайнов $(p_1(u), n_1(u))$ и $(p_2(u), n_2(u))$ должны совпадать. Однако в данной работе будут использоваться лишь полностью гладкие поверхности (рис. 4).

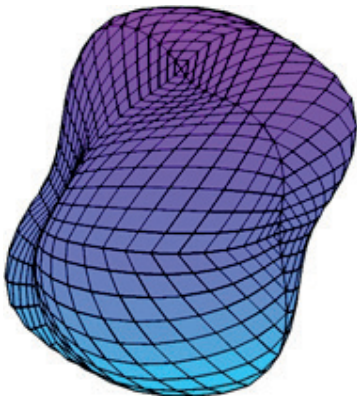


Рис. 4. Гладкая поверхность на основе многогранника с 8 гранями в трехмерном пространстве.

Таким образом, нами предложена новая концепция построения замкнутых гладких поверхностей на основе многогранников в пространстве произвольной размерности. Как видно из рис. 4, на основе многогранника с малым количеством граней можно получать поверхности сложной формы.

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

3. Формирование оптимизационной задачи

Устойчивость систем (1), (2) гарантирована, если значение (4) будет отрицательно в любой точке x поверхности уровня функции Ляпунова. В этом случае задача оптимизации заключается в поиске таких параметров поверхности, при которых это условие выполнено. Для поверхности, предложенной в предыдущем разделе, такими параметрами являются: вершины многогранника p_i, на основе которого строится поверхность, опорные нормали n_λ каждой гиперграни многогранника, а также параметр α, участвующий в уравнениях (16). Тогда уравнение (4) можно записать в следующем виде:

$$\max_{y\in F(x)}(grad\, v(x), y) = \max_k(A_k s(u), n(u)) = \max_k(A_k p(u)h(u), n(u)). \quad (18)$$

Заметим, что при выполнении условий (17) значение $h(u)$ будет положительным и не повлияет на знак выражения (18). Учитывая этот факт, а также избавляясь от операции взятия максимума, запишем (18) в виде системы неравенств (19), которые должны соблюдаться при любом u:

$$(A_k p(u), n(u)) < 0,\ \ k = 1,...,2^m. \quad (19)$$

Итак, для определения устойчивости необходимо решить систему равенств и неравенств (16), (17), (19), что может быть сделано с использованием численных методов оптимизации [9, 10].

Однако при решении данной задачи возникает ряд трудностей:

во-первых, уравнения и неравенства (16), (17), (19) являются нелинейными, и, как следствие, процесс оптимизации может сходиться к недопустимому решению, даже когда существует допустимое;

во-вторых, неравенства (19) должны быть справедливы при любом u, т.е. задача относится к теории полубесконечной оптимизации (*англ. semi-infinite programming*) и требует особых решений, поэтому в [2] на каждой итерации основного процесса оптимизации предлагалось решать вспомогательную оптимизационную задачу поиска точки x на поверхности, где достигается максимальное значение для (4).

Очевидно, что для решения первой проблемы необходимо свести задачу оптимизации к наиболее простому виду, что позволит гарантировать ее однозначную разрешимость при любых начальных условиях. Обыкновенные поверхности Безье, использованные в [2], таким ресурсом не обладают, однако предлагаемые в настоящей работе поверхности вида (13) обеспечат возможность существенно упростить процесс оптимизации. Действительно, уравнения и неравенства (16), (17), (19) зависят от опорных нормалей n_λ линейно. Следовательно, при фиксированных вершинах многогранника p_i и параметре α, для решения (16), (17), (19) могут быть применены методы линейного программирования, например, методы внутренней точки [14].

Для непосредственного применения методов линейного программирования необходимо аппроксимировать (19) некоторой конечномерной системой неравенств. Например, для каждой поверхности $s_r(u)$, где r – номер гиперграни, на основе которой построена поверхность, можно ввести сетку значений барицентрических координат u_μ по следующему принципу:

$$u_\mu = \frac{1}{l_u}\mu,\ |\mu| = l_u \ .$$

При этом l_u является некоторым натуральным числом, и чем оно больше, тем плотнее сетка значений барицентрических координат. Схематично покрытие поверхности точками $s_r(u_\mu)$ для размерности $d = 3$ показано на рис. 5.

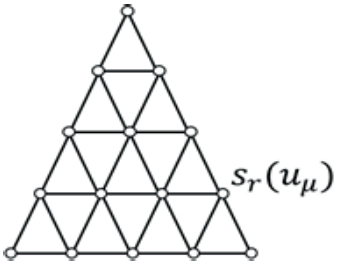


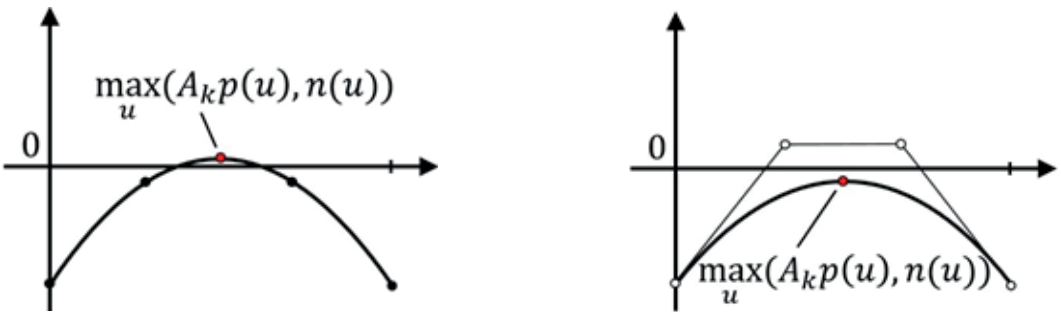
Рис. 5. Покрытие гиперграни сеткой точек $s_r(u_\mu)$ в трехмерном пространстве.

Подставляя u_μ вместо u в (19), получим конечномерную систему неравенств, линейных относительно опорных нормалей n_λ, которую с учетом других линейных ограничений (16), (17) можно решить с помощью численных методов линейного программирования. Однако данный подход не гарантирует, что неравенство (19) для найденной поверхности уровня функции Ляпунова выполняется при любом u, что легко понять, проанализировав

рис. 6а, на котором проиллюстрирована аппроксимация сплайна (19) в двумерном случае. Из рис. 6а видно, что хотя неравенство (19) выполняется в узлах u_μ (черные точки), между узлами оно может нарушаться (красная точка). В то же время, если задачу не удается решить даже для определенных u_μ, то ее точно нельзя будет решить для всех u. Отсюда следует, что предложенная схема приводит к необходимым условиям устойчивости (для заданных конфигураций многогранника и значения α), но не к достаточным. Повышение l_u способствует уплотнению сетки значений, а в пределе приведет к тому, что точки $s_r(u_\mu)$ будут покрывать всю поверхность s_r. Таким образом, при увеличении l_u возможно уточнение верхней оценки границы устойчивости системы, а в пределе условия перейдут в необходимые и достаточные (для данных конфигурации многогранника и значения α).

Рассмотрим, как можно получить нижнюю оценку границы устойчивости. По аналогии с (14) и (17), можно заметить, что (19) является сплайном Безье порядка l+1, где l – порядок сплайна $n(u)$. Значит, можно вычислить его опорные узлы, линейно зависящие от опорных нормалей n_λ и неравенства (19) заменить на конечномерную систему неравенств для опорных узлов. Это возможно благодаря тому, что если все опорные узлы сплайна Безье отрицательны, то и весь сплайн Безье будет также отрицательным. Действительно, многочлены Бернштейна-Безье неотрицательны и не равны все одновременно нулю при любом u, следовательно, отрицательные опорные узлы, умноженные на многочлены Бернштейна-Безье, в сумме дадут отрицательное число. Таким образом, если для всех гиперграней f_r и для всех матриц A_k опорные узлы сплайнов $(A_k p_r(u), n_r(u))$ отрицательны, то система устойчива. Однако, если решение подобной оптимизационной задачи не найдено, то это еще не означает, что система (1), (2) неустойчива.

На рис. 6 изображена ситуация, когда два центральных опорных узла (белые точки) сплайна $(A_k p_r(u), n_r(u))$ больше нуля, но весь сплайн располагается ниже нуля. Поэтому данная схема приводит к достаточным условиям устойчивости (для заданных конфигураций многогранника и значения α). Хорошо известно [6, 8], что любой Безье-сплайн порядка l можно представить в виде сплайна Безье большего порядка $l_u > l$. Повышение l_u приведет к уплотнению сетки опорных узлов сплайна, а сами опорные узлы будут располагаться «ближе» к поверхности сплайна и в пределе покроют всю поверхность. Значит, увеличение l_u способствует уточнению нижней границы устойчивости, а в пределе условия устойчивости перейдут в необходимые и достаточные.



а б

Рис. 6. Аппроксимация сплайна (19) в двумерном случае с помощью: а – сетки значений (черные точки) на сплайне; б – сетки опорных узлов (белые точки) сплайна.

Очевидно, что истинная граница устойчивости находится между верхней и нижней оценкой. Увеличение l_u приведет к сближению оценок, а сами оценки будут стремиться к некоторому значению, которое и станет истинной границей устойчивости. В то же время с ростом l_u растет и количество неравенств, и, как следствие, поиск поверхности уровня требует больших вычислительных ресурсов. На практике значение l_u приходится выбирать меньше 15 уже при размерности пространства $d = 4, 5$.

Стоит отметить, что в описанных выше случаях с ростом l_u в основном происходит добавление «неактивных» неравенств, т.е. тех неравенств, которые будут заведомо выполнены в точке решения. С целью уменьшения общего количества неравенств предлагается задачу поиска уровня функции Ляпунова представить в виде итерационного процесса, причем каждая итерация состоит из двух основных этапов:

- на первом этапе для каждой гиперграни и для каждой матрицы A_k осуществляется поиск барицентрических координат u_{max}, для которых значение $(A_k p_r(u), n_r(u))$ максимально. Если значение $(A_k p_r(u_{max}), n_r(u_{max})) > 0$, то в множество $\mathcal{M}$ добавляется подмножество $\{r, k, u_{max}\}$, где r – номер гиперграни.
- на втором этапе для каждого подмножества множества $\mathcal{M}$ формируется неравенство $(A_k p_r(u_{max}), n_r(u_{max})) < 0$. Полученная система линейных неравенств в совокупности с линейными ограничениями (16), (17) решается с помощью численных методов линейного программирования. На основе решения задачи линейного программирования обновляются опорные нормали n_λ каждой гиперграни.

Если на первом этапе для всех u_{max} значение $(A_k p_r(u_{max}), n_r(u_{max}))$ отрицательно, то система устойчива, а построенная поверхность является поверхностью уровня функции Ляпунова. Если на втором этапе задачу линейного программирования решить не удается, то при заданной конфигурации многогранника, выбранном параметре α и заданном порядке сплайна Безье $n(u)$ построить поверхность уровня функции Ляпунова невозможно. Повторение этой ситуации для достаточно большого порядка $n(u)$ свидетельствует о неустойчивости системы. Обе перечисленные выше ситуации являются критерием остановки итерационного процесса. Параметр α в данной работе предлагается брать равным единице.

Теперь можно перейти к описанию полного *алгоритма* построения сплайн-поверхности уровня функции Ляпунова для систем (1) и (2):

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

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

Шаг 3. Строится выпуклая оболочка множества точек, в которое входят как p_i, так и $-p_i$, для чего, например, целесообразно использовать алгоритм *beneath-beyound*, описанный в [1, 13].

Шаг 4. Для каждой гиперграни полученного центрально-симметричного многогранника вводятся опорные нормали n_λ из (5). Соответствующие опорные нормали n_λ на общем ребре двух смежных граней должны совпадать при дальнейшем изменении в ходе оптимизации. Параметр α принимается равным единице.

Шаг 5. С помощью численных методов нелинейной оптимизации [9, 10] решается задача поиска максимального значения $(A_k p_r(u), n_r(u))$. Если данное значение больше нуля,

то в множество $\mathcal{M}$ добавляется подмножество $\{r, k, u_{max}\}$, где r – номер гиперграни, k – номер матрицы, u_{max} – значение барицентрических координат, при которых достигается максимальное значение. Если для всех гиперграней и для всех A_k максимальное значение $(A_k p_r(u), n_r(u))$ меньше нуля, то система устойчива, а построенная поверхность является поверхностью уровня функции Ляпунова; в противном случае осуществляется переход к шагу 6.

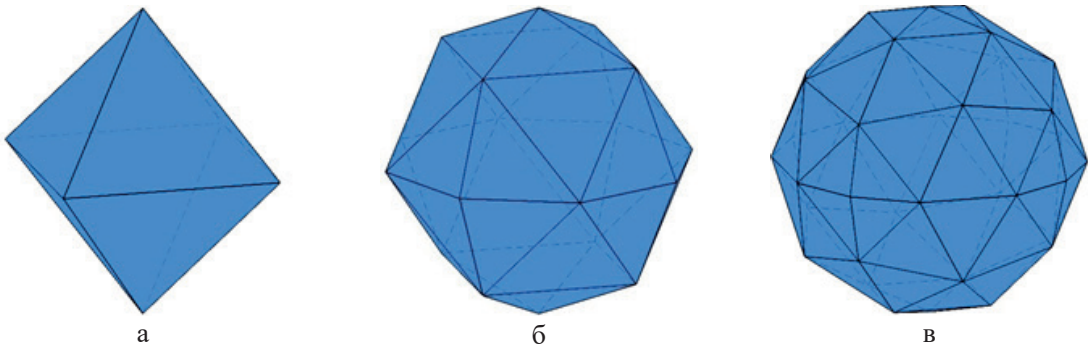
Шаг 6. Для каждого подмножества множества $\mathcal{M}$ формируется неравенство $(A_k p_r(u_{max}), n_r(u_{max})) < 0$. Полученная система линейных относительно n_λ неравенств вместе с линейными ограничениями (16), (17) решается с помощью методов линейного программирования. На основе решения задачи линейного программирования обновляются опорные нормали n_λ каждой грани. В случае успешного завершения оптимизационного процесса необходимо перейти к шагу 5. Если решение найти не удалось, то определить устойчивость при заданной структуре многогранника, параметре α и порядке сплайна $n(u)$ невозможно. Последнее при высоком порядке $n(u)$ свидетельствует о неустойчивости системы (1), (2).

Теперь можно перейти к оценке эффективности алгоритма на примере исследования конкретной системы.

4. Анализ работы алгоритма

В заключение проведем анализ влияния конфигурации исходного многогранника, а также порядка сплайна Безье $n(u)$ на работу алгоритма. В качестве критериев оценки работы алгоритма примем точность определения границы устойчивости и время расчета поверхности уровня. Для анализа воспользуемся результатами численных экспериментов по анализу устойчивости системы 3-го порядка из [1, 2].

В первую очередь определим степень влияния количества граней исходного многогранника на работу алгоритма. Зададим многогранники с различной конфигурацией: с 8 гранями (рис. 7а), с 32 гранями (рис. 7б) и с 72 гранями (рис.7в).



а б в

Рис. 7. Многогранники с различным количеством граней:
8 граней (а), 32 грани (б), 72 грани (в).

Определим граничный коэффициент системы $\delta_{гр}$ (под граничным понимается наибольший коэффициент, при котором метод дает положительное заключение об устойчивости) и время работы алгоритма t_a при различных конфигурациях многогранника и порядке l Безье-сплайна $n(u)$. Далее количество граней многогранника обозначим символом R, а порядок сплайна Безье $n(u) – l$. Результаты численных экспериментов сведены в таблицу.

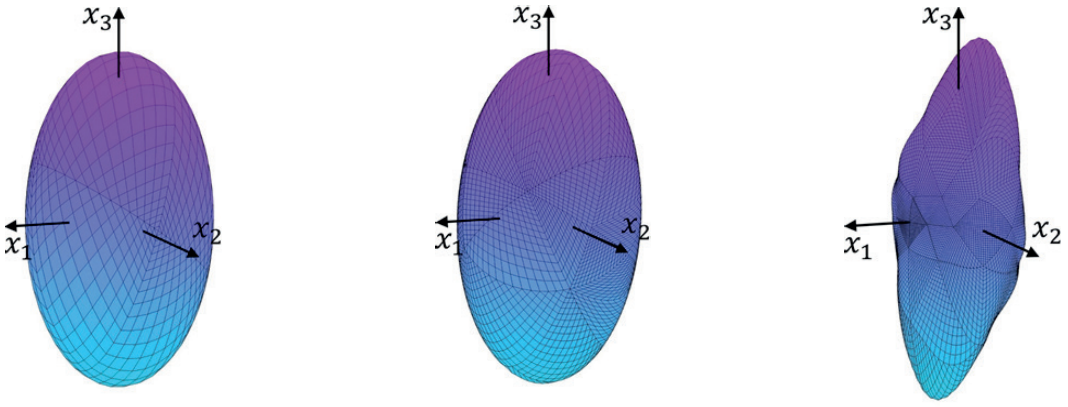
Как и следовало ожидать, наиболее точную оценку границы устойчивости $\delta_{гр} = 4.19$ удается получить при $R = 72$ и $l = 8$. Соответствующая поверхность уровня изображена на рис. 8в. Однако время расчета поверхности составляет 3136 с, что значительно усложняет использование поверхности с таким количеством граней и порядком сплайна $n(u)$ в задачах построения областей устойчивости на плоскости или в пространстве параметров системы. В то же время нужно заметить, что, начиная с $l = 5$, изменение граничного коэффициента $\delta_{гр}$ происходит только во втором знаке после запятой, при этом время расчета для $l = 5$ почти в пять раз меньше. Для $R = 8$, $l = 8$ значение граничного коэффициента на 4% меньше максимального и составляет $\delta_{гр} = 4.02$, а время работы алгоритма в 10 раз меньше, чем при $R = 72$, $l = 8$. Таким образом, границу устойчивости системы с менее, чем 5%-ной погрешностью можно определить даже с помощью поверхности, построенной на основе многогранника с малым количеством граней.

Значение граничного коэффициента и время работы алгоритма при различных конфигурациях многогранника

Количество граней многогранника	Порядок сплайна Безье $n(u)$	Граничный коэффициент	Время работы алгоритма, с
8	3	2.52	3
	4	2.86	19
	5	3.74	67
	6	3.85	114
	7	3.92	216
	8	4.02	314
32	3	2.52	14
	4	3.69	239
	5	4.02	188
	6	4.04	429
	7	4.13	520
	8	4.14	933
72	3	3.08	39
	4	3.99	275
	5	4.15	648
	6	4.16	863
	7	4.17	1643
	8	4.19	3136

Из анализа данных таблицы следует вывод, что граничные коэффициенты, рассчитанные для поверхностей $R = 8$, $l = 3$ и $R = 32$, $l = 3$, совпадают. Данный эффект является следствием совпадения построенных поверхностей уровня (рис. 8а,б). Оказывается, система уравнений (16) в этом случае разрешима только, если конструируемая поверхность является эллипсоидом. Вывод справедлив и для $l < 3$, и для других конфигураций многогранника. Так как поверхностью уровня положительно определенной квадратичной формы также является эллипсоид, то в любом случае оценки границы устойчивости системы, полученные на основе предлагаемого нами алгоритма, будут не хуже, чем у методов, основанных на построении квадратичной формы. То же самое касается некоторых частотных критериев, выполнение которых является достаточным (при определенных условиях необходимым

и достаточным) условием существования квадратичной функции Ляпунова [15]. Сравнивая рис. 8а, б и 8в, становится понятно, почему частотные критерии и квадратичные функции Ляпунова являются лишь достаточными условиями устойчивости: с помощью эллипсоидов просто невозможно воспроизвести ту сложную форму поверхности уровня функции Ляпунова, которой обладает система, находящаяся на границе устойчивости.



а б в

Рис. 8. Поверхность уровня функции Ляпунова в трехмерном пространстве:
а – $R = 8$, $l = 3$; б – $R = 32$, $l = 3$; в – $R = 72$, $l = 8$.

Суммируя все вышесказанное, можно дать следующие рекомендации по использованию предлагаемого алгоритма определения устойчивости:

- количество вершин p_i принять равным d; в качестве вершин взять $p_i = e_i$, а в качестве многогранника – выпуклую оболочку множества точек $\{p_i, -p_i\}$;
- порядок сплайна Безье $n(u)$ принять равным $l > d$, рассчитать значение граничного коэффициента;
- постепенно увеличивать l до тех пор, пока значение $\delta_{гр}$ не замедлит свой рост; при достаточно большом l погрешность оценки границы устойчивость составит около 5%, при $l = 1$ оценка $\delta_{гр}$ эквивалентна оценке, получаемой с помощью кругового критерия и квадратичной функции Ляпунова;
- для уточнения оценки увеличить количество граней R многогранника;

Заключение

В работе предложена новая схема построения функций Ляпунова, определяющих необходимые и достаточные условия устойчивости нелинейных нестационарных систем. Главным ее преимуществом является разработка нового типа поверхности, на основе которого строится процедура поиска поверхности уровня функции Ляпунова. Она выгодно отличается от обычных сплайнов Безье тем, что вычисление нормалей к данной поверхности не требует применения определителей матриц, а условия гладкости выражаются в виде обычных линейных равенств относительно параметров поверхности. Это позволяет свести задачу поиска поверхности уровня к итерационной процедуре, основным этапом которой является решение системы линейных равенств и неравенств. Кроме того, в предложенной схеме реализуется возможность построения функций Ляпунова для систем, находящихся вблизи границы устойчивости, за приемлемое время. Алгоритм построения поверхности уровня отличается простотой и гарантирует сходимость при любом началь-

ном многограннике. Даны конкретные рекомендации по выбору параметров алгоритма для получения точных оценок границ устойчивости.

Работа выполнена при поддержке РНФ 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. Бердников В.П. Модифицированный алгоритм определения полных областей устойчивости нестационарных нелинейных систем // Российский технологический журнал. 2018. Т. 6. № 3. С. 39–53.

3. Молчанов А.П., Пятницкий Е.С. Функции Ляпунова, определяющие необходимые и достаточные условия абсолютной устойчивости нелинейных нестационарных систем управления I // Автоматика и телемеханика. 1986. № 3. С. 63–73.

4. 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.

5. 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.

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

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

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

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

10. Bonnans J., Gilbert J.C., Lemarechal C., Sagastizabal C.A. Numerical Optimization: Theoretical and Practical Aspects. Berlin: Springer, 2006. 490 p.

11. Moreira L.S. Geometric analogy and products of vectors in n dimensions // Advances in Linear Algebra & Matrix Theory. 2013. V. 3. № 1. P. 1–6.

12. Petersen K.B., Pedersen M.S. The matrix cookbook. Technical University of Denmark, 2012. 72 p.

13. 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.

14. Roos C., Terlaky T., Vial J.-P. Interior point methods for linear optimization. Springer, 2006. 501 p.

15. Lipkovich M., Fradkov A. Equivalence of MIMO circle criterion to existence of quadratic Lyapunov function // IEEE Transactions on Automatic Control. 2016. V. 61. Iss. 7. P. 1895–1899.

References:

1. Berdnikov V.P. Algorithm of determination of non-stationary nonlinear systems full stability areas. *Rossiyskiy tekhnologicheskiy zhurnal* (Russian Technological Journal). 2017; 5(6): 55-72. (in Russ.)

2. Berdnikov V.P. Modified algorithm of determination of non-stationary nonlinear systems full stability areas. *Rossiyskiy tekhnologicheskiy zhurnal* (Russian Technological Journal). 2018; 6(3): 39-53. (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 I. *Avtomatika i telemekhanika* (Automation and Remote Control). 1986; (3): 63-73. (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 II. *Avtomatika i telemekhanika* (Automation and Remote Control). 1986; (4): 5-15. (in Russ.)

5. 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): 38-49. (in Russ.)

6. Farin G. Triangular Bernstein-Bezier patches. Computer Aided Geometric Design. 1986; 3(2): 83-127.

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

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

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

10. Bonnans J., Gilbert J.C., Lemarechal C., Sagastizabal C.A. Numerical Optimization: Theoretical and Practical Aspects. Berlin: Springer, 2006. 490 p.

11. Moreira L.S. Geometric analogy and products of vectors in n dimensions. *Advances in Linear Algebra & Matrix Theory*. 2013; 3(1): 1-6.

12. Petersen K.B., Pedersen M.S. The matrix cookbook. Technical University of Denmark, 2012. 72 p.

13. Barber C.B., Dobkin D.P., Huhdanpaa H. The Quickhull algorithm for convex hulls. *ACM Transactions on Mathematical Software*. 1995; 22(4): 469-483.

14. Roos C., Terlaky T., Vial J.-P. Interior point methods for linear optimization. Springer, 2006. 501 p.

15. Lipkovich M., Fradkov A. Equivalence of MIMO circle criterion to existence of quadratic Lyapunov function. *IEEE Transactions on Automatic Control*. 2016; 61(7): 1895-1899.

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).

Для цитирования: Бердников В.П. Повышение эффективности процедуры построения сплайн-функций Ляпунова для нелинейных нестационарных систем // Российский технологический журнал. 2018. Т. 6. № 5. С. 25–44. DOI: 10.32362/2500-316X-2018-6-5-25-44.

For citation: Berdnikov V.P. Improving efficiency of the procedure of Lyapunov spline-functions construction for nonlinear nonstationary systems. *Rossiyskiy tekhnologicheskiy zhurnal* (Russian Technological Journal). 2018; 6(5): 25-44. (in Russ.). DOI: 10.32362/2500-316X-2018-6-5-25-44.