

PACS 02.30.Yy

© 2009 г. Е.Ф. САЧКОВА
(Институт программных систем РАН, Переславль-Залесский)

ПРИБЛИЖЕННОЕ РЕШЕНИЕ ДВУХТОЧЕЧНЫХ ГРАНИЧНЫХ ЗАДАЧ ДЛЯ СИСТЕМ С ЛИНЕЙНЫМИ УПРАВЛЕНИЯМИ¹

Рассматривается двухточечная граничная задача для нелинейных систем с линейным управлением без ограничений. Для трехмерных по состоянию систем приводится описание локального алгоритма приближенного решения задачи управления. На основе этого алгоритма построены алгоритмы приближенного решения задачи управления на трехмерном интегральном многообразии (орбите) n -мерных по состоянию систем с двумерным линейным управлением. Работа алгоритмов демонстрируется на примерах.

1. Введение

В данной работе рассматривается задача конструктивного управления нелинейными системами с двумерным линейным управлением без ограничений. На основе геометрического метода – метода нильпотентной аппроксимации (см. [1, 2]), автором разработан алгоритм перемещения трехмерной нелинейной системы с двумя линейными управлениями из заданного начального состояния в малую окрестность заданного финального состояния. Этот алгоритм реализован в системе компьютерной математики Maple [3]. Написанная программа осуществляет итерационный процесс последовательного приближения системы к цели с помощью управлений другой, более простой, управляемой системы, являющейся аппроксимацией исходной. В [4] найдены точные решения двухточечной граничной задачи для нильпотентных аппроксимаций в нескольких классах управлений, их компьютерная реализация составляет библиотеку управлений для программы. На конкретном примере с помощью компьютерной программы показана эффективность алгоритма в сравнении со стандартным методом аппроксимации скобки Ли.

Широкие возможности программы обусловлены эффективностью как алгоритма, так и численных и символьных расчетов, осуществляемых системой Maple, поэтому она может быть полезна при решении большого числа прикладных задач, возникающих при управлении мобильными роботами [5], спутниками (при неограниченных управлениях) и иными импульсными системами [6, 7], качением твердых тел [8, 9]. Приводится пример апробации программы при решении задачи управления ориентацией сферы, которая катится по плоскости без проскальзывания и прокручивания. Рассмотрены управления, ранее не апробированные: кусочно-постоянные с одним переключением (см. [4]); управления, построенные с помощью линейных векторных полей на плоскости, имеющих особенность типа центр и типа фокус.

¹ Работа поддержана Российским фондом фундаментальных исследований (проект № 06-01-00330).

В настоящей работе рассматривается применение алгоритма приближенного решения двухточечной задачи для трехмерных нелинейных систем с двумя линейными управлениями к n -мерным ($n > 3$) нелинейным управляемым системам с двумя линейными управлениями и трехмерным многообразием полной управляемости, называемым орбитой. Построен алгоритм решения глобальной задачи управления на орбите. Приводится пример апробации алгоритма для четырехмерной системы с трехмерной орбитой.

Разработанные алгоритмы будут полезны для реализации обобщенных траекторий предельных систем для систем с неограниченными управлениями [6, 10].

2. Постановка задачи управления

Рассматривается управляемая система вида

$$(1) \quad \dot{x} = u_1 X_1(x) + u_2 X_2(x), \quad x \in \mathbb{R}^3, \quad u = (u_1, u_2) \in \mathbb{R}^2,$$

где

$$(2) \quad X_1, X_2, X_3 = [X_1, X_2] = \frac{\partial X_2}{\partial x} X_1 - \frac{\partial X_1}{\partial x} X_2,$$

— линейно независимые гладкие векторные поля в $\mathbb{R}^3$, с заданными граничными условиями и точностью:

$$(3) \quad x(0) = x^0, \quad x(T) = x^1, \quad x^0, x^1 \in \mathbb{R}^3, \quad T > 0, \quad \varepsilon > 0.$$

Требуется найти кусочно-непрерывные управления $u(t) = (u_1(t), u_2(t))$, чтобы переместить систему (1), удовлетворяющую условию (2), из начального состояния x^0 в ε -окрестность конечного состояния x^1 за время $T > 0$.

Задача (1)–(3) разрешима в $\mathbb{R}^3$, так как из условия (2) следует полная управляемость системы (1) в $\mathbb{R}^3$ ([8], с. 73, 78).

В [4] рассмотрен подход к решению задачи (1)–(3), основанный на методе нильпотентной аппроксимации системы (1), (2) в окрестности точки x^1 .

Напомним необходимые сведения. Система вида (1) называется *нильпотентной*, если ее алгебра Ли

$$\text{Lie}(X_1, X_2) = \text{span}(X_1, X_2, [X_1, X_2], \dots, [X_{i_1}, [X_{i_2}, \dots, [X_{i_k}, X_{i_{k+1}}] \dots]], \dots)$$

нильпотентна, т.е. для некоторого номера N все скобки Ли длины N $[X_{i_1}, [X_{i_2}, \dots, [X_{i_N}, X_{i_{N+1}}] \dots]] = 0$ для любых $i_1, \dots, i_N, i_{N+1} \in \{1, 2\}$. Для систем (1), (2) нильпотентные аппроксимации, согласно теореме Белаиша [1], имеют $N = 2$.

При решении задачи (1)–(3) в алгоритме используются две управляемые системы: исходная система (1), (2) и ее нильпотентная аппроксимация в окрестности точки x^1 , вычисленная в некоторых специальных координатах, связанных с исходной системой (называемых привилегированными координатами):

$$(4) \quad \begin{aligned} \dot{z}_1 &= u_1, \quad \dot{z}_2 = u_2, \\ \dot{z}_3 &= u_1(c_{11}z_1 + c_{12}z_2) + u_2(c_{21}z_1 + c_{22}z_2), \quad c_{12} \neq c_{21}, \end{aligned}$$

которая сводится заменой переменных

$$(5) \quad G(z) = \left(z_1, z_2, \frac{1}{c_{21} - c_{12}} \left(z_3 - \frac{c_{21} + c_{12}}{2} z_1 z_2 - \frac{c_{11}}{2} z_1^2 - \frac{c_{22}}{2} z_2^2 \right) \right)$$

к симметричной нильпотентной системе

$$(6) \quad \dot{y}_1 = u_1, \quad \dot{y}_2 = u_2, \quad \dot{y}_3 = (u_2 y_1 - u_1 y_2)/2.$$

Заметим, что граничным состояниям (x^0, x^1) исходной системы соответствуют граничные состояния $(y^0, 0)$ системы (6).

Метод приближенного решения задачи управления (1)–(3) заключается в последовательном приближении системы (1) к целевой точке x^1 с помощью управлений системы (6), вычисляемых на каждой итерации и точно переводящих систему (6) в целевую точку.

Решения задачи управления

$$(7) \quad y(0) = y^0, \quad y(T) = 0, \quad T > 0$$

для системы (6) найдены в пяти классах управлений: в тригонометрическом, кусочно-постоянном с одним переключением, оптимальном в смысле минимума функционала субримановой длины [4]; центральном и фокусном, построенных с помощью линейных векторных полей на плоскости, имеющих соответствующую особенность типа центр и типа фокус.

3. Локальный алгоритм

Напомним алгоритм приближенного решения задачи управления (1)–(3). Этот итерационный алгоритм основан на методе нильпотентной аппроксимации. Из общей теории [11] следует, что для любой точки x^1 существует радиус сходимости $\delta > 0$ этого алгоритма, т.е. такое число $\delta = \delta(x^1) > 0$, что для всех точек x^0 , $|x^0 - x^1| < \delta$, построенная далее в алгоритме последовательность приближений q^n сходится к x^1 . Будем далее решать задачу перемещения системы (1) из точки x^0 в точку x^1 при условии $|x^0 - x^1| < \delta$; такую задачу управления будем называть *локальной*.

Обозначим через $\mathcal{U}$ класс управлений, используемых в алгоритме для перемещения системы (например, оптимальных в смысле некоторого функционала, тригонометрических, кусочно-постоянных). Напомним, что используемая далее матрица $F(x)$ имеет вид:

$$(8) \quad F(x) = (X_1 \mid X_2 \mid X_3),$$

т.е. составлена по столбцам из векторов X_1, X_2 правой части системы (1) и их коммутатора X_3 .

Алгоритм приближенного решения локальной задачи управления (1)–(3):

- 1) **Проверка условия достижения цели:** если $|x^0 - x^1| < \varepsilon$, то цель достигнута и алгоритм останавливается. Далее предполагается, что $|x^0 - x^1| \geq \varepsilon$.
- 2) **Вычисление нильпотентной аппроксимации** исходной системы (1), (2) в окрестности целевой точки x^1 : вычисляются коммутатор $X_3 = [X_1, X_2]$, матрица $F(x) = (X_1, X_2, X_3)(x)$, коэффициенты c_{ij} , $i, j = 1, 2$.
- 3) **Итерационный процесс.** В качестве начального приближения на первой итерации берется $q^0 = x^0$. Пусть q^{n-1} – приближение к целевой точке x^1 , полученное на $(n-1)$ -й итерации.
 - а) Выбирается класс управлений $\mathcal{U}$.
 - б) Вычисляются координаты начальной точки q^{n-1} в привилегированных координатах, центрированных в целевой точке x^1 : $z^{n-1} = F^{-1}(q^{n-1})(q^{n-1} - x^1)$.
 - в) Вычисляются координаты y^{n-1} начальной точки q^{n-1} в системе координат (y_1, y_2, y_3) системы (6): $y^{n-1} = G(c_{ij}, z^{n-1})$, где отображение G задано формулой (5).
 - г) По формулам работы [4] вычисляются управления $\hat{u}^n \in \mathcal{U}$, переводящие систему (6) из точки y^{n-1} в точку $0 \in \mathbb{R}^3$ за время T .

д) Решается задача Коши для исходной системы (1) с управлениями $\hat{u}^n$:

$$\dot{x} = \hat{u}_1^n(t)X_1(x) + \hat{u}_2^n(t)X_2(x), \quad x(0) = q^{n-1}, \quad t \in [0, T];$$

обозначим ее решение через $x^n(t)$.

е) В качестве следующего приближения берется точка $q^n = x^n(T)$.

ж) Проверяется условие достижения цели: если $|q^n - x^1| < \varepsilon$, то цель достигнута и алгоритм останавливается. Если $|q^n - x^1| \geq \varepsilon$, то совершается переход к следующей итерации, к п. 3), а, и в качестве начального приближения берется q^n . Из сходимости алгоритма при условии $|x^0 - x^1| < \delta$ следует, что на некоторой итерации N выполнится условие $|q^N - x^1| < \varepsilon$, и алгоритм остановится.

4) **Приближенное решение локальной задачи управления** дается последовательным применением управлений $\hat{u}^1, \dots, \hat{u}^N$, вычисленных на каждой итерации и перепараметризованных соответствующим образом:

$$(9) \quad u(t) = \begin{cases} N\hat{u}^1(Nt), & t \in [0, T/N], \\ N\hat{u}^2(Nt - T/N), & t \in [T/N, 2T/N], \\ \dots\dots\dots \\ N\hat{u}^N(Nt - (N-1)T/N), & t \in [(N-1)T/N, T]. \end{cases}$$

Смысл этих формул в следующем: если, например, управление $\hat{u}^1(t)$ переводит точку q^0 в точку q^1 на отрезке времени длины T , то управление $N\hat{u}^1(Nt)$ переводит точку q^0 в точку q^1 на отрезке времени длины T/N , и т.д.; в результате управление $u(t)$ определено на отрезке $t \in [0, T]$. Управление $u(t) = u(x^0, x^1, T, \varepsilon; t)$, полученное с помощью формул (9), переводит систему (1) за время $T > 0$ из точки x^0 в точку x^1 с заданной точностью $\varepsilon > 0$, следовательно, является приближенным решением локальной задачи управления (1)–(3).

С помощью этого локального алгоритма можно построить и *глобальный* алгоритм, введя по некоторому правилу промежуточные узлы $x_1^1, \dots, x_k^1$ так, что $x_1^1 = x^0$, $x_k^1 = x^1$ и $|x_{i+1}^1 - x_i^1| < \delta(x_i^1)$.

Локальный алгоритм приближенного решения задачи управления (1)–(3) реализован в виде компьютерной программы FindControlLoc, написанной на входном языке системы Maple. Все управления реализованы в виде пяти процедур программы FindControlLoc и составляют библиотеку управлений NilpControls.

Отметим существенные характеристики программы:

- возможность рассматривать произвольные системы вида (1), (2);
- возможность выбирать произвольные граничные условия (3) (близкие в смысле δ);
- решение задачи управления (1)–(3) в пяти классах управлений.

В качестве иллюстрации работы программы FindControlLoc рассмотрим следующий пример. Управляемая система, описывающая качение сферы по плоскости без прокручивания и проскальзывания, описана в [8, 9]. Рассмотрим подсистему этой системы, описывающую изменение ориентации сферы. Переходя в этой подсистеме от ортогональных 3×3 матриц к кватернионам и применяя проекцию на трехмерное пространство, получим следующую систему:

$$(10) \quad \begin{aligned} \dot{x}_1 &= -x_3u_1 + x_2u_2, \\ \dot{x}_2 &= -\sqrt{1-x_1^2-x_2^2-x_3^2}u_1 - x_1u_2, \\ \dot{x}_3 &= x_1u_1 - \sqrt{1-x_1^2-x_2^2-x_3^2}u_2, \\ x &= (x_1, x_2, x_3) \in B^3 = \{x_1^2 + x_2^2 + x_3^2 < 1\}, \quad u = (u_1, u_2) \in \mathbb{R}^2. \end{aligned}$$

С помощью компьютерной программы FindControlLoc задача управления (1)–(3) для системы (10) решена в пяти классах управлений библиотеки NilpControls.

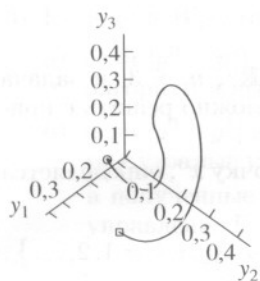


Рис. 1. Оптимальная траектория $x(t)$.

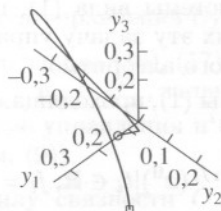


Рис. 2. Фокусная траектория $x(t)$.

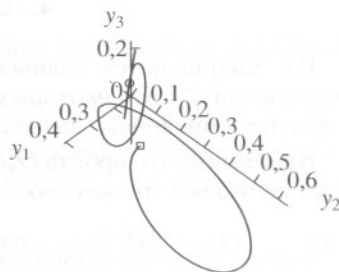


Рис. 3. Тригонометрическая траектория $x(t)$.

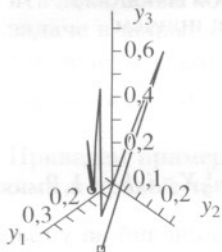


Рис. 4. Кусочно-постоянная траектория $x(t)$.

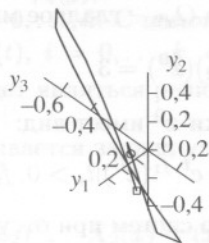


Рис. 5. Центральная траектория $x(t, 1, 4, -1)$.

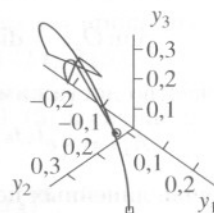


Рис. 6. Центральная траектория $x(t, 0, 5, -3)$.

Векторные поля из правой части системы (10):

$$X_1 = \left(-x_3, -\sqrt{1 - x_1^2 - x_2^2 - x_3^2}, x_1 \right)^T, \quad X_2 = \left(x_2, -x_1, -\sqrt{1 - x_1^2 - x_2^2 - x_3^2} \right)^T$$

удовлетворяют условию (2); даны граничные условия:

$$x^0 = [0,3555263893, 0,2583050417, 0,1359392951],$$

$$x^1 = [0,1381591491, 0, 0,05841275134];$$

$$\text{время } T = 1, \text{ точность } \varepsilon = 10^{-6}.$$

Начальное расстояние между точками x^0, x^1 равно 0,3463818366.

Вычисленные коэффициенты нильпотентной аппроксимации (4) в точке x^1 имеют вид

$$C = \begin{pmatrix} -0,06987008492 & -0,5 \\ 0,5 & -0,0698700849 \end{pmatrix}.$$

Результаты работы программы FindControlLoc на всей библиотеке NilpControls:

Класс управлений $\mathcal{U}$	Число итераций
Optimal	$N = 6$
Trig	$N = 5$
PieceConst	$N = 5$
Centre (1, 4, -1)	$N = 8$
Centre (0, 5, -3)	$N = 4$
Focus	$N = 3$.

Траектории системы (1), соответствующие каждому классу управлений, приведены на рис. 1–6. Траектории типа центр приведены для двух наборов параметров управлений.

4. Системы с трехмерной орбитой

Рассматриваются нелинейные системы вида (1), где $x \in \mathbb{R}^n$, $n > 3$, и задача управления (3). В некоторых случаях эту задачу управления можно решить с применением описанного выше локального алгоритма.

Напомним, что орбита O_{x^0} системы (1), проходящая через точку x^0 , определяется как следующее множество:

$$(11) \quad O_{x^0} = \{e^{t_k f_k} \circ \dots \circ e^{t_2 f_2} \circ e^{t_1 f_1}(x^0) | t_i \in \mathbb{R}, f_i = u_1^i X_1 + u_2^i X_2, i, k = 1, 2, \dots\},$$

где $e^{t_{k+1} f_{k+1}} \circ e^{t_k f_k} = e^{t_{k+1} f_{k+1}}(e^{t_k f_k}(x_{k-1}^0))$, $k = 1, 2, 3, \dots$, — точка, полученная движением вдоль поля f_{k+1} в течение времени t_{k+1} из точки $e^{t_k f_k}(x_{k-1}^0)$, $x_0^0 = x^0$.

Пусть для n -мерной системы выполняется условие $\dim \text{Lie}(X_1, X_2)(x^0) = 3$. Из теоремы об орбите [8] следует, что O_{x^0} — гладкое многообразие, причем

$$(12) \quad \dim O_{x^0} = \dim \text{Lie}(X_1, X_2)(x^0) = 3.$$

Множество достижимости из точки x^0 имеет вид:

$$(13) \quad A_{x^0} = \{e^{t_k f_k} \circ \dots \circ e^{t_2 f_2} \circ e^{t_1 f_1}(x^0) | t_i > 0, f_i = u_1^i X_1 + u_2^i X_2, i, k = 1, 2, \dots\}.$$

В случае линейных по управлению систем при отсутствии ограничений на управление множество достижимости A_{x^0} совпадает с орбитой O_{x^0} . Следовательно, решение задачи управления (1)–(3) при $n > 3$ существует тогда и только тогда, когда $x^1 \in O_{x^0}$. В этом случае задача управления в n -мерном пространстве состояний сводится к задаче управления на трехмерном многообразии O_{x^0} :

$$(14) \quad \dot{x} = u_1 X_1(x) + u_2 X_2(x), \quad x \in O_{x^0}, \quad \dim O_{x^0} = 3, \quad u \in \mathbb{R}^2,$$

с граничными условиями и точностью

$$(15) \quad x^0, x^1 \in O_{x^0}, \quad T > 0, \quad \varepsilon > 0.$$

5. Метод карт

Для решения задачи (14), (15) построим алгоритм, использующий локальный алгоритм (см. раздел 3) и метод теории гладких многообразий — метод карт. Суть его заключается в следующем. Так как орбита является трехмерным многообразием, то можно построить покрытие орбиты открытыми трехмерными областями и на каждой области ввести локальные координаты (карты). Тогда задача на многообразии будет сведена к серии задач на открытых областях трехмерного евклидова пространства, что делает возможным применение локального алгоритма.

Алгоритм 1.

- 1) Строится окрестность $W_{x^0} \subset O_{x^0}$ точки x^0 и локальные координаты $\varphi: \mathbb{R}^3 \supset B_{y^0}^3 \rightarrow W_{x^0}$ такие, что $W_{x^0} \ni x = \varphi(y)$, $y \in B_{y^0}^3 = \{y \in \mathbb{R}^3 \mid \|y\| < 1\}$, $y^0 = \varphi^{-1}(x^0)$.
- 2) В системе (14) делается замена переменных $x = \varphi(y)$:

$$(16) \quad \dot{y} = u_1 Y_1(y) + u_2 Y_2(y), \quad u \in \mathbb{R}^2, \quad y \in B_{y^0}^3,$$

и пересчитываются краевые условия:

$$(17) \quad y^0, y^1 \in B_{y^0}^3, \quad T > 0, \quad \varepsilon > 0.$$

3) Если $x^1 \in W_{x^0}$, то $y^1 = \varphi^{-1}(x^1) \in B_{y^0}^3$; в этом случае:

- а) если $|y^0 - y^1|$ достаточно мало, то непосредственно применяется локальный алгоритм и вычисляются управления $U_{B_{y^0}^3}(t)$;
 - б) если $|y^0 - y^1|$ велико, то задача (16), (17) является глобальной: вводятся промежуточные узлы $p_1, \dots, p_m \in B_{y^0}^3$, задача сводится к $m+1$ локальной задаче и полученные локальные управления $u^i(t)$, $i = 1, \dots, m+1$, объединяются в управление $U_{B_{y^0}^3}(t)$ (см. (9)).
- 4) Если $x^1 \notin W_{x^0}$, то в силу связности O_{x^0} можно построить последовательность промежуточных узлов $q_1, q_2, \dots, q_k \in O_{x^0}$ и соответствующую им последовательность окрестностей $W_{x^0}, W_{q_1}, \dots, W_{q_k} \in O_{x^0}$ так, чтобы $x^0, q_1 \in W_{x^0}$, $q_i, q_{i+1} \in W_{q_i}$, $i = 1, \dots, k-1$, $q_k, x^1 \in W_{q_k}$. В каждой окрестности можно построить локальные координаты $x = \varphi_i(y)$, $i = 0, \dots, k$, и свести задачу (14), (15) к $k+1$ задаче в шарах $B_{y_i}^3 \in \mathbb{R}^3$, $i = 0, \dots, k$. С помощью локального алгоритма можно вычислить управления $U_{B_{y_i}^3}(t)$, $i = 0, \dots, k$, объединить их по принципу (9) в управление $U(t)$, которое будет являться решением задачи (14), (15).

Приведем пример. Рассматривается задача управления (1)–(3) в случае $n = 4$ с полями в правой части системы:

$$(18) \quad X_1(x) = (-x_3, -x_4, x_1, x_2)^T, \quad X_2(x) = (x_2, -x_1, -x_4, x_3)^T,$$

с краевыми условиями и точностью:

$$(19) \quad x^0 = (0, 0, 0, 1), \quad x^1 = (1, 0, 0, 0), \quad T = 1, \quad \varepsilon = 10^{-2}.$$

Известно [10], что орбитой O_{x^0} этой системы является единичная трехмерная сфера S^3 : $x_1^2 + x_2^2 + x_3^2 + x_4^2 = 1$. Очевидно, что x^0, x^1 принадлежат орбите S^3 .

Следуя алгоритму 1, построим карту, содержащую точку x^0 :

$$x_4 = \sqrt{1 - x_1^2 - x_2^2 - x_3^2}, \quad x_1^2 + x_2^2 + x_3^2 < 1.$$

Очевидно, что x^1 не принадлежит этой карте, поэтому построим карту в окрестности точки x^1 :

$$x_1 = \sqrt{1 - x_2^2 - x_3^2 - x_4^2}, \quad x_2^2 + x_3^2 + x_4^2 < 1.$$

Введем промежуточный узел $q_1 = (0, 9, 0, 0, \sqrt{1 - (0,9)^2}) \in S^3$, принадлежащий обеим картам.

Задача (18), (19) сведена к двум трехмерным задачам.

Первая задача:

$$(20) \quad \begin{aligned} \dot{x}_1 &= -x_3 u_1 + x_2 u_2, \\ \dot{x}_2 &= -\sqrt{1 - x_1^2 - x_2^2 - x_3^2} u_1 - x_1 u_2, \\ \dot{x}_3 &= x_1 u_1 - \sqrt{1 - x_1^2 - x_2^2 - x_3^2} u_2, \\ x &= (x_1, x_2, x_3) \in B_1^3 = \{x_1^2 + x_2^2 + x_3^2 < 1\} \end{aligned}$$

с краевыми условиями:

$$(21) \quad (x_1^0, x_2^0, x_3^0) = (0, 0, 0), \quad (x_1^1, x_2^1, x_3^1) = (0, 9, 0, 0), \quad T = 1, \quad \varepsilon = 10^{-2}.$$

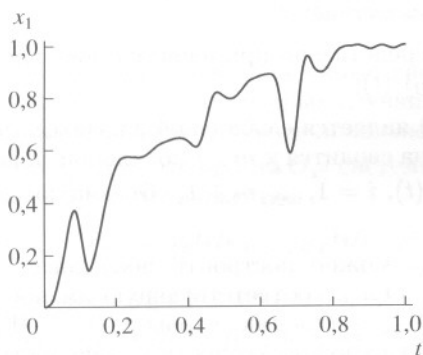


Рис. 7. $x_1(t)$.

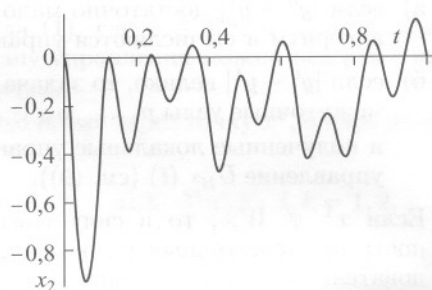


Рис. 8. $x_2(t)$.

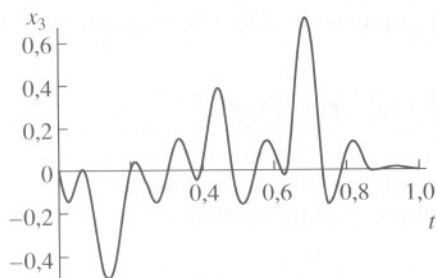


Рис. 9. $x_3(t)$.

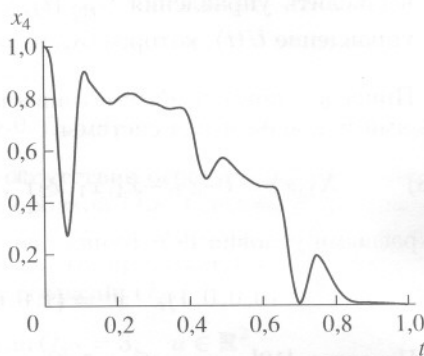


Рис. 10. $x_4(t)$.

Первая задача является глобальной, поэтому вводится промежуточный узел $p_1 = (0,65, 0, 0) \in B_1^3$.

Вторая задача:

$$\begin{aligned}
 \dot{x}_2 &= -x_4 u_1 - \sqrt{1 - x_2^2 - x_3^2 - x_4^2} u_2, \\
 \dot{x}_3 &= \sqrt{1 - x_2^2 - x_3^2 - x_4^2} u_1 - x_4 u_2, \\
 \dot{x}_4 &= x_2 u_1 + x_3 u_2, \\
 x &= (x_2, x_3, x_4) \in B_2^3 = \{x_2^2 + x_3^2 + x_4^2 < 1\}
 \end{aligned}
 \tag{22}$$

с краевыми условиями:

$$\begin{aligned}
 (x_2^0, x_3^0, x_4^0) &= (-0,01479997, -0,01655002, 0,457388015), \\
 (x_2^1, x_3^1, x_4^1) &= (0, 0, 0), \quad T = 1, \quad \varepsilon = 10^{-2}.
 \end{aligned}
 \tag{23}$$

Используя тригонометрические управления в локальном алгоритме, получили решение задачи (18), (19). Графики компонент четырехмерной траектории приведены на рис. 7–10.

6. Метод аппроксимации скобки Ли

Для решения задачи управления (14), (15) построим алгоритм, основанный на свойстве некоммутативности векторных полей в правой части системы (14). Это

свойство отражает следующее соотношение [8] между потоками полей X_1 , X_2 и их коммутатора $X_3 = [X_1, X_2]$:

$$(24) \quad e^{-tX_2} \circ e^{-tX_1} \circ e^{tX_2} \circ e^{tX_1}(x) = e^{t^2 X_3}(x) + O(t^3).$$

Формула (24) конструктивна. Она позволяет организовать перемещение системы (14) вдоль поля X_3 непосредственно с помощью полей X_1 и X_2 посредством управлений вида $u_1 = (\pm 1, 0)$, $u_2 = (0, \pm 1)$.

Для отыскания требуемого временного промежутка строится виртуальная вполне управляемая на орбите система:

$$(25) \quad \dot{x} = u_1 X_1(x) + u_2 X_2(x) + u_3 X_3(x), \quad x \in O_{x^0}, \quad u \in \mathbb{R}^2,$$

с краевыми условиями (15). Тогда, используя управления вида $(0, 0, 1)$, $(0, 1, 0)$, $(1, 0, 0)$, можно построить систему уравнений относительно переменных t_1, t_2, t_3 :

$$(26) \quad e^{t_1 X_1} \circ e^{t_2 X_2} \circ e^{t_3 X_3}(x^0) = x^1$$

при условии

$$(27) \quad x^0, x^1 \in O_{x^0}.$$

Для решения задачи (26), (27) потребуется введение локальных координат на орбите, что сведет n -мерную систему к трехмерной.

Введем некоторые определения: пусть $e^{-T_c/4 X_2} \circ e^{-T_c/4 X_1} \circ e^{T_c/4 X_2} \circ e^{T_c/4 X_1}(q^{i-1}) = q^i$ – элементарный цикл, $i = 1, \dots, N$, $q^0 = x^0$; T_c – время движения по циклу, N – количество циклов, S – общее время движения с помощью циклов. Тогда управление $U_c(t) = (U_{1c}(t), U_{2c}(t))$, $t \in [0, T_c]$ каждого цикла имеет вид:

$$(28) \quad U_c(t) = \begin{cases} (1, 0), & \text{если } t \in [0, T_c/4], \\ (0, 1), & \text{если } t \in [T_c/4, T_c/2], \\ (-1, 0), & \text{если } t \in [T_c/2, 3T_c/4], \\ (0, -1), & \text{если } t \in [3T_c/4, T_c]. \end{cases}$$

Пусть T_1, T_2, T_3 – решение уравнения (26), (27), N – количество циклов.

Алгоритм 2.

- 1) Вычисляется $S = 4\sqrt{T_3 N}$.
- 2) Вычисляется $T_c = S/N$.
- 3) Вычисляется кусочно-постоянное управление:

$$(29) \quad U_{conc}(t) = \begin{cases} U_c(t), & \text{если } t \in [0, S/N], \\ \vdots & \\ U_c\left(t - \frac{N-1}{N}S\right), & \text{если } t \in \left[\frac{N-1}{N}S, S\right], \\ (0, 1), & \text{если } t \in [S, S+T_1], \\ (1, 0), & \text{если } t \in [S+T_1, S+T_1+T_2]. \end{cases}$$

- 4) Вычисляется точка p^N , в которую приходит система (14) за N итераций с помощью управлений (29):
если $|x^1 - p^N| < \varepsilon$, то (29) – решение задачи управления (14), (15);
если $|x^1 - p^N| > \varepsilon$, то необходимо увеличить N .
- 5) Для того чтобы общее время движения было равно T , данному в условии (15), необходимо сделать перепараметризацию времени t .

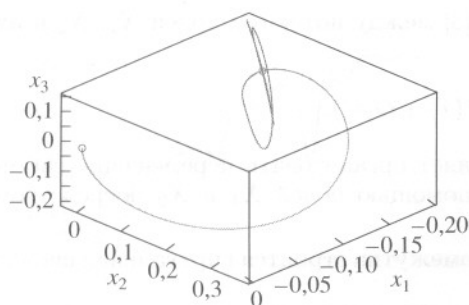


Рис. 11.

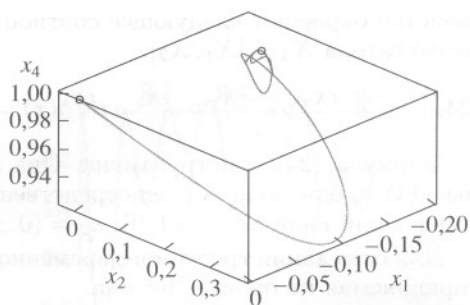


Рис. 12.

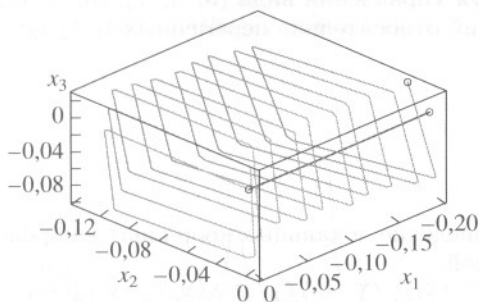


Рис. 13.

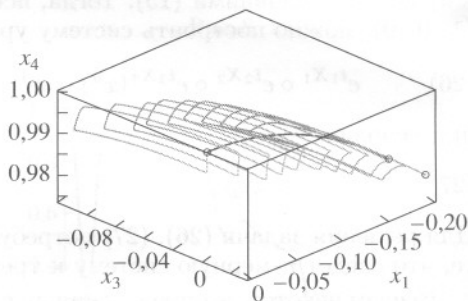


Рис. 14.

Проведем оценку погрешности Δ для алгоритма 2: из (24) следует

$$(30) \quad \Delta = NO \left(\frac{T_3}{N} \right)^{3/2} = \frac{K}{\sqrt{N}}, \quad K - \text{const.}$$

Отсюда видно, что алгоритм 2 сходится со скоростью $N^{-1/2}$.

Приведем следующий пример. Рассматривается задача управления (18) с крайними условиями:

$$(31) \quad x^0 = (0, 0, 0, 1), \quad x^1 = (-0,198669, 0, 0, 0,980067).$$

Несложно проверить, что $x^0 \in S^3$, $x^1 \in S^3$ с вычислительной погрешностью.

Применим к решению задачи (18), (31) алгоритм 1 и алгоритм 2, сравним количество итераций для каждого алгоритма при различных значениях погрешности ε .

Применим алгоритм 1.

Точки x^0 , x^1 принадлежат одной координатной окрестности точки x^0 : нижней полусфере $x_4 = -\sqrt{(1 - x_1^2 - x_2^2 - x_3^2)}$. Исходное расстояние между начальным и конечным состоянием $|x^0 - x^1| = 0,2$. Сведя задачу к трехмерной, можно непосредственно применить локальный алгоритм с тригонометрическими управлениями.

Приведем табл. 1 зависимости количества итераций N_1 алгоритма 1 от заданной точности ε .

На рис. 11 и 12 приведены проекции траектории системы (18), полученные с помощью локального алгоритма, в шар $S^3 \cap \{x_4 = 0\}$ и в шар $S^3 \cap \{x_3 = 0\}$ соответственно.

Применим алгоритм 2.

Таблица 1

ε	$N_1 = N_1(\varepsilon)$
10^{-1}	1
10^{-2}	2
10^{-3}	2
10^{-4}	3

Таблица 2

ε	$N_2 = N_2(\varepsilon)$
10^{-1}	1
10^{-2}	80
10^{-3}	>300

Рассмотрим частный случай траектории виртуальной системы (25) при $(T_1, T_2, T_3) = (0, 0, 0, 1)$. В этом случае система перемещается только вдоль поля X_3 и $e^{T_3 X_3}(x^0) = x^1$.

Приведем табл. 2 зависимости количества итераций N_2 алгоритма 2 от заданной точности ε .

На рис. 13 и 14 приведены проекции траектории системы (18), полученные с помощью метода аппроксимации скобки Ли, в шар $S^3 \cap \{x_4 = 0\}$ и в шар $S^3 \cap \{x_3 = 0\}$ соответственно.

Сравнение полученных результатов показывает, что алгоритм 1 гораздо эффективнее алгоритма 2.

7. Заключение

В настоящей работе продемонстрированы широкие возможности локального алгоритма приближенного решения двухточечной граничной задачи для систем с двумерным линейным управлением. Показана его эффективность в сравнении со стандартным методом аппроксимации скобки Ли. Разработанный алгоритм может быть полезен как для большого числа прикладных задач управления, так и для теории импульсного управления.

Автор выражает благодарность профессору В.И. Гурману за внимание к этой работе.

СПИСОК ЛИТЕРАТУРЫ

1. Bellaïche A., Bellaïche A., Risler J.-J. Eds. Sub-Riemannian geometry. Basel, Switzerland: Birkhäuser. 1996. P. 1–78.
2. Laferriere G., Sussmann H.J., Zexiang Li, J.F. Canny Eds. Nonholonomic Motion Planning. The Kluwer Int. Ser. in Engin. Comput. Sci. 1992. V. 192.
3. Дьяконов В. Maple 6: уч. курс. СПб.: Питер, 2001.
4. Сачкова Е.Ф. Решение задачи управления для нильпотентной системы // Дифференц. уравнения. 2008. № 12. С. 1704–1707.
5. Laumond J.P. Nonholonomic motion planning for mobile robots // Lecture Notes Control Inform. Sci., 1998. 229. Springer, P. 343.
6. Гурман В.И. Принцип расширения в задачах оптимального управления. 2-е изд. М.: Наука, 1997.
7. Дыкта В.А., Самсонов О.Н. Оптимальное импульсное управление с приложениями. М.: Физматлит, 2000.
8. Аграчев А.А., Сачков Ю.Л. Геометрическая теория управления. М.: Физматлит, 2005.
9. Jurdjevic V. Geometric control theory. Cambridge University Press, 1997.
10. Гурман В.И., Сачков Ю.Л. Представление и реализация обобщенных решений управляемых систем с неограниченным годографом // АиТ. 2008. № 4. С. 72–80.
11. Jean F. Lectures on Dynamical and Control Systems. Trieste, 2003.

Статья представлена к публикации членом редколлегии А.П. Курдюковым.

Поступила в редакцию 01.10.2008