

УДК 517.977

## ПАРАЛЛЕЛЬНЫЙ ПРОГРАММНЫЙ КОМПЛЕКС РЕШЕНИЯ НЕГОЛОНОМНЫХ ЗАДАЧ УПРАВЛЕНИЯ

*(Работа выполнена в рамках ФЦП «Исследования и разработки по приоритетным направлениям развития научно-технологического комплекса России на 2007–2013 годы», ГК № 07.514.11.4033)*

А.П. Маштаков

*(Институт программных систем РАН, г. Переславль-Залесский, alexey.mashtakov@gmail.com)*

Описан метод управления нелинейными системами с линейным управлением на основе нильпотентной аппроксимации. Представлен алгоритм приближенного решения конструктивной задачи управления пятимерными системами такого вида в классах кусочно-постоянных и оптимальных управлений для аппроксимирующей системы. Приведенный алгоритм был положен в основу параллельного программного комплекса MotionPlanning235, предназначенного для решения поставленной задачи.

**Ключевые слова:** нильпотентная аппроксимация, оптимальное управление, параллельные алгоритмы и программы.

### PARALLEL SOFTWARE PACKAGE FOR NONHOLONOMIC CONTROL PROBLEMS

*Mashtakov A.P. (Program Systems Institute, Russian Academy of Sciences, Pereslavl-Zalesky, alexey.mashtakov@gmail.com)*

**Abstract.** A motion planning problem for nonlinear five-dimensional systems is considered. Parallel software package MotionPlanning235 was developed to solve this problem in class of piecewise constant and optimal controls. Nilpotent approximation is used to obtain an approximate solution with a necessary precision.

**Keywords:** nilpotent approximation, optimal control, parallel algorithms and programs.

Перевод механической системы в требуемую конфигурацию является одной из важнейших задач робототехники и инженерии: это вывод спутника на орбиту, управление мобильным роботом, вращение твердого тела в руке робота-манипулятора и др. В математике подобная задача формализуется как конструктивная задача управления.

Имеется динамика механической системы, описываемая системой дифференциальных уравнений с функциональными параметрами (управления). Требуется найти управляющие воздействия из заданного класса функций, при применении которых система переходит из начального в заданное целевое состояние.

В работе рассматривается двухточечная задача управления следующего вида:

$$\dot{q} = u_1(t) X_1(q) + u_2(t) X_2(q), \quad (1)$$

$$q(0) = q_0, q(T) = q_1, \quad (2)$$

где пространство состояний  $q \in Q$  – это связное пятимерное гладкое многообразие,  $\dim(Q)=5$ ; управление принимает значения на двумерной плоскости,  $(u_1, u_2) \in \mathfrak{R}$ ; гладкие векторные поля  $X_1(q)$ ,  $X_2(q)$  удовлетворяют условию полного ранга [1] на многообразии  $Q$ . Требуется найти управление  $u(t)=(u_1(t), u_2(t))$ , переводящее систему (1) за время  $T>0$  из начального состояния  $q_0$  в терминальное  $q_1$  с любой ранее заданной точностью  $\varepsilon>0$ , то есть в

такое состояние  $\tilde{q}_1$ , что  $dist(q_1, \tilde{q}_1) < \varepsilon$ , где  $dist$  – расстояние на многообразии  $Q$  (например, если  $Q=\mathfrak{R}^5$ , то  $dist(q_1, \tilde{q}_1) = \sqrt{\sum_{i=1}^5 (q_1^i - \tilde{q}_1^i)^2}$ ).

Системы вида (1) характеризуются тем, что размерность пространства управлений меньше размерности пространства состояний  $\dim(\mathfrak{R}^2) = 2 < \dim(Q)=5$ , однако любые точки пространства состояний могут соединяться траекторией системы. В теории управления такие системы называются вполне неголономными. Нелинейная система (1), линейно зависящая от управлений, количество которых меньше размерности пространства состояний, характеризуется тем, что возможность передвижения по различным направлениям неодинакова. Величина смещения в направлении полей  $X_1(q)$ ,  $X_2(q)$  за малое время  $t$  есть  $O(t)$ , в направлении коммутатора  $X_3(q)=[X_1, X_2](q)-O(t^2)$ , в направлении коммутаторов более высокого порядка  $X_4(q)=[X_1, X_3](q)$  и  $X_5(q)=[X_2, X_3](q)-O(t^3)$ . В силу этой анизотропии пространства состояний задача управления для таких систем весьма нетривиальна.

В последнее время конструктивная задача управления активно изучается в нелинейной теории управления. Удовлетворительное решение она имеет лишь для некоторых специальных классов

систем [2]. Однако пятимерные системы общего положения с двумя управлениями имеют вектор роста (2, 3, 5), потому эти результаты неприменимы к таким системам.

В данной работе представлен способ отыскания приближенного решения задачи (1)–(2), основанный на построении нильпотентной аппроксимации. Идея метода в том, что исходная нелинейная система заменяется приближенной нильпотентной системой, для которой точно решается задача управления. Затем найденные управления подставляются в исходную систему.

Если состояние, достигнутое после применения найденного управления, отличается от желаемого в пределах допустимой погрешности, задача считается решенной, иначе процедура повторяется. Управления, точно решающие нильпотентную систему, дают приближенное решение исходной задачи управления в малой окрестности целевой точки. Метод нильпотентной аппроксимации применим к задачам управления общего вида; существенно только умение решать задачу управления для нильпотентной аппроксимации. Этот метод предложен в [3].

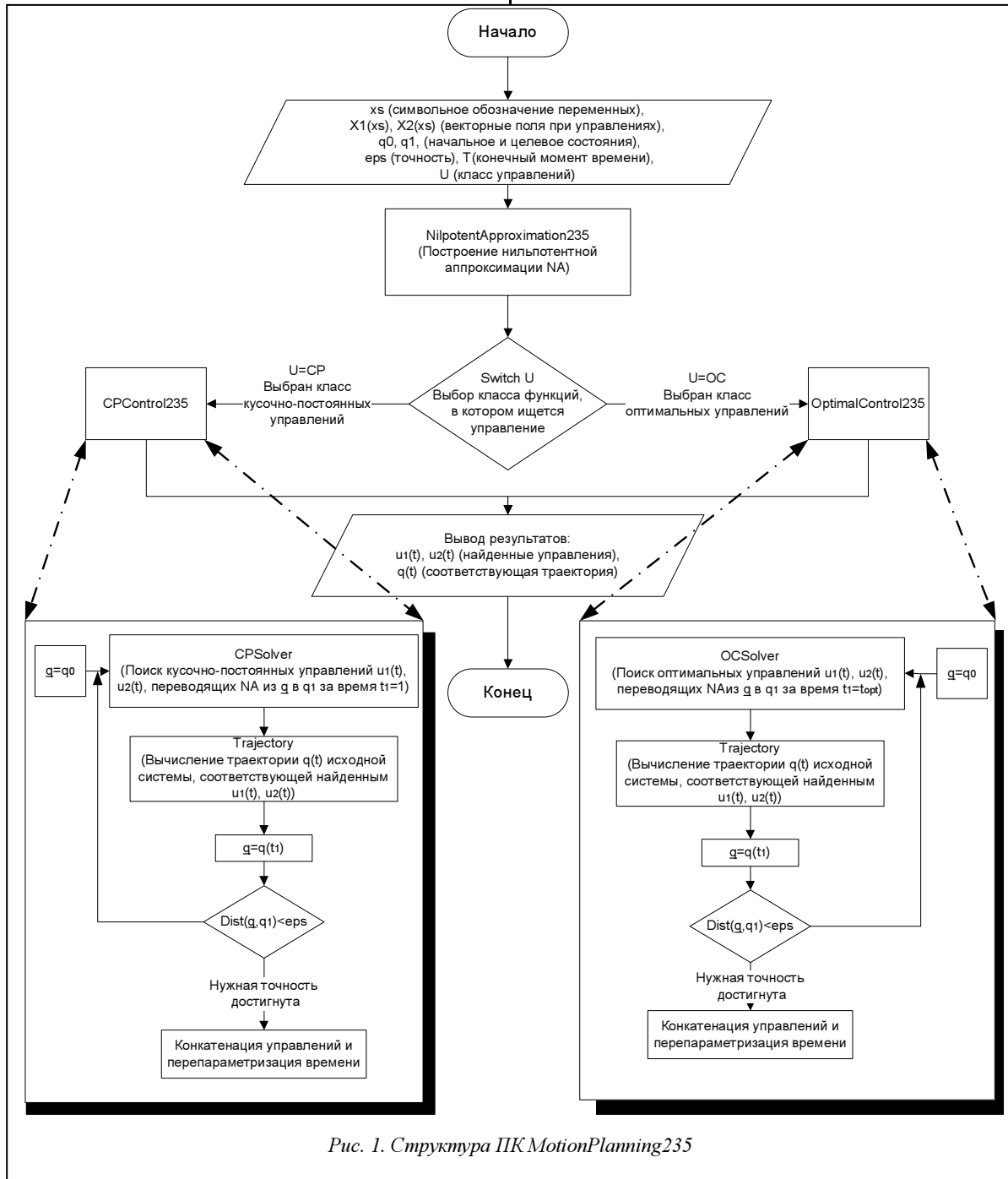


Рис. 1. Структура ПК MotionPlanning235

В данной работе рассматривается задача управления нелинейными пятимерными системами общего вида, линейно зависящими от двухмерного управления. Конкретными примерами систем такого вида, которые в робототехнике имеют самостоятельное значение из-за своего прикладного значения, являются система, моделирующая качение шара по плоскости без прокручивания и проскальзывания, и система, моделирующая движение машины с двумя прицепами по плоскости.

**Алгоритм поиска приближенного управления на основе вычисления траекторий аппроксимирующей системы и его реализация ПК MotionPlanning235.** Итерационный алгоритм приближенного решения задачи (1)–(2) основан на локальном приближении исходной нелинейной системы нильпотентной канонической системой (3), для которой задача управления решается точно на каждой итерации.

Итак, имеется исходная система в начальном состоянии  $q_0$ . Требуется найти управление, переводящее систему в конечное состояние  $q_1$  с предварительно заданной точностью  $\varepsilon$ . Поиск приближенного решения осуществляется следующим образом.

1. В окрестности конечной точки  $q_1$  строится аппроксимирующая система, для которой точно решается задача управления.

2. Найденные управления подставляются в исходную систему. Если достигнутое состояние не попадает в  $\varepsilon$ -окрестность конечного состояния, значит, нужная точность не достигнута, алгоритм повторяется снова для исходной системы, где в качестве начального состояния выбирается состояние, достигнутое на предыдущей итерации алгоритма, иначе вычисления прекращаются.

В среде Mathematica 8 (<http://www.wolfram.com/mathematica/>) был реализован параллельный программный комплекс **MotionPlanning235** (рис. 1) решения задачи (1)–(2). Он является дополнительным пакетом для системы Wolfram Mathematica (MotionPlanning235.m) и состоит из следующих основных модулей:

- **NilpotentApproximation235** строит нильпотентную аппроксимацию NA (см. (3)) системы (1) и возвращает замену переменных  $\tau=A^\circ\Phi$ , в которых NA имеет канонический вид;

- **CPControl235** решает задачу (1)–(2) в классе кусочно-постоянных управлений;

- **OptimalControl235** решает задачу (1)–(2) в классе оптимальных управлений.

**Модуль NilpotentApproximation235 построения нильпотентной аппроксимации.** Локальное приближение управляемой системы другой, более простой, широко используется в теории управления. Обычно в качестве локальной аппроксимации используется линеаризация управляемой системы. Однако для линейных по управлению систем вида (1) линеаризация дает слишком грубое приближе-

ние. Так как размерность управления меньше размерности состояния, линеаризация не может быть вполне управляемой. Естественной заменой линейной аппроксимации в этом случае является нильпотентная аппроксимация – наиболее простая система, сохраняющая структуру управляемости исходной системы (в частности, сохраняется такой важный инвариант, как вектор роста).

В данной работе использован алгоритм вычисления нильпотентной аппроксимации для линейных по управлению систем, предложенный в [2]. Этот алгоритм был конкретизирован для систем вида (1), а именно, вычислена замена переменных для перехода в привилегированные координаты  $A:q \rightarrow z$ , а векторные поля нильпотентной аппроксимации  $\hat{X}_1(z)$ ,  $\hat{X}_2(z)$  в привилегированных координатах  $z_i$ ,  $i=1, \dots, 5$ , были выражены через векторные поля исходной системы  $X_1(z)$ ,  $X_2(z)$  и их производные. Кроме того, алгоритм Беллаиша дополнен следующим образом: выполняется переход в систему координат  $y$  (замена переменных  $\Phi:z \rightarrow y$ ), в которой нильпотентная аппроксимация имеет канонический вид

$$\begin{cases} \dot{y}_1 = u_1, \\ \dot{y}_2 = u_2, \\ \dot{y}_3 = \frac{1}{2}(y_1 u_2 - y_2 u_1), \\ \dot{y}_4 = \frac{1}{2}(y_1^2 + y_2^2) u_2, \\ \dot{y}_5 = -\frac{1}{2}(y_1^2 + y_2^2) u_1, \end{cases} \quad (3)$$

а граничные условия представлены как

$$y(0)=y^0, \quad y(T)=(0, 0, 0, 0, 0). \quad (4)$$

**Модуль CPControl235 решения задачи в классе кусочно-постоянных управлений.** В программном комплексе **MotionPlanning235** реализован модуль **CPControl235** решения задачи управления нильпотентной канонической системой (3) с граничными условиями (4) в классе кусочно-постоянных функций. При этом время можно перепараметризовать так, что  $T=1$ . Требуется найти управление  $(u_1(t), u_2(t))$  в классе кусочно-постоянных функций на отрезке  $t \in [0, 1]$ , переводящие систему из произвольного начального состояния  $y^0 \in \mathcal{R}^5$  в начало координат. Доказано следующее утверждение.

**Предложение.** Для решения задачи управления (3)–(4) достаточно управлений с тремя точками переключения:

$$u_{1,2} = \begin{cases} \alpha_{1,2}, & t \in [0, 0,25), \\ \beta_{1,2}, & t \in [0,25, 0,5), \\ \gamma_{1,2}, & t \in [0,5, 0,75], \\ \delta_{1,2}, & t \in (0,75, 1]. \end{cases}$$

Коэффициенты  $\alpha$ ,  $\beta$ ,  $\gamma$ , и  $\delta$  определяются из системы пяти алгебраических уравнений с восе-

мью неизвестными. Получается трехпараметрическое семейство решений, причем для любого начального состояния  $y^0 \in \mathcal{R}^5$  существует способ зафиксировать свободные параметры так, чтобы получалось решение без особенностей. В модуле **CPControl235** фиксация свободных параметров осуществляется так, чтобы в соответствующей траектории не было больших отклонений, а именно, используется критерий  $\max |u_i| \rightarrow \min$ .

**Модуль OptimalControl235 решения задачи в классе оптимальных управлений.** Для управляемой системы (3) с граничными условиями (4) рассматривается задача оптимального управления с естественным интегральным критерием (функционалом субримановой длины)

$$\int_0^T \sqrt{u_1^2 + u_2^2} dt \rightarrow \min. \quad (5)$$

Эта задача известна в теории управления как обобщенная задача Дидоны, а в субримановой геометрии – как субриманова задача с вектором роста (2, 3, 5). Она имеет богатую историю и была детально теоретически изучена в [4]. Основным результатом теоретических исследований явилось описание структуры экспоненциального отображения и первых времен Максвелла вдоль экстремальных траекторий. На основе этого задача оптимального управления (3)–(5) была сведена к задаче решения пяти алгебраических уравнений в неэлементарных функциях с пятью неизвестными. Явно решить эту систему практически невозможно из-за сложности получившихся формул, поэтому предложено использовать численные методы.

Приведем некоторые результаты из статьи [4], необходимые для дальнейшего изложения. Семейство экстремальных траекторий в задаче оптимального управления параметризуется экспоненциальным отображением  $\exp$ , переводящим пару (вектор сопряженных переменных, время) в соответствующую точку экстремальной траектории. Прообраз  $L$  и образ  $M$  экспоненциального отображения  $\exp: L \rightarrow M$  разбиваются поверхностями разреза на четыре области ( $L = \bigcup_{i=1}^4 L_i$  в прообразе и

$M = \bigcup_{i=1}^4 M_i$  в образе), таких, что  $L_i$  переходит в  $M_i$

и экспоненциальное отображение является диффеоморфизмом на этих областях. В свою очередь, каждая область  $L_i$  разбивается на непересекающиеся множества  $C_1$  и  $C_2$ , в которых экспоненциальное отображение задается разными формулами. Задача построения оптимального синтеза состоит в обращении экспоненциального отображения. При этом разбиение  $L$  на  $L_i$  и  $M$  на  $M_i$  известно, а разбиение на  $C_i$  неизвестно. В задаче (3)–(5) была найдена двухмерная группа симметрий, состоящая из вращений и растяжений. Факторизация задачи по этой группе уменьшает размерность задачи с пяти до трех. Получается система из трех уравне-

ний с тремя неизвестными:

$$\begin{cases} P(u, v, k) = P_1, \\ Q(u, v, k) = Q_1, \\ R(u, v, k) = R_1. \end{cases} \quad (6)$$

Итак, для построения оптимального синтеза в задаче (3)–(5) требуется решить систему (6) относительно  $u, v, k$  при заданной правой части  $P_1, Q_1, R_1$ . Заметим, что функции  $P(u, v, k), Q(u, v, k)$  и  $R(u, v, k)$  выражены в эллиптических функциях Якоби и эллиптических интегралах первого и второго рода и имеют сложную аналитическую запись. Известно, что при любой правой части (за исключением особых множеств меньшей размерности) система (6) имеет единственный корень.

Для решения поставленной задачи был разработан модуль **OptimalControl235**. Схема его работы представлена на рисунке 2. Пользователь легко может организовать параллельное вычисление корня на нескольких ядрах процессора. Для этого требуется указать системе Mathematica 8 (функция недоступна в более ранних версиях) количество ядер и изменить при необходимости функцию **SolveHen** (выбрать, какие алгоритмы решения системы – AlgorithmA, AlgorithmB, AlgorithmC, AlgorithmD – будут выполняться одновременно и сколько именно). Кроме стандартного метода Ньютона (AlgorithmA) и метода хорд (AlgorithmB), пользователю предлагаются гибридные методы (AlgorithmC, AlgorithmD). Проведенное обширное тестирование показало, что в большинстве случаев первым результат выдает AlgorithmA или AlgorithmB, хотя и встречаются ситуации, когда гибридные методы справляются с задачей быстрее. Поэтому по умолчанию предлагается делить доступные ядра поровну между AlgorithmA и AlgorithmB. Отметим, что одновременный запуск одного и того же алгоритма на разных ядрах имеет смысл, так как во всех алгоритмах используется генератор случайных начальных приближений. Так как основное время вычисления занимает выбор удачного начального приближения, использование нескольких ядер дает существенное ускорение.

**Дополнительные функции ПК MotionLanning235.** Помимо основных модулей **NilpotentApproximation235, CPControl235** и **OptimalControl235**, решающих задачу (1)–(2), пользователю предоставляются некоторые дополнительные инструменты для представления результатов и слежения за процессом вычислений. Речь идет о следующих функциях:

1. **TrajectoryNatPar[X1, X2, {u1,u2,T}, q0, xs]** возвращает траекторию  $q(t)$  системы (1), где  $t \in [0, T]$ , соответствующую управлениям  $\{u_1, u_2\}$ , выходящую из точки  $q_0$ .

2. **PlotTrajectoryNatPar[trajectory, q0, q1, T]** строит разным цветом графики пяти компонент заданной траектории  $trajectory(t)$  за время  $t \in [0, T]$ ,

а также отмечает на нем заданные состояния  $q_0$  и  $q_1$ . Эта функция может использоваться для визуальной оценки достигнутого состояния и найденных траекторий. Пользователь может включить режим работы ПК, в котором графики найденных траекторий будут выводиться на экран на каждой итерации.

3. **PlotTrajectoryOtklNatPar[trajectory,  $q_0$ ,  $q_1$ , T]** строит разным цветом графики отклонений пяти компонент заданной траектории  $trajectory(t)$  за

проверки решений, найденных с помощью ПК **MotionPlanning235**.

5. **AnimateBall[trajectory, T,  $q_0$ ,  $q_1$ , N]** визуализирует качение без прокручиваний и проскальзываний сферы по траектории  $trajectory(t)$  (аналогично функции **AnimateCar**).

**Пример использования ПК MotionPlanning235.** Продемонстрируем работу ПК MotionPlanning235 на задаче о перемещении по плоскости машины с двумя прицепами. Состояние системы

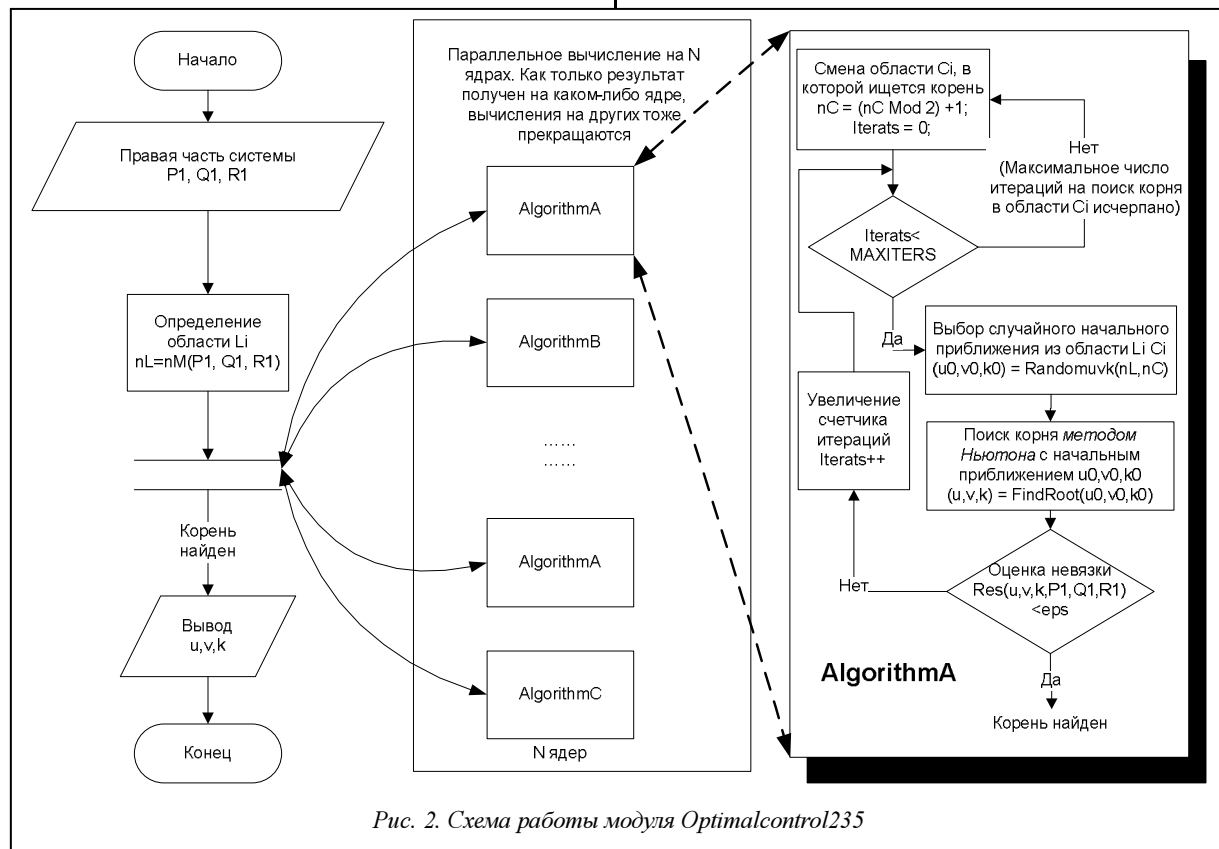


Рис. 2. Схема работы модуля Optimalcontrol235

время  $t \in [0, T]$  от состояния  $q_1$ .

4. **AnimateCar[trajectory, T,  $q_0$ ,  $q_1$ , N, {{xmin, xmax}, {ymin, ymax}}]** строит анимацию движения машины с двумя прицепами при движении по траектории  $trajectory(t)$  за время  $t \in [0, T]$  из состояния  $q_0$ . Область, по которой движется машина с прицепами, ограничивается прямоугольником  $\{\{xmin, xmax\}, \{ymin, ymax\}\}$ . В результате на жестком диске создается последовательность из  $N$  кадров. Последовательность кадров – это упорядоченный набор изображений .png, который впоследствии легко может быть собран в видеофайл стандартными утилитами. Размер изображения является параметром, доступным пользователю (по умолчанию 1024×768 пикселей). Помимо положения машины и прицепов, на каждый кадр пунктиром наносится состояние  $q_1$ . Функция предназначена как для самостоятельного использования (для моделирования системы «машина с двумя прицепами» и ее исследования), так и для

описывается пятью координатами  $(x, y, \theta, \varphi_1, \varphi_2) \in Q = \mathbb{R}^2 \times S^3$ . Здесь  $(x, y) \in \mathbb{R}^2$  – координаты центра машины на плоскости;  $\theta$  – угол, задающий ориентацию машины на плоскости;  $\varphi_1$  – угол, задающий положение первого прицепа относительно машины;  $\varphi_2$  – угол, задающий положение второго прицепа относительно первого прицепа. Динамика системы имеет вид

$$\begin{cases} \dot{x} = \cos(\theta)u_1, \\ \dot{y} = \sin(\theta)u_2, \\ \dot{\theta} = u_3, \\ \dot{\varphi}_1 = -\sin(\varphi_1)u_1 + (-1 - \cos(\varphi_1))u_2, \\ \dot{\varphi}_2 = (\sin(\varphi_1 - \varphi_2) + \sin(\varphi_1))u_1 + \\ + (\cos(\varphi_1 - \varphi_2) + \cos(\varphi_1))u_2. \end{cases} \quad (7)$$

Зададим начальное состояние  $q_0 = (0, 0, \pi/4, \pi/4, -\pi/4)$  и целевое  $q_1 = (-0,252, -0,339, 1,085, 0,514, -1,281)$ , в которое система должна перейти с точ-

ностью  $\varepsilon=10^{-3}$ .

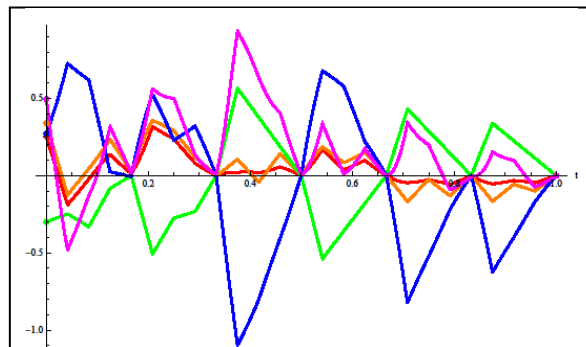


Рис. 3. Результат работы CPControl235

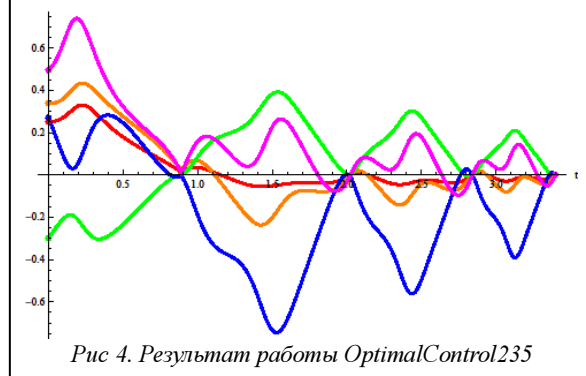


Рис. 4. Результат работы OptimalControl235

На рисунках 3 и 4 приведены результаты работы модулей **CPControl235** и **OptimalControl235**. В случае применения кусочно-постоянных управлений алгоритму потребовалось шесть итераций для достижения требуемой точности, а для оптимальных управлений – три. При использовании кусочно-постоянных управлений система совершает большой маневр (траектория имеет большую

амплитуду отклонений от целевого состояния).

В заключение следует отметить, что рассматриваемый в работе метод приближенного решения двухточечной задачи управления (1)–(2) был реализован в параллельном программном комплексе **MotionPlaning235**. Комплекс испытан на двух прикладных задачах (задача о качении шара по плоскости и задача управления машиной с двумя прицепами). В случаях, когда граничные условия не были слишком далеки, комплекс успешно решал поставленную задачу управления. При далеких граничных условиях алгоритм не сходиллся, что соответствует теоретическому обоснованию метода (нильпотентная аппроксимация является локальным приближением исходной системы). В будущем планируется расширить функционал ПК для решения глобальной задачи управления путем ее сведения к последовательности локальных задач. В настоящее время ПК **MotionPlaning235** является удобным и надежным средством решения локальной задачи (1)–(2).

#### Литература

1. Аграчев А.А., Сачков Ю.Л. Геометрическая теория управления. М.: Физматлит, 2005. 391 с.
2. Bellaïche A. The tangent space in sub-Riemannian geometry // Sub-Riemannian Geometry, A. Bellaïche and J.J. Risler, Eds. Basel, Switzerland: Birkh auser, 1996, pp. 1–78.
3. Laferriere G., Sussmann H.J. A differential geometric approach to motion planning // Nonholonomic Motion Planning, Zexiang Li and J.F. Canny Eds. Basel, Switzerland: Kluwer, 1992.
4. Сачков Ю.Л. Полное описание стратов Максвелла в обобщенной задаче Дидоны // Мат. сб. 2006. Т. 197. № 6. С. 111–160.
5. Laumond J.P. Lecture Notes in Control and Information Science. Springer. 1998. № 229. 343 с.
6. Murray R.M., Sastry S.S. Steering controllable systems, Proc. 29th IEEE Conf. Dec. and Control. Honolulu, Hawaii, 1990, pp. 408–412.