

Применение гибридного мультиагентного метода интерполяционного поиска в задаче о стабилизации спутника

Пантелеев А.В. *, Каранэ М.М.С. **

Московский авиационный институт (национальный исследовательский университет), МАИ, Волоколамское шоссе, 4, Москва, А-80, ГСП-3, 125993, Россия

*e-mail: avpanteleev@inbox.ru

**e-mail: mmkarane@mail.ru

Статья поступила 27.02.2021

Аннотация

В статье рассматривается решение задачи о стабилизации спутника с помощью алгоритма поиска оптимального программного управления, основанного на применении разложений по системам базисных функций и мультиагентного метода условной оптимизации. В частности, в качестве базисных функций используются кусочно-постоянные, кусочно-линейные, квадратичные и кубические сплайны, а в качестве мультиагентного метода используется гибридный мультиагентный метод интерполяционного поиска. Этот метод предложен для оптимизации функций многих переменных и основан на построении интерполяционных кривых Безье, Катмулла–Рома и В-сплайнов на основе информации о положении агентов, образующих текущую популяцию, и решении задачи одномерной параметрической оптимизации для реализации исследуемого и фронтального поиска, а также использовании идей роевого интеллекта и миграционных алгоритмов. На основе предложенных алгоритмов разработано

программное обеспечение и исследована его эффективность при решении задачи стабилизации положения спутника, управляемого двигателями. Проведен сравнительный анализ влияния выбора базисной системы и параметров интерполяционного метода на точность решаемой прикладной задачи.

Ключевые слова: оптимальное программное управление, мультиагентные алгоритмы, оптимизация, программное обеспечение, стабилизация спутника.

Введение

В современной математике большой интерес представляют задачи поиска оптимального управления, возникающие во многих отраслях науки, авиационной и ракетно-космической технике [1-4]. Одним из самых значимых результатов в теории оптимального управления является принцип максимума, который может быть применен к большинству известных задач данного класса. При решении прикладных задач принцип максимума наиболее актуален для систем, в которых программное управление является релейным. С точки зрения получения приближенных решений для нелинейных систем общего вида основной проблемой является нахождение решения соответствующей двухточечной краевой задачи. Поэтому разработка прямых методов нахождения оптимального программного управления остается актуальной.

Можно выделить несколько подходов к численному решению прикладных задач оптимального управления, которые в большинстве случаев нельзя решить аналитически.

Говоря о численных алгоритмах решения задачи оптимального управления [5-8], можно рассмотреть две большие группы методов: прямые методы и косвенные методы.

В косвенных методах используется вариационное исчисление, чтобы определить условия оптимальности первого порядка задачи оптимального управления. Решение задачи оптимального управления сводится к решению многоточечной краевой задачи. Одним из основных косвенных алгоритмов считается метод стрельбы. Кроме того, можно выделить метод множественной стрельбы и косвенный метод коллокаций [5].

Прямые методы принципиально отличаются от косвенных. В них траектория и/или управление исходной задачи оптимального управления аппроксимируются некоторым подходящим образом. Если аппроксимируется только управление, то такой метод называется методом параметризации управления. Данный подход используется, например, в прямом методе стрельбы и прямом методе множественной стрельбы. Если аппроксимируются и траектория, и управление, то метод называется методом параметризации траектории и управления. Этот способ применяется в методе локальных коллокаций и псевдоспектральном методе. Решение задачи оптимального управления в описанных случаях аппроксимации сводится к задаче нелинейного программирования.

В статье рассматривается алгоритм поиска оптимального управления одним классом нелинейных систем, основанный на параметрической оптимизации закона управления с использованием нового мультиагентного алгоритма интерполяционного поиска.

Мультиагентные алгоритмы [9-12] успешно применяются во многих областях науки, например, в оптимизации технических систем [13], в оптимальном управлении, машинном обучении, различных областях искусственного интеллекта и в других областях науки и техники. В таких алгоритмах используются процессы, основанные на взаимодействии группы элементов, которые способны обмениваться информацией и выполнять определенные действия, ведущие к получению результата. Мультиагентные алгоритмы довольно просты в применении и показывают приемлемые результаты за разумное вычислительное время. Хотя эти алгоритмы не имеют строгого обоснования и не гарантируют сходимости, но, тем не менее, они успешно применяются на практике.

Постановка задачи поиска оптимального программного управления

Пусть поведение модели объекта управления описывается обыкновенным дифференциальным уравнением

$$\dot{x}(t) = f(t, x(t), u(t)) , \quad (1)$$

где x – вектор состояния системы, $x = (x_1, \dots, x_n)^T \in R^n$; u – вектор управления, $u = (u_1, \dots, u_q)^T \in U \subseteq R^q$, U – некоторое заданное множество допустимых значений управления, определяемое прямым произведением отрезков $[a_1, b_1] \times \dots \times [a_q, b_q]$;

$t \in T = [t_0, t_1]$ – промежуток времени функционирования системы; моменты начала процесса t_0 и окончания t_1 заданы; $f(t, x, u)$ – непрерывная вектор-функция; R^n – n -мерное евклидово пространство.

Начальное условие $x(t_0) = x_0$ задает начальное состояние системы.

Определим множество допустимых процессов $\mathbf{D}(t_0, x_0)$ как множество пар $d = (x(\cdot), u(\cdot))$, которые включают траекторию $x(\cdot)$ и управление $u(\cdot)$ (где $\forall t \in T: x(t) \in R^n, u(t) \in U$, функции $x(\cdot)$ непрерывны и кусочно-дифференцируемы, а $u(\cdot)$ кусочно-непрерывны), удовлетворяющие уравнению (1) с заданным начальным условием.

На множестве $\mathbf{D}(t_0, x_0)$ определим функционал качества управления

$$I(d) = F(x(t_1)), \quad (2)$$

где $F(x)$ – заданная непрерывная функция.

Требуется найти такую пару $d^* = (x^*(\cdot), u^*(\cdot)) \in \mathbf{D}(t_0, x_0)$, что

$$I(d^*) = \min_{d \in \mathbf{D}(t_0, x_0)} I(d).$$

Для систем (1), в которых структура оптимального программного управления согласно принципу максимума является релейной, предлагается искать приближенное решение в виде функции насыщения, которая должна гарантировать выполнение заданных ограничений на управление параллелепипедного вида. Аргументы функции насыщения предлагается искать в виде линейной комбинации заданных базисных функций, которые используются в спектральном методе анализа и синтеза нелинейных систем управления [14-16]. Предварительно с помощью

линейных преобразований промежутков функционирования системы $[t_0, t_1]$ приводится к стандартному отрезку $[0, 1]$.

Алгоритм поиска оптимального программного управления

Шаг 1. Инициализация. Задать параметры мультиагентного алгоритма. Задать начальный масштаб усечения по времени $L=1$, промежуток возможных значений коэффициента в разложении $c_0 \in [c_{0_1}, c_{0_2}]$, $\varepsilon = 5\%$.

Шаг 2. Сгенерировать начальную популяцию, состоящую из NP особей, определяемых коэффициентами c_i в разложении $g(t) : \{c_0^j, c_1^j, \dots, c_{L-1}^j\}$, $j \in \overline{1, NP}$, где $c_i \in [c_{i_1}, c_{i_2}]$, $i \in \overline{0, L-1}$.

Шаг 3. По сгенерированным коэффициентам сформировать управление в виде функции насыщения sat , гарантирующей выполнение ограничений на управление:

$$u_j^m(t) = \text{sat} \{g_j(t)\}, j \in \overline{1, q}; m \in \overline{1, NP},$$

$$\text{где } \forall t \in T \quad \text{sat } g_j(t) = \begin{cases} a_j, & g_j(t) \leq 0, \\ b_j, & g_j(t) > 0, \end{cases} \quad g_j(t) = \sum_{i=0}^{L-1} c_i^j S(i, t).$$

В качестве базисной функций $S(i, t)$, можно использовать сплайны:

$$S(i, t) = Sp^* \left(\frac{t}{h} - i \right), \quad h = \frac{1}{L-1}, \quad i = 0, 1, \dots, L-1 - \text{финитные функции, порожденные}$$

сплайнами, на отрезке $[0, 1]$:

$$Sp^*(t) = \begin{cases} 2^{p-1}(1+t)^p, & t \in [-1, -\frac{1}{2}], \\ 1 - 2^{p-1}|t|^p, & t \in [-\frac{1}{2}, \frac{1}{2}], \\ 2^{p-1}(1-t)^p, & t \in [\frac{1}{2}, 1], \\ 0, & t \notin [-1, 1], \end{cases}$$

где $Sp^*(t)$ при $p = 0$ задает кусочно-постоянный сплайн, при $p = 1$ – кусочно-линейный (крышки), при $p = 2$ – квадратичный, при $p = 3$ – кубический.

Шаг 4. Проинтегрировать NP систем дифференциальных уравнений (1) с управлениями $u^1(t), \dots, u^{NP}(t)$, например, методом Рунге-Кутты 4-го порядка. Для каждого агента получить соответствующие траектории $(x_1^1(\cdot), \dots, x_1^{NP}(\cdot)), \dots, (x_n^1(\cdot), \dots, x_n^{NP}(\cdot))$ и вычислить значения функционала I^1, \dots, I^{NP} .

Шаг 5. Выполнить очередную итерацию выбранного метода минимизации функционала (2). Получить новые положения особей $1', \dots, NP'$ (векторы значений коэффициентов разложения по системе базисных функций). Перейти к шагу 3.

Шаг 6. Цикл (шаг 3 – шаг 5) заканчивается по достижению заданного числа итераций. Выбирается лучшая особь (набор коэффициентов c_i). Соответствующие ей управление и траектория, а также значение функционала $I_{c_i}^*$ принимаются за приближенное решение задачи с найденными коэффициентами $c_i^{*L}, i \in \overline{0, L-1}$ при фиксированном масштабе усечения L .

Шаг 7. Если $I_{c_i^L}^* < I_{c_i^{L-1}}^*$ и $\delta = \frac{|I_{c_i^L}^* - I_{c_i^{L-1}}^*|}{|I_{c_i^{L-1}}^*|} \cdot 100\% > \varepsilon$ (условия проверяются при

$L \geq 1$), то положить $L = L + 1$ и перейти к шагу 2. Если $I_{c_i^L}^* \geq I_{c_i^{L-1}}^*$, процедура поиска оптимального программного управления завершается и выбирается управление с коэффициентами c_i^{*L-1} .

Стратегия поиска решения гибридным мультиагентным методом интерполяционного поиска

Метод основан на процессах, происходящих в среде, имеющей множество агентов. Агенты имеют возможность обмениваться информацией для того, чтобы найти решение задачи оптимизации.

Стратегия поиска этого гибридного алгоритма включает интерполяционный поиск, который использует несколько точек текущей популяции и сводит задачу нахождения новых решений к задачам одномерной параметрической минимизации, метод роевого интеллекта для параметрической минимизации вдоль интерполяционной кривой [17] и самоорганизующийся метод миграции (self-organizing migrating algorithm, SOMA) [18].

Минимизируемая целевая функция $f(x)$ называется функцией приспособленности, а вектор параметров $x = (x_1, x_2, \dots, x_n)^T$ целевой функции – агентом. Предполагаемым решением поставленной задачи оптимизации может быть каждый вектор из множества допустимых решений, т.е. $x = (x_1, x_2, \dots, x_n)^T \in D = [a_1, b_1] \times \dots \times [a_n, b_n]$. При решении задачи используются

конечные наборы $I = \{x^j = (x_1^j, x_2^j, \dots, x_n^j)^T, j = 1, 2, \dots, NP\} \subset D$ возможных решений (агентов), называемые популяциями, где x^j – агент с номером j , а NP – размер популяции.

Гибридный мультиагентный метод интерполяционного поиска [19] имитирует эволюцию начальной популяции $I_0 = \{x^j, j = 1, 2, \dots, NP \mid x^j = (x_1^j, x_2^j, \dots, x_n^j)^T \in D\}$ и представляет собой итерационный процесс, исследующий множество D .

Процедура поиска решения начинается с генерации начальной популяции агентов $x^j, j = \overline{1, NP}$ на множестве D , используя равномерный закон распределения.

Первой фазой поиска является исследующий интерполяционный поиск. Для его реализации в популяции выбираются четыре агента P_1, P_2, P_3, P_4 , среди них лидер $P_1 = x^{(1)}$, а в качестве P_2, P_3, P_4 – три случайных агента популяции, отличных друг от друга и от $x^{(1)}$. Для их обработки используется кривая Безье. Она проходит внутри выпуклой оболочки, образованной выбранными четырьмя точками (агентами). При значении параметра $t = 0$ она проходит через P_1 , а при $t = 1$ через точку P_4 (в этом разделе t – параметр). Далее находится решение задачи параметрической оптимизации

$$x^{\text{Безье}4} = \arg \min_{t \in [0,1]} f[(1-t)^3 P_1 + 3(1-t)^2 t P_2 + 3(1-t) t^2 P_3 + t^3 P_4],$$

и популяция пополняется новым агентом $x^{\text{Безье}4}$.

Для исследования новых областей применяется В-сплайновая кривая, которая формируется по четырем случайным агентам популяции P_1, P_2, P_3, P_4 , отличным друг

от друга, $x^B = \arg \min_{t \in [0,1]} f \left[\frac{1}{2} [-t(1-t)^2 P_1 + (2 - 5t^2 + 3t^3) P_2 + t(1 + 4t - 3t^2) P_3 - t^2(1-t) P_4] \right]$.

Она, как правило, не проходит ни через одну точку, лежит в выпуклой оболочке, порожденной четырьмя вершинами. В результате популяция пополняется еще одним новым агентом x^B .

Второй фазой поиска является миграция популяции [18]. При этом в популяции выбирается агент-лидер (наилучшее решение) $x^{(1)}$. Все остальные агенты популяции $x^{(j)}, j = 2, \dots, NP$ двигаются по направлению к лидеру, совершая дискретные шаги, число которых равно $nstep$, причем половина этих шагов делается к лидеру, а затем в том же направлении совершается еще столько же шагов. Новое положение агента популяции определяется наилучшим достигнутым в ходе этого поиска решением. Направление поиска задается вектором $PRTVector$, координаты которого равны нулю или единице. Если координата равна нулю, поиск по данной координате не проводится, а если единице, то проводится. Таким образом, решение задачи ищется по всем координатам вектора $PRTVector$, равным одновременно единице. Положение лидера в процессе миграции не изменяется.

Третьей фазой метода является фронтальный поиск, который служит для уточнения итогового решения задачи. При этом используются интерполяционные кривые, для формирования которых используется информация о положении трех первых или четырех лидеров среди агентов популяции.

Среди агентов популяции $x^{(1)}, \dots, x^{(NP)}$, расположенных по возрастанию соответствующего им значения функции приспособленности выбираются три лидера $P_1 = x^{(1)}, P_3 = x^{(2)}, P_2 = x^{(3)}$, по которым формируется кривая Безье (при $t=0$ она проходит через P_1 , а при $t=1$ через точку P_3). Далее решается задача параметрической минимизации $x^{\text{Безье3}} = \arg \min_{t \in [0,1]} f[(1-t)^2 P_1 + 2(1-t)t P_2 + t^2 P_3]$, а популяция в итоге пополняется новым агентом $x^{\text{Безье3}}$.

Для продолжения поиска выбираются два лидера, через которых проходит интерполяционная кривая Катмулла–Рома (Catmull-Rom), а также два последующих решения: $P_1 = x^{(3)}, P_2 = x^{(1)}, P_3 = x^{(2)}, P_4 = x^{(4)}$ (при $t=0$ проходит через P_2 , а при $t=1$ через P_3). Далее решается задача параметрической минимизации

$$x^{\text{CR}} = \arg \min_{t \in [0,1]} f \left[\frac{1}{2} [-t(1-t)^2 P_1 + (2-5t^2+3t^3) P_2 + t(1+4t-3t^2) P_3 - t^2(1-t) P_4] \right],$$

а популяция пополняется новым агентом x^{CR} .

Для аналогичного поиска используется кривая Безье, формируемая по четырем лидерам популяции $P_1 = x^{(1)}, P_4 = x^{(2)}, P_2 = x^{(3)}, P_3 = x^{(4)}$ (при $t=0$ она проходит через P_1 , а при $t=1$ через точку P_4). Далее решается задача параметрической минимизации

$$x^{\text{Безье4}} = \arg \min_{t \in [0,1]} f [(1-t)^3 P_1 + 3(1-t)^2 t P_2 + 3(1-t) t^2 P_3 + t^3 P_4],$$

и популяция пополняется новым агентом $x^{\text{Безье4}}$.

Для фронтального поиска может быть использована В-сплайновая кривая, которая формируется по четырем лидерам популяции

$P_1 = x^{(1)}, P_4 = x^{(2)}, P_2 = x^{(3)}, P_3 = x^{(4)}$. Кривая не проходит ни через одну выбранную точку, но лежит в выпуклой оболочке, порожденной этими вершинами. Далее находится решение задачи параметрической оптимизации

$$x^B = \arg \min_{t \in [0,1]} f \left[\frac{1}{2} [-t(1-t)^2 P_1 + (2 - 5t^2 + 3t^3) P_2 + t(1 + 4t - 3t^2) P_3 - t^2(1-t) P_4] \right].$$

В результате популяция пополняется еще одним новым агентом x^B .

Процедура параметрической одномерной минимизации вдоль интерполяционных кривых выполняется на первой и третьей фазах поиска. Для этого используется биоинспирированный метод роевого интеллекта – метод, имитирующий поведение популяции криля [12, 20, 21]. Данный алгоритм основан на результатах анализа поведения стай криля – рачков, внешне напоминающих креветок. Их позиции меняются под действием трех факторов: присутствия других членов популяции, необходимости поиска пищи, случайных блужданий. Обычно движение популяции криля определяется двумя целями: увеличением плотности криля и достижением пищи.

В начале процесса поиска минимума генерируется популяция криля из NP' особей на интервале $t \in [0,1]$ с помощью равномерного закона распределения.

Предполагается, что движение j -го члена популяции криля происходит согласно

уравнению $\frac{dx^j}{dt} = V^j$, где x^j – положение, а V^j – скорость, которая складывается из

трех составляющих. Первая составляющая определяется влиянием соседей (членов популяции, входящих в некоторую окрестность j -го элемента определенного

радиуса), наилучшего элемента во всей популяции и информации о своей старой скорости. Вторая составляющая определяется движением в сторону источника пищи (за него принимается «центр масс» популяции), информацией о старой скорости в поисках пищи, памятью своего наилучшего результата за все итерации. Третья составляющая имитирует случайные блуждания особи, уменьшающиеся с ростом числа итераций. Для оживления процесса поиска применяются операции скрещивания и мутации, используемые в других эволюционных методах и методе дифференциальной эволюции. Процедура поиска минимума вдоль интерполяционной кривой завершается при достижении заданного числа итераций. В качестве ответа из последней популяции криля выбирается особь, которой соответствует наименьшее значение целевой функции вдоль параметрической кривой, а исходная популяция пополняется новым членом.

Задача одномерной параметрической минимизации может быть решена и классическими методами одномерной минимизации: дихотомии, золотого сечения, Фибоначчи и т.д. [22].

Гибридный мультиагентный метод интерполяционного поиска заканчивает работу после того, как будет исчерпано заданное число итераций. Агент с наименьшим значением целевой функции в последней популяции принимается в качестве приближенного решения.

Постановка задачи о стабилизации спутника

Рассматривается задача гашения вращательного движения спутника с помощью установленных на нем двигателей [23, 24]. Движение твердого тела

относительно центра инерции после перехода к безразмерным переменным, имеет вид:

$$\begin{aligned}\dot{p}(t) &= u_1/6, \\ \dot{q}(t) &= u_2 - 0,2rp, \\ \dot{r}(t) &= 0,2(u_3 + pq),\end{aligned}$$

где p, q, r – проекции угловой скорости на главные центральные оси инерции, а u_1, u_2, u_3 – управления, которые характеризуют тяги двигателей, расположенных на спутнике. Ограничение на управление: $U = [-200, 200] \times [-200, 200] \times [-200, 200]$.

Начальное состояние системы задано: $t_0 = 0, p(t_0) = 24, q(t_0) = 16, r(t_0) = 16$. В момент окончания функционирования системы $t_1 = 1$ должны выполняться условия: $p(t_1) = q(t_1) = r(t_1) = 0$.

Функционал качества управления характеризует затраты топлива при работе реактивных двигателей и степень выполнения конечных условий:

$$I_1(x_0, d) = \underbrace{\int_{t_0}^{t_1} [|u_1(t)| + |u_2(t)| + |u_3(t)|] dt}_{I(x_0, d)} + R_1 p(t_1)^2 + R_2 q(t_1)^2 + R_3 r(t_1)^2,$$

где R_i – параметры штрафа. Преобразуем задачу Больца к задаче Майера (2) с помощью введения дополнительного дифференциального уравнения:

$$\dot{s}(t) = |u_1(t)| + |u_2(t)| + |u_3(t)|, \quad s(t_0) = 0,$$

$$I_1(x_0, d) = s(t_1) + R_1 p(t_1)^2 + R_2 q(t_1)^2 + R_3 r(t_1)^2. \quad (3)$$

Требуется найти оптимальное управление и соответствующую ему траекторию, минимизирующие функционал (3) при заданных начальных условиях.

Анализ полученных результатов и выводы

На основе описанных алгоритмов было разработано программное обеспечение, позволяющее находить оптимальное программное управление для описанного класса задач. Результатом работы программы является значение критерия (2) и значения координат вектора состояния в конечный момент времени функционирования системы. Помимо этого, вычисляются наилучшие значения коэффициентов в разложении по базисным функциям, найденные в результате поиска при помощи гибридного мультиагентного алгоритма интерполяционного поиска. По окончании поиска оптимального управления программное обеспечение позволяет изобразить графики траекторий и управления. Задача решается при использовании разных типов сплайнов: кусочно-постоянного, кусочно-линейного, квадратичного и кубического. В связи с этим проводится сравнительный анализ результатов, вычисленных в этих четырех случаях, а также решение сравнивается с известным, полученным в [23]. Как следует из постановки задачи, требуется задать значения коэффициентов штрафной функции, чтобы выполнилось конечное условие $p(t_1) = q(t_1) = r(t_1) = 0$. При решении задачи были подобраны следующие коэффициенты: $R_1 = R_2 = R_3 = 10000$.

Исходя из вышесказанного, рассмотрим четыре случая поиска оптимального программного управления. Первый случай – в качестве базисных функций взят кусочно-постоянный сплайн. Найдено наилучшее количество коэффициентов в разложении $L_1 = 8$, $L_2 = 8$, $L_3 = 2$ для u_1, u_2 и u_3 , соответственно. Также определены наилучшие параметры гибридного мультиагентного алгоритма интерполяционного

поиска [19]: $NP = 40$; $I_{\max} = 400$; $M_1 = 2$; $M_2 = 5$; $PRT = 0,01$; $nstep = 5$; $b_2 = 8$.

Результаты численного эксперимента приведены в таблице 1, а на рисунке 2 изображены графики траекторий p, q, r и управления u_1, u_2, u_3 .

Таблица 1.

Результаты работы алгоритма для кусочно-постоянного сплайна

Координаты точек $(p(t_1), q(t_1), r(t_1))$	(0,0004; -0,0001; 0,0002)
Коэффициенты разложения для u_1	(-145,67; -134,49; -145,8; -146,74; -146,67; -147,08; -147,75; -133,25)
Коэффициенты разложения для u_2	(-41,2; -25,03; 1,07; 38,01; 24,78; 43,53; 32,16; 41,87)
Коэффициенты разложения для u_3	(0,03; 0,05)
Значение функционала I	169,42002

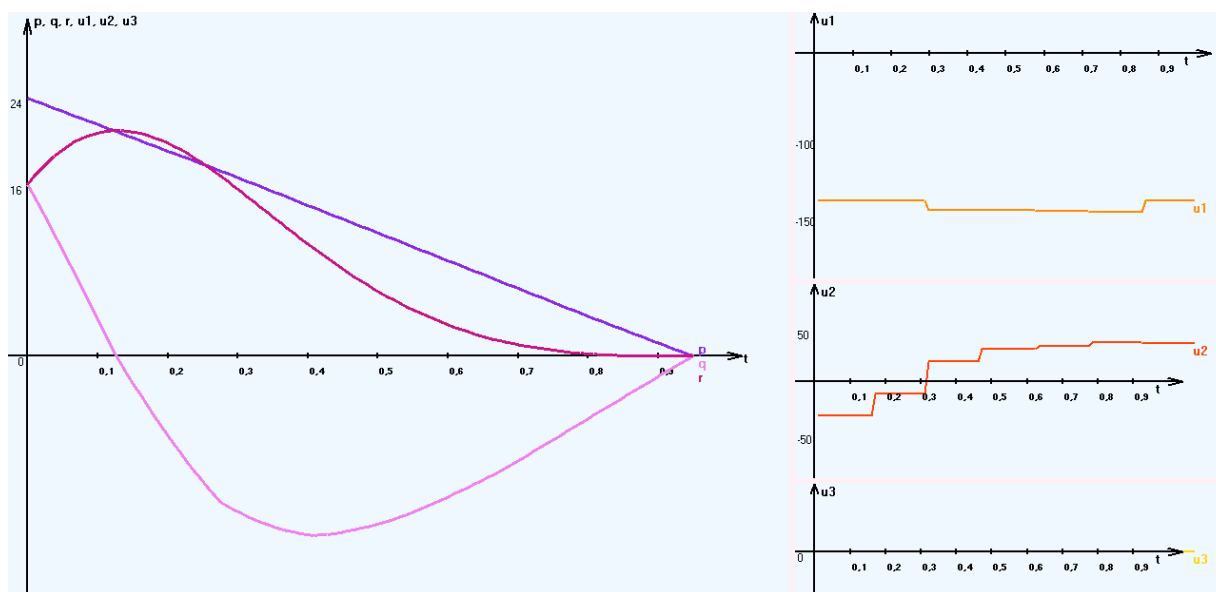


Рисунок 2. Графики траекторий и управления в случае кусочно-постоянного сплайна

Второй случай – в качестве базисных функций взят кусочно-линейный сплайн.

Найдено наилучшее количество коэффициентов в разложении $L_1 = 8$, $L_2 = 8$, $L_3 = 2$

для u_1, u_2 и u_3 , соответственно. Также определены наилучшие параметры гибридного мультиагентного алгоритма интерполяционного поиска [19]: $NP = 40$; $I_{\max} = 400$; $M_1 = 2$; $M_2 = 5$; $PRT = 0,01$; $nstep = 5$; $b_2 = 8$. Результаты численного эксперимента приведены в таблице 2, а на рисунке 3 изображены графики траекторий p, q, r и управления u_1, u_2, u_3 .

Таблица 2.

Результаты работы алгоритма для кусочно-линейного сплайна

Координаты точек $(p(t_1), q(t_1), r(t_1))$	(0,0052; 0,0002; -0,0023)
Коэффициенты разложения для u_1	(-145,78; -127,67; -148,05; -144,31; -148,19; -147,76; -148,31; -141,14)
Коэффициенты разложения для u_2	(-28,8; -38,16; 10,5; 42,61; 20,38; 40,27; 35,26; 32,15)
Коэффициенты разложения для u_3	(0,09; 0,02)
Значение функционала I	169,42325

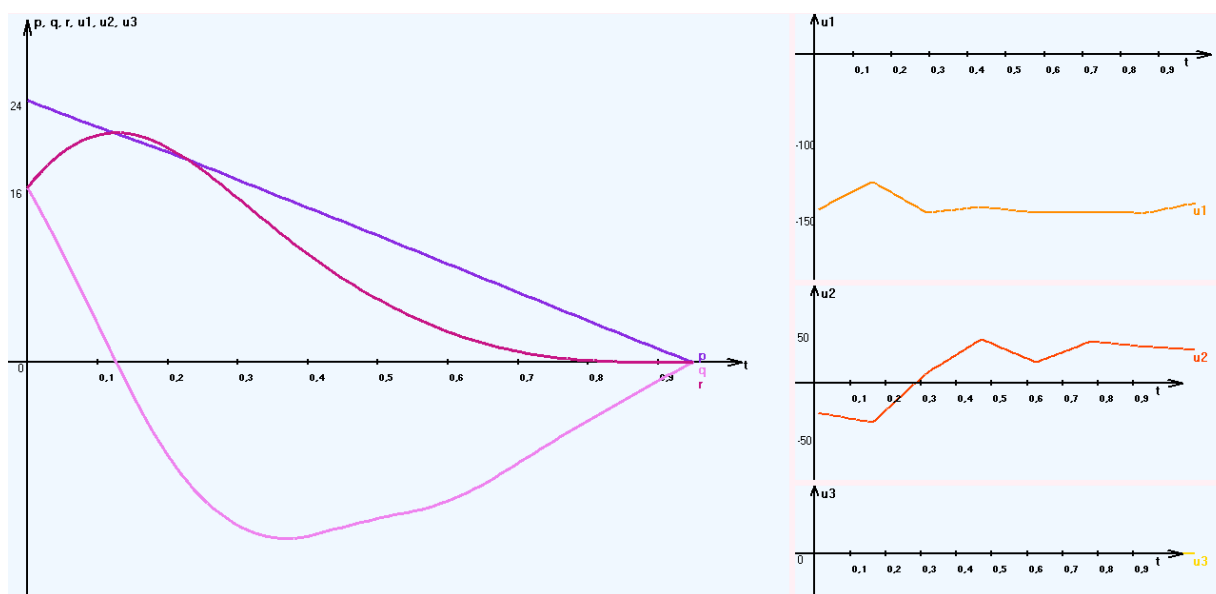


Рисунок 3. Графики траекторий и управления в случае для кусочно-линейного сплайна

Третий случай – в качестве базисных функций взят квадратичный сплайн. Найдено наилучшее количество коэффициентов в разложении $L_1 = 4$, $L_2 = 3$, $L_3 = 2$ для u_1, u_2 и u_3 , соответственно. Также определены наилучшие параметры гибридного мультиагентного алгоритма интерполяционного поиска [19]: $NP = 40$; $I_{\max} = 600$; $M_1 = 2$; $M_2 = 5$; $PRT = 0,01$; $nstep = 5$; $b_2 = 8$. Результаты численного эксперимента приведены в таблице 3, а на рисунке 4 изображены графики траекторий p, q, r и управления u_1, u_2, u_3 .

Таблица 3.

Результаты работы алгоритма для квадратичного сплайна

Координаты точек $(p(t_1), q(t_1), r(t_1))$	(0,014; -0,0109; 0,0001)
Коэффициенты разложения для u_1	(-134,12; -143,29; -146,64; -149,48)
Коэффициенты разложения для u_2	(-36,252; 32,6207; 36,4115)
Коэффициенты разложения для u_3	(0,0372; 0,0096)
Значение функционала I	171,81301

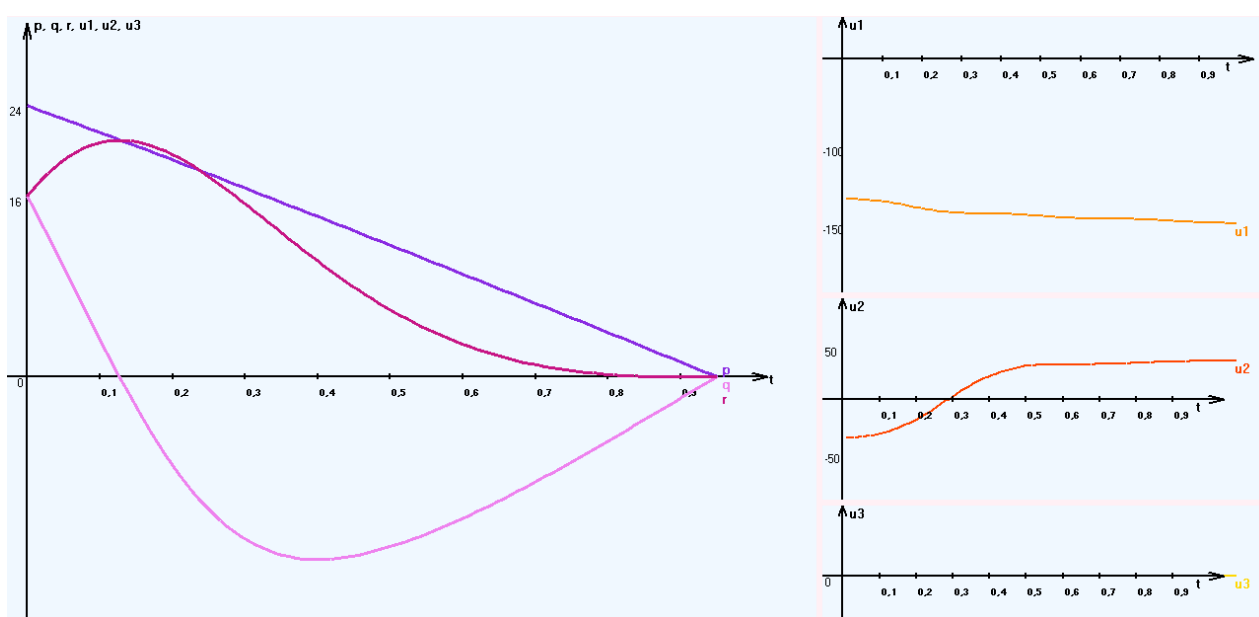


Рисунок 4. Графики траекторий и управления в случае квадратичного сплайна

Четвертый случай – в качестве базисных функций взят кубический сплайн. Найдено наилучшее количество коэффициентов в разложении $L_1 = 8$, $L_2 = 8$, $L_3 = 2$ для u_1 , u_2 и u_3 , соответственно. Также определены наилучшие параметры гибридного мультиагентного алгоритма интерполяционного поиска [19]: $NP = 40$; $I_{\max} = 400$; $M_1 = 2$; $M_2 = 5$; $PRT = 0,01$; $nstep = 5$; $b_2 = 8$. Результаты численного эксперимента приведены в таблице 4, а на рисунке 5 изображены графики траекторий p, q, r и управления u_1, u_2, u_3 .

Таблица 4.

Результаты работы алгоритма для кубического сплайна

Координаты точек $(p(t_1), q(t_1), r(t_1))$	(0,0015; -0,0009; 0,0007)
Коэффициенты разложения для u_1	(-143; -140,25; -146,93; -146,57)
Коэффициенты разложения для u_2	(-47,58; 17,68; 33,48; 38,28)
Коэффициенты разложения для u_3	(0,06; 0,03)
Значение функционала I	169,42035

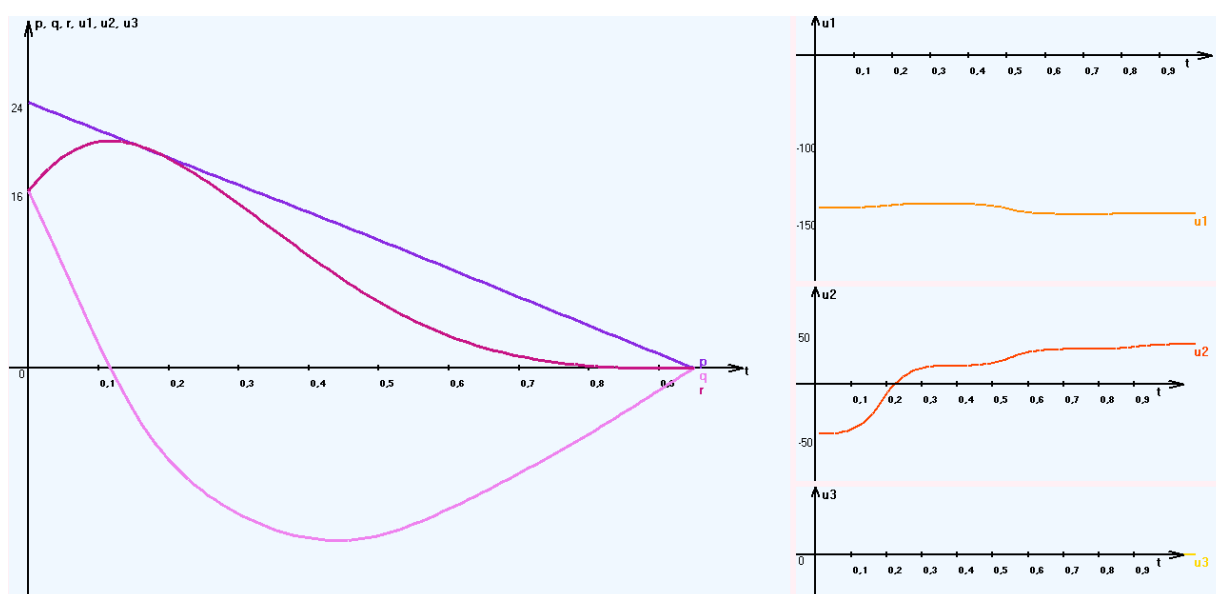


Рисунок 5. Графики траекторий и управления в случае кубического сплайна

В таблице 5 приведем итоговые численные результаты, полученные в четырех случаях, и сравним их с известным решением, полученным в [23]. Результат из [23] представлен на рисунке 6, а значение функционала $I = 169,42$.

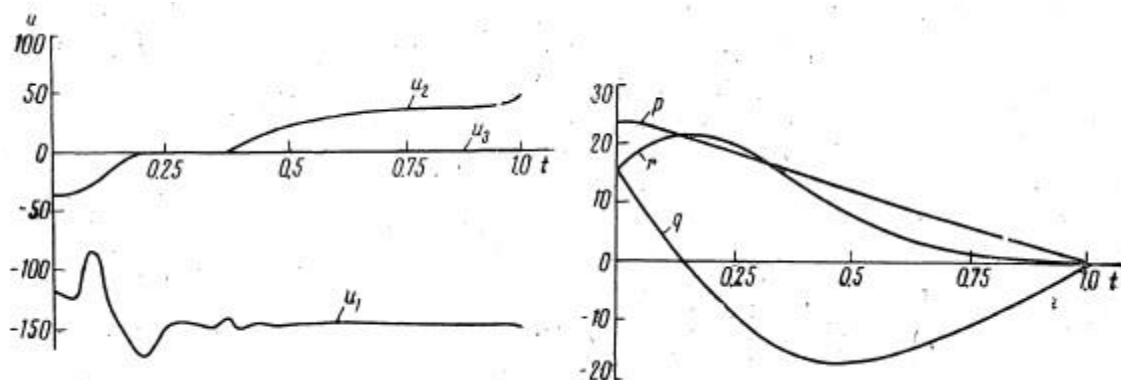


Рисунок 6. Программное управление и соответствующие ему траектории из [23]

Таблица 5.

Сравнение полученных результатов

Подход	Значение функционала I	Затраченное время
Кусочно-постоянные сплайны	169,42	2 мин. 33 с.
Кусочно-линейные сплайны	169,42	2 мин. 45 с.
Квадратичные сплайны	171,81	2 мин. 30 с.
Кубические сплайны	169,42	2 мин. 26 с.
Известное решение из [23]	169,42	-

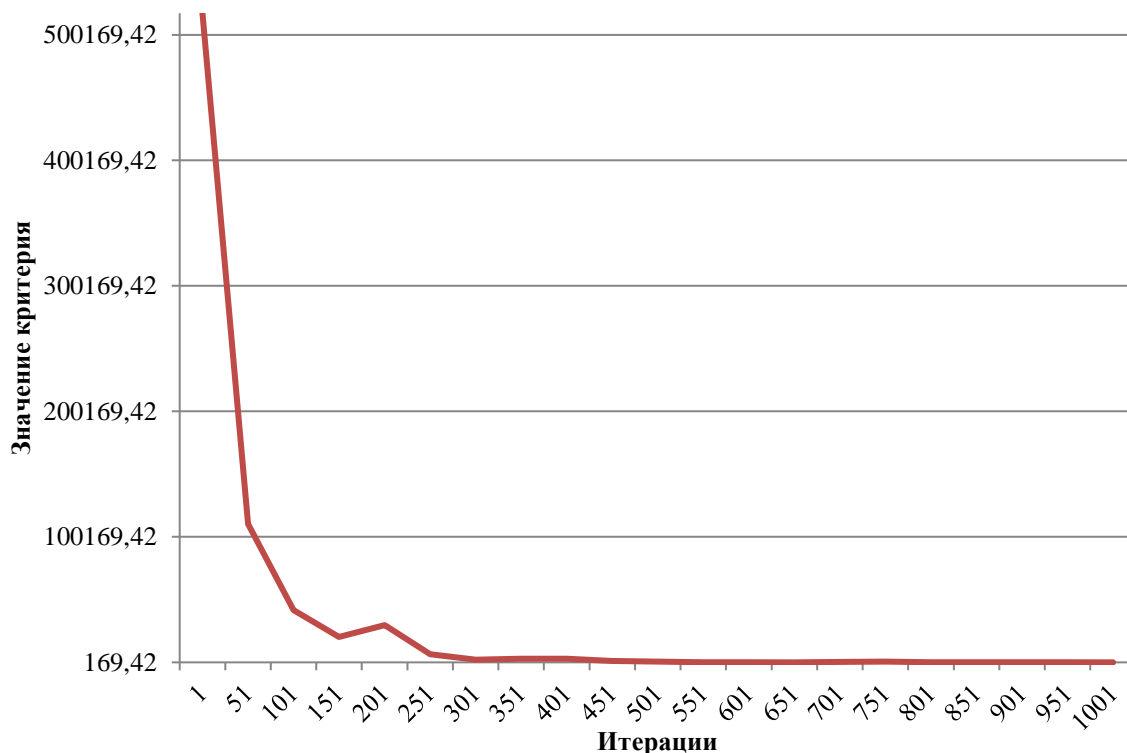


Рисунок 6. График зависимости значения критерия с ростом числа итераций при использовании кусочно-постоянного сплайна

Во всех четырех случаях при использовании сплайнов алгоритм успешно справился с поиском оптимального программного управления для гашения вращательного движения спутника, что можно наблюдать как из вычисленных характеристик, приведенных в таблицах 1 – 4, так и из графиков траекторий и управлений, изображенных на рисунках 2 – 5. Подобранные базисные функции подошли для решения поставленной задачи, но можно заметить, что при использовании квадратичного сплайна значение минимизируемого функционала получилось хуже известного из [23]. В остальных же случаях вычисленное значение критерия близко к оптимальному при использовании небольших вычислительных затрат. Также в таблице 5 указаны вычислительные затраты при решении задачи о стабилизации спутника: все расчеты были проведены на процессоре Intel Core i7

4700HQ с частотой 2.4 GHz. На рисунке 6 изображен график зависимости значения критерия с ростом числа итераций при использовании кусочно-постоянного сплайна (для остальных трех рассматриваемых типов сплайнов характер изменения значений критерия аналогичен). Как видно из графика, при увеличении числа итераций точность метода повышается, а значения функционала приближаются к известному из [23]. Следовательно, описанный алгоритм поиска оптимального программного управления на основе гибридного мультиагентного алгоритма интерполяционного поиска успешно справляется с прикладной задачей стабилизации спутника и может быть использован на практике.

Библиографический список

1. Panovskiy V.N., Panteleev A.V. Meta-heuristic interval methods of search of optimal in average control of nonlinear determinate systems with incomplete information about its parameters // Journal of Computer and System Sciences International, 2017, no. 56 (1), pp. 52 - 63. URL: <https://doi.org/10.1134/s1064230717010117>
2. Luus R. Iterative dynamic programming. Chapman and Hall/CRC, Boca Raton, USA, 2000, 344 p.
3. Panteleev A.V., Pis'mennaya V.A. Application of a memetic algorithm for the optimal control of bunches of trajectories of nonlinear deterministic systems with incomplete feedback // Journal of Computer and System Sciences International, 2018, no. 57 (1), pp. 25 - 36. URL: <https://doi.org/10.1134/s1064230718010082>

4. Pantelev A.V., Metlitskaya D.V. An application of genetic algorithms with binary and real coding for approximate synthesis of suboptimal control in deterministic systems // Automation and Remote Control, 2011, no. 72 (11), pp. 2328 - 2338. URL: <https://doi.org/10.1134/S0005117911110075>
5. Rao A.V. A Survey of Numerical Methods for Optimal Control // Advances in the Astronautical Sciences, 2010, no. 135 (1), pp. 1 – 32.
6. Федоренко Р.П. Приближенное решение задач оптимального управления. – М.: Наука, 1978. – 488 с.
7. Немыченков Г.И. Управление пучками траекторий стационарных систем автоматного типа при наличии дискретных неточных измерений // Труды МАИ. 2019. № 104. URL: <http://trudymai.ru/published.php?ID=102203>
8. Вернигора Л.В., Казмерчук П.В. Оптимизация траекторий КА с малой тягой методом линеаризации // Труды МАИ. 2018. № 106. URL: <http://trudymai.ru/published.php?ID=105759>
9. Beheshti Z., Shamsuddin S. M. H. A review of population-based meta-heuristic algorithms // International Journal of Advances in Soft Computing and its Applications, 2013, no. 5 (1), pp. 1 - 35.
10. Brownlee J. Clever Algorithms: Nature-Inspired Programming Recipes, LuLu, Morrisvill, USA, 2011, 423 p.
11. Пантелеев А.В., Скавинская Д.В., Алешина Е.А. Метаэвристические алгоритмы поиска оптимального программного управления. – М.: ИНФРА-М, 2016. – 396 с.

12. Karane M.M.S. Comparative analysis of multi-agent methods for constrained global optimization // IV international conference on information technologies in engineering education, Moscow, Russia, 23–26 October 2018, pp. 128 - 133. DOI: [10.1109/INFORINO.2018.8581711](https://doi.org/10.1109/INFORINO.2018.8581711)
13. Погарская Т.А. Оптимизация расстановки крепежных элементов при сборке самолетов на основе метода прямого поиска на адаптивных сетках // Труды МАИ. 2020. № 110. URL: <http://trudymai.ru/published.php?ID=112929>. DOI: [10.34759/trd-2020-110-18](https://doi.org/10.34759/trd-2020-110-18)
14. Rybakov K.A. Solving the nonlinear problems of estimation for navigation data processing using continuous particle filter // Gyroscopy and Navigation, 2019, no. 10 (1), pp. 27 - 34. DOI: [10.17285/0869-7035.2018.26.4.082-095](https://doi.org/10.17285/0869-7035.2018.26.4.082-095)
15. Averina T., Rybakov K. Systems with regime switching on manifolds // Proceedings of the 2018 14th International Conference «Stability and Oscillations of Nonlinear Control Systems» (Pyatnitskiy's Conference) (STAB), IEEE, 2018, pp. 1 - 3. DOI: [10.1109/STAB.2018.8408345](https://doi.org/10.1109/STAB.2018.8408345)
16. Пантелеев А.В., Каранэ М.С. Параметрический синтез оптимального программного управления на основе спектрального метода и мультиагентных алгоритмов оптимизации // Известия Института инженерной физики. 2020. № 3 (57). С. 74 - 78.
17. Blum C., Li X. Swarm Intelligence in Optimization // Swarm Intelligence, Natural Computing Series, Springer, Berlin, 2008, pp. 43 - 85. DOI: [10.1007/978-3-540-74089-6_2](https://doi.org/10.1007/978-3-540-74089-6_2)

18. Zelinka I. SOMA — Self-Organizing Migrating Algorithm // New Optimization Techniques in Engineering. Studies in Fuzziness and Soft Computing. Springer, Berlin, Heidelberg, 2004, 141 p. URL: https://doi.org/10.1007/978-3-540-39930-8_7
19. Pantelev A., Karane M. Hybrid multi-agent optimization method of interpolation search // AIP Conference Proceedings 2181, 020028, 2019. URL: <https://doi.org/10.1063/1.5135688>
20. Bacanin N., Pelevic B., Tuba M. Krill herd (KH) algorithm for portfolio optimization. In: Mathematics and Computers in Business, Manufacturing and Tourism // Proceedings of the 14th Intern. Conf. on Mathematics and Computers in Business and Economics (MCBE 13), Baltimore, USA, 2013, pp. 39 – 44.
21. Gandomi A.H., Alavi A.H. Krill herd: A new bio-inspired optimization algorithm // Communications in Nonlinear Science and Numerical Simulation, 2012, no. 17 (12), pp. 4831 - 4845. DOI: [10.1016/j.cnsns.2012.05.010](https://doi.org/10.1016/j.cnsns.2012.05.010)
22. Floudas C.A., Pardalos P.M. Encyclopedia of Optimization, Springer US, 2009, 4622 p.
23. Крылов И.А. Численное решение задачи об оптимальной стабилизации спутника // Вычислительная математика и физика. 1968. № 8 (1). С. 284 - 291.
24. Хрусталёв М.М., Халина А.С. Идентификаторы пониженной размерности в задаче стабилизации беспилотного летательного аппарата в неспокойной атмосфере // Труды МАИ. 2018. № 102. URL: <http://trudymai.ru/published.php?ID=99065>

Application of hybrid multi-agent interpolation search method to the satellite stabilization problem

Pantelev A.V.* , Karane M.M.S.**

*Moscow Aviation Institute (National Research University), MAI, 4,
Volokolamskoe shosse, Moscow, A-80, GSP-3, 125993, Russia*

*e-mail: avpantelev@inbox.ru

**e-mail: mmkarane@mail.ru

Abstract

The article regards the problem solution of satellite stabilization using the algorithm for finding the optimal open-loop control based on application of the expansion in terms of the system of basis functions and the multi-agent method.

Multi-agent algorithms are quite popular now and used in various fields of science. Multi-agent algorithms are not rigorously justified and convergence guarantee, however they are successfully applied in practice and demonstrate acceptable results in reasonable computational time. The article proposes a hybrid multi-agent interpolation search algorithm based on the Bezier, Catmull-Rom and B-splines interpolation curves construction. The curves are being constructed based on information on the agents, form the current population, position. In addition, the algorithm needs to solve the problems of one-dimensional parametric optimization to realize the exploratory and frontal search. Beyond that point, the described multi-agent method employs the ideas of swarm intelligence and migration algorithms.

Besides, the algorithm under consideration for the optimal open-loop control search is based on application of the expansion in terms of the system of basis functions. This

approach is quite popular and actively used in the spectral method for nonlinear control systems analysis and synthesis. The authors propose to search for the optimal open-loop control in the form of a saturation function, which argument is a linear combination of the given basis functions with some coefficients to be found. Particularly, piecewise constant, piecewise linear, quadratic and cubic splines are being used as basis functions. In addition, the saturation function should guarantee the specified constraints fulfillment on the parallelepipedic-type control.

Based on the proposed algorithm, the software for the optimal open-loop control search has been developed, and its algorithm efficiency has been studied when solving the problem of position stabilizing of a satellite driven by engines. Comparative analysis of the effect of basis system selection was performed as well. As it can be seen from the numerical experiment, the algorithm successfully copes with the problem of the satellite position stabilizing and finds a solution close to optimal when using all the above-mentioned systems of basis functions. In addition, the interpolation method parameters were selected so that to achieve a high accuracy of the applied problem being solved.

Keywords: optimal open-loop control, multi-agent algorithms, optimization, software, satellite stabilization.

References

1. Panovskiy V.N., Pantelev A.V. Meta-heuristic interval methods of search of optimal in average control of nonlinear determinate systems with incomplete information about its

- parameters, *Journal of Computer and System Sciences International*, 2017, no. 56 (1), pp. 52 - 63. URL: <https://doi.org/10.1134/s1064230717010117>
2. Luus R. *Iterative dynamic programming*. Chapman and Hall/CRC, Boca Raton, USA, 2000, 344 p.
3. Pantelev A.V., Pis'mennaya V.A. Application of a memetic algorithm for the optimal control of bunches of trajectories of nonlinear deterministic systems with incomplete feedback, *Journal of Computer and System Sciences International*, 2018, no. 57 (1), pp. 25 - 36. URL: <https://doi.org/10.1134/s1064230718010082>
4. Pantelev A.V., Metlitskaya D.V. An application of genetic algorithms with binary and real coding for approximate synthesis of suboptimal control in deterministic systems, *Automation and Remote Control*, 2011, no. 72 (11), pp. 2328 - 2338. URL: <https://doi.org/10.1134/S0005117911110075>
5. Rao A.V. A Survey of Numerical Methods for Optimal Control, *Advances in the Astronautical Sciences*, 2010, no. 135 (1), pp. 1 - 32.
6. Fedorenko R.P. *Priblizhennoe reshenie zadach optimal'nogo upravleniya* (Approximate solution of optimal control problems), Moscow, Nauka, 1978, 488 p.
7. Nemychenkov G.I. *Trudy MAI*, 2019, no. 104. URL: <http://trudymai.ru/eng/published.php?ID=102203>
8. Vernigora L.V., Kazmerchuk P.V. *Trudy MAI*, 2018, no. 106. URL: <http://trudymai.ru/eng/published.php?ID=105759>

9. Beheshti Z., Shamsuddin S. M. H. A review of population-based meta-heuristic algorithms, *International Journal of Advances in Soft Computing and its Applications*, 2013, no. 5 (1), pp. 1 - 35.
10. Brownlee J. *Clever Algorithms: Nature-Inspired Programming Recipes*, LuLu, Morrisvill, USA, 2011, 423 p.
11. Panteleev A.V., Skavinskaya D.V., Aleshina E.A. *Metaevristicheskie algoritmy poiska optimal'nogo programmogo upravleniya* (Meta-heuristic algorithms for finding optimal open-loop control), Moscow, INFRA-M, 2016, 396 p.
12. Karane M.M.S. Comparative analysis of multi-agent methods for constrained global optimization, *IV international conference on information technologies in engineering education*, Moscow, Russia, 23–26 October 2018, pp. 128 - 133. DOI: [10.1109/INFORINO.2018.8581711](https://doi.org/10.1109/INFORINO.2018.8581711)
13. Pogarskaya T.A. *Trudy MAI*, 2020, no. 110. URL: <http://trudymai.ru/eng/published.php?ID=112929>. DOI: [10.34759/trd-2020-110-18](https://doi.org/10.34759/trd-2020-110-18)
14. Rybakov K.A. Solving the nonlinear problems of estimation for navigation data processing using continuous particle filter, *Gyroscopy and Navigation*, 2019, no. 10 (1), pp. 27 - 34. DOI: [10.17285/0869-7035.2018.26.4.082-095](https://doi.org/10.17285/0869-7035.2018.26.4.082-095)
15. Averina T., Rybakov K. Systems with regime switching on manifolds, *Proceedings of the 2018 14th International Conference “Stability and Oscillations of Nonlinear Control Systems” (Pyatnitskiy’s Conference) (STAB)*, IEEE, 2018, pp. 1 - 3. DOI: [10.1109/STAB.2018.8408345](https://doi.org/10.1109/STAB.2018.8408345)

16. Pantelev A.V., Karane M.S. *Izvestiya Instituta inzhenernoi fiziki*, 2020, no. 3 (57), pp. 74 - 78.
17. Blum C., Li X. *Swarm Intelligence in Optimization*, *Swarm Intelligence, Natural Computing Series*, Springer, Berlin, 2008, pp. 43 - 85. DOI: [10.1007/978-3-540-74089-6_2](https://doi.org/10.1007/978-3-540-74089-6_2)
18. Zelinka I. *SOMA - Self-Organizing Migrating Algorithm*, *New Optimization Techniques in Engineering. Studies in Fuzziness and Soft Computing*. Springer, Berlin, Heidelberg, 2004, 141 p. URL: https://doi.org/10.1007/978-3-540-39930-8_7
19. Pantelev A., Karane M. Hybrid multi-agent optimization method of interpolation search, *AIP Conference Proceedings* 2181, 020028, 2019. URL: <https://doi.org/10.1063/1.5135688>
20. Bacanin N., Pelevic B., Tuba M. Krill herd (KH) algorithm for portfolio optimization. In: *Mathematics and Computers in Business, Manufacturing and Tourism, Proceedings of the 14th Intern. Conf. on Mathematics and Computers in Business and Economics (MCBE 13)*, Baltimore, USA, 2013, pp. 39 – 44.
21. Gandomi A.H., Alavi A.H. Krill herd: A new bio-inspired optimization algorithm, *Communications in Nonlinear Science and Numerical Simulation*, 2012, no. 17 (12), pp. 4831 - 4845. DOI: [10.1016/j.cnsns.2012.05.010](https://doi.org/10.1016/j.cnsns.2012.05.010)
22. Floudas C.A., Pardalos P.M. *Encyclopedia of Optimization*, Springer US, 2009, 4622 p.
23. Krylov I.A. *Vychislitel'naya matematika i fizika*, 1968, no. 8 (1), pp. 284 - 291.
24. Khrustalev M.M., Khalina A.S. *Trudy MAI*, 2018, no. 102. URL: <http://trudymai.ru/eng/published.php?ID=99065>