

DOI: 10.18698/0236-3933-2016-1-33-50

УДК 519.85:517.977.58

## ПРИМЕНЕНИЕ ОБОБЩЕННОГО ИНВЕРСНОГО ИНТЕРВАЛЬНОГО МЕТОДА ГЛОБАЛЬНОЙ УСЛОВНОЙ ОПТИМИЗАЦИИ В ЗАДАЧЕ ПОИСКА ОПТИМАЛЬНОГО ПРОГРАММНОГО УПРАВЛЕНИЯ

**А.В. Пантелеев, В.Н. Пановский**

Московский авиационный институт (национальный исследовательский университет), Москва, Российская Федерация  
e-mail: avpantelev@inbox.ru; panovskiy.v@yandex.ru

*Разработано алгоритмическое и программное обеспечение обобщенного инверсного интервального метода глобальной условной оптимизации, а также метод его применения для решения задачи нахождения оптимального программного управления нелинейными детерминированными непрерывными динамическими системами. Разработана обобщенная модульная схема алгоритма (имеющая два заменяемых модуля проверки и сжатия), использующего операцию инвертор. Приведены доказательства теорем о сходимости метода, решения прикладных задач управления химическим процессом и преследования маневрирующей цели перехватчиком.*

**Ключевые слова:** интервальные методы, глобальная условная оптимизация, оптимальное управление.

## APPLICATION OF THE GENERALIZED INVERSE INTERVAL METHOD OF GLOBAL CONSTRAINED OPTIMIZATION FOR OPTIMAL PROGRAM CONTROL PROBLEM

**A.V. Panteleev, V.N. Panovskiy**

Moscow Aviation Institute (National Research University),  
Moscow, Russian Federation  
e-mail: avpantelev@inbox.ru; panovskiy.v@yandex.ru

*The algorithmic and program software for the generalized inverse interval method of global constrained optimization as well as its application technique are developed for searching of the optimal program control of nonlinear deterministic continuous dynamical systems. The generalized module algorithm scheme (with two changeable check and compressibility modules) using the inverter operation was developed. The convergence theorems proofs, solutions of applied control problems (chemical process control and pursuit of a maneuvering target by an interceptor) are given.*

**Keywords:** interval methods, global constrained minimization, optimal control.

**Введение.** В современной математике достаточно большое внимание уделяется решению задач глобальной оптимизации [1–3], возникающих в процессе проектирования конструкций самолетов, вертолетов, космических аппаратов, когда появляется необходимость оптимизации

характерных параметров (масса, дальность полета, аэродинамические характеристики). Существующие численные методы используют разнообразные подходы, однако их применение связано с большими вычислительными затратами, излишними требованиями к постановке задачи, трудностями в достижении сходимости метода.

В качестве математического аппарата, определяющего предлагаемые в настоящей работе алгоритмы, используется интервальный анализ [4–6]. Существующие интервальные методы оптимизации можно подразделить на методы безусловной (алгоритмы Мура – Скелбоу, Ичиды – Фуджи, Дюсселя, интервальный алгоритм “имитации отжига”, метод случайного интервального дробления и др. [7–9]) и условной (методы Хансена, Мура и др. [10, 11]) оптимизации. Эвристические интервальные алгоритмы были применены в работе [12] для решения ряда прикладных задач. Предлагаемый интервальный метод глобальной условной оптимизации в процессе поиска использует операцию “инвертор”, имеет модульную структуру и относится к группе детерминированных методов [4]. В частных случаях реализаций модулей он совпадает с методом дихотомии прямого образа, методом отсечки и стохастической отсечки виртуальных значений, которые относятся к группе методов дробления графика [9, 13]. Показано, как общие задачи поиска оптимального программного управления нелинейными непрерывными детерминированными динамическими системами могут быть сведены к задаче интервальной условной оптимизации функций многих переменных и решены приближенно с помощью разработанных интервальных алгоритмов. Приведены примеры задачи управления химическим процессом, в которой имеется как локальный, так и глобальный экстремум, а также задачи преследования маневрирующей цели перехватчиком. На указанных примерах проиллюстрирована эффективность предложенного подхода.

**Используемые понятия и обозначения.** Основная идея интервального анализа — окружение вещественных чисел интервалами, а вещественных векторов — интервальными векторами (параллелотопами) [4]. Для обозначения интервала используются строчные латинские буквы, заключенные в квадратные скобки ( $[a]$ ,  $[b]$ ,  $[c]$ , ...), или представление  $([a_l; a_u], [b_l; b_u], [c_l; c_u], \dots)$ , где указывается нижняя и верхняя границы. Для задания параллелотопа применяется то же обозначение с полужирным начертанием букв ( $[a]$ ,  $[b]$ ,  $[c]$ , ...), или прямое произведение интервалов  $([1; 3] \times [2; 3] \times \dots \times [10; 11])$ .

Для произвольного интервала  $[x]$  определены следующие параметры: нижняя граница  $\underline{x} = lb([x]) = \sup\{\xi \in \mathbb{R} \cup \{-\infty, \infty\} \mid \forall \zeta \in [x], \xi \leq \zeta\}$ ; верхняя граница  $\overline{x} = ub([x]) = \inf\{\xi \in \mathbb{R} \cup \{-\infty, \infty\} \mid \forall \zeta \in [x], \zeta \leq \xi\}$ ; ширина непустого интервала  $\omega([x]) = \overline{x} - \underline{x}$ ; средняя

точка ограниченного и непустого интервала  $mid([x]) = \frac{[x] + \overline{[x]}}{2}$ .

Перечисленные параметры определены и для параллелотопов: нижняя и верхняя границы, а также средняя точка становятся векторами, ширина рассчитывается как максимальное значение ширины всех образующих параллелотоп компонентов.

Интервальной оболочкой множества  $X \subset \mathbb{R}^n$  называется параллелотоп с наименьшей шириной, который содержит множество  $X$ . Если множество  $X$  берется в квадратные скобки, это означает, что рассматривается интервальная оболочка  $[X]$  этого множества.

Пусть  $\circ$  — некоторая бинарная операция, тогда  $[x] \circ [y] = [\{\xi_1 \circ \xi_2 \mid \xi_1 \in [x], \xi_2 \in [y]\}]$ . Обозначим через  $f$  некоторый унарный оператор, тогда  $f([x]) = \{f(\xi) \mid \xi \in [x]\}$ . Описанная бинарная операция позволяет определить арифметические операции над интервалами [4].

Множество интервалов обозначается как  $\mathbb{IR}$ , множество интервальных векторов —  $\mathbb{IR}^n$ . Пусть имеется некоторая функция  $f: \mathbb{R}^n \rightarrow \mathbb{R}$ . Функция  $[f](\cdot)$  называется интервальной функцией включения для  $f: \mathbb{IR}^n \rightarrow \mathbb{IR}$ , если  $f([\mathbf{x}]) = \{f(\xi) \mid \xi \in [\mathbf{x}]\} \subseteq [f]([\mathbf{x}]), \forall [\mathbf{x}] \in \mathbb{IR}^n$ . Функция включения позволяет получить априорную оценку множества значений функции, даже если оно не является выпуклым или связным (если вместо переменных используются интервалы и соответствующие арифметические операции, то оценка называется оценкой прямого образа функции).

Инвертор  $INV(f, [s], [l], \varepsilon)$  — функция, которая по заданному интервалу  $[l]$ , функции  $f$  и параметру точности  $\varepsilon$  находит в области поиска  $[s]$  множество параллелотопов  $P$ , такое, что  $\forall [\mathbf{x}] \in P$  справедливо выражение  $[\mathbf{x}] \subseteq [s], \omega([\mathbf{x}]) \geq \varepsilon, [f]([\mathbf{x}]) \subseteq [l]$ , или  $[\mathbf{x}] \subseteq [s], \omega([\mathbf{x}]) < \varepsilon, [f]([\mathbf{x}]) \cap [l] \neq \emptyset$ . Инвертор может быть реализован с помощью алгоритма SIVIA [4].

**Алгоритм SIVIA.** Последовательность выполнения указанного алгоритма приведена ниже.

Шаг 1. Задать функцию  $f$ , область поиска  $[s]$ , интервал  $[l]$  и параметр точности  $\varepsilon$ . Создать пустое множество параллелотопов  $P$  и множество  $X = \{[s]\}$ .

Шаг 2. Для каждого параллелотопа  $[\mathbf{x}] \in X$  найти оценку прямого образа  $[f]([\mathbf{x}])$ . Проверить выполнение следующих условий (параллелотоп после проверки удаляется из множества  $X$ ):

а) если  $[f]([\mathbf{x}]) \cap [l] = \emptyset$ , то параллелотоп  $[\mathbf{x}]$  не добавляется в множество  $P$ ;

б) если  $[f]([\mathbf{x}]) \subseteq [l]$ , или  $[f]([\mathbf{x}]) \cap [l] \neq \emptyset, \omega([\mathbf{x}]) < \varepsilon$ , то добавить параллелотоп  $[\mathbf{x}]$  в множество  $P$ ;

в) в противном случае — задать параллелотопы  $[\mathbf{x}]' = [x_1] \times \dots \times [mid([x_k]); \overline{x_k}] \times \dots \times [x_n]$  и  $[\mathbf{x}]'' = [x_1] \times \dots \times [x_k; mid([x_k])] \times \dots \times [x_n]$ , где  $k$  — номер компоненты параллелотопа  $[\mathbf{x}]$  с наибольшей шириной, и добавить их в множество  $X$ . Параллелотопы  $[\mathbf{x}]'$  и  $[\mathbf{x}]''$  являются левым и правым результатами бисекции, примененной к параллелотопу  $[\mathbf{x}]$ .

Шаг 3. Если  $X \neq \emptyset$ , то перейти к шагу 2, в противном случае — к шагу 4.

Шаг 4. Вернуть множество параллелотопов  $P$ .

**Обобщенный инверсный интервальный метод глобальной условной оптимизации. Постановка задачи интервальной  $\varepsilon$ -минимизации и стратегия поиска решения.** Пусть имеются параллелотоп  $[s]$ , задающий множество допустимых решений, целевая функция  $f : \mathbb{R}^n \rightarrow \mathbb{R}$ , малое число  $\varepsilon > 0$ . Требуется найти параллелотоп  $[p]^*$ , такой, что

$$[p]^* \subseteq [s], \omega([p]^*) \leq \varepsilon, \nexists [x] \subseteq [s], \omega([x]) \geq \varepsilon : \overline{[f]}([x]) < \underline{[f]}([p]^*). \quad (1)$$

В основе метода лежит идея оценки значения функции на области поиска в виде целевого интервала. На каждой итерации алгоритма целевой интервал делится на два новых интервала [13]. Далее выполняется операция проверки того, в каком из полученных интервалов содержится значение минимума функции. Выбранный интервал вновь обозначается как целевой интервал и начинается следующая итерация. Поскольку первичная оценка значения функции на области поиска может быть неточной [4], целесообразно уточнить ее с помощью применения операции сжатия [13]. Операции проверки и сжатия выделены в алгоритме как отдельные заменяемые модули вследствие неединственности возможного описания этих процедур. Схема алгоритма представлена на рис. 1. Предлагаемый алгоритм называется обобщенным инверсным интервальным, так как при поиске наилучшего параллелотопа используются заменяемые модули сжатия и проверки, а также операция инвертор.

В качестве заменяемых модулей в алгоритме использованы следующие операторы.

1. *Оператор проверки*, на вход которого подается множество параллелотопов  $X$ , проверяемый интервал  $[l]$  и параметр ширины  $w$ . Результатом его работы является статус проверки “удачный”. Это свидетельствует о том, что в множестве параллелотопов  $X$  есть параллелотоп  $[p]$ , для которого выполнено условие  $\omega([p]) \geq w, [f]([p]) \subseteq [l]$ , или  $\omega([p]) < w, [f]([p]) \cap [l] \neq \emptyset$ . Статус проверки “неудачный” в противном случае. В алгоритме предлагается использовать:

— проверку до первого подходящего элемента First True operator (FT-operator);

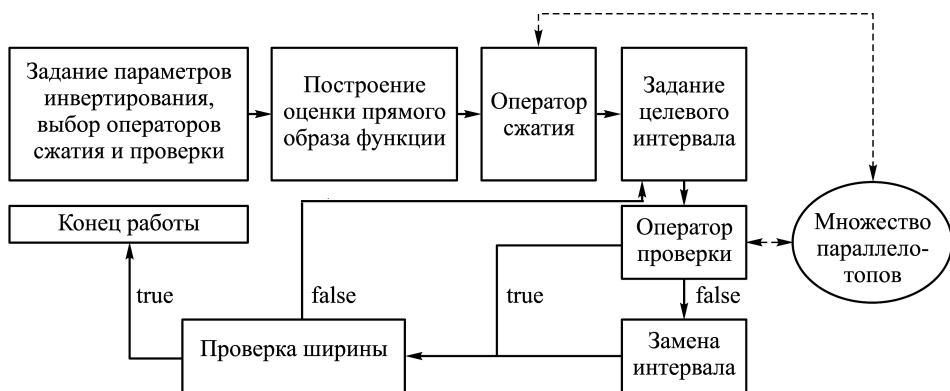


Рис. 1. Схема обобщенного инверсного интервального алгоритма

— проверку до первого подходящего элемента, множество  $X$  заменяется множеством, выработанным инвертором First True with Renewal operator (FTR-operator).

2. *Оператор сжатия*, на вход которого подается параллелотоп  $[s]$ , задающий область поиска. Результат его работы — сжатие оценки прямого образа функции на области поиска. В алгоритме предлагается применять:

- сжатие на основе разбиения области поиска, которое реализуется путем представления каждой компоненты параллелотопа, описывающего область поиска, в виде объединения непересекающихся интервалов Search Area Split operator (SAS-operator);
- сжатие на основе значений функций в случайно сгенерированных точках, которые являются интервальными векторами нулевой ширины Random Point Sample operator (RPS-operator).

**Алгоритм обобщенного инверсного интервального метода.** Последовательность выполнения указанного алгоритма приведена ниже.

Шаг 1. Задать область поиска  $[s]$ ; параметр точности, характеризующий размер параллелотопа в множестве  $P$ , которое будет вырабатываться инвертором  $\varepsilon > 0$ ; параметр точности для остановки алгоритма  $\zeta > 0$ . Выбрать оператор сжатия SAS с параметрами  $w$  и  $A$  или RPS с параметром  $A$ , а также оператор проверки FT или FTR с параметром  $w$ . Задать множество  $X = \{[s]\}$ .

Шаг 2. С помощью функции включения  $[f]$  найти оценку прямого образа  $[\Upsilon]$  функции  $f$  на области поиска  $[s]$ .

Шаг 3. Заменить полученную оценку  $[\Upsilon]$  интервалом  $[\tilde{\Upsilon}]$ , определенным с помощью выбранного оператора сжатия.

Шаг 4. С помощью операции бисекции разделить интервал  $[\Upsilon] = [y_l; y_u]$  на два интервала  $[\Upsilon_l] = \left[ y_l; \frac{y_l + y_u}{2} \right]$  и  $[\Upsilon_u] = \left[ \frac{y_l + y_u}{2}; y_u \right]$ .

Шаг 5. Проверить интервал  $[\Upsilon_l]$  выбранным оператором проверки.

Шаг 6. Если статус проверки “удачный”, то перейти к шагу 7, если нет — к шагу 8.

Шаг 7. Если  $\omega([\Upsilon_l]) < \zeta$ , то найти  $P = INV(f, [s], [\Upsilon_l], \varepsilon)$  и перейти к шагу 9. В противном случае принять интервал  $[\Upsilon]$  равным  $[\Upsilon_l]$  и перейти к шагу 4.

Шаг 8. Положить интервал  $[\Upsilon]$  равным  $[\Upsilon_u]$ . Если  $\omega([\Upsilon]) < \zeta$ , то найти  $P = INV(f, [s], [\Upsilon], \varepsilon)$  и перейти к шагу 9. В противном случае — перейти к шагу 4.

Шаг 9. Выбор параллелотопа.

Шаг 9.1. Принять  $i = 1$ , множество  $P_\varepsilon = P = \{[p]_i\}$ .

Шаг 9.2. Если  $\omega([p]_i) \leq \varepsilon$ , где  $[p]_i \in P_\varepsilon$ , то перейти к шагу 9.4, иначе — к шагу 9.3.

Шаг 9.3. Добавить в множество  $P_\varepsilon$  параллелотопы  $[p_1]_i \times \dots \times [mid([p_k]_i); \overline{p_k}]_i \times \dots \times [p_n]_i$  и  $[p_1]_i \times \dots \times [p_k]_i; mid([p_k]_i)] \times \dots \times [p_n]_i$ , где  $k$  — номер компоненты параллелотопа  $[p]_i$  с наибольшей шириной, удалить  $[p]_i$  из множества  $P_\varepsilon$ .

Шаг 9.4. Увеличить  $i$  на единицу. Если  $i \leq |P_\varepsilon|$ , где  $|\cdot|$  — мощность множества, то перейти к шагу 9.2. В противном случае перейти к шагу 9.5.

Шаг 9.5. Для каждого  $[p]_i \in P_\varepsilon$  найти  $[f]([p]_i)$  и выбрать  $[p]^* = \text{Arg} \min_{[p]_i \in P_\varepsilon} [f]([p]_i)$ .

**Алгоритмы реализации операторов проверки.** На вход всех операторов проверки подается множество параллелотопов  $X = \{[x]_i\}$ , проверяемый интервал  $[l]$  и параметр ширины  $w$ .

**Алгоритм FT-оператора.** Последовательность выполнения указанного алгоритма приведена ниже.

Шаг 1. Принять  $i = 1$ .

Шаг 2. Выполнить следующие действия:

1) если  $[f]([x]_i) \cap [l] = \emptyset$ , то продолжить работу;  
2) если  $[f]([x]_i) \subseteq [l]$ , или  $[f]([x]_i) \cap [l] \neq \emptyset$ ,  $\omega([x]_i) < w$ , то проверка завершена “удачно”;

3) в случае невыполнения действий 1 и 2 найти параллелотопы  $[x]' = [x_1]_i \times \dots \times [mid([x_k]_i); \overline{x_k}]_i \times \dots \times [x_n]_i$  и  $[x]'' = [x_1]_i \times \dots \times [x_k]_i; mid([x_k]_i)] \times \dots \times [x_n]_i$ , где  $k$  — номер компоненты параллелотопа  $[x]_i$  с наибольшей шириной, и добавить их в множество  $X$ .

Шаг 3. Если  $i < |X|$ , то увеличить значение  $i$  на единицу и перейти к шагу 2, а если  $i = |X|$ , то проверка завершена “неудачно”.

**Алгоритм FTR-оператора.** Последовательность выполнения указанного алгоритма приведена ниже.

Шаг 1. Принять  $i = 1$ . Задать множество  $\tilde{X} = X = \{[x]_i\}$ .

Шаг 2. Выполнить следующие действия (сам параллелотоп  $[\mathbf{x}]_i$  после проверки удаляется из множества  $X$ ):

1) если  $[f]([\mathbf{x}]_i) \cap [l] = \emptyset$ , то продолжить работу;

2) если  $[f]([\mathbf{x}]_i) \subseteq [l]$ , или  $[f]([\mathbf{x}]_i) \cap [l] \neq \emptyset$ ,  $\omega([\mathbf{x}]_i) < w$ , то проверка завершена “удачно”;

3) в случае невыполнения действий 1 и 2 найти параллелотопы  $[\mathbf{x}]' = [x_1]_i \times \dots \times [mid([x_k]_i); \overline{[x_k]_i}] \times \dots \times [x_n]_i$  и  $[\mathbf{x}]'' = [x_1]_i \times \dots \times [\overline{[x_k]_i}; mid([x_k]_i)] \times \dots \times [x_n]_i$ , где  $k$  — номер компоненты параллелотопа  $[\mathbf{x}]_i$  с наибольшей шириной, и добавить их в множество  $X$ .

Шаг 3. Если  $i < |X|$ , то увеличить значение  $i$  на единицу и перейти к шагу 2, а если  $i = |X|$  — перейти к шагу 4.

Шаг 4. Заменить множество  $X$  множеством  $\tilde{X}$ , сообщить, что проверка завершена “неудачно”.

**Алгоритмы реализации операторов сжатия.** На вход всех операторов сжатия подается параллелотоп  $[s]$ .

*Алгоритм SAS-оператора* (для работы необходим параметр ширины  $w$  и число попыток  $A$ ). Последовательность выполнения указанного алгоритма приведена ниже.

Шаг 1. Найти  $[\tilde{Y}] = [f]([s])$ .

Шаг 2. Для каждой компоненты вектора  $[s_i]$  найти такое наименьшее целое число  $n_i$ , что  $\frac{\omega([s_i])}{n_i} \leq w$ .

Шаг 3. Каждый интервал  $[s_i]$  представить в виде объединения непересекающихся интервалов  $\bigcup_{j=1}^{n_i} [s_i^j]$ . Таким образом, создается разбиение начального параллелотопа  $[s] = \bigcup_{k=1}^N [s]_k$ , где  $N = \prod_{i=1}^n n_i$ .

Шаг 4. Найти  $[\tilde{Y}] = \left[ \min \left( \left\{ [f]([s]_k) \right\}_{k=j_1, \dots, j_A}, [\tilde{Y}] \right), \min \left( \left\{ \overline{[f]([s]_k)} \right\}_{k=j_1, \dots, j_A}, \overline{[\tilde{Y}]} \right) \right]$  — результирующий сжатый интервал, где  $j_1, \dots, j_A$  — случайно выбранные натуральные числа из интервала  $[1; N]$ .

*Алгоритм RPS-оператора* (для работы необходимо число точек  $A$ ). Последовательность выполнения указанного алгоритма приведена ниже.

Шаг 1. Генерировать  $A$  случайных точек  $\xi_i \in [s]$ ,  $i = 1, \dots, A$ .

Шаг 2. Найти  $[\tilde{Y}] = \left[ \underline{[f]([s])}, \min_{i=1, \dots, A} f(\xi_i) \right]$  — результирующий сжатый интервал.

**Оценка качества решения. Теорема 1.** *Обобщенный инверсный интервальный метод находит решение задачи интервальной  $\varepsilon$ -минимизации.*

◀ Поскольку на шаге выбора (шаг 9) параллелотоп выбирается из множества  $P_\varepsilon$ , его ширина никогда не превысит выбранного значения  $\varepsilon$  (вследствие способа построения множества  $P_\varepsilon$ ). На каждом цикле алгоритм изначально с помощью инвертора вырабатывает множество параллелотопов  $P$ , в котором для любого параллелотопа  $[p] \in P$  справедливы соотношения  $[p] \subseteq [s]$ ,  $\omega([p]) \geq \varepsilon$  и  $[f]([p]) \subseteq [\Upsilon_l]$ , или  $[p] \subseteq [s]$ ,  $\omega([p]) < \varepsilon$  и  $[f]([p]) \cap [\Upsilon_l] \neq \emptyset$ . Поэтому переход к анализу  $[\Upsilon_u]$  свидетельствует о том, что множество  $P$  пусто, т.е.  $\forall [x] \subseteq [s]$ ,  $\omega([x]) < \varepsilon$  выполнено  $[f]([p]) \subset [\Upsilon_u]$ . Таким образом, в ходе каждого цикла алгоритма однозначно определяется интервал  $[y]$ , такой, что  $\nexists [x] \subseteq [s]$ , для которого выполнено условие  $\overline{[f]([x])} < [y]$ . Это исключает возможность существования параллелотопа  $[x] \subseteq [s]$ ,  $\omega([x]) \geq \varepsilon$ , для которого  $\overline{[f]([x])} < [f]([p]^*)$ , т.е. справедливо условие (1). ▶

**Теорема 2.** *Пусть  $x^* = \text{Arg min}_{x \in [s]} f(x)$ , а  $[p]^*$  – решение задачи интервальной  $\varepsilon$ -минимизации обобщенным инверсным интервальным методом, тогда  $f(x^*) \in [f]([p]^*)$ .*

◀ Необходимо показать, что для параллелотопа  $[p]^*$  одновременно выполняются неравенства  $f(x^*) \geq \underline{[f]([p]^*)}$ ,  $f(x^*) \leq \overline{[f]([p]^*)}$ . Докажем выполнение первого неравенства. Пусть  $f(x^*) < \underline{[f]([p]^*)}$ , тогда  $\exists [\tilde{p}]$  – это такое решение интервальной задачи интервальной  $\varepsilon$ -минимизации, что  $f(x^*) \in [f]([\tilde{p}])$ . Откуда следует  $[f]([\tilde{p}]) < \underline{[f]([p]^*)}$ , что невозможно, так как  $[p]^*$  является решением задачи интервальной  $\varepsilon$ -минимизации и при этом  $[p]^* = \text{Arg min}_{[p]_i \in P_\varepsilon} [f]([p]_i)$ . Следовательно, первое неравенство выполнено. Докажем выполнение второго неравенства. Пусть  $f(x^*) > \overline{[f]([p]^*)}$ , тогда  $[f]([p]^*) = \emptyset$ , что невозможно, поскольку  $[p]^* \neq \emptyset$ . Таким образом, второе неравенство выполнено. Два неравенства выполнены, теорема 2 доказана. ▶

**Следствие теоремы 2.** *В параллелотопе  $[p]^*$  есть точка  $\tilde{x}$ , такая, что  $f(\tilde{x}) - f(x^*) \leq \omega([f]([p]^*))$ .*

**Оптимальное интервальное программное управление непрерывными системами.** *Постановка задачи.* Поведение модели объекта управления описывается дифференциальным уравнением

$$\dot{x}(t) = f(t, x(t), u(t)), \quad (2)$$

где  $t$  – непрерывное время;  $t \in T = [t_0, t_1]$  – заданный промежуток времени функционирования системы;  $x \in \mathbb{R}^n$  – вектор состояния системы;  $u \in [u] \subset \mathbb{R}^q$  – вектор управления;  $[u]$  – множество допустимых значений управления, представляющее собой параллело-



топ;  $f(t, x, u) = (f_1(t, x, u), \dots, f_n(t, x, u))^T$  — непрерывная вектор-функция.

Начальное состояние системы (2) задано и равно

$$x(t_0) = x_0. \quad (3)$$

Конечное состояние  $x(t_1)$  должно удовлетворять условиям

$$\Gamma_i(x(t_1)) = 0, \quad i = 1, \dots, l, \quad (4)$$

где  $0 \leq l \leq n$ ;  $\Gamma_i(x)$  — непрерывно дифференцируемые функции;  $\left\{ \partial_i(x)/\partial x_1, \dots, \partial_i(x)/\partial x_n \right\}$ ,  $i = 1, \dots, l$  — система векторов, линейно независимая для  $\forall x \in \mathbb{R}^n$ .

При управлении используется информация только о непрерывном времени  $t$ , т.е. применяется программное управление. Множество допустимых процессов  $D(t_0, x_0)$  определяется как множество пар  $d = (x(\cdot), u(\cdot))$ , включающих в себя траекторию  $x(\cdot)$  и кусочно-непрерывное управление  $u(\cdot)$ , где  $\forall t \in T \quad u(t) \in [\mathbf{u}]$ , удовлетворяющих уравнению состояния (2), начальному (3) и конечному (4) условиям.

На множестве допустимых процессов определен функционал качества управления

$$I(d) = \int_{t_0}^{t_1} f^0(t, x(t), u(t)) dt + F(x(t_1)), \quad (5)$$

где  $f^0(t, x, u)$ ,  $F(x)$  — заданные непрерывные функции.

Требуется найти пару  $d^* = (x^*(\cdot), u^*(\cdot)) \in D(t_0, x_0)$ , при которой достигается минимальное значение функционала (5) на множестве допустимых процессов.

Основным элементом предлагаемого подхода к решению поставленной задачи является ее последовательное сведение к соответствующей задаче нелинейного программирования, а затем — к задаче интервальной  $\varepsilon$ -минимизации и решение последней с помощью разработанного обобщенного инверсного интервального метода глобальной условной оптимизации.

Для более простого вычисления интегрального члена в критерии (5) к системе (2) добавляется уравнение  $\dot{x}_{n+1}(t) = f^0(t, x(t), u(t))$  с начальным условием  $x_{n+1}(t_0) = 0$ , тогда значение функционала качества (5) определяется по выражению  $I = x_{n+1}(t_1) + F(t_1, x(t_1))$ .

Для сведения поставленной задачи с ограничениями (4) к задаче со свободным правым концом траектории к терминальному члену функционала добавляются либо классические штрафные функции, характеризующие степень выполнения условий (4), либо их интервальный

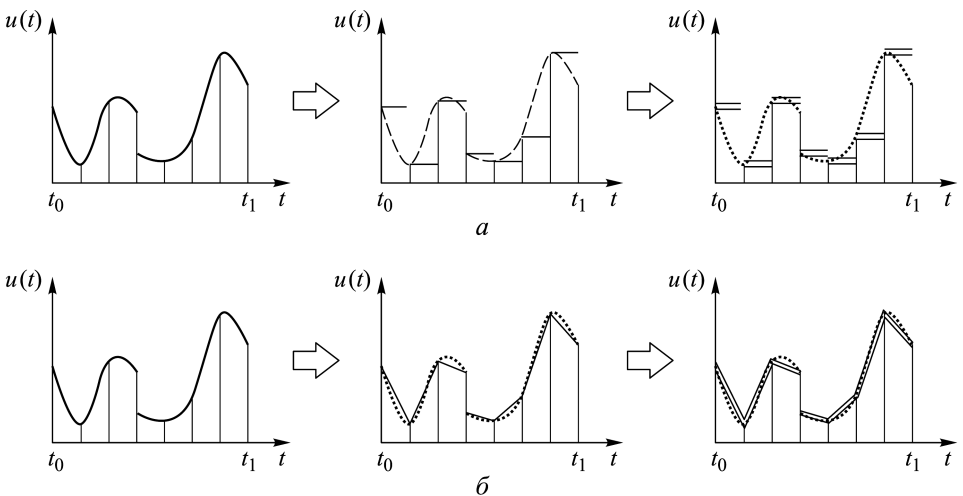
аналог вида

$$\sum_{i=1}^l R_i h_{\infty}^2([\Gamma_i(x(t_1)); \Gamma_i(x(t_1))], [-\xi_i; \xi_i]), \quad (6)$$

где  $h_{\infty}([a], [b]) = \left[ \min(h_{\infty}^0([a], [b]), h_{\infty}^0([a], [b])), \max(h_{\infty}^0([a], [b]), h_{\infty}^0([a], [b])) \right]$  – метрика Хаусдорфа;  $h_{\infty}^0([a], [b]) = \inf \left\{ r \mid [a] \subseteq [b] + [-r; r], r \geq 0 \right\}$  – мера близости интервалов;  $R_i$  – параметры штрафных функций;  $\xi_i > 0$  – допустимая погрешность выполнения конечного условия, задаваемая пользователем.

Искомое оптимальное программное управление предлагается искать в двух различных классах функций: кусочно-постоянных и кусочно-линейных. Поскольку выполняется поиск наилучшего управления в каждом классе, не совпадающем с классом кусочно-непрерывных функций, найденное управление является субоптимальным, рассматриваемым в качестве кандидата в решение задачи. При решении задачи применяется интервальный метод, поэтому каждому описанному управлению сопоставляется интервальный аналог, задаваемый интервальным вектором.

Для кусочно-постоянного интервального управления (рис. 2, а) необходимо задать значения функции  $u(t)$  в  $N$  моментах времени  $\tau_i = t_0 + \frac{t_1 - t_0}{N}i, i = 0, \dots, N - 1$ , т.е. управлению можно однозначно поставить в соответствие интервальный вектор  $[u] = \underbrace{[u_1(\tau_0)] \times \dots \times [u_q(\tau_0)]}_{[u(\tau_0)]} \times \dots \times \underbrace{[u_1(\tau_{N-1})] \times \dots \times [u_q(\tau_{N-1})]}_{[u(\tau_{N-1})]}$ . Тогда



**Рис. 2.** Замена кусочно-непрерывного управления интервальным кусочно-постоянным (а) и кусочно-линейным (б) управлениями

интервальное управление находится по формуле  $[u](t) \in [u(\tau_i)] = = [\underline{u}(\tau_i); \bar{u}(\tau_i)]$ ,  $\tau_i \leq t < \tau_{i+1}$ ,  $i = 0, \dots, N - 1$ .

Для кусочно-линейного интервального управления (рис. 2, б) необходимо задать дополнительное значение управления  $u(t)$  на последнем временном интервале. Поэтому интервальный вектор, который ставится в соответствие управлению, имеет вид  $[u] = = \underbrace{[u_1(\tau_0)] \times \dots \times [u_q(\tau_0)]}_{[u(\tau_0)]} \times \dots \times \underbrace{[u_1(\tau_N)] \times \dots \times [u_q(\tau_N)]}_{[u(\tau_N)]}$ . Интерваль-

ное управление определяется по формуле  $[u](t) \in \frac{\tau_{i+1} - t}{\tau_{i+1} - \tau_i} [u(\tau_i)] + + \frac{t - \tau_i}{\tau_{i+1} - \tau_i} [u(\tau_{i+1})]$ ,  $\tau_i \leq t < \tau_{i+1}$ ,  $i = 0, \dots, N - 1$ .

Интервальной траекторией называется решение уравнения (2) с начальным условием (3), соответствующее заданному интервальному управлению. Если система (2), описывающая поведение модели объекта управления, не жесткая, то для ее численного интегрирования применяют явные методы (методы Эйлера, Эйлера – Коши, методы Рунге – Кутты различных порядков и др.). Если система (2) жесткая, то используют неявные методы численного интегрирования (неявный метод Эйлера, метод трапеций, методы Адамса – Моултона, Гира и др.). Вычисление правых частей системы (2) проводится по правилам интервальной арифметики [4].

Отметим, что описанные процедуры могут быть использованы при других способах параметризации управления, например, кусочно-полиномиальном, или при разложении по системам ортонормированных базисных функций.

**Решение прикладных задач.** На основе приведенного алгоритма написана программа поиска субоптимального интервального программного управления. Среда разработки Microsoft Visual Studio, язык программирования C#.

Для исследования эффективности разработанного метода и программного обеспечения использованы принципы, изложенные в работе [14]: полученное приближенное решение необходимо сравнить с точным решением (если известно), с результатами применения принципа максимума, а также с результатами применения других методов.

**Задача управления химическим процессом.** Рассмотрим решение задачи оптимального управления химическим процессом [15, 16]. Модель объекта управления описывается системой дифференциальных уравнений

$$\begin{aligned} \dot{x}_1(t) &= -(2+u)(x_1+0,25) + (x_2+0,5) \exp\left(\frac{25x_1}{x_1+2}\right); \\ \dot{x}_2(t) &= 0,5 - x_2 - (x_2+0,5) \exp\left(\frac{25x_1}{x_1+2}\right); \\ \dot{x}_3(t) &= x_1^2 + x_2^2 + 0,1u^2, \end{aligned} \quad (7)$$

где  $x_1, x_2$  — отклонения от безразмерной установившейся температуры и безразмерной установившейся концентрации;  $x_3$  — величина, определяющая значение критерия качества при  $t = t_1, t \in [0; 0,78]$ ; начальное состояние  $x(0) = (0,09; 0,09; 0)^T$ , ограничение на управление  $[u] = [-10; 10]$ ; критерий качества управления  $I = x_3(0,78) \rightarrow \min$ .

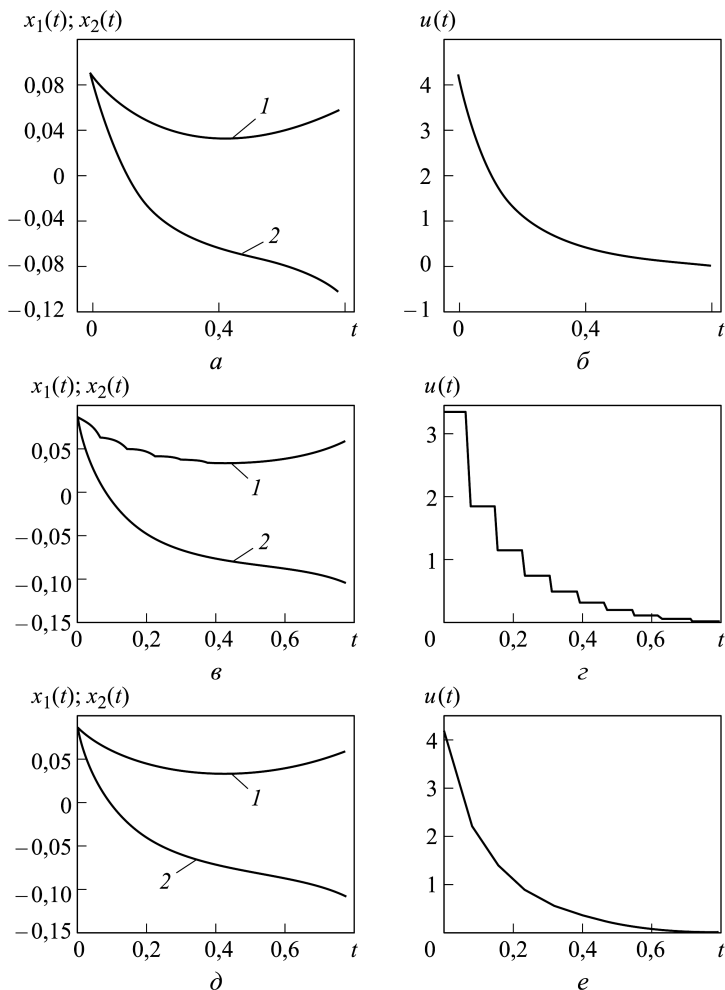
Результаты решения приведенной задачи методом динамического программирования ( $I = 0,13336$ ) в виде зависимостей изменений компонент векторов состояния и управления представлены на рис. 3, а, б [15, 16]. На этом же рисунке приведены результаты решения задачи, полученные с помощью принципа максимума Понтрягина (локальный минимум  $I = 0,244425$  и глобальный минимум  $I = 0,133094$ ), алгоритма Нелдера–Мида ( $I = 0,1333666$ ) и генетических алгоритмов ( $I = 0,13341$ ).

Параметры обобщенного инверсного интервального метода: в качестве метода дискретизации выбран явный метод Рунге–Кутты четвертого порядка; области поиска  $[s] = \underbrace{[-10; 10] \times \dots \times [-10; 10]}_N$  для

кусочно-постоянного управления и  $[s] = \underbrace{[-10; 10] \times \dots \times [-10; 10]}_{N+1}$

для кусочно-линейного управления,  $\varepsilon = \zeta = 0,00001$ ;  $N = 10$ . Результаты работы алгоритма и исследования эффективности алгоритма представлены на рис. 3, в, г и в табл. 1.

Сравнивая значения критерия оптимальности (см. табл. 1) со значениями, приведенными в работах [15, 16], можно сделать следующий вывод: разработанный алгоритм позволяет найти решение, близкое к глобальному минимуму, не застревая в окрестности локального минимума, существующего в рассматриваемой задаче. Его эффективность при заданной точности сравнима с эффективностью алгоритмов применения принципа максимума, итерационного динамического программирования, метода Нелдера–Мида и эволюционных алгоритмов. Кусочно-линейное управление предпочтительнее по функционалу, чем кусочно-постоянное. Полученные результаты иллюстрируют применимость теорем 1 и 2, поскольку находится решение задачи  $\varepsilon$ -минимизации и гарантируется принадлежность глобального оптимума к получаемому интервалу значений критерия оптимальности.



**Рис. 3.** Зависимости изменения компонент векторов состояния  $x_1$  (1) и  $x_2$  (2) (а) и управления  $u$  (б), найденные в работах [15, 16] и с помощью разработанного алгоритма в случае кусочно-постоянного (в) и кусочно-линейного (з) управлений

**Задача преследования.** Рассмотрим решение задачи преследования в трехмерном пространстве [17]. Поведение модели объекта управления описывается системой дифференциальных уравнений

$$\begin{aligned}
 \dot{x}(t) &= V_x(t), \quad \dot{V}_x(t) = u_x(t), \quad \dot{x}^t(t) = V_x^t(t), \quad \dot{V}_x^t(t) = u_x^t(t), \\
 \dot{y}(t) &= V_y(t), \quad \dot{V}_y(t) = u_y(t), \quad \dot{y}^t(t) = V_y^t(t), \quad \dot{V}_y^t(t) = u_y^t(t), \quad (8) \\
 \dot{z}(t) &= V_z(t), \quad \dot{V}_z(t) = u_z(t), \quad \dot{z}^t(t) = V_z^t(t), \quad \dot{V}_z^t(t) = u_z^t(t),
 \end{aligned}$$

где  $r = (x, y, z)^T$ ,  $r^t = (x^t, y^t, z^t)^T$  — радиус-векторы, описывающие положение перехватчика и цели;  $V = (V_x, V_y, V_z)^T$ ,  $V^t = (V_x^t, V_y^t, V_z^t)^T$  — векторы скоростей перехватчика и цели;  $u = (u_x, u_y, u_z)^T$ ,  $u^t = (u_x^t, u_y^t, u_z^t)^T$  — ускорения перехватчика (управления) и цели.

## Результат работы алгоритма

Операторы проверки и сжатия	Функционал качества $I$
<i>Кусочно-постоянное управление</i>	
FT ( $w = 0,001$ ), SAS( $w = 2, A = 25$ )	[0,139861; 0,140337]
FTR ( $w = 0,001$ ), SAS ( $w = 2, A = 25$ )	[0,139549; 0,140376]
FT ( $w = 0,001$ ), RPS ( $A = 100$ )	[0,139831; 0,140791]
FTR ( $w = 0,001$ ), RPS ( $A = 100$ )	[0,139642; 0,140509]
<i>Кусочно-линейное управление</i>	
FT ( $w = 0,001$ ), SAS( $w = 2, A = 25$ )	[0,133087; 0,133121]
FTR ( $w = 0,001$ ), SAS ( $w = 2, A = 25$ )	[0,133092; 0,133101]
FT ( $w = 0,001$ ), RPS ( $A = 100$ )	[0,133095; 0,133118]
FTR ( $w = 0,001$ ), RPS ( $A = 100$ )	[0,133001; 0,133195]

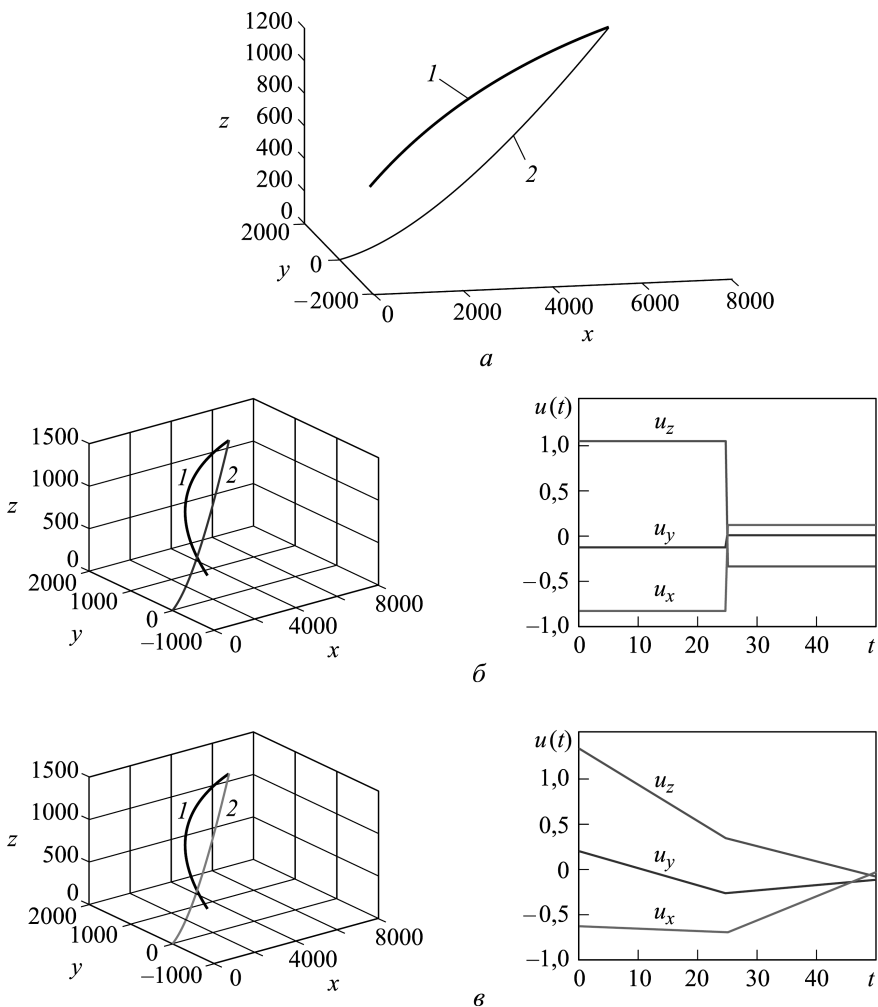
Предполагается, что цель движется с постоянным ускорением  $u^t = (1; -2; 0,1)^T$  м/с<sup>2</sup>; начальное положение цели  $r^t = (500; -600; 500)^T$  м; начальная скорость цели  $V^t = (100; 100; 10)^T$  м/с; начальное положение перехватчика  $r = (0, 0, 0)^T$  м; начальная скорость перехватчика  $V = (150; 40; 5)^T$  м/с. Время осуществления перехвата (координаты цели и перехватчика равны):  $t_1 = 50$  с. Функционал качества управления (5) задается в виде

$$I = (x(t_1) - x^t(t_1))^2 + (y(t_1) - y^t(t_1))^2 + (z(t_1) - z^t(t_1))^2.$$

Для сглаживания искомого интервального управления в функционал добавим “штрафное” слагаемое  $\Delta I = 0,001 \sum_{j=0}^M (u_{x,j}^2 + u_{y,j}^2 + u_{z,j}^2)$ , где  $u_{x,j} = u_x(\tau_j)$ ,  $u_{y,j} = u_y(\tau_j)$ ,  $u_{z,j} = u_z(\tau_j)$ ,  $M = N - 1$  для кусочно-постоянного управления,  $M = N$  для кусочно-линейного управления. В результате решается задача минимизации функционала  $I + \Delta I$ .

Параметры обобщенного инверсного интервального метода: в качестве метода дискретизации выбраны явная и неявная схемы Эйлера; область поиска  $[s] = \underbrace{[-2; 2] \times \dots \times [-2; 2]}_{3 \cdot N}$  для кусочно-постоянного управления и  $[s] = \underbrace{[-2; 2] \times \dots \times [-2; 2]}_{3 \cdot (N+1)}$  для кусочно-линейного управления;  $\varepsilon = 0,001$ ;  $\zeta = 0,1$ ;  $N = 2$ . Зависимость движения цели и перехватчика, полученная в работе [17], представлена на рис. 4, а. Результаты работы алгоритма приведены на рис. 4, б, в и в табл. 2.

Сравнительный анализ управления и траекторий, найденных интервальным методом и методом пропорциональной навигации, свиде-



**Рис. 4.** Зависимость движения цели и перехватчика, полученная с помощью алгоритма пропорциональной навигации [17] (а), зависимости движения цели и перехватчика, законов изменения управления, найденные с помощью разработанного алгоритма при кусочно-постоянном (б) и кусочно-линейном (в) управлениях

тельствует о близости полученных результатов. Максимальное суммарное отклонение цели и перехватчика  $\Delta = \Delta_x + \Delta_y + \Delta_z = 0,09 + 0,14 + 0,07 = 0,3$  для кусочно-постоянного управления и  $\Delta = \Delta_x + \Delta_y + \Delta_z = 0,06 + 0,08 + 0,14 = 0,28$  для кусочно-линейного управления, что соответствует требованиям, предъявляемым к точности решения задачи преследования. Здесь  $\Delta_y = \max |y(t_1) - y^t(t_1)|$ ,  $\Delta_z = \max |z(t_1) - z^t(t_1)|$  – погрешности выполнения терминальных условий по каждой координате. Кусочно-линейное управление предпочтительнее по функционалу, чем кусочно-постоянное (см. табл. 2), поскольку приводит к меньшим значениям ширины интервала неопределенностей значений критерия оптимальности. Отметим, что

## Результат работы алгоритма

Операторы проверки и сжатия	Функционал качества $I$ для схем Эйлера	
	Явная	Неявная
<i>Кусочно-постоянное управление</i>		
FT ( $w = 0,01$ ), SAS( $w = 2, A = 25$ )	[0; 0,0943]	[0; 0,0912]
FTR ( $w = 0,01$ ), SAS ( $w = 2, A = 25$ )	[0; 0,0997]	[0; 0,0973]
FT ( $w = 0,01$ ), RPS ( $A = 100$ )	[0; 0,0995]	[0; 0,0996]
FTR ( $w = 0,01$ ), RPS ( $A = 100$ )	[0; 0,0990]	[0; 0,0988]
<i>Кусочно-линейное управление</i>		
FT ( $w = 0,01$ ), SAS( $w = 2, A = 25$ )	[0; 0,0921]	[0; 0,0893]
FTR ( $w = 0,01$ ), SAS ( $w = 2, A = 25$ )	[0; 0,0909]	[0; 0,0910]
FT ( $w = 0,01$ ), RPS ( $A = 100$ )	[0; 0,0887]	[0; 0,0884]
FTR ( $w = 0,01$ ), RPS ( $A = 100$ )	[0; 0,0972]	[0; 0,1087]

каждому управлению соответствуют интервалы, содержащие глобальный минимум.

**Заключение.** Предложен новый детерминированный обобщенный инверсный интервальный метод условной глобальной оптимизации функций многих переменных, сформулированы и доказаны теоремы о свойствах метода. Разработан алгоритм для решения задач поиска субоптимального программного управления непрерывными нелинейными детерминированными динамическими системами в классе кусочно-постоянных и кусочно-линейных управлений. Решены две прикладные задачи: управления химическим процессом и преследования цели перехватчиком. К преимуществам предложенного алгоритма относится возможность гарантии принадлежности глобального оптимума получаемому интервалу значений критерия оптимальности.

## ЛИТЕРАТУРА

1. *Пантелеев А.В., Летова Т.А.* Методы оптимизации. Практический курс. М.: Логос, 2011. 424 с.
2. *Пантелеев А.В., Метлицкая Д.В., Алешина Е.А.* Методы глобальной оптимизации. Метаэвристические стратегии и алгоритмы. М.: Вузовская книга, 2013. 248 с.
3. *Пантелеев А.В.* Применение эволюционных методов глобальной оптимизации в задачах оптимального управления детерминированными системами. М.: Изд-во МАИ, 2013. 160 с.
4. *Jaulin L., Kieffer M., Didrit O., Walter E.* Applied interval analysis. London: Springer-Verlag, 2001. 384 p.
5. *Moore R.E.* Interval analysis. Englewood Cliffs, Prentice Hall, 1966. 145 p.
6. *Шарый С.П.* Конечномерный интервальный анализ. Новосибирск: XYZ, 2010. 606 с.



7. *Ratschek H., Rokne J.* New computer methods for global optimization. Chichester: Horwood, 1988. 229 p.
8. *Shary S.P.* Randomized algorithms in interval global optimization // *Numerical Analysis and Applications*. 2008. Vol. 1. P. 376–389.
9. *Shary S.P.* A surprising approach in interval global optimization // *Reliable Computing*. 2001. P. 497–505.
10. *Hansen E.* Global optimization using interval analysis. New York: Marcel Dekker, 2004. 515 p.
11. *Moore R.E.* Methods and applications of interval analysis. Philadelphia: SIAM, 1979. 190 p.
12. *Пановский В.Н.* Прикладное применение интервального метода взрывов // *Труды МАИ*. 2014. № 73.  
URL: <http://www.mai.ru/science/trudy/published.php?ID=48451>
13. *Пановский В.Н.* Применение аппарата интервального анализа для поиска глобального экстремума функций // *Труды МАИ*. 2012. № 51.  
URL: <http://www.mai.ru/science/trudy/published.php?ID=28953>
14. *Федоренко Р.П.* Приближенное решение задач оптимального управления. М.: Наука, 1978. 488 с.
15. *Пантелеев А.В., Метлицкая Д.В.* Применение генетических алгоритмов с бинарным и вещественным кодированием для приближенного синтеза субоптимального управления детерминированными системами // *Автоматика и телемеханика*. 2011. № 11. С. 117–129.
16. *Luus R.* Iterative dynamic programming. New York: Chapman & Hall, 2000. 331 p.
17. *Tewari A.* Advanced control of aircraft, spacecraft and rockets. New York: John Wiley & Sons, 2011. 448 p.

## REFERENCES

- [1] Panteleev A.V., Letova T.A. *Metody optimizatsii. Prakticheskiy kurs [Methods of Optimization. Practical Course]*. Moscow, Logos, 2011. 424 p.
- [2] Panteleev A.V., Metlitskaya D.V., Aleshina E.A. *Metody global'noy optimizatsii. Metaevristicheskie strategii i algoritmy [Methods of Global Optimization. Metaheuristic Strategies and Algorithms]*, Moscow, Vuzovskaya kniga Publ., 2013. 248 p.
- [3] Panteleev A.V. *Primenenie evolyutsionnykh metodov global'noy optimizatsii v zadachakh optimal'nogo upravleniya determinirovannymi sistemami [Application of Evolutionary Methods of Global Optimization in Problems of Finding the Optimal Control of Determined Systems]*, Moscow, MAI Publ., 2013. 160 p.
- [4] Jaulin L., Kieffer M., Didrit O., Walter E. *Applied interval analysis*. London, Springer-Verlag, 2001. 384 p.
- [5] Moore R.E. *Interval analysis*. Englewood Cliffs, Prentice Hall, 1966. 145 p.
- [6] Shary S.P. *Konechnomernyy interval'nyy analiz [Finite-dimensional Interval Analysis]*, Novosibirsk, XYZ Publ., 2010. 606 p.
- [7] Ratschek H., Rokne J. *New computer methods for global optimization*. Chichester, Horwood, 1988. 229 p.
- [8] Shary S.P. Randomized algorithms in interval global optimization. *Numerical Analysis and Applications*, 2008, vol. 1, pp. 376–389.
- [9] Shary S.P. A surprising approach in interval global optimization. *Reliable Computing*, 2001, pp. 497–505.
- [10] Hansen E. *Global optimization using interval analysis*. N.Y., Marcel Dekker, 2004. 515 p.
- [11] Moore R.E. *Methods and applications of interval analysis*. Philadelphia, SIAM, 1979. 190 p.

- [12] Panovskiy V.N. Application of the interval explosion method. *Electronnyy zhurnal "Trudy MAI"*, 2014, iss. 73. Available at: <http://www.mai.ru/science/trudy/published.php?ID=48451>
- [13] Panovskiy V.N. Application of the interval analysis for the search of the global extremum of functions. *Electronnyy zhurnal "Trudy MAI"*, 2012, iss. 51. Available at: <http://www.mai.ru/science/trudy/published.php?ID=28953>
- [14] Fedorenko R.P. *Priblizhennoe reshenie zadach optimal'nogo upravleniya* [Approximate Solution of the Optimal Control Problems]. Moscow, Nauka Publ., 1978. 488 p.
- [15] Panteleev A.V., Metlitskaya D.V. An application of genetic algorithms with binary and real coding for approximate synthesis of suboptimal control in deterministic systems. *Avtomat. i Telemekh.*, 2011, no. 11, pp. 117–129 (in Russ.).
- [16] Luus R. *Iterative dynamic programming*. N.Y.: Chapman & Hall, 2000. 331 p.
- [17] Tewari A. *Advanced control of aircraft, spacecraft and rockets*. N.Y.: John Wiley & Sons, 2011. 448 p.

Статья поступила в редакцию 18.12.2014

Пантелеев Андрей Владимирович — д-р физ.-мат. наук, профессор, заведующий кафедрой “Математическая кибернетика” МАИ (НИУ).

Московский авиационный институт (национальный исследовательский университет), Российская Федерация, 125993, Москва, Волоколамское ш., д. 4.

Panteleev A.V. — Dr. Sci. (Phys.-Math.), Professor, Head of Mathematical Cybernetics department, Moscow Aviation Institute (National Research University).

Moscow Aviation Institute (National Research University), Volokolamskoe shosse 4, Moscow, 125993 Russian Federation.

Пановский Валентин Николаевич — аспирант кафедры “Математическая кибернетика” МАИ (НИУ).

Московский авиационный институт (национальный исследовательский университет), Российская Федерация, 125993, Москва, Волоколамское ш., д. 4.

Panovskiy V.N. — post-graduate student of Mathematical Cybernetics department, Moscow Aviation Institute (National Research University).

Moscow Aviation Institute (National Research University), Volokolamskoe shosse 4, Moscow, 125993, Russian Federation.

#### **Просьба ссылаться на эту статью следующим образом:**

Пантелеев А.В., Пановский В.Н. Применение обобщенного инверсного интервального метода глобальной условной оптимизации в задаче поиска оптимального программного управления // Вестник МГТУ им. Н.Э. Баумана. Сер. Приборостроение. 2016. № 1. С. 33–50. DOI: 10.18698/0236-3933-2016-1-33-50

#### **Please cite this article in English as:**

Panteleev A.V., Panovskiy V.N. Application of the generalized inverse interval method of global constrained optimization for optimal program control problem. *Vestn. Mosk. Gos. Tekh. Univ. im. N.E. Baumana, Priborost.* [Herald of the Bauman Moscow State Tech. Univ., Instrum. Eng.], 2016, no. 1, pp. 33–50.  
DOI: 10.18698/0236-3933-2016-1-33-50