МАТЕМАТИКА 2015. Т. 25. Вып. 4

УДК 517.977.58

© *А. А. Зимовец, А. Р. Матвийчук*

ПАРАЛЛЕЛЬНЫЙ АЛГОРИТМ ПРИБЛИЖЕННОГО ПОСТРОЕНИЯ МНОЖЕСТВ ДОСТИЖИМОСТИ НЕЛИНЕЙНЫХ УПРАВЛЯЕМЫХ СИСТЕМ [1]

Статья посвящена исследованию эффективности применения технологии параллельных вычислений на многопроцессорных системах с общей памятью для задач приближенного расчета множеств достижимости нелинейных управляемых систем в конечномерном евклидовом пространстве. В рамках исследования предложен параллельный алгоритм приближенного построения множеств достижимости, основанный на пошаговой вычислительной схеме с использованием узлов «кубических» сеток для аппроксимации множеств. Предложенный алгоритм предназначен для проведения расчетов на ЭВМ архитектуры SMP и решает вопросы разделения задачи на отдельные подзадачи, синхронизации работы параллельных частей алгоритма и равномерного распределения нагрузки между процессорами. Численное моделирование примеров на ЭВМ с двумя 4-ядерными процессорами с использованием предложенного в статье параллельного алгоритма показало высокую эффективность применения технологии параллельных вычислений для расчета множеств достижимости сеточными методами.

Ключевые слова: множество достижимости, параллельный алгоритм, управляемая система, сеточный метод.

Введение

В работе рассматривается нелинейная управляемая система, функционирующая на конечном промежутке времени и в конечномерном пространстве.

В ряде важнейших задач теории управления, таких как, например, задача о сближении управляемой системы с целью, задача об оптимальном быстродействии, задача об обводе препятствий в фазовом пространстве системы, одним из эффективных подходов к построению решений является подход, в котором основным компонентом разрешающей конструкции является множество достижимости. В ряде задач о сближении множество разрешимости строится как интегральная воронка управляемой системы, записанной в так называемом «обратном» времени. Интегральная воронка управляемой системы представляет собой множество в пространстве позиций, (временны́ми) сечениями которого являются множества достижимости системы. При конструировании решений упомянутых задач управления основная тяжесть и основной объем вычислений приходятся на конструирование множеств достижимости системы.

Отметим, что во многих задачах оптимального управления множество достижимости дает нам представление о возможностях конкретной динамической системы, а построение этого множества зачастую является обязательным этапом при решении задач. Поэтому крайне важно уметь выделять в фазовом пространстве системы ее множества достижимости, отвечающие различным моментам времени. К сожалению, эффективное аналитическое описание множеств возможно лишь в редких случаях. В связи с этим актуальна задача приближенного построения множеств достижимости и, соответственно, интегральных воронок управляемых систем. Настоящая работа продолжает и дополняет исследования [1–7], связанные с этой тематикой.

[1] Работа выполнена при финансовой поддержке РФФИ (проект № 14–01–00486) и Программы Президиума РАН «Математические задачи современной теории управления».

МАТЕМАТИКА 2015. Т. 25. Вып. 4

§ 1. Постановка задачи и общая схема ее решения

Пусть на промежутке времени $[t_0, \vartheta]$, $t_0 < \vartheta < \infty$, задана управляемая система

$$\frac{dx}{dt} = f(t, x, u), \quad x \in \mathbb{R}^n. \tag{1.1}$$

Здесь u — вектор управления, удовлетворяющий включению $u \in P$; P — компакт в евклидовом пространстве $\mathbb{R}^p$.

Предполагается, что выполнены следующие условия.

Условие 1. Вектор-функция $f(t, x, u)$ определена и непрерывна на множестве $[t_0, \vartheta] \times \mathbb{R}^n \times P$, и для любой ограниченной и замкнутой области $\mathcal{D}$ в $[t_0, \vartheta] \times \mathbb{R}^n$ существует такая постоянная $L = L(\mathcal{D}) \in (0, \infty)$, что

$$\|f(t, x_*, u) - f(t, x^*, u)\| \leqslant L \|x_* - x^*\|, \quad (t, x_*, u) \text{ и } (t, x^*, u) \text{ из } \mathcal{D} \times P.$$

Условие 2. Существует такая постоянная $\mu \in (0, \infty)$, что

$$\|f(t, x, u)\| \leqslant \mu(1 + \|x\|), \quad (t, x, u) \in [t_0, \vartheta] \times \mathbb{R}^n \times P.$$

Условие 3. Множество $F(t, x) = f(t, x, P) = \{f(t, x, u) \colon u \in P\}$ выпукло при любых $(t, x) \in \in [t_0, \vartheta] \times \mathbb{R}^n$.

Здесь $\|f\|$ — норма вектора f в евклидовом пространстве.

Приведем некоторые известные определения тех понятий, о которых идет речь в настоящей работе. Под допустимым управлением $u(t)$, $t \in [t_0, \vartheta]$, понимаем измеримую вектор-функцию $u : [t_0, \vartheta] \to P$. Символом $X(t^*, t_*, x_*) \subset \mathbb{R}^n$ $(t_0 \leqslant t_* < t^* \leqslant \vartheta$, $x_* \in \mathbb{R}^n)$ обозначим множество достижимости системы (1.1), соответствующее моменту t^* и начальному условию $x(t_*) = x_*$. Символом $X(t_*, x_*) = \bigcup\limits_{t^* \in [t_*, \vartheta]} \big(t^*, X(t^*, t_*, x_*)\big) \subset [t_0, \vartheta] \times \mathbb{R}^n$ обозначим интегральную воронку системы (1.1) с начальной позицией $(t_*, x_*) \in [t_0, \vartheta] \times \mathbb{R}^n$. Здесь $(t^*, X^*) = \big\{(t^*, x^*) \colon x^* \in X^*\big\}$, $X^* \subset \mathbb{R}^n$.

При условиях, которые наложены на систему (1.1), множество достижимости $X(t^*, t_*, x_*)$ системы (1.1) есть в то же время и множество достижимости дифференциального включения

$$\frac{dx}{dt} \in F(t, x), \quad x(t_*) = x_*. \tag{1.2}$$

Учитывая это, получаем, что множество $X(t^*, t_*, x_*)$ есть замкнутое множество в $\mathbb{R}^n$. Принимая во внимание еще и ограниченность множества $X(t^*, t_*, x_*)$, получаем, что $X(t^*, t_*, x_*)$ — компакт в $\mathbb{R}^n$. Более того, оказывается, что если X_* — компакт в $\mathbb{R}^n$, то и $X(t^*, t_*, X_*) = \bigcup\limits_{x_* \in X_*} X(t^*, t_*, x_*)$ также компакт в $\mathbb{R}^n$. Можно сказать, что $X(t^*, t_*, X_*)$ есть множество всех тех точек $x^* \in \mathbb{R}^n$, в которые в момент $t^* \in [t_*, \vartheta]$ приходит управляемая система (1.1) (дифференциальное включение (1.2)) под воздействием всевозможных допустимых управлений $u(t)$, $t \in [t_*, t^*]$, отправляясь в момент t_* из множества X_*.

Нас будет интересовать задача приближенного конструирования множества $X(\vartheta, t_0, X_0)$. Это приближенное конструирование мы связываем с дискретизацией промежутка времени $[t_0, \vartheta]$ и фазового пространства $\mathbb{R}^n$ системы (1.1).

При приближенном конструировании множества $X(\vartheta, t_0, X_0)$ (X_0 — компакт в $\mathbb{R}^n$) мы принимаем во внимание полугрупповое свойство множеств достижимости $X(t) = X(t, t_0, X_0)$, $t \in [t_0, \vartheta]$:

$$X(t^*) = X\big(t^*, t_*, X(t_*)\big), \quad t_0 \leqslant t_* < t^* \leqslant \vartheta.$$

Дискретизация промежутка $[t_0, \vartheta]$ в сочетании с применением полугруппового свойства множеств достижимости задает определенное направление для приближенного конструирования множеств $X(t)$, $t \in [t_0, \vartheta]$ и, в частности, множества $X(\vartheta) = X(\vartheta, t_0, X_0)$.

Охарактеризуем схематично это направление.

Сначала вводится разбиение $\Gamma = \{t_0, t_1, \ldots, t_N = \vartheta\}$ промежутка $[t_0, \vartheta]$ с одинаковыми шагами $\Delta_i = t_{i+1} - t_i = \Delta = \frac{1}{N}(\vartheta - t_0)$, $i = \overline{0, N-1}$, где диаметр $\Delta = \Delta(\Gamma)$ считаем малым числом.

Дискретному разбиению Γ промежутка $[t_0, \vartheta]$ отвечает следующее рекуррентное соотношение:

$$X(t_i) = X\big(t_i, t_{i-1}, X(t_{i-1})\big), \quad i = \overline{1, N}.$$

Если бы на каждом промежутке $[t_{i-1}, t_i]$ разбиения Γ мы обладали возможностью при малых $\Delta = \Delta(\Gamma)$ точно вычислять множество $X\big(t_i, t_{i-1}, X(t_{i-1})\big)$, то, продвигаясь последовательно по шагам $[t_{i-1}, t_i]$ вперед, мы бы в конечном итоге на последнем шаге $[t_{N-1}, \vartheta]$ вычислили и множество $X(\vartheta)$.

Поскольку осуществить точное вычисление множеств $X(t_i)$ мы не в состоянии, то обратимся к приближенному вычислению на каждом шаге множеств $X(t_i)$, $t_i \in \Gamma$. При этом мы используем идеологию ломаных Эйлера. Использование конструкций типа ломаных Эйлера при проведении приближенных вычислений означает для системы (1.1) (равно и для д. в. (1.2)) подмену при малых $\delta = t^* - t_* > 0$ множеств $X(t^*, t_*, x_*)$ (локальных множеств достижимости системы (1.1)) множеством $\widetilde{X}(t^*, t_*, x_*) = x_* + \delta F(t_*, x_*) = \big\{x_* + \delta f_* \colon f_* \in F(t_*, x_*)\big\}$. Эти множества более приемлемы для вычислений, чем множества $X(t^*, t_*, x_*)$. Так, они являются выпуклыми компактами в $\mathbb{R}^n$, гомотетичными множеству $F(t_*, x_*)$, а в ряде задач они допускают эффективное аналитическое описание.

Введя обозначение $\widetilde{X}(t^*, t_*, X_*) = \bigcup\limits_{x_* \in X_*} \widetilde{X}(t^*, t_*, x_*)$, мы можем приближенно вычислять множества $X(t_i)$, $i = \overline{0, N}$ как последовательность множеств $X_a(t_i) = X_a(t_i, t_0, X_0)$, $i = \overline{0, N}$, согласно рекуррентному соотношению

$$X_a(t_0) = X_0, \quad X_a(t_i) = \widetilde{X}\big(t_i, t_{i-1}, X_a(t_{i-1})\big), \quad i = \overline{1, N}. \tag{1.3}$$

В этих соотношениях буква a — символ аппроксимации.

Тот факт, что последовательность множеств $X_a(t_i)$, $i = \overline{0, N}$, является аппроксимацией для последовательности $X(t_i)$, $i = \overline{0, N}$, подтверждается следующим предельным соотношением:

$$\lim_{\Delta \downarrow 0} \max_{t_i \in \Gamma} d\big(X(t_i), X_a(t_i)\big) = 0. \tag{1.4}$$

Здесь $d\big(X_*, X^*\big) = \max\big\{h(X_*, X^*), h(X^*, X_*)\big\}$ — хаусдорфово расстояние между множествами X_* и X^* в $\mathbb{R}^n$; $h(X_*, X^*) = \max\limits_{x_* \in X_*} \min\limits_{x^* \in X^*} \|x_* - x^*\|$.

В случае когда невозможно вычислять (точно) или описывать аналитически множества (1.3), возникает необходимость в коррекции отображения $(t^*, t_*, X_*) \mapsto \widetilde{X}(t^*, t_*, X_*)$, которая бы приводила к скорректированной рекуррентной последовательности (1.3), вычислимой точно.

Ориентируясь на такую коррекцию последовательности (1.3), мы подменяем отображение $(t^*, t_*, X_*) \mapsto X(t^*, t_*, X_*)$, задающее (1.3), некоторым отображением

$$(t^*, t_*, X_*) \mapsto X^{(\delta)}(t^*, t_*, X_*) \in \operatorname{comp}(\mathbb{R}^n), \quad \delta = t^* - t_* > 0, \tag{1.5}$$

которое порождает последовательность

$$X^a(t_i) = X^{(\delta)}\big(t_i, t_{i-1}, X^a(t_{i-1})\big), \quad i = \overline{1, N}, \tag{1.6}$$

удовлетворяющую соотношению (аналогу соотношения (1.4))

$$\lim_{\Delta \downarrow 0} \max_{t_i \in \Gamma} d\big(X(t_i), X^a(t_i)\big) = 0. \tag{1.7}$$

Здесь $X^a(t_0) = X_0^{(\Delta)}$ — некоторое задаваемое нами конечное множество в $\mathbb{R}^n$, стесненное неравенством $d\left(X_0, X_0^{(\Delta)}\right) \leqslant \sigma^*(\Delta)$, где $\sigma^*(\delta)$ — положительная функция на $(0, \infty)$, для которой $\sigma^*(\Delta) \downarrow 0$ при $\Delta \downarrow 0$.

Отметим, что предельное соотношение (1.7) придает последовательности (1.6) статус аппроксимационной последовательности.

Теперь дадим более детальное описание отображения (1.5). Для этого определим отображение $(t_*, x_*) \mapsto F^{(\delta)}(t_*, x_*)$ на $[t_0, \vartheta] \times \mathbb{R}^n$, зависящее от параметра $\delta = t^* - t_* > 0$, как отображение, аппроксимирующее отображение $(t_*, x_*) \mapsto F(t_*, x_*)$ таким образом, чтобы выполнялись следующие условия

Условие 4.

$$\sup_{(t_*,x_*)\in[t_0,\vartheta]\times\mathbb{R}^n} d\big(F(t_*,x_*), F^{(\delta)}(t_*,x_*)\big) \leqslant \varphi^*(\delta),$$

где выбранная нами функция $\varphi^*(\delta)$ на $(0,\infty)$ такова, что $\varphi^*(\delta) \downarrow 0$ при $\delta \downarrow 0$.

Условие 5. Для любой ограниченной и замкнутой области $\mathcal{D} \subset [t_0, \vartheta] \times \mathbb{R}^n$ множества $F^{(\delta)}(t_*, x_*)$ при любых $(t_*, x_*) \in \mathcal{D}$ состоят из конечного числа точек.

Далее введем в рассмотрение ту ограниченную и замкнутую область $\mathbb{D} \subset [t_0, \vartheta] \times \mathbb{R}^n$, в которой будут заведомо содержаться все множества $\big(t_i, X(t_i)\big)$, $i = \overline{0, N}$, а также всевозможные их аппроксимации, возникающие в процессе конструирования.

А именно, зададим такое $\mu_0 \in (0,\infty)$, что $h\big(X_0, \{\mathbf{0}\}\big) < \mu_0$; здесь $\mathbf{0}$ — нуль в пространстве $\mathbb{R}^n$. Полагаем

$$\mathbb{D} = \Big\{(t,x)\colon t \in [t_0,\vartheta], x \in B\big(\mathbf{0};\mu(t)\big)\Big\} \subset [t_0,\vartheta]\times\mathbb{R}^n,$$

где

$$\mu(t) = \Big(\mu_0 + \sigma^*(\vartheta - t_0) + (t-t_0)\big(\mu + \varphi^*(\vartheta - t_0)\big)\Big)e^{\mu(t-t_0)}, \quad t \in [t_0,\vartheta]$$

и число μ определено в условии 2; $B(\mathbf{0},\gamma)$ — замкнутый шар в $\mathbb{R}^n$ с центром в $\mathbf{0}$ и радиуса γ.

Как нетрудно заметить, область $\mathbb{D}$ включает в себя интегральную воронку дифференциального включения $\frac{dx}{dt} \in B\big(\mathbf{0};\varphi^*(\vartheta - t_0) + \mu(1+\|x\|)\big)$ на промежутке $[t_0,\vartheta]$ с начальным множеством $\mathbb{D}(t_0) = B\big(\mathbf{0};\mu_0 + \sigma^*(\vartheta - t_0)\big)$.

Отметим, что область $\mathbb{D}$ выбрана нами так, что при некотором достаточно малом $\varepsilon_* \in (0,\infty)$ имеет место $X(t,t_0,X_0) + B(\mathbf{0};\varepsilon_*) \subset \mathbb{D}(t)$, $t \in [t_0,\vartheta]$.

Именно эту область $\mathbb{D}$ и отвечающие ей константы $L = L(\mathbb{D})$ и $K = K(\mathbb{D}) = \max\big\{\|f\|\colon f = f(t,x,u), (t,x,u) \in \mathbb{D}\times P\big\}$ мы имеем в виду в последующих рассуждениях и оценках. Таким образом, каждое множество

$$\begin{gathered} X^{(\delta)}(t^*,t_*,x_*) = x_* + \delta F^{(\delta)}(t_*,x_*) = \big\{x_* + \delta f_*\colon f_* \in F^{(\delta)}(t_*,x_*)\big\}, \\ (t_*,x_*) \in \mathbb{D}, \quad t^* \in [t_*,\vartheta], \end{gathered} \tag{1.8}$$

мы в состоянии вычислить за приемлемое время.

Множества $X^{(\delta)}(t^*,t_*,x^*)$, $(t_*,x_*) \in \mathbb{D}$, есть те самые компоненты аппроксимации множества $X(\vartheta) = X(\vartheta,t_0,X_0)$, которую мы будем проводить в следующем разделе. Отметим, что, исходя из определения множеств $X^{(\delta)}(t^*,t_*,x_*)$, справедлива важная оценка (см., например, [5])

$$\sup_{(t_*,x_*)\in\mathbb{D}} d\big(X(t^*,t_*,x_*), X^{(\delta)}(t^*,t_*,x_*)\big) \leqslant \xi(\delta), \quad t^* \in [t_*,\vartheta]. \tag{1.9}$$

Здесь функция $\xi(\delta)$ положительна на $(0,\infty)$ и имеет вид $\xi(\delta) = \delta\xi^*(\delta)$, где $\xi^*(\delta) \downarrow 0$ при $\delta \downarrow 0$.

Оценка (1.9) показывает, что хаусдорфовы отклонения локальных аппроксимационных множеств достижимости $X^{(\delta)}(t^*,t_*,x_*)$ на $\mathbb{D}$ от локальных точных множеств достижимости $X(t^*,t_*,x_*)$ имеют более высокий порядок малости по $\delta = t^* - t_* > 0$, чем первый. Это дает нам уверенность в том, что аппроксимации множества $X(\vartheta) = X(\vartheta,t_0,X_0)$, уже глобальные на промежутке $[t_0,\vartheta]$ и компонующиеся из множеств $X^{(\delta)}(t^*,t_*,x_*)$, $(t_*,x_*) \in \mathbb{D}$ как из элементарных «кирпичиков», будут давать хорошие приближения этого множества $X(\vartheta)$.

Итак, введем на базе отображения $(t_*, x_*) \mapsto F^{(\delta)}(t_*, x_*)$ следующее отображение (1.5), заданное равенством

$$X^{(\delta)}(t^*, t_*, X_*) = \bigcup_{\overline{x}_* \in X_*} X^{(\delta)}(t^*, t_*, \overline{x}_*), \tag{1.10}$$

$t_0 \leqslant t_* < t^* \leqslant \vartheta$, X_* — конечное множество в $\mathbb{R}^n$, $\delta = t^* - t_*$.

Отображение $(t^*, t_*, X_*) \mapsto X^{(\delta)}(t^*, t_*, X_*)$ удовлетворяет неравенству

$$d\big(X(t^*, t_*, X_*), X^{(\delta)}(t^*, t_*, X_*)\big) \leqslant \xi(\delta), \tag{1.11}$$

$t_0 \leqslant t_* \leqslant t^* \leqslant \vartheta$, X_* — конечное множество в $\mathbb{R}^n$, $\delta = t^* - t_*$.

Принимая во внимание оценку (1.11), имеющую место для всех множеств $(t_*, X_*) \subset \mathbb{D}$, получаем, что последовательность множеств $X^a(t_i)$, $t_i \in \Gamma$ (1.6), удовлетворяет соотношению (1.7). В частности, множества $X^a(t_N)$, $t_N \in \Gamma$, сходятся в хаусдорфовой метрике к множеству $X(\vartheta)$ при $\Delta = \Delta(\Gamma) \downarrow 0$.

Приведенные рассуждения составляют теоретическую основу для разработки конструкций, аппроксимирующих множества достижимости.

§ 2. Последовательный и параллельный алгоритмы приближенного построения множеств достижимости

В статье предлагается реализация описанной выше схемы приближенного построения множеств достижимости на ЭВМ путем дискретизации пространства $\mathbb{R}^n$ узлами «кубической» сетки Λ_h с шагом h. Кроме того, для получения конечного набора векторов пространство $\mathbb{R}^p$ предлагается разбить узлами «кубической» сетки Λ_q с шагом q, что приведет к подмене множества P с бесконечным количеством векторов множеством P^q, состоящим из конечного набора векторов управлений.

Ввиду достаточно общих предположений относительно функции $f(t, x, u)$ мы считаем, что эффективное аналитическое описание множеств $F(t_*, x_*)$, $(t_*, x_*) \in \mathbb{D}$, отсутствует, поэтому аппроксимацию этих множеств мы будем строить в 2 этапа. На первом этапе мы вычисляем множество точек $F^q(t_*, x_*) = \big\{f(t_*, x_*, u)\colon u \in P^q\big\}$. На втором этапе мы сопоставляем точкам множества $F^q(t_*, x_*)$ ближайшие к ним узлы «кубической» сетки Λ_h с шагом h, формируя таким образом множество $F^a(t_*, x_*)$, которое аппроксимирует множество $F(t_*, x_*)$.

Процесс приближенного построения множества достижимости на ЭВМ можно разделить на две стадии: подготовительную и вычислительную. Подготовительная стадия включает в себя следующие этапы:

(1) дискретизация промежутка времени $[t_0, \vartheta]$ разбиением $\Gamma = \{t_0, t_1, \ldots, t_N = \vartheta\}$ с одинаковыми шагами $\Delta_i = t_{i+1} - t_i = \Delta = \frac{1}{N}(\vartheta - t_0)$, $i = \overline{0, N-1}$;

(2) сопоставление компакту P множества P^q, состоящего из узлов «кубической» сетки Λ_q с шагом q;

(3) построение множества $X^a(t_0)$, аппроксимирующего начальное множество X_0 и состоящего из узлов «кубической» сетки Λ_{h_Δ} с шагом $h_\Delta = h\Delta$.

Здесь и далее для аппроксимаций $X^a(t_i)$, $i = \overline{0, N-1}$, вместо сетки Λ_h будем рассматривать сетку Λ_{h_Δ}. Это связано с тем, что используемое для вычислений соотношение (1.8) представляет собой комбинацию преобразований масштабирования и параллельного переноса относительно множеств $F^a(t_*, x_*)$. Следовательно, все точки множеств $X^a(t_i)$, $t_i \in \Gamma$, построенные с использованием этого соотношения, будут расположены в узлах «кубической» сетки Λ_{h_Δ} с шагом $h_\Delta = h\Delta$ при условии, что начальные точки также были расположены в узлах этой сетки.

Вычислительная стадия выполняется последовательно для каждого $t_i \in \Gamma$, $i = \overline{0, N-1}$, и состоит из следующих этапов:

(1) сопоставление множествам $F(t_i, x_*)$, $x_* \in X^a(t_i)$, множеств $F^a(t_i, x_*)$, состоящих из узлов «кубической» сетки Λ_h с шагом h;

(2) построение аппроксимаций локальных множеств достижимости точек $x_* \in X^a(t_i)$ с использованием соотношения (1.8);

(3) объединение аппроксимаций локальных множеств достижимости точек $x_* \in X^a(t_i)$ в соответствии с соотношением (1.10).

Приведем простейший последовательный алгоритм, реализующий описанную схему приближенного построения множеств достижимости нелинейных управляемых систем.

Алгоритм 1. Последовательный алгоритм приближенного построения множества достижимости

1. Цикл 1: для каждого i от 0 до $N-1$ выполнить
2. Провести инициализацию структур данных
3. Цикл 2: для каждой точки $x_* \in X^a(t_i)$ выполнить
4. Цикл 3: для каждого вектора управления $u \in P^q$ выполнить
5. Вычислить $f_* = f(t_i, x_*, u)$
6. Сопоставить точке f_* ближайшую к ней точку $\widehat{f_*}$ сетки Λ_h
7. Вычислить $\widehat{x}_* = x_* + \Delta \widehat{f_*}$
8. Если $\widehat{x}_* \notin X^a(t_{i+1})$, то построить $X^a(t_{i+1}) = X^a(t_{i+1}) \bigcup \{\widehat{x}_*\}$
9. Конец цикла 3
10. Конец цикла 2
11. Конец цикла 1
12. Сохранить множество $X^a(t_N)$ на жесткий диск

При необходимости в алгоритме 1 между шагами 10 и 11 также можно добавить операцию сохранения множества $X^a(t_{i+1})$ на жесткий диск. Добавление этой операции немного замедлит работу алгоритма, но при этом позволит использовать алгоритм 1 для приближенного построения интегральной воронки системы (1.1) на промежутке времени $[t_0, \vartheta]$, а также прерывать и возобновлять вычисления на любом шаге цикла 1. Инициализация структур данных в операции 2 включает в себя такие действия, как предварительное выделение оперативной памяти, задание начального состояния и т. п.

Вследствие достаточно общих предположений относительно функции $f(t, x, u)$ представляется затруднительным дать точное описание взаимосвязи между параметрами Δ, h и q, а следовательно, и оценить вычислительную сложность алгоритма 1. В работе [6, с. 22] для аналогичной вычислительной схемы, использующей конструкции на базе ломаных Эйлера, взаимосвязь между параметрами h и Δ устанавливается в виде $h = \Omega\sqrt{\Delta}$, где Ω — некоторое положительное число. Для выбора параметра q можно использовать следующее поддающееся автоматизации эмпирическое правило: уменьшаем величину q до тех пор, пока это приводит к изменению количества точек в каком-либо из множеств $X^a(t_i)$, $i = \overline{1, N}$.

Достоинством алгоритма 1 является простота реализации для пространств любой размерности. Основной недостаток алгоритма 1 состоит в том, что во многих случаях для получения удовлетворительного ответа требуется неприемлемо большое время вычислений. Это вынуждает нас искать пути и подходы, направленные на сокращение времени счета.

Уменьшать время вычислений для приближенного построения множества достижимости системы (1.1) можно различными способами. Так, например, в работе [7] предлагается метод, позволяющий при определенных условиях сократить объем вычислений за счет использования точек только из некоторого приграничного слоя множеств $X^a(t_i)$, $i = \overline{0, N-1}$. Другой возможный способ уменьшения времени вычислений состоит в более полном использовании ресурсов и технологий современных ЭВМ. Среди таких технологий можно выделить конвейерную обработку команд, суперскалярную архитектуру, векторную обработку данных, внеочередное выполнение команд, различные многоядерные и многопроцессорные архитектуры как с общей, так и с распределенной памятью и т. д. При этом весь комплекс таких технологий можно разделить на две категории: технологии одной из этих категорий предоставляются программисту в качестве инструмента для организации наиболее эффективного вычислительного процесса, технологии из другой категории программисту недоступны и задействуются в вычислительной системе автоматически при выполнении программы. Далее мы исследуем влияние технологии параллельных вычислений на примере симметричной многопроцессорной системы (SMP-системы) на скорость вычисления множества достижимости с использованием модифицированного для SMP-систем алгоритма 1.

Класс симметричных многопроцессорных систем сейчас является одним из наиболее доступных классов ЭВМ: к нему относится большинство современных персональных компьютеров и серверов. Компьютеры этой архитектуры характеризуются тем, что несколько равноправных процессоров совместно работают с общими модулями оперативной памяти и периферийными устройствами. Равноправие здесь понимается в том смысле, что любой из процессоров может выполнять любой процесс, будь то процесс операционной системы или пользовательский процесс.

Рассмотрим вариант ускорения вычислительного процесса в симметричных многопроцессорных системах за счет распараллеливания алгоритма 1. Для этого предварительно зададим множество A в форме n-мерного прямоугольного параллелепипеда, которое назовем ареной. Размер арены A выберем таким образом, чтобы она вмещала в себя все сечения $\mathbb{D}(t) = \{x \in \mathbb{R}^n \colon (t, x) \in \mathbb{D}\}$ области $\mathbb{D}$, определенной нами в предыдущем параграфе. Построенная таким образом арена будет заведомо вмещать в себя все множества $X^a(t_i)$, $i = \overline{0, N}$. Выбранная нами форма арены A позволяет разбить ее на набор более мелких областей в форме n-мерных прямоугольных параллелепипедов одинакового размера. Такое разбиение арены A положено нами в основу приведенного далее алгоритма.

Алгоритм 2. Параллельный алгоритм приближенного построения множества достижимости

1. Цикл 1: для каждого i от 0 до $N-1$ выполнить
2. Провести инициализацию структур данных
3. Войти в параллельную секцию
4. Цикл 2: выбрать следующий блок точек b из очереди текущего потока и выполнить
5. Цикл 3: выбрать из блока b следующую точку $x_* \in X^a(t_i)$ и выполнить
6. Построить $V = \varnothing$
7. Цикл 4: для каждого $u \in P^q$ выполнить
8. Вычислить $f_* = f(t_i, x_*, u)$
9. Сопоставить точке f_* ближайшую к ней точку $\widehat{f}_*$ сетки Λ_h
10. Вычислить $\widehat{x}_* = x_* + \Delta\widehat{f}_*$
11. Построить $V = V \bigcup \{\widehat{x}_*\}$

12. Конец цикла 4
13. Цикл 5: для каждой точки $\widehat{x}_* \in V$ выполнить
14. Определить область арены A, содержащую $\widehat{x}_*$, и соответствующий ей блок $\widehat{b}$
15. Если $\widehat{b} \notin B_{i+1}$, то выполнить
16. Войти в критическую секцию $\big(1, hash(\widehat{b})\big)$
17. Построить $B_{i+1} = B_{i+1} \bigcup \{\widehat{b}\}$
18. Добавить блок $\widehat{b}$ в очередь потока № $hash(\widehat{b})$
19. Выйти из критической секции $\big(1, hash(\widehat{b})\big)$
20. Конец если
21. Конец цикла 5
22. Войти в критическую секцию $\big(2, 1\big)$
23. Цикл 6: для каждой точки $\widehat{x}_* \in V$ и соответствующего ей блока $\widehat{b}$ выполнить
24. Если $\widehat{x}_* \notin \widehat{b}$, то добавить точку $\widehat{x}_*$ в блок $\widehat{b}$
25. Конец цикла 6
26. Выйти из критической секции $\big(2, 1\big)$
27. Конец цикла 3
28. Конец цикла 2
29. Выйти из параллельной секции
30. Конец цикла 1
31. Сохранить множество $X^a(t_N)$ на жесткий диск

Здесь B_i — наборы непересекающихся блоков, в каждом из которых содержится хотя бы одна точка множества $X^a(t_i)$, $i = \overline{0, N}$. Наборы B_i таковы, что каждый блок такого набора соответствует какой-либо из областей, на которые разбита арена A, и в сумме все блоки множества B_i содержат все точки соответствующего ему множества $X^a(t_i)$. Символом $hash(\widehat{b})$ обозначена хэш-функция от адреса ячейки памяти в структуре данных, соответствующей множеству B_{i+1}, в которой должен быть размещен указатель на блок $\widehat{b}$. Эта функция принимает целочисленные значения в промежутке от 1 до количества параллельно выполняемых потоков. Параллельной секцией в алгоритме 2 мы называем участок кода, который может выполняться одновременно несколькими вычислительными потоками.

В алгоритме 2 можно выделить три основные части. В первой части, состоящей из операций 6–12, выполняется построение локального множества достижимости V точки x_* из блока b. Во второй части, состоящей из операций 13–21, выполняется создание блоков, которые будут задействованы для хранения множества $X^a(t_{i+1})$. В третьей части, состоящей из операций 22–26, вычисленные в первой части точки добавляются в соответствующие им блоки.

Рассмотрим алгоритм 2 более подробно. Поскольку алгоритм 2 относится к классу параллельных алгоритмов, то нас будет интересовать, каким образом в алгоритме 2 даются ответы на следующие вопросы:

(1) разделение задачи на отдельные подзадачи;

(2) синхронизация работы параллельных частей алгоритма и доступа к общим участкам оперативной памяти (для тех частей подзадач, которые не могут выполняться независимо друг от друга);

(3) равномерное распределение нагрузки между процессорами.

Рассмотрим, каким образом в алгоритме 2 решается вопрос разделения задачи на отдельные подзадачи. Для этого заметим, что построение локального множества достижимости одной точки множества $X^a(t_i)$, $0 \leqslant i < N$, может выполняться независимо от построения локального множества достижимости любой другой точки множества $X^a(t_i)$, так как для выполнения этих операций не требуется доступ к общим участкам памяти. Таким образом, для разбиения задачи на отдельные подзадачи мы можем воспользоваться методом геометрической декомпозиции. В нашем случае этот метод состоит в том, что исходная арена A разбивается на набор областей одинакового размера в форме n-мерных прямоугольных параллелепипедов и в качестве отдельной подзадачи рассматривается задача построения множества достижимости точек множества $X^a(t_i)$, расположенных в одной такой области, и добавления вычисленных точек в структуру данных, предназначенную для хранения множества $X^a(t_{i+1})$. В алгоритме 2 указанные действия соответствуют операциям 3–29. Однако лишь некоторые из этих операций могут выполняться независимо друг от друга. Гораздо сложнее дело обстоит с добавлением вычисленных точек в множество $X^a(t_{i+1})$, поскольку вычислительные потоки не могут модифицировать соответствующую этому множеству структуру данных независимо друг от друга. Процесс добавления точек в множество $X^a(t_{i+1})$ должен быть строго синхронизирован, чтобы ни в какой момент времени никакие два потока не модифицировали одни и те же участки памяти. Так мы переходим к вопросу синхронизации работы параллельных частей алгоритма.

Как правило, операционная система предоставляет программисту определенный набор примитивов синхронизации: критические секции, семафоры, барьеры и т. д. Таким образом, программист освобождается от технических аспектов вопроса синхронизации и его задача сводится к составлению корректного алгоритма. В нашем случае используются два механизма: критические секции и барьеры. Напомним, что критической секцией называют участок программного кода, в котором осуществляется доступ к вычислительным ресурсам, не допускающим одновременного использования несколькими вычислительными потоками. Таким образом, соответствующий примитив синхронизации не позволяет нескольким потокам одновременно входить в критическую секцию с одним и тем же номером. Если какой-либо из потоков в данный момент находится в критической секции, то все остальные потоки, желающие войти в ту же самую критическую секцию, должны ждать, пока поток, находящийся в критической секции, не выйдет из нее. Функция же барьерной синхронизации состоит в том, чтобы вычислительные потоки, дошедшие до определенной точки в программе, дождались, пока до той же самой точки не дойдут все остальные потоки. Барьерная синхронизация используется в операции 29 алгоритма 2, чтобы одни вычислительные потоки не начинали построение множества достижимости для следующего момента времени до тех пор, пока все остальные потоки не произведут все свои расчеты для текущего момента времени.

Процесс добавления точек в множество $X^a(t_{i+1})$ реализован во второй и третьей частях алгоритма 2. В этих частях используется синхронизация работы вычислительных потоков при помощи механизма критических секций. Рассмотрим вопросы синхронизации в каждой из этих частей по отдельности.

Во второй части алгоритма 2 выполняется создание блоков $\widehat{b}$ и добавление этих блоков в структуру данных, соответствующую множеству B_{i+1}. Организация синхронной работы вычислительных потоков в этой части алгоритма зависит от структуры данных, используемой для хранения множества B_{i+1}. Если для хранения множества B_{i+1} использовать массив указателей на блоки $\widehat{b}$ (поскольку размер арены A считается известным, то мы можем заранее рассчитать размер этого массива в процессе выполнения операции 2), то операция изменения одного элемента такого массива может выполняться независимо от операций изменения других элементов массива. Таким образом, задача синхронизации работы вычислительных потоков

сводится к запрету нескольким потокам одновременно выполнять модификацию одно и того же указателя в массиве, соответствующем множеству B_{i+1}. Это достигается путем использования критической секции в операции 16 алгоритма 2. Поскольку нам требуется исключить одновременный доступ нескольких потоков к отдельным ячейкам массива, а не ко всему массиву в целом, то мы могли бы каждой ячейке сопоставить свой уникальный номер критической секции. Тогда время ожидания потоков на входе в соответствующую критическую секцию было бы минимальным, но по причинам, связанным с распределением вычислительной нагрузки между процессорами, мы будем использовать в операции 16 количество номеров критической секции, равное количеству вычислительных потоков. В этом случае один и тот же номер критической секции будет соответствовать сразу нескольким ячейкам массива, поэтому ожидание потоков на входе в критическую секцию будет происходить в процессе вычислений чаще, чем в случае использования уникальных номеров критических секций. Однако, как показывает практика, такой подход оказывается вполне оправдан. Отметим также, что, строго говоря, после входа в критическую секцию необходимо повторить проверку, выполняемую в операции 15.

В третьей части алгоритма выполняется добавление вычисленных в первой части точек в соответствующие им блоки. В приведенном алгоритме в операции 22 используется всегда один и тот же номер критической секции. Этот подход хорошо работает при небольшом количестве процессоров в SMP-системе. В общем случае можно использовать и большее количество номеров критической секции вплоть до количества блоков в множестве B_{i+1}, поскольку модификация одного блока может выполняться независимо от модификации других блоков. Причина использования лишь одного номера критической секции для SMP-систем с небольшим количеством процессоров состоит в том, чтобы на каждую пару операций входа и выхода из критической секции приходилось как можно больше операций добавления точек в соответствующие им блоки. Таким образом мы уменьшаем долю технических затрат на выполнение операций синхронизации в программе.

Перейдем теперь к вопросу распределения нагрузки между процессорами. Для решения этого вопроса в алгоритме 2 применена концепция использования очередей заданий для каждого потока. В качестве единичного задания выступает задача расчета множества достижимости точек одного из блоков множества B_i. Таким образом, задача равномерного распределения нагрузки между процессорами сводится к задаче равномерного заполнения очередей потоков с учетом того, что каждый блок может содержать различное количество точек множества $X^a(t_i)$. В результате было решено использовать псевдослучайное распределение блоков по очередям потоков. Это достигается путем выбора в операции 18 алгоритма 2 очереди потока в соответствии со значением хэш-функции $hash(\widehat{b})$ блока $\widehat{b}$. Отметим, что в качестве такой функции можно выбирать некриптографическую хэш-функцию, поскольку в рассматриваемом алгоритме не требуется одно из важнейших свойств криптографических хэш-функций, а именно устойчивость к восстановлению прообраза. В результате мы получаем, что критическая секция в операциях 16–19 имеет двойное назначение: она синхронизирует модификацию указателей в массиве, соответствующем множеству B_{i+1}, и добавление блоков $\widehat{b}$ в очереди потоков.

§ 3. Пример численного моделирования

В качестве примера рассмотрим задачу построения множества достижимости четырехколесной тележки, динамика которой описывается уравнением

$$\begin{cases} \dot{x_1} = \cos x_3, \\ \dot{x_2} = \sin x_3, \\ \dot{x_3} = u. \end{cases} \tag{3.1}$$

Здесь (x_1, x_2) — координаты точки, расположенной посередине между задними колесами тележки; x_3 — угол поворота тележки; u — угловая скорость тележки, величиной которой мы можем управлять в пределах $|u| \leqslant 1$. В начальный момент времени $t_0 = 0$ тележка находится в положении $x(0) = (0, 0, 0)$. Требуется приближенно построить множество достижимости

системы (3.1) в момент времени $\vartheta = 5$. Арену A зададим в виде куба с ребром длиной 20 и центром в начале координат. Контур приближенно построенного на ЭВМ множества достижимости приведен на рисунке 1.

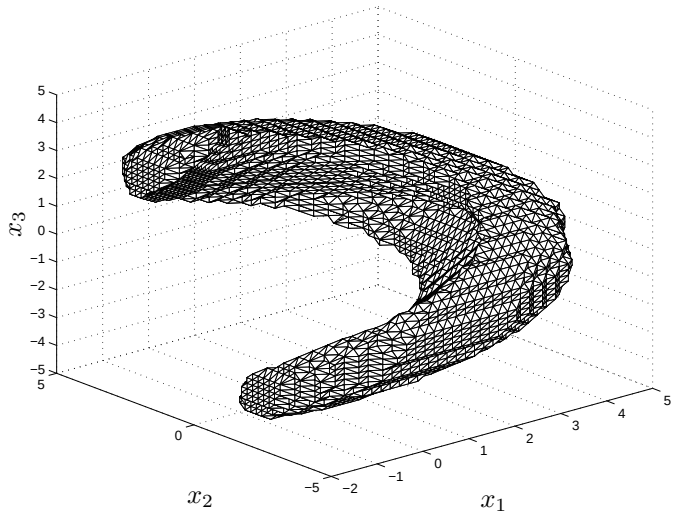

Рис. 1. Множество достижимости системы (3.1) в момент времени $t = 5$

Исследуем зависимость времени вычислений от количества задействованных вычислительных потоков с параметрами вычислений $\Delta = 0.05$, $h = 0.2$, $h_\Delta = 0.01$ и $q = 0.05$. В качестве вычислительной платформы используем серверную платформу Intel S5000PSL с двумя 4-ядерными процессорами Intel Xeon E5335 (2.0 ГГц) и 8 Гб оперативной памяти типа FB-DIMM DDR2 (PC2-5300). Считаем, что один поток выполняется на одном вычислительном ядре SMP-системы. Символом k_i обозначим величину T_1/T_i, где i — количество потоков, T_1 — время вычисления множества достижимости одним потоком, T_i — время вычисления множества достижимости i потоками. В таблице 3.1 приведены время вычислений в минутах для различного количества потоков и получившиеся коэффициенты ускорения относительно однопоточного варианта.

Таблица 3.1

Эффективность распараллеливания вычислений при использовании алгоритма 2

Кол-во потоков, i	Время счета (мин), T_i	Коэфф. ускорения, k_i
1	146.2	1
2	76.0	1.92
3	50.5	2.89
4	38.4	3.81
5	31.1	4.70
6	26.2	5.58
7	22.8	6.42
8	20.1	7.26

На рисунке 2 изображен график зависимости времени вычислений в минутах от количества задействованных потоков. На рисунке 3 изображен график зависимости коэффициента ускорения от количества задействованных потоков. Отметим, что одно из достоинств параллельного алгоритма 2 состоит в том, что он практически не требует использования дополнительной памяти по сравнению с последовательным алгоритмом 1.

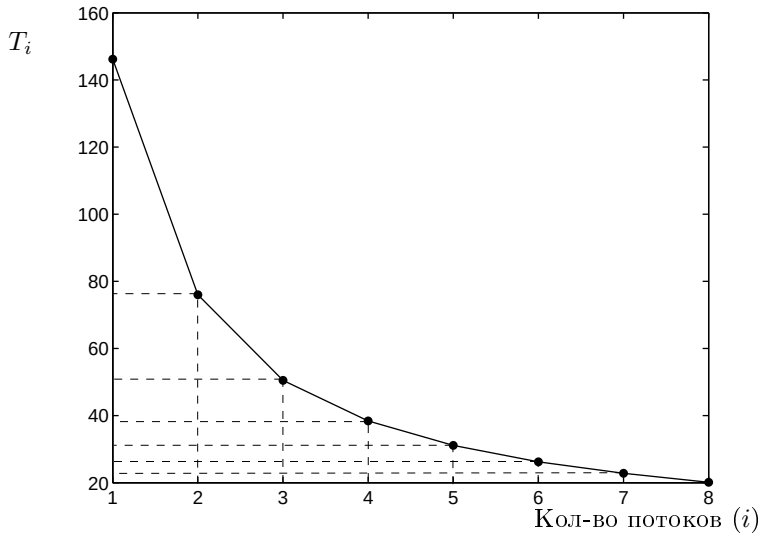

Рис. 2. Зависимость времени вычислений от количества потоков

Заключение

В статье рассмотрены два алгоритма приближенного построения множеств достижимости нелинейных управляемых систем. Алгоритм 1 является одним из наиболее простых алгоритмов решения поставленной задачи. Он легко реализуем в пространствах любой размерности и при незначительных модификациях позволяет производить расчеты для систем, в которых накладываются дополнительные ограничения на значения фазовой переменной. Алгоритм 2 использует ту же вычислительную схему, что и алгоритм 1, и имеет те же преимущества. Однако при тех же требованиях к объему оперативной памяти алгоритм 2 позволяет получить результат значительно быстрее. В процессе тестирования алгоритма 2 на 8-ядерной SMP-системе была получена практически линейная зависимость коэффициента ускорения от количества задействованных ядер. Таким образом, разработанный параллельный алгоритм показал высокую степень масштабируемости в рамках выбранной компьютерной архитектуры.

СПИСОК ЛИТЕРАТУРЫ

1. Красовский Н.Н. Теория управления движением. М.: Наука, 1968. 476 с.
2. Никольский М.С. Об одном методе аппроксимации множества достижимости для дифференциального включения // Журн. вычисл. матем. и матем. физ. 1988. Т. 28. № 8. С. 1252–1254.
3. Комаров В.А., Певчих К.Э. Об одном методе аппроксимации множеств достижимости дифференциальных включений с заданной точностью // Журн. вычисл. матем. и матем. физ. 1991. Т. 31. № 1. С. 153–157.
4. Гусейнов Х.Г., Моисеев А.Н., Ушаков В.Н. Об аппроксимации областей достижимости управляемых систем // Прикладная математика и механика. 1998. Т. 62. № 2. С. 179–187.
5. Ушаков В.Н., Матвийчук А.Р., Ушаков А.В. Аппроксимация множеств достижимости и интегральных воронок дифференциальных включений // Вестник Удмуртского университета. Математика. Механика. Компьютерные науки. 2011. Вып. 4. С. 23–39.
6. Пахотинских В.Ю. Построение решений в задачах управления на конечном промежутке времени: дис. ... канд. физ.-матем. наук / Челябинский государственный университет. Челябинск, 2005. 160 с.
7. Зимовец А.А. Метод приграничного слоя для приближенного построения множеств достижимости управляемых систем // Вестник ЮУрГУ. Сер. Математика. Механика. Физика. 2013. Т. 5. № 1. С. 18–25.

Поступила в редакцию 16.10.2015

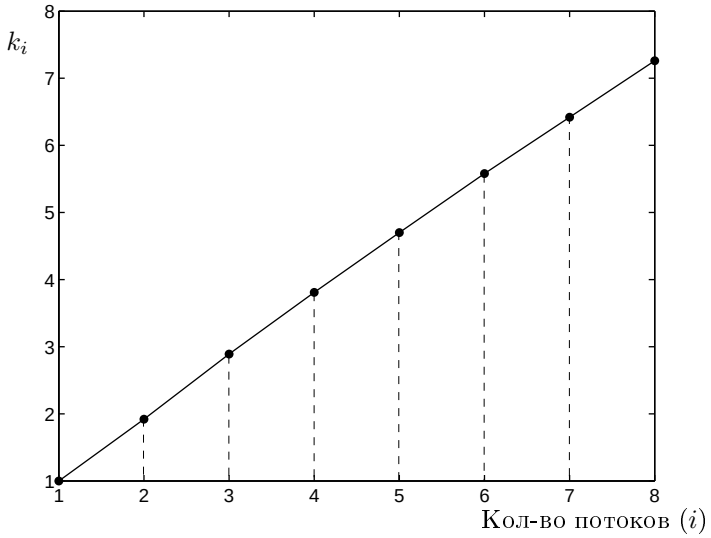

Рис. 3. Зависимость коэффициента ускорения от количества потоков

Зимовец Артём Анатольевич, к. ф.-м. н., ведущий математик, отдел динамических систем, Институт математики и механики им. Н. Н. Красовского УрО РАН, 620990, Россия, г. Екатеринбург, ул. С. Ковалевской, 16.
E-mail: aazimovets@gmail.com

Матвийчук Александр Ростиславович, к. ф.-м. н., заведующий отделом, отдел системного обеспечения, Институт математики и механики им. Н. Н. Красовского УрО РАН, 620990, Россия, г. Екатеринбург, ул. С. Ковалевской, 16.
E-mail: matv@uran.ru

A. A. Zimovets, A. R. Matviichuk
A parallel algorithm for constructing approximate attainable sets of nonlinear control systems

Keywords: attainability set, parallel algorithm, control system, grid-based method.

MSC: 93B40

The paper investigates the effectiveness of shared memory parallel programming approach for constructing approximate attainable sets of nonlinear control systems in a finite-dimensional Euclidean space. In this study, we propose a parallel iterative algorithm for constructing approximate attainable sets employing a regular Cartesian grid for spatial discretization. The proposed algorithm has been designed for implementation on SMP systems and handles such issues as data decomposition, threads synchronization and distribution of work between multiple threads. Numerical experiments on a system with two quad-core processors confirmed a high efficiency of shared memory parallel programming approach for applying grid-based methods to construct approximate attainable sets.

REFERENCES

1. Krasovskii N.N. *Teoriya upravleniya dvizheniem* (Theory of motion control), Moscow: Nauka, 1968, 476 p.
2. Nikol'skii M.S. A method of approximating an attainable set for a differential inclusion, *USSR Computational Mathematics and Mathematical Physics*, 1988, vol. 28, no. 4, pp. 192–194.
3. Komarov V.A., Pevchikh K.E. A method of approximating attainability sets for differential inclusions with a specified accuracy, *USSR Computational Mathematics and Mathematical Physics*, 1991, vol. 31, no. 1, pp. 109–112.
4. Guseinov Kh.G., Moiseyev A.N., Ushakov V.N. The approximation of reachable domains of control systems, *Journal of Applied Mathematics and Mechanics*, 1998, vol. 62, no. 2, pp. 169–175.

5. Ushakov V.N., Matviichuk A.R., Ushakov A.V. Approximations of attainability sets and of integral funnels of differential inclusions, *Vestn. Udmurt. Univ. Mat. Mekh. Komp'yut. Nauki*, 2011, no. 4, pp. 23–39 (in Russian).
6. Pakhotinskikh V.Yu. Constructing solutions for control problems on a finite time interval, *Cand. Sci. (Phys.-Math.) Dissertation*, Chelyabinsk, 2005, 160 p. (In Russian).
7. Zimovets A.A. A boundary layer method for the construction of approximate attainability sets of control systems, *Vestn. Yuzhno-Ural. Gos. Univ. Ser. Mat. Mekh. Fiz.*, 2013, vol. 5, no. 1, pp. 18–25 (in Russian).

Received 16.10.2015

Zimovets Artem Anatol'evich, Candidate of Physics and Mathematics, Leading Mathematician, Department of Dynamical Systems, N.N. Krasovskii Institute of Mathematics and Mechanics, Ural Branch of the Russian Academy of Sciences, ul. S. Kovalevskoi, 16, Yekaterinburg, 620990, Russia.
E-mail: aazimovets@gmail.com

Matviichuk Aleksandr Rostislavovich, Candidate of Physics and Mathematics, Head of Department, Department of Information Technologies, N.N. Krasovskii Institute of Mathematics and Mechanics, Ural Branch of the Russian Academy of Sciences, ul. S. Kovalevskoi, 16, Yekaterinburg, 620990, Russia.
E-mail: matv@uran.ru