

О. В. Фесько

Программный комплекс оптимизации динамических систем на множествах управлений

Аннотация. Представлены результаты разработок алгоритмической и программной части программного комплекса оптимизации динамических систем на множествах управлений простой структуры, в котором реализован параллельный алгоритм поиска кусочно-линейного управления нелинейной динамической системы, удобного для практического применения. Применение алгоритма и анализ эффективности распараллеливания на кластерном вычислительном устройстве демонстрируется на ряде примеров.

Ключевые слова и фразы: задача оптимального управления, кусочно-постоянное управление, кусочно-линейное управление, параллельный алгоритм, Т-система, OpenTS, язык программирования ++, графический интерфейс пользователя.

Введение

Рассматривается задача оптимального управления процессами, которые описываются системами дифференциальных уравнений. Подобные задачи имеют многочисленные приложения в механике космического полета, в вопросах управления химическими или ядерными реакторами и т.д.

При исследовании различных динамических систем с управлением большое значение имеет поиск простых оптимальных законов управления, реализуемых на практике. Предложен алгоритм поиска решения задачи оптимального управления на множестве кусочно-линейных функций посредством сведения исходной задачи к задаче условной минимизации функции многих переменных. Демонстрируется применение алгоритма и анализ эффективности распараллеливания в среде исполнения OpenTS [4].

1. Постановка задачи

Рассмотрим задачу оптимального управления для системы

$$(1) \quad \begin{cases} \dot{x}(t) = f(t, x(t), u(t)), & t \in [t_0, t_1], \\ x(t_0) = x_0, \end{cases}$$

где $x = (x_1, \dots, x_n)^T \in \mathbb{R}^n$, $x_i(t)$, $i = \overline{1, n}$ — кусочно-дифференцируемы, управление $u(t)$ — кусочно-линейно разрывно, т. е.

$$u(t) = \frac{(w_{2i+1} - w_{2i})t - (\tau_i w_{2i+1} - \tau_{i+1} w_{2i})}{\tau_{i+1} - \tau_i}, t \in [\tau_i, \tau_{i+1}], i = \overline{0, m},$$

где m — число точек переключения,

$$w_- \leq w_i \leq w_+, w_i \in \mathbb{R}, i = \overline{0, 2m+1}$$

$$t_0 = \tau_0 < \tau_1 < \tau_2 < \dots < \tau_{m+1} = t_1, m \geq 0, m \in \mathbb{Z},$$

с критерием качества

$$(2) \quad F(x(t_1)) \rightarrow \min.$$

В работах [1, 3] найдены условия, при которых функция F будет дифференцируема по параметрам w_i , $i = \overline{0, 2m+1}$, а исходную задачу (1), (2) можно свести к задаче условной минимизации функции многих переменных $F(x(t_1)) = F(x(t_1; w)) = G(w)$ на множестве W , т. е.

$$G(w) \rightarrow \min_{w \in W}, W = \{w \in \mathbb{R}^{2m+2} : w_- \leq w_i \leq w_+, i = \overline{0, 2m+1}\}.$$

2. Алгоритм решения задачи

Для минимизации многомерной функции $G(w)$ был выбран широко применяемый градиентный метод с некоторыми модификациями [1].

Схема решения поставленной задачи оптимального управления (1), (2) следующая (рис.1):

Шаг 1: Основная область поиска управлений W разбивается на непересекающиеся подобласти. В каждой из подобластей программа работает по нижеследующей схеме.

Шаг 2: Подается некоторый начальный вектор управлений, область поиска, точность вычислений и прочие входные данные.

Шаг 3: При данном управлении решается задача Коши для системы дифференциальных уравнений, ищется траектория, вычисляется значение целевого функционала.

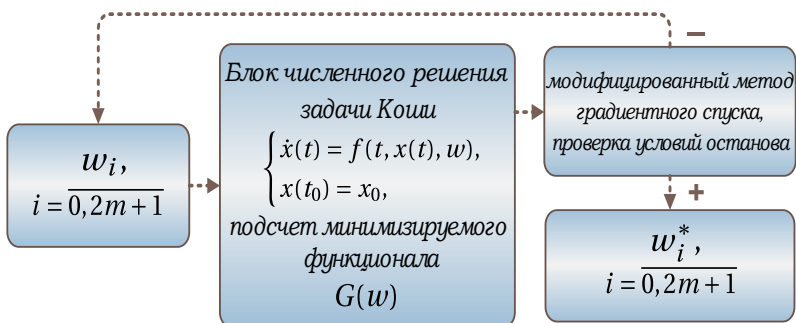


Рис. 1. Схема работы программы

Шаг 4: Вызывается функция поиска очередного приближения.

Шаг 5: Проверка условий останова. Если условия не выполняются, переходим к шагу 3, иначе алгоритм прекращает свою работу.

3. Тестовые примеры

Для программы решения задачи оптимального управления на множестве кусочно-линейных управлений было разработано несколько тестовых примеров.

Пример 1.

Имеется дифференциальная система с начальными условиями

$$(3) \quad \begin{cases} \dot{x}_1(t) = u(t), & t \in [0, 2], \\ \dot{x}_2(t) = u^2(t) + x_1^2(t), \\ x_1(0) = 1, & x_2(0) = 0 \end{cases}$$

с критерием качества

$$(4) \quad F(x(t_1)) = x_2(2) \rightarrow \min,$$

где

$$u(t) = \frac{(w_{2i+1} - w_{2i})t - (\tau_i w_{2i+1} - \tau_{i+1} w_{2i})}{\tau_{i+1} - \tau_i}, t \in [\tau_i, \tau_{i+1}], i = \overline{0, m}$$

с одной и двумя точками переключения ($m = 1, 2$). Моменты переключений: $\tau_1 = 1$ (при $m = 1$) и $\tau_1 = \frac{2}{3}$, $\tau_2 = \frac{4}{3}$ (при $m = 2$). На рис. 2(а) и 2(б) представлены графики управлений, найденных теоретически и программно (с одной и двумя точками переключения).

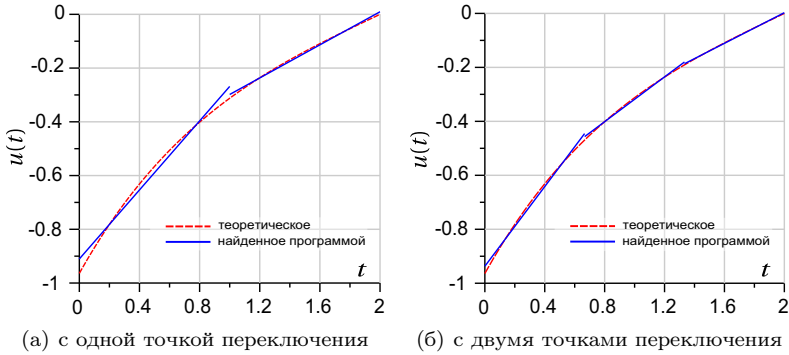


Рис. 2. Сравнение программно и теоретически найденных управлений

При этом полученное программно значение критерия при $m=1$ составило 0.964527, а при $m=2$ — 0.964133.

Пример 2.

Рассмотрим нелинейную динамическую систему с управлением, полученную в [5] при аппроксимации модели движения вертолета в вертикальной плоскости в нештатной ситуации:

$$\begin{cases} \dot{x}_1(t) = -0.441 \times 10^{-3} \sqrt{(x_1^2 + x_2^2)} x_1 - 9.8 u_1, & x(0) = (0, 0, 29.6, 0)^T, \quad t \in [0, t_F], \\ \dot{x}_2(t) = -0.441 \times 10^{-3} \sqrt{(x_1^2 + x_2^2)} x_2 + 0.122 x_3^2 F_T(x_1, x_2, x_3, u_1, u_2) - 9.792, \\ \dot{x}_3(t) = f_3(x_1, x_2, x_3, u_1, u_2, \tilde{N}) + 0.192 x_3^{-1} (N - 357), & \dot{x}_4(t) = x_2, \end{cases}$$

где $x_1(t)$, $x_2(t)$ — горизонтальная и вертикальная составляющие вектора скорости, $x_3(t)$ — угловая скорость вращения несущего винта, $x_4(t)$ — высота, $u_1(t)$ — угол отклонения вектора тяги от вертикали, $u_2(t)$ — общий шаг несущего винта, $F_T(t)$, f — композиционные полиномы, $\tilde{N}$, N — константы.

В [5] по критерию $F_0 = x_4(t_F) \rightarrow \min$ с применением специальных методов теории оптимального управления было найдено приближенно оптимальное решение $(\bar{x}(t), \bar{u}(t))$. Однако $\bar{u}(t)$ имеет весьма сложную структуру. Данное управление изображено на рис. 3(a) и 3(б) пунктиром.

Поставим модельную задачу поиска управления более простой структуры, а именно, кусочно-линейного управления с фиксированным набором моментов переключения: $\tau_1 = 2.5$, $\tau_2 = 8.5$ (для $u_1(t)$)

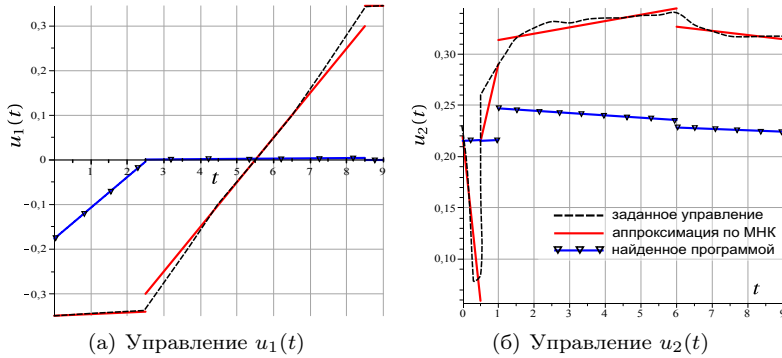


Рис. 3. Сравнение полученных управлений

и $\tau_3 = 0.5$, $\tau_4 = 1$, $\tau_5 = 6$ (для $u_2(t)$), которое реализует движение, близкое к $\bar{x}(t)$. Существенным будем считать фазовое ограничение вида $x_2(t) \leq 0$.

Одним из возможных способов построения такого управления видится аппроксимация по методу наименьших квадратов (МНК). Найденное управление по МНК представлено на рис. 3(а) и 3(б) красной линией, а соответствующие ему траектории — на рис. 4(а) и 4(б). Заметно, что условие $x_2(t) \leq 0$ нарушается уже в начале движения.

Воспользуемся теперь вышеописанной процедурой получения кусочно-линейного управления совместно со специальным образом составленным функционалом

$$F(x(t_F)) = \int_0^{t_F} \left(|\min(0; x_1(t))| + \beta(x_2(t) - \bar{x}_2(t))^2 \right) dt \rightarrow \min.$$

Полученные таким способом управления и соответствующие траектории изображены на рис. 3 и 4. Согласно данным графикам, при этом подходе существенное фазовое ограничение $x_2(t) \leq 0$ выполняется на всем рассматриваемом отрезке времени.

4. Параллельные версии программ на языке T++

Программа решения задачи оптимального управления была реализована на языке T++, т.к. вариант программ на C++ естественным образом ориентирован на параллельные вычисления. Система

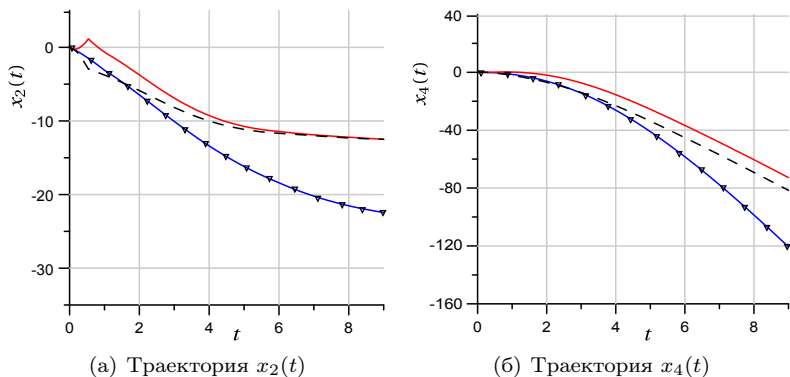


Рис. 4. Соответствующие траектории

параллельного программирования — Т-система [4, 6, 7] — позволяет без участия пользователя, в динамике выполнять распараллеливание кусков кода программы, планировку вычислений, распределение данных по узлам и пр. Т-система предоставляет среду исполнения для языка программирования высокого уровня Т++, который является параллельным диалектом языка С++. Данная система параллельного программирования — собственная разработка Института программных систем им. А.К. Айламазяна РАН.

Программная реализация параллельного алгоритма задачи поиска кусочно-линейных управлений была успешно апробирована на ряде методических и модельных (нештатная посадка вертолета) примеров [2].

Гранулой параллелизма в данных задачах был выбран непосредственно градиентный метод (независимый поиск минимума в подобластях). Вычисления проводились на суперкомпьютере семейства «СКИФ» [8] ИПС им. А.К. Айламазяна РАН.

Алгоритм содержит крупные гранулы параллелизма, вычисляемые независимо для каждого набора входных данных. Причем заранее неизвестно, сколько итераций потребуется для достижения указанной точности и неизвестно поэтому как будут загружены отдельные узлы вычислительной установки. Динамическое распараллеливание имеет преимущества в случаях, когда требуется выравнивание

ТАБЛИЦА 1. Анализ эффективности параллельной версии программы

Число узлов, p	Время, t_p	Ускорение, t_1/t_p
1	2182.976	1
2	1319.838	1.65
4	826.408	2.64
6	526.569	4.15
8	454.782	4.8
10	412.992	5.29
12	352.079	6.2

нагрузки в гетерогенных и/или меняющихся со временем вычислительных системах, а также для задач, обладающих гранулами различной тяжести.

В таблице 1 представлены результаты ускорения параллельного варианта программы на различном числе узлов для задачи (3), (4) с двумя точками переключения. При этом рассматривалось разбиение основной области на 32 подобласти. Время расчета заметно сокращается при увеличении числа узлов.

5. Графический интерфейс пользователя

Помимо прямого взаимодействия пользователя с высокопроизводительной вычислительной системой параллельной архитектуры, для организации удобного и интерактивного взаимодействия пользователя с кластерным вычислительным устройством (КВУ) был реализован простой пользовательский графический интерфейс (рис. 5).

Чтобы приступить к вычислениям, пользователю требуется авторизоваться на управляющей ЭВМ КВУ (рис. 6).

На рис. 7 представлены области:

- (1) Панель инструментов. Обеспечивает возможность загрузки заранее сформированных входных параметров задачи, сохранения параметров, организации запуска и останова вычислений.
- (2) Область задания входных параметров задачи.
 - $m \in \mathbb{Z}_+$. Число дифференциальных уравнений системы.
 - $x_0 \in \mathbb{R}^n$. Значения на левом конце траектории.

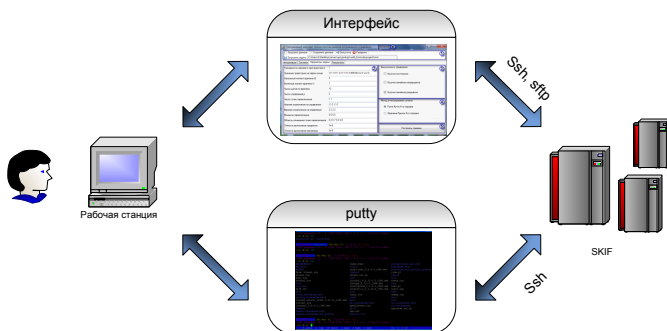


Рис. 5. Варианты взаимодействия пользователя с КВУ

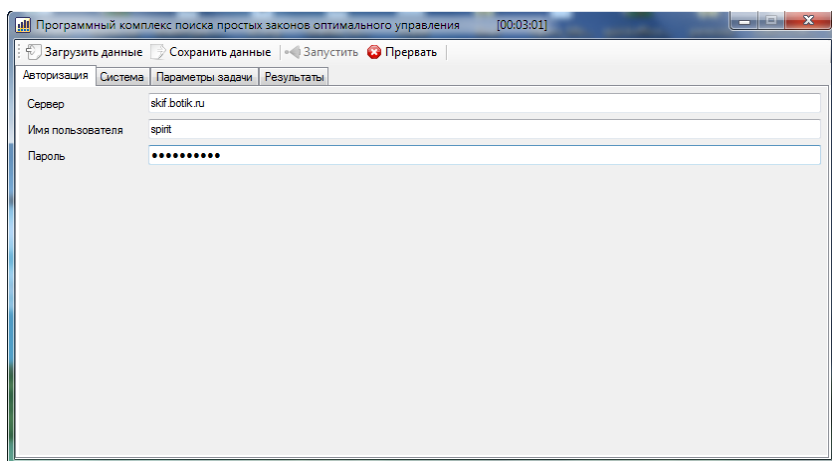


Рис. 6. Ввод логина и пароля

- $t_0, t_1 \in \mathbb{R}$. Начальный и конечный моменты времени ($t_1 > t_0$).
- $h \in \mathbb{Z}_+$. Число шагов по времени для интегрирования системы.
- $p \in \mathbb{Z}_+$. Число управлений системы.
- $c \in \mathbb{Z}_+^p$. Число точек переключения по каждому управлению.
- $low \in \mathbb{R}^{p+\sum_i c_i}$. Нижние ограничения на компоненты вектора u .

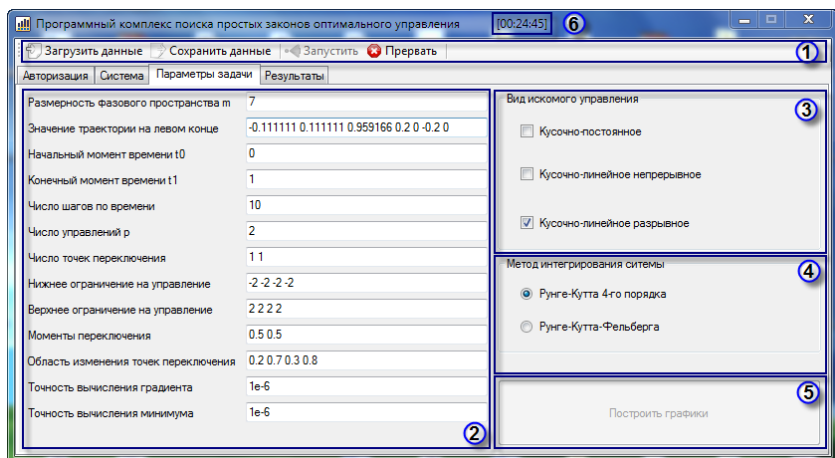


Рис. 7. Ввод параметров задачи

- $up \in \mathbb{R}^{p+\sum_i c_i}$. Верхние ограничения на компоненты вектора u , причем $up_j > low_j$.
- $\tau \in \mathbb{R}^{\sum_i c_i}$. Моменты переключений для каждого управления.
- $region \in \mathbb{R}^{2\sum_i c_i}$. Область изменения каждой точки переключения.
- $\epsilon, epsilon \in \mathbb{R}_+$. Точности вычислений минимума и градиента соответственно.

- (3) Область выбора желаемого типа поиска управления.
- (4) Область выбора предпочитаемого метода интегрирования системы.
- (5) Кнопка, запускающая визуализацию полученных данных в виде графиков.
- (6) Таймер.

В графическом интерфейсе пользователя предусмотрена возможность задания правых частей управляемой системы (область 1), а также терминального функционала качества (область 2) в виде синтаксически правильных математических выражений языка программирования C++ (рис. 8).

В случае успешного завершения вычислений программа оповещает пользователя (рис. 9).

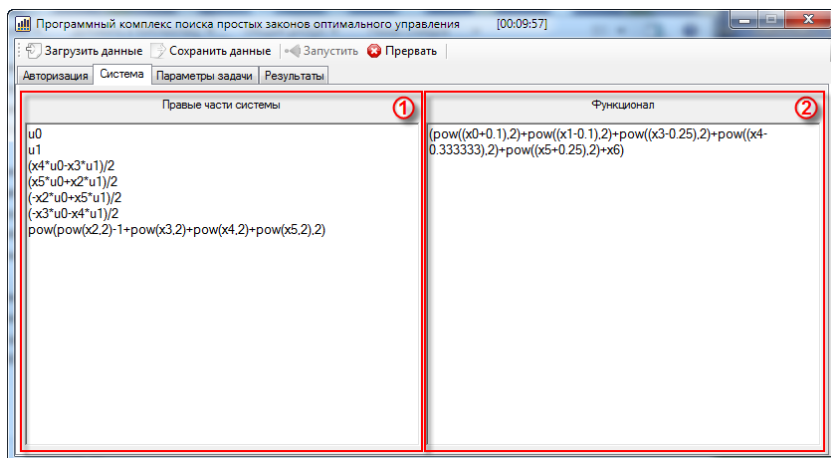


Рис. 8. Ввод системы ОДУ и функционала

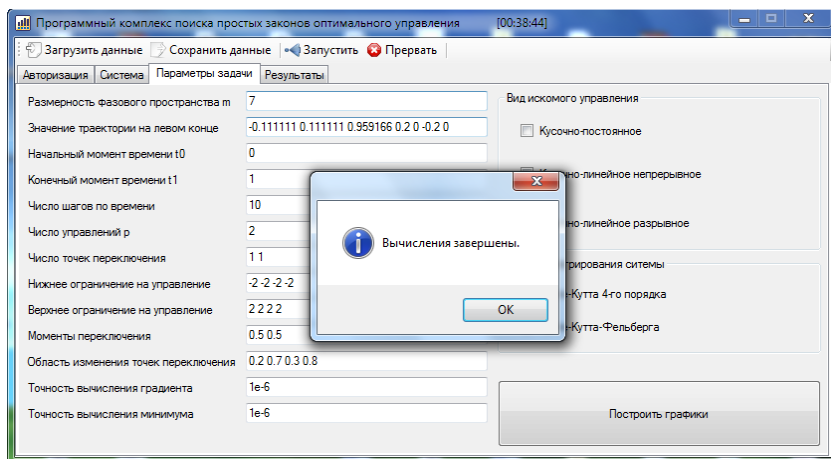


Рис. 9. Уведомление об успешном завершении вычислений

По окончании работы программы в текстовом поле вкладки «Результаты» перечисляются переданные с сервера названия выходных

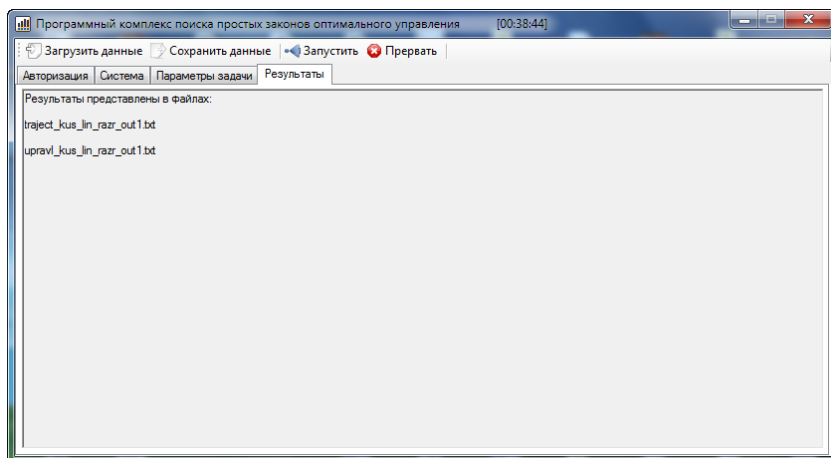


Рис. 10. Результирующие файлы

файлов, в которых представлены найденные управления и траектории (рис. 10). Их можно посмотреть в текущей рабочей директории пользователя.

После нажатия кнопки «Построить графики» появляются два окна: с графиками управлений и траекторий (рис. 11, 12). Имеется возможность сохранять графики в различные форматы. Более того, с помощью манипулятора типа мышь или тачпад на графике можно выделить область, требующую масштабирования (для более детального рассмотрения графика).

Заключение

Предложенный алгоритм поиска начального управления может быть непосредственно применен к задачам поиска управления для непрерывных динамических систем.

Метод естественным образом ориентирован на параллельную программно-алгоритмическую реализацию. Соответствующая параллельная программа (Т-программа) успешно применяется к исследованию

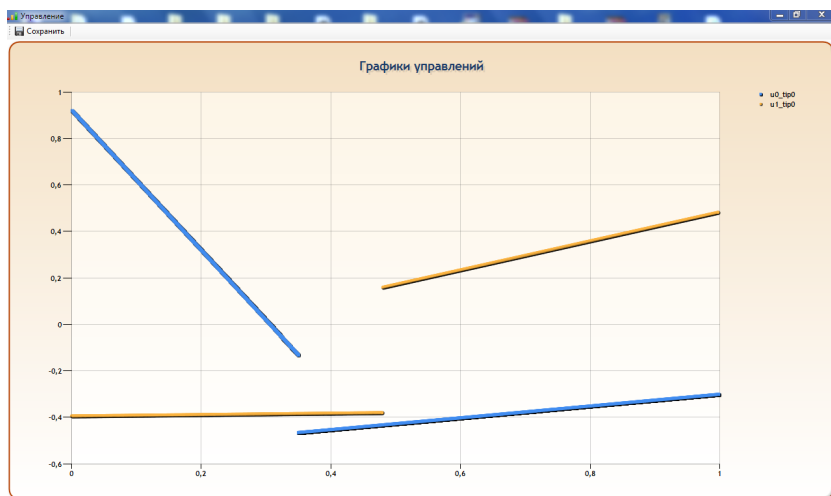


Рис. 11. Графики управлений

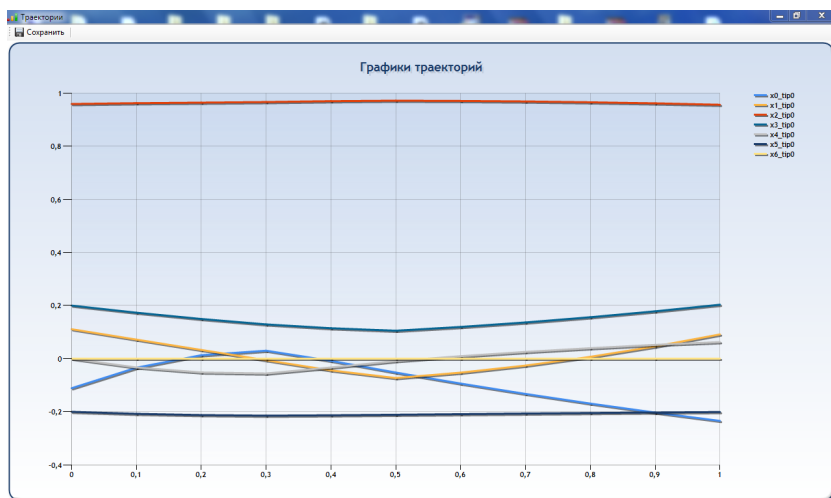


Рис. 12. Соответствующие траектории

тестовых и прикладных задач оптимального управления с целью поиска начального приближения для последующего уточнения известными методами улучшения. Показана хорошая эффективность распараллеливания указанного класса алгоритмов. Представлен графический интерфейс пользователя.

Список литературы

- [1] Фесько О. В. *Параллельный алгоритм оптимизации динамических систем на множестве кусочно-линейных управлений* // Вестник БГУ, 2010, № 9, с. 79–87. ↑1, 2
- [2] Фесько О. В. *Параллельный алгоритм оптимизации динамических систем с управлением* // Параллельные вычислительные технологии (ПаВТ2010): Труды международной научной конференции : Электронный ресурс. — Электронное издание. — Уфа : Челябинск: Издательский центр ЮУрГУ, 2010. — ISBN 978–5–696–03987–9, с. 689. ↑4
- [3] Фесько О. В. *Оптимизация динамических систем на множестве кусочно-постоянных управлений* // Наукоемкие информационные технологии : Труды молодежной научно-практической конференции. — Переславль-Залесский : Изд-во «Университет города Переславля», 2009, с. 206–217. ↑1
- [4] Абрамов С. М., Адамович А. И., Инюхин А. В., Московский А. А., Роганов В. А., Шевчук Ю. В., Шевчук Е. В. *T-система с открытой архитектурой* // Суперкомпьютерные системы и их применение SSA'2004 : Труды Международной научной конференции. — Минск, 2004, с. 18–22. ↑1, 4
- [5] Гурман В. И., Квоков В. Н., Ухин М. Ю. *Приближенные методы оптимизации управления летательным аппаратом* // Автоматика и телемеханика, 2008, № 4, с. 191–201. ↑3
- [6] Абрамов С. М., Кузнецов А. А., Роганов В. А. *Кроссплатформенная версия T-системы с открытой архитектурой* // Параллельные вычислительные технологии (ПаВТ'2007, Челябинск) : Труды международной научной конференции. — Челябинск : Изд-во ЮУрГУ, 2007, с. 115–121. ↑4
- [7] Абрамов С. М., Загоровский И. М., Коваленко М. Р., Матвеев Г. А., Роганов В. А. *Миграция от MPI к платформе OpenTS: эксперимент с приложениями PovRay и ALCMD* // Программные системы: теория и приложения (Переславль-Залесский, 2006) : Международная конференция. — М. : Наука-Физматлит, 2006, с. 265–275. ↑4
- [8] Абрамов С. М., Адамович А. И., Коваленко М. Р., Слепухин А. Ф., Парамонов Н. Н. *Кластерные системы семейства суперкомпьютеров «СКИФ»* // Научный сервис в сети Интернет (Новороссийск, 2003) : Труды Всероссийской научной конференции. — М. : Изд-во МГУ, 2003, с. 147–151. ↑4

O. V. Fesko. *Software system of dynamical systems optimization in control set.*

ABSTRACT. The developing results of algorithmic and programmed part of software system intended for dynamical systems optimization in the prime control set are introduced. This part implements parallel algorithm of the piecewise linear control (that is handy) searching for nonlinear dynamical system. The algorithm application and efficiency analysis of code parallelization in a computer cluster is demonstrated by a number of examples.

Key Words and Phrases: optimal control problem, piecewise constant control, piecewise linear control, parallel algorithm, T-system, OpenTS, T++ parallel programming language, graphical user interface.

Образец ссылки на статью:

О. В. Фесько. Программный комплекс оптимизации динамических систем на множествах управлений // Программные системы: теория и приложения : электрон. научн. журн. 2010. № 4(4), с. 53–66. URL: http://psta.psiras.ru/read/psta2010_4_53-66.pdf