

А. О. Блинов, В. И. Гурман, Е. А. Трушкова, В. П. Фраленко

Программный комплекс улучшения и оптимизации законов управления

Аннотация. Разработан программный комплекс улучшения и оптимизации управления для приложений в различных областях. В программном комплексе успешно реализованы на кластерном вычислительном устройстве алгоритмы аппроксимации, оптимизации и улучшения приближенно оптимального управления для различных динамических процессов с управлением. Исследование возможностей и эффективности комплекса проведено на ряде модельных и прикладных задач, включая многокомпонентную модель региона, модель движения вертолета в нештатной ситуации, модель спуска космического челнока в атмосфере с минимальным теплоподводом.

1. Введение

Работа посвящена разработке и реализации методов синтеза оптимальных целевых законов управления, что является кардинальной проблемой теории и практики управления. Ее решение связано с бесконечномерными обратными задачами, аппроксимируемыми при численной реализации многомерными задачами, требующими неограниченно растущих вычислительных ресурсов для приближения к точным решениям. Это приводит к неизбежному выводу о необходимости нового подхода — априорно приближенного, позволяющего связать точность приближения с размерностью аппроксимаций и, соответственно, с мощностью вычислительных средств, и реализовать такой подход на современных высокопроизводительных вычислительных системах параллельной архитектуры. Парадигма параллельных вычислений в высшей степени соответствует самой природе рассматриваемых задач, связанных с множественностью однотипных операций на верхнем уровне, таких как:

- решение обратной задачи через множество решений прямой (прямо или косвенно);
- формирование и численная реализация полей управлений (ситуационных управлений).

В статье отражены результаты разработки, реализации и исследования экспериментального варианта комплекса улучшения и оптимизации законов управления для приложений в различных областях ПК ISCON (Improvement and Synthesis of Control) с распараллеливанием и переносом на кластерное вычислительное устройство (КВУ).

В следующем разделе 2 описываются главные компоненты комплекса: графический интерфейс, сервер управления, управляющие модули и набор исполняемых модулей. Разделы 3 и 4 содержат постановки задач, на которых проводятся вычислительные эксперименты в рамках тестирования ПК и основные рекомендации пользователям по работе в ПК.

2. Описание программного комплекса

Программный комплекс предназначен для моделирования сложных динамических процессов, а также решения оптимизационных задач и задач улучшения управления для различных прикладных областей на КВУ. Для этого в нем реализованы алгоритмы аппроксимации, оптимизации и улучшения приближенно оптимального управления. Главными компонентами комплекса являются графический интерфейс, сервер управления, управляющие модули и набор исполняемых модулей.

В графическом интерфейсе происходит ввод начальных данных, постановка задачи, выбор метода решения задачи, управление потоками данных, визуализация и сохранение результатов. Сервер управления участвует в обеспечении пользователям доступа к возможностям комплекса, принимает запросы на решение выбранных задач с выбранными пользователем настройками. Управляющие модули принимают полученную от сервера управления информацию и выполняют развертывание полигона для вычислений, запуская в дальнейшем либо локально, либо удаленно исполняемый модуль решаемой задачи. Так же они обеспечивают сбор выходных данных и их передачу обратно серверу управления или пользователю.

Основной идеей при разработке архитектуры системы было обеспечение гибкости и расширяемости. Выбранная модульная схема программного комплекса обеспечивает расширяемость и масштабируемость. Гетерогенность вычислительной среды поддержана включением в архитектуру ПК управляющих модулей, связывающих физически разделенные компоненты ПК. В зависимости от набора исполняемых модулей, ПК может использоваться при создании систем,

относящихся к различным задачам оптимизации и управления. Гибкость системы обеспечивается возможностью подключения доступных вычислительных устройств за счет конфигурирования управляющих модулей.

Из особенностей ПК можно выделить:

- Использование параллелизма на различных уровнях: параллельное выполнение решаемых задач, внутренний параллелизм модулей.

- Модули системы реализуются в виде исполняемых файлов и могут содержать как последовательную, так и параллельную реализацию алгоритма.

Управление ПК поддерживается графическим интерфейсом, интегрированным с сервером управления.

Область применения программного комплекса определяется реализованными методами (исполняемыми модулями), предназначенными для построения аналитического описания модели по имеющимся статистическим данным (алгоритм аппроксимации по методу наименьших квадратов (МНК)), улучшения приближенно оптимальной программы управления динамической системой, моделирования и исследования упругих систем. Указанные методы могут быть применены, например, к задачам управления экономическими системами, движением летательных аппаратов (вертолеты, космические челноки).

3. Описание задач, решаемых с помощью ПК

3.1. Программа аппроксимации моделей динамических систем

При работе с моделями реальных динамических систем

$$\dot{x}(t) = f(t, x(t), u(t)), \quad x \in \mathbb{R}^n, \quad u \in \mathbb{R}^p,$$

типичны ситуации, когда модель имеет структуру, к которой невозможно применить тот или иной метод исследования или алгоритм синтеза управления. В этих случаях, для того, чтобы упростить систему, предлагается применять алгоритм аппроксимации многомерных функций многих переменных $f(t, x, u)$ (при фиксированных моментах времени) по методу наименьших квадратов многомерными

полиномами [1]

$$\varphi(z) = \sum_{j=1}^m \psi_j g_j(z), \quad z = (z^1, \dots, z^{n+p}) = (x, u) \in \mathbb{R}^{n+p},$$

где $\{g_j(z) = \prod_{i=1}^{n+p} (z^i)^{k_i(j)}\}$ — некоторый набор заданных базисных функций, $k_i(j)$ — целые положительные числа, а $\{\psi_j\}$ — соответствующий набор коэффициентов, подлежащих определению. Решается следующая задача минимизации (МНК):

$$S = \sum_{i=1}^{\beta} (\varphi(z_i) - f(t, x_i, u_i))^2 \rightarrow \min_{\{\psi_i\}}$$

где β — количество узлов аппроксимации. Эта задача сводится к решению системы линейных алгебраических уравнений с постоянными коэффициентами.

Для этого надо:

- 1) сформировать с помощью узлов аппроксимации и базисных функций приближающего полинома матрицу и столбец свободных членов для системы линейных алгебраических уравнений,
- 2) решить полученную систему.

Реализована параллельность указанного метода в первой части. Область формирования исходных данных разбивается на части, в каждой из которых строится матрица и столбец свободных членов для системы уравнений с помощью исходных значений в узлах текущей части. Общая матрица получается в этом случае как сумма всех построенных «частичных» матриц (это же справедливо и для свободных членов).

Алгоритм реализован в Т-системе (на языке программирования Т++) на КВУ семейства «СКИФ» [1].

3.1.1. Входные данные задачи

Входными данными задачи являются:

1. размерность фазового пространства n (целое число большее нуля);
2. размерность пространства управлений p (целое число большее нуля);
3. правая часть управляемой системы $f(t, x, u)$ — n -мерная функция от $(n + p)$ переменных (при фиксированном значении t);

4. нижние ограничения на фазовую траекторию и управление $((n + p)$ -мерный действительный вектор);

5. верхние ограничения на фазовую траекторию и управление $((n + p)$ -мерный действительный вектор);

6. базисные функции (j векторов размера $(n + p)$, содержащих степени вхождения переменных z^i в базисную функцию g_i , т. е. вектора вида $(k_1(j), k_2(j), \dots, k_{n+p}(j))$).

3.1.2. Выходные данные задачи

Выходными данными задачи являются коэффициенты аппроксимирующих полиномов (n действительных векторов размера j).

3.1.3. Настройка ПК по решению описанной задачи

Настройка ПК по решению описанной задачи произведена для правой части динамической системы вида:

$$(1) \quad f(x^1, x^2, x^3, x^4, u^1, u^2) = (f_1(x, u), f_2(x, u), f_3(x, u), f_4(x, u))^T,$$

где

(2)

$$\begin{aligned} f_1(x, u) &= x^1 - 0.000004413602941\sqrt{(x^1)^2 + (x^2)^2}x^1 - 0.098u^1, \\ f_2(x, u) &= x^2 - 0.000004413602941\sqrt{(x^1)^2 + (x^2)^2}x^2 + \\ &+ 0.001217794118(x^3)^2 \left(-0.04504670322 + 0.002934957373x^3 - \right. \\ &- 0.00005375103213(x^3)^2 + x^1(0.005307639750 - 0.000291577926x^3 + \\ &+ 0.00000474579887(x^3)^2) + x^2(0.00586815863 - 0.000587450425x^3 + \\ &+ 0.00001158950057(x^3)^2) + u^1(0.07527467316 - 0.003228550181x^3 + \\ &+ 0.00004177445652(x^3)^2) + u^2(0.4564552109 - 0.007347112224x^3 + \\ &+ 0.0001468137088(x^3)^2) \left. \right) - 0.09799999999, \\ f_3(x, u) &= 1.001251011x^3 + 0.02526105823 - 0.00005189698900(x^3)^2 + \\ &+ 0.01x^1(0.5281399869 - 0.03512379548x^3 + 0.0006299973572(x^3)^2) + \\ &+ 0.01x^2(-1.228470885 + 0.07975873768x^3 - 0.001487652282(x^3)^2) + \\ &+ 0.01u^1(8.462949172 - 0.5230654017x^3 + 0.009933002308(x^3)^2) + \\ &+ 0.01u^2(75.61380577 - 5.458013542x^3 + 0.08086676638(x^3)^2) - \\ &- \frac{0.071153846151}{x^3}, \\ f_4(x, u) &= x^4 + 0.01x^2. \end{aligned}$$

Для проведения аппроксимации указанной функции $f(x, u)$, например, линейной функцией

$$\varphi(x, u) = \psi_1 + \psi_2x^1 + \psi_3x^2 + \psi_4x^3 + \psi_5x^4 + \psi_6u^1 + \psi_7u^2$$

в области изменения переменных

$$0.278 \leq x^1 \leq -3.2, \quad -3.2 \leq x^2 \leq 0, \quad 24.6 \leq x^3 \leq 30.8, \quad -6 \leq x^4 \leq 0, \\ -0.349 \leq u^1 \leq 0.349, \quad -0.349 \leq u^2 \leq 0.349$$

входные данные следует задать в виде:

1. размерность фазового пространства $n = 4$;
2. размерность пространства управлений $p = 2$;
3. правая часть управляемой системы $f(x, u)$ (см. выше);
4. нижние ограничения на фазовую траекторию и управление (6-мерный действительный вектор): (0.278, -3.2, 24.6, -6, -0.349, 0.07853);
5. верхние ограничения на фазовую траекторию и управление (6-мерный действительный вектор): (7.5, 0, 30.8, 0, 0.349, 0.349);
6. базисные функции (7 векторов размерности 6, содержащих степени вхождения переменных в текущую базисную функцию), т. е. вектора:

$$(0, 0, 0, 0, 0, 0), \\ (1, 0, 0, 0, 0, 0), \\ (0, 1, 0, 0, 0, 0), \\ (0, 0, 1, 0, 0, 0), \\ (0, 0, 0, 1, 0, 0), \\ (0, 0, 0, 0, 1, 0), \\ (0, 0, 0, 0, 0, 1).$$

Для оценки эффективности распараллеливания программы проведён запуск ПК на различном числе узлов и замер времени работы в каждом случае. Полученные данные представлены в таблице 1.

Таблица 1. Анализ эффективности параллельной версии программы улучшения управления.

Число узлов: n	Время работы программы: t_n	Ускорение: t_1/t_n	Число узлов: n	Время работы программы: t_n	Ускорение: t_1/t_n
1	3338.348	1	9	631.214	5.289
2	1779.791	1.876	10	596.175	5.600
3	1237.142	2.698	11	588.017	5.677
4	880.248	3.793	12	596.195	5.599
5	729.924	4.574	13	589.926	5.659
6	631.720	5.285	14	586.519	5.692
7	632.003	5.282	15	597.649	5.586
8	586.202	5.695	16	579.739	5.758

Эти данные позволяют сделать вывод об эффективном распараллеливании указанного класса алгоритмов лишь для небольшого числа узлов (1–8).

3.2. Программа улучшения управления

Предполагается, что модель движения в общем случае представляет собой дискретную управляемую систему, терминальный функционал качества, ограничения на управления типа неравенств, фазовые ограничения типа неравенств (различные внутри и на правом конце заданного фиксированного промежутка времени):

$$\begin{aligned} x(t+1) &= f(t, x(t), u(t)), \quad t \in T = \{t_I, t_I + 1, \dots, t_F\}, \\ x(t_I) &= x_I, \\ u(t) &\in D_u = \{u : T \setminus \{t_F\} \rightarrow \mathbb{R}^p \mid u_l \leq u(t) \leq u_u, t \in T \setminus \{t_F\}\}, \\ x_l &\leq x(t) \leq x_u, t \in T \setminus \{t_F\}, \quad x_{lF} \leq x(t_F) \leq x_{uF}, \\ I &= F_0(x(t_F)) \rightarrow \min, \quad u_l, u_u \in \mathbb{R}^p, x_I, x_l, x_u, x_{lF}, x_{uF} \in \mathbb{R}^n. \end{aligned}$$

Производится замена этой задачи општрафованной, т. е. задачей без фазовых ограничений, с помощью введенных функций штрафа типа срезок:

$$\begin{aligned} x(t+1) &= f(t, x(t), u(t)), \quad t \in T, \quad x(t_I) = x_I, \\ z(t+1) &= z(t) + t_F^{-1} \delta(x(t)), \quad z(t_I) = 0, \\ F(x(t_F), z(t_F)) &= \beta_0 F_0(x(t_F)) + \beta_1^T z(t_F) + \beta_2^T \delta_F(x(t)) \rightarrow \min, \end{aligned}$$

где

$$\begin{aligned} \beta_0 &\in \mathbb{R}, \beta_1, \beta_2 \in \mathbb{R}^n, \quad \delta^i(x) = -\min\{0, x^i - x_l^i\} + \max\{0, x^i - x_u^i\}, \\ \delta_F^i(x) &= -\min\{0, x^i - x_{lF}^i\} + \max\{0, x^i - x_{uF}^i\}, \quad i = 1, \dots, n. \end{aligned}$$

Задача улучшения ставится следующим образом: пусть известен допустимый элемент $m^I = (u^I(t), x^I(t))$, требуется найти допустимый элемент $m^{II} = (u^{II}(t), x^{II}(t))$, такой, что $F(x^{II}(t_F), z^{II}(t_F)) < F(x^I(t_F), z^I(t_F))$.

Итерационное улучшение основано на линейно-квадратических аппроксимациях соотношений Беллмана в среднем в окрестности текущего приближения полиномами второго порядка. Предусмотрены регуляторы, настраиваемые так, чтобы каждая итерация была наиболее эффективной.

Общие конструкции метода улучшения управления приведены в [2, 3], где на основе принципа оптимальности Кротова элемент m^{II}

ищется путем аппроксимации решения следующей задачи:

$$\begin{aligned} y(t+1) &= g(t, y(t), v(t)), \quad t \in T, \quad y(t_I) = 0, \\ s(t+1) &= s(t) + t_F^{-1} \delta(y(t) + x^I(t)) - t_F^{-1} \delta(x^I(t)), \quad s(t_I) = 0, \\ y^0(t+1) &= y^0(t) + \frac{1}{2} v^T(t) v(t), \quad y^0(t_I) = 0, \\ G_\alpha(y^0(t_F), y(t_F), s(t_F)) &= \alpha y^0(t_F) + \\ &+ (1 - \alpha) F(y(t_F) + x^I(t_F), s(t_F) + z^I(t_F)) \rightarrow \min, \end{aligned}$$

где α — некоторое число из отрезка $[0, 1]$ (регулятор метода), $y = x - x^I$, $s = z - z^I$, $v = u - u^I$, $g(t, y, v) = f(t, y + x^I, v + u^I) - f(t, x^I, u^I)$.

Будем искать функцию Кротова в виде

$$\varphi(t, y^0, y, s) = w(t) + \psi^0(t) y^0 + \psi^T(t) y + \gamma^T(t) s,$$

где значения $w(t)$, ψ^0 , ψ , γ находятся из следующих приближенных соотношений Кротова-Беллмана:

$$\varphi(t_F, y^0, y, s) \approx -G_\alpha(y^0, y, s),$$

$$\varphi(t, y^0, y, s) \approx \max_v \varphi((t+1, y^0 + \frac{1}{2} v^T v, g(t, y, v),$$

$$s + t_F^{-1} \delta(y + x^I) - t_F^{-1} \delta(x^I)), \quad t = t_F - 1, \dots, t_I.$$

При этом управление (в форме синтеза), на котором достигается максимум, обозначим через $\hat{v}(t, y)$.

Опишем одну итерацию алгоритма метода улучшения. По данному начальному приближению $m^I = (u^I(t), x^I(t))$ выбираем весовые коэффициенты функционала $F(x(t_F), z(t_F))$ из следующих условий:

$$\begin{aligned} \beta_0 = 1, \beta = (\beta_1^1, \dots, \beta_1^n, \beta_2^1, \dots, \beta_2^n)^T &= 0, \quad \text{если } J_0 = \{1, \dots, 2n\}, \\ \beta_0 = 0, \beta^j = 0, j \in J_0, \beta^j = \frac{P}{S h^j}, j \in J, & \quad \text{если } J_0 \neq \{1, \dots, 2n\}, \end{aligned}$$

где

$$\begin{aligned} h &= (z^T(t_F), \delta_F^T(t_F))^T, \quad J_0 = \{j \in \{1, \dots, 2n\} | h^j \leq 0.001\}, \\ J &= \{1, \dots, 2n\} \setminus J_0, \quad P = \prod_{j \in J} h^j, \quad S = \sum_{j \in J} \frac{P}{h^j}. \end{aligned}$$

Фиксируем набор параметров метода. Разложив правые части соотношений Кротова-Беллмана при фиксированном $t \in T$ в ряд до членов второго порядка в окрестности нуля и заменив производные их разностными аналогами (шаги разностных схем — дополнительные параметры метода), получим управление

$$u^{II}(t, x) = \hat{v}(t, x - x^I(t)) + u^I(t), \quad t \in T \setminus \{t_f\},$$

и соответствующий элемент

$$m^{II} = (u^{II}(t) = u^{II}(t, x^{II}(t)), x^{II}(t)).$$

Если улучшение произошло, проводим следующую итерацию, выбрав в качестве начального элемента m^{II} . В противном случае берем другой набор параметров метода или останавливаем итерации.

Построенный алгоритм естественным образом ориентирован на параллельные вычисления. Алгоритм реализован в Т-системе (язык программирования T++) на КВУ семейства «СКИФ» [4].

3.2.1. Входные данные задачи

Входными данными задачи являются:

1. размерность фазового пространства n (целое число большее нуля);
2. размерность пространства управлений p (целое число большее нуля);
3. начальный момент времени t_I (любое действительное число);
4. конечный момент времени t_F (действительное число большее t_I);
5. число отрезков разбиения временного интервала m (целое число большее нуля);
6. правая часть управляемой системы $f(t, x, u)$ — n -мерная функция от $(n + p + 1)$ переменной;
7. минимизируемый функционал $F_0(t_F, x(t_F))$ — функционал от n переменных;
8. начальное значение фазовой траектории x_I (n -мерный действительный вектор);
9. нижние ограничения на фазовую траекторию внутри временного отрезка (n -мерный действительный вектор), вектор-индикатор наличия/отсутствия этих ограничений (n -мерный вектор, состоящий из нулей и единиц);
10. верхние ограничения на фазовую траекторию внутри временного отрезка (n -мерный действительный вектор, каждая компонента которого больше соответствующей компоненты вектора из 9.), вектор-индикатор наличия/отсутствия этих ограничений (n -мерный вектор, состоящий из нулей и единиц);
11. нижние ограничения на фазовую траекторию в момент времени t_F (n -мерный действительный вектор), вектор-индикатор наличия/отсутствия этих ограничений (n -мерный вектор, состоящий из нулей и единиц);

12. верхние ограничения на фазовую траекторию в момент времени t_F (n -мерный действительный вектор, каждая компонента которого больше соответствующей компоненты вектора из 11.), вектор-индикатор наличия/отсутствия этих ограничений (n -мерный вектор, состоящий из нулей и единиц);

13. нижние ограничения на управление (p -мерный действительный вектор);

14. верхние ограничения на управление (p -мерный действительный вектор, каждая компонента которого больше соответствующей компоненты из 13.);

15. минимальное время переключения управления (p -мерный действительный вектор, каждая компонента которого положительна);

16. начальная программа управления ((pm) -мерный действительный вектор или имя текстового файла).

При этом данные 1–8 являются обязательными.

3.2.2. Выходные данные задачи

Вывод результатов работы программы осуществляется в текстовый файл. В нем указывается набор параметров; номер итерации, на которой достигнуто наибольшее улучшение; далее таблицей идут столбцы результатов (число строк равно числу временных моментов): временной момент, значения траектории в этот момент (покоординатно), значения управлений в этот момент (покоординатно), значения отклонений от допустимого множества (покоординатно); в заключение приводится достигнутое значение целевой функции. Предусмотрена также возможность вывода управления в форме синтеза ($u(t, x) = A(t)x + B(t)$): вывод двух текстовых файлов, один из которых содержит матрицу коэффициентов при переменной x (матрицу $A(t)$), другой — матрицу коэффициентов свободных членов (матрицу $B(t)$).

Формат файла выходных результатов позволяет производить построение графиков для наглядного представления произошедшего улучшения.

3.2.3. Настройка ПК по решению описанной задачи

Настройка ПК по решению описанной задачи произведена, в частности, для задачи улучшения начального приближенно оптимального

управления для нелинейной системы, полученной при аппроксимации модели движения вертолета в нештатной ситуации [3]:

$$x(t+1) = f(x(t), u(t)), \quad t \in \{0, 1, \dots, t_F\}, \quad x \in \mathbb{R}^4, u \in \mathbb{R}^2,$$

где правая часть системы $f(x, u)$ определена согласно формулам 1 и 2.

Заданы начальные значения фазовых переменных, ограничения на фазовые переменные во время и в конце маневра, ограничения на управления:

$$\begin{aligned} x(0) &= (10, 0, 29.6, 0)^T, \\ u_l &= (-0.348, 0.08)^T, \quad u_u = (0.348, 0.348)^T, \\ x_l &= (0, -5, 24.6, -\infty)^T, \quad x_u = (+\infty, 0, 30.8, +\infty)^T, \\ x_{lF} &= (0, -3.2, 24.6, -\infty)^T, \quad x_{uF} = (7.5, 0, 30.8, +\infty)^T. \end{aligned}$$

Требуется минимизировать конечную высоту $F_0(x(t_F)) = x^4(t_F)$, что равносильно максимизации нижней границы опасной зоны аварийной посадки.

Для проведения улучшения одного из вариантов начальной программы управления входные данные следует задать, например, в виде:

1. размерность фазового пространства $n = 4$;
2. размерность пространства управлений $p = 2$;
3. начальный момент времени $t_I = 0$;
4. конечный момент времени $t_F = 9.47$;
5. число отрезков разбиения временного интервала $m = 947$;
6. правая часть управляемой системы $f(x, u)$ (см. выше);
7. минимизируемый функционал $F_0(x(t_F))$ (см. выше);
8. начальное значение фазовой траектории $x_I = (10, 0, 29.6, 0)$;
9. нижние ограничения на фазовую траекторию внутри временного отрезка $(0, -5, 24.6, 0)$, вектор-индикатор ограничений $(1, 1, 1, 0)$;
10. верхние ограничения на фазовую траекторию внутри временного отрезка $(7.5, 0, 30.8, 0)$, вектор-индикатор ограничений $(0, 1, 1, 0)$;
11. нижние ограничения на фазовую траекторию в момент времени t_F $(0, -3.2, 24.6, 0)$, вектор-индикатор ограничений $(1, 1, 1, 0)$;
12. верхние ограничения на фазовую траекторию в момент времени t_F $(7.5, 0, 30.8, 0)$, вектор-индикатор ограничений $(1, 1, 1, 0)$;
13. нижние ограничения на управление $(-0.348, 0.08)$;
14. верхние ограничения на управление $(0.348, 0.348)$;

15. минимальное время перекладки управления (0.7, 0.35);

16. начальная программа управления берется из файла с именем `upr_nach.txt`.

Описанная программа улучшения управления для дискретных динамических систем была успешно применена к исследованию имитационной модели маневров безопасной нештатной посадки вертолета с определением границы безопасной зоны. Результаты позволили сделать вывод о повышении границы опасной зоны на 15% против начального приближения при сохранении качественного характера динамики управлений и состояния [3]. Результаты работы программного комплекса для входных числовых данных, описанных выше, представлены на рис. 1. Вычисления проводились для 256 различных наборов параметров метода улучшения. В результате работы программы удалось уменьшить значение целевого функционала, удовлетворив при этом всем ограничениям (см. рис. 1).

Для оценки эффективности распараллеливания программы проведен запуск ПК на различном числе узлов и замер времени работы в каждом случае. Полученные данные представлены в таблице 2.

Таблица 2. Анализ эффективности параллельной версии программы улучшения управления.

Число узлов: n	Время работы программы: t_n	Ускорение: t_1/t_n
1	1029.85	1
3	351.99	2.93
5	218.83	4.71
7	159.60	6.45
9	130.71	7.88
11	110.29	9.34
13	93.69	10.99
15	90.10	11.43

Эти данные позволяют сделать вывод об эффективном распараллеливании указанного класса алгоритмов.

4. Работа с программным комплексом

Графический интерфейс запускается из командной строки (файл `gui.exe`) или с помощью манипулятора типа «мышь». После этого пользователю становится доступна его базовая форма. Созданную задачу можно сразу же запустить на КВУ.

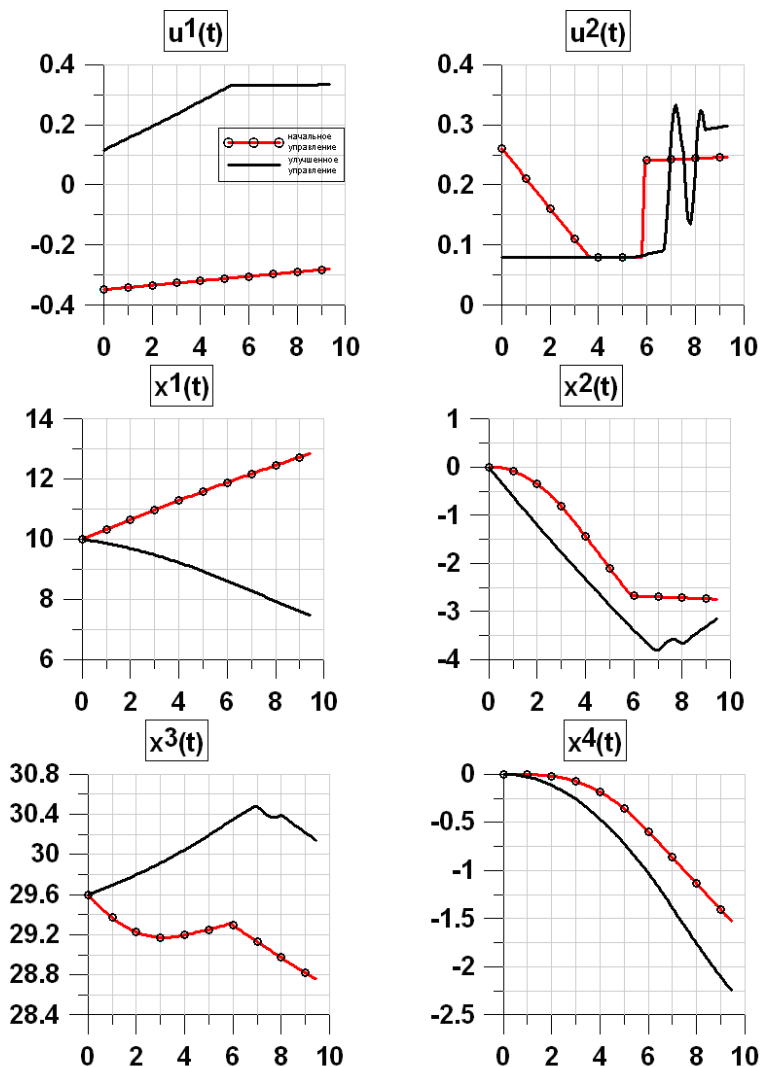


Рис. 1. Начальные и улучшенные значения управлений и соответствующих траекторий.

Входные данные представлены на вкладке «Параметры» области D и списком внешних файлов в форме «Отправить файлы» в области E (см. рис. 2).

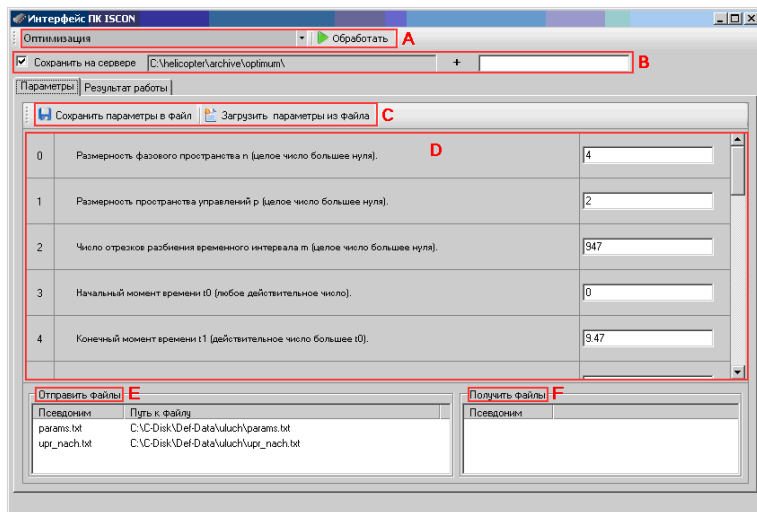


Рис. 2. Основная форма графического интерфейса.

На рис. 2 выделены шесть областей:

1. область A, в которой находится инструмент выбора решаемой задачи, кнопка запуска ее обработки;
2. область B с необходимыми настройками — включение/выключение архивации результатов обработки созданных задач на сервере с указанием поддиректории;
3. область C с кнопками сохранения параметров решаемой задачи в файл и загрузки из файла;
4. область D, содержащая необходимые параметры решаемой задачи с их расшифровкой;
5. область E с управляемой группой файлов, необходимых для решения задачи (загружаются с компьютера пользователя графического интерфейса на управляющий модуль);

6. *область F* с управляемой группой файлов, получаемых после решения задачи (загружаются на компьютер пользователя графического интерфейса от управляющего модуля).

При использовании графического интерфейса параметризация запуска вычислительных модулей решается автоматически согласно выбранным настройкам в *области D* и загружаемым файлам из *области E*. Другие настройки задаются при конфигурировании сервера управления и при создании управляющего модуля.

В случае успешного завершения вычислений появится окно с текстом «Работа над задачей успешно завершена». Если же во время работы счет задачи был остановлен пользователем, появится первое сообщение об остановке схемы обработки. Второе сообщение говорит о принудительном разрыве связи с сервером управления. Оно может появиться и в случае физического обрыва канала связи. Для контроля за активностью сервера управления можно обратиться к log-файлу сервера управления `runtime.log`.

По завершении работы исполняемого модуля, графический интерфейс самостоятельно загрузит с сервера управления файлы из *области F* при условии их физического существования и ненулевого размера (сохранение происходит в папку `data`). Произойдет автоматическое переключение на вкладку результатов работы, представляющей собой текстовый редактор: в случае успешного завершения вычислений появится окно с текстом «Работа над задачей успешно завершена».

5. Заключение

Разработан программный комплекс, в котором реализованы методы и алгоритмы решения оптимизационных задач и задач улучшения управления для различных прикладных областей на кластерном вычислительном устройстве. Парадигма параллельных вычислений в высшей степени соответствует самой природе рассматриваемых задач, связанных с множественностью однотипных операций на верхнем уровне. В программном комплексе успешно реализованы алгоритмы аппроксимации, оптимизации и улучшения приближенно оптимального управления.

Программный комплекс содержит: сервер управления (средство управления и контроля комплексом), управляющие модули (инструменты формирования среды для решения поставленных задач), исполняемые модули (выполняют счет поставленных задач) и интерфейс пользователя (средство запуска счета параметризованных задач).

Для параллельной реализации программного комплекса была использована гетерогенная аппаратная среда. Компоненты программного комплекса физически разделены. Графический интерфейс, сервер управления и управляющие модули работают на платформе IBM PC, а аппаратная платформа для исполняемых модулей вообще не фиксируется. В составе программного комплекса в частности есть исполняемые модули, работающие на аппаратной платформе IBM PC, модули, выполняющиеся на аппаратной платформе суперкомпьютеров «СКИФ» кластерного уровня. Аппаратная платформа суперкомпьютеров «СКИФ» включает: управляющую ЭВМ (фронтенд), вычислительные узлы кластерного уровня; системную сеть КВУ (SCI), объединяющую вычислительные узлы; вспомогательную сеть (семейства Ethernet, с поддержкой TCP/IP), объединяющую управляющую ЭВМ и вычислительные узлы.

Такая гибкость при работе с исполняемыми модулями возможна из-за активного использования протокола SSH (Secure Shell) при построении управляющих модулей, сетевого протокола прикладного уровня, позволяющего производить удаленное управление операционной системой и передачу файлов.

Исследование возможностей и эффективности ПК ISCON проведено на ряде модельных и прикладных задач, включая многокомпонентную модель региона, модель движения вертолета в нештатной ситуации, модель спуска космического челнока в атмосфере с минимальным теплоподводом, модели упругих систем. Результаты позволяют сделать вывод о значительном ускорении вычислительного процесса, близкого к линейному до определенного (оптимального) количества вычислительных узлов. Это подтверждает теоретический прогноз о высокой эффективности распараллеливания.

Список литературы

- [1] Бельшев Д. В., Блинов А. О., Фраленко В. П. *Параллельный алгоритм аппроксимации моделей управляемых систем.* // Труды Четвертой Международной конференции «Параллельные вычисления и задачи управления» РАСО'2008. Москва, 27-29 октября 2008 г. — Москва: Институт проблем управления им. В. А. Трапезникова РАН., 2008. — ISBN 978-5-91450-016-7, с. 968–978. ↑3.1
- [2] Гурман В. И. Принцип расширения в задачах управления. — Москва: Наука. Физматлит, 1997. — 288 с. ↑3.2
- [3] Гурман В. И., Квоков В. Н., Ухин М. Ю. *Приближенные методы оптимизации управления летательным аппаратом.* // Автоматика и телемеханика, № 3, 2008, с. 191–201. ↑3.2, 3.2.3
- [4] Коваленко М. Р., Матвеев Г. А., Осипов В. И., Трушкова Е. А. *Параллельный алгоритм улучшения управления.* // Труды Четвертой Международной конференции «Параллельные вычисления и задачи управления» РАСО'2008. Москва, 27-29 октября 2008 г. — Москва: Институт проблем управления им. В. А. Трапезникова РАН., 2008. — ISBN 978-5-91450-016-7, с. 979–984. ↑3.2

ИССЛЕДОВАТЕЛЬСКИЙ ЦЕНТР ПРОЦЕССОВ УПРАВЛЕНИЯ ИПС РАН

ИССЛЕДОВАТЕЛЬСКИЙ ЦЕНТР ИСКУССТВЕННОГО ИНТЕЛЛЕКТА ИПС РАН

A. O. Blinov, V. I. Gurman, E. A. Trushkova, V. P. Fralenko. *Software package of improvement and optimization of control laws* // Proceedings of Program Systems institute scientific conference “Program systems: Theory and applications”. — Pereslavl-Zalesskij, v. 1, 2009. — p. 25–41. — ISBN 978-5-901795-16-3 (*in Russian*).

АБСТРАКТ. А software package of improving and optimization of control processes for applications in various fields is developed. Its algorithms are successfully realized on the cluster computing device. The capacity and effectiveness of the package are investigated in a number of model examples and applications such as multicomponent model of the region, model of the helicopter maneuvers in emergencies, model of the space shuttle descending in the atmosphere with a minimum of heat supply.