На правах рукописи

Маджара Тарас Игоревич

ИНТЕЛЛЕКТУАЛЬНАЯ СИСТЕМА ДЛЯ РЕШЕНИЯ ЗАДАЧ ОПТИМАЛЬНОГО УПРАВЛЕНИЯ С ВЫЧИСЛИТЕЛЬНЫМИ ОСОБЕННОСТЯМИ

05.13.01 – Системный анализ, управление
и обработка информации

АВТОРЕФЕРАТ

диссертации на соискание ученой степени
кандидата технических наук

Владивосток
2011

Работа выполнена в лаборатории оптимального управления Института динамики систем и теории управления Сибирского отделения РАН (ИДСТУ СО РАН).

Научный руководитель: доктор технических наук
Горнов Александр Юрьевич

Официальные оппоненты: чл.-корр. РАН,
доктор физико-математических наук, профессор
Федотов Анатолий Михайлович

доктор технических наук, профессор
Жирабок Алексей Нилович

Ведущая организация: Институт программных систем
им. А.К. Айламазяна РАН (г. Переславль-Залесский)

Защита состоится « 05 » декабря 2011 г. в 14 часов на заседании диссертационного совета Д 005.007.01 в Институте автоматики и процессов управления ДВО РАН по адресу: 690041, г. Владивосток, ул. Радио, 5.

С диссертацией можно ознакомиться в библиотеке Института автоматики и процессов управления ДВО РАН.

Автореферат разослан « 03 » ноября 2011 г.

Ученый секретарь
диссертационного совета Д 005.007.01, к.т.н. А.В. Лебедев

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Актуальность работы. В настоящее время при решении сложных практических задач, направленных на создание или изучение объектов и процессов в самых разных областях человеческой деятельности, все чаще используются методы системного анализа. Одним из наиболее востребованных средств исследования закономерностей функционирования и развития таких объектов и процессов стали задачи оптимального управления (ЗОУ), охватывающие широкий спектр проблем, таких как динамика полета вертолетов [В.И. Гурман, В.А. Батурин], самолетов и других летательных аппаратов на различных этапах полета [А.И. Тятюшкин, R. Pytlak, R.B. Viner], управление космическими [Р.П. Федоренко] и подводными [M. Chyba, T. Haberkorn, S.B. Singh, R.N. Smith, S.K. Choi] аппаратами, ядерными [Л.Т. Ащепков] и биохимическими [S. Park, W.F. Ramirez] реакторами и многих других. Естественным продолжением теоретических разработок в области численных методов решения ЗОУ стала их реализация на ЭВМ в виде многочисленных комплексов программ. Однако было отмечено, что подавляющее большинство успешно решенных практических задач потребовало привлечения авторов этих программных комплексов, одновременно выступающих в роли экспертов по оптимизации. Решение каждой конкретной практической задачи нуждалось в ручном поиске оригинального вычислительного сценария, предусматривающего многократный запуск комплекса с уточнением алгоритмических параметров и анализом промежуточных результатов, осуществляемых самим разработчиком. Необходимость его привлечения объяснялась многими факторами, в числе которых объективная трудность решения задач оптимизации динамических систем, основную роль в преодолении которой играет не столько простое применение самого численного метода, сколько наличие у пользователя опыта и глубинных знаний предметной области. Понимание этого факта диктовало необходимость дальнейшего усовершенствования методов и средств решения ЗОУ. В частности, вместо использования численного метода в виде последовательности шагов было предложено разработать интеллектно-вычислительный метод, представляющий собой процесс логического анализа, формирующего ту или иную последовательность вычислений, причем отдельные вычисления, в свою очередь, «встроены» в процесс анализа. Сам метод оптимизации в этом случае перестает существовать в традиционном его понимании и превращается в гибрид метода логического вывода и простых вычислений [Коршунов, Коткин, 1991]. Одним из малоисследованных классов ЗОУ, требующих применения экспертного опыта в процессе численного решения, является класс задач с вычислительными особенностями, вызывающими аварийные отказы («АВОСТы») оптимизационных алгоритмов, и, как следствие, не допускающих прямое применение существующих средств оптимизации.

На основании рассмотренных выше задач и требований можно выделить следующие актуальные направления в разработке методов численного решения сложных прикладных ЗОУ с использованием современных программных систем:

1. Исследование класса задач оптимального управления с вычислительными особенностями (ЗОУВО).
2. Формализация накопленного экспертами опыта решения задач рассматриваемого класса.
3. Разработка высокоадаптивных интеллектуальных технологий, позволяющих интегрировать экспертные знания в существующие средства численной оптимизации.

Целью работы является повышение эффективности и надежности существующих средств оптимизации сложных динамических систем рассматриваемого класса путем применения методов искусственного интеллекта, в частности методов продукционной логики. Для этого необходимо решить следующие задачи:

– создание структур данных и алгоритмов, реализующих расчетные методики эксперта при решении задач рассматриваемого класса;
– разработка интеллектуальных программных компонент, осуществляющих принятие решений в ходе управления вычислительным процессом;
– адаптация существующих программных средств под современные вычислительные среды и их интеграция со средствами интеллектуализации;
– проверка работоспособности предложенных вычислительных технологий на тестовых, модельных и содержательных задачах.

Методы и средства исследования. При выполнении работы использовались методы теории оптимального управления, элементы теории построения экспертных систем, метод вычислительного эксперимента, методы построения комплексов прикладных программ, методы искусственного интеллекта и инструментальная среда для разработки экспертных систем CLIPS.

Научная новизна:

1. На множестве ЗОУ выделен класс задач с вычислительными особенностями, описаны подходы к их регуляризации и сформулированы количественные критерии эффективности численного решения.
2. Впервые предложено семейство интеллектуальных алгоритмов, формализующих механизм принятия решения экспертом-вычислителем при численном решении задач рассматриваемого класса.

3. Впервые для оценки и повышения эффективности функционирования средств численной оптимизации динамических систем разработан и применен интеллектуальный динамический планировщик (ИДП).
4. Сформирована оригинальная коллекция задач рассматриваемого класса, включающая в себя как известные, так и специально сконструированные тестовые задачи.

Практическая значимость диссертационной работы состоит в разработке и реализации технологий, совершенствующих существующие средства численного анализа сложных систем с использованием современных методов искусственного интеллекта. Результаты диссертации использованы при реализации проектов, поддержанных грантами РФФИ № 00-01-00731-а «Многометодные процедуры оптимального управления», № 02-01-00889-а «Приближенные методы решения вырожденных задач оптимального управления», № 02-07-90343-в «Internet-технология поддержки удаленного пользователя пакета прикладных программ «OPTCON-2» при решении сложных задач оптимального управления», № 05-01-00477-а «Алгоритмы локально-оптимального синтеза управления с использованием нетейлоровских аппроксимаций условий Кротова и уравнения Гамильтона–Якоби–Беллмана», № 05-01-00659-а «Автоматизация интеллектуального обеспечения методов решения задач оптимального управления», № 09-07-00267-а «Вычислительные технологии интеллектуального анализа временных рядов на основе математических методов теории управления», РГНФ № 09-02-00650 «Разработка компьютеризованных методик для исследования социально значимых медико-экологических проблем региона». Практическая значимость работы подтверждена Актами о практическом использовании в ИАиЭ СО РАН и в ИК им. Борескова СО РАН. Результаты диссертационного исследования используются в учебном процессе НГУ (при подготовке студентов по направлению 230100 – «Информатика и вычислительная техника»).

Достоверность полученных результатов. Разработка и реализация интеллектуального динамического планировщика, представленного в диссертации, проведена с использованием признанного инструментария и в соответствии с теорией построения экспертных систем. Достоверность результатов вычислений обусловлена корректным применением математического аппарата и зарекомендовавших себя программных средств для решения ЗОУ. Для всех решенных задач условия оптимальности (линеаризованный принцип максимума Понтрягина) проверены и выполняются. Работоспособность разработанных технологий подтверждена вычислительными экспериментами на пакете тестовых задач.

Апробация работы. Основные результаты диссертации докладывались и обсуждались на российских и международных конференциях и школах-семинарах: XI Байкальская школа-семинар «Методы оптимизации и их приложения» (Иркутск, 1998), 10-я юбилейная международная конференция по вычислительной механике и современным прикладным программным средствам (Переславль-Залесский, 1999), XII Байкальская международная конференция «Методы оптимизации и их приложения» (Иркутск, 2001), Международная конференция «Математика, ее приложения и математическое образование» (Улан-Удэ, 2002), IV конференция молодых ученых «Навигация и управление движением» (Санкт-Петербург, 2002), школа-семинар молодых ученых «Математическое моделирование и информационные технологии» (Иркутск–Ангасолка, 2002), конференция ИДСТУ СО РАН «Ляпуновские чтения» (Иркутск, 2002), Международная конференция «Вычислительные и информационные технологии в науке, технике и образовании» (Алматы, Казахстан, 2004, 2008), Всероссийская конференция «Математика, информатика, управление» (Иркутск, 2004), Всероссийская конференция молодых ученых по математическому моделированию и информационным технологиям (Кемерово, 2005), Всероссийская конференция «Информационные и математические технологии в науке, технике и образовании», (Северобайкальск, 2005), Международная конференция «Алгоритмический анализ неустойчивых задач» (Екатеринбург, 2008), XIII Байкальская Всероссийская конференция «Информационные и математические технологии в науке и управлении» (Иркутск, 2008), Международная конференция «Вычислительные и информационные технологии в науке, технике и образовании» (Копаоник, Врнячка Баня, Сербия, Будва, Черногория, 2009, 2011).

Публикации. По теме диссертации опубликовано 16 научных работ, в том числе 3 статьи в рекомендованных ВАК научных журналах, 1 – в научном периодическом издании и 12 статей и тезисов в сборниках трудов конференций различного уровня.

Структура работы. Диссертация состоит из введения, четырех глав, заключения и списка литературы, содержащего 103 наименования. Общий объем работы составляет 149 страниц, в тексте содержится 24 рисунка.

Основные защищаемые положения:

1. Структуры данных и алгоритмы их обработки, реализующие расчетные методики эксперта при исследовании задач оптимального управления с вычислительными особенностями.
2. Вычислительная технология, позволяющая решать задачи оптимального управления рассматриваемого класса в автоматизированном режиме.
3. Архитектура и программная реализация интеллектуального динамического планировщика и программных интерфейсных компонент, позволяющих конструктивно преодолевать нештатные ситуации, возникающие при работе алгоритмов оптимизации.

СОДЕРЖАНИЕ РАБОТЫ

Во введении обосновывается актуальность работы, ставятся цели и задачи исследования и приводятся основные положения.

Первая глава посвящена рассмотрению задач оптимального управления с вычислительными особенностями и методов, позволяющих успешно находить их решения. В разделах 1.1 и 1.2 приводится общая постановка задач оптимального управления, содержится обзор существующих теоретических разработок и программных комплексов, освещаются проблемы, возникающие при численном исследовании ЗОУ.

Пусть задан отрезок времени $T=[t_0,t_1]$, на котором определены вектор-функции $u(t)\in R^m$, $x(t)\in R^n$, $t\in T$, задающие управление и фазовое состояние некоторого объекта в момент времени t и удовлетворяющие системе дифференциальных уравнений

$$dx/dt = f(t,x,u), \qquad x(t_0)=x_0. \tag{1}$$

Наложены ограничения на управление: $u(t)\in U\subset R^m$, $t\in T$. (2)

Вектор-функция $x(t)$ предполагается кусочно-дифференцируемой, а $u(t)$ кусочно-непрерывной. Множество пар $(x(t),u(t))$, удовлетворяющих перечисленным условиям, называют множеством допустимых пар и обозначают через D. На множестве D задан целевой функционал $I=F(x(t_1))$. Решение задачи оптимального управления состоит в поиске улучшающей последовательности $\{x^s(t),u^s(t)\}\subset D$, на которой $I(x^s,u^s)\to\inf\limits_{(x,u)\in D} I(x,u),\ s\to\infty.$ Функции $x^*(t),u^*(t): I(x^*,u^*)=\min\limits_{(x,u)\in D} I(x,u)$ называются оптимальной траекторией и оптимальным управлением соответственно. Все итерационные алгоритмы численного решения ЗОУ предполагают априорное наличие некоторой допустимой пары, состоящей из начального управления $u^0(t)$ и соответствующей ему траектории $x^0(t)$, являющихся стартовым элементом улучшающей последовательности $\{x^s(t),u^s(t)\}\subset D$, генерируемой алгоритмами.

Под задачей оптимального управления с вычислительными особенностями будем понимать ЗОУ, которая:

1. Имеет решение (множество допустимых не пусто).
2. Имеет хотя бы одно управление, при котором в системе (1) возникает одно из условий: а) нарушены условия роста[1], гарантирующие существование решения на всем промежутке времени; б) нарушены области определения элементарных математических функций, входящих в правую часть; в) значения переменных выходят за границы возможностей машинного представления чисел с плавающей точкой.

[1] Филлипов А.Ф. Дифференциальные уравнения с разрывной правой частью.

При решении такого типа задач с использованием программных комплексов в улучшающей последовательности $\{x^s(t), u^s(t)\} \subset D$, генерируемой алгоритмом оптимизации, порождается элемент, использование которого в последующих процедурах вычислительного метода приводит к аварийному завершению работы («АВОСТу»).

Одним из подходов, позволяющих успешно решать ЗОУВО, можно назвать метод продолжения по параметру[2]. В разд. 1.3 приводится описание нескольких классов постановочных параметров, изменяющих постановку задачи и погружающих ее в семейство аппроксимирующих вспомогательных задач. Например, вместо управляемой системы (1) рассматривается система («*pf*-параметризация»)

$$dx_i / dt_i = p_i f_i(t, x, u),\ x_i(t_0) = x_i^0. \tag{3}$$

Здесь $p = \{p_i\}$ – векторный параметр, $p_i \in (0,1]$, $i = \overline{1,n}$, n – размерность фазового пространства. Очевидно, что при $p_1 = p_2 = \ldots = p_n = 1$ система (3) полностью идентична исходной. На практике, обычно, рассматривается частный случай построения вектора параметра, когда $p_1 = p_2 = \ldots = p_n = p$. Далее, значение параметра p, при котором соответствующая ему задача идентична исходной, будем обозначать p^*. Помимо представленного выше вида параметризации в разделе описаны такие методы построения аппроксимирующего семейства, как изменение начальных условий задачи Коши (1), изменение границ областей определения элементарных функций, входящих в правую часть динамической системы, ослабление или усиление ресурса управления путем параметризации ограничений (2), построение составных целевых функционалов специального вида, учитывающих специфику задачи и др. Далее формулируется итерационный метод нахождения оптимального управления в исходной задаче на основе решений серии задач из параметрического семейства. В основе метода лежит экспертная гипотеза о том, что две задачи из параметрического семейства, порождаемые двумя близкими по значению параметрами, имеют также близкие решения. Как правило, путем вычислительного эксперимента в аппроксимирующем семействе удается найти последовательность ЗОУ, решение каждой из которых позволяет строить «хорошее» начальное управление для последующей, а в итоге и для исходной задачи. *Информация о возможных способах параметризации задачи и применения метода продолжения по параметру носит экспертный характер, а значит метод не может быть применен пользователем, не имеющим достаточной квалификации.*

Во второй главе формализуется ряд расчетных методик эксперта при решении ЗОУВО методом продолжения по параметру. В разделах 2.1 и 2.2 описаны основные структуры данных: локальные вычислительные схемы и

[2] Холодниок М., Клич А., Кубичек М., Марек М. Методы анализа нелинейных динамических моделей.

вычислительные стратегии. Пусть последовательность ЗОУ параметрического семейства порождается значениями параметра p, изменяющимися в пределах отрезка $P=[p_0, p^*]$. Пусть также отрезок P разбит на N участков $P_i=\left[p_i^0, p_i^{k_i}\right]$, $i=\overline{0, N-1}$, внутри каждого из которых два соседних значения параметра p_i^j, p_i^{j+1} отличаются на фиксированную величину Δp_i, которую назовем шагом по параметру на данном участке.

Тройки $S_i=(p_i^0, p_i^{k_i}, \Delta p_i)$, $i=\overline{0, N-1}$, назовем *локальными вычислительными схемами*. Пусть локальная вычислительная схема S_i *успешна*, если серия ЗОУ, порождаемая последовательностью значений параметра p_i^j, $j=\overline{0, k_i}$ этой схемы, обрабатывается алгоритмом «продолжения по параметру» в штатном режиме (без возникновения «АВОСТа»).

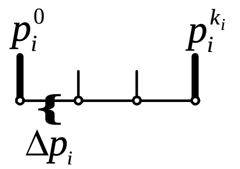

Рис. 1. Локальная вычислительная схема

Вычислительной стратегией назовем упорядоченное множество, состоящее из N локальных вычислительных схем $s=\left\{S_0, S_1, ..., S_{N-1}\right\}=\{S_i\}_{i=0}^{N-1}$ на отрезке P. Оно определяет ход всего процесса решения исходной ЗОУ, задавая начальное значение постановочного параметра $p=p_0$ и дальнейший «кусочно-постоянный» шаг его изменения на всем интервале P. Пример вычислительной стратегии представлен на рис. 2.

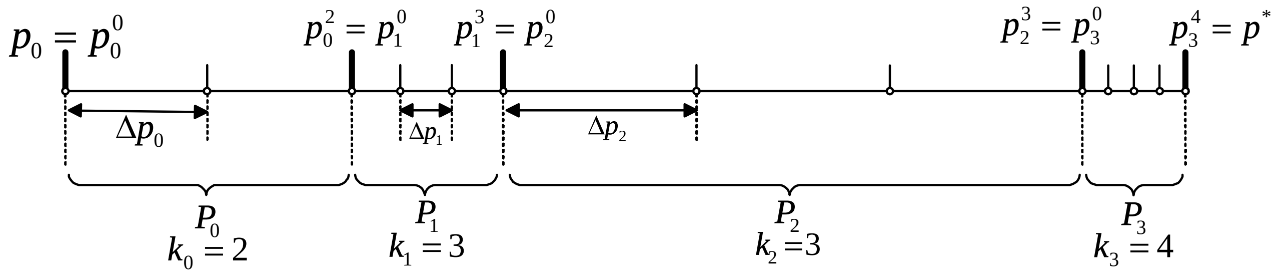

Рис 2. Вычислительная стратегия

Будем говорить, что вычислительная стратегия *успешна*, если все составляющие ее локальные схемы успешны.

Далее в разд. 2.3 вводится три критерия сравнения вычислительных стратегий. В разд. 2.4 расчетные методики формализуются в виде многовариантных итерационных алгоритмов «улучшения» с использованием введенных критериев. Алгоритмы имеют структуру «если-то», что позволяет реализовать их в виде правил-продукций. Нахождение первой успешной вычислительной стратегии эквивалентно построению решения исходной ЗОУВО.

Обозначим $Z\{p,u_p^0\}$ – ЗОУ из параметрического семейства, соответствующая некоторому значению параметра p с начальным управлением u_p^0. Выражением $Z\{p,u_p^0\}\to\varnothing$ обозначим возникновение нештатной ситуации, а выражением $Z\{p,u_p^0\}\to u_p^*$ – факт успешного вычисления оптимального управления u_p^* в задаче $Z\{p,u_p^0\}$. u^0 – начальное управление в исходной задаче. В таких обозначениях ход метода продолжения по параметру в пределах некоторой локальной вычислительной схемы $S=(p^0,p^M,\Delta p)$ описывается последовательностью задач

$$Z\{S,v_S^0\}=\left\{Z\{p^i,v_i^0\}\right\}_{i=0}^M,$$

где $p^i=p^0+i\Delta p,\quad v_0^0=v_S^0=u_{p^0}^0,\quad v_i^0=v_{i-1}^*,\quad Z\{p^i,v_i^0\}\to v_i^*=u_{p^i}^*$.

Если локальная схема успешна, то $Z\{S,v_S^0\}\to v_S^*$.

Аналогично, ход метода продолжения по параметру, задаваемый стратегией $s=\{S_i\}_{i=0}^{N-1}$, можно описать последовательностью задач $Z\{s,u^0\}=\left\{Z\{S_i,v_{S_i}^0\}\right\}_{i=0}^{N-1}$, где $v_{S_0}^0=u^0$, $v_{S_i}^0=v_{S_{i-1}}^*$. Если вычислительная стратегия s успешна, то $v_{S_{N-1}}^*$ – оптимальное управление в исходной задаче.

Алгоритмы поиска успешной стратегии разбиты на 3 группы:

<u>1-я группа:</u> Сужение области поиска в пространстве вычислительных стратегий. Алгоритмы данного этапа являются наиболее простыми и всего лишь устанавливают границы возможного варьирования значений параметра и направление обхода их последовательности в зависимости от выбранного способа параметризации исходной задачи. Например,

<u>**если**</u>: pf-параметризация, <u>**то**</u>: $P=[p_0,1]$, $0<p_0<1$, $p_i^k<p_i^{k+1}$.

<u>2-я группа:</u> Выбор значения p_0 с одновременным генерированием начальной вычислительной стратегии s_0 путем выдвижения и последующей проверки через вычислительный эксперимент соответствующих экспертных гипотез.

2.1. Полагаем p_0=0,9. <u>**Если**</u> $Z\{0{,}9,u^0\}\to\varnothing$, <u>**то**</u> переходим к п. 2.2, иначе p_0 найдено, в качестве s_0 выбирается стратегия (0,9; 1; 0,1) и осуществляется переход на этап ее улучшения.

2.2. Построим итерационную процедуру поиска p_0. Обозначим через $p_0^{(k)}$ и $s_0^{(k)}$ соответственно приближение p_0 и начальную стратегию s_0 на k-й итерации, положив $p_0^{(0)}=p^*=1$, $s_0^{(0)}$ - пусто.

2.3. Пусть проведено k итераций. Последовательно выдвигаются и проверяются три гипотезы $p_0^{(k+1)}=ap_0^{(k)}$, где $a=\{0{,}5;0{,}2;0{,}1\}$: <u>**если**</u> $Z\{0{,}5p_0^{(k)},u^0\}\to u_p^*$,

то завершаем процедуру, полагая $p_0 = 0{,}5 p_0^{(k)}$ и $s_0 = \left\{ s_0^{(k)},\ (0{,}5 p_0^{(k)};\ 0{,}6 p_0^{(k)};\ 0{,}1 p_0^{(k)}),\ (0{,}6 p_0^{(k)};\ p_0^{(k)};\ 0{,}2 p_0^{(k)}) \right\}$;

если $Z\{0{,}5 p_0^{(k)}, u^0\} \to \varnothing, Z\{0{,}2 p_0^{(k)}, u^0\} \to u_p^*$, **то** завершаем процедуру, полагая $p_0 = 0{,}2 p_0^{(k)}$ и $s_0 = \left\{ s_0^{(k)},\ (0{,}2 p_0^{(k)};\ p_0^{(k)}; 0{,}2 p_0^{(k)}) \right\}$;

если $Z\{0{,}5 p_0^{(k)},\ u^0\} \to \varnothing, Z\{0{,}2 p_0^{(k)},\ u^0\} \to \varnothing, Z\{0{,}1 p_0^{(k)}, u^0\} \to u_p^*$,

то завершаем процедуру, полагая $p_0 = 0{,}1 p_0^{(k)}$ и $s_0 = \left\{ s_0^{(k)},\ (0{,}1 p_0^{(k)};\ 0{,}4 p_0^{(k)};\ 0{,}1 p_0^{(k)}),\ (0{,}4 p_0^{(k)};\ p_0^{(k)};\ 0{,}2 p_0^{(k)}) \right\}$;

если $\forall a \in \{0{,}5; 0{,}2; 0{,}1\}\ \ Z\{a p_0^{(k)}, u^0\} \to \varnothing$, **то** полагается $p_0^{(k+1)} = 0{,}1 p_0^{(k)}$, $s_0^{(k+1)} = \left\{ s_0^{(k)}, (0{,}1 p_0^{(k)}; 0{,}2 p_0^{(k)}; 0{,}05 p_0^{(k)}), (0{,}2 p_0^{(k)}; 0{,}4 p_0^{(k)}; 0{,}1 p_0^{(k)}), (0{,}4 p_0^{(k)};\ p_0^{(k)}; 0{,}2 p_0^{(k)}) \right\}$ и пункт 2.3 повторяется (осуществляется переход на следующую итерацию).

3-я группа: Итерационное улучшение полученной на предыдущем этапе вычислительной стратегии s_0. Основная задача 3-й группы алгоритмов – исключить из начальной вычислительной стратегии неуспешные локальные вычислительные схемы и заменить их найденными успешными. Обозначим через $s_{(k)} = \left\{ S_i^{(k)} \right\}_{i=0}^{N_k - 1}$ вычислительную стратегию на k-й итерации, состоящую из N_k локальных вычислительных схем $S_i^{(k)}$ с шагом $\Delta p_i^{(k)}$. Обозначим также через $m(\Delta p_i^{(k)})$ – мантиссу шага $\Delta p_i^{(k)}$. Приведем общую схему алгоритмов 3-й группы. Пусть проведено k итераций.

3.1. **Если** $Z\{s_{(k)}, u^0\} \to u^*$, **то** $s_{(k)}$ – успешна и u^* найдено,

иначе $\exists j : Z\{S_i^{(k)}, v^0_{S_i^{(k)}}\} \to u^*_{S_i^{(k)}},\ i = \overline{0, j-1},\ \ Z\{S_j^{(k)}, v^0_{S_j^{(k)}}\} \to \varnothing$.

3.2. Последовательно выдвигаются и проверяются 2 или 3 гипотезы об эффективном уменьшении шагов по параметру в локальных схемах $S_j^{(k)}, S_{j+1}^{(k)}$: $\Delta \overline{p}_j^{(k)} = a \Delta p_j^{(k)}$, $\Delta \overline{p}_{j+1}^{(k)} = b \Delta p_{j+1}^{(k)}$, где a и b выбираются из таблицы:

		если $m(\Delta p_{j+1}^{(k)})$		
		1	2	5
если $m(\Delta p_j^{(k)})$	1	**то** a={0.5; 0.2; 0.1} b={1.0; 0.5; 0.2}	**то** a={0.5; 0.2; 0.1} b={1.0; 0.5; 0.25}	**то** a={0.5; 0.2; 0.1} b={1.0; 0.2; 0.1}
	2	**то** a={0.5; 0.25; 0.1} b={1.0; 0.5; 0.2}	**то** a={0.5; 0.25; 0.1} b={1.0; 0.5; 0.25}	**то** a={0.5; 0.25; 0.1} b={1.0; 0.2; 0.1}
	5	**то** a={0.2; 0.1} b={1.0; 0.5}	**то** a={0.2; 0.1} b={1.0; 0.5}	**то** a={0.2; 0.1} b={1.0; 0.2}

Таб.1. Выбор коэффициентов алгоритма

3.3. **<u>если</u>** $Z\{\overline{S}_j^{(k)}, v^0_{S_j^{(k)}}\} \to u^*_{S_i^{(k)}}$, **<u>то</u>** полагается

$s_{(k+1)} = \{S_0^{(k)},\dots,\ S_{j-1}^{(k)},\ \overline{S}_j^{(k)}, \overline{S}_{j+1}^{(k)},\dots,\ S_{N_k-1}^{(k)}\}$ и процесс переходит на следующую итерацию с п. 3.1.

<u>если</u> $\forall a\ Z\{\overline{S}_j^{(k)}, v^0_{S_j^{(k)}}\} \to \varnothing$, **<u>то</u>** 1) локальная вычислительная схема $S_j^{(k)}$ разбивается на M одношаговых схем $\{S_j^i\}_{i=0}^{M-1}$ с фиксированным $\Delta p_j^{(k)}$, где M – количество шагов в схеме $S_j^{(k)}$; 2) полагается $s_{(k+1)} = \{S_0^{(k)},\dots,S_{j-1}^{(k)}, S_j^0, S_j^1,\dots,S_j^{M-1}, \overline{S}_{j+1}^{(k)},\dots,S_{N_k-1}^{(k)}\}$ и процесс переходит на следующую итерацию с п. 3.1.

Таким образом, каждая итерация алгоритмов 2-й и 3-й группы разбита на две части: 1) вычислительный эксперимент по решению серии задач из параметрического семейства (часть «если»); 2) модификация вычислительной стратегии в зависимости от результата эксперимента (часть «то»). Коэффициенты дробления, стоящие в алгоритмах при значениях параметров и шагов, а также сами механизмы построения вычислительных стратегий получены экспертами в результате практического опыта решения задач рассматриваемого класса и являются, вообще говоря, недоступными широкому кругу пользователей.

Третья глава посвящена описанию технологий реализации автоматизированного программного обеспечения для решения ЗОУ OPTCON/SMART. В разд. 3.1 проведен краткий анализ традиционного подхода к построению программных средств для решения ЗОУ, позволивший сделать вывод о целесообразности его модификации в целях повышения уровня автоматизации и надежности работы комплексов численной оптимизации динамических систем, в частности, на рассматриваемом классе задач. Далее в разд. 3.2 предлагается расширить существующую архитектуру до двухуровневой иерархии, представленной на рис. 3. На рисунке черными стрелками обозначены управляющие воздействия, а серыми – информационные потоки.

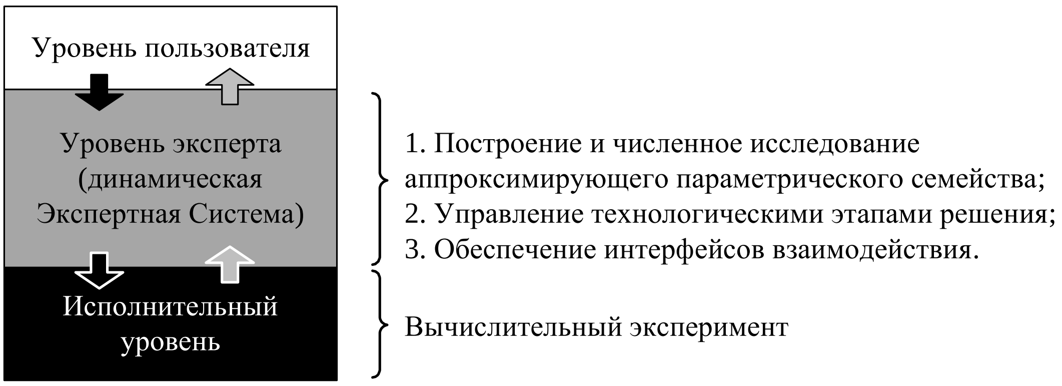

Рис. 3

Исполнительный уровень реализуется вычислительным комплексом для решения ЗОУ, построенным по традиционной схеме с тем отличием, что он управляется не непосредственно конечным пользователем, а уровнем эксперта. Язык взаимодействия экспертного и исполнительного уровней основан на XML.

Программные компоненты, реализующие верхний и нижний уровень архитектуры, названы *интеллектуальным динамическим планировщиком* (ИДП) и *исполнительным модулем* (ИМ) соответственно.

В разд. 3.3 представлена общая архитектура автоматизированной системы OPTCON/SMART, схематически отображаются информационные потоки и управляющие воздействия между ее элементами и описаны средства программной реализации каждой компоненты.

Разд. 3.4 посвящен реализации датчиков нештатных ситуаций – оригинальных программных компонентов, целью работы которых является своевременное информирование ИМ о возникновении «АВОСТа» и одновременно недопущение аварийного завершения процесса в операционной системе. Работа датчиков OPTCON/SMART основана на низкоуровневых механизмах обработки исключительных ситуаций, предлагаемых Intel-совместимыми процессорами.

В разд. 3.5 приведена архитектура (рис. 4) и реализация ИДП, описаны программные механизмы его взаимодействия с ИМ. На рисунке черными и серыми стрелками обозначены, соответственно, управляющие воздействия и информационные потоки.

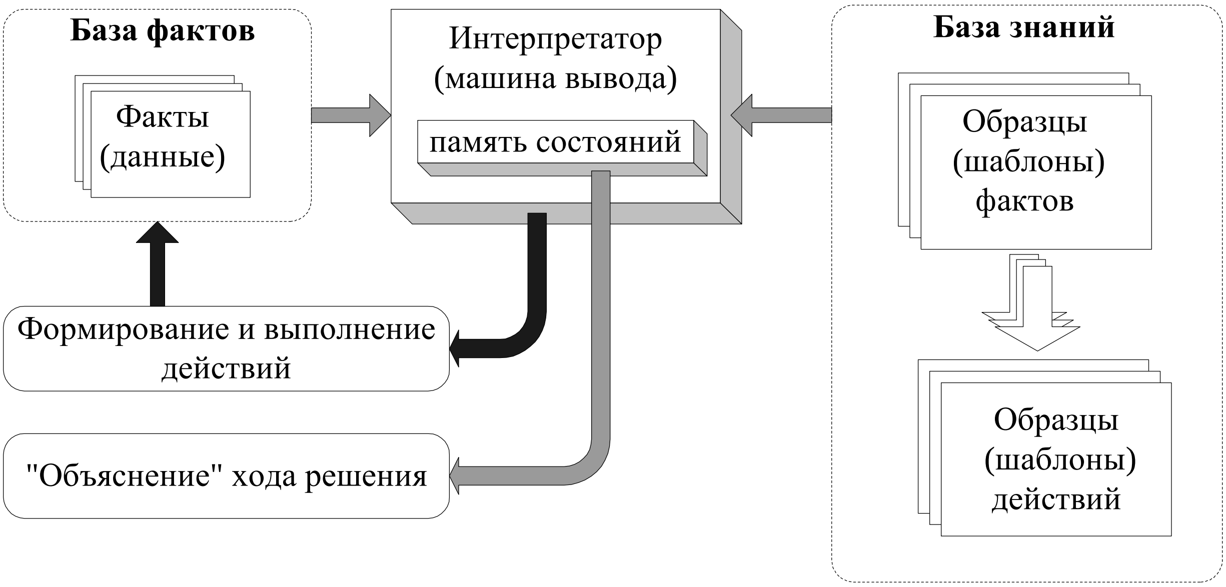

Рис.4. Архитектура ИДП

Планировщик реализован в операционной среде декларативного программирования CLIPS и состоит из трех основных компонентов: 1. *База фактов (рабочая память)*. Содержит как статические данные о решаемой задаче (заданные пользователем априори), так и динамические, т.е. полученные в ходе решения. Важнейшей динамической информацией, содержащейся в базе фактов, является набор локальных вычислительных схем и их текущий статус (успешна, не успешна, не проверена); 2. *База знаний (правил)*. Содержит формализованные в виде правил-продукций алгоритмы, представленные в главе 2, а также правила, управляющие технологическими этапами решения серий задач; 3. *Машина вывода (интерпретатор правил)*. Механизм, непосредственно реализующий расчетные методики эксперта-вычислителя на основе правил, содержащихся в Базе знаний.

В разд. 3.5.2 на языке CLIPS описаны структуры данных, содержащиеся в Базе фактов: локальные вычислительные схемы, содержащие атрибуты экспертных гипотез; состояния вычислительного процесса; аппроксимирующие семейства. Например, шаблон неупорядоченного факта CLIPS, описывающего локальную вычислительную схему, задается следующим образом:

```
(deftemplate ls (slot p0 (type INTEGER))(slot p1 (type INTEGER))
 (slot dp (type INTEGER))(slot status)(multislot role)))
```

Здесь `p0` – слот, содержащий начальное значение постановочного параметра; `p1` – слот, содержащий конечное значение постановочного параметра; `dp` – слот, содержащий шаг по параметру на промежутке от `p0` до `p1`, `status` – слот, описывающий текущий статус схемы. Содержимое данного слота может принимать одно из 3-х значений: **`active`** – активная локальная схема (локальная схема, обрабатываемая в настоящий момент исполнительным модулем); **`none`** – пассивная локальная схема (локальная схема, планируемая к обработке исполнительным модулем, либо еще не проверенная гипотеза об эффективном уменьшении шага по параметру); **`success`** - успешная схема (локальная схема, полностью обработанная исполнительным модулем в штатном режиме без возникновения нештатной ситуации). `role` – слот, описывающий роль схемы в процессе принятия решения. Может принимать два значения: **`std`** - обычная локальная схема; **`spec <N>`** - гипотеза об эффективном уменьшении шага по параметру, проверяемая либо планируемая к проверке исполнительным модулем.

В разд. 3.5.3 описываются принципы построения Базы знаний. Правила-продукции группируются по следующим категориям: резидент нештатных ситуаций; конструктор вычислительных схем; конструктор начального состояния; «супервайзер» вычислительного процесса. LHS правил представляют собой набор условных элементов – ограничений, используемых для того, чтобы определить удовлетворяет ли некоторый факт данному условию. RHS правил определяются набором шаблонов некоторых действий, которые необходимо сформировать и выполнить в случае «срабатывания» данного правила.

RHS правил содержат действия следующих двух основных категорий: а) управление набором локальных вычислительных схем (добавление, модификация, удаление, проверка на успешность); б) управление ИМ (формирование вычислительного эксперимента, конфигурация и вызов ИМ).

В разд. 3.5.3 описана машина вывода. Работа машины вывода традиционно представляет собой цикл, на котором последовательно выполняются следующие три операции: 1) *сопоставление* – LHS каждого правила сопоставляется с текущими элементами Базы фактов. Результат сопоставления – активирование одного или нескольких правил базы знаний; 2) *выбор* (разрешение конфликта) одного правила, наиболее подходящего по заданному критерию, если активировано несколько правил; 3) *действие* – формирование и выполнение всех действий, заключенных в RHS выбранного правила.

Критерием останова является отсутствие в базе фактов элементов, способных активировать хотя бы одно правило.

Разд. 3.6 содержит описание менеджера программной постановки – специализированного модуля, основными задачами которого являются автоматическая параметризация математических выражений, составляющих программную постановку задачи, а также обеспечение пользователя возможностью свободного именования переменных модели.

В разд. 3.7 приведена архитектура и программная реализация исполнительного модуля. Сформулированы требования к вычислительному ядру, позволяющие ему работать под управлением ИДП; описан перечень возможностей вычислительного ядра OPTCON-III. Приводится XML-спецификация протокола взаимодействия ИДП и ИМ.

Совокупность всех представленных в третьей главе технологий позволяет успешно интегрировать существующие средства численного решения задач оптимального управления с интеллектуальным динамическим планировщиком. При этом расчетные методики эксперта-вычислителя, представленные в главе 2 и записанные в виде правил ИДП (фактически – в отдельном текстовом файле), могут свободно корректироваться и дополняться, не требуя доработки самого исполнительного модуля.

Четвертая глава посвящена исследованию возможностей предложенных технологий по решению тестовых, модельных и содержательных ЗОУВО. В разд. 4.1 изложены основные принципы построения коллекции тестовых задач рассматриваемого класса, приводятся постановки и решения 15-ти задач коллекции. Для каждой тестовой задачи строятся аппроксимации множества достижимости с выделением на его плоскости зон попадания траекторий в нештатные ситуации. Ниже приводятся постановка и решение (рис.5) одной из тестовых задач коллекции. Вычислительная особенность этой задачи выражается в нарушении алгоритмом оптимизации области определения функции квадратного корня. На рис. 6, отражающем ход процесса решения, N –

номер вычислительного эксперимента, *p* – значение постановочного параметра. Белые точки на графике обозначают возникновение нештатной ситуации в конкретном эксперименте, а черные – его штатное завершение. На рис. 7, представляющем аппроксимацию множества достижимости, черным цветом обозначена граница зоны возникновения нештатных ситуаций. В результате применения предлагаемого подхода было получено значение функционала $I^* = 8.24918*10^{-4}$.

$$\dot{x}_1 = p(x_2 + u);$$
$$\dot{x}_2 = p\left(x_1 - u - \sqrt{x_1^2 - 0.5x_2^2}\right);$$
$$x(0) = (1,\ 1),\quad t \in [0,\ 4.5];$$
$$I\ (x,u)\ \ =\ \ x_1^2(4.5) + x_2^2(4.5) \to \min;$$
$$p \in (0,1],\ |u| \le 1,\ u^0(t) = 1.$$

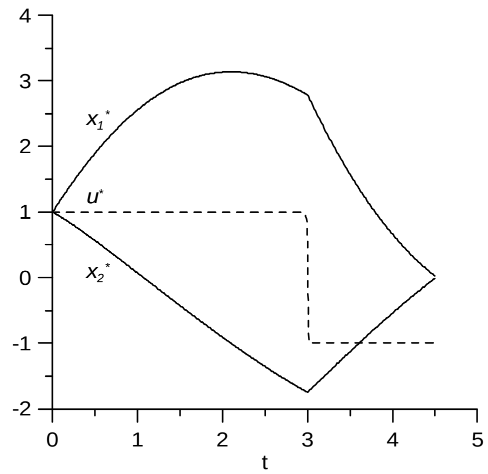

Рис.5. Решение тестовой задачи

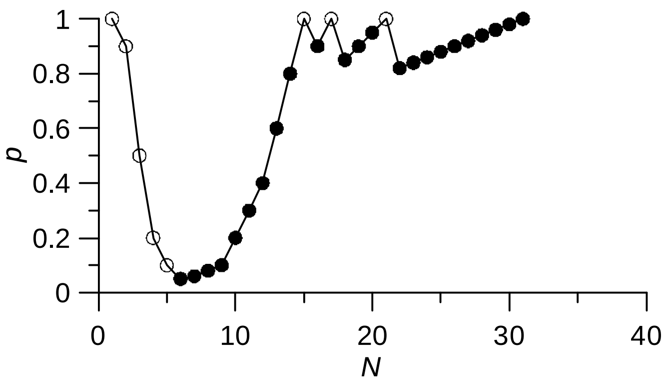

Рис. 6. Ход вычислительного процесса

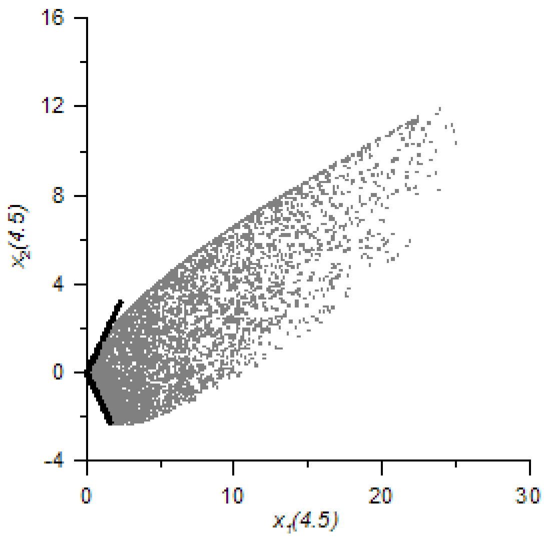

Рис.7. Аппроксимация множества достижимости

В разд. 4.2 приводятся решения нескольких модельных задач исследуемого класса, в частности, задачи о брахистохроне и задачи Годдарда. В разд. 4.3 – 4.5 решены также содержательные задачи об оптимальном управлении биореактором и оптимальном маневре дельтаплана.

Задача о вертикальном взлете ракеты (задача Годдарда). Данный вариант задачи описывает движение ракеты в атмосфере следующей динамической системой:

$$\dot{h} = pv;\ \dot{v} = p\frac{T - 310v^2 e^{-500(h-1)}}{m} - \frac{1}{h^2};\ \dot{m} = -2Tp;$$
$$h(0) = 1,\ \ v(0) = 0,\ \ m(0) = 1.$$

Здесь $h(t)$ – высота полета ракеты относительно центра планеты, $v(t)$ – вертикальная скорость подъема, $0 \le |T(t)| \le 3.5$ – тяга двигателя, $m(t)$ – масса ракеты, p – постановочный параметр. Целью задачи является максимизация высоты полета в конечный момент времени при выполнении терминального ограничения $m(t_f) = 0.6m(0)$. Управлением в задаче служит тяга $T(t)$. Наилучшее из известных для данного варианта задачи значение функционала $h(t_f) = 1.01283$ при $t_f = 0.2$. С использованием разработанной системы было получено значение $h(t_f)$=1.01283 при том же значении t_f. При этом $m(t_f) = 0.59995$.

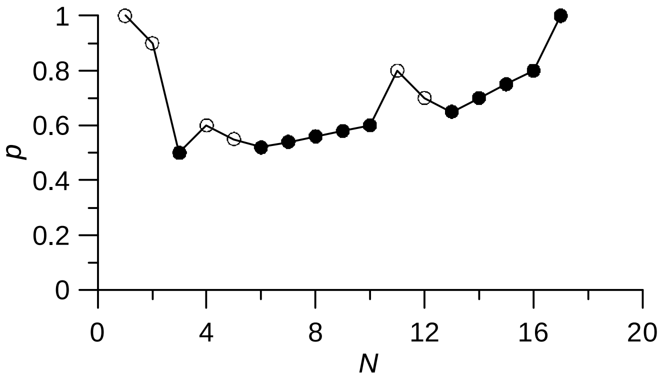

Рис. 8. Ход вычислительного процесса

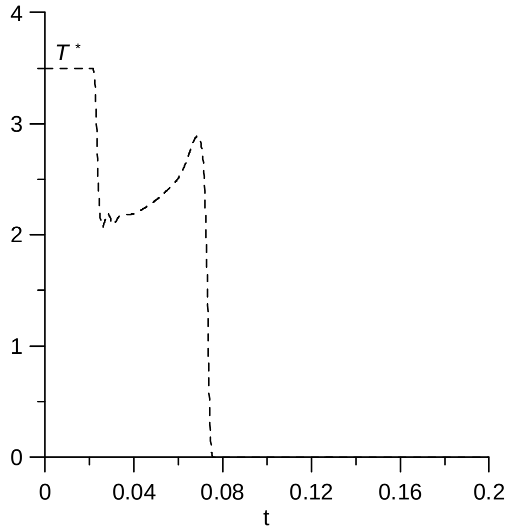

Рис. 9. Тяга двигателя

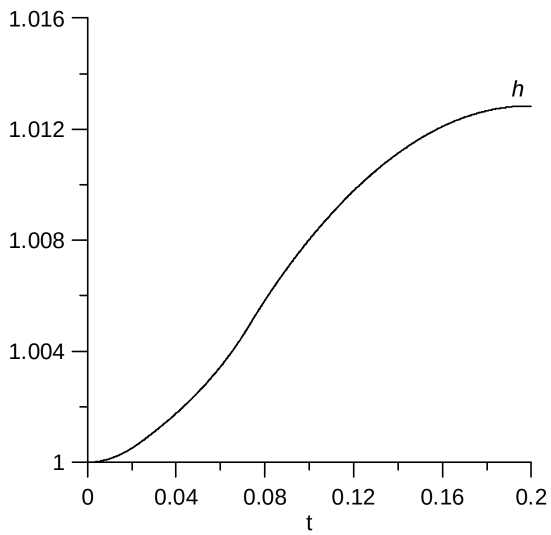

Рис.10. Высота полета

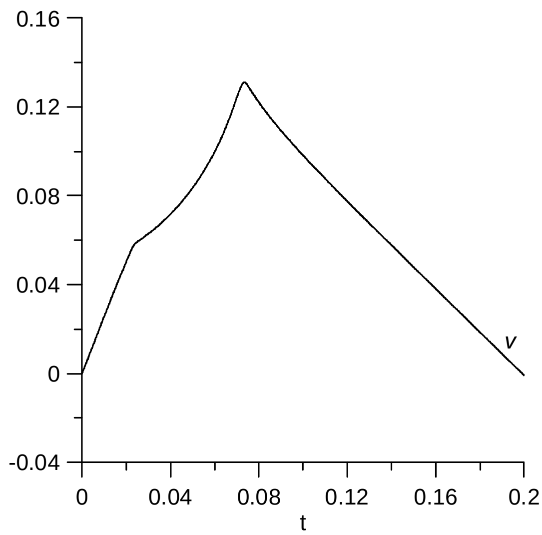

Рис. 11. Вертикальная скорость

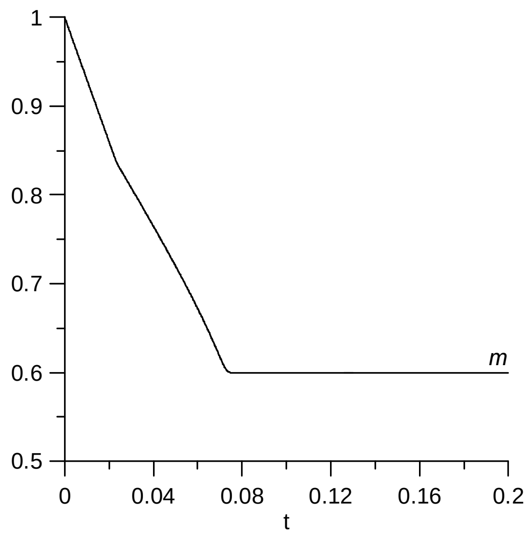

Рис. 12. Масса ракеты

В заключении сформулированы основные результаты диссертационной работы и выводы из проведенных исследований. Приводятся возможные направления дальнейшей работы.

ОСНОВНЫЕ РЕЗУЛЬТАТЫ РАБОТЫ

1. Разработано семейство оригинальных интеллектуальных алгоритмов, реализующих расчетные методики эксперта-вычислителя и позволяющих существенно повысить эффективность существующих средств оптимизации для ЗОУ.
2. На основе разработанных алгоритмов реализован интеллектуальный динамический планировщик (ИДП), а также ряд программных интерфейсных компонент, позволяющих интегрировать его с существующими комплексами для численного решения ЗОУ.
3. Разработано специализированное программное обеспечение OPTCON/SMART, включающее интеллектуальный динамический планировщик, позволяющий проводить решение ЗОУ в автоматизированном режиме. Работоспособность проверена на тестовых, модельных и прикладных задачах.

ОСНОВНЫЕ ПУБЛИКАЦИИ ПО ТЕМЕ ДИССЕРТАЦИИ

1. Маджара Т.И. Технология поиска начального приближения при численном решении задач оптимального управления // Вычислительные технологии. – 2004. – Т. 9 (3). – С. 111–119.
2. Маджара Т.И. Подход к численному решению задач оптимального управления с вычислительными особенностями // Современные технологии. Системный анализ. Моделирование. – 2008. – № 3 (1). – С. 24–29.

3. Маджара Т.И., Горнов А.Ю. Коллекция тестовых задач оптимального управления с вычислительными особенностями // Современные технологии. Системный анализ. Моделирование. – 2009. – № 3 (23). – С. 49–56.
4. Батурин В.А., Маджара Т.И., Урбанович Д.Е. Автоматизация решения задач оптимального управления с использованием базы знаний // Оптимизация, управление, интеллект. – 2004. – № 8. – С. 46–55.
5. Маджара Т.И. Автоматический анализатор вида задачи в автоматизированной системе для решения задач оптимального управления // Тр. XII Байкальской междунар. Конф. «Методы оптимизации и их приложения». Иркутск, 2001. – Т. 2. – С. 120-123.
6. Маджара Т.И. Автоматизированная система для решения задач оптимального управления // Сб. материалов междунар. конф. «Математика, ее приложения и математическое образование». Улан-Удэ, 2002. – С. 265–272.
7. Батурин В.А., Маджара Т.И., Урбанович Д.Е. Автоматизация решения задач оптимального управления // Сб. тр. III всерос. конф. «Математика, информатика, управление». Иркутск, 2004.
8. Маджара Т.И. Интеллектуальный динамический планировщик для решения одного класса задач оптимального управления с вычислительными особенностями // Сб. тр. XIII Байкальской всерос. конф. «Информационные и математические технологии в науке и управлении». Иркутск, 2008. – С. 213–221.
9. Маджара Т.И. Технология решения задач оптимального управления с вычислительными особенностями // Сб. тр. междунар. конф. «Вычислительные и информационные технологии в науке, технике и образовании». Алматы, Казахстан, 2008. – С. 333–341.

Личный вклад. Все основные результаты, представленные в диссертационной работе, получены автором лично. В работах [4, 7] автором предложена архитектура и описаны основные компоненты интеллектуальной системы. В работе [3] автором сформирована часть коллекции ЗОУВО, получены их решения и построены аппроксимации множеств достижимости.

Маджара Тарас Игоревич

Интеллектуальная система для решения задач оптимального управления с вычислительными особенностями

Автореферат

Подписано к печати 28.10.2011 Усл. п.л. 1.0 Уч.-изд.л. 0.8
Формат 60x84/16 Тираж 100 экз. Заказ 16

Редакционно-издательский отдел Учреждения Российской академии наук Института динамики систем и теории управления Сибирского отделения РАН
664033, г. Иркутск, ул. Лермонтова, 134.