

На правах рукописи

Сениченков Юрий Борисович

Основы теории и средства моделирования
гибридных систем

05.13.11 – Математическое и программное обеспечение
вычислительных машин, комплексов и компьютерных сетей

АВТОРЕФЕРАТ

диссертации на соискание ученой степени
доктора технических наук

Санкт-Петербург - 2005

Работа выполнена в государственном образовательном учреждении
высшего профессионального образования
«Санкт-Петербургский Государственный Политехнический Университет»

Официальные оппоненты:

доктор технических наук, профессор Холоднов Владислав Алексеевич,
доктор технических наук, профессор Александров Анатолий Михайлович,
доктор технических наук, профессор Росс Сергей Иванович.

Ведущая организация:

Санкт-Петербургский институт информатики и автоматизации РАН
(СПИИРАН)

Защита состоится 7 апреля 2005 года в 16 часов на заседании
диссертационного совета Д 212.229.18 ГОУ ВПО «Санкт-Петербургский
Государственный Политехнический Университет» по адресу:
194021, Санкт-Петербург, Политехническая 21, ауд. 325.

С диссертацией можно ознакомиться в библиотеке ГОУ ВПО «СПбГПУ».

Автореферат разослан 5 марта 2005 г.

Ученый секретарь

диссертационного совета

д.т.н., проф.

Шашихин В.Н.

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Разработка и исследование универсальных программных комплексов автоматизированного проектирования, анализа и визуализации сложных динамических систем для различных естественнонаучных и технических областей остаются одними из ключевых проблем теории и практики вычислительного эксперимента.

Актуальность темы.

Термин «сложные динамические системы» сегодня чаще всего употребляется для обозначения класса динамических систем, который мог бы называться иерархическими событийно-управляемыми динамическими системами переменной структуры и состава. Под сложностью систем понимается – 1) сложность поведения и 2) сложность структуры, 3) переменный, зависящий от времени состав моделируемых объектов и 4) большое число решаемых уравнений. Сложные динамические системы отличаются высоким уровнем абстракции, трудно воспринимаются человеком и требуют визуализации и на стадии проектирования модели, и при ее анализе.

Сложные динамические системы чрезвычайно востребованы практикой и для их моделирования во всем мире создаются программные комплексы визуального моделирования, включающие в себя графические языки проектирования и отладки моделей, управления вычислительным экспериментом, средства визуализации поведения и интерактивного вмешательства в ход эксперимента.

Сложность поведения модели, прежде всего, связана с наличием нескольких режимов, смена которых обусловлена наступлением внешних или внутренних событий. Сегодня уже не достаточно изучать каждый режим отдельно. Необходимо строить модель всего изучаемого объекта и исследовать ее как единый объект. Для описания и исследования сложных объектов требуются новые математические модели.

Сложность структуры в сочетании со сложностью поведения также требует решения новых научных задач. Современные методы проектирования основаны на блочном, агрегатном подходе, в котором новая модель собирается в основ-

ном из унифицированных компонентов и немногочисленных уникальных блоков. Затем она автоматически представляется в виде большой, трудно воспринимаемой человеком, требующей численного решения, системы уравнений. Для построения модели используются два типа блоков – блоки с входами-выходами, пришедшими из теории управления, и блоки с контактами, успешно используемыми при проектировании механических, гидравлических и электрических систем. Сложные модели, состоящие из компонентов различной физической природы и оснащенные цифровыми системами управления, требуют одновременного использования блоков различного типа. Автоматическое построение совокупной системы уравнений, форма и число которых зависят от времени, и выбор подходящего для нее численного метода становится актуальной теоретической и технической задачей.

В последнее время интерес к графическим языкам моделирования настолько вырос, что их стали рассматривать независимо от реализующих их программных сред. Появились языки, претендующие на звание универсальных. Это – Универсальный Язык Моделирования (UML) и язык «физического» моделирования Modelica. В то же время наиболее распространенным остается подход, реализованный совокупностью компонентов Simulink, Stateflow, SimMechanics, SimPowerSystems пакета Matlab.

Графический язык компонента Simulink (восходящий еще к эпохе аналоговых машин) и язык Modelica поддерживают работу с моделями, режимы поведения и, следовательно, структура решаемых уравнений которых практически не меняются во времени.

Настоятельная практическая потребность моделировать событийно-управляемые объекты заставила авторов Matlab ввести еще один компонент – Stateflow, с помощью которого непрерывные и дискретные модели компонента Simulink управляются картами состояния Харела, а необходимость поддерживать «физическое» моделирование – компоненты SimMechanics, SimPowerSystems.

Первоначально авторы языка Modelica предполагали, что в рамках «физического» моделирования им удастся проектировать модели с различными режимами поведения. Однако им не удалось справиться с трудностями, возникающими в этом случае при формировании совокупной системы. Возможность менять форму и число уравнений при различных режимах в последних версиях языка ограничена.

Описывать и изучать объекты со сложным поведением можно с помощью гибридных систем.

Понятие гибридной системы было введено в начале 90-х годов прошлого столетия с целью распространить методы исследования конечных автоматов на сложные динамические системы с непрерывным временем. Появились программные комплексы символьной верификации, такие как NuTech. Однако возможности этого подхода пока еще ограничены и позволяют исследовать только очень простые гибридные системы.

В середине 90-х автором было предложено использовать гибридные автоматы как основу графических языков программных комплексов для визуального моделирования и численного исследования сложных динамических систем.

Этот подход был в дальнейшем реализован в семействе графических оболочек Model Vision. За рубежом одновременно с первым универсальным пакетом MV 2.1 семейства появился программный комплекс SHIFT для моделирования задач частного типа, а позже гибридные автоматы для численного моделирования стали использовать авторы программных оболочек Ptolemy II и Charon.

Цель работы.

Целью работы является разработка основ теории гибридных систем, необходимой для создания универсальных графических оболочек, предназначенных для численного моделирования иерархических, многокомпонентных, событийно-управляемых систем, состоящих из компонентов с различными типами связей, поведение которых представлено гибридными автоматами, и воплощение теоретических положений в конкретных программных продуктах.

Основные задачи.

Для достижения этой цели ставились следующие задачи. Разработать:

- требования к графическим оболочкам, обеспечивающие быстрое, эффективное и надежное моделирование и численное исследование событийно-управляемых динамических систем;
- основы теории открытых гибридных систем, графическое представление которых в виде открытых гибридных автоматов может быть положено в основу нового графического языка моделирования и поддерживающего его математического обеспечения;
- методы синхронизации открытых гибридных систем и их алгоритмы;
- структуру математического обеспечения, поддерживающего формализм открытых гибридных автоматов, и численные библиотеки на различных алгоритмических языках.

Методы исследования

Для решения поставленных задач использовались теория систем, в частности теория непрерывно-дискретных или гибридных систем, систем с переменной структурой, а также разделы численного анализа, посвященные вычислительным методам линейной алгебры, решению систем нелинейных алгебраических, обыкновенных дифференциальных и алгебро-дифференциальных систем.

Научная новизна

Впервые в мировой практике при создании программных комплексов визуального моделирования сложных динамических систем использован разработанный автором формализм открытых гибридных автоматов.

В результате были разработаны основы теории открытых гибридных систем, что позволило автоматизировать процесс проектирования и численного исследования многокомпонентных иерархических событийно-управляемых динамических систем, в состав которых могут входить блоки различной физической природы со сложным поведением.

Систематизирована теория изолированных гибридных систем. Ее изложение упрощено за счет введения нового понятия – «примитивного гибридного авто-

мата», ориентированного на использование гибридных автоматов для численного моделирования, и расширяющего класс исследуемых моделей до систем алгебро-дифференциальных уравнений

Предложен новый принцип синхронизации открытых гибридных систем. Два метода синхронизации, традиционного дискретного и нового непрерывного типа, реализованы в виде алгоритмов формирования совокупной системы алгебро-дифференциальных уравнений для иерархических функциональных схем с блоками различного типа.

Для автоматизированного построения и анализа совокупной системы:

- Построена классификация уравнений, требующих предварительной проверки корректности. Она позволяет применять для каждого выделенного типа уравнений более простые численные методы, чем это необходимо для решения систем алгебро-дифференциальных уравнений общего вида.
- Обобщено понятие структурной вырожденности, использовавшееся для систем линейных алгебраических уравнений большой размерности, и доказано, что для проверки корректности всех выделенных типов уравнений может быть применен единый алгебраический подход, в основе которого лежит аналог LU-разложения матрицы инцидентности.
- Разработан новый подход, на основании которого можно строить различные алгоритмы поиска точки переключения методом установления. Предложены алгоритмы поиска точек переключения для различных типов предикатов.

Практическая ценность.

На базе теории открытых гибридных автоматов разработана структура математического обеспечения, необходимого для моделирования и численного исследования гибридных систем. В частности, разработана структура эвристических управляющих программ, необходимых для автоматического, без участия пользователя, решения уравнений следующих типов: а) систем нелинейных алгебраических уравнений, б) систем обыкновенных дифференциальных уравнений, с) систем алгебро-дифференциальных уравнений, возникающих при построе-

нии совокупной системы уравнений по иерархическим функциональным схемам с блоками с входами и выходами и блоками с контактами.

Предложенная структура математического обеспечения реализована в виде численных библиотек: на языке Фортран для семейства графических оболочек Model Vision Studium, на языке Java для графической среды Anylogic, как объектно-ориентированный код на языке Java для проекта «Открытая физика». Показаны преимущества и эффективность разработанного программного обеспечения при решении реальных прикладных задач.

Разработанные графические оболочки (MV 2.1 – свидетельство о регистрации № 950277 от 08. 1995, MVS 3.0 - свидетельство о регистрации № 990643 от 09. 1999) уже на протяжении десяти лет используются во многих научных организациях для научной и конструкторской работы, что подтверждено актами о внедрении.

Они успешно используются и в учебном процессе десятков университетов и даже школах: Model Vision в России и странах СНГ (Казахстан, Узбекистан, Украина), проект «Открытая физика» - в США и пакет EJS – в Испании.

Графические оболочки семейства Model Vision по выразительности языка превосходят широко известные компоненты Simulink и Stateflow пакета Matlab и не уступают им по мощности численных библиотек.

Создаваемая в настоящее время версия Model Vision 4.0 решает более широкий класс задач, чем пакет Dymola с входным языком Modelica и компоненты физического моделирования SimMechanics, SimPowerSystems пакета Matlab.

Достоверность результатов

Достоверность результатов подтверждается сравнением качества и точности численного моделирования в пакетах семейства Model Vision с результатами моделирования в других пакетах и решением большого количества практических задач. Только в недавно изданном «Справочнике по расчету гидравлических и вентиляционных систем» графическая среда Model Vision использована для решения более 300 задач.

Апробация работы

Результаты работы с 1996 по 2004 год докладывались более чем на 20 конференциях. На ежегодных отечественных конференциях: «Математика, компьютер, образование», «Компьютерное моделирование», «Математические методы в технике и технологиях», «Дифференциальные уравнения и их приложения», «Информационные технологии в моделировании и управлении» в Санкт-Петербурге и Москве. На международных конференциях – «Информатика и теория управления» (ICI&C) в Санкт-Петербурге, «Winter Simulation Conference» в Орландо (США), и на конференциях международных ассоциаций WSES (в Греции), IEEE (в Словении), CoLoS (в США, Германии, Франции, Словении).

Реализация и внедрение результатов работы

Численные библиотеки семейства Model Vision (www.exponenta.ru) используются также в графической оболочке AnyLogic (www.xjtek.com), проекте «Открытая физика» (США) (<http://webphysics.davidson.edu>). Семейство графических оболочек Model Vision применяется не только в Санкт-Петербурге: в государственном политехническом университете (<http://dcn.nord.nw.ru>, <http://tmm.spbstu.ru>), технологическом институте, институте международных образовательных программ (www.imop.spbstu.ru), международном банковском институте, университете водного транспорта, государственном университете аэрокосмического приборостроения и военно-космической академии им А.Ф. Можайского, но в других отечественных университетах и институтах Москвы, Нижнего Новгорода, Новосибирска, Иркутска, Ростова-на-Дону, Пскова, Пензы для научных исследований и обучения. Графические оболочки используются в странах СНГ - Узбекистане, Казахстане и на Украине.

Актами о внедрении подтверждено использование графических оболочек в ООО «Наука» (Санкт-Петербург), ООО «ПО Киришинефтеоргсинтез», Министерстве Обороны Российской Федерации.

Публикации.

Список публикаций содержит 43 работы: книги и учебники - 4, научные статьи – 12, доклады конференций - 14 и научно-методические статьи - 13, в которых описывается теория и технология моделирования гибридных систем в семействе графических оболочке Model Vision, обосновываются основные научные положения, демонстрируются возможности пакетов для научных исследований и обучения естественнонаучным дисциплинам.

Структура и объем работы

Работа состоит из введения, пяти глав, заключения и приложения. Объем работы - 306 стр., список цитированной литературы – 127 наименований, рисунков – 98, таблиц - 84. В приложение вынесен список статей зарубежных авторов, посвященных гибридным системам.

Содержание работы

Во введении дается краткая характеристика работы: ее актуальность, научная новизна, достоверность и практическая значимость результатов.

В первой главе приводится обзор работ зарубежных и отечественных авторов, посвященных гибридным системам. Анализируются различные определения гибридных систем. Приводится краткая характеристика существующих программных средств моделирования сложных динамических систем, анализируются их недостатки. Формулируются современные требования к графическим оболочкам, и предлагается новый подход к построению графических оболочек на базе гибридных автоматов.

Отмечается, что существуют два подхода к исследованию гибридных систем. Первый подход базируется на обобщении конечных автоматов, сводящемуся к оснащению гибридного автомата различными моделями непрерывного времени. Несмотря на существующие успехи на этом пути, его возможности пока еще ограничены и могут быть использованы только для анализа достаточно простых систем.

Второй подход основан на обобщении классических динамических систем. Этот подход давно уже развивается отечественными учеными (Цыпкин А.З., Бромберг П.В., Андронов А.А., Уткин В.И., Емельянов С.В., Коровин С.К., Филиппов А.Ф., Глушков В.М., Бусленко Н. П.) и пригоден для исследования практически важных сложных динамических систем. Приводится краткий обзор основных отечественных работ в области гибридных систем, больше известных в нашей стране как непрерывно-дискретные системы.

Указывается, что ценность второго подхода заключается, прежде всего, в том, что появляется возможность применять к гибридным системам методы исследования непрерывных систем.

Автором предлагается следующая классификация программных средств моделирования и исследования сложных динамических систем.

Среди программных сред можно выделить специализированные, направленные на решение конкретных прикладных задач, и универсальные, способные справиться практически с любой исследовательской задачей.

Среди универсальных сред существуют несколько групп, отличающиеся типами моделей и методами их построения. Это - математические пакеты, наиболее приспособленные для изучения изолированных классических динамических систем, и пакеты компонентного моделирования с блоками с входами выходами и с блоками с контактами, предназначенные для моделирования иерархических компонентных систем.

Математическое обеспечение сред компонентного моделирования решает, по сравнению с математическими пакетами, другие математические задачи: автоматического синтеза совокупной системы из уравнений отдельных компонент с учетом связей блоков; проверки корректности итоговой системы и преобразования ее к форме, необходимой программной реализации численного метода; автоматического выбора численного метода для достоверного воспроизведения поведения системы.

В работе делается вывод, что наиболее перспективными следует считать оболочки способные одновременно моделировать:

- однокомпонентные системы с непрерывным поведением,
- однокомпонентные системы с гибридным поведением,
- многокомпонентные системы с компонентами типа «вход-выход-состояние»,
- многокомпонентные системы с компонентами, взаимодействующими через контакты,
- многокомпонентные системы с переменным числом компонентов.

Для построения таких оболочек, по мнению автора работы, следует использовать и развивать теорию гибридных систем.

Во второй главе построены основы теории открытых гибридных систем, предлагается новый тип синхронизации, систематизирована теория изолированных гибридных систем.

В работе предлагаются следующие новые определения гибридных систем, ориентированные на численное моделирование. Под изолированной гибридной системой H понимается кортеж

$$H = \{s, f(s), g(s), S_0, pred(t, s(\tau)), Inv(t, s(\tau)), Init(s(\tau))\},$$

где

- $s \in \mathfrak{R}^n$ - вектор переменных состояния размерности n ;
- $f(s), g(s) \in \mathfrak{R}^n$ – заданные вектор-функции или правые части уравнений

$$\frac{ds_1}{dt} = f(t, s_1, s_2),$$

$$0 = g(t, s_1, s_2)$$

$$s = \{s_1, s_2\}; \quad s(0) = s_0 \in S_0, \quad t \in [0, \infty)$$

Составляющие вектора s_1, s_2 называются дифференциальной и алгебраической компонентами решения соответственно.

- S_0 - множество начальных условий, включающее в себя все начальные условия, порождаемые функцией инициализации в процессе функционирования гибридной системы;

- $pred(t, s(t)) : \mathcal{R}^1 \times \mathcal{R}^n \rightarrow \{false, true\}$ - булевская функция, определенная на решениях алгебро-дифференциального уравнения, выделяющая особые состояния, или события, приводящие к смене поведения;
- $Inv(t, s(t)) : \mathcal{R}^1 \times \mathcal{R}^n \rightarrow \{false, true\}$ - булевская функция, определяющая свойства решения;
- $Init(s(t)) : \mathcal{R}^n \rightarrow \mathcal{R}^n$ - вещественная функция, ставящая в соответствие значению решения в правой, конечной, точке t_-^* текущего промежутка интегрирования значение новых начальных условий на новом временном промежутке в его левой, начальной, точке $t_+^* : s(t_+^*) = Init(s(t_-^*))$.
- τ - гибридное время.

Определим гибридное время τ либо как конечную последовательность $\tau_H = \{\tau_i\}_1^N$, где $\tau_i = (Pre_gap_i, [t_{i-1}^*, t_i^*], gap_i)$, $i = 1, 1..N$; $t_0^* = 0$ и $[t_{i-1}^*, t_i^*]$ являются замкнутыми интервалами, и только последний интервал полуоткрытый - $\tau_N = [t_N^*, \infty)$, либо как последовательность $\tau_H = \{\tau_i\}_1^\infty$, состоящую только из замкнутых конечных интервалов. Элементы гибридного времени Pre_gap_i, gap_i назовем временными щелями очередного такта (тика) $\tau_i = (Pre_gap_i, [t_{i-1}^*, t_i^*], gap_i)$ $\tau = \{\tau_1, \tau_2, \dots\}$ гибридного времени, где

- t^* - время срабатывания перехода, или первая точка, в которой становится истинным предикат,

- $pre(s(t^*))$ - значение функции на правом $pre(t^*)$ (конечном) конце текущего временного интервала,

- $post(s(t^*))$ - значение функции на левом $post(t^*)$ (начальном) конце следующего временного интервала,

- Pre_gap_i - временная щель для вычисления согласованных начальных условий и проверки предиката на левом конце нового промежутка $[t_{i-1}^*, t_i^*] = [t_{i-1}^*, pre(t_i^*)]$ очередного длительного поведения,

- $[t_{i-1}^*, t_i^*]$ - временной интервал текущего длительного поведения (непрерывное поведение),
- gap_i - временная щель для вычисления новых начальных условий на правом конце текущего промежутка $[t_i^*, t_{i+1}^*] = [post(t_i^*), t_{i+1}^*]$ очередного длительного поведения.

Изолированную гибридную систему удобно изображать в виде примитивного гибридного автомата (Рис. 1).

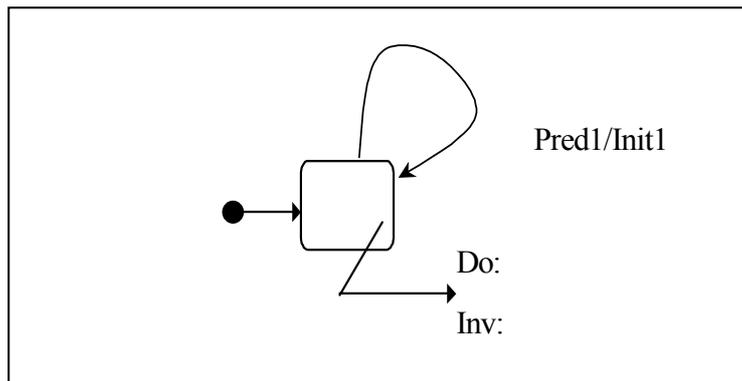


Рис. 1 Примитивный изолированный гибридный автомат

Работа изолированного примитивного автомата описывается следующим алгоритмом:

Алгоритм работы примитивного автомата.

1. *Инициализация.*

$$t_0^* = 0; i := 0; \quad ;$$

$$post(s_1(t_0^*)) := pre(s_1(t_0^*)) := s_{10};$$

$$post(s_2(t_0^*)) := pre(s_2(t_0^*)) := s_{20}$$

здесь s_{20} либо согласованное начальное условие для алгебраической составляющей, либо только начальное приближение к нему.

2. Pre_gap_i : *Начинается текущий промежуток гибридного времени τ_i . Вычисление новых согласованных начальных условий и проверка предиката.*

$i := i + 1$; Решаем уравнение $0 = g(t_{i-1}^*, post(s_{10}(t_{i-1}^*)), post(s_{20}(t_{i-1}^*)))$ относительно $post(s_{20}(t_{i-1}^*))$ с начальным приближением $pre(s_{20}(t_{i-1}^*))$.

→ Аварийный выход: не смогли найти согласованные начальные условия.

Вычисляем предикат на левом конце текущего промежутка $pred(post(s(t_{i-1}^*)))$

Если предикат истинен $pred(post(s(t_{i-1}^*))) = true$, то

завершается текущий интервал гибридного времени τ_i

$t_i^* := t_{i-1}^*$; $pre(s(t_i^*)) := post(s(t_{i-1}^*))$;

переходим к 4.

иначе

начинается текущее длительное поведение

3. $[t_{i-1}^*, t_i^*]$: Текущее длительное поведение.

Решение уравнений

$$\frac{ds_{10}}{dt} = f(t, s_{10}, s_{20});$$

$$0 = g(t, s_{10}, s_{20})$$

с согласованными начальными условиями, до тех пор пока не станет истинным предикат $pred(s(t))$.

→ Аварийный выход: не смогли найти решить уравнения.

Как только предикат стал истинным, выполняем

завершается текущий интервал гибридного времени τ_i

$t_i^* := t$: $pred(s(t)) = true$; $pre(s(t_i^*)) := s(t)$

4. gap_i : Завершающее мгновенное поведение. Инициализация началь-

ных условий для нового интервала: $post(s(t_{i-1}^*)) := Init(pre(s(t_{i-1}^*)))$;

здесь $post(s_{10}(t_{i-1}^*))$ - новое начальное условие для дифференциальной составляющей, а $post(s_{20}(t_{i-1}^*))$ - возможно, только приближение к нему.

Переходим к 2.

Конец Алгоритма работы примитивного автомата.

В большинстве языков моделирования гибридный автомат вводится как автомат с несколькими длительными состояниями. Граф, соответствующий

гибридной системе с несколькими длительными состояниями, с приписанными узлам системами дифференциальных уравнений и дугами, помеченными предикатами функциями инициализации, называется картой поведения или гибридным автоматом.

В работе доказано, что любой детерминированный гибридный автомат с несколькими состояниями может быть сведен к примитивному гибриднему автомату. С теоретической точки зрения более удобен примитивный гибридный автомат, поэтому именно он использован для систематизации теории изолированных гибридных систем. Однако, с практической точки зрения, более удобен автомат с несколькими состояниями, поэтому карты состояния положены в основу графического языка.

В рассмотренном изолированном примитивном автомате все переменные состояния были равноправными. Для построения модели из компонентов в изолированном примитивном гибридном автомате

$$H = \{w, F(w), G(w), W_0, pred(t, w(t)), Inv(t, w(t)), Init(w(t))\},$$

представим множество переменных w в виде трех непересекающихся множеств $w = In \cup Out \cup S$ и $In \cap Out \cap S = \emptyset$ и будем называть элементы x множества In входами, элементы y множества Out - выходами, а элементы s множества S - переменными состояниями. Запишем уравнения длительных состояний в виде

$$\begin{aligned} \frac{ds_1}{dt} &= f(s_1, s_2, x, t) \\ 0 &= g(s_1, s_2, y, x, t); s = \{s_1, s_2\} \in S; x \in In; y \in Out \end{aligned}$$

и назовем полученный автомат открытым примитивным автоматом «вход-выход-состояние».

Два открытых примитивных автомата «вход-выход-состояние» Hio_1^o и Hio_2^o называются объединенными в блок-схему $\langle Hio_1^o \parallel Hio_2^o \rangle$, если хотя бы один выход или вход одного из них равен входу или выходу другого (связаны между собой). Соответствующие пары переменных называются связными. Не связные переменные называются свободными.

Образование блок-схем или связывание открытых гибридных автоматов по входам и выходам определяется следующими правилами:

1. Любой выход может быть связан с любым числом входов.
2. Любой вход можно быть связан только с одним выходом.

Композицией $Hio_1^O \parallel Hio_2^O$ двух независимых примитивных открытых гибридных «вход-выход-состояние» Hio_1^O и Hio_2^O

$$Hio_1^O = \{w_1, f_1(t, x_1, s_1^1, s_2^1), g_1(t, x_1, y_1, s_1^1, s_2^1), S_{01}, pred_1(t, w_1), Inv_1(t, w_1), Init_1(t, w_1)\},$$

$$Hio_2^O = \{w_2, f_2(t, x_2, s_1^1, s_2^1), g_2(t, x_2, y_2, s_1^1, s_2^1), S_{02}, pred_2(t, w_2), Inv_2(t, w_2), Init_2(t, w_2)\},$$

$$w = \{x, y, s\};$$

называется примитивный автомат с объединенной системой уравнений

$$\frac{ds_1^1}{dt} = f_1(t, s_1^1, s_2^1, x_1)$$

$$0 = g_1(t, s_1^1, s_2^1, x_1, y_1);$$

$$\frac{ds_1^2}{dt} = f_2(t, s_1^2, s_2^2, x_2)$$

$$0 = g_2(t, s_1^2, s_2^2, x_2, y_2);$$

двумя предикатами исходных автоматов и доопределенными функциями инициализации. Доопределение функций инициализации необходимо во временных точках, где смена поведения наступает только у одного автомата. В этом случае второй автомат лишь приостанавливает свое текущее поведение и продолжает его же, как только второй автомат придет в новое состояние. Таким образом, доопределение происходит так же, как и в случае изолированных автоматов.

Блок-схеме $Hio_1^O \parallel Hio_2^O$ двух открытых примитивных автоматов «вход-выход-состояние» Hio_1^O и Hio_2^O

$$Hio_1^O = \{w_1, f_1(t, x_1, s_1^1, s_2^1), g_1(t, s_1^1, s_2^1, x_1, y_1), S_{01}, pred_1(t, w_1), Inv_1(t, w_1), Init_1(t, w_1)\},$$

$$Hio_2^O = \{w_2, f_2(t, x_2, s_1^2, s_2^2), g_2(t, s_1^2, s_2^2, x_2, y_2), S_{02}, pred_2(t, w_2), Inv_2(t, w_2), Init_2(t, w_2)\},$$

$$w = \{x, y, s\};$$

соответствует автомат с объединенной системой уравнений

$$\frac{ds_1^1}{dt} = f_1(t, s_1^1, s_2^1, x_1)$$

$$0 = g_1(t, s_1^1, s_2^1, x_1, y_1);$$

$$\frac{ds_1^2}{dt} = f_2(t, s_1^2, s_2^2, x_2)$$

$$0 = g_2(t, s_1^2, s_2^2, x_2, y_2);$$

и новыми уравнениями связи $x_1^1 = y_1^2; x_1^2 = y_1^1$; где - x_1^1 и x_1^2 - векторы, составленные из связных входов первого и второго автоматов, y_1^1 и y_1^2 - векторы, составленные из связных выходов, двумя предикатами и новой, доопределенной функцией инициализации.

В открытых гибридных автоматах возможны различные трактовки одновременности событий во временной щели – «дискретная» и «непрерывная». При традиционной дискретной синхронизации новые согласованные условия вычисляются последовательно для каждого блока, и результат вычислений может зависеть от порядка обхода блоков (различного упорядочивания одновременных событий во временной щели). При предложенной в работе непрерывной синхронизации обеспечивается вычисление новых начальных условий решаемых уравнений независимо от порядка обхода.

Различные способы синхронизации реализованы в виде алгоритмов, позволяющих формировать совокупную систему уравнений по функциональным схемам. Компонентные модели можно строить, используя не только «входы-выходы», но и контакты. Для таких блоков также вводится понятие блок-схемы и композиции.

В третьей главе рассматриваются проблемы предварительного анализа структуры совокупной системы и проблемы продвижения модельного времени. Для структурного анализа уравнений разработан метод, аналогичный методу для анализа больших разреженных систем линейных алгебраических уравнений. Построены методы поиска точки переключения для предикатов различного вида, и предложен новый подход, позволяющий создавать различные модификации метода установления, применяемого для поиска точки переключения. Обсуждаются стратегия выбора шага для визуализации поведения.

При блочном моделировании достаточно сложно предсказать заранее, какую форму примет итоговая система уравнений и какого типа она окажется. В то же время от формы и типа уравнений во многом зависит эффективность решения, поэтому в структуре математического обеспечения необходимо иметь блок предварительного анализа уравнений.

Требования к решению формируются при проведении вычислительного эксперимента - пользователь может потребовать максимально возможной скорости решения без визуализации поведения или детального графического воспроизведения решения. Эти задачи возложены на блок продвижения модельного времени, который содержит чрезвычайно важный для гибридных систем блок поиска точек смены поведения - точек переключения.

С практической точки зрения важно предоставить пользователю как можно больше свободы при описании поведения отдельных компонентов и не заставлять его использовать трудновоспринимаемые синтаксические формы, свойственные процедурным языкам программирования. Под свободной формой будем понимать форму, в которой уравнения первого порядка не разрешены относительно первых производных $F\left(\frac{ds}{dt}, s, t\right) = 0$. Ориентация на свободную форму оправдана еще и тем, что это не только удобная форма представления исходной информации, но и форма, естественным образом возникающая при использовании блоков с контактами.

Свободная форма редко используется в программных реализациях численных методов. Более распространенной является форма $A(s) \frac{ds}{dt} = f(s, t)$, с возможно вырожденной матрицей $A(s)$.

В каждом конкретном случае систему, передаваемую решателю, следует автоматически проверять и приводить из свободной формы к более простой, если это возможно. Эту работу выполняет блок предварительного структурного анализа уравнений. Структурному анализу можно подвергать как системы отдельного блока, так и всю совокупную систему уравнений.

Структурный анализатор выделяет:

- *вычисляемые последовательности формул,*
- *структурно невырожденные системы нелинейных алгебраических уравнений,*

Для каждого из типов уравнений – последовательность формул, система нелинейных уравнений и система нелинейных уравнений - вводится матрица структуры Q , столбцам которой соответствуют вычисляемые переменные, взятые в том же порядке, как они появляются в последовательности формул или уравнений, а строкам – формулы или уравнения. К матрице структуры применяется аналог алгоритма исключения Гаусса, и анализируется построенная матрица. Например, последовательность формул оказывается вычисляемой, если существует матрица перестановок P_1 , такая что матрица $P_1(Q)P_1^T$ имеет блочно-диагональный вид, и каждый блок является строго нижней треугольной матрицей.

Алгебраические методы можно применить и к дифференциальным уравнениям, если уравнения в свободной форме $F\left(\frac{ds}{dt}, s, t\right) = 0$ привести к виду

$$\frac{ds}{dt} = z$$

$$F(z, s, t) = 0$$

Одновременно можно найти структурный индекс полученной системы.

Сформированная система передается решателю, задача которого найти численное решение в конкретной временной точке. Выбор этой конкретной точки определяется блоком продвижения модельного времени и зависит от режима работы пакета.

Таких режимов два: решение с визуализацией поведения и решение без визуализации. В первом случае требование качества визуализации накладывает ограничение на выбор возможной величины шага продвижения модельного времени, так как необходимо воспроизвести графики с учетом всех особенностей поведения, во втором ограничений на величину шага не существует. Вторым случаем характерен для решения задач оптимизации и любых других расчетов большого числа вариантов с различными значениями параметров, когда нас интересуют не промежуточные, а конечные результаты.

Как в первом, так и во втором случае возникает еще одно требование, связанное с моделированием гибридных систем: при численном интегрировании в любом режиме должны быть правильно определены точки смены поведения, или точки переключения.

К основным проблемам поиска точки переключения следует отнести построение надежного алгоритма локализации очередной точки переключения с заданной точностью (глобальный поиск) и построение быстрого алгоритма поиска локализованной точки (локальный поиск).

Первая проблема связана с выбором величины шага модельного времени. Решение второй проблемы зависит от того, какая информация о свойствах решения доступна численному алгоритму.

Смена поведения возможна после истечения заданного времени, получения сообщения, наступления истинности заданного предиката. Первые два случая не вызывают затруднений, так как явно определяют точку переключения, однако последний случай может потребовать больших вычислительных затрат и привести к качественным ошибкам.

Условие смены поведения можно записывать в виде булевской функции $Pr_b : boolean \rightarrow boolean$ и предиката $Pr_g : \text{arg of any MVS type} \rightarrow boolean$. Обе

формы допускают применение алгоритмов, опирающихся только на вычисление значений функций, но не их производных, и потому достаточно медленных.

В гибридных системах поиск точки переключения может осложняться тем, что точка смены поведения может лежать на границе множества допустимых значений фазового вектора: а) текущего длительного состояния, б) нового длительного состояния, в) обоих состояний, и решение не может быть продолжено за точку переключения.

Поиск в условиях, когда можно вычислить только значение предиката в окрестности точки переключения.

В данном случае применяется метод деления отрезка пополам, и в качестве нового значения модельного времени выбирается значение правой точки окончательного локализирующего интервала. Новые начальные условия в этой точке вычисляются по значениям текущих переменных на левом конце окончательного локализирующего интервала.

Поиск в условиях, когда можно вычислить не только значение функции, определяющей точку переключения, но и ее производную.

В этом случае, после того как точка переключения локализована, вместо неравенств, описывающих смену поведения, строятся уравнения, и применяется различные модификации метода Ньютона.

Поиск в условиях, когда точка переключения определяется как корень уравнения на решении, описывающем поведение.

Здесь используются те же алгоритмы, что и в предыдущем пункте, только теперь уравнения уже заданы и формировать их не нужно.

Рассмотрим особенности локализации корня в условиях непродолжаемости решения за точку переключения.

Одним из способов поиска точки переключения является переход от поиска корня нелинейного алгебраического уравнения к поиску стационарного режима дифференциального уравнения, для которого положение равновесия является корнем исходного алгебраического уравнения. В вычислительной математике такой прием называется поиском корня методом установления. Очевид-

ным достоинством этого приема является то, что автоматически обеспечивается невозможность перейти за точку переключения.

Пусть требуется найти корень функции $g(s) = 0$ на решении уравнения $\frac{ds}{dt} = f(s), s(0) = s_0$. В написанной системе осуществим переход от независимой переменной t к новой независимой переменной h . Переход к новой независимой переменной и дифференцирование функции $g(s)$ по независимой переменной приводит к системе:

$$\begin{aligned}\frac{ds}{dh} &= f(s) \frac{dt}{dh}; \\ \frac{dg}{dh} &= \frac{\partial g}{\partial s} \frac{\partial s}{\partial t} \frac{dt}{dh} = \frac{\partial g}{\partial s} f(s) \frac{dt}{dh};\end{aligned}$$

Как видно из последней записи, зависимость $\frac{dt}{dh}$ пока произвольна, и ее можно выбрать так, чтобы корень исходного алгебраического уравнения оказался бы устойчивой стационарной точкой новой системы. Обозначим эту зависимость через R , положив $\frac{dt}{dh} = R$. Выбор функции в виде $R = -g(s) \left(\frac{\partial g}{\partial s} \right)^{-1}$ обеспечивает устойчивость и приводит к хорошо известным уравнениям метода установления

$$\begin{aligned}\frac{ds}{dh} &= - \left(\frac{\partial g}{\partial s} \right)^{-1} g(s) f(s), s(0) = s_0; \\ \frac{dg}{dh} &= -g(s) f(s), g(0) = g(s_0); \\ \frac{dt}{dh} &= - \left(\frac{\partial g}{\partial s} \right)^{-1} g(s); t(0) = 0\end{aligned}$$

Четвертая глава посвящена разработанной структуре математического обеспечения для моделирования гибридных систем. Предлагаются структуры конкретных управляющих программ («решателей»), вызывающих различные программные реализации численных методов автоматически, без участия пользователя. Обсуждаются вопросы эффективности, приводятся результаты

тестирования и предлагаются новые наборы тестовых примеров. Предложенные структуры иллюстрируются на примере библиотеки численных методов графической среды MVS.

Схематически библиотеку численных методов MVS можно изобразить следующим образом (табл.1).

Таблица 1

| | | |
|---------|---------|---------|
| BLAS | FLOAT | |
| LINPACK | EISPACK | MINPACK |
| LA | GNA | |
| NAE_SLV | ODE_SLV | DAE_SLV |

Программы библиотеки представляют собой иерархическую структуру. Уровни (строки таблицы) являются независимыми группами, использующими только программы предыдущих уровней (табл. 2).

Таблица 2

| | |
|---------|---|
| BLAS | Систематизированная коллекция подпрограмм, которая реализует основные элементарные операции линейной алгебры. |
| FLOAT | Подпрограммы этого раздела вычисляют основные характеристики машинной арифметики. |
| LINPACK | Систематизированная коллекция подпрограмм предназначена для решения систем линейных алгебраических уравнений с матрицами различного типа. |
| EISPACK | Систематизированная коллекция подпрограмм предназначена для решения проблемы собственных значений. |
| MINPACK | Систематизированная коллекция подпрограмм предназначена для нахождения минимумов функционалов и решения систем нелинейных алгебраических уравнений. |
| LA | Подпрограммы этой группы реализуют основные матричные операции. |
| GNA | Группа подпрограмм, предназначенная для решения основных задач численного анализа общего назначения. |
| NAE_SLV | Группа подпрограмм, предназначенная для решения систем нелинейных алгебраических уравнений. |
| ODE_SLV | Группа подпрограмм, предназначенная для решения систем |

| | |
|---------|--|
| | обыкновенных дифференциальных уравнений. |
| DAE_SLV | Группа подпрограмм, предназначенная для решения систем алгебро-дифференциальных уравнений. |

Для решения каждой из основных задач (систем нелинейных алгебраических, дифференциальных и алгебро-дифференциальных уравнений), пользователю предлагается на выбор несколько программных реализаций численных методов.

1. Автоматические решатели. Их назначение - автоматически осуществить подбор наименее трудоемкой программной реализации численного метода для решения конкретной задачи, а в случае неудачи предоставить пользователю максимально подробную информацию о встреченных трудностях. Анализируя поведение автоматического решателя, пользователь может затем выбрать конкретный метод.

2. Программные реализации для решения задач конкретного типа. Если пользователю известен тип решаемой задачи, то достаточно просто выбрать один из предлагаемых на выбор методов соответствующего класса. Однако пользователь имеет дело не непосредственно с программной реализацией из соответствующей систематизированной коллекции, а с управляющей программой, анализирующей коды завершения вызываемой подпрограммы и, в случае необходимости, меняющей ее параметры так, чтобы обеспечить получение решения на всем интервале моделирования. Например, если программная реализация не может обеспечить получение решения с указанной пользователем точностью, то управляющая программа сначала попытается изменить точность, а если это не поможет, то только тогда прекратит счет и выдаст аварийное сообщение.

3. Программные реализации для отладки. Цель этой группы методов - продвигать модельное время вперед как можно дальше, чтобы дать возможность пользователю понять природу ошибок. К этой группе относятся явный и неявный методы Эйлера с автоматическим выбором шага и метод Рунге-Кутты четвертого порядка с постоянным шагом.

Эффективность созданных численных библиотек подтверждается результатами тестирования и многочисленными численными экспериментами, примеры которых приведены в конце главы.

Предлагается и иллюстрируется примерами новый способ построения тестовых примеров для проверки качества программных реализаций численных методов решения систем нелинейных алгебраических уравнений.

В пятой главе приводятся примеры, подтверждающие эффективность созданного программного обеспечения. Примеры условно разделены на две группы. В первую вошли работы научных коллективов, использовавших оболочку Model Vision для решения новых научных и инженерных задач и опубликовавших результаты своих исследований. Приводятся аргументы, поясняющие, почему авторы использовали оболочку Model Vision вместо широко распространенных в нашей стране зарубежных графических оболочек. Во вторую группу вошли коллективы, применившие оболочку Model Vision для учебных целей и написавшие свои руководства по ее использованию.

В заключении подводятся итоги работы, формулируются основные выводы.

В приложении приводится список зарубежных публикаций, посвященных гибридным системам.

Выводы и результаты

Разработаны основы теории открытых гибридных систем. Теоретические положения использованы при построении программных комплексов для визуального моделирования и численного исследования сложных динамических систем.

Основные выводы и результаты работы сводятся к следующему.

1. Современные программные комплексы плохо приспособлены для моделирования объектов со сложным поведением. Разработаны требования к программным комплексам для моделирования иерархических событийно-управляемых динамических систем переменной структуры и состава. Для удов-

летворения этим требованиям предложен новый подход к моделированию и численному исследованию сложных динамических систем на базе гибридных систем.

2. Существующие методы символьной верификации сложных динамических систем не удовлетворяют потребностям практики. Для решения практических задач предложено конструктивное определение изолированной гибридной системы, позволяющее использовать гибридные автоматы для численного моделирования и расширить класс исследуемых моделей до систем алгебро-дифференциальных уравнений. Систематизирована теория изолированных гибридных систем и упрощено ее изложение на базе введенного понятия примитивного гибридного автомата.

3. Разработаны основы теории открытых гибридных систем. Введено и формализовано понятие открытых гибридных систем, на базе которого автоматически строится совокупная система алгебро-дифференциальных уравнений по иерархическим функциональным схемам из блоков с входами и выходами и блоков с контактами.

4. Доказано, что традиционная синхронизация гибридных систем не всегда правильно описывает поведение сложных динамических систем. Предложен новый принцип синхронизации открытых гибридных систем. Два метода синхронизации – дискретного и непрерывного типа - реализованы в виде алгоритмов формирования систем алгебро-дифференциальных уравнений, описывающих поведение гибридной системы с синхронизацией дискретного и непрерывного типа.

5. Предложена классификация уравнений, возникающих при использовании языка, в основе которого лежат гибридные автоматы, и требующих предварительной проверки корректности. Она позволяет применять для каждого выделенного типа уравнений более простые численные методы, чем это необходимо для решения систем алгебро-дифференциальных уравнений общего вида.

6. Обобщено понятие структурной вырожденности, используемое для систем линейных алгебраических уравнений большой размерности, позволяющее

алгоритмизировать анализ структурной вырожденности всех выделенных типов уравнений. Для проверки корректности всех выделенных типов уравнений предложен единый алгебраический подход, в основе которого лежит аналог LU-разложения матрицы инцидентности.

7. Разработан новый подход, на основании которого возможно построение различных алгоритмов поиска точки переключения методом установления. Предложены и реализованы в виде алгоритмов методы поиска точки переключения для предикатов различного типа.

8. Разработана структура математического обеспечения, необходимого для численного моделирования гибридных систем и структура управляющих программ, необходимых для автоматического решения уравнений следующих типов: систем нелинейных алгебраических уравнений; систем обыкновенных дифференциальных уравнений; систем алгебро-дифференциальных уравнений, возникающих при построении совокупной системы уравнений на основании иерархических функциональных схем с блоками с входами и выходами и блоков с контактами.

Построен новый набор тестов для проверки программных реализаций численных методов решения систем нелинейных алгебраических уравнений, позволяющих конструировать системы, имеющие неединственное решение.

9. Разработаны численные библиотеки для семейства графических оболочек Model Vision (Fortran) и для среды Anylogic (Java). Они также использованы в проекте «Открытая физика» (Physlet Physics <http://webphysics.davidson.edu>) (Java) и пакете EJS (Java) (www.colos.org). Графические оболочки Model Vision (www.exponenta.ru) и Anylogic (www.xjtek.com), в которых наиболее полно воплощены результаты диссертации, получили признание специалистов, о чем, например, свидетельствуют аннотации, размещенные на сайте «Hybrid System Software» (<http://www-er.df.op.dlr.de/cacsd/hds/software.shtml>).

10. Графические оболочки Model Vision и Anylogic широко используются для решения научных и инженерных задач, которые сложно или невозможно ре-

шать с помощью других программных средств, что подтверждается десятками опубликованных работ различных научных коллективов.

11. Графическая оболочка Model Vision успешно используется в учебном процессе многих университетов России и в странах СНГ. Ее численные библиотеки для проекта «Открытая физика» используются в США, для графической среды EJS - в Испании. Графической оболочке Model Vision выделена специальная страница на хорошо известном в нашей стране образовательном сайте (www.exponenta.ru). Информация о среде Model Vision и посвященных ей книгах размещена на федеральном портале «Инженерное образование».

12. Показаны преимущества и эффективность разработанного программного обеспечения при решении реальных прикладных задач.

Список опубликованных работ.

книги и учебники

1. Сениченков Ю.Б. Моделирование. Вычислительный практикум: Учеб. пособие./ Веселова И.Ю. - СПб.: Изд-во СПбГТУ, 1999. – С.107.
2. Сениченков Ю.Б. Визуальное моделирование сложных динамических систем.- / Колесов Ю.Б.-СПб.: Изд-во «Профессионал», 2000. – С. 242.
3. Сениченков Ю.Б. Практическое моделирование сложных динамических систем. / Бенькович Е.С., Колесов Ю.Б.- СПб.: Изд-во БХВ, 2001.- С. 441.
4. Сениченков Ю.Б. Численное моделирование гибридных систем. - СПб.: Изд-во СПбГПУ, 2004.- С. 206.

научные статьи

5. Сениченков Ю.Б. Библиотека программ для решения ОДУ. / Колесов Ю.Б. // Вычислительные, измерительные и управляющие системы: Тр. СПбГТУ. № 462. СПб: Изд-во СПбГТУ,1996. - С. 116-122.
6. Сениченков Ю.Б. Экспертная оценка качества программных реализаций численных методов для решения ОДУ. / Воскобойников С.П., Устинов С.М // Вычислительные, измерительные и управляющие системы: Тр. СПбГТУ. № 462. СПб: Изд-во СПбГТУ,1996. С. 122-141.

7. Сениченков Ю.Б. Современное программное обеспечение для проведения вычислительного эксперимента. / Колесов Ю.Б., Расторгуев И.В. // Вычислительные, измерительные и управляющие системы: Тр. СПбГТУ. № 468. СПб: Изд-во СПбГТУ, 1997. С. 95-98.
8. Сениченков Ю.Б. Свидетельство об официальной регистрации программ для ЭВМ «Model Vision Studium версия 3.0» / Инихов Д.Б., Колесов Ю.Б. № 990643- М.: Роспатент, 6.09.1999.
9. Сениченков Ю.Б. Объектно-ориентированный подход к моделированию сложных динамических систем. / Колесов Ю.Б. // Вычислительные, измерительные и управляющие системы: Тр. СПбГТУ. № 476. СПб: Изд-во СПбГТУ, 1998. С.19-23.
10. Senichenkov Yu. Test problems for Hybrid Simulators. / Kolesov Yu. // Recent Advances in Theoretical and Applied Mathematics. Singapore: World Scientific Publishing Co. Pte. Ltd., 2000. P. 214-220.
11. Senichenkov Yu. Object-Oriented Modeling in Model Vision Studium./ Kolesov Yu. // Recent Advances in Signal Processing and Computer Technologies. Singapore: World Scientific Publishing Co. Pte. Ltd., 2001. P. 413-418.
12. Сениченков Ю.Б. Современные подходы к компьютерному моделированию динамических систем. / Колесов Ю.Б. // Научно-технические ведомости СПбГТУ. -2002. - № 3. С. 93-102.
13. Сениченков Ю.Б. Компьютерное моделирование в научных исследованиях и образовании. / Колесов Ю.Б. // Exponenta_Pro. Математика в приложениях. - 2003. - № 1. С. 4-11.
14. Сениченков Ю.Б. Программная поддержка активного вычислительного эксперимента. / Колесов Ю.Б. // Научно-Технические ведомости. 2004. № 1. С. 177-182.
15. Сениченков Ю.Б. Синхронизация событий при использовании гибридных автоматов для численного моделирования сложных динамических систем. / Колесов Ю.Б. // Научно-Технические ведомости. 2004. № 1. С. 202-207.

16. Сениченков Ю.Б. Моделирование сложных динамических систем. // Научно-Технические ведомости. 2005. № 1. С. 198-207.

доклады на конференциях

17. Senichenkov Yu. Hybrid systems and numerical solution of ODE. / Kolesov Yu. // Differential equations and applications. The first International conference. St. Petersburg, 1996. P.105-107.

18. Senichenkov Yu. Model Vision 3.0 for Windows. - The graphical environment for complex dynamical system design. / Kolesov Yu. // ICI&C'97: International conference on Informatics and Control. St. Petersburg, 1997. P. 764-768.

19. Senichenkov Yu. Visual specification language for event-driven hierarchical dynamical systems with variable structure. / Kolesov Yu. // ICI&C'97: International conference on Informatics and Control. St. Petersburg, 1997. P. 704-711.

20. Сениченков Ю.Б. Объектно-ориентированный подход к моделированию сложных динамических систем. / Колесов Ю.Б. // Фундаментальные исследования в технических вузах: Научно-техническая конференция. С.-Петербург, 1997. С. 107

21. Senichenkov Yu. Model Vision's 3.0 ODE Solver. / Kolesov Yu., Pariajskaj E. // Differential equations and applications: The second International conference. St. Petersburg, 1998. P.122-124.

22. Senichenkov Yu. Model Vision 3.0 - graphical environment for complex dynamic systems modeling. / Kolesov Yu. // COLOS-98. Maribor. 1998. P.34-41.

23. Сениченков Ю.Б. Учебные и методические материалы по теме «Моделирование»./ Бенькович Е.С., Лосева Е. А. // Информационные технологии в моделировании и управлении: II межд. конференция. С.-Петербург. 2000. С. 65-67.

24. Сениченков Ю.Б. Семейство Model Vision. / Колесов Ю.Б. // Информационные технологии в моделировании и управлении: II межд. конференция, С.-Петербург, 2000. С.188-190.

25. Сениченков Ю.Б. Применение MVS в химии и биологии. / Колесов Ю.Б. // Математические методы в технике и технологиях: ММТТ-2000. С.-Петербург, 2000. С.154-157.

26. Senichenkov Yu. Java engine for UML based hybrid state machines. / Borshchev A., Kolesov Yu. // Proceedings of Winter Simulation Conference. Orlando, California, USA, 2000. P. 1888-1897.

27. Сениченков Ю.Б. Объектно-ориентированное моделирование в графической среде Model Vision Studium. / Колесов Ю.Б. // Математика, компьютер, образование: 8-я межд. конференция. Пущино, 2001. С.242-244.

28. Сениченков Ю.Б. Model Vision Studium - инструмент для объектно-ориентированного моделирование сложных динамических систем. / Колесов Ю.Б.// Гибридные системы. Model Vision Studium: 2-я межд. конференция. С.-Петербург, 2001. С.5- 46.

29. Сениченков Ю.Б., Компьютерное моделирование/ Колесов Ю.Б. // Компьютерное моделирование 2003: 4-тая межд. конф-ция. СПб, 2003. С. 4-6.

30. Senichenkov Yu. A composition of open hybrid automata. / Kolesov Yu. // EUROCON 2003: IEEE, Region 8. Ljubljana, Slovenia, 2003. P. 332-335.

научно-методические публикации

31. Сениченков Ю.Б. Математическое моделирование и вычислительный эксперимент. // Компьютерные инструменты в образовании.-1998.-№ 1.С.24-32.

32. Сениченков Ю.Б. Как быстро и качественно "приготовить" интерактивную модель динамической системы. / Колесов Ю.Б. //Компьютерные инструменты в образовании.- 1998.- №3-4. С. 21-30.

33. Сениченков Ю.Б. Математическое моделирование в картинках или рисуем поведение динамических систем с помощью MODEL VISION. / Колесов Ю.Б. // Компьютерные инструменты в образовании.- 1998. - № 5. С.17-24.

34. Ю.Б. Сениченков. Имитационное моделирование сложных динамических систем. / Колесов Ю.Б. // Компьютерпресс. - 2000.- № 3. CD-приложение. С.12.

35. Сениченков Ю.Б. Model Vision Studium – компьютерная лаборатория для моделирования и исследования сложных динамических систем. / Колесов Ю.Б. // Компьютерпресс.- 2000. - № 3. CD-приложение. С.24.

36. Сениченков Ю.Б. Model Vision Studium – Виртуальный стенд для изучения и проектирования сложных динамических систем. Краткий обзор возможностей. / Колесов Ю.Б. // Компьютерпресс. - 2000. - № 3. CD-приложение. – С.5.
37. Сениченков Ю.Б. Компьютерные инструменты педагога. // Компьютерные инструменты в образовании. 2002. - № 2. С. 81-84.
38. Сениченков Ю.Б. Школа моделирования 2003. Урок 1. / Колесов Ю.Б. // Компьютерные инструменты в образовании. -2003.- № 1. С. 39-49.
39. Сениченков Ю.Б. Школа моделирования 2003. Урок 2. Моделирование движения объекта в среде MVS. / Макарова Н.В., Титова Ю.Ф. // Компьютерные инструменты в образовании. -2003.- №2. С. 75-83.
40. Сениченков Ю.Б. Школа моделирования 2003. Урок 3. Возможности компьютерного эксперимента в среде MVS. / Макарова Н.В., Титова Ю.Ф. // Компьютерные инструменты в образовании. -2003.- № 3. С. 55-64.
41. Сениченков Ю.Б. Школа моделирования 2003. Урок 4. Моделирование систем. / Макарова Н.В., Титова Ю.Ф. // Компьютерные инструменты в образовании. -2003. -№ 4. С. 61-71.
42. Сениченков Ю.Б. Школа моделирования 2003. Урок 5. Имитационное моделирование. / Макарова Н.В., Титова Ю.Ф. // Компьютерные инструменты в образовании.- 2003. - № 5. С. 61-71.
43. Сениченков Ю.Б. Школа моделирования 2003. Урок 6. Графические возможности MVS. / Колесов Ю.Б., Макарова Н.В., Титова Ю.Ф. // Компьютерные инструменты в образовании.- 2003.- № 6. С. 65-71.

Лицензия ЛР № 020593 от 07.08.97

**Налоговая льгота – Общероссийский классификатор продукции
ОК 005-93, т.2; 953004 – научная и производственная литература**

Подписано в печать 10.02.2005. Формат 60x84/16

Усл. печ. л. 2,0. Уч.-изд. л. 2,0. Тираж 100. заказ

**Отпечатано с готового оригинал-макета, предоставленного автором, в
типографии Издательства Политехнического университета.
195251, Санкт-Петербург, Политехническая, 29.**