ПРОГРАММНЫЙ СТЕНД ИСПЫТАНИЙ И ОТЛАДКИ МАТЕМАТИЧЕСКИХ МОДЕЛЕЙ СЛОЖНЫХ ДИНАМИЧЕСКИХ СИСТЕМ

А. В. Тишков[1], М. Н. Тишкова[2], А. А. Мусаев[3]

Санкт-Петербургский институт информатики и автоматизации РАН
199178, Санкт-Петербург, 14-я линия В.О., д.39
[1]<avt74@yahoo.com>, [2]<mromashova@hotmail.com>,
[3]<musayev@yandex.ru>

УДК 665.63
А. В. Тишков, М. Н. Тишкова, А. А. Мусаев. **Программный стенд испытаний и отладки математических моделей сложных динамических систем //** Труды СПИИРАН, Вып. 2, т. 2. — СПб.: Наука, 2005.
Аннотация. *Рассмотрена задача построения программного стенда, предназначенного для испытаний, отладки и верификации математических моделей, используемых в различных задачах наблюдения и управления. Стенд включает в себя базу знаний математических моделей, программный генератор тестовых сигналов и анализатор качества функционирования моделей. Реализация стенда осуществлена на основе java-технологий.* — Библ. 3 назв.

UDC 665.63
A. V. Tishkov, M. N. Tishkova, A. A. Musaev. **Software test bench for mathematical models debugging and verification**// SPIIRAS Proceedings. Issue 2, vol. 2. — SPb.: Nauka, 2005.
Abstracts. *The problem of software test bench development for debugging and verifications of the mathematical models used in various observations and control tasks is considered. The test bench includes knowledge base of mathematical models, the program generator of test signals and the quality of models functioning analyzer. Realization of the test bench is carried out on the basis of java-technologies*. — Bibl. 3 items.

1. Введение и постановка задачи

Современная прикладная математика предоставляет разработчикам систем мониторинга и управления обширный арсенал эффективных математических моделей, используемых для решения задач оценивания, идентификации, распознавания, прогнозирования и т.п.

В частности, широко применяются методы статистической обработки данных — параметрического оценивания, проверки гипотез, многомерного статистического анализа. Среди современных технологий компьютерной математики следует указать алгоритмы искусственных нейронных сетей, генетические алгоритмы, эволюционные методы и другие.

Указанные модели относятся к общей методологии Data Mining и широко используются при решении задач поддержки принятия решения и оптимизации процессов управления.

Однако непосредственное применение существующих математических моделей и отвечающих им программных комплексов сопряжено с существенными трудностями. Проблема состоит в несоответствии статистической структуры исходных данных (результатов мониторинга) совокупности ограничений, гарантирующих сохранение функциональной эффективности используемых математических моделей.

Как правило, результаты мониторинга образуют нестационарные нелинейные временные ряды, содержащие как систематические, так и случайные

Труды СПИИРАН. 2005. Вып. 2. Т. 2. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2005. Issue 2. V. 2. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

ошибки. При этом статистические характеристики погрешностей измерений заранее неизвестны и эволюционируют во времени. В результате этого прямое применение математических моделей в конкретных задачах мониторинга и управления часто приводит к существенному снижению точности оценки состояния объектов управления, уменьшению достоверности прогноза и эффективности формируемых решений. Возникает необходимость в периодической перенастройке и отладке алгоритмов управления и соответствующего программного обеспечения. Значительно возрастают сроки разработки и внедрения программно-алгоритмических средств управления.

В связи с указанным, возникает проблема создания специализированного программного комплекса, ориентированного на процессы автоматизированной отладки, испытаний и верификации математических моделей, используемых в качестве основы для разработки современных средств мониторинга и управления сложными динамическими системами. При этом предполагается, что разрабатываемый системный комплекс будет содержать базы знаний типовых математических моделей, используемых в задачах мониторинга и управления, как произвольной размерности, так и пониженной размерности (2 или 3), что позволяет визуализировать реакцию указанных моделей на различные типы возмущающих воздействий.

Формирование частных возмущений осуществляется специализированным блоком, позволяющим, совместно с блоком автоматизированного анализа качества функционирования модели, заранее получить сведения об устойчивости той или иной модели к различным типам вариаций параметров и структуры входных данных.

Адекватность тестовых сигналов реальным процессам, примеры которых хранятся в соответствующей БД комплекса, проверяется на основе использования статистических метрик и критериев согласия.

Аналоги подобных программных комплексов авторам неизвестны.

2. Описание стенда

Программный стенд испытания и отладки (ПСИО ММ) математических моделей (ПСИО ММ) разрабатывался на основе:

- пакета программ генерации имитационных и тестовых процессов, разрабатываемых в среде MatLAB [1, 2];
- блока процедур дескриптивного анализа данных;
- пакета визуализируемых (размерности 2–3) статистических моделей виртуального мониторинга, разрабатываемых в среде MatLAB;
- уже существующих моделей виртуальных анализаторов, разработанных ранее в различных средах (MatLAB, Excel, Borland Power Builder).

В основу разрабатываемого ПСИО ММ положены средства тестирования двух базовых классов математических моделей, основанных на статистическом и кибернетическом подходах. Класс статистических моделей включает в себя множество моделей виртуального анализа и прогнозирования, основанные на методах корреляционного, регрессионного, дискриминантного и компонентного анализа, а также на алгоритмах сплайн-аппроксимации.

Класс кибернетических моделей включает в себя множество моделей виртуального анализа и прогнозирования, основанные на технологиях Data Mining (нейронные сети, генетические алгоритмы, метод группового учета аргументов, нечеткая логика и др.).

Труды СПИИРАН. 2005. Вып. 2. Т. 2. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2005. Issue 2. V. 2. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

Общая структура ПК ИОВ ММ приведена на рис. 1.

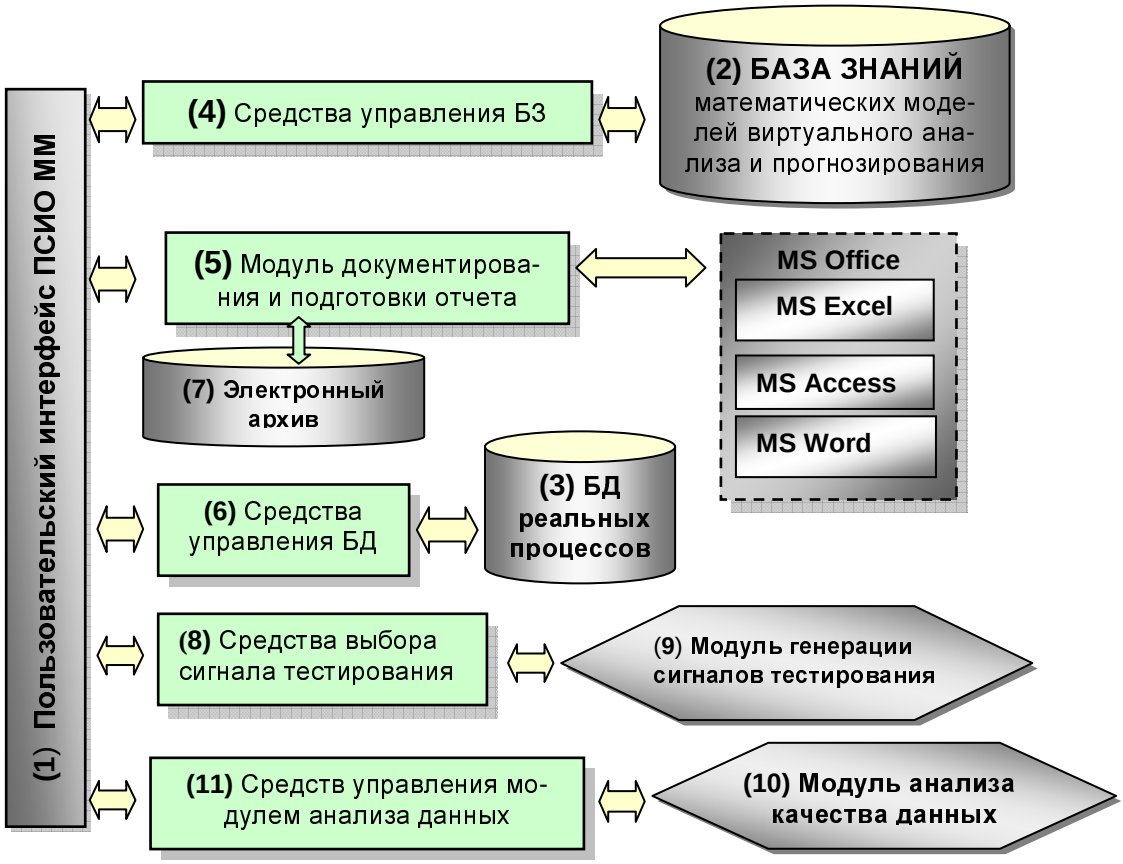

Рис 1. Общая структура ПСИО ММ.

Основные блоки и модули ПСИО ММ, а также входящие в состав этих блоков модели и подсистемы перечислены в табл. 1.

Управление ПСИО ММ осуществляется через программный графический интерфейс, позволяющий обеспечить необходимые взаимодействия с другими подсистемами программного стенда в соответствии со схемой на рис. 1.

Подсистемы комплекса имеет возможность оперативного взаимодействия с подсистемами Office 2000 (MS Excel, MS Access, MS Word) и MatLAB 6.

3. Особенности реализации программного стенда

В ходе реализации программного испытательного стенда математических моделей, был разработан набор абстрактных Java-классов, на которых основываются такие компоненты системы, как генераторы исходных данных, а так же классы «математические модели». Базовые классы предназначены для упрощения дальнейшей разработки классов-потомков, носителей функциональной нагрузки в системе. В них собраны все типичные поля и методы, которые впоследствии использовались всеми классами-потомками, вне зависимости от их предназначения и особенностей. Такой подход ведёт к уменьшению объёма кода программы, повышает надёжность и способствует лучшему восприятию кода и структуры системы в целом сторонними разработчиками.

Труды СПИИРАН. 2005. Вып. 2. Т. 2. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2005. Issue 2. V. 2. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

Таблица 1

№	Основные блоки и модули	Модели и подсистемы
1.	БЗ математических моделей, включающая в себя следующие базовые математические модели и их программы	Статистическая модель восстановления регрессионной зависимости
		Статистическая модель дискриминантного анализа
		Статистическая модель кластерного анализа
		Статистическая модель компонентного анализа
		Статистическая модель сплайн-аппроксимации
		Нейросетевая модель с обратным распространением ошибки
		Нейросетевая модель на основе карт Кохонена
		Нейросетевая модель Хопфилда
		Модель эволюционного программирования
		Модели на основе генетических алгоритмов
2.	Программный модуль генерации имитационных сигналов	Базовый программный генератор случайных величин с гауссовским и равномерным законами распределения
		Программный генератор 2-х, 3-х и m зависимых переменных с заданными корреляционными структурами
		Программный генератор случайных последовательностей, содержащих тренды полиномиального типа
		Программный генератор случайных последовательностей, содержащих тренды колебательного типа
		Программный генератор нелинейных последовательностей, содержащие странные аттракторы
		Программный генератор негауссовских случайных последовательностей (распределения экспоненциальное, Лапласа, Коши, максимального и минимального значения, двойное показательное, логистическое и др.)
		Программный генератор засоренных случайных последовательностей (модели Хьюбера и Тьюки)
3.	Модуль анализа качества данных	Модуль дескриптивной статистики
		Модуль анализа трендов
		Модуль анализа распределения данных
4.	База данных реальных данных	
5.	Электронный (документальный) архив	

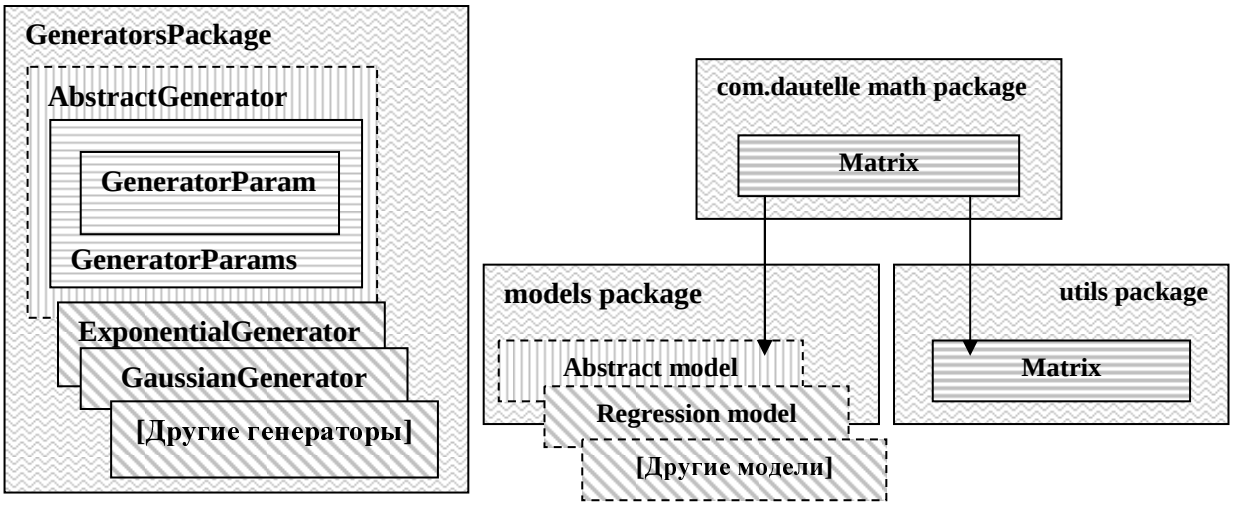

Рис. 2. Диаграмма основных классов стенда.

Труды СПИИРАН. 2005. Вып. 2. Т. 2. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2005. Issue 2. V. 2. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

Применение абстрактных классов позволило унифицировать коды классов-потомков, то есть стандартизовать их API, что является обязательным условием для обеспечения возможности расширения системы в дальнейшем.

Помимо базовых классов и их наследников, реализующих основные компоненты системы, были запрограммированы вспомогательные классы, предназначенные для взаимодействия этих компонент с пользователем через web-интерфейс и друг с другом, для обмена данными.

На рис. 2 показана упрощённая диаграмма классов испытательного стенда. Вертикальной штриховкой отмечены абстрактные классы, наклонной — классы-модели и генераторы исходных данных (набор этих классов может изменяться пользователями, в процессе работы с системой), горизонтальной — вспомогательные классы-утилиты. Области с волновой штриховкой характеризуют принадлежность классов к тем или иным пакетам.

4. Пакет генараторов случайных величин

Базовым классом в пакете генераторов исходных данных является класс AbstractGenerator. От которого наследуются содержательные классы, подгружаемые пользователем программного комплекса. В данном пункте приведена реализация класса AbstractGenerator (листинг 1) и генератор для показательного распределения.

В приведенном листинге поле desc содержит описание генератора на русском языке, которое отображается пользователю системы.

Поле data содержит результат работы генератора — матрицу (в частных случаях вектор или число).

Поле params предназначено для хранения параметров в случае параметрического генератора, например, параметр лямбда для показательного распределения.

Значение параметров задается пользователем системы при запуске генератора. Поле rnd содержит стандартный датчик случайных чисел из пакета java.util. Наконец, поля rows и cols задают размер генерируемой матрицы.

Метод generateOne() предназначен для получения следующего случайного значения. Методы generate() создают набор значений случайных величин. Конструктор AbstractGenerator() предназначен только для вызова в конструкторах наследуемых классов и является общей инициализирующей частью для всех генераторов.

Создавать экземпляры класса AbstractGenerator путем вызова данного конструктора невозможно, поскольку класс является абстрактным. Метод clear() очищает матрицу сгенерированных значений. Методы saveToDB() и saveToFile() позволяют сохранить результаты генерации, причем первый требует переопределения в классах-потомках, желающих сохранять данные в базе данных.

Методы setDesc() и getDesc() позволяют устанавливать и отображать описание класса генератора в пользовательском интерфейсе ПСИО ММ. Метод validateParams() позволяет контролировать значения параметров генератора, задаваемых пользователем.

Для реализации этого метода классы-потомки абстрактного класса GeneratorParams() должны реализовать метод isCorrect().

Пример класса, реализующего показательное распределение, приведен на листинге 2.

Труды СПИИРАН. 2005. Вып. 2. Т. 2. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2005. Issue 2. V. 2. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

5. Заключение. Применение результатов разработки

Программный комплекс является унифицированным изделием и пригоден для использования в любых предметных областях, связанных с разработкой и использованием математических моделей сложных динамических систем.

Элементы данного комплекса апробированы при построении математических моделей, используемых в задачах оптимизации управления технологическими процессами.

```
package generators;

import java.io.*;
import java.util.*;
import java.lang.IllegalArgumentException;
import com.dautelle.math.*;
import com.dautelle.xml.*;

abstract public class AbstractGenerator {

      protected static String desc = "";
      protected Matrix data;
      public GeneratorParams params;
      Random rnd;
      protected int rows;
      protected int cols;

      abstract public double generateOne();

      public Matrix generate() {
            return generate(1, 1);
      }

      public Matrix generate(int count) {
            return generate(1, count);
      }

      public Matrix generate(int rows, int cols) {
            this.rows = rows;
            this.cols = cols;
            for(int i = 0; i < rows; i++) {
                  for(int j = 0; j < rows; i++) {
                        data.set(i, j, Float64.valueOf(generateOne()));
                  }
            }
            return data;
      }

      public AbstractGenerator() {
            clear();
            params = new GeneratorParams();
            rnd = new Random(System.currentTimeMillis());
      }

      public void clear() {
            data.clear();
      }

      public void saveToDb() {
```


Труды СПИИРАН. 2005. Вып. 2. Т. 2. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2005. Issue 2. V. 2. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

```
      }

      public void saveToFile(String fileName) throws IOException {
        ObjectWriter writer = new ObjectWriter();
        writer.setNamespace("", "generators");
        File file = new File(fileName);
        writer.write(data, file);
      }

      protected void setDesc(String desc) {
            this.desc = desc;
      }

      public String getDesc() {
            return this.desc;
      }

      public boolean validateParams() {
            for(int i = 0; i < params.size(); i++) {
                  if(!(params.get(i).isCorrect()))
                        return false;
            }
            return true;
      }

}
```

Листинг 1. Реализация класса AbstractGenerator

```
package generators;

import java.util.*;
import java.lang.Math;
import com.dautelle.math.*;

public class ExponentialGenerator extends AbstractGenerator {

      public static String description = "показательное распределение";
      private Random rnd;

      public ExponentialGenerator() {
            params.add("lambda", 0, 0, 0, 100, "Лямбда");
      }

      public double generateOne() {
            return ((-1.0 / params.get("lambda")) * rnd.nextGaussian());
      }

}
```

Листинг 2. Реализация генератора для показательного распределения

Литература

[1] *Макшанов А. В., Мусаев А. А.* Формирование случайных чисел с заданным законом распределения //В кн.: Сборник алгоритмов и программ типовых задач. — ВИКИ им. А. Ф. Можайского, 1981, №5, с. 181–189.

[2] *Мусаев А. А. Устойчивые методы определения движения. – Л.: МО СССР, 1984. — 232 с.*