компьютер
в учебном
процессе

Колесов Юрий Борисович

УДК 681.3.06 (075.8)

ВЫЧИСЛИТЕЛЬНЫЕ ЭКСПЕРИМЕНТЫ В ПАКЕТЕ RMD

Аннотация

Статья посвящена автоматизации сложных вычислительных экспериментов, требующих многократного вычисления характеристик, зависящих от параметров. Среди вычислительных экспериментов можно выделить «типовые», часто проводимые исследователями и проектировщиками, методики которых достаточно хорошо обоснованы теоретически: статистические, стохастические, анализ чувствительности. Описаны типы экспериментов, которые удалось автоматизировать в среде визуального моделирования RandModelDesigner. Приведены примеры конкретных экспериментов.

Ключевые слова: объектно-ориентированное моделирование, среды визуального моделирования сложных динамических систем, вычислительный эксперимент.

ВВЕДЕНИЕ

Модели создаются с определенными целями. Можно выделить два «крайних» случая использования моделей. В первом случае встраиваемая модель используется многократно как элемент сложной технической системы в качестве имитатора какого-то реального объекта. Примерами таких систем являются тренажеры, испытательные стенды и т. п. Для встраиваемых моделей RMD обеспечивает генерацию готового программного модуля (DLL) для платформы «Intel-Windows», которым внешнее приложение может управлять через специальный интерфейс. Во втором случае модель используется однократно для анализа ее поведения в определенных условиях. Основной интерес для пользователя в этом случае представляет не сама модель, а результаты ее анализа.

В любом случае модель необходимо предварительно отладить, то есть проверить ее адекватность, найти и устранить ошибки. Отладка, как правило, сводится к воспроизведению набора сценариев, для которых хотя бы качественно известны правильные результаты. Для встраиваемой модели ее успешная работа на заданном контрольном наборе сценариев является критерием сдачи модели в эксплуатацию.

Реализация простого сценария и типовые «аналитические» вычислительные эксперименты (получение параметрической зависимости, определение вероятности события, определения математического ожидания и среднеквадратического отклонения величины, анализ глобальной чувствительности, нахождение оптимальных значений параметров) поддерживаются визуальной моделью RMD. Нестандартные вычислительные эксперименты требуют использования для создания плана эксперимента входного языка моделирования. Ниже на примерах показано использование всех этих способов.

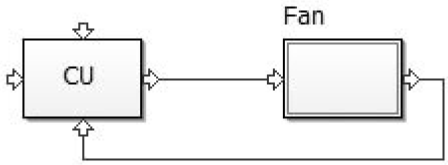

Рис. 1. Система регулирования скорости вращения вентилятора

РЕАЛИЗАЦИЯ ПРОСТОГО СЦЕНАРИЯ

Простым будем называть сценарий, задающий только изменения значений переменных модели в фиксированных временных точках. Предположим, что мы хотим изучить поведение системы регулирования скоростью вращения вентилятора (СРСВВ) при различных программных значениях угловой скорости и различных значениях коэффициента усиления регулятора Ke. Уберем из модели СРСВВ генератор программного сигнала, оставив собственно систему регулирования (рис. 1).

Запустим визуальную модель и с помощью команды главного меню «Сервис / План» создадим следующий план прогона модели (рис. 2).

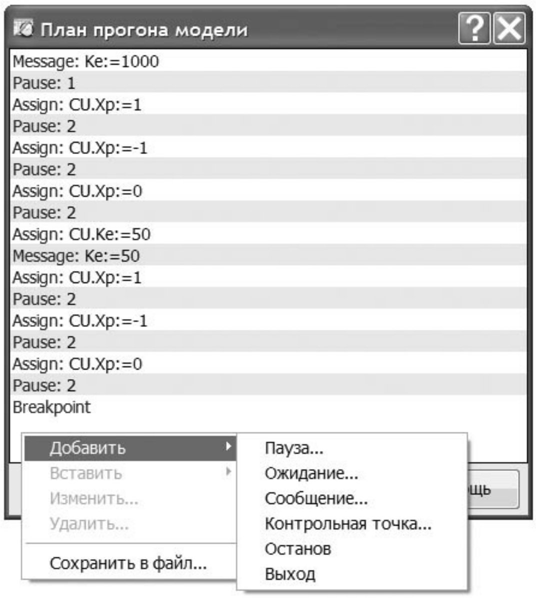

Рис. 2. План прогона модели СРСВВ

Запустив модель с этим планом, получим графики программного и реального значений скорости вращения вентилятора, показанные на рис. 3.

РЕАЛИЗАЦИЯ СЛОЖНОГО СЦЕНАРИЯ

Пусть мы хотим подавать на вход СРСВВ синусоидальное программное значение скорости вращения, уменьшая через 1 единицу времени значение коэффициента CU.Ke в два раза (начальное значение 100), а затем сделать то же самое для ступенчатых воздействий. Реализовать такой сценарий стандартными средствами плана прогона модели невозможно. Используем собственное поведение класса «Model». Для этого преобразуем класс «Model» в гибридный и создадим карту поведения, показанную на рис. 4. При переходе из начального состояния в состояние «Синус» устанавливается исходное значение CU.Ke = 100. В состоянии «Синус» выполняется деятельность – экземпляр непрерывного локального класса «Генерация_синусоиды», поведение которого задается уравнением

CU.Xp = sin((2*pi)/T*Time)

Срабатывающий с интервалом 1 переход обеспечивает уменьшение значения CU.Ke

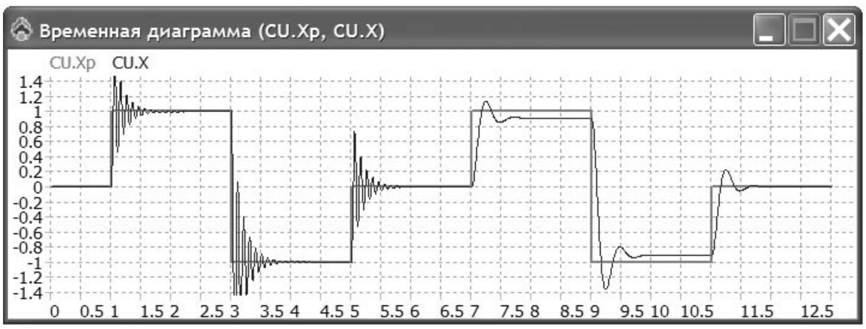

Рис. 3. Графики программного и реального значений скорости вращения вентилятора

в два раза. В формуле использовано глобальное время «Time» чтобы не допустить разрыва синусоиды при срабатывании этого перехода – значение локального времени «time» обнуляется при каждом входе в состояние.

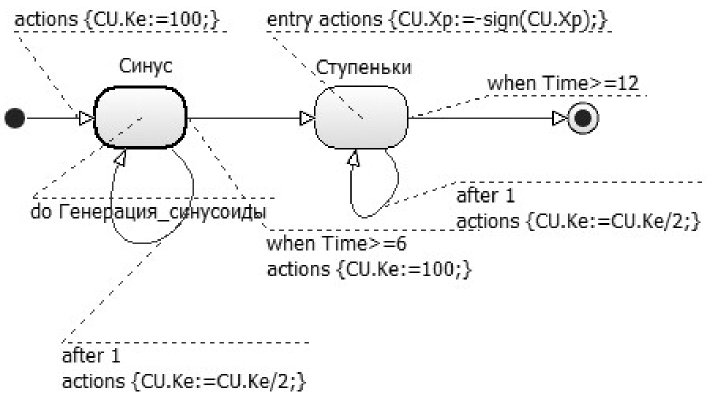

Рис. 4. Карта поведения класса «Model», реализующая сложный сценарий внешних воздействий

При переходе через 6 единиц времени из состояния «Синус» в состояние «Ступеньки» устанавливается исходное значение CU.Ke = 100. Заметим, что в условии срабатывания этого перехода не может использоваться событие «after 6», так как каждый раз при входе в состояние «Ступеньки» устанавливается значение входа CU.Xp, равное единице со знаком, обратным знаку предыдущего значения. С интервалом 1 срабатывает внешний переход, в действиях которого коэффициент CU.Ke уменьшается в два раза. Каждый раз при срабатывании этого перехода осуществляется новый вход в состояние «Ступеньки» и выполнение входных действий, изменяющих значение CU.Xp. Наконец, по истечении 12 единиц времени вычислительного эксперимента модель переходит в конечное состояние.

Запустив модель с этой картой поведения, получим графики программного и реального значений скорости вращения вентилятора, показанные на рис. 5.

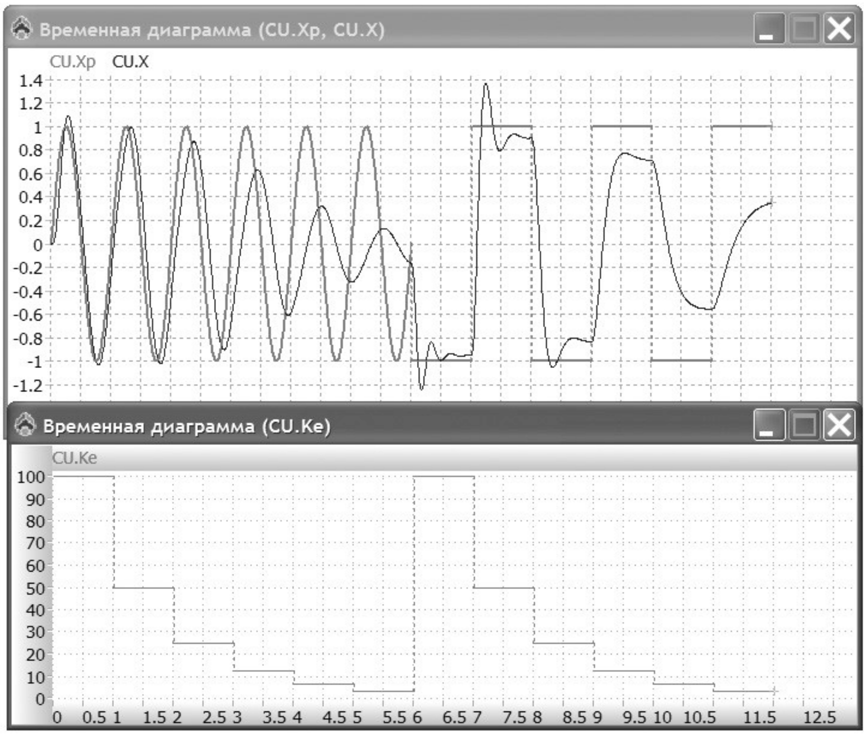

Рис. 5. Графики программного и реального значений скорости вращения вентилятора, а также коэффициента усиления регулятора при выполнении сложного плана вычислительного эксперимента

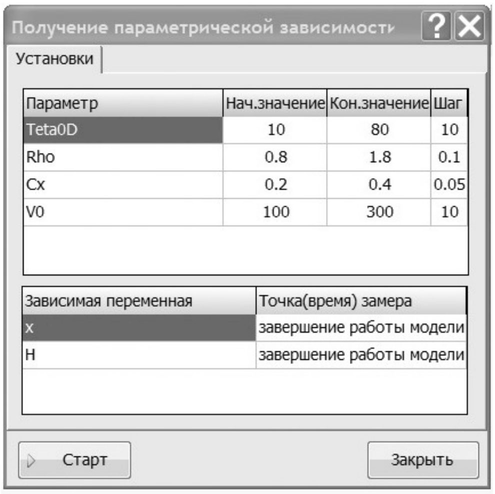

Рис. 6. Установки для вычислительного эксперимента «Получение параметрической зависимости»

ПОЛУЧЕНИЕ ПАРАМЕТРИЧЕСКОЙ ЗАВИСИМОСТИ

Получение зависимости какой-либо переменной модели от параметра – это один из наиболее распространенных вычислительных экспериментов. Пусть нам нужно найти зависимости дальности полета снаряда «x» и максимальной высоты траектории «H» (пример «Missile») от угла бросания в градусах «Teta0D», плотности воздуха «Rho», коэффициента аэродинамического сопротивления «Cx» и начальной скорости «V0».

Сначала решим эту задачу с помощью встроенных средств визуальной модели. Запустим визуальную модель и выполним команду главного меню «Моделирование / Типовой вычислительный эксперимент / Па-

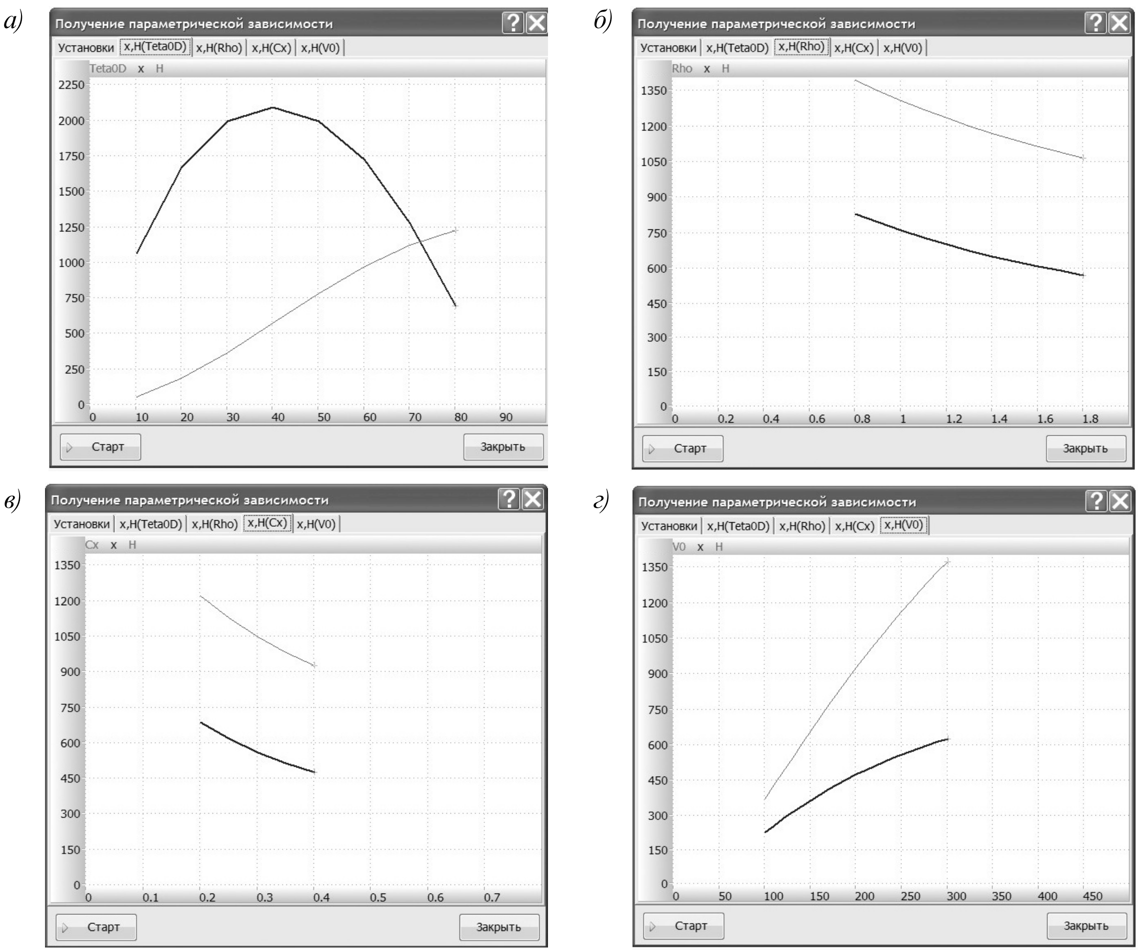

Рис. 7. Зависимости дальности полета (жирная линия) и высоты траектории снаряда от угла бросания (*а*), плотности воздуха (*б*), коэффициента аэродинамического сопротивления (*в*) и начальной скорости (*г*)

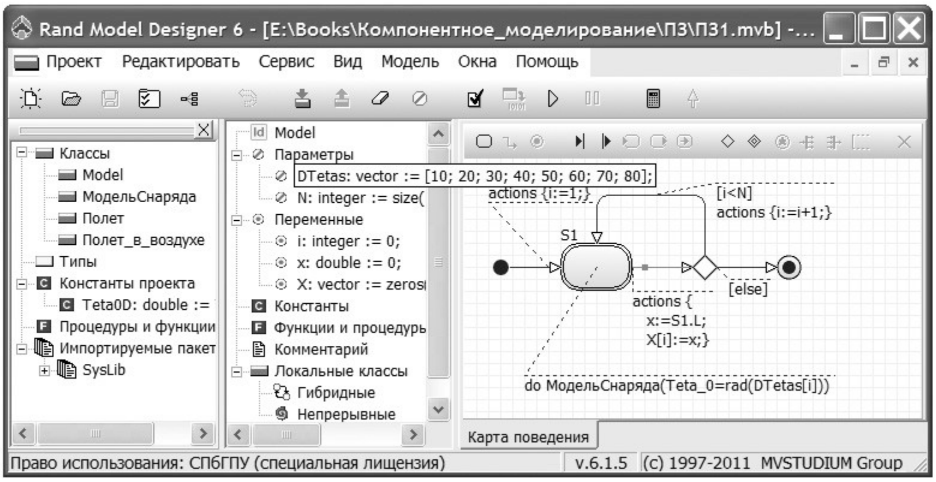

Рис. 8. Получение зависимости дальности полета снаряда от угла бросания – вариант 1

раметрическая зависимость…». На страничке «Установки» окна «Получение параметрической зависимости» (рис. 6) нужно задать:

– перечень варьируемых параметров с указанием начального и конечного значения параметра, а также шага изменения значения;

– перечень зависимых переменных с указанием точки замера значения (в данном случае по завершении работы модели).

Далее нажимаем кнопку «Старт» и получаем требуемые графики (рис. 7).

Попробуем теперь получить параметрическую зависимость дальности полета снаряда от угла бросания «вручную» – средствами входного языка моделирования. Сохраним модель снаряда как отдельный класс «МодельСнаряда», а поведение класса «Model» используем для задания плана эксперимента. Множество углов бросания определим в параметре «DTetas». Модель снаряда припишем состоянию «S1» в качестве локальной деятельности с действительным значением угла бросания «Teta_D0 = rad(DTetas[i])» (рис. 8). Безусловный переход из состояния «S1» в точку ветвления происходит по завершению карты поведения модели снаряда, то есть после падения снаряда. Дальность полета в *i*-м испытании запоминается в X[i].

Перебрав все значения вектора «DTetas», получим искомую зависимость (диаграмма X(DTetas)) плюс сами тестовые траектории (рис. 9).

В той же самой карте поведения локальную деятельность можно выполнять в независимом – «ортогональном» – времени (рис. 10). В этом случае время в модели будет продвигаться только дискретным образом. Непрерывное время будет инкапсулировано в объекте – экземпляре класса «МодельСнаряда» и каждый раз будет стартовать с нуля.

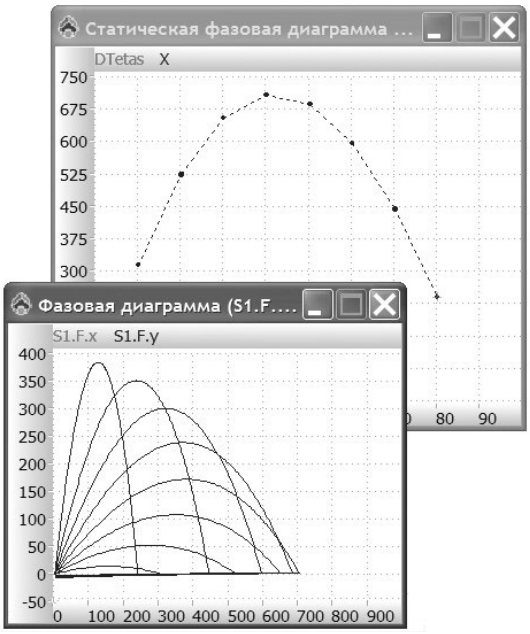

Рис. 9. Полученная зависимость дальности полета от угла бросания

Наконец, логику перебора множества углов бросания можно реализовать оператором цикла в выходных дискретных действиях (рис. 11).

Результаты для всех вариантов будут одинаковы.

«Ручное» задание плана вычислительного эксперимента имеет смысл использовать в случаях, когда типовые вычислительные эксперименты не вполне соответствуют решаемой задаче (например, желательно получить не только итоговую зависимость, но и все тестовые траектории).

СТОХАСТИЧЕСКИЕ ВЫЧИСЛИТЕЛЬНЫЕ ЭКСПЕРИМЕНТЫ

Пусть значения некоторых величин в исследуемой модели являются случайными. Для модели снаряда это могут быть угол бросания, начальная скорость, коэффициент аэродинамического сопротивления, параметры атмосферы и др. Для модели проекта – это длительности работ и затраты. Для экономических моделей это курсы валют, цены на энергоносители и др. Возможно также использование множества сценариев внешних воздействий с указанной вероятностью реализации каждого сценария. Предполагается, что нам известны законы распределения этих случайных величин. Целью исследования является нахождение закона распределения или его характеристик (математическое ожидание, дисперсия) некоторых конечных величин (дальности полета снаряда, времени выполнения проекта, валового регионального продукта и т. п.) или вероятности некоторого события (попадания в заданный диапазон дальности, выполнения проекта в срок, удвоения валового продукта и т. п.).

В тех случаях, когда построение аналитических зависимостей для определения вероятности некоторого события или математического ожидания некоторой случайной величины по той или иной причине затруднительно, применяется метод статистического имитационного моделирования (метод статистических испытаний, метод Монте-Карло) [1]. Метод предполагает «розыгрыш» случайных параметров исследуемой системы и получение с помощью вычислительного эксперимента множества «реализаций» фазовой траектории этой системы (испытаний), результаты которых обрабатываются обычными методами математической статистики. Предполагается наличие корректных генераторов случайных величин в инструментальных средствах моделирования.

Вероятность некоторого события p определяется как частота $\overline{P}$

$$p \approx \overline{P} = \frac{M}{N},$$

где N – число испытаний, M – число испытаний, в которых произошло данное событие.

Вероятность того, что вероятность p и частота $\overline{P}$ различаются не более чем на ε, определяется по формуле [5]

$$P(|\overline{P} = p| < \varepsilon) = 2\Phi\left(\frac{\varepsilon\sqrt{N}}{\sqrt{p(1-p)}}\right),$$

где $\Phi(x) = \frac{1}{2\pi}\int_0^x e^{-\frac{t^2}{2}} d$ – функция Лапласа.

Подставляя вместо p в ходе эксперимента ее ориентировочное значение или ча-

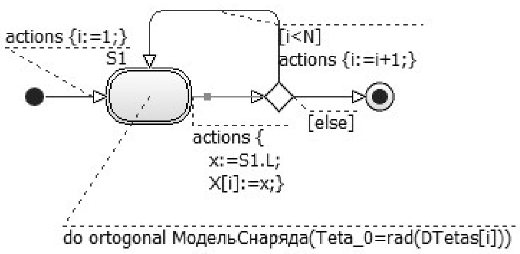

Рис. 10. Получение зависимости дальности полета снаряда от угла бросания – вариант 2

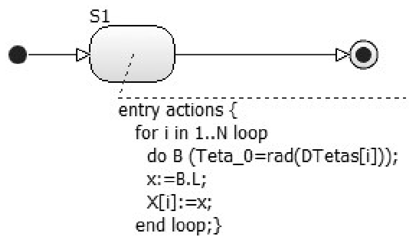

Рис. 11. Получение зависимости дальности полета снаряда от угла бросания – вариант 3

стоту, можно прекращать вычислительный эксперимент, когда вычисленная вероятность P превысит заданную величину.

Математическое ожидание m_x случайной величины x приближенно определяется как среднее арифметическое $\overline{X}$

$$m_x \approx \overline{X} = \frac{\sum_{i=1}^{N} x_i}{N},$$

где x_i – значение величины в i-м испытании.

Вероятность того, что математическое ожидание и частота отличаются не более чем на ε определяется по формуле [5]

$$P(|\,\overline{X} - m_x\,| < \varepsilon) = 2\Phi(\frac{\varepsilon\sqrt{N}}{\sigma_x}),$$

где $\sigma_x \approx \sqrt{\frac{1}{N}\sum_{i=1}^{N} x_i^2 - \overline{X}^2}$ – среднеквадратическое отклонение величины x. Таким образом, вычислительный эксперимент можно прекращать, когда вероятность P превысит заданную величину. Величина N может быть также известна заранее из некоторых содержательных соображений.

Исследуем модель снаряда в предположении, что угол бросания «Teta0D», плотность воздуха «Rho», коэффициент аэродинамического сопротивления «Cx» и начальная скорость «V0» являются случайными величинами. Попробуем найти вероятность падения снаряда на интервале дальности 2050..2100 м, а также математическое ожидание и среднеквадратическое отклонение дальности полета снаряда. Воспользуемся встроенными возможностями визуальной модели (команда «Моделирование / Типовой вычислительный эксперимент»).

Если законы распределения случайных параметров не заданы в самой модели, как в нашем случае, то их нужно задать в настройках вычислительного эксперимента (рис. 12, 13). Кроме того, для определения вероятности события необходимо указать логическое выражение, определяющее, произошло событие или нет, точку вычисления этого предиката, абсолютную погрешность вычисления вероятности события, а также требуемое значение вероятности (рис. 12).

Для определения математического ожидания и среднеквадратического отклонения нужно указать соответствующие переменные модели, точки их замера, абсолютные погрешности, а также требуемое значение вероятности P (рис. 13). В результате этого

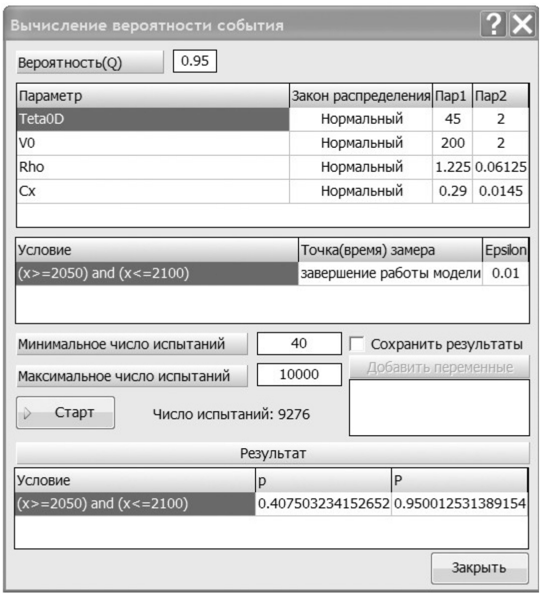

Рис. 12. Окно вычислительного эксперимента для определения вероятности падения снаряда на интервале 2050..2100 метров

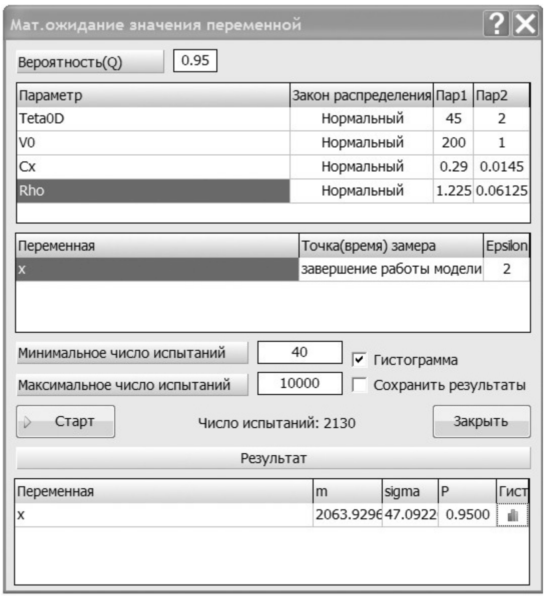

Рис. 13. Окно вычислительного эксперимента для определения математического ожидания и среднеквадратического отклонения дальности полета снаряд

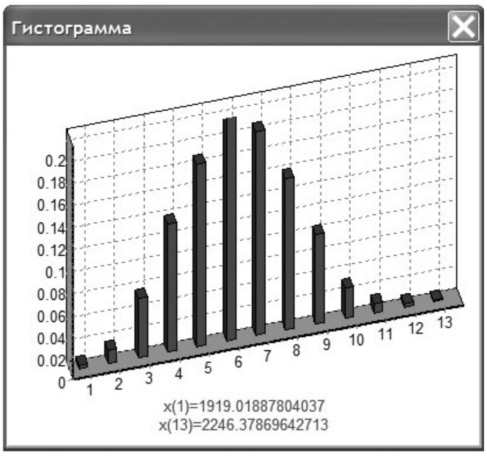

Рис. 14. Частотная гистограмма дальности полета снаряда

эксперимента можно также получить частотную гистограмму для каждой величины (рис. 14).

Вычислительный эксперимент автоматически останавливается при достижении вероятностью *P* требуемого значения или при превышении максимального числа испытаний. В результате проведенных экспериментов мы определили, что вероятность снаряда на интервале дальности 2050..2100 м с вероятностью 0.95 равна 0.40 .. 0.42, математическое ожидание дальности падения с вероятностью 0.95 равно 2062 .. 2066 м, а среднеквадратическое отклонение 45..49 м.

Результаты испытаний (все фазовые траектории модели) можно сохранить для последующей обработки специальными статистическими программами.

Попробуем теперь решить ту же задачу средствами входного языка моделирования. Этот способ актуален для проведения нестандартных вычислительных экспериментов (например, случайные параметры не являются независимыми), а также для тех случаев, когда в описании поведения самой модели некоторые величины определяются методом Монте-Карло (то есть проводится «локальный» стохастический эксперимент). В этом случае план эксперимента нужно задавать в карте поведения какого-то гибридного класса, например класса «Model». В стандартном пакете «Statistics» имеются абстрактные классы – заготовки планов стохастических экспериментов.

Гибридный класс «StochasticExp_mx» является заготовкой плана стохастического эксперимента по определению математического ожидания и среднеквадратического отклонения некоторой величины (рис. 15). Экземпляр этого класса имеет следующие параметры: (см. листинг 1).

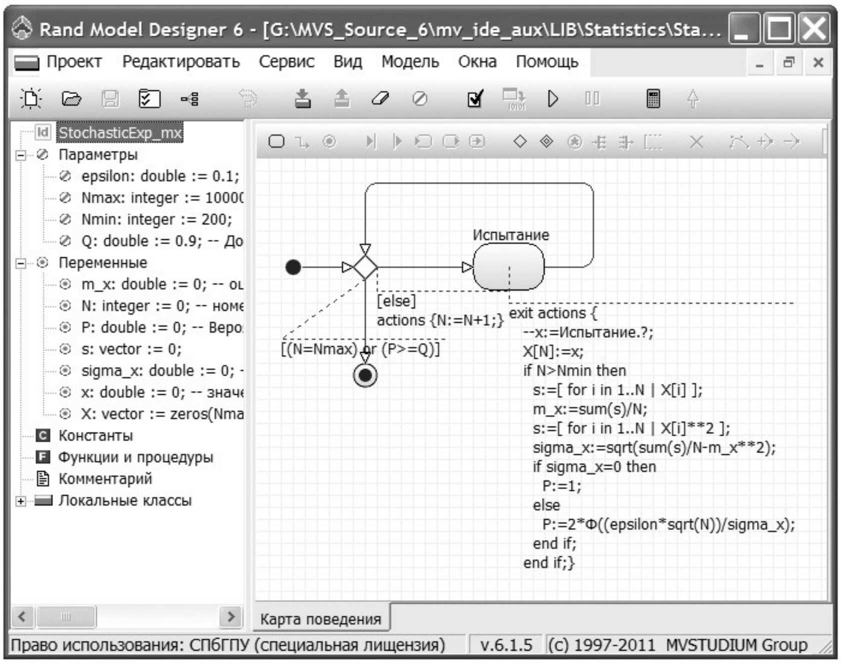

Рис. 15.

Листинг 1

```
parameterNmax: integer := 10000; — предельное число испытаний
parameterepsilon: double := 0.1; — абсолютная погрешность
parameterQ: double := 0.9; — допустимая вероятность определения m_x с
погрешностью epsilon
parameterNmin: integer := 200; — минимальное число испытаний
Переменные:
m_x: double := 0; — оценка математического ожидания величины x
N: integer := 0; — номер текущего испытания
P: double := 0; — Вероятность определения m_x с погрешностью меньшей
чем epsilon
sigma_x: double := 0; — среднеквадратическое отклонение
x: double := 0; — значение искомой величины в N-м испытании
X: vector := zeros(Nmax); — вектор значений величины x в N испытаниях
```

Для планирования конкретного стохастического эксперимента нужно определить класс «Model» как производный от «Statistics. StochasticExp_mx», приписать состоянию «Испытание» в качестве локальной деятельности экземпляр исследуемой модели с указанием нужных случайных значений и вместо комментария в выходных действиях состояния «Испытание» указать нужную переменную модели.

В качестве примера рассмотрим задачу определения математического ожидания и среднеквадратического отклонения дальности полета снаряда при случайных значениях угла бросания и начальной скорости, распределенных по нормальному закону. В классе «Model», производном от «Statistics. StochasticExp_mx» состоянию «Испытание», в качестве локальной деятельности приписан объект класса «МодельСнаряда» с действительными значениями параметров «Teta_0» и «V_0», распределенных по нормальному закону (рис. 16).

В переопределенных выходных действиях состояния «Испытание» дополнительно переменной «x» присваивается дальность полета снаряда, достигнутая в конкретном испытании, а также вызывается процедура

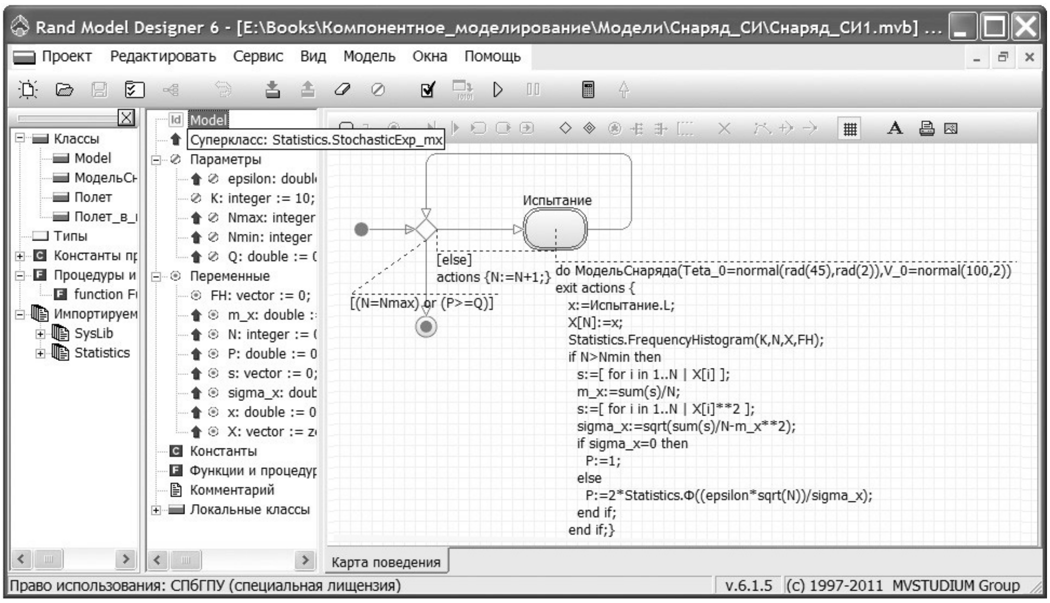

Рис. 16. План стохастического эксперимента
по определению математического ожидания дальности полета снаряда

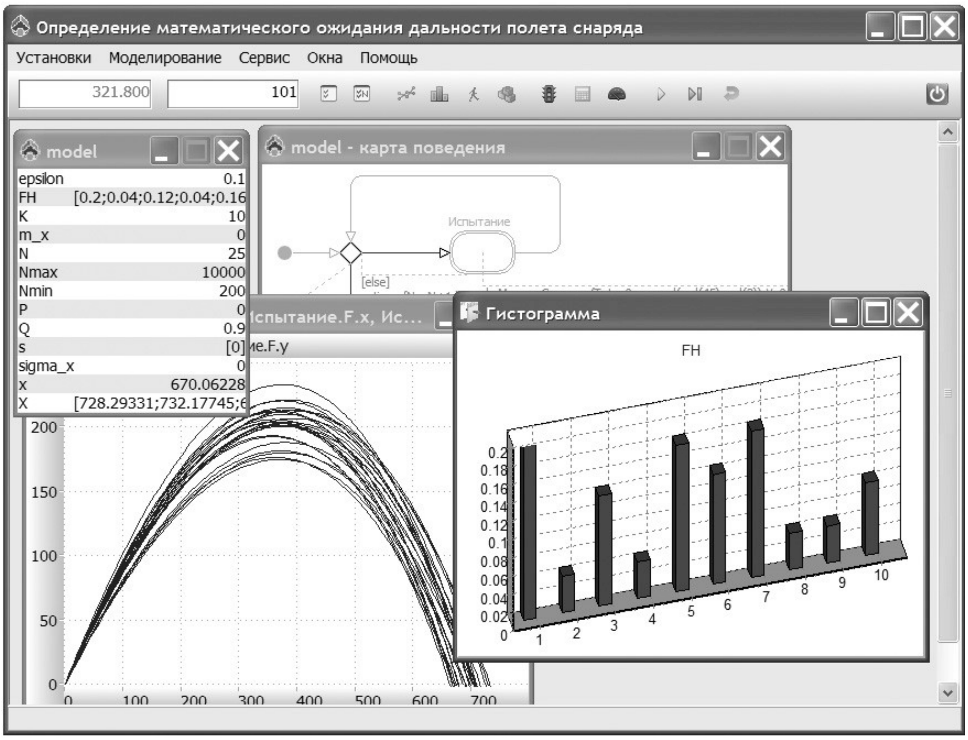

Рис. 17. Вычислительный эксперимент по определению математического ожидания дальности полета снаряда с заданием плана эксперимента в поведении класса «Model»

формирования частотной гистограммы из пакета «Statistics».

Запустив визуальную модель, можно увидеть в динамике формирование закона распределения на частотной гистограмме и трубку случайных траекторий (рис. 17). Для ускорения вычислений после отладки лучше отключить вычисление частотной гистограммы и показ траекторий, а также выполнять модель снаряда в ортогональном времени.

Гибридный абстрактный класс «StochasticExp_p» является заготовкой плана стохастического эксперимента по определению вероятности некоторого события (рис. 18). Параметры в этом классе аналогичны классу «StochasticExp_mx», а переменные имеют следующий смысл:

n: double := 0; – число испытаний, в которых цель достигнута,

N: integer := 0; – номер текущего испытания,

p: double := 0; – вероятность достижения цели,

P: double := 0; – вероятность определения p с погрешностью, меньшей чем epsilon,

ЦельДостигнута: boolean := false; – флаг достижения цели в текущем испытании.

Для планирования конкретного стохастического эксперимента нужно определить класс «Model» как производный от «Statistics.StochasticExp_p», приписать состоянию «Испытание» в качестве локальной деятельности экземпляр исследуемой модели с указанием нужных случайных значений и вместо комментария в выходных действиях состояния «Испытание» указать оператор, вычисляющий флаг достижения цели по значениям переменных модели.

В качестве иллюстрации использования этого класса рассмотрим известную задачу определения числа π методом Монте-Карло. Для этого выбрасываются случайные точки внутри некоторого квадрата и ищется вероятность попадания во вписанный в этот квадрат круг. Полагая, что эта вероятность p равна отношению площадей круга и квадрата, получаем, что $\frac{\pi R^2}{(2R)^2} = p$ и $\pi = 4p$.

Создадим проект, в котором импортируем пакет «Statistics» и определим класс «Model» как производный от «Statistics.StochasticExp_p» (рис. 19).

Добавим определения нужных переменных. Переопределим комментарий класса. В данном случае локальная деятельность сводится лишь к «разыгрыванию» случайных значений координат «x» и «y», поэтому поместим эти действия просто во входные действия состояния «Испытание». В выходных действиях переопределим оператор вычисления значения унаследованной переменной «ЦельДостигнута». Наконец, в действиях перехода в конечное состояние поместим операторы вычисления оценки значения числа π (рис. 19). Зададим действительные значения параметров модели (epsilon=0.01, Nmax=20000, Nmin=200, Q=0.99). Запустим визуальную модель.

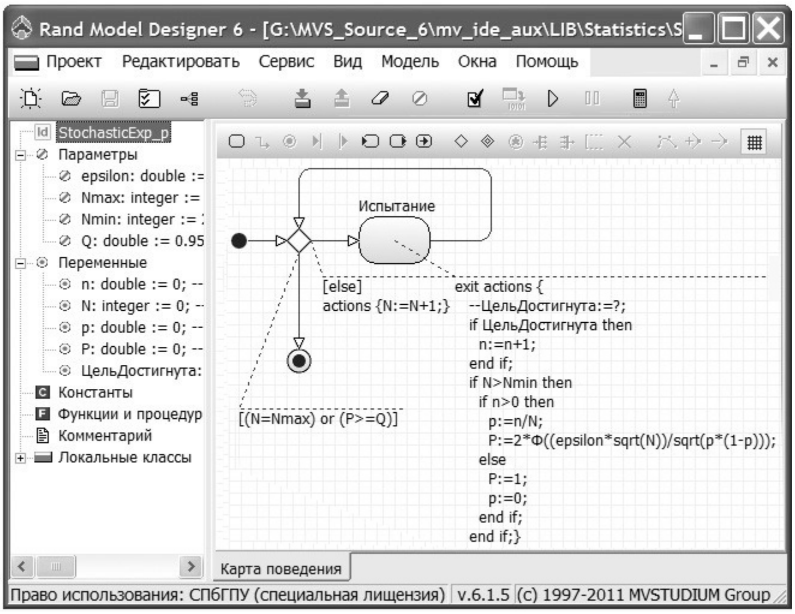

Рис. 18. Класс – заготовка плана стохастического эксперимента по определению вероятности некоторого события

Закроем окно карты поведения, так как оно сильно тормозит работу модели, и окно диаграммы за ненадобностью. Добавим для

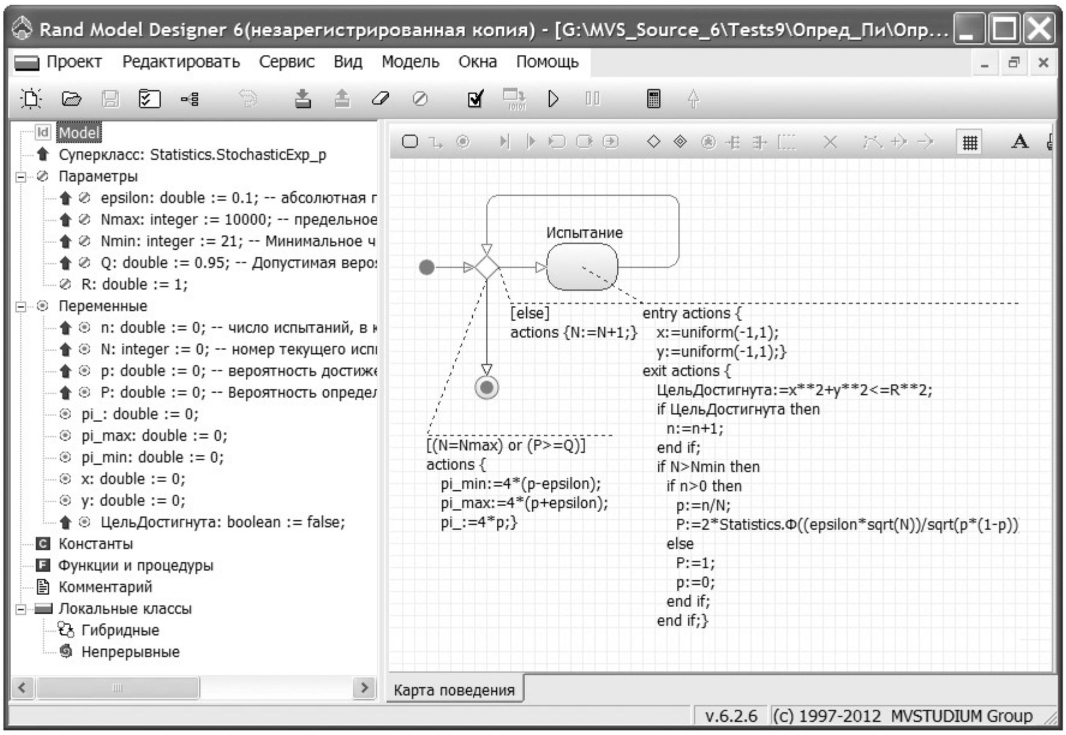

Рис. 19. План стохастического эксперимента по определению значения числа

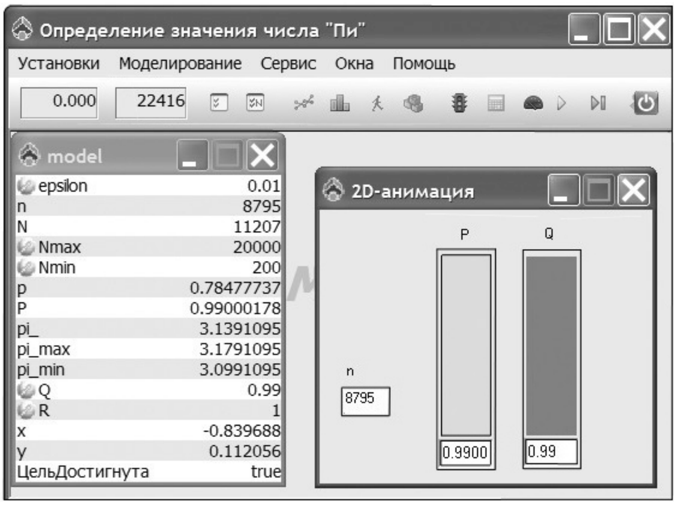

Рис. 20. Результаты эксперимента по оценке значения числа π

наглядности окно 2D-анимации, на котором поместим линейные индикаторы, отображающие значения переменных «P» и «Q», а также цифровой индикатор для переменной «n». Эксперимент завершится, когда столбики индикаторов станут одной высоты. Получаем оценку $\pi = 3.10..3.18$ (рис. 20).

ОПТИМИЗАЦИЯ

Целью оптимизационного вычислительного эксперимента является нахождение совокупности оптимальных значений выбранных параметров модели, при которых значение заданной целевой функции максимально или минимально с учетом ограничений. Ограничения могут быть заданы в виде допустимых диапазонов значений оптимизируемых параметров, а также в виде соотношений между этими параметрами – равенств и неравенств.

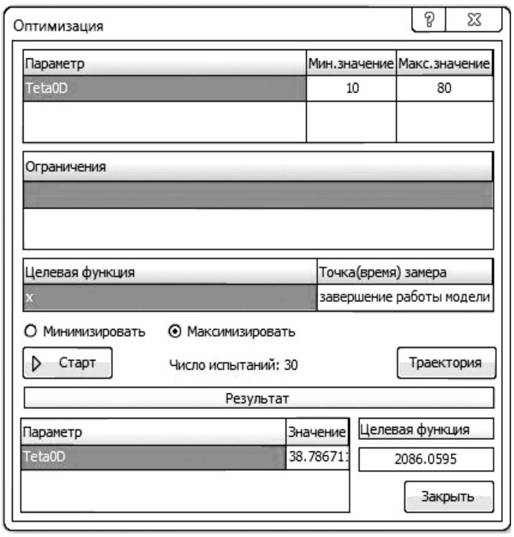

Рис. 21. Окно вычислительного эксперимента по нахождению угла бросания максимальной дальности

Рассмотрим задачу определения оптимального угла бросания снаряда, при котором дальность его полета максимальна. Сначала решим эту задачу стандартными средствами визуальной модели. Откроем окно типового вычислительного эксперимента «Оптимизация» и укажем в нем (рис. 21):

– параметр «Teta0D» как параметр оптимизации с диапазоном изменения 10..80 градусов;

– выражение для целевой функции – в данном случае это переменная «x» – с указанием момента вычисления – в данном случае по окончании прогона модели.

Нажимаем кнопку «Старт» и получаем угол бросания максимальной дальности при движении в воздухе – 38.8 градуса.

Если мы теперь попробуем найти угол бросания, при котором модуль скорости «V» максимален или минимален на 3-й секунде полета, то получим значение «Teta9D» 10 градусов для первого случая и 80 градусов для второго (рис. 22), то есть на границах допустимого диапазона значений.

Бывают модели, в которых решение задач оптимизации входит в определение функциональности модели.

Оптимизацию можно провести и средствами входного языка. Это необходимо в тех случаях, когда функционирование самой модели включает в себя определение максимального или минимального значения некоторой величины. Рассмотрим этот способ на том же примере определения угла бросания максимальной дальности.

Сохраним класс «Model» как класс «Модель_снаряда». Переменную «x» сделаем внешней – выходом. Экземпляр этого клас-

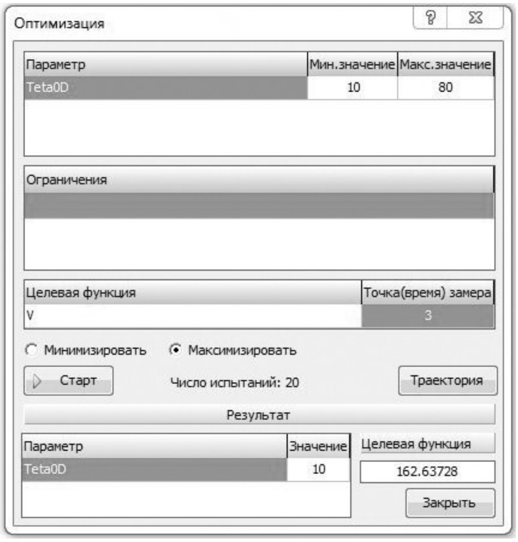

Рис. 22. Окно вычислительного эксперимента по определению угла бросания максимальной скорости

са – объект «Снаряд» – поместим в структуру класса «Model» (рис. 24). В дискретных действиях класса «Model» поместим оператор поиска максимального значения целевой функции «Дальность» по переменной «teta» (рис. 23).

Если бы снаряд летел в безвоздушном пространстве, то для целевой функции можно было бы написать формулу, вычисляющую результат ($\dfrac{2\cdot V_0^2\cdot\sin\theta_0\cdot\cos\theta_0}{g}$). Но в нашем случае дальность полета может быть найдена только численно. Следовательно, определение целевой функции должно включать в себя «эксперимент в эксперименте». Для этого используем оператор выполнения подмодели «do» (рис. 25).

Каждый раз при вычислении целевой функции данный оператор запускает в ортогональном времени модель движения снаряда в воздухе с углом бросания «teta0» градусов. Выполнение оператора завершается после падения снаряда. При этом значение

Рис. 23. Оптимизация средствами входного языка – карта поведения

времени самой модели не изменяется. Значения переменных объекта «Снаряд» после выполнения оператора соответствуют моменту падения снаряда. Значение видимой извне переменной «x» возвращается как значение целевой функции. Запустив модель, получим тот же оптимальный угол – 38.8 градусов.

АНАЛИЗ ГЛОБАЛЬНОЙ ЧУВСТВИТЕЛЬНОСТИ

Анализ глобальной чувствительности – это ранжирование варьируемых параметров

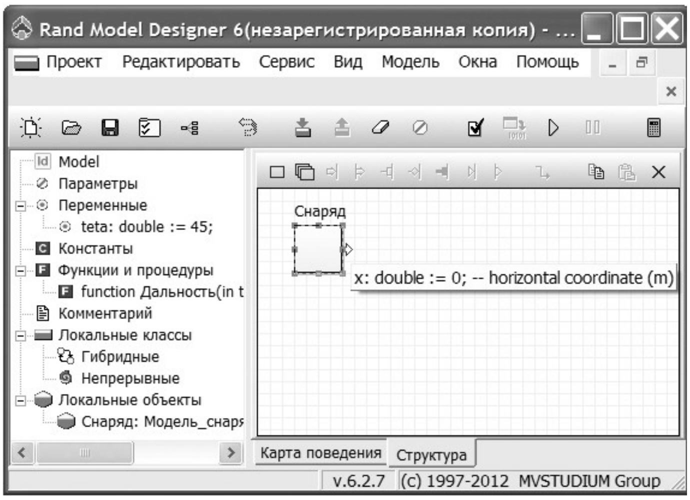

Рис. 24. Оптимизация средствами входного языка – структура

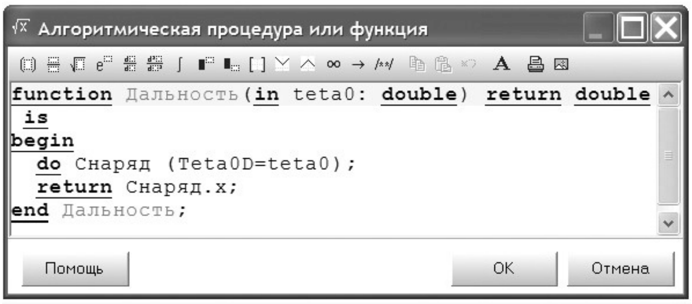

Рис. 25. Оптимизация средствами входного языка – целевая функция

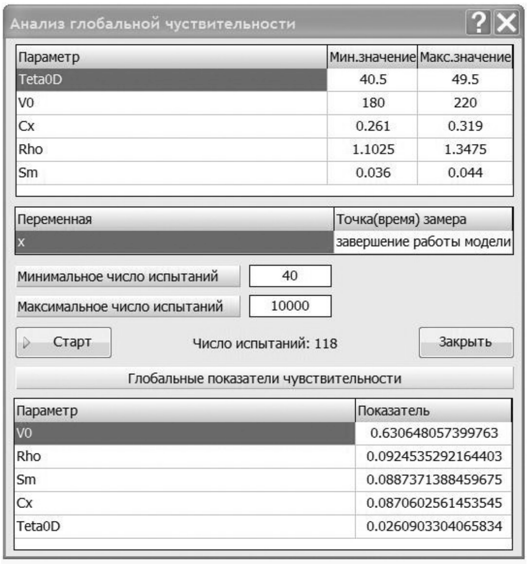

Рис. 26. Окно вычислительного эксперимента по анализу глобальной чувствительности

модели по степени влияния на значение некоторой выходной переменной модели при независимом изменении всех варьируемых параметров в заданных диапазонах. Анализ глобальной чувствительности позволяет, например, определить, неточность каких исходных данных в наибольшей степени влияет на результаты моделирования. При решении этой задачи определяются так называемые глобальные показатели чувствительности (И.М. Соболь. Глобальные показатели чувствительности для изучения нелинейных математических моделей // Матем. моделирование. 2005, Т. 17. № 9. С.43–52) для каждого варьируемого параметра, и эти параметры упорядочиваются по значению этого показателя. Для определения глобальных показателей чувствительности выполняется специальный стохастический вычислительный эксперимент, в плане которого использован алгоритм, разработанный И.М. Соболем.

Решим задачу по определению степени влияния разбросов значений угла бросания, начальной скорости, коэффициента аэродинамического сопротивления, плотности воздуха и миделева сечения снаряда на величину дальности полета снаряда. Для этого откроем окно соответствующего типового эксперимента (рис. 26) и зададим:

– перечень варьируемых параметров модели с указанием диапазонов их изменения;

– выходную переменную модели.

После нажатия кнопки «Старт» начинают вычисляться глобальные показатели чувствительности, одновременно параметры упорядочиваются по значениям этих показателей. В данном случае оказывается, что влияние начальной скорости на порядок превосходит влияние всех остальных параметров.

Литература

1. *Вентцель Е.С.* Исследование операций. М.: «Советское радио», 1972.

Abstract

Automation of computing experiments demanding recurrent executable model run for building parametric dependences is considered. Typical experiments such as statistical and stochastic experiments, sensitivity analysis are high-usage tools by researchers and engineers. These types of experiments were automatized in visual environment for modeling and simulation complex dynamical systems under the name Rand Model Designer. The article is illustrated by concrete examples.

Keywords: object-oriented modeling, visual environment for modeling and simulation complex dynamical systems, computer experiment.

Колесов Юрий Борисович,
доктор технических наук,
главный научный сотрудник
ФГУП «СП-Центр»,
ybk@mail.ru