# Тема 4 (дополнения) АЛГОРИТМЫ ИМИТАЦИИ МНОГОМЕРНЫХ СЛУЧАЙНЫХ ВОЗДЕЙСТВИЙ

## 3.1. Имитация многомерных случайных величин

Рассмотренные в ранее известные методы формирования многомерных случайных величин не оптимальны с точки зрения возможности создания эффективного алгоритма, пригодного для использования как в программных моделях, так и для реализации аппаратными методами. Поэтому, для имитации многомерных случайных величин целесообразно применять алгоритмы формирования одномерных случайных величин, использующие один случайный процесс и допускающие параллельное формирование разрядов случайной величины. В связи с этим возникает задача перехода от многомерной функции распределения вероятностей к эквивалентной ей одномерной функции распределения и последующем преобразовании значений одномерной случайной величины в значения составляющих случайного вектора.

Эквивалентными будем называть две такие функции распределения вероятностей *F*(*x*1, …, *xn*) и *F*\*(*y*) случайных величин *Ẍ* и *Y*, значения которых равны на взаимно соответствующих значениях аргументов *ẍ* и *y*.

Для решения задачи перехода от многомерной функции распределения *F*(*x*1, …, *xn*) к эквивалентной одномерной функции *F*\*(*y*) рассмотрим представление математических объектов. Тензорным называется математические объекты, числовая запись которых зависит от системы координат по определенному закону. Тензоры ранга ноль (абсолютным скаляром, скаляром) называется объект, который в координатной системе x определяется функцией α(*x*1, …, *xn*), а в координатной системе y -- функцией α\*(*y*1, …, *yn*), связанной с α(*x*1, …, *xn*) в каждой точке пространства соотношением α\*=α [18].

Пример такого преобразования для двумерной функции распределения представлено на рис. 4д.1.



Рисунок 4д.1. Преобразование двумерного распределения в эквивалентное одномерное распределение

Инвариантное преобразование позволяет перейти к эквивалентной одномерной функции распределения вероятностей и использовать при формировании случайной величины алгоритм имитации одномерных случайных величин, обеспечивающее высокое быстродействие и минимальное количество исходных факторов. При тензорном представлении объектов преобразование координат должно осуществляться таким образом, чтобы имело место взаимно однозначное соответствие между различными системами координат.

В дальнейшем предполагается, если специально не оговорено иное, что многомерные случайные величины определены на единичном пространстве.

Таким образом, рассмотренное представление многомерной случайной величины позволяет:

* При формировании многомерных случайных величин использовать алгоритмы формирования многомерных случайных величин за счет предварительного преобразования многомерной функции распределения вероятностей в эквивалентную ей одномерную функцию; при этом необходимое быстродействие обеспечивается выбором соответствующего алгоритма формирования одномерных случайных величин;
* Использовать один исходный случайный процесс для имитации многомерных случайных величин;
* Выполнять преобразование эквивалентной одномерной случайной величины Y в многомерную случайную величину Ẍ путем разбиения двоичного числа на группы разрядов без аппаратурных и временных затрат.

Из вышеизложенного следует сделать вывод о возможности разработки алгоритмов формирования многомерных случайных величин, ориентированных на использование в системах имитационного моделирования.

4д.2. Алгоритм поразрядного формирования многомерных случайных величин.

Преимущества рассмотренного представления многомерного закона распределения случайной величины реализуется лишь при использовании быстродействующего алгоритма формирования одномерной случайной величины. Наиболее распространенный алгоритм формирования – алгоритм последовательной выборки, описываемый рекуррентным соотношением [19]

(3.15)

где *yik* – сформированное значение одномерной случайной величины;

*P*(*y*j) (*j*=1, …, *m*) – вероятности соответствующих значений случайной величины Y;

*ξik* – равномерно распределенное число, сформированное в k-том испытании.

Широкое распространение этого алгоритма объясняется простотой его реализации и простой подготовки исходных данных *P*(*yj*) (*j*=1, …, *m*). Недостатком алгоритма последовательной выборки является относительно низкое быстродействие, определяемое величиной *n*1 среднего числа циклов сравнения (3.15):

(3.16)

К недостаткам алгоритма следует также отнести асинхронный (непериодический) характер формирования значений случайной величины.

Отмеченные недостатки могут быть преодолены при использовании алгоритма интервальной выборки, описанного в работах [20].

Сущность алгоритма заключается в том, что условие

) (4д.1)

проверяется для специально упорядоченных значений *F*(*yl*). В этом случае обеспечивается синхронный характер формирования значений случайной величины Y и высокое быстродействие, определяемое числом шагов формирования, равным разрядности значений случайной величины *Y*.

Исходная информация для алгоритма поразрядного формирования многомерных случайных величин представляет собой значения эквивалентной функции распределения вероятностей, размещаемой в ячейки памяти таким образом, что проверка условия (4д.1) позволяет определить, какому интервалу значений относится формируемое значение случайной величины *Y*. Получаемый на очередном шаге процесса формирования код *L* значения случайной величины определяет адрес значения эквивалентной функции распределения *F*(*y*) для проверки условия (3.17) в следующем шаге формирования.

На рис. 3.3 приведена диаграмма процесса формирования кода случайной величины Y. В вершинах диаграммы записаны коды случайной величины и значения эквивалентной функции распределения *F*(*y*), используемые на данном шаге формирования. Знаком «\*» обозначены еще не сформированные разряды кода. Аргумент эквивалентной функции распределения *F*(*y*) рассматривается как двоичное целое число I, номер интервала.



Рисунок 4д.2. Диаграмма процесса формирования кода случайного числа по алгоритму интервальной выборки

Схема алгоритма поразрядного формирования многомерных случайных величин с заданной многомерной функцией распределения вероятностей, осонованного на алгоритме интервальной выборки, приведен на рис. 3.4.

НАЧАЛО

Формирование массивов

А

Формирование

равновероятного числа

Вычисление

Выборка сформирована

КОНЕЦ

А

А

3

4

5

6

7

8

9

нет

да

нет

да

нет

да

Рисунок 3.4. Схема алгоритма поразрядного формирования многомерных случайных величин

Имитация многомерных случайных величин с использованием алгоритма поразрядного формирования осуществляется в следующей последовательности.

1. Формирование массива модифицированной функции распределения F\*(L). Исходные данные для рассматриваемого алгоритма представляют собой многомерной случайной величины *Ẍ*={*X*1, …, *XN*} *p*(*x*1, …,*xN*); *i* – ая составляющая *Xi* задается *ri* – разряднымым двоичным кодом. При этом имеет место соотношение

(3.23)

В блоке 2 (рис. 3.4) выполняется преобразование многомерной плотности распределения *p*(*x*1, …,*xN*) в эквивалентную одномерную функцию распределения *F*(*y*), а затем *F*(*y*) преобразуется в модифицированную функцию *F*\*(*L*). Преобразование функции *p*(*x*1, …,*xN*) в функцию *F*(*y*) осуществляется в соответствии с (3.1), (3.2), (3.6) следующим образом:

(3.24)

Перейдем от двоичных дробей y к целым двоичным числам *I* согласно (3.28).

Под модифицированной функцией распределения *F\**(*L*) будем понимать такую функцию, описывающую распределение случайной величины *Y*, аргумент которой *L* на шаге моделирования *H* совпадает с текущим кодом случайной величины Y, полученным на предыдущем шаге *H*-1. При реализации алгоритма интервальной выборки текущий код случайной величины *Y* получается из начального кода 00…00 путем занесения в соответствующий разряд цифры, полученной на данном шаге формирования (блок 5, рис.3.4).

Аргумент функции *F*(*L*) и аргумент модифицированной функции *F*\*(*L*) связаны между собой соотношением:

(3.25)

Возможные значения аргумента L на шаге формирования *H* принадлежит множеству

(3.26)

2. Проверяется, сформирован ли заданный объем M выборки многомерной случайной величины *Ẍ*. Если заданный объем сформирован, ты моделирование заканчивается, если фактический объем меньше заданного, то процедура формирования числа повторяется.

3. Формируется равномерно распределенное число *ηi*€[0,1[. Поскольку процедуры формирования равновероятных случайных чисел исследованы достаточно полно, то этот вопрос специально не рассматривается.

4. Проверяется выполнение неравенства ηH≤*F*(*L*H). Таким образом проверяется, в какой интервал попадает формируемое значение случайной величины *Y*. Если данное неравенство выполняется, то формируемое значение Y относится к интервалу *I*≥I*H*=φ(*L,H*), и осуществляется переход к блоку 5.

5. Формируется очередной разряд значения многомерной случайной величины *Y*. Формирование реализуется путем занесения случайной величины *Y* . Формирование реализуется путем занесения в разряд с весом 0 или 1 в зависимости от результатов анализа, выполненного в блоке 4.

6. Проверяется, сформировано ли заданное количество разрядов кода случайной величины Y. Если формирование не закончено , то выполняется переход к блоку 3 и преобразования блоков 3,4,5,6 повторяются. Если , т.е. Y сформирован, то выполняется переход к блоку

7. На основе кода Y формируются коды составляющих X1, …, XN. В соответствии с (3.11) выполняются преобразования

(3.27)

Полученные целые двоичные числа *J1,J2,…,JN* связаны с составляющими соотношениями

|  |  |
| --- | --- |
|  | (3.28) |

аналогичными соотношению (3.18).

При необходимости полученные коды *J1, J2,…,JN* могут быть преобразованы (приведены в соответствующий диапазон значений) (блок 8). После этого выполняется анализ объёма сформированной выборки (блок 9).

Определим основные характеристики алгоритма поразрядного формирования многомерных случайных величин.

Из приведенных графиков следует, что наибольшим быстродействием (минимальным временем *T*) характеризуется алгоритм поразрядного формирования, наиболее низким быстродействие обладает алгоритм Неймана.

|  |
| --- |
|  |
| Рисунок 3.5. Зависимость времени формирования значений многомерной случайной величины от её размерности |

## 3.3.Алгоритм k-разрядного формирования случайных величин

Анализ алгоритма поразрядного формирования многомерных случайных величин показывает, что процедура формирования может быть ускорена, если на каждом шаге формирования уточнять интервал значений эквивалентной одномерной случайной величины *Y* с точностью не до одного, а до *k* разрядов. Диаграмма процесса формирования кода случайной величины при 2-разрядном формировании приведен на рис. 3.6. Как и в алгоритме поразрядного формирования, исходной информацией для алгоритма k-разрядного формирования служат значения эквивалентной одномерной функции распределения . На каждом шаге формирования кода случайной величины *Y* производится сравнение упорядоченных значений функции с кодом равновероятностного случайного числа и в зависимости от результатов сравнения формируются *k* очередных разрядов кода случайной величины *Y.*

|  |
| --- |
|  |
| 0 |

|  |
| --- |
| … |
| 1 |
| . . . . . . . . . . . . . . . . . . . . . . . |
| . . .  11…11  11…10  00…01  00…00 |

Рисунок 3.6. Диаграмма процесса формирования кода случайной величины *Y* при 2-разрядном формировании

НАЧАЛО

Формирование массивов

А

Формирование

равновероятного числа

Вычисление

Выборка сформирована?

КОНЕЦ

А

А

2

3

6

7

8

нет

да

нет

да

…

…

5

9

Рисунок 3.7. Схема алгоритма *k*-разрядного формирования многомерных случайных величин

Обоснуем алгоритм *k*-разрядного формирования. Пусть имеется случайная величина . Задана функция распределения . Если удовлетворяет последовательности уравнений

|  |  |
| --- | --- |
| , | (3.31) |

Причём , а , где

|  |  |
| --- | --- |
|  | (3.32) |
|  |  |

– *k-*разрядные двоичные коды, сформированные на шагах ; ; – равновероятные случайные числа (; то случайная величина описывается функцией распределения вероятностей .

Из соотношений (3.31), (3.32) последовательно получаем:

………

Следовательно

|  |  |
| --- | --- |
|  | (3.33) |

Что и требовалось доказать.

Схема алгоритма *k-*разрядного формирования многомерных случайных величин с заданной многомерной функцией распределения вероятностей приведена на рис. 3.7.

Имитация осуществляется в следующей последовательности:

1. Формируются массивы модифицированной функции распределения для многомерной случайной величины , заданной многомерной плотностью распределения . Под модифицированной функцией распределения будем понимать упорядоченную функцию распределения в соответствии с определением, данным в п 2.2. При выполнении алгоритма *k-*разрядного формирования многомерных случайных величин текущий код случайной величины *Y* формируется начиная с кода путём вычисления на каждом шаге *k-*разрядного кода и занесения этого кода в соответствующие разряды кода *Y* (блоки ).

Аргументы функции и модифицированных функций распределения связаны между собой соотношениями:

|  |  |
| --- | --- |
|  | (3.34) |

Причём аргумент *L* (текущее значение случайной величины *Y*) на шаге *H* принадлежит счётному множеству :

|  |  |
| --- | --- |
|  | (3.35) |

2. Проверяется, сформирован ли заданный объём выборки *M* для многомерной случайной величины (блок 9). Если заданное значение объёма выборки не достигнуто, выполняется переход к блоку 3, в противном случае формирование заканчивается.

3. Выполняется формирование равномерно распределённого случайного числа .

4. Вычисляются значения коэффициентов по формуле

|  |  |
| --- | --- |
|  | (3.36) |

Здесь функция определяется следующим образом

|  |  |
| --- | --- |
|  | (3.37) |

Коэффициенты с точностью до двоичных разрядов определяют интервал, на который попадает формируемое значение случайной величины .

5. Вычисляются очередных разрядов формируемого значения случайной величины . Для этого суммируются коэффициенты . Полученный разрядный двоичный код заносится в разряды с по .

Таким образом, на данном шаге формируется текущий код , равный

|  |  |
| --- | --- |
|  | (3.38) |

6. Производится проверка окончания формирования очередного значения случайной величины . Если сформированы не все разряды, выполняется переход к блоку 3, и цикл, включающий блоки 3, 4, 5, 6 повторяется, в противном случае осуществляется переход к блоку 7.

7. Код преобразуется в коды составляющих на основе соотношений (3.23), (3.24).

8. Коды преобразуются (масштабируются) – блок 8.

3.5.Алгоритм имитации многомерных случайных величин с программируемым быстродействием

Рассмотренные в гл. 2 задачи, использующие имитацию многомерных случайных воздействий, характеризуются различными требованиями к быстродействию и точности формирования многомерных случайных воздействий с заданными статистическими характеристиками. Это определяет необходимость создания алгоритма, позволяющего управлять скоростью формирований значений многомерных случайных воздействий и точностью воспроизведения заданных характеристик, для обеспечения перестройки алгоритма имитации в соответствии с требованиями решаемой задачи.

Указанная цель может быть достигнута, если формировать составляющие многомерной случайной величины не одну за другой, а одновременно вырабатывать один или несколько разрядов каждой из составляющих и останавливать процесс формирования в необходимый момент. Процесс формирования эквивалентной одномерной случайной величины при этом соответствует диаграмме, представленной на рис. 2.6, однако эквивалентная одномерная функция распределения получается из многомерной функции распределения в результате преобразования, отличного от преобразования (2.1), (2.2) Рассматривая как тензор ранга ноль, перейдем от системы координат к системе координат

Поскольку необходимо поразрядное формирование каждой из составляющих , представим их в виде

|  |  |
| --- | --- |
|  | (3.57) |

где – разрядность двоичных слов, используемых для представления составляющих

Координаты определим через координаты системой соотношений

|  |  |
| --- | --- |
|  | (3.58) |

Функция распределения и связаны соотношением

|  |  |
| --- | --- |
|  | (3.59) |

Данное преобразование двумерной плотности распределения вероятностей представлено на рис. 3.11.

Достаточным условием взаимно однозначного соответствия наборов и () является однозначное преобразование переменной в значения переменных . Данное условие выполняется за счет соответствующего выбора коэффициентов .

При выборе коэффициентов для обеспечения минимума аппаратурных затрат при реализации алгоритма имитации необходимо обеспечить минимальную разрядность переменной . Кроме того, для удобства аппаратурной реализации алгоритма коэффициенты должны представлять собой степени числа 2. Минимальная разрядность переменной достигается при выполнении следующих соотношений

|  |  |
| --- | --- |
|  | (3.60) |

Из соотношение (3.60) следует, что

|  |  |
| --- | --- |
|  | (3.61) |

Полагаем , тогда матрица коэффициентов имеет вид:

|  |  |
| --- | --- |
|  | (3.62) |

При задании коэффициентов согласно (3.62) множество возможных значений состоит из точек. Рассмотрим обратное преобразование значений в значения составляющих . В множество входят элементы

|  |  |
| --- | --- |
|  | (3.63) |

Определим множества следующим образом

|  |  |
| --- | --- |
|  | (3.64) |

Из (3.64) следует, что равно

|  |  |
| --- | --- |
|  | (3.65) |

Из соотношений (3.63) и (3.64) получаем

|  |  |
| --- | --- |
|  | (3.66) |

Таким образом, соотношения (3.64), (3.66) определяют обратное преобразование одномерной случайной величины в составляющие многомерной случайной величины . Схема алгоритма с программируемыми параметрами для имитации многомерных случайных величин приведена на рис. 3.7.

Формирование эквивалентной одномерной случайной величины выполняется на блоках 3, 4, 5, 6. В блоке 6 производится анализ получения заданной разрядности . Если заданное количество разрядов сформировано, то в блоке 7 выполняется преобразование значения в значения составляющих . При необходимости воспроизводимая функция распределения подвергается равномерной аппроксимации. С этой целью в блоке 8 формируются аппроксимирующие равновероятные случайные числа , которые затем суммируются с кодами .

Объем памяти, необходимый для реализации данного алгоритма, определяется соотношением

|  |  |
| --- | --- |
|  | (3.67) |

где – разрядности кодов функции распределения, – разрядность каждой из составляющих . Точность имитации определяется соотношением (3.56).

Оценка быстродействия алгоритма с программируемыми параметрами получается на основе формулы, полученной для времени формирования алгоритма разрядного формирования (3.40). Подставляя в выражение (3.40) вместо величины , определяющей количество шагов формирования, величину получаем

|  |  |
| --- | --- |
|  | (3.68) |

Максимальное значение имеет место в том случае, когда формируются все разрядов каждой из составляющих . Следовательно, применение алгоритма с программируемыми параметрами позволяет значительно повысить быстродействие при имитации, если требования к точности снижены. Кроме того, данный алгоритм универсален в том смысле, что он позволяет обеспечить различные требования по точности и быстродействию.

3.6. Алгоритмы имитации многосвязных марковских цепей и импульсных случайных процессов по заданной многомерной функции распределения

Алгоритмы имитации многосвязных марковских цепей и импульсных случайных процессов по заданной многомерной функции распределения вероятностей базируется на алгоритмах формирования многомерных случайных величин.

При имитации многосвязной марковской цепи по заданным условным функциям распределения определяется многомерная функция распределения F() и текущее значение формируется как значения одной составляющей на основе известных значений составляющих . Схема алгоритма имитации многосвязных марковских цепей приведена на рис. 2.12. Основой данного алгоритма является алгоритм -разрядного формирования многомерных случайных величин (блоки 3, 4, 5). В качестве значения случайного процесса в момент времени принимается значение составляющей , полученное на данном шаге формирования (блок 6):

|  |  |
| --- | --- |
| , | (3.69) |

где – разрядность двоичных кодов, используемых для представления составляющих многомерной случайной величины. Значение эквивалентной одномерной случайной величины для следующего шана формирования формируется путем сдвига предыдущего значения на разрядов в сторону старших разрядов и записи в освободившиеся младшие разряды кода :

|  |  |
| --- | --- |
|  | (3.70) |

Быстродействие данного алгоритма (время формирования очередного значения ) может быть определено на основе соотношения (2.40) с учетом того, что цикл формирования включает всего один шаг. Таким образом, время формирования одного значения марковской цепи cоставляет:

|  |  |
| --- | --- |
|  | (3.71) |

Необходимый объем памяти для данного алгоритма определяется выражением

|  |  |
| --- | --- |
| , | (3.72) |

задающим объем информации о многомерной функции распределения вероятностей .

Методические погрешности алгоритма имитации многосвязных марковских цепей определим как погрешности воспроизведения заданной многомерной функции распределения вероятностей. Эти погрешности могут быть оценены соотношением

|  |  |
| --- | --- |
|  | (3.73) |

получаемым из (2.56) с учетом разрядности кодов составляющих многомерной случайной величины .

При формировании импульсных случайных процессов по заданной многомерной функции распределения используется процедура, которая включает формирований многомерной случайной величины представляющей собой параметры импульсного процесса, и последующее функциональное преобразование этой многомерной случайной величины в импульсный случайны процесс (рис. 3.13).

Если импульсный процесс определяется тремя параметрами: амплитудой , длительностью и интервалом следования импульсов , и коррелированы только два параметра (например, амплитуда и длительность импульсов), то трехмерная функция плотности распределения вероятностей многомерной случайной величины представляется в виде

|  |  |
| --- | --- |
|  | (3.74) |

В блоке 3 выполняется формирование кодов по заданной с использованием одного из рассмотренных алгоритмов формирования многомерных случайных величин. Выбор конкретного алгоритма осуществляется на основе требований к быстродействию и точности алгоритма.

В блоке 4 выполняется формирование значений импульсного процесса по заданной форме импульсов

|  |  |
| --- | --- |
|  | (3.75) |

и сформированному значению амплитуды импульса

|  |  |
| --- | --- |
|  | (3.76) |

В блоке 5 формируется временной интервал соответствующий паузе между соседними импульсами

|  |  |
| --- | --- |
|  | (3.77) |

Быстродействие алгоритма имитации импульсных процессов определяется двумя факторами:

* скоростью формирования значений параметров ;
* скоростью функционального преобразования, выполняемого в блоке 4.

Полагая, что преобразование выполняется с использованием одной операции умножения для каждого из моментов времени , где M – количество точек задания импульса, и что коды формируются с использованием алгоритма поразрядного формирования, можно сделать вывод о том, что общее быстродействие алгоритма имитации импульсных случайных процессов определяются скоростью формирования параметров . При заданной дискретности формирования импульсов и времени формирования параметров , где – разрядность кодов параметров, получаем минимальный интервал формирования значений импульсного процесса , равный

|  |  |
| --- | --- |
|  | (3.78) |

|  |
| --- |
| Начало  Формирование массивов  Формирование равновероятного числа  Выборка сформирована?  Конец  … 2  2  3  5  6  7  да  нет |
| Рис. 2.12 Блок-схема алгоритма имитации многосвязных марковских цепей |
| Начало  Формирование массивов  Формирование    Формирование временного интервала  Выборка сформирована?  Конец  2  3  4  5  6  да  нет |
| Рис. 3.13 Блок-схема алгоритма имитации импульсных случайных процессов |