**4. МОДЕЛИРОВАНИЕ СЛУЧАЙНЫХ ФАКТОРОВ**

Имитационная модель позволяет исследовать поведение различных систем с учетом влияния случайных факторов. Эти факторы в зависимости от их природы могут быть отражены в модели как случайные события, случайные величины (дис­кретные или непрерывные), или как случайные функции (процессы).

В основе всех методов и приемов моделирования случайных факторов лежит использование базовой случайной величины -- случайных чисел, имеющих равномерное распределение на интер­вале [0; 1].

**4.1.Построение датчиков БСВ**

**4.1.1. Датчики БСВ**

***Базовой*** случайной величиной (БСВ) в статистическом моделировании называют непрерывную случайную величину *Z*, равномерно распределенную на интервале (0,1). Ее плотность распределения вероятностей имеет вид:



Математическое ожидание M[z] и дисперсия D[z] БСВ составляют



соответственно.

БСВ моделируется на ЭВМ с помощью *датчиков* БСВ. Датчик БСВ  это устройство или программа, выдающая по запросу одно или несколько независимых значений *z1* , ..., *zn* БСВ.

Датчики БСВ могут быть трех типов: табличные, физические и программные.

***Табличный*** датчик БСВ  это просто таблица случайных чисел. Основной недостаток такого датчика  ограниченное количество случайных чисел в таблицах. А в статистическом эксперименте часто требуется не ограниченное заранее их количество.

***Физический*** датчик БСВ  это специальное радиоэлектронное устройство в ЭВМ, содержащее источник электронного шума. Шум преобразуется в случайные числа с распределением. Недостатки физического датчика БСВ: невозможность повторения каких-либо ранее полученных реализаций *z1*, ... , *zn* без их предварительной записи в память ЭВМ, схемная нестабильность и сложность тиражирования датчика.

***Программный*** датчик БСВ обычно вычисляет значения *z1*, *z2*,..., по какой-либо рекуррентной формуле типа

*zi = f ( zn),*

при заданном стартовом значении *z0*.

Заданное значение *z0* полностью определяет всю последовательность реализаций *z1*, *z2*,..., поэтому *z* часто называют *псевдослучайной* величиной. Но ее статистические свойства идентичны свойствам "чисто случайной" последовательности, что и обеспечивает успех статистического моделирования.

Программный датчик БСВ имеет следующие преимущества: простота создания датчика, простота применения, простота тиражирования, надежность, быстродействие, высокая точность достижения необходимых статистических свойств, сравнимая с точностью представления вещественных чисел, компактность, повторяемость, когда это нужно, любых последовательностей случайных значений без их предварительного запоминания.

В дальнейшем мы будем рассматривать только программные датчики БСВ.

Имея датчик БСВ *Z*, можно промоделировать любые случайные факторы: непрерывные или дискретные случайные величины (как простые, так и многомерные), случайные события, случайные процессы и поля и т.д. Для этого достаточно соответствующим образом преобразовать последовательность *z1*, *z2*, ... . Поэтому БСВ *Z* и называют базовой.

Теоретически в качестве базовой можно было бы взять почти любую случайную величину. Использование СВ *z* с распределением обусловлено технологическими соображениями: простотой и экономичностью датчика, простотой преобразования *Z* в другие случайные факторы, относительной простотой тестирования датчика.

**4.1.2.Метод середины квадрата**

Метод середины квадрата предложен для получения псевдослучайных чисел Д. фон Нейманом в 1946 г. Один из вариантов этого метода заключается в следующем.

1. Возьмем произвольное *n*-разрядное число.
2. Возведем полученное число в квадрат и, если необходимо, добавим к результату слева нули до 2*n*-разрядного числа.
3. Возьмем четыре цифры из середины 2*n*-разрядного в качестве нового случайного *n*-разрядного числа.
4. Если нужны еще случайные числа, то перейдем к пункту 2.

*Например*, если взять в качестве начального числа 1994, то из него получается следующая последовательность псевдослучайных чисел: 9760 2576 6357 4114 9249 5440 5936 2360 5696 4444 7491 1150 3225 4006 0480 2304 3084 5110 1121 2566 ...

Сам по себе метод середины квадрата не получил широкого распространения, так как выдает "больше чем нужно малых значений". Но открытый в нем принцип используется во многих, если не во всех, более поздних датчиках БСВ. Этот принцип состоит в вырезании нескольких цифр из результата какой-либо операции над числами.

**4.1.3. Мультипликативный конгруэнтный метод**

Так называемый мультипликативный конгруэнтный датчик БСВ задается двумя параметрами: модулем *m* и множителем *k*. Обычно это достаточно большие целые числа.

При заданных *m*, *k* числа *z1*, *z2*, ..., вычиcляются по рекуррентной формуле:

*Ai* = (*kAi -1*) mod *m*, *i* = 1, 2,...,                      (4.1)

*zi = Ai / m*,

где *m * модуль,

*k * множитель,

*A0*  начальное значение,

mod  операция вычисления остатка от деления *kAi -1* на *m*.

Таким образом, *A1* определяется как остаток от деления  *kA0* на *m*; *A2* - как остаток от деления *kA1* на *m* и т.д. Поскольку все числа *Ai*  это остатки от деления на *m*, то 0   *Ai* < *m*. Разделив последнее неравенство на *m*, видим, что 0  *Ai / m*< 1, т. е. 0  *zi* <1.

Из неравенства 0  *Ai* < *m* вытекает также, что датчик (4.1) дает периодическую последовательность *Ai*. Действительно, число всех возможных остатков от 0 до *m* - 1 равно *m* и, рано или поздно, на каком-то шаге *i* обязательно появится значение *Ai*, уже встречавшееся ранее. С этого момента последовательность *Ai* “зациклится".

Длина периода *T* будет не больше *m -* 1. Например, если встретится остаток *Ai*= 0, то далее, согласно (4.1), будет *Ai+ 1* = 0, *Ai+ 2* = 0, ... , т.е. длина периода *T* = 1. Ненулевых же остатков в интервале 0 *Ai* < *m* всего *m -* 1, и, если все они войдут в период, будет *T* = *m -* 1. Это имеет место, например, при *m* = 13, *k* = 7; в этом случае ряд *Ai* выглядит так:

1, 7, 10, 5, 9, 11, 12, 6, 3, 8, 4, 2,  1, 7,... .  
\\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/

*T* = *m -* 1 = 12

Поскольку в качестве случайной можно использовать лишь подпоследовательность *Ai* внутри одного периода, то параметры датчика выбирают так, чтобы длина периода *T* была максимальной. С учетом ограничения *T m -* 1 модуль *m* берут максимально возможным. Чтобы упростить вычисление остатков по (4.1), для двоичных ЭВМ часто берут *m* = 2*n*. Рекомендуется также брать достаточно большой множитель *k*, причем взаимно простой с *m*.

В можно найти подробные рекомендации по выбору параметров *m*, *k* и начального значения *A0* . Заметим, однако, что в настоящее время не известны правила, которые гарантировали бы высокое качество датчика без его специального статистического тестирования.

Датчик (4.1) называют мультипликативно-конгруэнтным потому, что он использует две основные операции  умножение (англ. multiplication) и вычисление остатка (в теории чисел  получение конгруэнтного числа). Можно было бы поэтому перевести его название и как "множительно-остатковый датчик".

Обратим внимание также и на то, что операция вычисления остатка воплощает здесь упоминавшийся в п. 4.1.2 неймановский принцип вытаскивания цифр. Это становится очевидным, если записывать числа в системе счисления с основанием *m*. Тогда операция *X* mod *m* означает выбор последней цифры из числа *X*. Для *m* = 2*n* операция *X* mod *m* означает также выделение последних *n* цифр из двоичной записи числа *X*.

В качестве примера рассмотрим таблицу параметров датчиков, предлагаемых в некоторых публикациях и программных продуктах.

|  |  |  |
| --- | --- | --- |
| Место использования датчика  (программный продукт или публикация) | модуль  *m* | множитель  *k* |
| Язык моделирования СИМУЛА | 2*35* | 5*16* |
| Пакеты LLRANDOM, IMSL | 2*31*  1 (простое число) | 16807 |
| Язык моделирования SIMSCRIPT | 2*31*  1 | 63036001 |

**4.1.4. Специализированные вычислительные устройства для формирования случайных чисел**

Опыт создания имитационных моделей показывает, что значительная часть вычислительных ресурсов тратиться на формирования случайных чисел. Для повышения эффективности систем имитационного моделирования необходимо предусмотреть методы и средства, позволяющие формировать базовые случайные величины с использование аппаратных средств. Такими средствами являются специализированные вычислительные устройства СВУ, которые подключаются как модули расширения ЭВМ.

Вычислительных устройства формирования случайных величин обладают всеми достоинствами как физических, так и программных датчиков случайных чисел: высоким быстродействием и стабильностью характеристик.

Примерами таких модулей могут быть разработанные в НИЛ 1.10 нашего университета *МОДУЛЬ ФОРМИРОВАНИЯ СЛУЧАЙНЫХ ЧИСЕЛ, ПРОЦЕССОВ И СИГНАЛОВ ДЛЯ ПЭВМ, КОНТРОЛЛЕР ГЕНЕРАТОРА СЛУЧАЙНОГО ПРОЦЕССА* и другие устройства. В качестве примера рассмотрим характеристики *КОНТРОЛЛЕРА ГЕНЕРАТОРА СЛУЧАЙНОГО ПРОЦЕССА (КГСП)*.

Контроллер генератора случайных про­цес­сов (КГСП) реализован в виде большой интегральной схемы на базовом матричном кристалле серии 1574ХМ2 и предназначен для построения генераторов импульсных и непре­рывных случайных про­цессов с управляемыми временными и вероятностными характери­сти­ками.

Осно­вные области применения КГСП – системы имитации радиолокационных и радио­технических сигналов, системы за­щиты инфо­рмации, датчики случайных и псевдослучайных чисел, аппаратные моде­ли сложных систем и т.д.

КГСП может быть использован в качестве:

* 40-разрядного генератора М-последовательности с произвольным начальным состоя­ни­ем, который обеспечивает генерацию последовательности максимальной длины, равной (240-1) периодов тактовой частоты;

*Наличие управляющего входа DI позволяет «за­пускать» генерацию М-последовательности про­изво­льного начального состо­яния и осуществлять случайную "развязку" генери­руемых чисел. Иници­ализация генератора М-последовательности осуществляется при помощи сигнала сброса, после которого при необходимости на вход DI пода­ется последовательность длиной 40 бит, определяющая начальное состояние гене­ратора. Подача на вход DI уровня "1" при нулевом состоянии регистра М-после­до­вательности блокирует работу генератора, так как сдвиговый регистр остается в нулевом состоянии.*

* 19-разрядного генератора псевдослучайных и случайных (при использовании входа внешней "развязки") чисел;

*Закон распределения чисел либо равномерно распределенный (вероятность появле­ния на выходах Q18 - Q0 "0" и "1" равна 0,5), либо в зависимости от значения на входах управления AS, BS, CS на соответствующем выходах (Q05, Q06, Q07) формируется числа с более вероятным появлением "0":*

* синхронного 19-разрядного счетчика с параллельным занесением произвольного или случайного начального значения;
* асинхронного 19-разрядного счетчика со сбросом;
* полного 11-разрядного сумматора.

Тактовая частота -- *30 МГц*

4.2. Характеристики датчиков базовых случайных величин

Практика показывает, что результаты имитационного моделирования существенно зависят от качества используемых последовательностей псевдослучайных чисел. Поэтому используемые в имитационной модели генераторы случайных чисел должны пройти тесты на пригодность. Основные анализируемые характеристики генерируемых датчиком последова­тельностей:

• равномерность;

• стохастичность (случайность);

• независимость.

Рассмотрим методы проведения такого анализа, наиболее часто применяемые на практике.

**4.2.1. Тестирование равномерности**

Обозначим равномерное распределение вероятностей на интервале (0,1) через *R*[0,1]. Тогда утверждение, что БСВ *Z* имеет распределение *R*[0,1], можно кратко записать в виде *z* ~ *R*[0,1].

С помощью статистических тестов проверяют два свойства датчика, делающих его точной моделью идеальной БСВ,  это *равномерность* распределения чисел *Zi*, выдаваемых датчиком на интервале (0,1), и их статистическая *независимость*. При этом числа *zi* рассматривают как реализации некоторой **СВ.**, т.е. как статистическую выборку.

Достаточно простым методом проверки равномерности распределения является частотный тест. Он основан на законе больших чисел и выполняется по следующему алгоритму.

1. Разобьем интервал (0,1) на *K* равных отрезков (например, *K* = 10).

2. Сгенерируем *n* чисел *z1*,..., *zn* с помощью тестируемого датчика БСВ (например, *n* = 100).

3. Подсчитаем, сколько чисел попало в каждый из *k* отрезков, т.е. найдем числа попаданий *n1*,...,*nk*.

1. Рассчитаем относительные частоты попаданий в отрезки:



5. Построим гистограмму частот на *K* отрезках интервала (0,1).

6. Повторим действия (2)  (5) для большего значения *n* (например, для *n* =10 000).

7. Оценим по полученным гистограммам сходимость каждой частоты к вероятности *p* = 1/*K* того, что БСВ попадет в *i*-й отрезок. Согласно закону больших чисел должно быть

 (4.2.)

Это значит, что высоты столбиков во второй гистограмме должны в целом быть ближе к уровню 1/*K*, чем в первой.

Тестирование датчика на равномерность можно совместить с оцениванием **математического ожидания *m\**** и дисперсии S2. Оценки ***m\**** и S2 рассчитываются соответственно по формулам:

 (4.3)

 (4.4)

С ростом *n* оценки  и  должны сходиться по вероятности к точным значениям *M*(*z*) = 1/2, *D*(*z*) = 1/12 = 0.08333... .

**4.2.2. Тестирование стохастичности**

Рассмотрим один из основных методов проверки – метод комбинаций. Суть его сводится к следующему. Выбирают достаточно большую последовательность случайных чисел xi и для нее определяют вероятность появления в каждом из xi ровно j единиц. При этом могут анализироваться как все разряды числа, так и только l старших. Теоретически закон появления j единиц в l разрядах двоичного числа может быть описан как биномиальный закон распреде­ления (исходя из независимости отдельных разрядов).

Тогда при длине выборки N ожидаемое число появлений случайных чисел xi с j единицами в проверяемых l:

**

Для полученной последовательности определяется эта же характеристика. Про­верка соответствия реального значения теоретическому выполняется с помощью одного из статистических критериев согласия.

**4.2.3. Тестирование независимости**

Простейшую проверку статистической независимости реализаций *z1*, *z2*, ..., можно осуществить, оценивая корреляцию между числами *zi* и *zi+s*, отстоящими друг от друга на шаг *s* >1.

Для вывода формулы, по которой можно рассчитать коэффициент корреляции чисел *zi* и *zi+ s* , рассмотрим две произвольные **с.в**. *x*, *y*. Коэффициент корреляции определяется для них формулой:

                 (4.5)

Если известно, что *x*, *y* ~ *R*[0,1], то *M*(*x*) = *M*(*y*) = 1/2 и *D*(*x*) = *D*(*y*) =1/12, то есть (4.5) принимает вид:

*R*(*x,y*) = 12 *M*(*xy*) - 3.                      (4.6)

Условимся рассматривать пару чисел (*zi* , *zi+ s*) как реализацию пары **СВ** (*x,y*). Тогда в выборке *z1*,..., *zn* имеем всего *n - s* реализаций этой пары:

(*z1* , *z1+ s*),(*z2* , *z2+ s*),... ,(*zn* , *zn+ s*).

По ним можно рассчитать оценку *R*' коэффициента корреляции *R*(*x,y*), заменяя в (4.6) математическое ожидание *M*(*xy*) соответствующим средним арифметическим:

                    (4.7)

С ростом *n* оценка *R*' должна приближаться к нулю, в противном случае датчик БСВ не отвечает требованию независимости.

Конечно, если *R*' сходится к нулю, то это еще не гарантирует наличие независимости, но все же один из тестов оказывается успешно выдержанным. При желании всегда можно продолжить испытания датчика другими методами.

Еще одна важная характеристика датчика СЧ — **длина отрезка периодичности *L****.* Если в основу работы датчика положен мультипликативный метод, то оценить L несложно: она определяется величиной константы *М.*

**4.3. Случайные события и их имитация**

**4.3.1.Имитация случайного события**

Пусть некоторое событие А происходит с вероятностью . Требуется воспроизвести факт наступления события А. Поставим в соответствие событию А событие В, состоящее в том, что *х* меньше либо равно, где х здесь и в дальнейшем – случайное число (СЧ) с равномерным на интервале (0,1) законом распределения. Вычислим вероятность события В:



Таким образом, события А и В являются равновероятными. Отсюда следует процедура имитации факта появления события А. Она сводится к проверке неравенства  меньше, либо равно Р, а алгоритм заключается в следующем:

1. С помощью датчика случайных чисел (СЧ) получают СЧ *Х*;

2. Проверяют выполнение неравенства Х меньше, либо равно ;

3. Если оно выполняется, то событие А – произошло, если нет – то произошло 

**4.3.2. Имитация сложного события**

Имитация сложного события, состоящего, например, из двух независимых элементарных событий А и В, заключается в проверке неравенств:

,

где  и – вероятности событий А и В, а *х1* и *х2* – СЧ с равномерным законом распределения.

В зависимости от исхода проверки неравенств (аналогично алгоритму 4.2.1.) делается вывод какой из вариантов:

имеет место.

**4.3.3. Имитация сложного события, состоящего из зависимых событий****.**

В случае, когда сложное событие состоит из элементарных зависимых событий А и В имитация сложного события производится с помощью проверки следующих неравенств:

В зависимости от того, какая из этих четырех систем неравенств выполняется, делается вывод о том, какой из этих четырех возможных исходов  имеет место.

В качестве исходных данных задаются, и условная вероятность , вероятность  может быть вычислена. По формуле полной вероятности:

,

где

, отсюда легко выразить 

**4.3.4. Имитация событий, составляющих полную группу**

Пусть событие Аi (i=1,n) составляют полную группу, тогда их вероятности Рi, таковы что:



Имитация факта появления одного из событий Аi (i=1,n) сводится к проверке следующих неравенств:



Выполнение К-го неравенства эквивалентно выполнению события АК. Описанный алгоритм называют иногда алгоритмом “розыгрыша по жребию”. Его можно интерпретировать как установление номера К-го отрезка длинной РK, на который пало СЧ х, при условии разбиения отрезка единичной длины на отрезки с длинами P1,P2,...Pn (рис 4.3.)

ris29

Рис. 4.3

**4.4. Имитация непрерывных случайных величин**

**4.3.1. Метод обратной функции**

Пусть непрерывная случайная величина Y задана своим законом распределения:

,

где *f(y)*– плотность распределения вероятностей, а *F(y)-* функция распределения вероятностей. Доказано, что случайная величина



распределена равномерно на интервале (0,1).

Отсюда следует, что искомое значение y может быть определено из уравнения:

 (4.8)

которое эквивалентно уравнению:

 (4.9)

где y – значение случайной величины Y, a x – значение СВ X. Решение уравнения (4.9) можно записать в общем виде через обратную функцию



Основной недостаток метода заключается в том, что интеграл (4.8) не всегда является берущимся, а уравнение (4.9) не всегда решается аналитическими методами.

**4.4.2. Метод Неймана (режекции)**

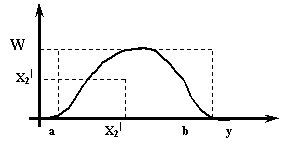
Метод Неймана, так же как метод обратной функции, является методом, позволяющим получить значения СВ в соответствии с заданным законом распределения. Этот метод является достаточно универсальным он применим для моделирования всех СВ, значения которых не выходят за пределы ограниченного интервала (a,b), а также для СВ, законы распределения которых можно аппроксимировать усеченными.

Метод Неймана состоит в следующем:

С помощью датчика случайных чисел получают пару чисел, распределенных равномерно на (0,1) x1 и x2.

Путем преобразований (по методу обратной функции получают два числа x1\* и x2\*, равномерно распределенных соответственно на интервалах (a,b) и (o,w), то есть

 и , где 



Из точек с координатами x1\* и x2\* выбирают те, которые попали “под колокол” функции *f(x)*, то есть те точки, для которых f(x1\*)<x2\*.

Если условие выполнено, то искомое значение y полагают равным x1\*.

**4.4.3. Алгоритм получения значения нормально распределенной случайной величины****.**

Нормальное распределение является наиболее часто встречающимся. Функция плотности распределения вероятностей для него имеет вид:



где *m* – математическое ожидание, а σ2– дисперсия. Согласно центральной предельной теореме теории вероятностей



распределена асимптотически нормально, если распределены одинаково.

Для практического получения значений *X* в качестве и выбирают равномерно распределенные случайные величины. При этом наиболее часто используют преобразование

 (4.10)

где *xi* – равномерно распределенные на (0,1) случайные числа. При к=12 формула приобретает вид наиболее удобной для расчетов, но она дает достаточно точные результаты уже для k=3,4. Формула (4.10) верна для центрированной (m=0) и нормированной ( =1) случайной величины.

Для получения y\*, распределенного нормально с произвольными m и σ, пользуются дополнительно преобразованием

y\*=m+σy (4.11)

**4.4.4. Алгоритм получения случайной величины, распределенной по Пуассону**

Закон Пуассона описывает число событий, происходящих за одинаковые промежутки времени, при условии независимости этих событий. Это распределение хорошо описывает количество вызовов телефонной станции за определенное время суток, заказов такси и т.д. Закон Пуассона называют законом появления редких событий.

В основе алгоритма получения случайных чисел, распределенных по закону распределения Пуассона, лежит предельная теорема Пуассона. В соответствии с этой теоремой, если *n* – количество событий велико, а *р* – вероятность успеха мала, то вероятность того, что при n испытаниях событие произойдет к раз равна:



Здесь *np*=*а*, где *а* – параметр закона Пуассона.

Процедура получения чисел, распределенных по закону распределения Пуассона, заключается в следующем:

* Положить *р* меньше, либо равно 0,1 (так как события являются редкими).
* Вычислить число испытаний *n=а/р*.
* Значение *х* – случайного числа с равномерным на интервале (0,1) законом распределения сравнить с *р*, если *х* меньше, либо равно *р*, то к счетчику событий добавляется 1.
* Проводится *n* испытаний, после чего содержимое счетчика можно считать случайным числом, распределенным по Пуассону.

Аналогично можно получить значения случайных величин, распределенных в соответствии с геометрическим, биноминальным и другими распределениями дискретных случайных величин.

**4.5.** **Алгоритмы получения значений систем случайных величин (случайных векторов).**

**4.5.1. Метод аналитических преобразований.**

Пусть системы непрерывных случайных величин (*x1, x2, …, xn*) задана условными законами распределения *xi* (*i*=1,*n*). По теореме умножения плотностей распределения: совместная функция плотности распределения вероятностей

*f(x1, x2, . . . xn)=f1(x1) f2(x2|x1) f3(x3|x1x2) . . . f1(xn| x1,x2, . . ., xn-1).*

Для системы двух случайных величин (*x1,x2*), алгоритм получения вектора ее значений сводится к следующему:

Вычисление частной функции плотности для *x1*:



Получение значения *X1* в соответствии с *f1*(*x*1) согласно любому методу, например, одному из описанных в предыдущем разделе.

Вычисление частной функции плотности для второй компоненты *x2* системы. Она может быть получена на основании теоремы умножения законов распределения:



Получение x2 – значения СВ X2 любым известным методом в соответствии с найденным законом ее распределения.

Алгоритм может быть обобщен для любого *n*. Однако, практические работы, выполняемые по этому методу, связаны с большими вычислительными трудностями, за исключением тех редких случаев, когда интегралы берутся. Поэтому разработаны другие методы, позволяющие решать задачу получения значений системы непрерывных случайных величин.

**4.4.2. Метод разложения по координатным случайным величинам.**

Пусть СНСВ задана в рамках теории корреляций: математическими ожиданиями компонент (m1, m2, . . . mn) и матрицей корреляционных моментов:

, 

Доказано, что  можно получить с помощью их разложения по координатам СВ xi:

 (4.12)

где *xi* - некоррелированные, центрированные, нормированные нормально распределенные СВ.

Коэффициенты  могут быть достаточно просто получены решением системы уравнений:

 (4.13)

Алгоритм получения значений СНСВ сводится к следующему:

* Решение системы нелинейных уравнений (4.13).
* Получение *n* значений *yi* нормированных, центрированных СВ, распределенных нормально.
* Вычисление xi i=(1,…,n) значений СВ, образующих систему непрерывных случайных величин в соответствии с (4.12).

**4.4.3. Алгоритм получения значений системы дискретных случайных величин**

Дискретный двумерный вектор CDCB задается двумерным законом распределения, т.е.

а) матрицей вероятностей , где Pij – вероятность совместного появления i-ого и j-ого значений соответственной первой и второй компоненты, причем:

.

б) двумя векторами возможных значений первой и второй компоненты {Ai}, {Bi}, .

Получение значений двумерной дискретной системы случайных величин может осуществляться по следующему алгоритму.

Вычисляют суммы , , .

Если *Х* - равномерно распределенное случайное число из интервала (0,1) такое, что , то считают, что *x1* компонента двумерной дискретной случайной величины получила k-ое значение.

Выбирают k-ую строку , вычисляют .

Если вновь полученное с помощью датчика случайных чисел *Х* такое, что вторая компонента получила S-е значение.

Замечание: В алгоритме используется правило “розыгрыша по жребию”, однако надо иметь в виду, что .