**МЕТОДИЧЕСКИЕ УКАЗАНИЯ**

для проведения лабораторных работ

по дисциплине **«Моделирование систем»**

#### ЛАБОРАТОРНАЯ РАБОТА № 1

#### «МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ

#### СЛУЧАЙНЫХ ВЕЛИЧИН»

1. Цель работы:

Изучение основных методов получения случайных величин и алгоритмов их моделирования. Создание программных датчиков случайных величин в среде Borland С++ 3.1.

**2. Теоретический материал для изучения**

## ВВЕДЕНИЕ

*Моделированием* называется метод, при котором исследуемый объект заменяется другим более простым объектом. Слово *функциональное* свидетельствует, что модель выполняет определенные функции исследуемого объекта. В зависимости от способа воплощения оригинала в модели различают физическое и математическое моделирование. В настоящее время в связи с широким использованием вычислительной техники наиболее популярно математическое моделирование. В этом случае исследуемый объект заменяется математическим алгоритмом, который описывает его функционирование. Математические модели могут быть непрерывными, дискретными или цифровыми. Модель называется непрерывной, если функции, входящие в ее состав, непрерывны по величине и во времени. Если функции, входящие в состав модели, дискретны во времени и непрерывны по величине, то модель считается дискретной. И, наконец, если функции, входящие в состав модели, дискретны как во времени, так и по величине, то модель считается цифровой. Большинство систем могут быть представлены в виде, показанном на рис.1.

Здесь возможны две ситуации. Первая заключается в том, что входное воздействие *Х(t)*, выходной процесс *Y(t)* и оператор преобразования *L[…]* определяются детерминированными функциями времени (т.е. их поведение во времени можно записать однозначно).

##### L[…]

*X(t)*

*Y(t)*

Рис. 1

Например, если генератор вырабатывает синусоидальное напряжение *Х(t) = U0*sin(ωt + ϕ), то значение напряжения можно определить в любой момент времени *t*. Вторая ситуация связана с тем, что входное воздействие *Х(t)* или оператор преобразования не могут быть описаны детерминированными функциями времени. В этом случае в каждый момент времени входное воздействие *Х(t)* или оператор преобразования с некоторой вероятностью могут принимать те или иные количественные значения из множества возможных. Для описания таких функций времени используются случайные процессы. Случайным можно назвать процесс, который не только является функцией времени, но и зависит от случайных факторов (т.е. функция времени, значение которой в любой момент времени – случайная величина (СВ)).

К примерам систем, характеризующихся случайными параметрами, можно отнести системы связи, у которых характеристики канала связи меняются случайным образом, а также локационные системы обнаружения отражений от целей на фоне случайных помех.

## 2.1. ОПИСАНИЕ СЛУЧАЙНЫХ ВЕЛИЧИН

Напомним методику описания случайных величин. Вероятность случайного события – это мера того, насколько велика возможность его возникновения. Вероятность изменяется от 0 до 1. Вероятность достоверного (которое точно произойдет) события равна 1. Напротив, вероятность невозможного события равна 0. Случайные величины являются более обобщенным понятием случайного события. Случайные величины могут быть дискретными или непрерывными. Дискретная случайная величина может принимать только определенные значения с определенной вероятностью. Например, на рис.2 показаны вероятности значений случайной величины *Х*.

***Р*** = 0.3 при ***Х*** = 1,

***Р*** = 0.3 при ***Х*** = 3,

***Р*** = 0.4 при ***Х*** = 5.

# Р

0

1

3

5

***Х***

Рис. 2

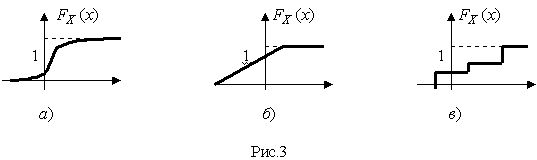
Для описания как дискретных, так и непрерывных случайных величин используется функция распределения вероятности. Пусть Х – случайная величина и х - ее любое значение. Функция распределения вероятности определяется следующим образом:

*Fx(x) = P(X ≤ x),*

т.е. она равна вероятности того, что случайная величина *Х* примет значение *X ≤ x*. Здесь и в дальнейшем большие буквы используются для обозначения случайных величин, а маленькие – для значений, принимаемых случайными величинами. Поскольку функция распределения вероятности представляет собой вероятность, то она удовлетворяет следующим свойствам:

1. 0 ***≤*** *FX(x) ≤ 1****,*** при x ∈(– ∞; + ∞),
2. *FX(*– ∞*)* = 0, *FX(+* ∞*)* = 1,
3. *FX(x)* – неубывающая функция,
4. *P(x1 < X < x2) = FX(x2)* – *FX(x1).*

На рис. 3 показаны примеры функций распределения вероятности.



Функция распределения вероятности не всегда удобна для расчетов. Часто удобнее использовать не саму функцию *FX(x)*, а ее производную. Она называется плотностью распределения вероятности:

*f(x) = dFX(x)/dx*.

Физический смысл *f(x)* состоит в том, что произведение *f(x)dx* представляет вероятность попадания случайной величины *Х* в интервал от *х* до *х + dx* , т.е.:

*f(x)dx = P(x ≤ X≤ x+dx).*

Свойства плотности распределения вероятности имеют вид:

 = 1 – вероятность достоверного события равна 1 ***(свойство 1)***,

 = *FX(x2)* – *FX(x1)* – вероятность попадания случайной величины в интервал от *x1* до *x2* ***(свойство 2)***.

**Пример.** Случайная величина *X* имеет плотность распределения вероятности следующего вида:

*f(х) = с*, при *-5 <x<-2*,

*f(х) = 0.1δ(x-4)*, при других *х* , где *δ(x)*- дельта-функция.

Это проиллюстрировано на рис. 4. Необходимо вычислить параметр *с*.

Этот пример относится к непрерывной случайной величине, которая может принимать и дискретное значение. Учитывая 1-е свойство плотности распределения вероятности и интегрирующее свойство дельта-функции, получим

 =  + = 3с + 0.1=1

# *f(x)*

###### Рис. 4

*0*

*с*

# *0.1δ(x-4)*

*4*

*x*

*-2*

*-5*

Из данных вычислений следует, что *с = 0.3*. Рассмотрим вопрос - как можно смоделировать случайную величину? В настоящее время в связи с большой популярностью вычислительной техники широко используется так называемый *алгоритмический способ*.

## 2.2. ПРИНЦИПЫ ПОСТРОЕНИЯ АЛГОРИТМИЧЕСКИХ ДАТЧИКОВ СЛУЧАЙНЫХ ВЕЛИЧИН

В этом случае воспроизведение случайных факторов реализуется с помощью специальных алгоритмов, что обеспечивает получение периодических детерминированных числовых последовательностей с большим периодом. Такие последовательности похожи на настоящие случайные и называются псевдослучайными или квазислучайными. Алгоритм получения периодической последовательности целых псевдослучайных чисел записывается следующим образом:

*g i = Z [a g i-1 / b],*

где *Z[…]* – остаток от деления целых чисел (например, *Z*[15/6] = 3), *i* = 1, 2,…... Здесь *a* и *b* – целые числа, при этом *a < b*. Например, если выбрать *а* = 7, *b* = 17, *g0* = 3, то получим следующую псевдослучайную последовательность чисел: 3, 4, 11, 9, 12, 16, 10, 2, 14, 13, 6, 8, 5, 17, 15, 3, 4, 11, 9,…. В этом случае период случайной последовательности равен 16, т.е. алгоритм позволяет получить последовательность из 16 целых чисел, каждое число в данной последовательности "условно" равновероятно принимает значение из интервала от 1 до 16. Такой алгоритм фактически имитирует случайную величину.

Практически во всех средах программирования присутствуют датчики случайных чисел, реализованные по алгоритмам, близким к приведенному ранее. Так, например, в среде Borland C++ есть функция **int random (int n)**, которая (при обращении к ней) выдает случайное целое число из интервала от *0* до *n*-1. Таким образом, написав в тексте программы строку ***a* = random (1000)**, мы получим значение (реализацию) случайной величины, которое будет присвоено переменной *а*. На рис. 5 приведена программа, которая формирует массив из 10 случайных целых чисел, равновероятно принимающих значение из интервала от 0 до 999, и выводит их на экран. В большинстве ситуаций, связанных с моделированием, в качестве исходного (или базового) для дальнейших преобразований требуется датчик случайных чисел с равномерной плотностью распределения вероятности из интервала от 0 до 1. График плотности распределения таких случайных чисел показан на рис. 6.

#define M10

#include <stdio.h>

#include <stdlib.h>

void main(void)

{ int z[M]; int i;

for (i=0; i<M; i++)

z[i]=random(1000);

printf(“z[%d]=%d\n”,

i,z[i]);

}

}

Рис. 5

###### Рис. 6

# *f(x)*

*0*

*1*

*x*

*1*

Очевидно, что если для получения случайного числа *х* воспользоваться функцией random в следующем варианте *х = random (N)/(float)N* , то при *N>>*1 (реально *N* порядка 10000) мы получим случайное число *х* с плотностью распределения вероятности, показанной на рис.6. На рис.7 приведена программа формирования массива случайных чисел с плотностью распределения вероятности *f(х) = 1* при 0 ≤ *x <* 1.

#define M10

#define N 30000

#include <stdio.h>

#include <stdlib.h>

void main(void)

{ float x[M]; int i;

for (i=0; i<M; i++)

x[i]=random(N)/(float)N;

}

Рис. 7

Используя датчик случайной величины *X* с равномерным распределением в интервале от 0 до 1, с помощью очевидного преобразования можно получить случайную величину *у* с равномерным распределением в интервале от *а* до *с*:

*y = a + (c – a)x.*

Рассмотрим теперь вопрос об экспериментальной оценке плотности распределения вероятности. Всегда есть неуверенность в том, что тот или иной алгоритм выдает случайные числа с требуемой плотностью распределения вероятности. Таким образом, задача заключается в экспериментальном исследовании *f(х)*. Считаем, что случайное число х попадает в некоторый диапазон, например *a ≤ x < b*, который мы, как правило, всегда знаем. Далее разбиваем этот интервал на *N* отрезков длиной *d = (b – a)/N*. Обращаясь к датчику случайной величины *М* раз, мы можем получить оценку вероятности попадания случайного числа в *n*-й отрезок *P(n) = Mn/M*, где *Mn* - число экспериментов, в результате которых величина *х* попала в *n*-й отрезок. Для пересчета вероятности попадания случайной величины в плотность распределения вероятности  *f(n)* необходимо поделить *Р(n)* на *d*. При увеличении *N*, *M* и большом отношении *M/N* мы получим экспериментальную оценку плотности распределения вероятности, близкую к истинной. Далее приведено формальное описание алгоритма.

Инициализируем массив *P(n) =* 0, где *n* изменяется в интервале от *0* до *N-1* включительно.

|  |  |
| --- | --- |
| **Цикл по *m* от *0* до *M*-1 (это цикл по экспериментам)** | |
| *x = sl\_vel* | *(обращение к датчику случайной величины);* |
| *n =* целая часть  [*N(x-a)/(b-a )*] | *(вычисление номера отрезка n, в который попало значение случайной величины);* |
| P(n) = P(n) + 1 | *( при попадании числа х в n - й отрезок добавляем единицу в элемент массива P(n)).* |
| **Завершение цикла по *m*** | |

Теперь для получения оценки плотности распределения вероятности *f(n)* необходимо поделить *P(n)* на *M*, а затем на *d*. Это выполняется в следующем цикле.

|  |
| --- |
| **Цикл по n от *0* до *N*-1 (это цикл по отрезкам)** |
| *f(n) = [P(n) /M]/ d* |
| **Завершение цикла по *n*** |

Описанная ранее методика статистической оценки плотности распределения вероятности выполнена в соответствии с общим методом моделирования Монте-Карло. Суть данного метода заключается в том, что для вычисления площади сложной фигуры необходимо случайно и равномерно формировать в прямоугольнике, окружающем фигуру, точки. Количество точек, попавших в эту фигуру, обозначим *MФ*, общее же количество "разбросанных" точек обозначим *M*. В этом случае при стремлении *M* к бесконечности отношение площади фигуры к площади прямоугольника (в котором равномерно разбрасываются точки) равно отношению *MФ/M*.

## 2.3. ТЕХНОЛОГИЯ ПОСТРОЕНИЯ МОДЕЛИРУЮЩИХ

## ПРОГРАММ В Borland C++ 3.1

Начнем с текста программы моделирование случайной величины равномерно распределенной в интервале от 1.5 до 3.5.

|  |  |
| --- | --- |
| **Программа 1 (исходный файл lect1\_1.cpp, выполняемый файл lect1\_1.exe)** | |
| #define N 500 | *// Число отрезков для вычисления плотности*  *// распределения вероятности* |
| #define M 100000L | *// Число экспериментов* |
| #include |  |
| #include "model.h" | // Подключение заголовочного файла с описанием  *// шаблонов функций графического вывода* |
| float sl vel(float a, float b); | // Описание функции, выдающей значение  *// случайной величины с равномерной плотностью*  *// распределения вероятности в интервале от а до b* |
| float f[N]; | // Массив для значений оценки плотности  *// распределения вероятности* |
| void main(void) |  |
| { long m;  float x, a = 1.5, b = 3.5, d;  int n ; | *// Индекс для номера эксперимента* |
| d = (b-a)/N ; | *//величина отрезка случайной величины х* |
| for(m =0; m < M; m++)  { x = sl\_vel(a,b);            n = (x - a) / (b - a) \* N;              f[n] = f[n] + 1;      } | *// Цикл по числу экспериментов* |
| for (n=0; n < N; n++) | // Пересчет вероятности в плотность распределения вероятности |
| f[n] = f[n] / ( M \* d ); |  |
| Init\_graph(); | *// Инициализация графической библиотеки* |
| graf\_1("plotnoct ", f, 0, N-1); | *// Вывод на график* |
| Close\_graph();  } | *// Переход в текстовый режим* |
| float sl\_vel( float a, float b) | // Функция, формирующая значение случайной  *//величины с равномерной плотностью*  *// распределения вероятности в интервале от a до b ( a x < b ).* |
| {  float x;      x = random(30000)/30000.;      x = a + (b-a) \* x;   return x; } |  |

Рассмотрим далее, каким образом выполнить данную программу в среде Borland - C++3.1. Во-первых, эта и дальнейшие программы будут использовать ряд сервисных графических функций ввода-вывода. Эти функции находятся в файле **model.cpp**. В свою очередь, описание шаблонов этих функций располагается в файле **model.h**. Поэтому порядок выполнения данной программы и всех последующих должен быть следующий:

1. Написать текст моделирующей программы в каком-нибудь файле, например, **my\_prog1.cpp** ( расширение **cpp** обязательно ).
2. Открыть новый проект. Для этого надо войти в меню **Project** и в разделе **Open project** ввести имя нового файла проекта, например, **my\_pro1.prj**. Расширение **prj** обязательно.
3. В этот новый проект надо включить Ваш файл (**my\_prog1.cpp**) и файл **model.cpp**. Для этого опять нужно войти в меню **Project** и выбрать раздел **Add Item**. Далее в появившемся окне последовательно выбрать **my\_prog1.cpp** (выделив и нажав кнопку **Add**) и **model.cpp** (аналогично).
4. Открыть раздел **Options** и в разделе **Linker** выбрать строку **Libraries**. В появившемся окне установите флажок напротив **Graphics Library**.
5. Теперь можно выполнить проект (меню **Run** раздел **Run**).

Следует обратить внимание на то, что в рабочем каталоге (каталог, из которого Вы переходите в среду – вызываете bc.exe) должны быть четыре файла model.cpp, model.h, egavga.bgi , trip.chr.

Далее перечислим ряд сервисных функций находящихся в файле **model.cpp**:

|  |  |
| --- | --- |
| 1) | функция, обеспечивающая вывод на график: void graf\_1(char \*stroka, float \*s1, int n\_begin, int n\_end). Здесь stroka – указатель на строку, содержащую название графика (это будет фактически подпись на графике), s1 – имя массива (указатель на массив), выводимого на график, n\_begin - индекс начального элемента массива, выводимого на график, n\_end – индекс последнего элемента массива, выводимого на график; |
| 2) | функция, обеспечивающая вывод двух массивов данных на график: void graf\_2( char \*stroka, float \*s1, float \*s2, int n\_begin, int n\_end). Новым параметром относительно предыдущей функции является: s2 – имя второго массива, выводимого на график. Максимальное значение по вертикальной оси на графике соответствует максимальному значению среди значений элементов обоих массивов. Соответственно минимальное значение по вертикальной оси на графике соответствует минимальному значению среди значений элементов обоих массивов; |
| 3) | функция, обеспечивающая вывод трех массивов данных на график void graf\_3( char \*stroka, float \*s1, float \*s2, float \*s3, int n\_begin, int n\_end ). Новым параметром относительно предыдущей функции является: s3 – имя третьего массива выводимого на график. Максимальное значение по вертикальной оси на графике соответствует максимальному значению среди значений элементов всех трех массивов. Соответственно минимальное значение по вертикальной оси на графике соответствует минимальному значению среди значений элементов всех трех массивов; |
| 4) | функция, обеспечивающая вывод двух массивов данных на график (с нормировкой значений): void graf\_2\_norm( char \*stroka, float \*s1, float \*s2, int n\_begin, int n\_end). Данная функция аналогично функции graf\_2 выводит на график два массива, однако нормировку производит для каждого массива свою. В итоге, если Вы выводите на график два массива, существенно различающихся по значениям, то каждый массив будет пронормирован к своим максимальному и минимальному значениям; |
| 5) | функция, обеспечивающая вывод трех массивов данных на график (с нормировкой значений): void graf\_3\_norm( char \*stroka, float \*s1, float \*s2, float \*s3, int n\_begin, int n\_end). Данная функция, аналогично функции graf\_3, выводит на график три массива, однако нормировку производит для каждого массива свою. В итоге, если Вы выводите на график три массива, существенно различающихся по значениям, то каждый массив будет пронормирован к своим максимальному и минимальному значениям; |
| 6) | следующая функция обеспечивает ввод с клавиатуры значения целочисленной переменной: void input\_int (char \*stroka, int \*perem). Здесь stroka – массив, содержащий подпись для окна ввода, рerem – адрес переменной, куда будет записано вводимое значение; |
| 7) | следующая функция обеспечивает ввод с клавиатуры значения вещественной переменной: void input\_float ( char \*stroka, float \*perem). Здесь stroka – массив, содержащий подпись для окна ввода, рerem – адрес переменной, куда будет записано вводимое значение; |
| 8) | для перехода в графический режим, в начале своей программы Вы должны поставить функцию, обеспечивающую инициализацию графики: void Init\_graph( void ); |
| 9) | в конце программы необходимо использовать функцию, которая обеспечивает обратный переход в текстовый режим: void Close\_graph( void ); |
| 10) | для получения случайных чисел, распределенных по гауссовскому закону, можно использовать функцию: float gauss (float m\_oz, float disp). Здесь m\_oz – математическое ожидание случайной величины, disp - дисперсия случайной величины. Файл **model.h** содержит описание заголовков этих функций, кроме того в нем определена константа: PI = 3.14159265 - число "пи". |

## 2.4. МОДЕЛИРОВАНИЕ ДИСКРЕТНОЙ СЛУЧАЙНОЙ

## ВЕЛИЧИНЫ

Случайная величина *Х* может принять значение *х1* с вероятностью *Р1*, значение *х2* с вероятностью *Р2*, …, значение *хn* с вероятностью *Рn*. Схематично это показано на рис. 8.

Для моделирования такой случайной величины можно воспользоваться датчиком случайной величины *E* с равномерным распределением в интервале от 0 до 1. Выданное датчиком значение *е* последовательно сравнивается следующим образом:

*Pn*

*x*

*xn*

*x3*

*x2*

*x1*

*0*

Рис. 8

если *е < P(x1)*, то принимаем *Х = x1*,

если *е < P(x1) + P(x2)*, то принимаем *Х = x2*,

……

если *е < P(x1) + P(x2)+ … +P(xn-1)*, то принимаем *Х = xn-1*,

если ни одно из предыдущих условий не выполнено, то принимаем *Х = хn*.

**Программа 2 (исходный файл lect1\_2.cpp, выполняемый файл lect1\_2.exe)**

Необходимо смоделировать случайную величину Х, которая может принимать четыре значения: 1, 27, 29 , 57 со следующими вероятностями:

Р (Х =1) = 0.05, Р (Х =27) = 0.45, Р (Х =29) = 0.49, Р(Х=57) = 0.01.

#define N 500

#include <stdlib.h>

#include "model.h"

float sl\_vel(void );

float realizasia[500];

void main(void)

{ int i, n;

for(i = 0 ; i < N; i++)

realizasia[i] = sl\_vel();

Init\_graph();

graf\_1(" realizasia random ", realizasia , 0, N-1);

Close\_graph();

}

float sl\_vel(void)

{

float x[4] = {1, 27 , 29 , 57 };

float P[3] = { 0.05 , 0.5 , 0.99 };

int i ; float x\_vihod , e;

e = random( 30000)/30000.;

x\_vihod = x[3] ;

for(i=0 ; i<3 ; i++)

{ if( e < P[i])

{ x\_vihod = x[i] ;

break ;

}

}

return x\_vihod;

}

## 2.5. МОДЕЛИРОВАНИЕ НЕПРЕРЫВНЫХ СЛУЧАЙНЫХ

## ВЕЛИЧИН

Универсальным методом моделирования непрерывных случайных величин является метод исключения. При моделировании случайной величины *Х* с плотностью распределения вероятности в интервале от *a* до *b* используется следующий алгоритм:

1. Получение от датчика случайных чисел с равномерной плотностью распределения вероятности в интервале от 0 до 1 двух независимых значений случайных величин: *e1*и *e2*.
2. Расчет *x1*= а + (b - а) *e1*, *x2* = *f*max *e2*,  где   *f*max - максимальное значение  *f(х)*.
3. Если *x2 ≤ f(x1)*, то *x1* представляет моделируемое значение случайной величины. Если данное неравенство не выполняется, то возвращаемся к пункту 1.

**Программа 3 (исходный файл lect1\_3.cpp, выполняемый файл lect1\_3.exe)**

Разработать функцию, формирующую (моделирующую) значения случайной величины с плотностью распределения вероятности f(x) = Asin(πx) на интервале 0 ≤ x < 1. Эта зависимость показана на следующем рисунке.

Вычислим *А*:  - вероятность достоверного события равна 1.

Рис. 9

*f(x)*

*x*

*0*

A

*1*

Далее, вычисляя интеграл, получаем:  =  = 1.

Из чего следует *А* = π/2.

#define N 500

#define M 30000L

#include

#include

#include "model.h"

float sl\_vel(void);

float f[N] ;

void main(void)

{ long i ; int n;

float x ;

for(i = 0 ; i < M ; i++)

{ x = sl\_vel();

n = x \* N ;

f[n] = f[n] + 1./M \* N ;

}

Init\_graph();

graf\_1("plotnost", f, 0, N-1);

Close\_graph();

}

float sl\_vel(void)

{

float , , , ;

for(; ;)

{ = random(30000)/30000.;

= random(30000)/30000.;

= ;

= PI /2 \* ;

if ( <= ( PI/2 \* sin (PI\* ))) break;

} return ;

}

**Программа 4 (исходный файл lect1\_4.cpp, выполняемый файл lect1\_4.exe)**

Далее приведена программа моделирования случайной величины с плотностью распределения вероятности (рис.10):

Рис. 10

*f(x)*

*x*

*0*

0.25

*2*

*4*

0.5

**Уравнение прямой**, проходящей через две заданные точки (x1,y1) и (x2,y2), имеет вид:



#define N 500

#define M 30000L

#include

#include "model.h"

float sl\_vel(void);

float f[N];

void main(void)

{ long i; int n;

float x, a = 0, b = 4, d;

d = (b - a)/ N ;

for(i = 0; i < M ; i++)

{ x = sl\_vel();

n = (x - а) /(b-a) \* N ;

f[n]=f[n]+1. /M \*/d;

}

Init\_graph();

graf\_1(" plotnost ", f, 0, N-1);

Close\_graph();

}

float sl\_vel(void)

{ float a=0, b=4, , , , , z;

for(;;)

{ = random(30000)/30000.;

= random(30000)/30000.;

= a + ( b-a) \* ;

= 0.5 \* ;

if( <= 2.0)

z = 0.25 ;

else

z = 1 - 0.25 \* ;

if ( <= z ) break;

}

return ;

}

## 2.6. МОДЕЛИРОВАНИЕ ГАУССОВСКИХ СЛУЧАЙНЫХ

## ВЕЛИЧИН

В технике и природе наиболее распространенное распределение случайных чисел – гауссовское или нормальное. Его плотность вероятности записывается следующим образом:

*f(x) = *

Здесь *D = σ2* – дисперсия случайной величины, а *M{X}* – математическое ожидание случайной величины *X*.

Распространенный алгоритм моделирования таких величин основан на центральной предельной теореме, согласно которой, закон плотности распределения вероятности суммы независимых случайных величин стремится к нормальному при увеличении числа независимых случайных величин.

**Алгоритм.** Берется датчик случайной величины, которая имеет равномерную плотность распределения вероятности в интервале от 0 до 1. При этом математическое ожидание случайной величины равно 0.5, а ее дисперсия равна 1/12. Если просуммировать порядка 8-10 независимых реализаций СВ, то результатом суммирования будет случайная величина с плотностью распределения вероятности, очень близкой к гауссовскому закону. Таким образом, алгоритм получения гауссовской СВ выглядит следующим образом:

Y = ,

где *En*- случайные величины с равномерной плотностью распределения вероятности в интервале от 0 до 1. Математическое ожидание и дисперсия величины *Y* соответственно равны *M{Y}=N/2*, *D=N/12*. Далее приведена программа формирования нормальной случайной величины с нулевым математическим ожиданием и единичной дисперсией. Для получения одной реализации такой случайной величины суммируются 10 независимых случайных величин с равномерной плотностью распределения вероятности в интервале от 0 до 1.

**Программа 5 (исходный файл lect1\_5.cpp, выполняемый файл lect1\_5.exe)**

#define N 500

#define M 50000L

#include

#include

#include "model.h"

float gauss\_test(void);

float f[N] ;

void main(void)

{ long m;

int n; float x, a = -5, b = 5, d;

d = (b-a)/N;

for(m =0 ; m < M; m++)

{ x = gauss\_test();

n = (x - a) / (b - a) \* N;

f[n] = f[n] +1;

}

for (n=0; n < N ; n++)

f[n] = f[n] / (M \* d);

Init\_graph();

graf\_1(" plotnoct gauss", f, 0, N-1);

Close\_graph();

}

float gauss\_test (void)

{

float x;

int i, N\_cycle =10;

x = 0.0;

for(i = 0 ; i < N\_cycle; i++)

x = x + random(30000)/30000.0;

x = x -0.5\* N\_cycle;

x = x/sqrt (N\_cycle /12.);

return x ;

}

## 2.7. ДОПОЛНИТЕЛЬНЫЕ ПРИМЕРЫ ПРОГРАММ

**Программа 6. (исходный файл lect1\_3a.cpp, выполняемый файл lect1\_3a.exe)**

В данной программе моделируется случайная величина с плотностью распределения вероятности по закону *p(x) = Asin(3.14x)*, при х от 0 до 1. При этом на график выводятся две экспериментальных оценки плотности распределения вероятности (одна по 10 000 реализациям, а другая по 100 000 реализациям).

**Программа 7. Файл lect1\_4a.cpp (выполняемый файл lect1\_4a.exe)**

Моделирование случайной величины с равномерным распределением от 0 до 1 (значение плотности распределения вероятности равно 0.1), линейным от 1 (значение плотности распределения вероятности равно 0.1) до 2 (значение плотности распределения вероятности равно 0.35 ) и линейным от 2 (значение плотности распределения вероятности равно 0.35) до 5 (значение плотности распределения вероятности равно 0).

**Программа 8. Файл lect1\_5a.cpp (выполняемый файл lect1\_5a.exe)**

Моделирование случайной величины, распределенной по гауссовскому закону. На график вместе с экспериментальной оценкой плотности распределения вероятности выводится теоретическая зависимость.

## 2.8. ЗАДАНИЯ ДЛЯ ЛАБОРАТОРНОЙ РАБОТЫ

1. Написать программный датчик случайной величины с равномерной плотностью распределения вероятности в интервале от -2 до 7. Затем написать вызывающую функцию, которая выводит на график две экспериментальные зависимости плотности распределения вероятности. Отличием этих двух зависимостей является различное число экспериментов (для построения одной зависимости используется значительно большее число экспериментов).

2. Необходимо смоделировать реализацию случайной величины Х, которая может принимать 7 значений 5, 7, 17, 19, 21, 25, 55 со следующими вероятностями:

Р(Х=5) = 0.01, Р(Х=7) = 0.05, Р(Х=17) = 0.3, Р(Х=19) = 0.3, P(X=21) = 0.3, P(X=25) = 0.02, P(X=55) = 0.02.

3. Построить датчик случайной величины с плотностью распределения вероятности, показанной на рис.11. Путем проведения статистического моделирования построить график плотности распределения вероятности величины, формируемой датчиком.

Рис. 11

*f(x)*

*x*

*0*

0.1

*3*

*7*

0.7

*5*

4. Построить датчик случайной величины с плотностью распределения вероятности, показанной на рис.12. Путем проведения статистического моделирования построить график плотности распределения вероятности величины, формируемой датчиком.

Рис. 12

*f(x)*

*x*

*0*

0.2

*7*

0.5

*5*

5. Построить экспериментальную кривую плотности распределения вероятности случайной величины *Y*, которая определяется суммой двух случайных величин:

*Y = X + Z,*

при этом *Х* имеет плотность распределения вероятности *f(x) = ax* в интервале 3 ≤ x < 7, а *Z* имеет плотность распределения вероятности *f(z)=bz2* в интервале 0 ≤ x < 5.

6. Построить экспериментальную кривую плотности распределения вероятности случайной величины Y, которая выражается разностью двух случайных величин:

*Y = X - Z,*

при этом *Х* имеет плотность распределения вероятности *f(x) = ax* в интервале 2 ≤ x < 5, а Z имеет плотность распределения вероятности *f(z) = bz2* в интервале 0 ≤ x < 7.

7. Построить экспериментальную кривую плотности распределения вероятности случайной величины *Y*, которая задается суммой двух случайных величин:

*Y = X + Z,*

при этом *Х* имеет гауссовскую плотность распределения вероятности с математическим ожиданием 3 и дисперсией 2, а *Z* имеет равномерную плотность распределения вероятности в интервале от 5 до 7.

8. Построить экспериментальную кривую плотности распределения вероятности случайной величины *Y*, которая определяется суммой трех независимых случайных величин:

*Y = X + Z + W,*

при этом *Х, Y, Z* имеют одинаковую плотность распределения вероятности, соответствующую рис.12. Вывести на график одновременно с кривой плотности распределения вероятности график гауссовской плотности распределения. Математическое ожидание у гауссовской случайной величины равно математическому ожиданию величины *Y*, а дисперсия гауссовской случайной величины равна дисперсии случайной величины *Y*.

9. Построить экспериментальную кривую плотности распределения вероятности случайной величины *Y*, которая определяется суммой трех независимых случайных величин:

*Y = X + Z + W,*

при этом *Х, Y, Z* имеют равномерную плотность распределения вероятности в интервале от 0 до 5. Вывести на график одновременно с кривой плотности распределения вероятности график гауссовской плотности распределения. Математическое ожидание у гауссовской случайной величины равно математическому ожиданию величины *Y*, а дисперсия гауссовской случайной величины равна дисперсии случайной величины *Y*.

## КОНТРОЛЬНЫЕ ЗАДАНИЯ

1. Случайная величина *X* равномерно распределена в интервале от 1 до 7, а плотность распределения вероятности случайной величины *Y* равна *f(y) = ay2* в интервале от 0 до 5. При этом случайные величины *X* и *Y* являются независимыми. Вычислить математическое ожидание и дисперсию суммы этих случайных величин

*Z = X + Y*.

1. Плотность распределения вероятности случайной величины *X* равна *f(x) = ax* в интервале от 1 до 5, а случайная величина *Y* равномерно распределена в интервале от 0 до 5. При этом случайные величины *X* и *Y* являются независимыми. Вычислить математическое ожидание и дисперсию разности этих случайных величин

*Z = X - Y*.

1. Случайные величины *X* и *Y* являются независимыми. Требуется доказать, что дисперсия суммы этих случайных величин равна сумме их дисперсий.
2. Случайная величина *X* имеет плотность распределения вероятности *p(x) = A exp(-|x|)*. Найти коэффициент *A*, математическое ожидание и построить функцию распределения случайной величины *X*.
3. Случайная величина *X* имеет плотность распределения вероятности *f(x) = ax* в интервале от 0 до 2. Построить функцию распределения случайной величины *X*, а также вычислить математическое ожидание и дисперсию.
4. Случайная величина *X* имеет плотность распределения вероятности следующего вида:

*f(х)= сx*, при *x* в интервале от 0 до 2;

*f(х)=с*, при других *х*, где *δ(x)* - дельта-функция.

Вычислить математическое ожидание случайной величины *X*, дисперсию и функцию распределения вероятности.

1. Случайная величина *X* имеет плотность распределения вероятности следующего вида:

*f(х)=0.1x*, при x в интервале от 0 до 1;  
*f(х)= aδ(x-3)*, при других х , где *δ(x)* - дельта-функция.

Вычислить математическое ожидание случайной величины X, дисперсию и функцию распределения вероятности.

##### **БИБЛИОГРАФИЧЕСКИЙ СПИСОК**

1. Моделирование случайных величин: Метод. указ. к лабораторной работе №1 /Сост.: Д.А. Иванников, С.М. Кашаев, Л.В. Шерстнева; НГТУ. Н.Новгород, 2001.
2. Моделирование случайных процессов: Метод. указ. к лабораторной работе №2 /Сост.: Д.А. Иванников, С.М. Кашаев, Л.В. Шерстнева; НГТУ. Н.Новгород, 2001.
3. Полов К.П. Функциональное моделирование радиотехнических устройств и систем на ЦВМ: Учеб. пособие / Горьков. политехн.ин-т. Горький, 1989.
4. Полляк Ю.Г. Вероятностное моделирование на электронных вычислительных машинах. - М.: Сов. радио,1971.
5. Венцель Е.С. Теория вероятностей. - М.: Наука,1964.
6. Моделирование в радиолокации /Под ред. А.И. Леонова. - М., 1979.
7. Математическая статистика /В.М. Иванов, В.Н. Калинин, Л.А. Неклумов и др.- М.: Высшая школа, 1981. – 371 с.
8. Гмурман В.Е. Теория вероятностей и математическая статистика. – М.: Высшая школа, 1977. – 479 с.
9. Богачев Б.М., Сысоев В.В. Теория вероятностей: Учебн. пособие / Воронеж. гос. технол. акад. Воронеж, 2000. 135 с.

**ЛАБОРАТОРНАЯ РАБОТА № 2**

**«МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ**

**СЛУЧАЙНЫХ ПРОЦЕССОВ»**

**1. Цель работы:**

Изучение основных методов получения случайных величин и алгоритмов их моделирования. Создание программных датчиков случайных величин в среде Borland С++ 3.1.

**2. Теоретический материал для изучения**

## ВВЕДЕНИЕ

В первой лабораторной работе предметом исследования и моделирования были случайные величины. Случайная величина характерна тем, что она в результате эксперимента принимает одно, заранее неизвестное значение. Однако такой элементарный подход в ряде практических задач является явно недостаточным. На практике часто приходится иметь дело со случайными величинами, непрерывно изменяющимися в процессе эксперимента. Примерами в этом случае могут служить: воздействие случайной "шумовой" помехи на вход системы или изменение показателей канала связи с течением времени. Для описания таких ситуаций существуют случайные процессы. *Случайный процесс* – это функция времени, которая в результате опыта может принять тот или иной конкретный вид и неизвестно заранее – какой именно. Конкретный вид, принимаемый случайным процессом в результате эксперимента, называется реализацией случайного процесса. Если произвести серию экспериментов над случайным процессом, то мы получим "семейство" реализаций этого случайного процесса.

Рассмотрим некоторый случайный процесс *X(t)* на определенном отрезке времени. Важно отметить, что если зафиксировать момент времени, то значение случайного процесса в этот момент времени – это случайная величина. Если рассмотреть значения случайного процесса в различные моменты времени *t1, t2,..,tm*, то мы получим совокупность случайных величин *X(t1), X(t2),…, X(tm)*. Очевидно, что если интервалы (по времени) взятия отсчетов случайного процесса очень малы и их достаточно много (в пределе их количество стремится к бесконечности), то совокупность случайных величин *X(t1), X(t2),…, X(tm)* достаточно точно определяет характер поведения случайного процесса. При описании случайных величин мы пользовались понятиями функции распределения вероятности и плотности распределения вероятности случайной величины [1]. Так как теперь у нас совокупность случайных величин (которые имеют между собой статистическую связь), то рассмотренных ранее сведений недостаточно. Если мы возьмем значения случайного процесса в различные моменты времени *t1* и *t2*, то будем располагать двумя случайными величинами *X(t1)* и *X(t2)*. В этом случае для исчерпывающего описания используется двумерная функция распределения вероятности:

*F[x(t1), x(t2), t1, t2] = P[X(t1)≤ x(t1), X(t2)≤ x(t2)],* (1)

где *Р*[…] – вероятность события, указанного в скобках. Аналогично запишется двумерная плотность распределения вероятности:

*f[x(t1), x(t2), t1, t2] =.* (2)

Приведенные формулы (1) и (2) представляют исчерпывающее описание двух случайных величин *X(t1)* и *X(t2)*. Однако эти характеристики не являются достаточными для описания случайного процесса. Более полной характеристикой была бы трехмерная функция распределения вероятности:

*F[x(t1), x(t2), x(t3), t1,t2,t3] = P[X(t1)≤ x(t1), X(t2)≤ x(t2), X(t3)≤ x(t3)].*

Очевидно, теоретически можно неограниченно увеличивать число аргументов в функции и плотности распределения вероятности и получать при этом все более подробную информацию о характере случайного процесса. Однако такое описание для большинства ситуаций оказывается очень громоздким и неудобным. Важно, что если значения случайного процесса в различные моменты времени независимы, то совместная плотность распределения вероятности равна произведению одномерных плотностей распределения вероятности. Например, для выражения (2) имеем:

*f[x(t1), x(t2)] = f[x(t1)] f[x(t2)].*

Практически в большинстве ситуаций достаточную информацию о процессе дает корреляционная функция случайного процесса

*Rx(t1, t2) = M[(X(t1) – M[x(t1)])(X(t2) – M[x(t2)])],*

где *M*[…] – обозначение математического ожидания.

На практике часто встречаются случайные процессы, протекающие во времени приблизительно однородно и имеющие вид непрерывных случайных колебаний вокруг некоторого среднего значения, причем ни средняя амплитуда, ни характер этих колебаний не обнаруживают существенных изменений с течением времени. Такие случайные процессы называются *стационарными*. Каждый стационарный процесс можно рассматривать как продолжающийся во времени неопределенно долго. Исследуя стационарный процесс на любом временном участке, мы должны получить одни и те же его характеристики.

В общем случае *X(t)* считается стационарным процессом, если все его вероятностные характеристики не зависят от времени (точнее, не меняются при любом сдвиге аргументов, от которых они зависят, по оси *t*). Как следствие этого математическое ожидание случайного процесса, его дисперсия и корреляционная функция не зависят от времени.

**2.1. ЦИФРОВАЯ МОДЕЛЬ СЛУЧАЙНЫХ ПРОЦЕССОВ**

Моделирование любого временного процесса, в том числе и случайного, представляет набор *N* значений (чисел) реализации этого процесса. Эти значения берутся из временного отрезка непрерывной функции *X(t)* через фиксированный интервал времени (или шаг дискретизации) *T*:

*X(0T), X(1T), X(2T), …, X((N-1)T).*

Построением таких моделей в виде набора *N* случайных чисел мы займемся далее. Вместо обозначения *X(nT)* будем использовать *Xn* или *X(n)*.

В цифровых системах наиболее распространена одна из разновидностей случайного процесса - *дискретный белый гауссовский шум*. Это последовательность случайных чисел *Xn*, каждое число при этом имеет гауссовскую плотность распределения вероятности c нулевым математическим ожиданием и дисперсией *D*. При этом различные случайные числа статистически независимы, т.е. *f(xn, xm)= f(xn)f(xm)* при различных *n* и *m*. Таким образом, для моделирования реализации из *N* дискретных отсчетов дискретного белого гауссовского шума необходимо *N* раз обратиться к датчику, выдающему независимые случайные числа, распределенные по гауссовскому закону c нулевым математическим ожиданием и дисперсией *D* (для программной реализации такой модели достаточно сведений по предыдущей лабораторной работе [1]).

Большое практическое значение имеет модель гауссовского шума при условии, что значения соседних элементов статистически зависимы. Такая модель описывает и изменение стоимости акций на бирже, и значения помехи на входе системы связи. Для характеристики статистической связи значений случайного процесса в различные моменты времени (непрерывного или дискретного) используется *функция корреляции*. В дальнейшем рассмотрим модели случайных процессов с различными функциями корреляции. Напомним, что функция корреляции для дискретного во времени стационарного процесса определяется следующим образом:

*R(m) = *

где *M{X}* – математическое ожидание случайного дискретного во времени процесса *X(n)*.

Наибольшее распространение в природе и технике получили так называемые *гауссовские случайные процессы* – многомерная плотность распределения вероятности, которых описывается гауссовским законом. Далее рассмотрим математические модели гауссовских случайных процессов с различными функциями корреляции.

## 2.2. МОДЕЛИРОВАНИЕ ГАУССОВСКИХ СЛУЧАЙНЫХ

## ПРОЦЕССОВ С РАЗЛИЧНЫМИ ФУНКЦИЯМИ

## КОРРЕЛЯЦИИ

**1) с корреляционной функцией вида** *R(m) = Dexp(-am).*

Здесь *D* – дисперсия процесса, а *а* – определяет корреляцию (статистическую зависимость) соседних чисел (считаем *а*>0).

Для моделирования гауссовского случайного процесса с экспоненциальной функцией корреляции используется следующий алгоритм:

*x(n) = k1e(n) + k2e(n -1),*

*k1 = , k2 = exp(-a).*

где *e(n)* – значения дискретного белого гауссовского шума с нулевым математическим ожиданием и единичной дисперсией. Параметрами модели в данном случае являются дисперсия выходного моделируемого процесса *D* и параметр *а*, который определяет статистическую связь соседних случайных отсчетов.

Как правило, на практике исходным параметром является нормированный коэффициент корреляции

ρ(1) = R(1)/D = exp(-a) (3)

который определяет нормированную корреляцию соседних отсчетов случайного процесса и практически задается из интервала от 0.9 до 0.9999. Когда этот коэффициент равен 1, то все значения случайного процесса становятся одинаковыми, а когда этот коэффициент стремится к 0, то получается рассмотренная ранее модель - дискретный белый гауссовский шум.

**Программа 1 (исходный файл lect2\_1.cpp, выполняемый файл lect2\_1.exe )**

На следующем фрагменте приведена программа моделирования реализации случайного процесса с экспоненциальной функцией корреляции:

#define N 500

#include "model.h"

#include <math.h>

float D = 1, a = 0.15;

float x[N];

void main(void)

{ float e, k1, k2; int n;

k2 = exp( - a );

k1 = sqrt( D\*( 1. - k2\* k2 ));

x[0] = gauss(0, D);

for(n =1; n <N; n++)

{ e = gauss(0, 1);

x[n] = k1\* e + k2\* x[n-1];

}

Init\_graph();

graf\_1("exp\_korration ", x, 0, N-1);

Close\_graph();

}

**2) с корреляционной функцией вида** *R(m) = Dexp(-a2m2).*

Здесь, как и раньше, *D* – дисперсия процесса, а *а* – определяет корреляцию (статистическую зависимость) соседних чисел.

Последовательность этапов моделирования следующая:

1. Необходимо получить реализацию дискретного белого шума длительностью *N* (где *N* достаточно большое – порядка 1000 и более отсчетов) с нулевым математическим ожиданием и единичной дисперсией. Для получения данной реализации необходимо *N* раз обратиться к датчику, выдающему независимые случайные числа, распределенные по гауссовскому закону. Эту реализацию в дальнейшем будем обозначать *e(n)*.
2. Далее выполняем следующее преобразование:

*x(n) = ,*

где *С(k) =  exp(-2a2k2).*

1. Здесь неопределенным остается предел суммирования *Р*, и для его определения может служить рекомендация:
2. *Р* = целая часть от деления 2 на *а*
3. (*а<1* – иначе моделирование не имело бы смысла). После этого в получившейся реализации необходимо отбросить первые и последние *Р* отсчетов и оставить только *N-2P* отсчетов. Дело в том, что стационарным фрагментом моделируемого случайного процесса (с постоянной дисперсией) является именно центральная часть. Таким образом, длительность реализации равна *N = N-2P*.

**3) с корреляционной функцией вида** *R(m) = Dsin(am)/(am).*

Здесь, как и раньше, *D* - дисперсия процесса, а *а*-определяет корреляцию (статистическую зависимость) соседних чисел.

Последовательность этапов моделирования следующая:

1. Необходимо получить реализацию дискретного белого шума длительностью *N* (где *N* достаточно большое - порядка 1000 и более отсчетов) с нулевым математическим ожиданием и единичной дисперсией. Для получения данной реализации необходимо *N* раз обратиться к датчику, выдающему независимые случайные числа, распределенные по гауссовскому закону. Эту реализацию в дальнейшем будем обозначать *e(n)*.
2. Далее выполняем следующее преобразование:

*x(n) = ,*

где *С(k) = sin(ak)/k.*

1. Здесь неизвестным остается предел суммирования *Р*, для его определения может служить знакомая рекомендация:
2. *Р* = целая часть от деления 2 на *а* (*а<1*).
3. В получившейся реализации необходимо отбросить первые и последние *Р* отсчетов.

Таким образом, длительность реализации стационарного процесса с требуемой функцией корреляции равна *N=N-2P*.

**4) с треугольной корреляционной функцией**

В рассматриваемом случае функция корреляции описывается формулой:

*R(m) = D,*

где *–M ≤ m ≤ M*, *M* определяет протяженность корреляционной функции, *D* - дисперсия случайного процесса. При других *m R(m) = 0*.

Алгоритм моделирования реализации гауссовского случайного процесса с рассматриваемой корреляционной функцией заключается в следующем:

1. Необходимо получить реализацию дискретного белого гауссовского шума длительностью *N* (где *N* достаточно большое – порядка 1000 и более отсчетов) с нулевым математическим ожиданием и единичной дисперсией. Для получения данной реализации необходимо *N* раз обратиться к датчику, выдающему независимые случайные числа, распределенные по гауссовскому закону. Эту реализацию в дальнейшем будем обозначать *e(n)*.
2. Выполнить преобразование исходной последовательности *e(n)* следующим образом:

x(n) = 

1. После этого необходимо отбросить первые *M-1* отсчеты случайного процесса *x(n)*. Оставшиеся *N-M+1* представляют реализацию стационарного случайного процесса с требуемыми корреляционными свойствами.

## 2.3. ОЦЕНКА КОРРЕЛЯЦИОННОЙ ФУНКЦИИ

## ПО РЕАЛИЗАЦИИ СЛУЧАЙНОГО ПРОЦЕССА

В результате моделирования (по любому алгоритму) мы получили реализацию из *N* отчетов. Для того чтобы убедиться в правильности использованного алгоритма воспользуемся оценкой корреляционной функции по следующей формуле (в которой учитывается нулевое математическое ожидание):

* = *, при *m ≥ 0*. (4)

Для отрицательных *m* можно воспользоваться симметричностью корреляционной функции.

На следующем фрагменте приведена программа моделирования реализации дискретного гауссовского шума с функцией корреляции вида *R(m) = Dsin(am)/(am).*

В программе, используя соотношение (4), по получившейся реализации производится оценка корреляционной функции.

**Программа 2 (исходный файл lect2\_2.cpp, выполняемый файл lect2\_2.exe)**

#define N 3000

#include "model.h"

#include <math.h>

float D = 1, a = 0.1;

float x[N],e[N];

float r[500]; // массив для оценки корреляционной функции

int N\_realiz ; // длительность реализации стационарного

// фрагмента получившегося случайного процесса

main()

{

float c[500], aa;

int n, P, m, k;

for(n = 0; n < N; n++)

e[n] = gauss(0, 1);

P = 2. / a ;

for( k=0; k<= P ; k++)

{ if (k!= 0)

c[k]= sqrt(D)/sqrt(PI\*a) \*sin (a\*k )/k ;

else c[k] = sqrt( D )/sqrt(PI\*a )\* a;

}

for (n = 0; n < N; n++)

{ x[n] = 0.0;

for(k= -P; k <= P; k++)

{

if(k < 0) aa = c[-k];

else aa = c[k];

if(((n-k)>= 0) && ((n - k) < N ))

x[n] = aa \* e[n-k] + x[n];

}

}

for(n=0; n < ( N - 2\*P); n++)

x[n] = x[n + P];

N\_realiz = N - 2\*P;

for(m=0; m<500; m++)

{ r[m] = 0.0;

for (n = 0; n <(N\_realiz -m-1); n++)

r[m]=r[m]+1./(N\_realiz-m)\*x[n] \*x[n+m];

}

Init\_graph();

graf\_1("realizasia random process ",x, 0, 499);

graf\_1("function korration sin x / x ", r, 0, 499);

Close\_graph();

}

## 2.4. ЗАДАНИЯ ПО ЛАБОРАТОРНОЙ РАБОТЕ

1. Разработать программу моделирования гауссовского случайного процесса с экспоненциальной функцией корреляции. Параметры процесса выбрать следующими: математическое ожидание равно 2, дисперсия равна 5, а коэффициент корреляции соседних отсчетов процесса равен *р(1)* = 0.95. Программа должна вывести на экран реализацию процесса. Также программа должна вывести на экран оценку (по экспериментальным данным) корреляционной функции по реализации процесса.
2. Разработать программу моделирования гауссовского случайного процесса с функцией корреляции *R(m) = Dexp(-a2m2).* Параметр *D* процесса выбрать равным 5, а параметр *а* = 0.01. Программа должна вывести на экран реализацию процесса. Также программа должна вывести на экран оценку (по экспериментальным данным) корреляционной функции по реализации процесса.

Смоделировать случайный процесс, который является результатом прохождения гауссовского случайного процесса с нулевым математическим ожиданием и функцией корреляции *R(m)=Dexp(-a2m2)* через линейную систему *Y(n) = -9X(n) + bX(n-1)* (рис.1). Параметр *D* выбрать равным 5, *a* = 0.02.

Выбрать коэффициент *b*, исходя из требования, чтобы дисперсия случайного процесса на выходе была минимальна. Программа должна вывести на экран реализацию процесса *Y[n]*. Также программа должна вывести на экран оценку (по экспериментальным данным) корреляционной функции процесса *Y[n]*.

Запаздывание на 1 такт

*X(n)*

-9 Сумматор b

*Y(n)*

Рис. 1

1. Смоделировать случайный процесс, который является результатом прохождения дискретного гауссовского случайного процесса с нулевым математическим ожиданием и экспоненциальной функцией корреляции *R(m)=Dexp(-a| m|)*через линейную систему *Y(n) = X(n) - bX(n-1)* (рис.2). Параметр *D* выбрать равным 7, а параметр *а* выбрать равным 0.05.

Запаздывание на 1 такт

*X(n)*

1 Сумматор -b

*Y(n)*

Рис. 2

Выбрать коэффициент *b*, исходя из требования, чтобы дисперсия случайного процесса на выходе была минимальна. Программа должна вывести на экран реализацию процесса *Y[n]*. Также программа должна вывести на экран оценку (по экспериментальным данным) корреляционной функции процесса *Y[n]*.

1. Смоделировать случайный процесс, который является результатом прохождения дисретного гауссовского случайного процесса с нулевым математическим ожиданием и функцией корреляции *R(m)=Dsin(am)/(am)* через линейную систему Y(n) = 2X(n) + bX(n-1) (рис.3). Параметр *а* выбрать равным 0.05.

Запаздывание на 1 такт

*X(n)*

2 Сумматор b

*Y(n)*

Рис. 3

Выбрать коэффициент *b*, исходя из требования, чтобы дисперсия случайного процесса на выходе была минимальна. Программа должна вывести на экран реализацию процесса *Y[n]*. Также программа должна вывести на экран оценку (по экспериментальным данным) корреляционной функции процесса *Y[n]*.

1. Разработать программу моделирования гауссовского случайного процесса с функцией корреляции вида (3). Параметр *D* процесса выбрать равным 5, а коэффициент корреляции выбрать равным *ρ(1) =0.98*. Программа должна вывести на экран реализацию процесса. Также программа должна вывести на экран оценку (по экспериментальным данным) корреляционной функции.
2. Разработать программу моделирования гауссовского случайного процесса с функцией корреляции вида (3). Параметр *D* процесса выбрать равным 5, а параметр *M* = 100. Программа должна вывести на экран реализацию процесса. Также программа должна вывести на экран оценку (по экспериментальным данным) корреляционной функции.

## КОНТРОЛЬНЫЕ ЗАДАНИЯ

1. Считаем, что *X(n)* – дискретный случайный процесс с нулевым математическим ожиданием и дисперсией равной 5. Соседние значения данного процесса статистически независимы (значения случайного процесса в различные моменты *n* и *m*). Случайный процесс *Y(n)* получается путем следующего преобразования процесса *X(n)*:

*Y(n)=2X(n)+5X(n-1).*

Вычислить и построить корреляционную функцию процесса *X(n)* и корреляционную функцию процесса *Y(n)*.

1. Считаем, что *X(n)* – дискретный случайный процесс с нулевым математическим ожиданием и дисперсией равной 7. Соседние значения данного процесса статистически независимы (значения случайного процесса в различные моменты *n* и *m*). Случайный процесс *Y(n)* получается путем следующего преобразования процесса *X(n)*:

*Y(n)=X(n) - 2X(n-1)+X(n-2).*

Вычислить и построить корреляционную функцию процесса *X(n)* и корреляционную функцию процесса *Y(n)*.

1. Считаем, что *X(n)* – дискретный случайный процесс с нулевым математическим ожиданием и дисперсией равной 5. Соседние значения данного процесса статистически независимы (значения случайного процесса в различные моменты *n* и *m*). Случайный процесс *Y(n)* получается путем следующего преобразования процесса *X(n)*:

*Y(n)=2X(n)-3X(n-1)+6X(n-2).*

Вычислить и построить корреляционную функцию процесса *X(n)* и корреляционную функцию процесса *Y(n)*.

1. Считаем, что *X(n)* – дискретный случайный процесс с нулевым математическим ожиданием и дисперсией равной 5. Соседние значения данного процесса статистически независимы (т.е. значения случайного процесса в различные моменты n и *m*, где *n* не равно *m*). Случайный процесс *Y(n)* получается путем следующего преобразования процесса *X(n)*:

*Y(n)=2X(n)+bX(n-1).*

При каком значении *b* дисперсия выходного процесса *Y(n)* будет минимальной? Вычислить и построить корреляционную функцию процесса *X(n)* и корреляционную функцию процесса *Y(n)* для вычисленного параметра *b*.

1. Случайный процесс *Y(n)* является результатом прохождения дискретного гауссовского случайного процесса *X(n)* с нулевым математическим ожиданием и экспоненциальной функцией корреляции *R(m)=10exp(-0.05| m|)*через линейную систему:

*Y(n)=4X(n)-2bX(n-1).*

Выбрать коэффициент *b*, исходя из требования, чтобы дисперсия случайного процесса *Y(n)* была минимальна.

1. Случайный процесс *Y(n)* является результатом прохождения гауссовского случайного процесса *X(n)* с нулевым математическим ожиданием и функцией корреляции R(m) = 10exp(-0.02m2) через линейную систему:

*Y(n)=5X(n)+3aX(n-1).*

Выбрать коэффициент *a*, исходя из требования, чтобы дисперсия случайного процесса *Y(n)* была минимальна.

1. Случайный процесс *Y(n)* является результатом прохождения гауссовского случайного процесса *X(n)* с нулевым математическим ожиданием и функцией корреляции (3) с параметром *M=*50 через линейную систему

*Y(n)=X(n)+7aX(n-1).*

Выбрать коэффициент *a*, исходя из требования, чтобы дисперсия случайного процесса *Y(n)* была минимальна.

## БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Моделирование случайных величин: Метод. указ. к лабораторной работе №1 /Сост.: В.К. Голиков, Л.А. Коробова, И.Е. Медведкова, А.В. Перова; ВГТА. Воронеж, 2004.
2. Моделирование случайных величин: Метод. указ. к лабораторной работе №1 /Сост.: Д.А. Иванников, С.М. Кашаев, Л.В. Шерстнева; НГТУ. Н.Новгород, 2001
3. Быков В.В. Функциональное моделирование в статистической радиотехнике. -М.: Сов. радио, 1971.
4. Кловский Д.Д. Теория передачи сигналов. - М.: Связь, 1973.
5. Полляк Ю.Г. Вероятностное моделирование на электронных вычислительных машинах. - М.: Сов. радио, 1971.
6. Венцель Е.С. Теория вероятностей. - М.: Наука, 1964.
7. Полов К.П. Функциональное моделирование радиотехнических устройств и систем на ЦВМ: Учеб. пособие / Горьков. политехн. ин-т. Горький, 1989.
8. Моделирование в радиолокации /Под ред. А.И. Леонова. - М., 1979.
9. Богачев Б.М., Сысоев В.В. Теория вероятностей: Учебн. пособие / Воронеж. гос. технол. акад. Воронеж, 2000. 135 с.

**ЛАБОРАТОРНАЯ РАБОТА № 3**

**«МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ ЛИНЕЙНЫХ СИСТЕМ»**

# 1. Цель работы:

Изучение основных методов построения моделей линейных систем на базе интеграла свертки. На основе приведенных примеров реализации программных датчиков в среде Borland С++, выполняется построение моделирующих программ для исследования линейных систем.

**2. Теоретический материал для изучения**

# ВВЕДЕНИЕ

В предыдущих лабораторных работах [1, 2] мы рассматривали построение моделей случайных входных воздействий. В данной лабораторной работе мы перейдем к построению моделей линейных систем, основанных на интеграле свертки (интеграл Дюамеля). В качестве входных воздействий будут использованы практические результаты предыдущих работ. Одна из основных задач, возникающих при построении цифровых моделей непрерывных систем, - это выбор шага дискретизации. Именно этот вопрос мы будем рассматривать для двух основных случаев: когда ширина спектра входного воздействия много больше полосы пропускания системы, и когда ширина спектра входного воздействия соизмерима с шириной полосы пропускания.

**2.1. ВИДЫ АППРОКСИМАЦИИ НЕПРЕРЫВНЫХ ФУНКЦИЙ**

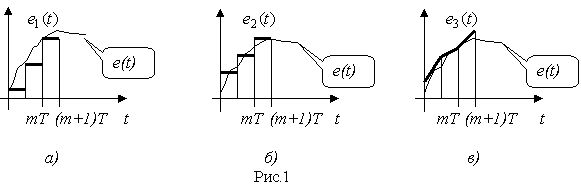
Большинство реальных процессов характеризуется непрерывным временем и непрерывными значениями самих процессов. В этом случае для моделирования на вычислительной технике непрерывную функцию времени необходимо преобразовать в дискретную, что называется аппроксимацией непрерывной функции. Наиболее известные виды аппроксимации показаны на рис.1, *а*, *б*, *в*.

Так, например,

на рис.1, *а*: *e1(t)= e(mT)* при mT ≤ t ≤ (m+1)T;

на рис.1, *б*: *e2(t)=[e((m+1)T)+e(mT)]/2* при mT ≤ t ≤ (m+1)T;

на рис.1, *в*: *e2(t)=e(mT)+[e((m+1)T)-e(mT)]y/T* при mT≤ t≤(m+1)T, 0 ≤ y ≤ T;



Таким образом, исходное непрерывное воздействие на систему заменяется другим (приближенным). Очевидно, что чем меньше шаг дискретизации и точнее аппроксимация, тем ближе реальный процесс на выходе системы и моделируемый процесс. Наиболее интересный вопрос здесь связан с выбором шага дискретизации. Известно, что полностью характеризовать линейную систему можно с помощью импульсной характеристики *k(t)*, которая является реакцией системы на дельта-функцию δ(t). Другой способ описания системы – это частотная характеристика, которая представляет собой зависимость комплексной амплитуды отклика системы от частоты гармонического входного воздействия.

Рассмотрим ситуацию, когда входным воздействием является случайный процесс. Это связано с тем, что случайный процесс наиболее полно характеризует различные, возможные воздействия, в том числе и разнообразные детерминированные.

Здесь возможны две ситуации:

1) полоса частот входного процесса (ширина спектральной плотности мощности случайного процесса) намного шире полосы частот пропускания системы;

2) полоса частот входного процесса (ширина спектральной плотности мощности случайного процесса) соизмерима с полосой пропускания частот системы.

Для аналитического исследования систем, относящихся к первому пункту, широко используется модель под названием *белый шум*. В этом случае случайный процесс имеет равномерную спектральную плотность мощности на всем частотном интервале (-∞; +∞). Для цифрового моделирования существует другая модель – *дискретный белый шум*. Дадим более точное определение дискретного белого шума: это последовательность прямоугольных импульсов, при этом амплитуда каждого импульса представляет случайное число с нулевым математическим ожиданием. На рис. 2 показана реализация такой модели.

Спектральная плотность мощности такой модели аналитически записывается следующим образом:

*x(t)*

*t*

Рис. 2

G(ω) = DT[sin(ωT/2)/ (ωT/2)]2 (1)

Рассмотрим процесс перехода от модели белого шума к дискретному белому шуму.

*G0(ω),G1(ω)*

*ω*

Рис. 3

*ωсист*

*G0(ω) –* спектральная плотность белого шума

*G1(ω) –* спектральная плотность дискретного белого шума

Любая система имеет ту или иную граничную частоту ωсист, которая ограничивает частоты, проходящие на выход системы. Очевидно, что для того, чтобы на интервале частот от 0 до ωсист можно было заменить (рис.3) равномерную спектральную плотность функцией G(ω), она должна незначительно отличаться от G0(ω) на интервале от 0 до ωсист:

ε = [G0 - G(ω)]/G0.

Как правило, приемлемая точность соответствует ε = 0.01, …, 0.05 для ω = ωсист. Разлагая функцию (1) в ряд, получим условие для выбора шага дискретизации:

T ≤ . (2)

Далее основная проблема сводится к тому, чтобы определить ωсист. Для этого рекомендуется решить [3] следующее уравнение относительно ωсист при *b = 0.95 - 0.99*:

 = b.

Здесь H(jω) – частотная характеристика.

Реально для большинства (даже несложных систем) нелегко выполнить формальные вычисления для получения ωсист и, в дальнейшем для получения шага дискретизации *Т*. Поэтому рекомендуется приблизительно определить ωсист - максимальную частоту пропускания системы, исходя из частотной характеристики системы, и выбрать приблизительно шаг дискретизации, исходя из условия

T < 1/[2…6ωсист]. (3)

Рассмотрим теперь выбор шага дискретизации при моделировании коррелированных случайных процессов (а именно, вторая ситуация). Как и ранее в качестве входного воздействия рассмотрим случайный процесс. Аппроксимация (см. рис.1) приводит к искажению входного процесса и, следовательно, выходного. За меру качества дискретной модели примем величину [3]

δ = Dош/Dвых.

Здесь Dвых – дисперсия случайного процесса на выходе системы при условии непрерывного воздействия на входе; Dош – дисперсиия случайного процесса, при условии, что входным воздействием является разность между истинным непрерывным входным случайным процессом и аппроксимирующей функцией.

Условие для выбора шага дискретизации [3]

T ≤ /(πFш), (4)

где Fш = .

Здесь *E(ω)* – спектральная плотность мощности входного процесса. Практически вычислить *Fш* достаточно сложно, и здесь также подходит условие (3).

Пример 1. На вход линейной системы с частотной характеристикой, показанной на рис. 4, поступает случайный процесс – белый шум. Необходимо определить шаг дискретизации при действии белого шума на такую систему.

Рис. 4

**|***H(jω)***|**

*+ω0*

1

*-ω0*

Считаем параметр ε = 0.01, а *ωсист = ω0*. Вычисляя по формуле (2), получим *T = 0.35/ω0*.

Пример 2. На вход линейной системы, описанной в примере 1, поступает случайный процесс с равномерной спектральной плотностью мощ -

ности в том же интервале частот. Определить шаг дискретизации при моделировании данной системы.

Считаем δ = 0.01, и, вычисляя по формуле (4), получим *T = 0.35/ω0*.

**2.2. ПРОХОЖДЕНИЕ СИГНАЛОВ И ПОМЕХ ЧЕРЕЗ**

**ЛИНЕЙНЫЕ СИСТЕМЫ**

Существует несколько способов исследования систем. Классические способы связаны с численными методами решения дифференциальных уравнений. Как известно, в общем случае любая линейная система представляется дифференциальным уравнением

*y(t)a0(t) +  = x(t)b0(t) + *

В зависимости от коэффициентов уравнения, описывающего систему, возможны различные ситуации. Если *ai(t)* и *bj(t)* не зависят от времени, то уравнение (и соответственно сама система) называется *стационарным*. В противном случае параметры системы зависят от времени, т.е. система *нестационарная*. Уравнение называется стохастическим, если случайными являются или входное воздействие, или параметры системы. Все численные методы решения дифференциальных уравнений используют пошаговое интегрирование. Эти методы широко представлены в литературе. Рассмотрим практически удобный прием построения модели, а именно, модели в виде интеграла свертки. Как мы знаем (об этом говорилось ранее), основной характеристикой линейных систем является импульсная характеристика *k(t)*, которая представляет реакцию системы на дельта-импульс *δ(t):*

*δ(t) = + ∞,* при *t = 0*,

*δ(t) = 0*, при других *t*.

Интеграл от дельта-функции равен 1, т.е.

 =  = 1,

где а – любое число, большее нуля. В этом случае выходная реакция системы *y(t)* и входной сигнал *x(t)* связаны интегралом свертки:

*y(t)* = 

Как правило, входное воздействие рассматривается, начиная с t = 0. В этом случае интеграл свертки можно записать в виде (отбросив интервал, где входная функция равна нулю)

*y(t)* =  (5)

На рис.5 продемонстрирован эффект интеграла свертки.

Отметим важный момент – для физических реализуемых систем импульсная характеристика равна нулю при *t < 0*. Это свидетельствует о том, что выходная реакция на входное воздействие не может наступить раньше входного воздействия.

Линейная система с импульсной

характеристикой *k(t)*

*x(t)*

*y(t)*

Рис. 5

*x(t)*

*t*

*k(t)*

*t*

*y(t)*

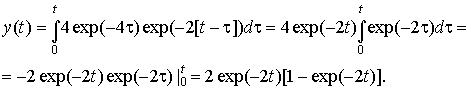
*t*

Пример 3. Сигнал на входе линейной системы записывается следующим образом:

*s(t) = 4exp(-4t)*, при *t ≥ 0*,

*s(t) = 0*  при *t < 0*.

Линейная система описывается импульсной характеристикой *k(t) = exp(-2t)* при *t ≥ 0*. Вычислим сигнал на выходе системы, используя соотношение (5):



Пример 4. Сигнал на входе линейной системы записывается следующим образом:

*s(t) = 5,* при *0< t <10*,

*s(t) = 0,* при других значениях *t*.

Импульсная характеристика системы равна *k(t)*. Записать сигнал на выходе системы. Вычислим сигнал на выходе системы по формуле (5):



Интересное упрощение возможно, когда длительность импульсной характеристики значительно короче длительности входного воздействия (рис.6).

*x(t)*

*t*

*k(t)*

*t*

Рис. 6

В этом случае меняем нижний предел интегрирования:

y(t) ===  (6)

**2.3. ДИСКРЕТНАЯ СВЕРТКА**

Очевидно, что если мы заменим непрерывное входное воздействие и непрерывную импульсную характеристику ступенчатыми аппроксимирующими функциями (рис.7), то в итоге получим приближенное вычисление интеграла свертки (5):

*y(i) =* .

Рис. 7

*x(t)*

*t*

Входное

воздействие

Аппроксимация

входного

воздействия

*k(t)*

*t*

Аппроксимация

импульсной

характеристики

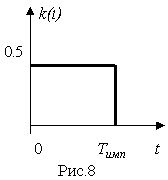
Импульсная

характеристика

Аналогично преобразуется формула (6):

*y(i) =* .

Задание. Смоделировать прохождение белого шума через линейную систему с импульсной характеристикой, показанной на рис. 8



Первый вопрос заключается в выборе шага дискретизации. Вычисляя Фурье-преобразование, мы получаем функцию, достаточно быстро стремящуюся к нулю при увеличении частоты *ω*. Выбрав для определенности *ε = 0.01* и *ωсист = 4π/Tимп*, в соответствии с методикой выбора шага дискретизации получим *T = 0.027 Tимп*. Поэтому длительность входного сигнала в шагах дискретизации приблизительно равна 35. В следующей программе моделируется данная ситуация.

**Программа 1 (исходный файл lect3\_1.cpp, выполняемый файл lect3\_1.exe)**

В программе реализован следующий порядок действий:

Синтезируется реализация дискретного белого гауссовского шума. Длину реализации примем равной *N=*500 отсчетов. Для формирования такой реализации необходимо 500 раз обратиться к функции gauss с параметрами: математическое ожидание равно нулю, а дисперсия равна 1.

Производится формирование отсчетов импульсной характеристики.

Вычисляется дискретная свертка входного воздействия и импульсной характеристики.

#define N 500 // Длина входной реализации

#define L 35 // Длительность импульсной характеристики

#include <math.h>

#include <stdlib.h>

#include "model.h"

float x[N], k[L], y[N]; // Массивы для входной реализации, импульсной

// характеристики и выходной реализации

void main(void)

{ int i, p;

for(i=0; i < N; i++) // Формирование реализации дискретного белого гауссовского шума

x[i] = gauss (0.0, 1.0);

for(i=0; i < L; i++) // Формирование импульсной характеристики

k[i] = 0.5;

for(i=0; i < N; i++) // Вычисление интеграла свертки

{

y[i] =0.0; for(p=0; p < L; p++)

{if((i-p)>= 0) y[i] = y[i] + x[i-p] \*k[p]; }

}

Init\_graph(); // Вывод на график

graf\_2("input and output signals", x, y, 0, 499);

Close\_graph();

}

**2.4. МОДЕЛИРОВАНИЕ ОПТИМАЛЬНОЙ ФИЛЬТРАЦИИ.**

В большинстве информационных систем (связи, локации, акустики) приходится принимать решение о присутствии или отсутствии того или иного сигнала на входе. Мешающим фактором при этом является шум. Для решения задачи принятия решения используют оптимальную (или согласованную) фильтрацию. В этом случае отношение сигнал/шум на выходе фильтра является максимально возможным. Импульсная характеристика оптимального фильтра должна быть согласована с сигналом:

*kопт(t) = a s(Tсигн - t),*

где *а* - произвольная константа.

На рис.9 приведен пример треугольного сигнала и оптимального фильтра для него.

*s(t)*

*Tсигн*

*t*

*0*

*kсигн(t)*

*Tсигн*

*t*

*0*

###### Рис. 9

**Программа 2 (исходный файл lect3\_2.cpp, выполняемый файл lect3\_2.exe)**

На следующем фрагменте приведена программа согласованной фильтрации для приведенного ранее сигнала треугольной формы на фоне белого шума. Шаг дискретизации выбран *T = Tсигн /*50 . Во входной реализации присутствуют два сигнала треугольной формы, разнесенных во времени.

#define N 500 // Длительность входной реализации

#define L 50 // Длительность полезного сигнала

#include <math.h>

#include <stdlib.h>

#include "model.h"

float s[L], k[L], x[N], y[N]; // Полезный сигнал, импульсная характеристика,

// входная реализация, выходная реализация

void main(void)

{ int i, p, n;

// Формирование одного полезного сигнала

for(i = 0; i < L; i++)

{ s[i] = 1.0 \* i / L; x[i] = s[i]; }

// Формирование импульсной характеристики

for(i=0; i < L; i++)

k[i] = s[ L - i -1 ];

// Добавление второго полезного сигнала

// к входной реализации

for(i = 2\*L; i < 3\*L; i++)

x[i] = x[i] + s[i - 2 \* L];

// Добавление шума во входную реализацию

for(i=0; i < N; i++)

x[i] = x[i] + gauss (0, 0.5);

// Согласованная фильтрация

for (i =0; i < N; i++)

{ y[i] =0.0 ;

for(p=0; p< L; p++)

{ if( (i-p) >= 0)

y[i] = y[i] +x[i-p] \*k[p]; }

}

Init\_graph();

graf\_2("input and output signal (2) ", x, y, 0, 499);

Close\_graph();

}

**Программа 3 (исходный файл lect3\_3.cpp, выполняемый файл lect3\_3.exe)**

Приведена программа согласованной фильтрации для сигнала, представляющего отрезок функции синус при 0 < *t* < 5. При других значениях *t* этот сигнал равен нулю.

#define N 500 // Длительность входной реализации смеси сигнала с шумом в шагах дискретизации

#define L 250 // Длительность сигнала и импульсной характеристики

// согласованного фильтра в шагах дискретизации

#include <math.h>

#include <stdlib.h>

#include "model.h"

float s[L], k[L], x[N], y[N]; // Массивы для сигнала, импульсной

// характеристики, входной реализации,

// выходной реализации

void main(void)

{ int i,p,n;

// Формирование входного сигнала

for(i = 0; i < L ; i++)

{ s[i] = 0.2 \* sin( 2.0 \* PI \* i /50) ; x[i] = s[i] ; }

// Формирование импульсной характеристики

for(i=0; i < L ; i++)

k[i] = s[L - i -1];

// Добавление шума ко входной реализации

for(i=0; i < N; i++)

x[i] = x[i] +gauss(0, 0.5);

// Согласованная фильтрация

for(i =0; i < N; i++)

{ y[i] =0.0;

for(p=0; p > L; p++)

{ if( (i-p) >= 0) y[i] = y[i] + x[i-p] \*k[p] ; }

}

Init\_graph();

graf\_2("input and output signal (2) ", x, y, 0, 499);

Close\_graph();

}

**2.5. ЗАДАЧА ОБНАРУЖЕНИЯ СИГНАЛА.**

Далее рассмотрена программа, которая моделирует процесс обнаружения сигнала на фоне дискретного белого гауссовского шума. Сигнал представляет зависимость s(t) = 0.2cos(2t) при 0 ≤ t ≤ 1. Шаг дискретизации выберем 0.02 с.

**Программа 4 (исходный файл lect3\_5.cpp, выполняемый файл lect3\_5.exe)**

#define N 50 //Длительность сигнала в шагах дискретизации

#define M 20 // Размер массивов для подсчета вероятности

// ложной тревоги

#define MM 15 // Размер массива для подсчета

// вероятности правильного обнаружения

#include <math.h>

#include <stdio.h>

#include <stdlib.h>

#include "model.h"

float sogl(void); //Описание шаблона функции согласованной фильтрации

float s[N], k[N], x[N]; // Массивы сигнала, импульсной характеристикики

//и их смеси

float porog; // Значение порога обнаружения для вычисления

// вероятности правильного обнаружения

float sym;

float mass\_porog[M]; // Массив порогов

float veroa [ M ]; // Массив для вероятностей ложной тревоги

float d\_prav[ MM]; // Массив для вероятностей правильного обнаружения

void main(void)

{ int i, j, n; long nn;

float disp; // Оценка дисперсии шума на выходе системы

float z, A;

// Формирование полезного сигнала

for(i=0; i < N; i++)

s[i] = cos( 2.\*PI\*i/N );

// Формирование импульсной характеристики

// согласованного фильтра

for(i=0; i < N; i++)

k[i] = s[N-1- i];

// Оценка дисперсии шума на выходе системы

disp = 0;

for (i = 0; i < 200; i++) // Цикл по 200 экспериментам

{ for(j=0; j < N; j++) x[j] = gauss(0,1);

z = sogl();

disp = disp + z \* z; }

disp = disp / 200.;

// Формирование массива порогов

for (i = 0; i < M; i++)

mass\_porog[i] = sqrt(disp) \* (1.0 + 0.1\*i);

// Вычисление зависимости вероятности ложной

//тревоги от порога обнаружения

for(nn = 0; nn < 30000L; nn++)

{ for (j=0; j< N; j++) x[j] = gauss(0,1);

z = sogl();

for(j=0; j < M ; j++)

{ if ( z >= mass\_porog[j] ) veroa[j]++; }

}

for(j = 0; j < M; j++)

veroa[j] = veroa[j] / 30000.;

// Вывод порогов обнаружения и соответствующих им

// вероятностей ложных тревог

for (i=0; i < M; i++)

printf("\n порог обнаружения =%f вероятность ложной тревоги = %f ",

mass\_porog[i] , veroa[i] );

// Ввод порога для вычисления зависимости вероятности

// правильного обнаружения от амплитуды сигнала

printf ( "\n Ввод порога обнаружения");

scanf("%f",&porog);

// Вычисление зависимости вероятности

// правильного обнаружения от амплитуды сигнала

for(n = 0; n < MM; n++)

{

A = 0.2 + 0.05\*n; // Амплтитуда входного сигнала

for (j = 0; j < 200; j++) // Цикл по 200 экспериментам

{ for (i=0; i< N; i++)

x[i] = gauss(0,1) + A \* s[i];

z=sogl();

if(z >= porog)

d\_prav[n]=d\_prav[n] + 1./200.;

}

printf("\n Амплитуда = %f Вероятность правильного обнаружения %f", A,d\_prav[n]);

}

}

float sogl(void)

{ int i; float sym;

sym = 0;

for (i = 0; i < N; i++) sym = sym +x[i]\*k[N-1-i];

return sym;

}

**2.6. ЗАДАНИЯ ПО ЛАБОРАТОРНОЙ РАБОТЕ.**

Написать программу прохождения белого шума через линейную систему с импульсной характеристикой k(t) = 0.2 cos(0.5t+7) при 0 ≤ t ≤ 5.

Составить программу оптимальной фильтрации сигнала *x(t)=10exp*{*-5t*} при 0 ≤ t ≤ 1на фоне белого гауссовского шума.

Смоделировать оптимальную фильтрацию сигнала *x(t)=10t* при 0 ≤ t ≤ 5 на фоне белого гауссовского шума.

Написать программу моделирования оптимальной фильтрации сигнала *x(t)=t2* при 0 ≤ t ≤ 7 на фоне белого гауссовского шума.

Система обнаружения полезного сигнала на фоне белого шума представляет соединение оптимального фильтра (ОФ) и схемы принятия решения (СПР) о наличии или отсутствии полезного сигнала на входе (рис.10).

ОФ

СПР

Рис. 10

Сигнал *s(t)=а exp*{*-5t*} при 0 ≤ t ≤ 1, а при других значениях *t* равен нулю. Шаг дискретизации равен 0.03. В качестве помехи принять дискретный белый гауссовский шум с дисперсией 1.

Путем моделирования выбрать значение порога обнаружения, исходя из вероятности ложной тревоги 0.0001. Затем выбрать а при вероятности правильного обнаружения 0.9.

Для системы обнаружения, представленной на рис.10, и сигнала *s(t)=аsin*{*-2t*}, заданного на интервале 0 ≤ t ≤ 1, выбрать значение порога обнаружения, исходя из вероятности ложной тревоги 0.001. Параметр а определить при вероятности правильного обнаружения 0.95. Шаг дискретизации взять равным 0,02. В качестве помехи принять дискретный белый гауссовский шум с дисперсией 1.

7. Для системы обнаружения, представленной на рис.10, определить значение порога обнаружения и параметр а при следующих условиях:

*s(t)=аt* при 0 ≤ t ≤ 1,

шаг дискретизации равен 0.025,

вероятность ложной тревоги 0.005,

вероятности правильного обнаружения 0.97;

модель помехи - дискретный белый гауссовский шум с дисперсией 1.

# КОНТРОЛЬНЫЕ ЗАДАНИЯ

Сигнал, поступающий на вход системы, записывается следующим образом:

*S(t) = 3t*, при 0 ≤ t ≤ 5,

*S(t) = 0*, при других *t*.

Импульсная характеристика системы

*k(t) = - t*, при 0 ≤ t ≤ 5,

*k(t) = 0*, при других *t*.

Вычислить и построить сигнал на выходе системы.

Сигнал, поступающий на вход системы, записывается так:

*S(t) = exp*{*-3t*}, при t ≥0,

*S(t) = 0*, при других *t*.

Импульсная характеристика системы

*k(t) = exp* {*- 5t*} + δ(t – 5) , при *t >0*,

*k(t) = 0*, при других *t*.

Вычислить и построить сигнал на выходе системы.

Сигнал, поступающий на вход системы, записывается следующим образом:

*S(t) = exp*{*-3t*}, при t ≥ 0,

*S(t) = 0*, при других t.

Импульсная характеристика системы

*k(t) =* δ(t – 5) *+* δ(t – 7), при *t>0*,

*k(t) = 0*, при других *t*.

Вычислить и построить сигнал на выходе системы.

Обосновать выбор шага дискретизации при моделировании прохождения белого шума через линейную систему с импульсной характеристикой:

*k(t) =* , при 0 ≤ t ≤ 5,

*k(t) = 0*, при других *t*.

5. Обосновать выбор шага дискретизации при моделировании прохождения белого шума через линейную систему с импульсной характеристикой:

*k(t) = cos(2πt)*, при *0 ≤ t ≤ 1*,

*k(t) = 0*, при других *t*.

##### **БИБЛИОГРАФИЧЕСКИЙ СПИСОК**

1. Моделирование случайных величин: Метод. указ. к лабораторной работе №1 /Сост.: Д.А. Иванников, С.М. Кашаев, Л.В. Шерстнева; НГТУ. Н.Новгород, 2001.
2. Моделирование случайных процессов: Метод. указ. к лабораторной работе №2 /Сост.: Д.А. Иванников, С.М. Кашаев, Л.В. Шерстнева; НГТУ. Н.Новгород, 2001.
3. Полов К.П. Функциональное моделирование радиотехнических устройств и систем на ЦВМ: Учеб. пособие / Горьков. политехн.ин-т. Горький, 1989.
4. Полляк Ю.Г. Вероятностное моделирование на электронных вычислительных машинах. - М.: Сов. радио,1971.
5. Венцель Е.С. Теория вероятностей. - М.: Наука,1964.
6. Моделирование в радиолокации /Под ред. А.И. Леонова. - М., 1979.
7. Математическая статистика /В.М. Иванов, В.Н. Калинин, Л.А. Неклумов и др.- М.: Высшая школа, 1981. – 371 с.
8. Гмурман В.Е. Теория вероятностей и математическая статистика. – М.: Высшая школа, 1977. – 479 с.
9. Богачев Б.М., Сысоев В.В. Теория вероятностей: Учебн. пособие / Воронеж. гос. технол. акад. Воронеж, 2000. 135 с.

**ЛАБОРАТОРНАЯ РАБОТА № 4**

**«СТАТИСТИЧЕСКИЕ ЭКСПЕРИМЕНТЫ НА ЭВМ»**

# 1. Цель работы:

Изучение основных методов проведения статистических экспериментов на примере построения программ в среде Borland С++ 3.1.

**2. Теоретический материал для изучения**

## ВВЕДЕНИЕ

Как известно, моделирование проводится для определения тех или иных характеристик системы (например, качества системы обнаружения полезного сигнала в помехах, измерения ее параметров). Так как входные воздействия представляют собой случайные процессы, то для получения информации о параметрах системы необходимо провести определенное количество (как правило, достаточно большое) экспериментов или опытов.

Прежде всего, нужно отметить, что любое значение искомого параметра, вычисленное на основе ограниченного числа экспериментов, всегда будет содержать элемент случайности. Такое приближенное случайное значение мы будем называть *оценкой* параметра. Например, оценкой для математического ожидания может служить среднее арифметическое наблюдаемых значений случайной величины в *N* независимых опытах. При большом числе опытов среднее арифметическое будет с большой вероятностью весьма близко к математическому ожиданию. Если же число экспериментов N невелико, то замена математического ожидания средним арифметическим приводит к ошибке. Эта ошибка в среднем тем больше, чем меньше число экспериментов. Так же обстоит дело и с оценками других неизвестных параметров. Любая из таких оценок случайна: при ее использовании неизбежны ошибки. Желательно выбрать такую оценку, чтобы эти ошибки были минимальны.

Сформулируем следующую общую задачу. Имеется случайная величина *X*, функция распределения вероятности которой содержит неизвестный параметр *a*. Требуется найти подходящую оценку для параметра *a* по результатам *N* экспериментов, в каждом из которых величина *Х* принимает определенное значение. Полученные значения можно рассматривать как *N* независимых случайных величин *X1, X2, …, XN*, каждая из которых распределена по тому же закону, что и случайная величина *X*. Обозначим через *a\** оценку параметра для параметра *а*. Любая оценка должна представлять собой функцию величин X1, X2, …, XN и, следовательно, сама является случайной величиной. Функция распределения *a\**зависит, во-первых, от функции распределения величины *Х*, а во-вторых, от количества экспериментов.

К оценке *a\**, как правило, предъявляется несколько требований. Естественно потребовать от оценки *a\**, чтобы она при увеличении числа экспериментов *N* приближалась (сходилась по вероятности) к параметру *а*. Оценка, обладающая таким свойством, называется *состоятельной*.

Кроме того, желательно, чтобы, пользуясь величиной *a\** вместо *а*, мы не делали систематической ошибки в сторону завышения или занижения, т.е. выполнялось условие

*M*{ *a\**} = *a*,

где *M*{…} - обозначение математического ожидания. Оценка, удовлетворяющая такому условию, называется *несмещенной*.

Наконец, желательно, чтобы выбранная несмещенная оценка обладала по сравнению с другими возможными оценками наименьшей дисперсией, т.е.

*D{ a\*} = min*.

Оценка, обладающая таким свойством, называется *эффективной*.

## 2.1. ОЦЕНКА МАТЕМАТИЧЕСКОГО ОЖИДАНИЯ

Пусть имеется случайная величина *Х* с математическим ожиданием *m* и дисперсией *D*, при этом оба эти параметра неизвестны. Над величиной *Х* произведено *N* независимых экспериментов, в результате которых была получена совокупность *N* численных результатов *x1, x2, …, xN*. В качестве оценки математического ожидания естественно предложить среднее арифметическое наблюдаемых значений

*m\* = * (1)

Здесь в качестве *xi* рассматриваются конкретные значения (числа), полученные в результате *N* экспериментов. Если взять другие (независимые от предыдущих) *N* экспериментов, то, очевидно, мы получим другое значение m\*. Если взять еще *N* экспериментов, то мы получим еще одно новое значение m\*. Обозначим через *Xi* случайную величину, являющуюся результатом *i*-го эксперимента, тогда реализациями *Xi* будут числа, полученные в результате этих экспериментов. Очевидно, что случайная величина *Xi* будет иметь такую же плотность распределения вероятности, что и исходная, случайная величина *Х*. Также считаем, что случайные величины *Xi* и *Xj* являются независимыми при *i*, не равном *j* (различные независимые друг относительно друга эксперименты). Поэтому формулу (1) перепишем в другом (статистическом) виде:

*m\* = * (2)

Покажем, что оценка является несмещенной:

*M[m\*] = M =  =  = M[X] = m.*

Таким образом, математическое ожидание выборочного среднего m\* равно истинному математическому ожиданию случайной величины *m*. Это достаточно предсказуемый и понятный факт. Следовательно, за оценку математического ожидания случайной величины можно принять выборочное среднее (2). Теперь возникает вопрос: что происходит с дисперсией оценки математического ожидания при увеличении числа экспериментов? Аналитические вычисления показывают, что

*D(m\*) = D/N,*

где *D(m\*)* – дисперсия оценки математического ожидания (2), а *D* – истинная дисперсия случайной величины *X*.

Из вышесказанного следует, что с ростом *N* (количества экспериментов) дисперсия оценки уменьшается, т.е. чем больше мы суммируем независимые реализации, тем ближе к математическому ожиданию мы получим оценку.

## 2.2. ОЦЕНКА ДИСПЕРСИИ

На первый взгляд наиболее естественной оценкой представляется

*D\* = * (3)

где *m\** вычисляется по формуле (2). Проверим, является ли оценка *D\**несмещенной. Формула (3) может быть записана следующим образом [1]:

*D\* =  - (m\*)2.*

Подставим в эту формулу выражение (2):

*D\* =  - (m\*)2 =  -  =*

*=  -  - 2 =  - 2.*

Найдем математическое ожидание оценки дисперсии:

*M[D\*] =  - .* (4)

Так как дисперсия случайной величины не зависит от того, какое математическое ожидание у случайной величины, примем математическое ожидание равным 0, т.е. *m* = 0.

Тогда

*M[] = D,  = ND,* (5)

*M[XiXj] = 0,* при *i ≠ j.* (6)

Последнее равенство следует из того, что эксперименты независимы, а математическое ожидание случайной величины в каждом эксперименте равно 0. Подставляя (5) и (6) в (4), получим:

*M[D\*] = D.*

Отсюда следует, что оценка *D\**не является несмещенной – ее математическое ожидание равно не *D*, а несколько меньше. Пользуясь оценкой *D\** вместо дисперсии *D*, мы получим систематическую ошибку. Чтобы ликвидировать это смещение, достаточно ввести поправку, умножив величину *D\** на *(N-1)/N*. Такую исправленную статистическую дисперсию мы и выберем в качестве оценки:

*D\*\* = *

Таким образом, если в результате *N* экспериментов мы располагаем набором *N* значений случайной величины

*x1, x2, …, xN*,

то для оценок математического ожидания и дисперсии необходимо воспользоваться следующими формулами:

 (7)

## 2.3. ИНТЕРВАЛЬНЫЕ ОЦЕНКИ

В п.1 и п.2 мы рассмотрели вопрос об оценке неизвестного параметра a одним числом. Такая оценка называется *точечной*. В ряде случаев требуется не только найти для параметра *а* подходящее численное значение, но и оценить его надежность и точность. Требуется знать - к каким ошибкам может привести замена параметра *а* его точечной оценкой, и с какой уверенностью можно ожидать, что ошибки не выйдут за известные пределы. Чтобы дать представление о точности и надежности оценки *a\**, в математической статистике пользуются так называемыми *доверительными интервалами* и *доверительными вероятностями*.

Рассмотрим в качестве примера задачу о доверительном интервале при оценке математического ожидания.

Пусть для параметра *а* получена из ряда экспериментов несмещенная оценка *a\**. Мы хотим оценить возможную при этом ошибку. Назначим некоторую достаточно большую вероятность *β* (например, *β = 0.99*), такую, что событие с вероятностью *β* можно считать практически достоверным, и найдем такое значение *e*, для которого

*P(|a\* - a| < e)* = *β* (8)

Тогда диапазон практически наиболее вероятных значений ошибки, возникающих при замене *a* на a\*, по модулю не будет превосходить *е*. Большие по абсолютной величине ошибки будут появляться с малой вероятностью *α = 1 - β*.

Равенство (8) означает, что с вероятностью *β* неизвестное значение параметра *а* попадает в интервал

*I = (a\* - e; a\* + e)*.

Необходимо отметить следующее обстоятельство. Ранее мы рассматривали близость случайной оценки к истинному значению оцениваемого параметра. Здесь ситуация несколько другая. Величина *а* не случайна, зато случаен интервал *I*. И величину *β* можно трактовать как вероятность того, что случайный интервал *I* накроет истинное значение параметра *а*. Вероятность *β* называется *доверительной вероятностью*, *I - доверительным интервалом*,

*a1 = a\* - e; a2 = a\* + e*

а – называются *доверительными границами*. Перейдем теперь к нахождению доверительных границ *a1* и *a2*. Пусть для параметра *а* существует несмещенная оценка *a\**. Если бы нам была известна функция распределения случайной величины (или плотность распределения вероятности) *a\**, то задача нахождения доверительного интервала была бы весьма проста: достаточно было бы найти такое значение *e*, для которого выполняется условие (8). Затруднение состоит в том, что функция распределения оценки *a\** зависит от функции распределения величины *Х* и, следовательно, от самого неизвестного параметра *а*.

В качестве другого примера рассмотрим задачу о доверительном интервале для математического ожидания.

Пусть произведено *N* независимых опытов над случайной величиной *Х*, характеристики которой (дисперсия *D* и математическое ожидание *m*) неизвестны. Для этих параметров получены оценки:

*m\* = , D\* = *

Требуется построить доверительный интервал *I*, соответствующий доверительной вероятности *β* для математического ожидания *m* величины *Х*.

При решении этой задачи воспользуемся тем, что величина *m\** представляет собой сумму *N* независимых случайных величин *Xi*, и, согласно центральной предельной теореме, при достаточно большом *N* ее закон близок к нормальному. Поэтому будем исходить из того, что величина *m\** распределена по нормальному закону. Характеристики этого закона – математическое ожидание и дисперсия – равны соответственно *m* и *D/N*. Найдем такую величину *е*, для которой

*P(|m\* - m| < e)* = *β* (9)

Для нормальной случайной величины (с нулевым математическим ожиданием и единичной дисперсией) функция распределения вероятности

*F(x) = *

С учетом этого формулу (9) запишем в виде:

*P(|m\* - m| < e)* = 2F(e/σ\*) – 1,

где σ\* = (D/N)0.5 – среднее квадратическое отклонение оценки.

Из уравнения

*2F(e/σ\*) – 1 = β*

находим значение *е*:

*e = (σ\*)2F-1[(1+β)/2]* (10)

где *F-1(x)* – функция, обратная *F(…)*, т.е. такое значение аргумента, при котором нормальная функция распределения равна *х*.

Дисперсия *D*, через которую выражена величина*σ\**, нам в точности неизвестна. В качестве ее ориентировочного значения можно воспользоваться оценкой *D\** или *D\*\** и положить приближенно

*σ\* =(D\*\*/N)0.5*.

Таким образом, решена задача построения доверительного интервала

*I = (m\* - e; m\* + e)*

где *е* определяется формулой (10).

Для удобства в табл. 1 приведены значения величины

*t = F-1[(1+β)/2]*.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| *β* | *t* | *β* | *t* | *β* | *t* | *β* | *t* |
| 0.8 | 1.282 | 0.86 | 1.175 | 0.91 | 1.694 | 0.97 | 2.169 |
| 0.81 | 1.310 | 0.87 | 1.513 | 0.92 | 1.750 | 0.98 | 2.325 |
| 0.82 | 1.340 | 0.88 | 1.554 | 0.93 | 1.810 | 0.99 | 2.576 |
| 0.83 | 1.371 | 0.89 | 1.597 | 0.94 | 1.880 | 0.9973 | 3.000 |
| 0.84 | 1.404 | 0.9 | 1.643 | 0.95 | 1.960 | 0.999 | 3.290 |
| 0.85 | 1.439 |  |  |  |  |  |  |

**Пример.** Пусть в результате проведения 30 опытов были получены 30 значений случайной величины *Х*:

10.5, 10.8, 11.2, 10.9, 10.6, 11.0, 10.8, 11.0, 11.6, 10.9, 10.5, 11.8, 10.2, 9.2, 10.2, 11.2, 10.3, 11.1, 11.8, 10.3, 10.7, 10.8, 11.2, 10.9, 10.1, 11.7, 10.8, 11.3, 11.0, 11.9.

Требуется найти оценку *m\** для математического ожидания *m* величины *X* и построить доверительный интервал, соответствующий доверительной вероятности *β = 0.8*.

Вычисляем *m\** = 10.87, *D\*\** = 0.49. Далее *σ\* =(D\*\*/N)0.5* = 0.12. По табл. 1 находим: t = 1.282.

Тогда

*e = tσ\* = 0.16;*

доверительные границы:

*m1* = *m\** - *0.16 = 10.71,*

*m2* = *m\** + *0.16 = 11.03;*

доверительный интервал:

*I = (10.71; 11.03).*

## 2.4. ОЦЕНКА ВЕРОЯТНОСТИ ПО ЧАСТОТЕ

## ПОЯВЛЕНИЯ СОБЫТИЯ

При проведении экспериментов часто приходится оценивать неизвестную вероятность события *P* по его частоте *P\** в *N* независимых экспериментах. Частота некоторого события в *N* независимых экспериментах есть не что иное, как среднее арифметическое наблюдаемых значений величины *Х*, которая в каждом отдельном опыте принимает значение 1 (если событие совершилось), или значение 0 (если событие не произошло):

*P\** = 

В [1] показано, что математическое ожидание величины *Х* равно *Р*, а ее дисперсия равна *Р(1-P)*. Математическое ожидание выборочного среднего равно *Р*:

*M[P\*] = P*,

т.е. оценка *P\** является несмещенной. Дисперсия величины *P\** равна:

*D[P\*] = P(1 - P)/N.*

Специфика этой задачи в том, что *Х* в данном случае – дискретная случайная величина только с двумя возможными значениями: 0 и 1. Сделаем ограничение практически всегда выполняемым - число экспериментов достаточно велико, так что выполняются условия:

*N(1-P) > 4, NP > 4.*

Если эти условия выполнены, то частоту *P\** можно считать распределенной по гауссовскому закону. Тогда параметры этого закона:

*m\* = P, σ\* = [(1 - P)P]0.5*

В [1] приведена методика оценки доверительного интервала, которую мы приведем далее. Границы интервала, в котором заключено истинное значение вероятности события, определяются следующим образом:

*P1 = P\* - t[(1 - P\*) P\*/N]0.5,*

*P2 = P\* + t[(1 - P\*) P\*/N]0.5.*

Здесь *P\** - конкретная оценка вероятности (частоты события), а *t* находится, исходя из заданной доверительной вероятности *β*:

*t = F-1[(1+β)/2]*.

**Пример.** Производится серия из *N* экспериментов с целью оценки вероятности некоторого события. В результате этой серии экспериментов получено значение *P\** = *0.34*. Построить доверительный интервал, в котором с вероятностью *0.85* вычисляется истинная вероятность рассматриваемого события.

По табл. 1 при *β = 0.85* находим: *t* = *1.439*. Умножая это значение на величину

* ≈ 0.0335*,

получим 0.048. Откуда искомый доверительный интервал *I ≈ (0.292; 0.388).*

## 2.5. РЕКУРРЕНТНЫЕ СООТНОШЕНИЯ ДЛЯ ДИСПЕРСИИ И МАТЕМАТИЧЕСКОГО ОЖИДАНИЯ

При моделировании удобно пользоваться рекуррентной формулой для оценки математического ожидания:

* =  =  + [1/N]xN,*

где ** - оценка математического ожидания по выборке *xi* из *N* отсчетов, ** - оценка математического ожидания по выборке *xi* из (*N-1*) отсчета. Аналогичная формула используется для рекуррентной оценки дисперсии:

* =  +[1/N][xn - ]2,* ,

где **- оценка математического ожидания по выборке *xi* из *N* отсчетов.

Далее можно получить несмещенную оценку дисперсии по формуле (7).

## 2.6. СРАВНЕНИЕ ХАРАКТЕРИСТИК ДВУХ СИСТЕМ

Очень часто моделирование проводится для сравнения качества двух или более систем. Очевидно, что для сравнения этих систем мы должны обеспечить максимально идентичные входные воздействия на каждую систему. Как правило, реально трудно обеспечить одинаковые условия их работы (например потому, что в каждой системе есть внутренние шумы неидеальных элементов, которые различны для разных систем). Компьютерное моделирование позволяет обеспечить одинаковые входные реализации для каждой системы. Поэтому процесс проведения статистического моделирования (для *N* реализаций или экспериментов) при решении данной задачи может быть описан следующим образом:

1) получение одной реализации входного воздействия (путем обращения к соответствующим датчикам случайных величин);

2) вычисление реакции первой системы на данное входное воздействие (вычисление того или иного показателя по итогам одного эксперимента);

3) вычисление реакции второй системы на данное входное воздействие (вычисление того или иного показателя по итогам одного эксперимента);

4) переход к первому пункту, если число экспериментов меньше или равно N. По результатам проведенных экспериментов вычисляются средние значения показателей каждой системы.

Если число проведенных экспериментов достигло требуемого значения, то программа моделирования завершает работу и выводит полученные результаты.

## 2.7. ОБРАЗЦЫ ПРОГРАММ

Далее приведены тексты двух программ, в которых рассматривается проведение статистических экспериментов.

**Программа 1(исходный файл lect4\_1.cpp, выполняемый файл lect4\_1.exe)**

Вычисление оценки математического ожидания случайной величины *E*, которая является результатом преобразования гауссовской случайной величины *X* (с нулевым математическим ожиданием и единичной дисперсией). Закон преобразования:

*e = x,* при *x > 0.5,*

*e = 0,* при *x ≤ 0.5.*

#define N 500

#include "model.h"

#include <math.h>

float e[500];

float m\_rek[500];

float m\_ist[500];

void main(void)

{ int a0; int i;

Init\_graph();

for(;;)

{ input\_int("press Enter for continue, (9-exit) ",&a0);

if( a0 = 9 ) break;

for(i=1; i < N ; i++) m\_ist[i] = exp(-0.5\*0.5/2.0)/sqrt(2\*PI);

for(i=1; i < N ; i++)

{ e[i] = gauss(0, 1); if( e[i] <= 0.5) e[i] = 0 ; }

m\_rek[1] = e[1] ;

for( i=2 ; i < N ; i++ )

m\_rek[i] = ( i -1.0)/i \*m\_rek[i-1] + 1.0/i \*e[i];

graf\_2(" matematem srednee ",m\_rek,m\_ist,1,N-1);

}

Close\_graph();

}

**Программа 2 (исходный файл lect4\_2.cpp, выполняемый файл lect4\_2.exe)**

Вычисление оценки дисперсии случайной величины *E*, которая является результатом преобразования гауссовской случайной величины *X* (с нулевым матемаматическим ожиданием и единичной дисперсией). Закон преобразования:

*e = x,* при *x ≥ 0,*

*e = 0,* при *x < 0.*

#define N 500

#include "model.h"

#include <math.h>

float e[500];

float d\_rek[500];

float d\_ist[500];

float m\_rek[500];

void main(void)

{ int a0; int i;

Init\_graph();

for(;;)

{ input\_int(" press Enter if 9 - exit ",&a0);

if(a0 = 9) break;

for(i=0 ; i < N ; i++) d\_ist[i] = ( PI -2.0 )/ PI;

for(i=0 ; i < N ; i++)

{

e[i] = gauss(0, 1);

if(e[i] < 0) e[i] = -e[i];

}

d\_rek[1] = 0;

m\_rek[1] = e[0];

for(i=2; i < N; i++)

m\_rek[i] = (i -1.0)/i \*m\_rek[i-1] + 1.0/i \*e[i];

for(i=2; i < N; i++)

d\_rek[i]=(i-1.)/i\*d\_rek[i-1]+1./i\*(e[i]-m\_rek[i] )\*(e[i]-m\_rek[i] );

graf\_2(" dispersia ",d\_rek,d\_ist,1,N-1); }

Close\_graph();

}

## 2.8. ЗАДАНИЯ К ЛАБОРАТОРНОЙ РАБОТЕ.

1. Написать программу рекуррентной оценки математического ожидания случайной величины Y. Эта случайная величина является результатом следующего преобразования гауссовской случайной величины Х (с математическим ожиданием 0 и дисперсией 1):

*y = x2*, при *х > 0*,

*y* = 0, при других *х*.

Вывести на график рекуррентную оценку математического ожидания в зависимости от количества отсчетов, участвующих в оценке. На этот же график вывести прямую истинного математического ожидания.

1. Выполнить задание 1 при условии, что случайная величина является результатом следующего преобразования гауссовской случайной величины *Х* :

*y = x*, при *х* > 0,

*y = -x*, при других *х*.

1. Выполнить задание 1 при условии, что случайная величина является результатом следующего преобразования гауссовской случайной величины *Х* :

*y = x2*, при любых *х*.

1. Написать программу рекуррентной оценки дисперсии случайной величины *Y*, которая является результатом следующего преобразования гауссовской случайной величины *Х* (с математическим ожиданием 0 и дисперсией 1):

*y = x*, при *х* > 0,

*y* = 0, при других *х*.

Вывести на график рекуррентную оценку дисперсии в зависимости от количества отсчетов, участвующих в оценке, и прямую истинного значения дисперсии.

1. Разработать программу оценки разности математических ожиданий на выходе двух систем. Система 1 представляет собой нелинейное преобразование вида *y* = *x* при *х* > 0.5 и *у* = *-х* при других *х*. Система 2 представляет собой нелинейное преобразование вида *y* = *x* при *х* > 0 и *y* = 0 при других *х*. В качестве входного воздействия выбрать гауссовский процесс с нулевым математическим ожиданием и единичной дисперсией. Вывести на график оценки разности математических ожиданий при идентичных входных реализациях, при различных входных реализациях и прямую теоретической разности математических ожиданий.
2. Выполнить задание 5, если система 1 представляет собой нелинейное преобразование вида *y = x2* при любых *х*, а система 2 представляет нелинейное преобразование вида *y* = *x* при *х* > 0 и *у* = -*х* при других *х*.
3. Выполнить задание 5, если система 1 представляет собой нелинейное преобразование вида *y = x2* при любых *х*, а система 2 представляет нелинейное преобразование вида *y* = *x* при *х* > 0 и *у* = 0 при других *х*.
4. Система обнаружения полезного сигнала на фоне белого шума представляет соединение оптимального фильтра (ОФ) и схемы принятия решения (СПР) о наличии или отсутствии полезного сигнала на входе (рис.1).

ОФ

СПР

Рис. 1

Импульсная характеристика оптимального фильтра согласована с сигналом *s(t)*=sin{-*2t*}, определенным на интервале 0< *t* <1. Шаг дискретизации равен 0.01. Модель помехи представляет собой дискретный белый гауссовский шум с дисперсией 1. Порог обнаружения в схеме принятия решения равен 15. Провести 10000 экспериментов и, считая доверительную вероятность равной 0.95, вычислить доверительный интервал.

1. Для системы обнаружения, представленной на рис.1, по результатам 10000 экспериментов вычислить доверительный интервал при следующих условиях:

*s(t)*=cos{*t*}, при 0 < *t* < 5,

шаг дискретизации равен 0.01, модель помехи - дискретный белый гауссовский шум с дисперсией 1, порог обнаружения в схеме принятия решения равен 17, доверительная вероятность равна 0.95.

## КОНТРОЛЬНЫЕ ЗАДАНИЯ

1. В результате эксперимента получена следующая реализация: 5, 7, 2, -2, 7. Построить рекуррентную оценку математического ожидания.
2. В результате эксперимента получена следующая реализация: 2, 1, 0, -2, 5. Построить рекуррентную оценку дисперсии.
3. Произведено 1000 экспериментов, связанных с обнаружением сигнала. В 720 случаях сигнал был обнаружен. Доверительная вероятность равна 0.95. Построить доверительный интервал.
4. Произведено 100 экспериментов, связанных с обнаружением сигнала. В 90 случаях сигнал был обнаружен. Доверительная вероятность равна 0.9. Построить доверительный интервал.
5. Произведено 20 экспериментов над случайной величиной Х. В результате экспериментов получены следующие результаты:  
   5.16, 5.35, 4.95, 4.84, 5.21, 5.43, 5.24, 5.17, 5.40, 4.87, 4.95, 4.97, 5.05, 5.10, 4.75, 4.84, 5.21, 5.44, 4.94, 5.21.

Найти оценку для математического ожидания величины *X* и построить доверительный интервал, соответствующий доверительной вероятности β = 0.09.

## БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Моделирование случайных величин: Метод. указ. к лабораторной работе №1 /Сост.: Д.А. Иванников, С.М. Кашаев, Л.В. Шерстнева; НГТУ. Н.Новгород, 2001.
2. Моделирование случайных процессов: Метод. указ. к лабораторной работе №2 /Сост.: Д.А. Иванников, С.М. Кашаев, Л.В. Шерстнева; НГТУ. Н.Новгород, 2001.
3. Полов К.П. Функциональное моделирование радиотехнических устройств и систем на ЦВМ: Учеб. пособие / Горьков. политехн.ин-т. Горький, 1989.
4. Полляк Ю.Г. Вероятностное моделирование на электронных вычислительных машинах. - М.: Сов. радио,1971.
5. Венцель Е.С. Теория вероятностей. - М.: Наука,1964.
6. Моделирование в радиолокации /Под ред. А.И. Леонова. - М., 1979.
7. Математическая статистика /В.М. Иванов, В.Н. Калинин, Л.А. Неклумов и др.- М.: Высшая школа, 1981. – 371 с.
8. Гмурман В.Е. Теория вероятностей и математическая статистика. – М.: Высшая школа, 1977. – 479 с.
9. Богачев Б.М., Сысоев В.В. Теория вероятностей: Учебн. пособие / Воронеж. гос. технол. акад. Воронеж, 2000. 135 с.

**ЛАБОРАТОРНАЯ РАБОТА №5**

**«Моделирование простейшего потока»**

* 1. **Цель работы:**

Изучить свойства и характеристики простейшего потока. Сравнить теоретические и модельные значения полученных характеристик.

# На примере построения программ в среде Pascal.

# 2. Теоретические сведения

## Свойства и характеристики простейшего потока

Простейший поток обладает следующими свойствами:

***- стационарность,***

***- отсутствие последействия,***

***- ординарность.***

***Стационарность*** означает, что с течением времени вероятностные характеристики потока не меняются. Стационарность потока равносильна постоянной плотности вероятности поступления вызовов в любой момент времени, иначе говоря, для стационарного потока вероятность поступления i вызовов за промежуток длиной Δt зависит только от величины промежутка и не зависит от его расположения на оси времени (1).

Pi(t +Δt ) = Pi(t1 +Δt ) = Pi(Δt) (1)

***Последействие*** означает зависимость вероятностных характеристик потока от предыдущих событий. Иными словами, вероятность поступления iвызовов в промежуток [t1,t2] зависит от числа, времени поступления и длительности обслуживания вызовов до момента t1*.* Для случайного потока без последействия условная вероятность поступления вызовов в промежутке [t1,t2]*,* вычисленная при любых предположениях о течении процесса обслуживания вызовов до момента *t1,* равна безусловной (2).

Pi( [t1, t2] )|t< t1 = Pi( [t1, t2] ) (2)

***Ординарность*** означает практическую невозможность группового поступления вызовов. Иначе говоря, вероятность поступления двух или более вызовов за любой бесконечно малый промежуток времени Δt есть величина бесконечно малая более высокого порядка, чем Δt, т.е.

=λΔt+o(Δt) (3)

К основным характеристикам случайного потока относят ***ведущую функцию, параметр и интенсивность***.

***Ведущая функция*** случайного потока  есть математическое ожидание числа вызовов в промежутке [t,t+Δt]*.* Функция  *-* неотрицательная, неубывающая и в практических задачах теории распределения информации непрерывна и принимает только конечные значения.

***Параметр потока*** λ(t) в момент времени t есть предел отношения вероятности поступления не менее одного вызова в промежутке [t,t+Δt] к величине этого промежутка Δt при: Δt →0.

 (4)

Параметр потока определяет плотность вероятности наступления вызывающего момента в момент *t.* Определение параметра равносильно предположению, что вероятность поступления хотя бы одного вызова в промежутке [t,t+Δt] с точностью до бесконечно малой пропорциональна промежутку и параметру потока λ(t):

 = λ(t) Δt +o(Δt) (5)

Для стационарных потоков вероятность поступления вызовов не зависит от времени, т. е.*,*

=,

поэтому параметр стационарного потока постоянный. Соответственно получаем

=λΔt+o(Δt) (6)

***Интенсивность*** стационарного потока μ есть математическое ожидание числа вызовов в единицу времени.

Если интенсивность характеризует поток вызовов, то параметр – поток вызывающих моментов. Поэтому всегда μ(t)≥λ(t)*,* а равенство имеет место только для ординарных потоков, когда в каждый вызывающий момент поступает только один вызов.

## Моделирование простейшего потока

Для простейшего потока вызовов длины промежутков zk = tk - tk-1 >0 времени между последовательными вызовами потока распределены по показательному закону с тем же параметром λ

P(z < t) = F(t) =  (7)

Это обстоятельство позволяет моделировать простейший поток вызовов на заданном промежутке времени при помощи метода Монте-Карло, который основывается на следующей теореме:

**Теорема:** Если ri *–* случайные числа, равномерно распределенные на (0,1), то возможное значение xi разыгрываемой непрерывной случайно величины Х с заданной функцией распределения F(х), соответствующее ri является корнем уравнения

F(xi) = ri. (8)

Согласно этой теореме, для получения последовательности случайных значений Z*k*, распределенных по показательному закону с параметром λ, требуется для каждого случайного числа , генерируемого на ПЭВМ датчиком псевдослучайных чисел, решить уравнение

1 - = ri, i =1,2,… (9)

Решая это уравнение относительно zi,имеем

zi = - ln(1-ri) (10)

или

zi = - ln(ri) i =1,2,… (11)

# Порядок выполнения работы

2.1. Сгенерировать случайные равномерно распределённые числа от 0 до 1.

2.2. По формуле zi = - ln(ri), где *i=1, 2, …* получить *Zi* для промежутков между вызовами.

2.3. λ = 10(N+1)/(N+4)(выз/мин*); где* N – номер по журналу.

2.4. На промежутке [T1 ; T2 ], T1 = N+1, T2 =N+4мин*.* получить последовательность *tk* моментов поступления вызовов.

tk = T1 + 

до тех пор пока tk ≤ T2

* 1. Полученные данные свести в таблицу:

|  |  |  |
| --- | --- | --- |
| *Ri* | *Zi* | *Tk* |
| *r1* | z*1* | *T1* |
| *r2* | z*2* | *T2* |
| . | . | . |
| . | . | . |

2.6. Провести статистическую обработку полученных результатов, для этого разделить заданный интервал на 24 равных промежутка длиной:

τ = , (мин).

Для каждого промежутка определить x (τ) – количество вызовов, попавших в промежуток, длиной τ*.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *N интервала* | *1* | *2* | *. . .* | *24* |
| *x(τ )* |  |  |  |  |

Получить таблицу статистического распределения случайной величины

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *x(τ )* | *0* | *1* | *2* | *. . .* |
| *Nk* | *n1* | *N2* | *n3* | *. . .* |

n = ∑ nk = 24

nk *-* количество интервалов в которое попало *к* вызовов.

2.7. Определить модельное значение параметра потока:

a =  - мат. ожидание числа вызовов в *к* интервале.

a = ⇒  = 

2.8. Для заданного ( λ ) и модельного значения (), определить:

1). Вероятность отсутствия вызовов P0 ( t ) за промежуток t = T2 - T1 ;

2). Вероятность поступления одного вызова P1 ( t ) ;

3). Вероятность поступления четырёх вызовов P4 ( t );

4). Вероятность поступления не менее пяти вызовов

P≥5 ( t )=1-( P0 + P1 + P2 + P3 + P4 );

5). Вероятность поступления менее трёх вызовов

P<3 ( t )= P0 + P1 + P2;

6). Вероятность поступления не более семи вызовов

P≤ 7 ( t )= P0 + . . . + P7;

7). Вероятность, что промежуток между вызовами *Zk*

P[ 0.1 < *Zk* < 0.5 ] = F(0.5) - F(0.1) .

2.9. Сделать выводы.

# Контрольные вопросы

* 1. По каким свойствам классифицируются случайные потоки?
  2. Дать определение свойствам:
* стационарность;
* ординарность;
* отсутствие последействия.
  1. Дать определения числовым характеристикам случайных потоков:
* параметр потока λ;
* интенсивность потока μ;
* ведущая функция потока.
  1. Для каких потоков совпадают значения параметра потока и интенсивности: λ = μ?
  2. По какому закону распределён промежуток между соседними вызовами в простейшем потоке?
  3. По какому закону распределена случайная величина, характеризующая количество вызовов простейшего потока, попавших в некоторый промежуток ?

**Библиографический список**

1. Введение в исследование операций: в 2-х книгах. Кн.2. Пер. с англ. – М.: Мир, 1985. – 496 с.
2. Математическая статистика /В.М. Иванов, В.Н. Калинин, Л.А. Неклумов и др.- М.: Высшая школа, 1981. – 371 с.
3. Гмурман В.Е. Теория вероятностей и математическая статистика. – М.: Высшая школа, 1977. – 479 с.
4. Богачев Б.М., Сысоев В.В. Теория вероятностей: Учебн. пособие / Воронеж. гос. технол. акад. Воронеж, 2000. 135 с.

**ЛАБОРАТОРНАЯ РАБОТА №6**

**«Суммирование случайных потоков»**

**1. Цель работы:**

Исследовать сумму двух простейших потоков и определить характеристики результирующего потока.

# На примере построения программ в среде Pascal.

# 2. Теоретические сведения

## Суммирование и разъединение простейших потоков

При объединении нескольких независимых простейших потоков образуется также простейший поток с параметром, равным сумме параметров исходных потоков. При разъединении поступающего простейшего потока с параметром λ на n направлений так, что каждый вызов исходного потока с вероятностью  поступает на i-е на правление, поток i-го направления также будет простейшим с параметром λPi. Эти свойства простейшего потока широко используются на практике, поскольку значительно упрощают расчёты стационарного оборудования и сетей связи.

## Экспериментальная проверка соответствия реального потока простейшему

В простейшем потоке промежутки z между соседними вызовами распределены по показательному (экспоненциальному) закону с параметром λ:

.

Определим математическое ожидание, дисперсию и среднеквадратическое отклонение промежутка *z*:



;

.

Полученное совпадение величин Mz и z характерно для показательного распределения. Это свойство на практике используют как критерий для первоначальной проверки соответствия гипотезы о показательном распределении полученным статистическим данным.

Другой способ проверки основывается на том, что количество вызовов простейшего потока, попавших в интервал времени ***t*** описывается распределением Пуассона:



Определим математическое ожидание Мi и дисперсию Di числа вызовов за промежуток t:

;

.

Совпадение математического ожидания и дисперсии числа вызовов за промежуток t означает соответствие реального потока простейшему. Допустим, для некоторого реального потока получен ряд чисел x1, x2, …, xn, характеризующий число вызовов, поступающих в *n* промежутков длиной t. Обычно принимают t = 15 мин. Рассчитываются среднее значение и несмещенная оценка дисперсии величины *x*:

;

.

В зависимости от степени совпадения величин  и Dx делается вывод о приемлемости модели простейшего потока. Для дальнейшего анализа можно использовать третий центральный момент, величина которого тоже равна .

# Порядок выполнения работы

* 1. Промоделировать два простейших потока. Использовать методику 2.1-2.6 л. р. №5

;

.

|  |  |  |  |
| --- | --- | --- | --- |
| *Nинт* | *1* | *. . .* | *24* |
| *x1(τ )* |  |  |  |
| *x2(τ )* |  |  |  |
| *x1+x2* |  |  |  |

* 1. Получить суммарный поток складывая x(τ )соответствующих интервалов. Построить графики х1(n), x2(n), x(n),

где n- № интервала,

х1 , x2 , x *-* количество вызовов, попавших в интервал для I, II и суммарного потока соответственно.

* 1. Для суммарного потока получить λсум модельное. Использовать методику п. 2.7 л. р. №5.
  2. Сравнить полученное значение λсум и *1+2* .
  3. Рассчитать оценки дисперсии и математического ожидания случайной величины x(τ ) - количество вызовов суммарного потока, попавших в интервал *τ.*
  4. Сделать выводы.

# Контрольные вопросы.

3.1. Какой поток образуется при объединении n простейших потоков?

3.2. Чему равны параметры потоков, образовавшихся при разъединении простейшего потока.

3.3. Какой способ проверки соответствия реального потока простейшему, используют:

а) если измерены промежутки между вызовами потока?

б) если подсчитано число вызовов, попавших в промежутки равной длины?

**Библиографический список**

1. Введение в исследование операций: в 2-х книгах. Кн.2. Пер. с англ. – М.: Мир, 1985. – 496 с.
2. Математическая статистика /В.М. Иванов, В.Н. Калинин, Л.А. Неклумов и др.- М.: Высшая школа, 1981. – 371 с.
3. Гмурман В.Е. Теория вероятностей и математическая статистика. – М.: Высшая школа, 1977. – 479 с.
4. Богачев Б.М., Сысоев В.В. Теория вероятностей: Учебн. пособие / Воронеж. гос. технол. акад. Воронеж, 2000. 135 с.

**ЛАБОРАТОРНАЯ РАБОТА № 7**

**«Анализ *V*-канальной СМО с явными потерями»**

**1. Цельработы:**

Исследовать 1-е распределение Эрланга и характеристики качества СМО с явными потерями.

# На примере построения программ в среде Pascal.

# 2.Теоретические сведения

# Первое распределение Эрланга, характеристики качества

На вход V-канальной СМО с явными потерями поступает простейший поток вызовов с параметром Λ (Эрл.).



Рис.1.1. Граф состояний СМО с явными потерями

Вероятности всех состояний системы (в установившемся режиме) дает первое распределение Эрланга:



К основным характеристикам качества обслуживания рассматриваемой СМО относятся:

**- Вероятность потерь по вр****емени.**



Формулу  обычно называют первой формулой Эрланга.

**- Вероятность потери вызова.**

Для простейшего потока вызовов

Pв = Λпот /Λ = Λ Pν / P = Pν = Eν(Λ)

Таким образом, вероятность потери вызова совпадает с вероятностью потерь по времени.

**- Интенсивность обслуженной нагрузки.**



**- Интенсивность потенциальной нагрузки**



Равенство интенсивностей потенциальной и поступающей нагрузок обусловливает равенство интенсивностей потерянной Λпот и избыточной R нагрузок:

Λпот = R = ΛEV(Λ)

Из чего непосредственно следует равенство потерь по нагрузке и по вызову. Таким образом, все три вида потерь равны между собой. Объясняется это двумя свойствами простейшей потока: стационарностью и отсутствием последействия.

# Порядок выполнения работы:

* 1. Построить графики распределения Pi для V-канальной СМО с явными потерями, если на вход поступает простейший поток вызовов с интенсивностью = 0 - 15 (Эрл). Число каналов обслуживания определяется по вариантам*.*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| *N, вар* | *1* | *2* | *3* | *4* | *5* | *6* |
| V | *3* | *4* | *5* | *3* | *4* | *5* |

2.2.Определить точки пересечения графиков Pi и Pi-1*,* значения Рi(0), Pi(), i=



2.3. Определить характеристики качества обслуживания для Λ= 10(N+1)/(N+4) (Эрл):

* Вероятность потери вызова Pb(Λ);
* Вероятность потерь по времени Pt(Λ);
* Вероятность потерь по нагрузке *Pн(*Λ);
* Обслуженную нагрузку Y;
* Избыточную нагрузку R;
* Потенциальную нагрузку A.

2.4. Сделать выводы.

# Контрольные вопросы.

3.1. Построить граф состояний системы M/M/V/L.

3.2. Записать I распределение Эрланга.

3.3. Привести I формулу Эрланга.

3.4. Дать определение основным видам нагрузки:

* потенциальная;
* избыточная;
* поступающая;
* потерянная;
* обслуженная.

3.5. Дать определение характеристикам качества СМО с явными потерями.

**Библиографический список**

1. Введение в исследование операций: в 2-х книгах. Кн.2. Пер. с англ. – М.: Мир, 1985. – 496 с.
2. Математическая статистика /В.М. Иванов, В.Н. Калинин, Л.А. Неклумов и др.- М.: Высшая школа, 1981. – 371 с.
3. Гмурман В.Е. Теория вероятностей и математическая статистика. – М.: Высшая школа, 1977. – 479 с.
4. Богачев Б.М., Сысоев В.В. Теория вероятностей: Учебн. пособие / Воронеж. гос. технол. акад. Воронеж, 2000. 135 с.

**ЛАБОРАТОРНАЯ РАБОТА № 8**

**«Моделирование реального процесса обслуживания для СМО с явными потерями»**

**1. Цель работы:**

Сравнить значения характеристик качества для СМО с явными потерями, полученных в результате моделирования и рассчитанных по первой формуле Эрланга.

# На примере построения программ в среде Pascal.

# 2.Теоретические сведения

# Моделирование процесса обслуживания в СМО.

Функция распределения промежутка между вызовами , а функция распределения длительности обслуживания . Программа моделирования содержит два генератора случайных величин Z и ξв соответсвтии с заданными функциями A(t) и B(t)*,* переменные to хранения момента поступления очередного вызова и t1, t2,..., tv для хранения момента освобождения i-го () канала.

Для упрощения пояснений примем v=3 и проанализируем работу алгоритма с момента поступления пятого вызова. Первый генератор формирует очередное случайное число z5, что соответствует поступлению пятого вызова to = z1 + z2 + z3 + z4 + z5. Предположим, что до момента to первый канал был занят четвертым вызовом, а второй и третий, соответственно вторым и третьим. Тогда: t1 = z1 + z2 + z3 + z4 + ξ4, t2 = z1 + z2 + ξ2, t3 = z1 + z2 + z3 + ξ3. Каждое из чисел t1 , t2,, t3 определяет момент освобождения соответствующего канала.

При последовательном занятии каналов значение toпоочередно сравнивается с t1 , t2,, tv, пока не обнаруживается ячейка с моментом освобождения . Пусть окажется что и , а . Это означает, что к моменту поступления пятого вызова первый и второй канал оставались занятыми, а третий уже освободился и может принять на обслуживание поступивший пятый вызов. Тогда t3 присваивается t0 . Затем генерируется случайное число ξ5, определяющее длительность обслуживания пятого вызова. Добавлением числа ξ5 к t3 пятый цикл завершается.

Шестой цикл начинается с генерации случайного числа z6. Как и прежде, t0 = t0+z6. Затем осуществляется поочередное сравнение содержимого нулевой ячейки с содержимым остальных ячеек. Если теперь окажется что, ,и , то шестой вызов будет потерян и на этом цикл закончится.

Для подсчета числа поступивших Квыз и потерянных Кпот. вызовов используются два счетчика. В первый добавляется единица при каждой генерации числа z, а во второй - при каждой потере вызова. Отношение Квыз/Кпот. даст по окончании очередной серии статистическую оценку потерь вызовов.

# Порядок выполнения работы:

2.1. Начальные условия моделирования:

1. Параметр поступающего потока: λ = 10 (N+1) / (N+4) (выз/мин), где N - номер по журналу.
2. Среднее время обслуживания и число каналов определяется вариантом:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| *N, вар* | *1* | *2* | *3* | *4* | *5* | *6* |
| *V* | *3* | *4* | *5* | *3* | *4* | *5* |
| *h,сек* | *45* | *60* | *90* | *60* | *90* | *120* |

1. В начале моделирования в системе занято два канала.
   1. Порядок моделирования.

Моделирование осуществлять на интервале: [t1,t2] *мин*.

t1=N+1, t2=N+200*,* где N - номер по журналу.

1. Поступление вызова моделируется аналогично лабораторной работе №1, запоминается в массиве переменной tпости подсчитывается счетчиком Квыз.
2. Процесс обслуживания моделируется по показательному закону распределения.

 ;

.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *r* | *z* |  | *tпост* | *tосв* | *N канала* |
| *r1* | *-* |  | *-* |  | 1 |
| *r2* | *-* |  | *-* |  | 2 |
| *r3* | *Z1* |  | *tn1* |  | 3 |
|  |  |  |  |  | *Потеря* |

1. Время освобождения канала определяется:



1. Каналы занимаются последовательно. Если к моменту поступления вызова заняты все каналы, то он теряется и подсчитывается количество потерянных вызовов Кпот.

2.3. Определить модельную вероятность потери вызова:



Кпот - количество потерянных вызовов;

Квыз - общее количество вызовов;

2.4. Определить Рв по I формуле Эрланга:

,

где Λ = λ *h.*

* 1. Сделать выводы.

# Контрольные вопросы.

3.1. Определение пропускной способности отдельных каналов при:

а) случайном занятии;

б) последовательном занятии.

3.2. Применение символики Кендала-Башарина.

**Библиографический список**

1. Введение в исследование операций: в 2-х книгах. Кн.2. Пер. с англ. – М.: Мир, 1985. – 496 с.
2. Математическая статистика /В.М. Иванов, В.Н. Калинин, Л.А. Неклумов и др.- М.: Высшая школа, 1981. – 371 с.
3. Гмурман В.Е. Теория вероятностей и математическая статистика. – М.: Высшая школа, 1977. – 479 с.
4. Богачев Б.М., Сысоев В.В. Теория вероятностей: Учебн. пособие / Воронеж. гос. технол. акад. Воронеж, 2000. 135 с.

**ЛАБОРАТОРНАЯ РАБОТА №9**

**«Исследование СМО с ожиданием»**

**1. Цель работы:**

Изучить второе распределение Эрланга и характеристики качества систем с очередями.

# На примере построения программ в среде Pascal.

# 2.Теоретические сведения

# Второе распределение Эрланга.

# Характеристики качества систем M/M/V/W.

V- канальная СМО обслуживает простейший поток вызовов. При занятости всех *v* выходов поступивший вызов становится в очередь и обслуживается после некоторого ожидания. Общее число вызовов, находящихся в системе на обслуживании и в очереди, обозначим  и назовем состоянием системы. При  величина i характеризует число занятых каналов в системе, при  число занятых каналов равно v, а разность i - v есть длина очереди. Параметр потока освобождений определяется числом занятых выходов и в первом случае зависит от состояния системы i*,* а во втором  имеет постоянное значение v*.*



Рис. Граф состояний СМО с ожиданием

Отметим, что при интенсивности поступающей нагрузки Λ, равной или большей числа выходов системы v*,* с вероятностью 1 постоянно будут заняты все выходы и длина очереди будет бесконечной. Поэтому, чтобы система могла функционировать нормально и очередь не росла безгранично, необходимо выполнить условие Λ < v.

Вероятность того, что система в установившемся режиме находится в состоянии i (Pi.) определяем по второму распределению Эрланга:

, ;

, 

К основным характеристикам качества обслуживания СМО с ожиданием относят:

**- Вероятность ожидания для посту****пившего вызова**

Для простейшего потока вызовов она совпадает с вероятностью занятости всех выходов в системе, т. е. с вероятностью потерь по времени:



Приведенное выражение называется второй формулой Эрланга.

**Интенсивность обслуж****енной нагрузки.**



Из-за отсутствия явных потерь сообщений интенсивность поступающей нагрузки совпадает с интенсивностью обслуженной и избыточная нагрузка отсутствует. Поскольку для простейшего потока интенсивность потенциальной нагрузки равна интенсивности поступающей, потерянная нагрузка также отсутствует. Однако не всегда в системе с ожиданием потери по нагрузке равны нулю. При обслуживании примитивного потока (данная модель здесь не рассматривается) источник за счет ожидания в среднем меньше находится в свободном состоянии, чем в системе без потерь. Это приводит к снижению интенсивности потока вызовов и поступающая нагрузка меньше потенциальной. И хотя все поступающие вызовы обслуживаются, потери по нагрузке имеют место.

Λ можно рассматривать как математическое ожидание числа занятых выходов, а v - Λ -соответственно как математическое ожидание числа свободных выходов.

**- Вероятность превышения длиной очереди заданной величины** **n***.*



**- Средняя длина очереди.**



Величина  есть интенсивность нагрузки, создаваемой ожидающими вызовами, а  - интенсивность потока задержанных вызовов, где каждый задержанный вызов в среднем ждет . Тогда



**- Средняя длительность ожидания.**

Из (3.10) и (3.11) следует



Средняя длительность ожидания для любого поступившего вызова



Величины и  выражены в условных единицах времени.

# Порядок выполнения работы:

Используя вторую формулу Эрланга, определить число каналов обслуживания, обеспечивающих заданную вероятность ожидания Р(γ > 0), если на вход системы поступают простейшие потоки с интенсивностью:

 Эрл,

 Эрл,

 Эрл.

Значения Р(γ >0) взять по вариантам:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *Nвар* | 1 | 2 | 3 | 4 | 5 |
| *Р(>0)* | 0,01 | 0,015 | 0,02 | 0,025 | 0,005 |

Привести таблицу и график зависимости DV():

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  | |  | |
| V |  | V |  | V |  |
|  |  |  |  |  |  |
| . |  | . |  | . |  |
| . |  | . |  | . |  |

(Построение трёх этих графиков производить в одной системе координат).

2.3. Для  и V=[1.5 Λ] определить:

* вероятность превышения очередью трёх вызовов;
* вероятность ожидания;
* среднее время ожидания для задержанного вызова;
* среднее время ожидания для любого вызова;
* среднюю длину очереди.

Сделать выводы.

# Контрольные вопросы.

3.1. Построить граф состояний системы M/M/V/W.

3.2. Определить вероятность любого состояния системы с ожиданием.

3.3. Вывести основные характеристики качества системы M/M/V/W.

3.4. Указать условие существования установившегося режима.

**Библиографический список**

1. Введение в исследование операций: в 2-х книгах. Кн.2. Пер. с англ. – М.: Мир, 1985. – 496 с.
2. Математическая статистика /В.М. Иванов, В.Н. Калинин, Л.А. Неклумов и др.- М.: Высшая школа, 1981. – 371 с.
3. Гмурман В.Е. Теория вероятностей и математическая статистика. – М.: Высшая школа, 1977. – 479 с.
4. Богачев Б.М., Сысоев В.В. Теория вероятностей: Учебн. пособие / Воронеж. гос. технол. акад. Воронеж, 2000. 135 с.

**Методические рекомендации**

**для оформления отчета**

Отчет должен содержать:

1. Титульный лист с указанием номера и темы лабораторной работы и Ф.И.О. выполнявшего.
2. Цель.
3. Результат выполнения работы в виде последовательности шагов выполнения лабораторной работы с теоретическим минимумом необходимым для ответов на вопросы домашнего задания. Результаты выполнения лабораторной работы.
4. А также, желательно указать трудные моменты освоения пройденного материала. Или же, наоборот, легкие задания, не привлекающие к себе должного внимания.