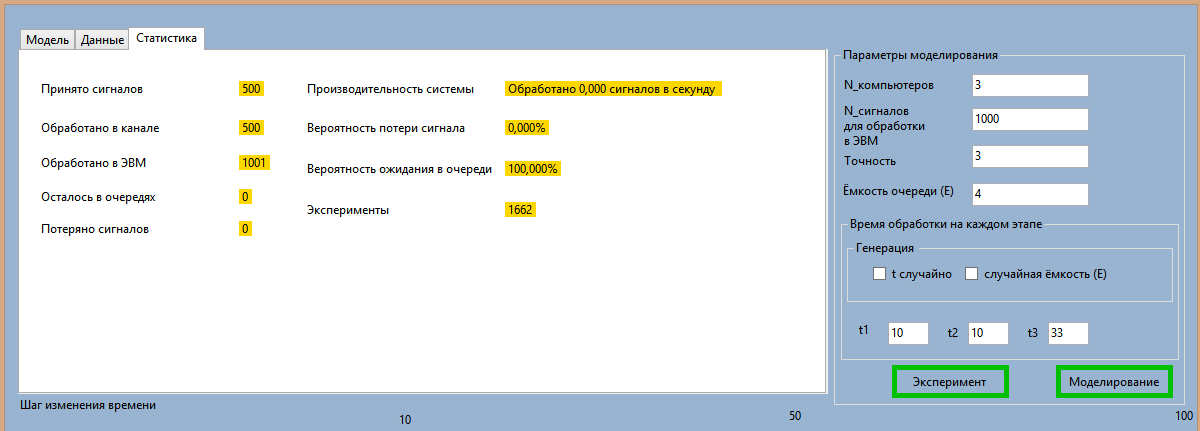
1. 1. студента группы ИТ – 42  
      Курбатовой Софьи Андреевны

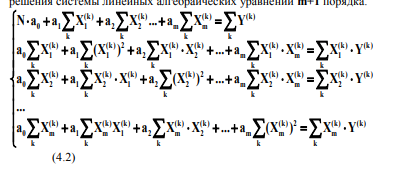
|  |  |  |  |
| --- | --- | --- | --- |
| Выполнение: |  | Защита |  |

* + 1. Регресионный анализ системы массового обслуживания

**Цель работы**: построить регрессионную модель системы массового обслуживания, экспериментируя с имитационной модель.

* + - * 1. Содержание работы
      1. Система обработки информации содержит мультиплексный канал и N ЭВМ. Сигналы поступают на вход канала через t1(мкс).
      2. В канале они предварительно обрабатываются в течение t2 (мкс). Затем они поступают на обработку в ту ЭВМ, где наименьшая очередь. Емкости входных накопителей в каждой ЭВМ - E. Время обработки сигнала в каждой из ЭВМ - t3 (мкс).
      3. Смоделировать процесс обработки 1000 сигналов.
      4. **Данные для детерминированной модели СМО:** N=3, t1=10, t2=10 , t3=33, Е=4.
      5. **Данные для стохастической модели СМО:** интервал t1 распределен по показательному закону с параметром λ1=0,1, интервалы t2, t3 распределены нормально с параметрами m2=10, m3=33, σ2=1,5, σ3=3; вследствие возмущающих воздействий емкости входных накопителей каждой из ЭВМ непрерывно меняются, поэтому величина E является стационарным случайным процессом с нормальным законом распределения и интервалом разброса [2... 6] (сигналы, находившиеся в накопителе до изменения его емкости и не вмещающиеся в него после изменения его емкости, уничтожаются).
      6. **Варьируемые параметры:** N.
      7. Показатели работы: производительность системы, стоимость обработки, вероятность переполнения накопителей.
      8. 1. Определить отклик регрессионной модели и совокупность факторов модели. В качестве отклика рекомендуется взять один из показателей работы системы, а факторами должны быть все величины, которые могут влиять на отклик.
      9. 2. Изменяя факторы модели при каждом прогоне модели, провести несколько десятков имитационных экспериментов с моделью, фиксируя значения отклика и факторов.
      10. 3. Решив систему уравнений (4.2), найти константы регрессионной модели a0, a1, a2, … am.
      11. 4. Проанализировать полученную зависимость. Установить, соответствуют ли знаки и величины констант регрессионной модели интуитивным представлениям о характере зависимости отклика от факторов.
      12. 5. Оценить адекватность регрессионной модели
          1. Ход работы
      13. Условные обозначения: Аij – активность, ФДi – функциональное действие, УЗij – условие запуска.
      14. В системе наблюдаются следующие функциональные действия (ФД):
      15. ФД1 – приход сигнала с интервалом t1
      16. ФД2 – обработка сигнала внутри канала
      17. ФД3 – поступление на обработку в ЭВМ с наименьшей очередью
      18. Предполагается наличие следующих активностей:
      19. А10 – Поступление сигнала в канал
      20. А21 – Обработка сигнала внутри канала
      21. А22 – Конец обработки и переход к следующему
      22. А31 – Определение ЭВМ с меньшей очередью (где емкость больше)
      23. А32 – Выполнение обработки сигнала в ЭВМ
      24. Кобрбсигн – количество обработанных сигналов.
      25. Квх – количество принятых(входных) сигналов
      26. Кпотерсигнал – количество сигналов, которые были потеряны
      27. 1. Отклик Y – количество потерянных сигналов.
      28. Факторы:
      29. X1 – t1 – время между поступлением сигналов [5..15] берем 5 случайных значений
      30. X2 – t2 – время обработки сигналов в канале [5..15] берем 5 случайных значений
      31. X3 – t3 – время обработки сигнала в ЭВМ [25..35] берем 5 случайных значений
      32. X4 – E – емкость накопителя очереди [2..6]
      33. 2. Было проведено 1662 эксперимента со всеми возможными комбинациями факторов.
      34. 

Моделирование

* + - 1. По полученным данным была составлена СЛАУ. Для этого в программу был добавлен следующий фрагмент, представленный на листинге 1, реализующий метод наименьших квадратов, используя результаты экспериментов над моделью СМО. Таким образом найдем неизвестные коэффициенты регрессии.
      2. 

СЛАУ для решения

* + - 1. Листинг 1 Функция для записи результата

static ResultLine Work(double NumPrepSignal)

{

CountExperiment(NumPrepSignal,false,false);

return new ResultLine(t1, t2, t3, Ecapcity, prepSignal);

* + - 1. }
      2. Листинг 2 Фрагмент появления случайных значений

foreach(var E in Enumerable.Range(2, 6).OrderBy(x => rnd.Next()).Take(5))

{ var numSignal = Convert.ToDouble(tbNumSignal.Text.Trim());

(t1, t2,t3,Ecapcity) = (T1, T2,T3,E);

var res = Work(numSignal);

resultLine.Add(res);

Experiment = ++experiments;

* + - 1. }

var writer = new Writer();

writer.WriterResultLine(resultLine);

var Slau = new SLAULine

{

a0 = Experiment,

a1 = resultLine.Select(x => x.X1).Sum(),

a2 = resultLine.Select(x => x.X2).Sum(),

a3 = resultLine.Select(x => x.X3).Sum(),

a4 = resultLine.Select(x => x.X4).Sum(),

b = resultLine.Select(x => x.Y).Sum()

};

var Slau1 = new SLAULine

{

a0 = resultLine.Select(x => x.X1).Sum(),

a1 = resultLine.Select(x => x.X1 \* x.X1).Sum(),

a2 = resultLine.Select(x => x.X2 \* x.X1).Sum(),

a3 = resultLine.Select(x => x.X3 \* x.X1).Sum(),

a4 = resultLine.Select(x => x.X4 \* x.X1).Sum(),

b = resultLine.Select(x => x.Y \* x.X1).Sum(),

};

var Slau2 = new SLAULine

{

a0 = resultLine.Select(x => x.X2).Sum(),

a1 = resultLine.Select(x => x.X1 \* x.X2).Sum(),

a2 = resultLine.Select(x => x.X2 \* x.X2).Sum(),

a3 = resultLine.Select(x => x.X3 \* x.X2).Sum(),

a4 = resultLine.Select(x => x.X4 \* x.X2).Sum(),

b = resultLine.Select(x => x.Y \* x.X2).Sum(),

};

var Slau3 = new SLAULine

{

a0 = resultLine.Select(x => x.X3).Sum(),

a1 = resultLine.Select(x => x.X1 \* x.X3).Sum(),

a2 = resultLine.Select(x => x.X2 \* x.X3).Sum(),

a3 = resultLine.Select(x => x.X3 \* x.X3).Sum(),

a4 = resultLine.Select(x => x.X4 \* x.X3).Sum(),

b = resultLine.Select(x => x.Y \* x.X3).Sum(),

};

var Slau4 = new SLAULine

{

a0 = resultLine.Select(x => x.X4).Sum(),

a1 = resultLine.Select(x => x.X1 \* x.X4).Sum(),

a2 = resultLine.Select(x => x.X2 \* x.X4).Sum(),

a3 = resultLine.Select(x => x.X3 \* x.X4).Sum(),

a4 = resultLine.Select(x => x.X4 \* x.X4).Sum(),

b = resultLine.Select(x => x.Y \* x.X4).Sum(),

};

var SLAU = new List<SLAULine>

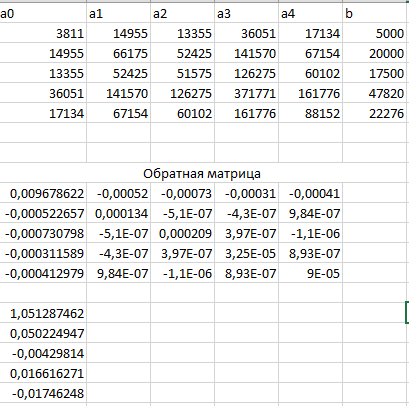
{

Slau, Slau1, Slau2, Slau3,Slau4

};

var slau = new Writer();

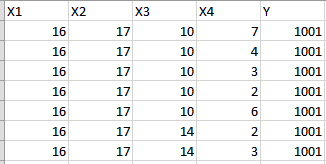
slau.WriterResultLine(SLAU);

* + - 1. }
      2. Полученные значения были записаны в таблицу и представлены на рисунке 4.2. Далее была вычислена обратная матрица и произведено умножение полученное обратной матрицы на вектор b.
      3. 

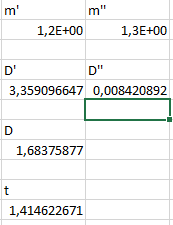
Параметры СЛАУ

* + - 1. Далее используя полученные значения параметров a0,a1,a2,a3,a4 выполним расчет регрессионной зависимости в следующем виде:
      2. 

Расчет регрессионной зависимости

* + - 1. В качестве β используем полученные a.
      2. Значение X были также сгенерированы в процессе моделирования. Фрагмент на рисунке ниже:
      3. 

Фрагмент расчета

* + - 1. По таблице распределения Стьюдента для числа степеней свободы β = n-1 находим наиболее близкое табличное значение tβтабл, не меньшее, чем tβ. Оценкой адекватности регрессионной модели будет величина α = 1-р, где р – вероятность, соответствующая табличному значению tβтабл.
      2. Оценим адекватность полученной модели:
      3. 

Оценка адекватности полученной модели

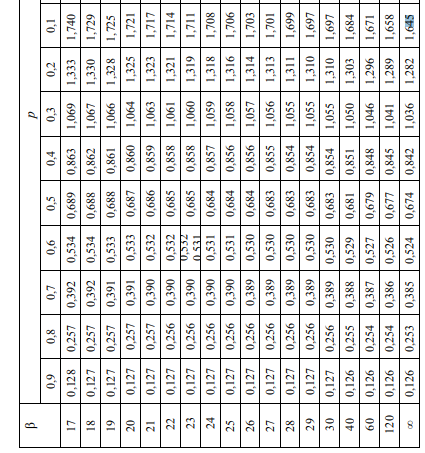
* + - 1. Можно заметить что значение полученного t=1,4. Выберем ближайшее значение по таблице и определим p = 0.1:
      2. 

Таблица с критериями распределения

* + - 1. Следовательно адекватность полученной модели 1-0.1 = 0.9, что говорит о регрессионная модель показала достаточную адекватность.
      2. **Вывод:** Таким образом в ходе выполнения лабораторной работы было осуществлено знакомство с регрессионной системой массового обслуживания. Проведенные опыты показали, что модель показала достаточный уровень адекватности.
      3. 