Министерство образования Республики Беларусь

Учреждение образования

«Белорусский государственный университет информатики и радиоэлектроники»

Кафедра программного обеспечения информационных технологий

ОТЧЁТ

по лабораторной работе №4

по курсу «Системный анализ и машинное моделирование»

на тему

«Построение аналитической и имитационной моделей непрерывно-стохастической системы»

Вариант 7

|  |  |
| --- | --- |
| Проверил:  Мельник Н.И. | Выполнил:  студент гр. 851001  Касперович В.Л. |

Минск 2011

1. ЦЕЛЬ РАБОТЫ

Построение аналитической и имитационных моделей непрерывно-стохастической системы и расчет параметров данной системы с использованием построенных моделей.

1. ИСХОДНЫЕ ДАННЫЕ К РАБОТЕ

На вход n – канальной СМО с отказами поступает поток заявок с интенсивностью λ = 6 заявок в час. Среднее время обслуживания одной заявки 0.8 часа. Каждая обслуженная заявка приносит доход 4 у.е. Содержание одного канала обходится 2 у.е./час. Определить экономически целесообразное количество каналов.

1. РАСЧЕТ МОДЕЛИ Q-СХЕМЫ

В данном случае мы имеем дело с системой вида M/M/n с отказами. Её схема представлена на рис.1.



Рис.1. Q-схема

Процесс, происходящий в этой системе, является процессом размножения и гибели. Исходя из этого, построим диаграмму интенсивностей переходов (ДИП), показанную на рис.2.

Максимальное число заявок в системе – n (по 1 заявке в каждом из каналов), минимальное – 0.

На основании ДИП построим систему уравнений по следующему правилу: встречные потоки через сечение ДИП всегда равны между собой:





Рис.2. ДИП Q-схемы

Обозначим ;

Исходя из того, что сумма вероятностей всех состояний системы равна 1 (нормировочное уравнение), получим:т.е. формулу Эрланга.

Общее число обработанных заявок будет равно  где *t* – период времени работы СМО, *A* – её абсолютная пропускная способность, *n* – число каналов; прибыль в таком случае составит  у.е. Затраты на функционирование системы будут равны  у.е.

Необходимо подсчитать экономически выгодное число каналов, т.е. найти максимум функции  поскольку *t*=const, для задачи экстремизации эту величину можно не учитывать, т.е. максимизировать функцию . При подставленных значениях  формула будет иметь вид .

Экстремизацию этой функции осуществим при помощи пакета MATLAB, выполнив следующий скрипт:

function [numchan,profit]=QScheme(inpint,proctime,chansum,oneprof,tm)

%Models effectivity of Erlang M/M/n Q-scheme.

for i=1:1:20

A(i) = ((inpint\*proctime)^i)/factorial(i);

B(i) = oneprof\*inpint\*(1-A(i)\*(1/(1+sum(A(1:i)))))-chansum\*i;

end;

plot(B);

[C,num] = max(B);

numchan=num;

profit=C\*tm;

При исходных данных, соответствующих варианту, получим: наиболее эффективное количество каналов – 6; прибыль, полученная при этом – 775 у.е.

При построении имитационной модели получено: наиболее эффективное количество каналов – 6; прибыль, полученная при этом – 850-900 у.е. (в среднем).

Исходный код имитационной модели находится в приложении А, графики зависимостей, построенные при аналитическом и имитационном моделировании соответственно, – в приложении Б.

1. ВЫВОДЫ

В данной работе была исследована непрерывно-стохастическая модель (Q-схема). Для нее был построена диаграмма интенсивностей переходов, на основе которой были посчитаны вероятности состояний СМО. Были посчитаны необходимые характеристики системы и сравнены с результатами, получившимися при имитационном моделировании.

ПРИЛОЖЕНИЕ А. ИСХОДНЫЙ КОД ИМИТАЦИОННОЙ МОДЕЛИ

– класс MainForm (интерфейс)

using System;

using System.Collections.Generic;

using System.Drawing;

using System.Windows.Forms;

using System.IO;

using System.Linq;

namespace SAMM\_5\_6

{

public partial class MainForm : Form

{

public MainForm()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

StreamWriter sw = new StreamWriter("1.csv");

double[] values = new double[20];

chart1.Series.Clear();

System.Windows.Forms.DataVisualization.Charting.Series DataSer\_1 = new System.Windows.Forms.DataVisualization.Charting.Series();

DataSer\_1.ChartType = System.Windows.Forms.DataVisualization.Charting.SeriesChartType.Spline;

DataSer\_1.Color = Color.Red;

DataSer\_1.Name = "";

for (int i = 1; i <= 20; i++)

{

QueueSystem qs = new QueueSystem(i, 100,Int32.Parse(textBox1.Text),Double.Parse(textBox2.Text),Double.Parse(textBox3.Text),Double.Parse(textBox4.Text));

values[i - 1] = qs.Operate();

sw.WriteLine(Convert.ToString(i) + ';' + Convert.ToString(values[i - 1]));

DataSer\_1.Points.AddXY((double)i, values[i - 1]);

}

sw.Close();

double max = values.Max();

label1.Text = "Целесообразное количество каналов: " + Convert.ToString(values.ToList().IndexOf(max) + 1);

label2.Text = "Максимальная прибыль: " + Convert.ToString(max) + " у.е.";

chart1.ResetAutoValues();

chart1.Series.Add(DataSer\_1);

}

}

}

– класс QueueSystem (моделирование СМО)

using System;

using System.Collections.Generic;

namespace SAMM\_5\_6

{

class QueueSystem

{

private List<int> channels; //хранится время для освобождения всех каналов

private List<int> timeDemands;

double t, val;

int numHours;

Random rnd, rnd1, rnd2;

int intens;

double proctime, sum, demandpay, profit = 0;

public QueueSystem(int numChannels, int \_numHours, int \_intens, double \_proctime, double \_sum, double \_demandpay)

{

timeDemands = new List<int>();

channels = new List<int>();

rnd = new Random();

rnd1 = new Random();

rnd2 = new Random();

for (int i = 0; i <= numChannels - 1; i++)

{

channels.Add(0);

}

numHours = \_numHours;

intens = \_intens;

proctime = \_proctime;

sum = \_sum;

demandpay = \_demandpay;

}

private void addNewDemand()

{

t = (float)rnd2.Next(800) / 1000\*proctime;

val = (rnd1.Next(1000000) % 2 == 0) ? (proctime - t) : (proctime + t);

int fr = channels.IndexOf(0);

if (fr >= 0) //есть свободный канал

{

channels[fr] = (int)(val \* 60);

}

}

private void Minute(int order)

{

foreach (int k in channels)

{

if (k == 1) profit += demandpay;

}

for (int i = 0; i <= channels.Count - 1; i++)

if (channels[i] != 0)

channels[i]--;

if (timeDemands.Count != 0)

{

while ((timeDemands.Count>0)&&(timeDemands[0] == order))

{

timeDemands.RemoveAt(0);

addNewDemand();

}

}

}

private void Hour()

{

for (int i = 0; i <= intens-1; i++)

{

timeDemands.Add(rnd.Next(60));

}

timeDemands.Sort();

for (int j = 0; j <= 59; j++)

{

Minute(j);

}

}

public double Operate()

{

for (int i = 0; i <= numHours - 1; i++)

{

Hour();

}

return (profit-numHours\*channels.Count\*sum);

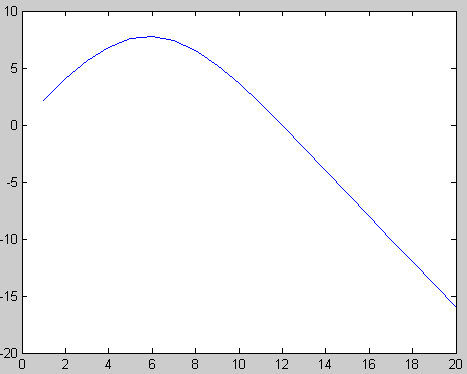
}

}

}

ПРИЛОЖЕНИЕ Б. ГРАФИКИ ЗАВИСИМОСТЕЙ

– при аналитическом моделировании:



– при имитационном моделировании:

