МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

ФГБОУ ВО «ТВЕРСКОЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Факультет прикладной математики и кибернетики

Направление 09.03.03 – Прикладная информатика

Профиль подготовки «Прикладная информатика в экономике»

**РАСЧЁТНО-ГРАФИЧЕСКАЯ РАБОТА**

по дисциплине «Имитационное моделирование»

Тема: «Моделирование системы обслуживания погрузочно-разгрузочных работ»

2020-2021 уч. год 4 курс 7 семестр

**Авторы:** студенты 4 курса

Руденок Михаил Александрович

Степанов Андрей Андреевич

**Руководители практики:** к.ф.-м.н., доцент кафедры математической статистики и системного анализа

Багрова Инна Александровна

**Оценка:** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Тверь, 2020

**Оглавление**

[Введение 3](#_Toc59212465)

[Постановка задачи 4](#_Toc59212466)

[Схема системы 5](#_Toc59212467)

[Блок-схемы 6](#_Toc59212468)

[Доверительный интервал 10](#_Toc59212469)

[График 11](#_Toc59212470)

[Код программы 12](#_Toc59212471)

[Заключение 16](#_Toc59212472)

[Список литературы 17](#_Toc59212473)

**Введение**

Имитационное моделирование – метод исследования, позволяющий строить модели, которые описывают процессы так, как они проходили бы в действительности. Актуальность данной темы заключается в том, что процедура моделирования дает возможность изучения поведения системы как в естественных, так и в создаваемых условиях. Кроме того, можно оценить как сам процесс в комплексе, так и отдельные его характеристики.

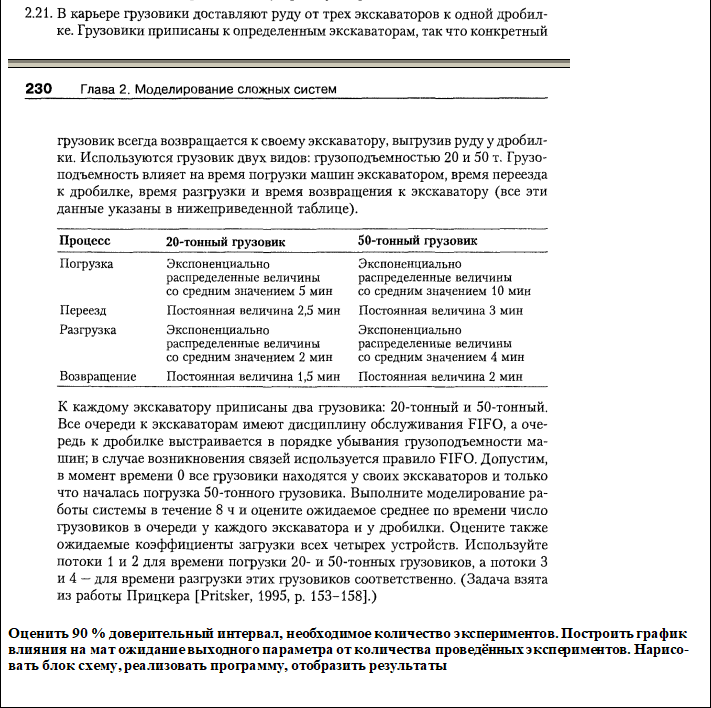
Основные цели имитационного моделирования:

• описать поведение системы;

• построить теории и гипотезы, которые могут объяснить наблюдаемое поведение;

• использовать эти теории для предсказания будущего поведения системы, т. е. тех воздействий, которые могут быть вызваны изменениями в системе или изменениями способов ее функционирования.

# Постановка задачи



**Схема системы**

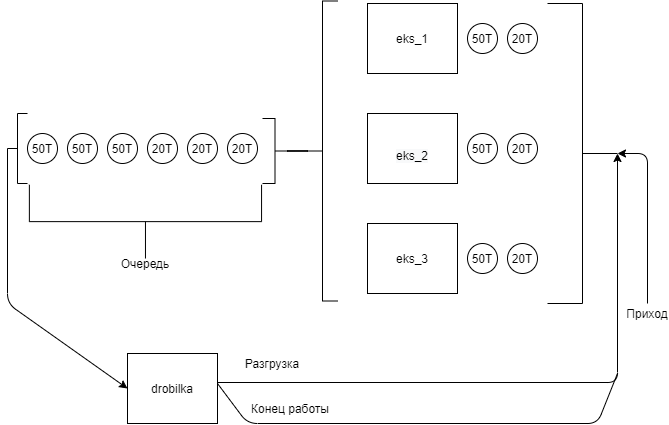
****

Схема.1. Схема работы системы с 3 экскаваторами и 1 дробилкой. При окончании работы грузовик возвращается к своему экскаватору.

**Блок схемы**

Изображение выглядит как текст

Автоматически созданное описание

Схема.2. Итерации программы

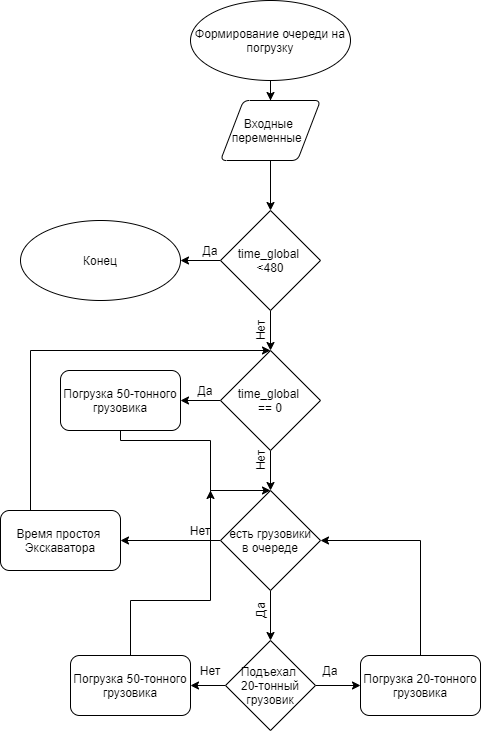


Схема.3. Формирование очереди на погрузку

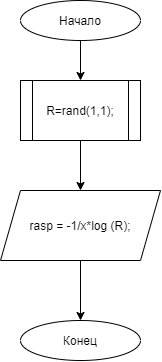


Схема.4. Моделирование времени погрузки для грузовиков

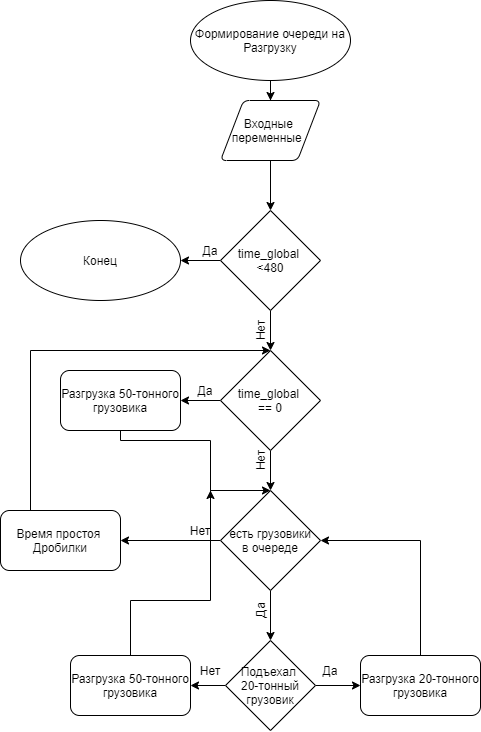


Схема.5. Формирование очереди на разгрузку

**Доверительный интервал**

Доверительные интервалы (англ. Confidence Intervals) одним из типов интервальных оценок, используемых в статистике, которые рассчитываются для заданного уровня значимости. Они позволяют сделать утверждение, что истинное значение неизвестного статистического параметра генеральной совокупности находится в полученном диапазоне значений с вероятностью, которая задана выбранным уровнем статистической значимости. Для построения доверительного интервала воспользуемся данной схемой:

* Если оценка такова, что для нее , то считается, что имеет абсолютную погрешность β
* Если мы будем выполнять повторные прогоны до тех пор, пока половина длины 100∙(1-α) процентного интервала, заданного формулой
* будет ≤ β (β>0), то

**График**

Одной из задач работы было построить график влияния на математическое ожидание выходных параметров (среднее по времени число грузовиков в очереди у каждого экскаватора и у дробилки) от количества проведённых экспериментов. В данном случае моделируется 100 тестов.

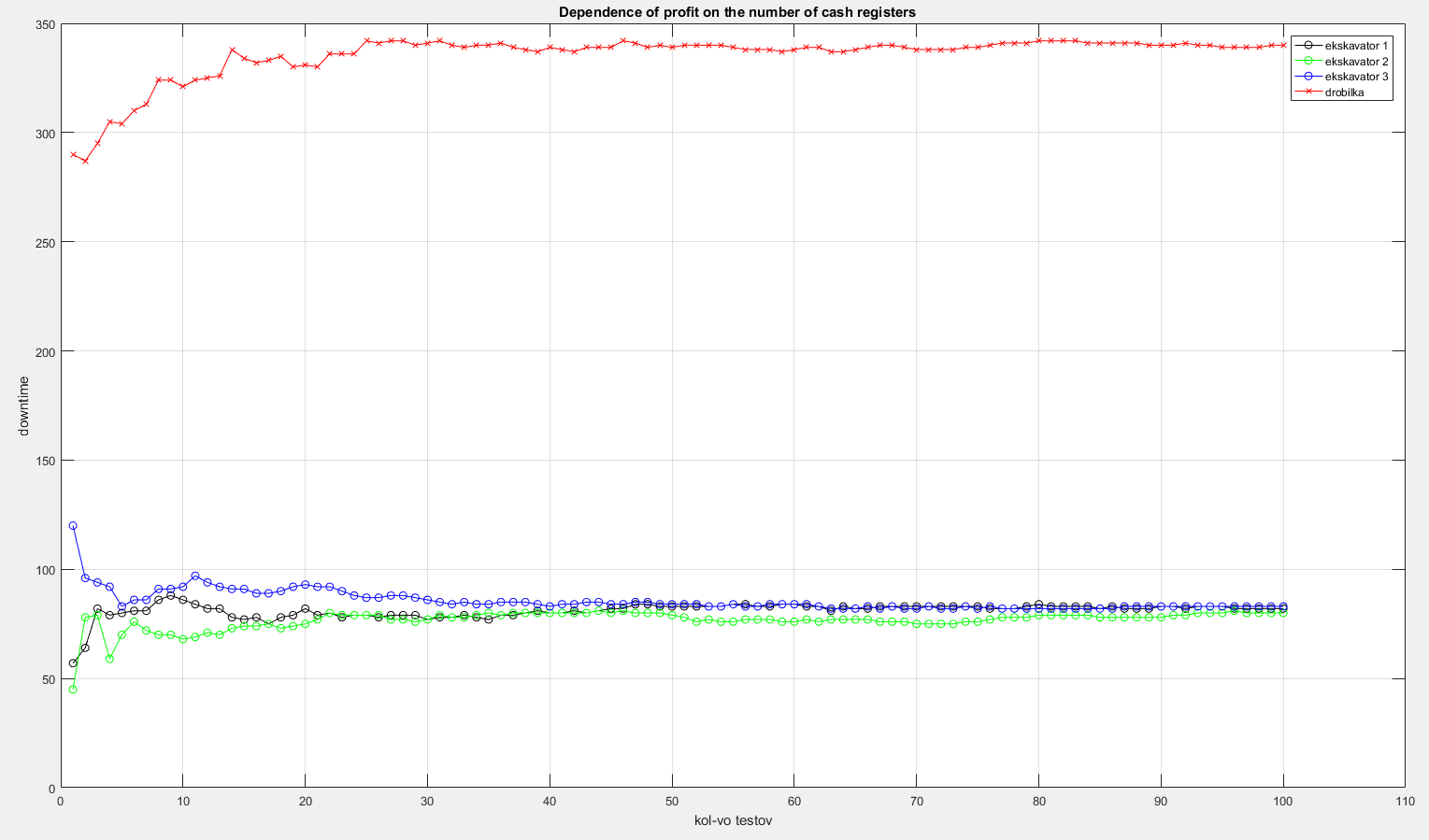


Рис.6. график влияния на математическое ожидание выходных параметров

**Код программы**

count = 0;

a = 0;

b = 0;

c = 0;

d = 0;

kol\_vo = 100;

xmin = 0;

xmax = 110;

for n=1:kol\_vo

count = count + 1;

am = ekskavator();

a = am(1) + a;

ac = a / count;

eks\_1(count,:) = [round(ac)];

s\_eks\_1 = sum(eks\_1 / kol\_vo);

bm = ekskavator();

b = bm(1) + b;

bc = b / count;

eks\_2(count,:) = [round(bc)];

s\_eks\_2 = sum(eks\_2 / kol\_vo);

cm = ekskavator();

c = cm(1) + c;

cc = c / count;

eks\_3(count,:) = [round(cc)];

s\_eks\_3 = sum(eks\_3 / kol\_vo);

dm = drobilka();

d = (dm(1) + d);

dc = d / count;

drobilka\_otvet(count,:) = [round(dc)];

plot(eks\_1,'marker','o','color', 'k')

hold on

plot(eks\_2,'marker','o','color', 'g')

plot(eks\_3,'marker','o','color', 'b')

plot(drobilka\_otvet,'marker','x','color', 'r')

xlim([xmin, xmax])

xlabel('kol-vo testov')

ylabel('downtime')

title(' Dependence of profit on the number of cash registers')

legend('ekskavator 1','ekskavator 2','ekskavator 3','drobilka')

grid on

hold off

end

ae =am(end);

de = dm(end);

percent\_zagruzhennosti\_drobilki = round((de \* kol\_vo)/480)

percent\_zagruzhennosti\_ekskavator = round((ae \* kol\_vo) / 480)

percent\_service\_gryzovik = round((((s\_eks\_1 + s\_eks\_2 + s\_eks\_3) / 3) \* kol\_vo)/480)

**function** rasp = xp\_rasp(x)

R=rand(1,1);

rasp = -1/x\*log (R);

**end**

**function** gryz\_waiting = ekskavator()

count = 0;

time\_global = 0;

pereezd50 = 3;

comeback50 =2;

pereezd20 = 2.5;

comeback20 =1.5;

razn20 = 0;

razn50 = 0;

gryz20\_pog = 0;

gryz\_waiting\_s = 0;

sum\_zagr\_eks = 0;

gryz\_afk = 0;

while time\_global <480

if (time\_global == 0)

gryz50\_pog = round(xp\_rasp(1/10));

time\_global = gryz50\_pog;

razn20 = time\_global;

zagr\_eks = gryz50\_pog;

else

razg20 = xp\_rasp(1/2);

razg50 = xp\_rasp(1/4);

gryz20\_afk = pereezd20 + razg20 + comeback20 + razn20;

gryz50\_afk = pereezd50 + razg50 + comeback50 + razn50;

if (gryz20\_afk > gryz50\_afk || time\_global == 0)

gryz50\_pog = round(xp\_rasp(1/10));

pog\_afk\_50 = gryz50\_pog + gryz50\_afk;

zagr\_eks = gryz50\_pog;

if (pog\_afk\_50 <= gryz20\_afk)

gryz20\_afk = 0;

gryz\_afk = gryz20\_afk;

else

razn20 = pog\_afk\_50 - gryz20\_afk;

gryz\_afk = razn20;

end

else

gryz20\_pog = round(xp\_rasp(1/5));

pog\_afk\_20 = gryz20\_pog + gryz20\_afk;

zagr\_eks = gryz20\_pog;

if (pog\_afk\_20 <= gryz50\_afk)

gryz50\_afk = 0;

gryz\_afk = gryz50\_afk;

else

razn50 = pog\_afk\_20 - gryz50\_afk;

gryz\_afk = razn50;

end

end

end

count = count + 1;

time\_global = time\_global + gryz50\_pog + gryz20\_pog + gryz\_afk;

gryz\_waiting\_s = round(gryz\_afk + gryz\_waiting\_s);

sum\_zagr\_eks = sum\_zagr\_eks + zagr\_eks;

end

gryz\_waiting(1,:) = [gryz\_waiting\_s ; sum\_zagr\_eks];

**end**

**function** drobilka\_z = drobilka()

time\_global = 0;

pereezd50 = 3;

count = 0;

ii = 0;

gryz20\_raz = 0;

gryz50\_raz = 0;

gryz\_afk = 0;

waiting\_gryzoviki = 0;

zagr\_drobilki = 0;

while time\_global < 480

count = count + 1;

if time\_global == 0

gryz50\_pog = round(xp\_rasp(1/10));

gryz50\_gp = gryz50\_pog + pereezd50;

gryz50\_raz = round(xp\_rasp(1/4));

time\_global = gryz50\_gp + gryz50\_raz;

raz\_gryz = gryz50\_raz;

else

for n=1:6

afk50 = round(AFK50(1/10));

afk20\_50 =round(cylce\_service(1/5,1/10));

if (ii == 3)

ii = 1;

else

ii = ii + 1;

end

i = afk50(ii);

x = afk20\_50(n);

if (x==i)

q= round(xp\_rasp(1/4));

gryz50\_raz = q + gryz\_afk;

gryz\_afk = gryz20\_raz;

raz\_gryz = q + raz\_gryz;

else

w = round(xp\_rasp(1/2));

gryz20\_raz = w + gryz\_afk;

gryz\_afk = gryz20\_raz;

raz\_gryz = w + raz\_gryz;

count = count + 1;

end

end

waiting\_gryzoviki = gryz\_afk + waiting\_gryzoviki;

zagr\_drobilki = raz\_gryz + zagr\_drobilki;

time\_global = gryz50\_raz + gryz20\_raz + time\_global;

end

end

drobilka\_z(1,:) =[waiting\_gryzoviki ;zagr\_drobilki];

**end**

**function** servise = cycle\_service(x,y)

count = 0;

pereezd20 = 2.5;

pereezd50 = 3;

for n = 1:3

count = count + 1;

gryz20\_pog = xp\_rasp(x);

s\_gr20 = round(gryz20\_pog + pereezd20);

gr20(count,:) = [s\_gr20];

end

count =0;

for n = 1:3

count = count + 1;

gryz50\_pog = xp\_rasp(y);

s\_gr50 = gryz50\_pog + pereezd50;

gr50(count,:) = [s\_gr50];

end

d = [gr20; gr50];

servise = sort(d);

**end**

**function** time\_p20 = AFK20(x)

count = 0;

pereezd20 = 2.5;

for n = 1:3

count = count + 1;

gryz20\_pog = xp\_rasp(x);

s\_gr20 = round(gryz20\_pog + pereezd20);

gr20(count,:) = [s\_gr20];

time\_p20 = sort(gr20)

end

**end**

**function** time\_p50 = AFK50(x)

count = 0;

pereezd50 = 3;

for n = 1:3

count = count + 1;

gryz50\_pog = xp\_rasp(x);

s\_gr50 = gryz50\_pog + pereezd50;

gr50(count,:) = [s\_gr50]

time\_p50 = sort(gr50)

end

**end**

# **Заключение**

В данной работе мы промоделировали работу систем на протяжении восьми часов. Построили график зависимости математического ожидания от выходных параметров. Также мы рассчитали необходимое количество экспериментов. Были составлены схемы имитационных моделей и описаны взаимодействия отдельно взятых модулей.

Программа была реализована c помощью MatLab. **MATLAB** — пакет прикладных программ для решения задач технических вычислений. Язык MATLAB является высокоуровневым интерпретируемым языком программирования, включающим основанные на матрицах структуры данных, широкий спектр функций, интегрированную среду разработки, объектно-ориентированные возможности и интерфейсы к программам, написанным на других языках программирования.

# Список литературы

1. Багрова И.А., Шахпаронян А.П. Имитационное моделирование – Тверской государственный университет 2014, 159 с.
2. Кельтон В.Д., Аверилл М. Л. Имитационное моделирование – Издательский дом «Питер» 2004, 847 c.
3. Шеннон Р. Имитационное моделирование систем – искусство и наука – Издательство «Мир» Москва 1978, 420 с.