Министерство образования и науки РФ

Федеральное государственное автономное образовательное учреждение высшего образования

Севастопольский государственный университет

Институт информационных технологий и управления в технических системах

Кафедра «Информационных Систем»

Отчет

по лабораторной работе №3

Определение показателей надежности систем

по статистическим данным для малой

выборки изделий.

Выполнил:

ст. гр. ИС22-д

Серегин А.В.

Проверил:

Колесников

Севастополь

2016

1. Цель работы.

Определение показателей надежности по статистическим данным

2. Вариант задания.

№ 21:

Время отказа: 1840, 2020, 2150, 2300, 2350, 2410, 2510, 2580, 2760, 2820, 3090, 3120, 3310.

3. Код программы.

Реализация библиотеки RIS:

#include "RIS.h"

#include <cmath>

double RIS::reliabilities(const uint64\_t& \_count, const uint64\_t& \_failsCount, const double& \_deltaTime)

{

return \_failsCount / double(\_count \* \_deltaTime);

}

double RIS::reliabilities(const uint64\_t& \_count, const uint64\_t& \_countLeft)

{

return (double)\_countLeft / (double)\_count;

}

double RIS::reliabilities(const double& \_failureIntensity, const double& \_workTime)

{

return std::exp(-\_failureIntensity \* \_workTime);

}

double RIS::failureRate(const uint64\_t& \_count, const uint64\_t& \_failsCount)

{

return 1.0 - double(\_failsCount) / double(\_count);

}

double RIS::failureIntensity(const double& \_failureRate, const double& \_reliabilities)

{

return \_failureRate / \_reliabilities;

}

double RIS::failureIntensity(const uint64\_t& \_count, const uint64\_t& \_failsCount, const double& \_deltaTime)

{

return 2.0 \* double(\_failsCount) / double(\_count + \_count - \_failsCount) / double(\_deltaTime);

}

double RIS::averageWorkTime(const uint64\_t& \_count, const uint64\_t& \_failsCount, const double& \_deltaTime)

{

return 1.0 / failureIntensity(\_count, \_failsCount, \_deltaTime);

}

double RIS::averageWorkTime(const double& \_failureIntensity)

{

return 1.0 / \_failureIntensity;

}

Main.cpp:

#include <QDebug>

#include <QVector>

#include <QTextStream>

int main()

{

QVector<double> timeOfFailure = {1840, 2020, 2150, 2300, 2350, 2410, 2510,

2680, 2760, 2820, 3090, 3120, 3310};

QVector<double> deltaTime(timeOfFailure.size() - 1);

QVector<double> reliabilities(timeOfFailure.size());

QVector<double> failureIntensity(timeOfFailure.size() - 1);

int entityesCount = timeOfFailure.size();

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

{

reliabilities[i] = (entityesCount - i - 1 + 0.7) / (entityesCount + 0.4);

deltaTime[i] = timeOfFailure[i + 1] - timeOfFailure[i];

failureIntensity[i] = 1.0 / (deltaTime[i] \* (sizeof(entityesCount - i - 1 - 0.7)));

}

reliabilities[entityesCount - 1] = (entityesCount - entityesCount + 0,7) / (entityesCount + 0.4);

qDebug() << qSetRealNumberPrecision(7) << deltaTime;

qDebug() << qSetRealNumberPrecision(7) << reliabilities;

qDebug() << qSetRealNumberPrecision(7) << failureIntensity;

return 0;

}

4. Расчетные данные.

На рисунке ниже приведены результаты вычислений, полученные в результате выполнения программы.

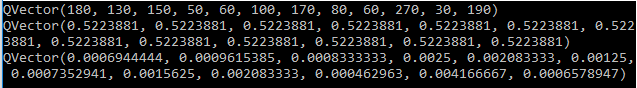


Рисунок 1 – Результаты выполнения программы.

Рисунок 2 – График зависимости вероятности

безотказной работы от времени.

Рисунок 3 – График изменения интенсивности

отказов во времени.

Выводы

В ходе лабораторной работы были проведены расчеты показателей надежности по статистическим данным.