Лабораторная работа № 8

TCP/AQM

Королёв Иван

Содержание

Список иллюстраций

Список таблиц

# 1 Цель работы

Реализовать модель TCP/AQM в xcos и OpenModelica.

# 2 Задание

1. Построить модель TCP/AQM в xcos;
2. Построить графики динамики изменения размера TCP окна и размера очереди ;
3. Построить модель TCP/AQM в OpenModelica;

# 3 Выполнение лабораторной работы 5

## 3.1 Реализация модели эпидемии в xcos

Зафиксируем начальные данные: **β = 1, ν = 0, 3, s(0) = 0, 999, i(0) = 0, 001, r(0) = 0**.

В меню моделирования устанавливаем переменные окружения (рис. 1)

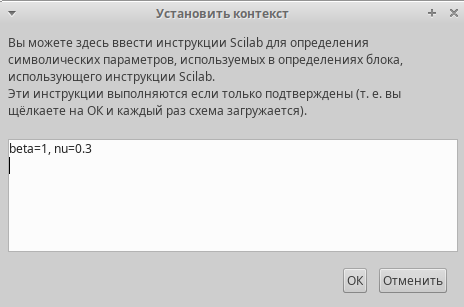


Рис. 1: beta, nu

Для реализации модели потребуется: \* CLOCK\_c — запуск часов модельного времени; \* CSCOPE — регистрирующее устройство для построения графика; \* TEXT\_f — задаёт текст примечаний; \* MUX — мультиплексер, позволяющий в данном случае вывести на графике сразу несколько кривых; \* INTEGRAL\_m — блок интегрирования \* GAINBLK\_f — в данном случае позволяет задать значения коэффициентов β и ν; \* SUMMATION — блок суммирования; \* PROD\_f — поэлементное произведение двух векторов на входе блока.

Добавляем эти блоки из палитры инструментов и строим с их помощью данную систему дифференциальных уравнений:

где – скорость заражения, – скорость выздоровления.

Реализованная модель эпидемии. Выходы трёх блоков интегрирования соединяем с мультиплексором.(рис. 2)

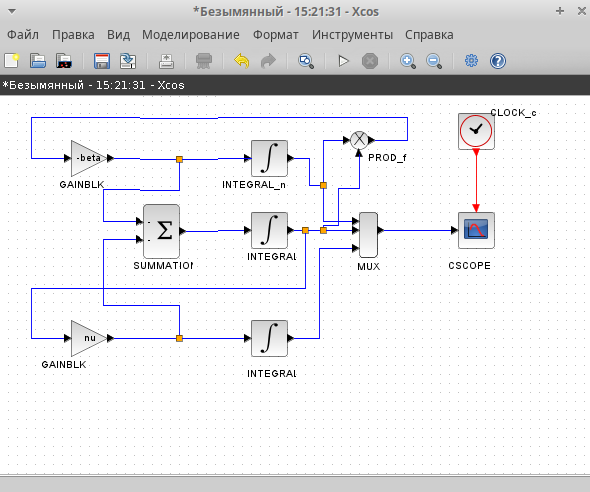


Рис. 2: Реализованная модель эпидемии

В параметрах верхнего блока интегрирования задаем значения s(0) = 0, 999, который отвечает за здоровых особей. (рис. 3)

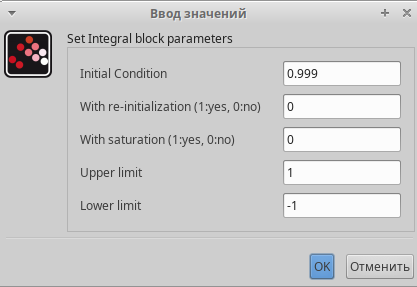


Рис. 3: Начальные значения для верхнего блока интегрирования

В параметрах среднего блока интегрирования задаем значения i(0) = 0, 001, который отвечает за переносчиков болезни. (рис. 4)

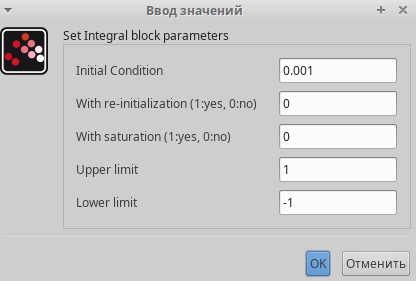


Рис. 4: Начальные значения для среднего блока интегрирования

В нижнем блоке интегрирования начальные значения по умолчанию заданы нулю, как в нашем условии. Данная часть отвечает за тех, кто имеет иммунитет.

Далее, устанавливаем конечное время интегрирования. Оно равно 30 (рис. 5)

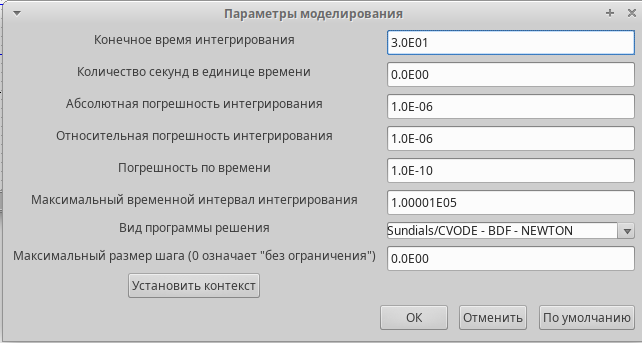


Рис. 5: Конечное время интегрирования

Результат моделирования представлен на (рис. 6), где черной линией обозначен график s(t) (динамика численности уязвимых к болезни особей), красная линия r(t) — динамику численности выздоровевших особей, наконец, зеленая линия i(t) — динамику численности заражённых особей. Пересечение трёх линий определяет порог эпидемии.

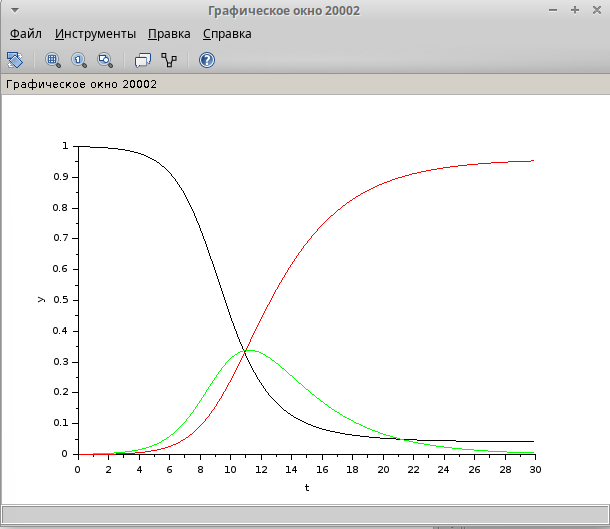


Рис. 6: Модель эпидемии при beta=1, nu=0.3

## 3.2 Реализация модели с помощью блока Modelica в xcos

В данном задании необходимо было реализовать такую же модель эпидемии при beta=1, nu=0.3, только с помощью блока Modelica в xcos. Для начала добавляем новый блок констант и блок реализации кода на Modelica. Таким образом выглядит наша модель (рис. 7)

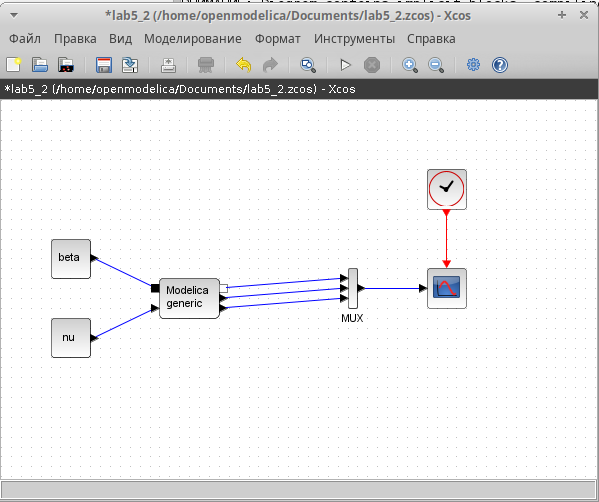


Рис. 7: Модель эпидемии

Указываем параметры для блока реализации. Переменные на входе (“beta”, “nu”) и выходе (“s”, “i”, “r”) блока заданы как внешние (“E”). (рис. 8)

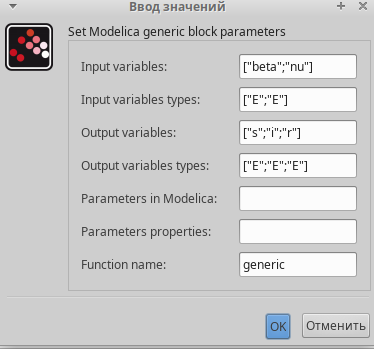


Рис. 8: Параметры блока реализации

Код на языке Modelica. Задаем переменные beta, nu. Указываем начальные значения для s, i, r и пишем систему уравнения. (рис. 9)

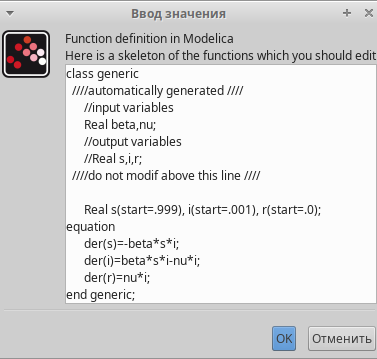


Рис. 9: Параметры блока реализации

Результат работы модели. Он идентичен с реализацией в xcos. (рис. 10)

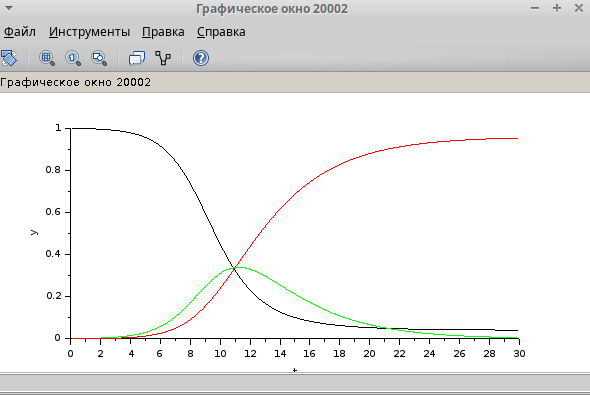


Рис. 10: Модель эпидемии Modelica

## 3.3 Выполнение упражнени построения модели эпидемии в OpenModelica

Код реализации модели эпидемии в OpenModelica. Задаем все начальные параметры с помощью parameter Real, как было в реализациях xcos. Записываем систему уравнения, реализация очень сильно схожа с реализацией с помощью блока Modelica в xcos (рис. 11)

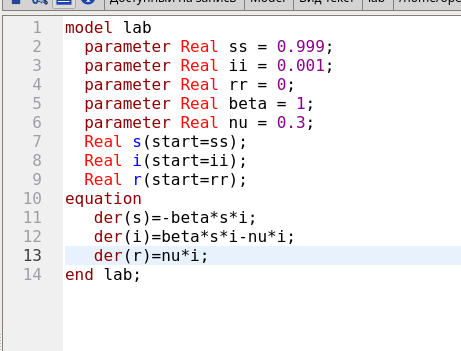


Рис. 11: Реализация модели эпидемии в OpenModelica

Результат модели. Результат идентичен с построением с помощью других способов, значит все выполнено правильно. (рис. 12)

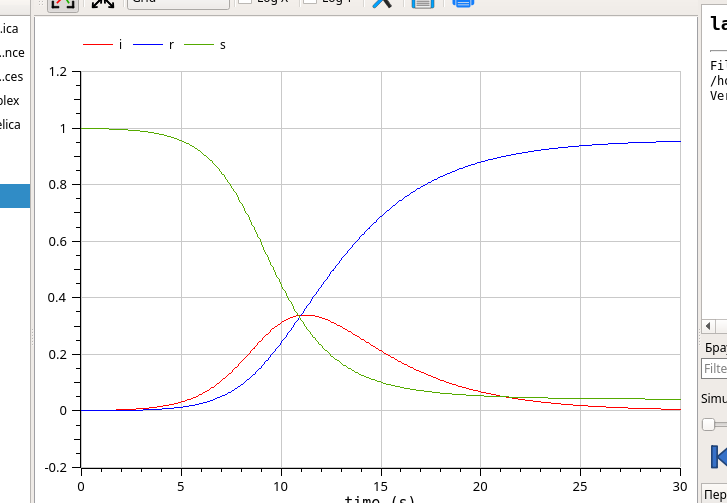


Рис. 12: Модель эпидемии в OpenModelica

## 3.4 Задание для самостоятельного выполнения. Реализация с помощью xcos

Необходимо реализовать такую же модель эпидемии, только с учётом процесса рождения / гибели особей в xcos (в том числе и с использованием блока Modelica), а также в OpenModelica.

Так выглядит система уравнения:

где — константа, которая равна коэффициенту смертности и рождаемости.

Реализуем эту модель в *xcos*. Тут нам понадобятся три блока суммирования и 4 блока констант (добавляется константа ).

В меню моделирования устанавливаем переменные окружения. (рис. 13) Реализация с помощью xcos. (рис. 14)

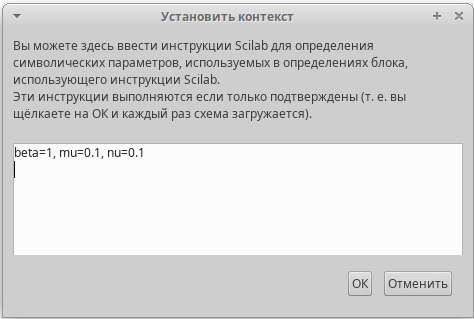


Рис. 13: Переменные окружения

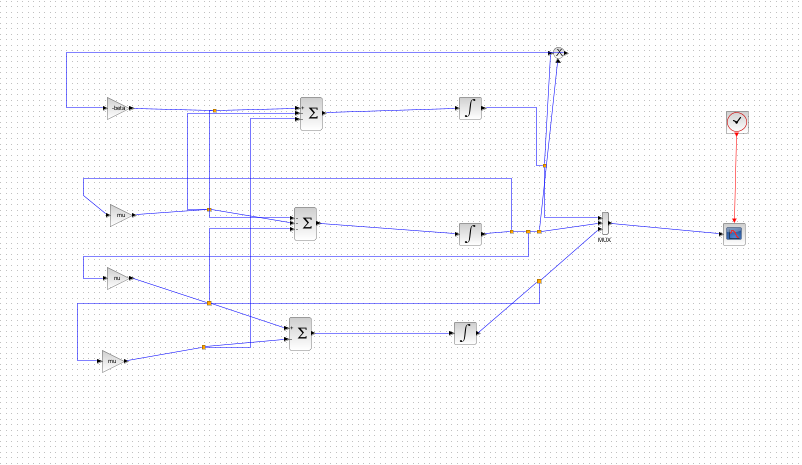


Рис. 14: Реализация модели эпидемии с учетом процесса рождения / гибели особей с помощью xcos

В параметрах блоков интегрирования нет изменений, указываем все начальные значения из предыдущих этапов выполнения.

Результат моделирования представлен на (рис. 15), где черной линией обозначен график s(t) (динамика численности уязвимых к болезни особей), красная линия r(t) — динамику численности выздоровевших особей, наконец, зеленая линия i(t) — динамику численности заражённых особей. Пересечение трёх линий определяет порог эпидемии.

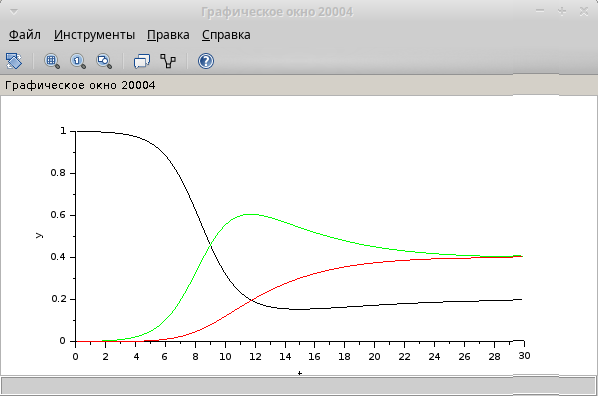


Рис. 15: Модель эпидемии при beta=1, nu=0.1, mu=0.1

## 3.5 Задание для самостоятельного выполнения. Реализация с помощью блока Modelica в xcos

Реализация с помощью блока Modelica в xcos. (рис. 16)

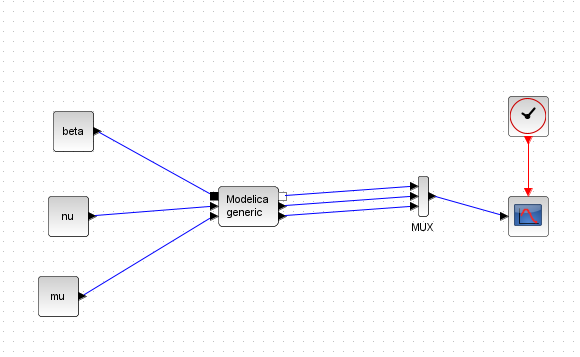


Рис. 16: Реализация модели эпидемии с учетом процесса рождения / гибели особей с помощью блока Modelica в xcos

Указываем параметры для блока реализации. Переменные на входе (“beta”, “nu”, “mu”) и выходе (“s”, “i”, “r”) блока заданы как внешние (“E”). (рис. 17)

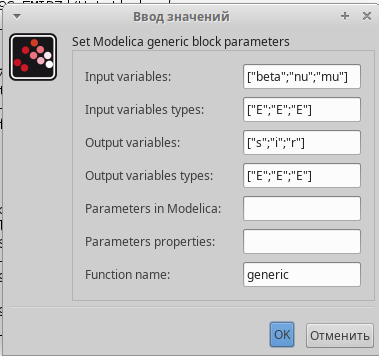


Рис. 17: Параметры блока реализации

Код на языке Modelica. Задаем переменные beta, nu, mu. Указываем начальные значения для s, i, r и пишем систему уравнения. (рис. 18)

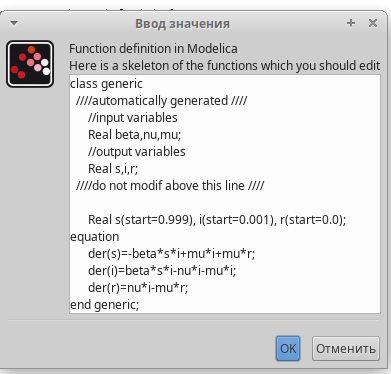


Рис. 18: Параметры блока реализации

Результат моделирования представлен на (рис. 19), где черной линией обозначен график s(t) (динамика численности уязвимых к болезни особей), красная линия r(t) — динамику численности выздоровевших особей, наконец, зеленая линия i(t) — динамику численности заражённых особей. Пересечение трёх линий определяет порог эпидемии.

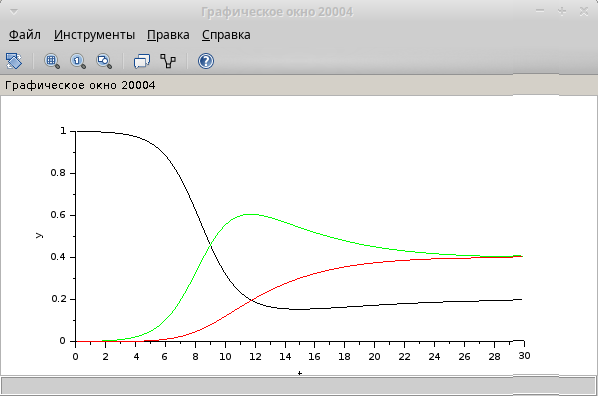


Рис. 19: Модель эпидемии при beta=1, nu=0.1, mu=0.1

## 3.6 Задание для самостоятельного выполнения. Реализация в OpenModelica

Код реализации модели эпидемии в OpenModelica. Задаем все начальные параметры с помощью parameter Real, как было в реализациях xcos. Записываем систему уравнения, реализация очень сильно схожа с реализацией с помощью блока Modelica в xcos (рис. 20)

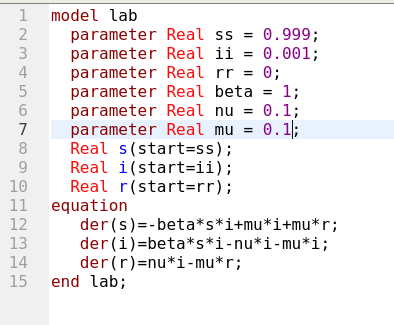


Рис. 20: Реализация модели с учетом процесса рождения / гибели особей эпидемии в OpenModelica

Результат модели. Результат идентичен с построением с помощью других способов, значит все выполнено правильно. (рис. 21)

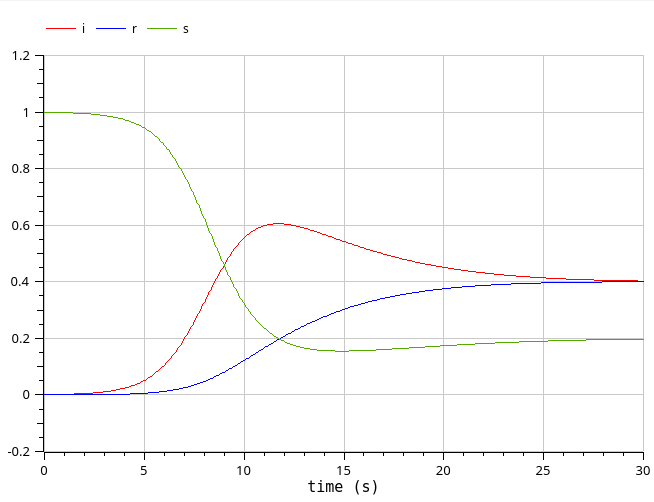


Рис. 21: Модель эпидемии с учетом процесса рождения / гибели особей в OpenModelica

## 3.7 Результаты на различных параметрах.

При mu=0.6, nu=0.1, beta=1 (рис. 22), (рис. 23)

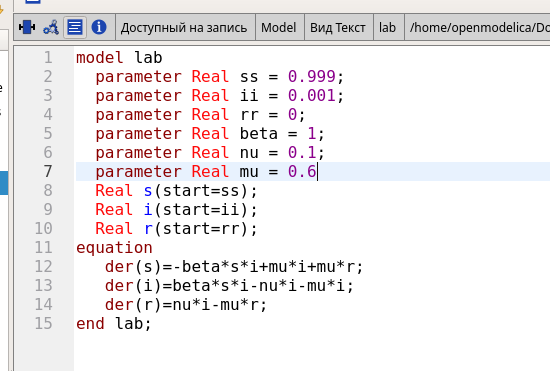


Рис. 22: Результаты на различных параметрах.

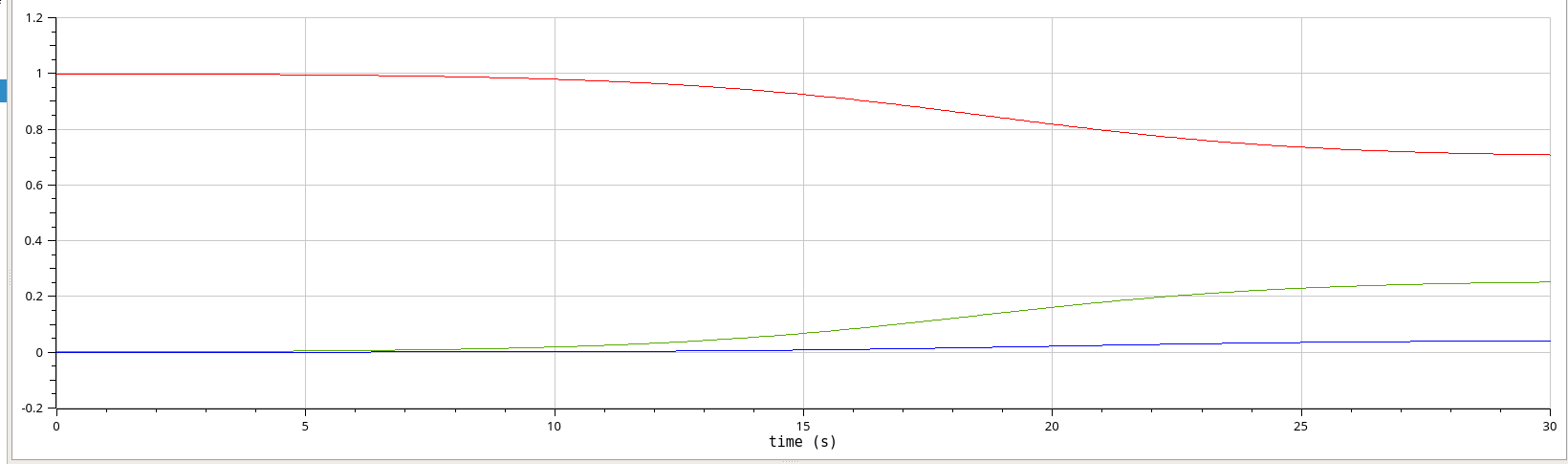


Рис. 23: Результаты на различных параметрах.

При mu=0.6, nu=0.6, beta=1 (рис. 24), (рис. 25)

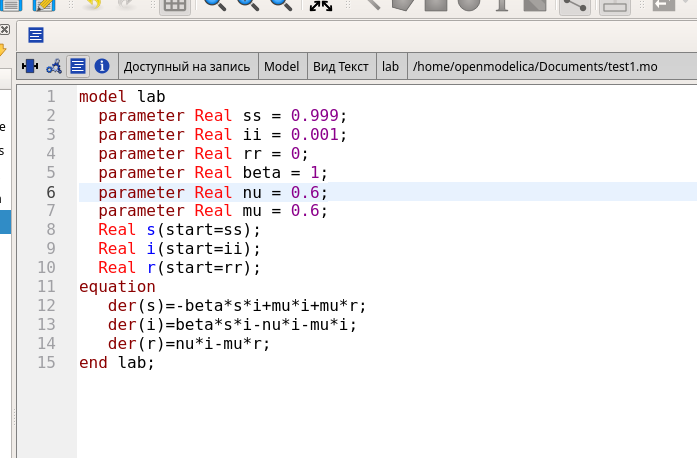


Рис. 24: Результаты на различных параметрах.

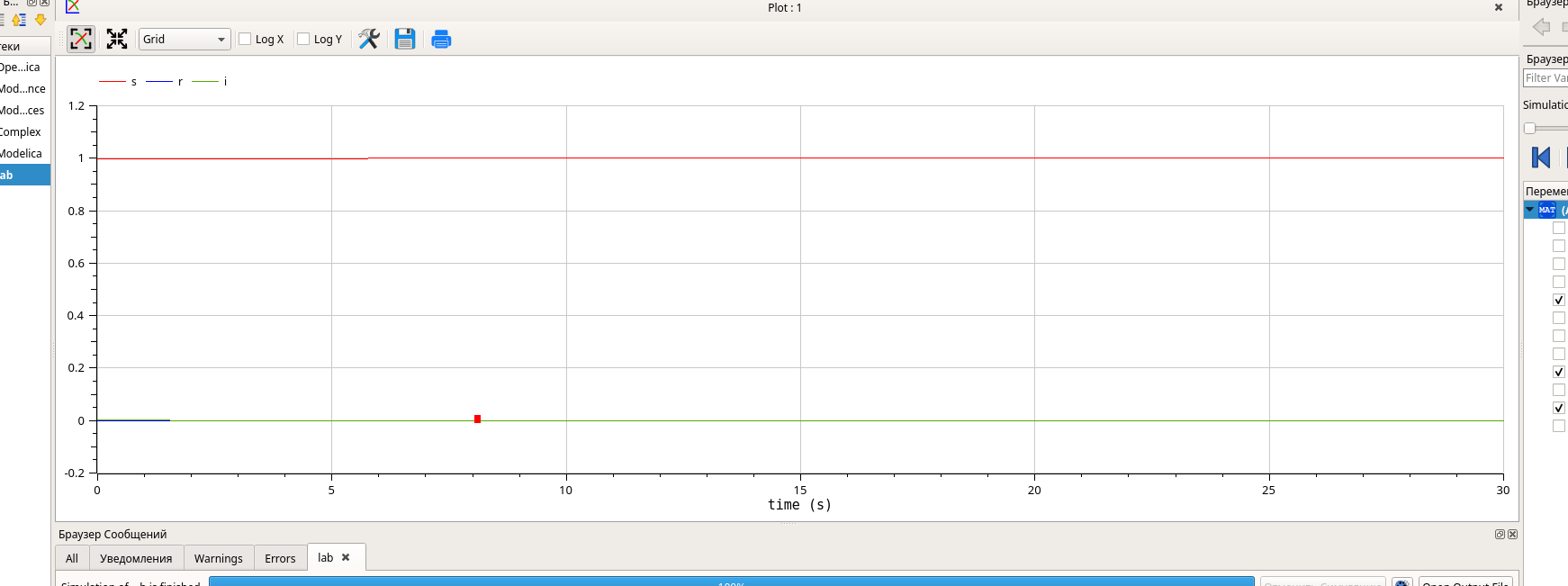


Рис. 25: Результаты на различных параметрах.

Исходя из анализа графиков, можно сделать вывод, что чем выше значение любого из параметров, тем быстрее система достигает стационарного состояния. При высоком коэффициенте заражения 𝛽 система быстро проходит через пик развития эпидемии и достигает стационарного состояния.

# 4 Выполнение лабораторной работы 6

## 4.1 Реализация модели в xcos

Для начала фиксируем начальные данные a=2,b=1,c=0.3,d=1.(рис. 26).

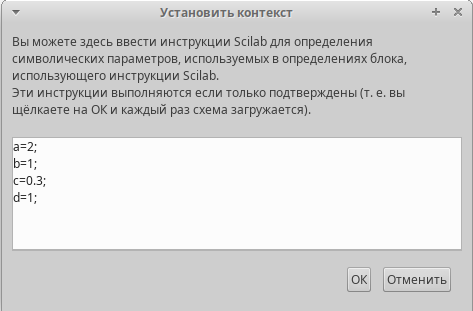


Рис. 26: Константы

Реализуем модель хищник-жертва с помощью блоков. Все блоки идентичны с предыдущей лабораторной, блок времени, блок произведение, интегрирования, суммы и тд. Только дополнительно потребуется блок регистрирующее устройство для построения фазового портрета. (CSCOPXY). Первое уравнение модели задано верхним блоком интегрирования, блоком произведения и блоками задания коэффициентов a и b. Второе уравнение модели задано нижним блоком интегрирования и блоками задания коэффициентов. (рис. 48).

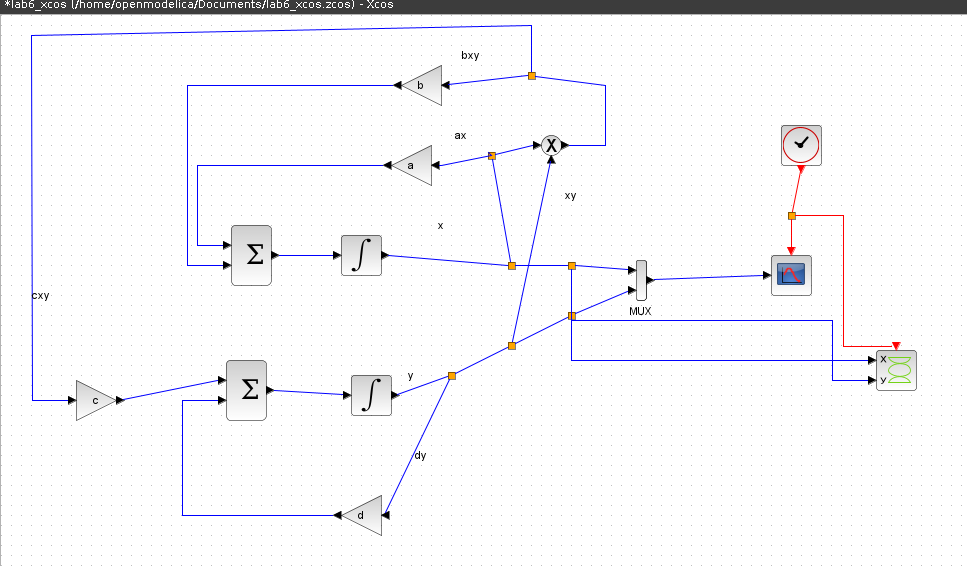


Рис. 27: Реализация модели

Задаем начальные значения для x и y в параметрах блоков интегрирования. (рис. 28), (рис. 29)

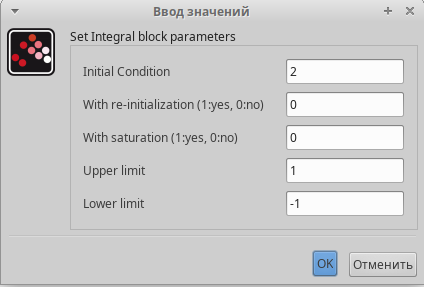


Рис. 28: Начальные значения

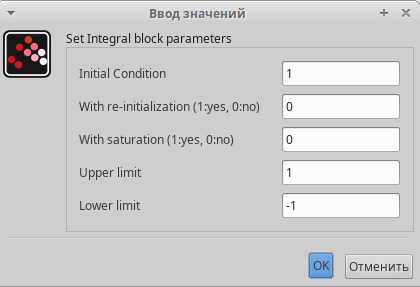


Рис. 29: Начальные значения

Устанавливаем конечное время интегрирования 30. (рис. 30)

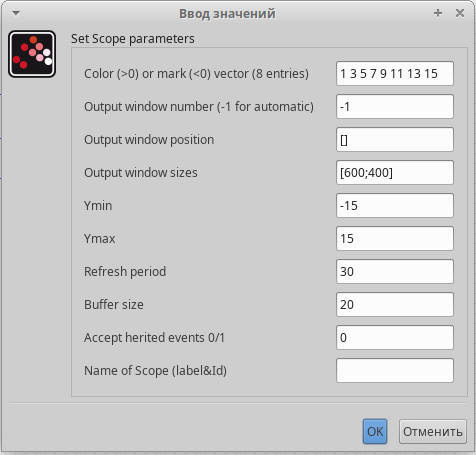


Рис. 30: конечное время интегрирования

Фазовый портрет. (рис. 31)

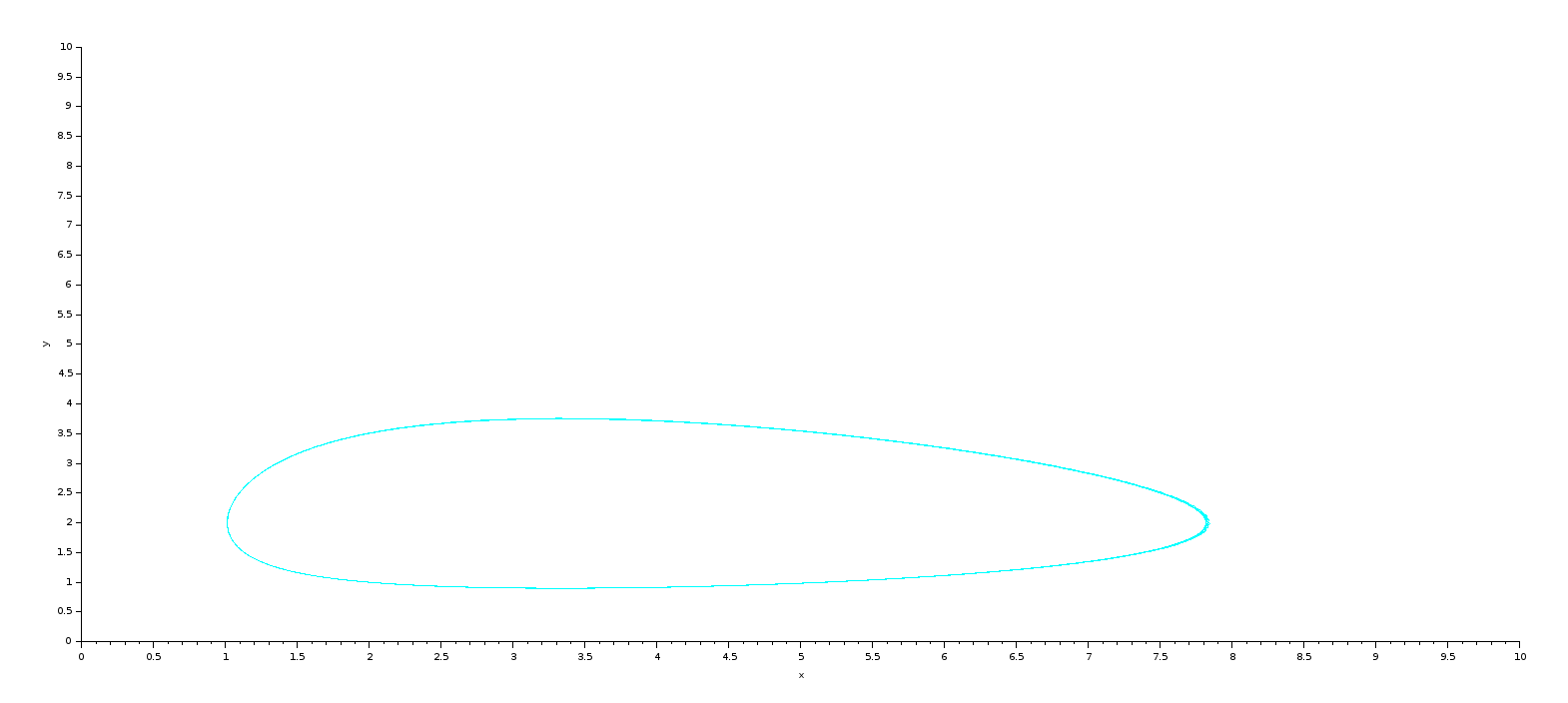


Рис. 31: Фазовый портрет.

Динамика изменения численности хищников и жертв. Черной линией обозначена динамика численности жертв. Зеленой линией обозначена динамика численности хищников. (рис. 32)

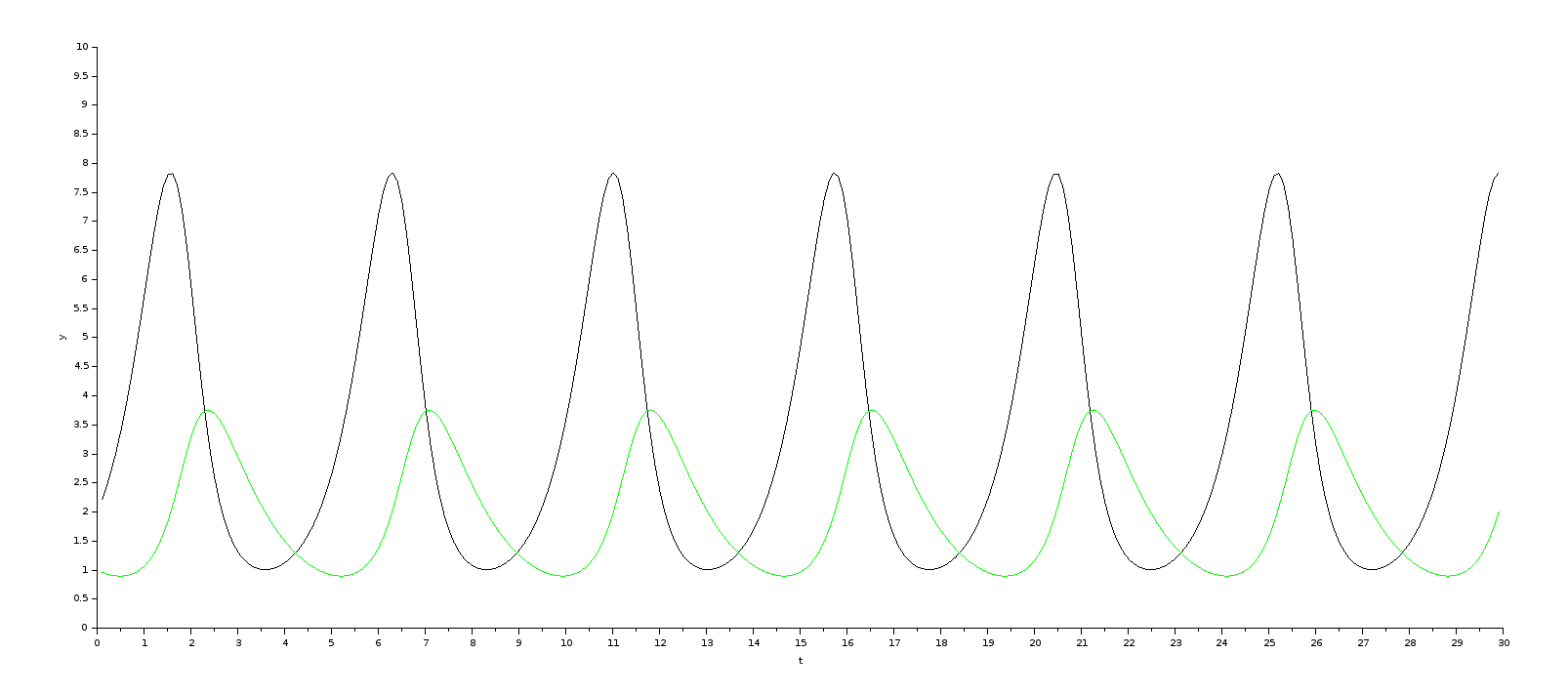


Рис. 32: Динамика изменения численности хищников и жертв

## 4.2 Реализация модели с помощью блока Modelica в xcos

Как и ранее, задаем значения коэффициентам a,b,c,d. Устанавливаем конечное время интегрирования. Реализуем модель. Нам понадобится блок моделирования, блок констант и регистрирующее устройство для построения фазового портрета и для построения графика. (рис. 33)

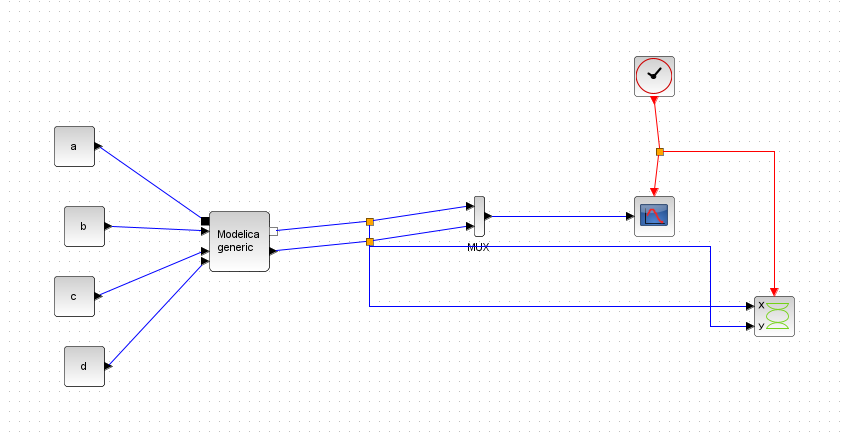


Рис. 33: Реализация модели

Параметры блока моделирования и программный код (рис. 34), (рис. 35)

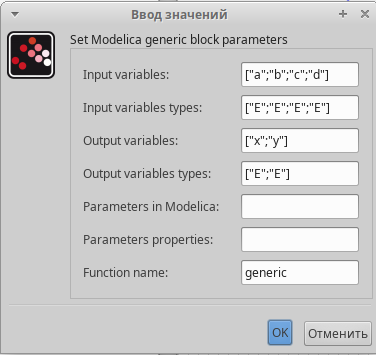


Рис. 34: Параметры блока моделирования

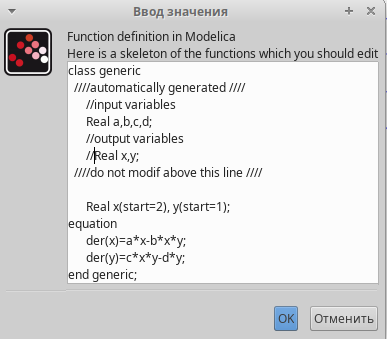


Рис. 35: Параметры блока моделирования

Фазовый портрет и график изменения численности популяций. Результат полностью идентичен с xcos. (рис. 36), (рис. 37)

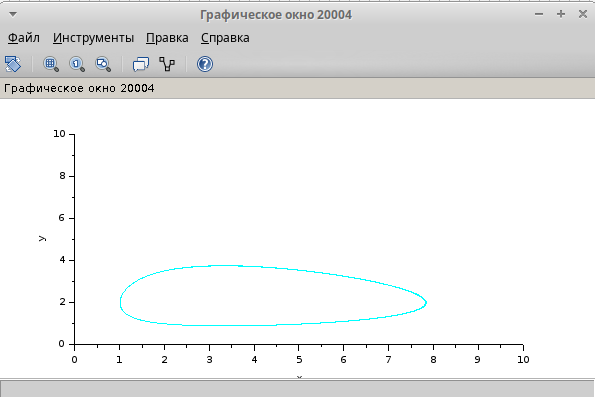


Рис. 36: Фазовый портрет

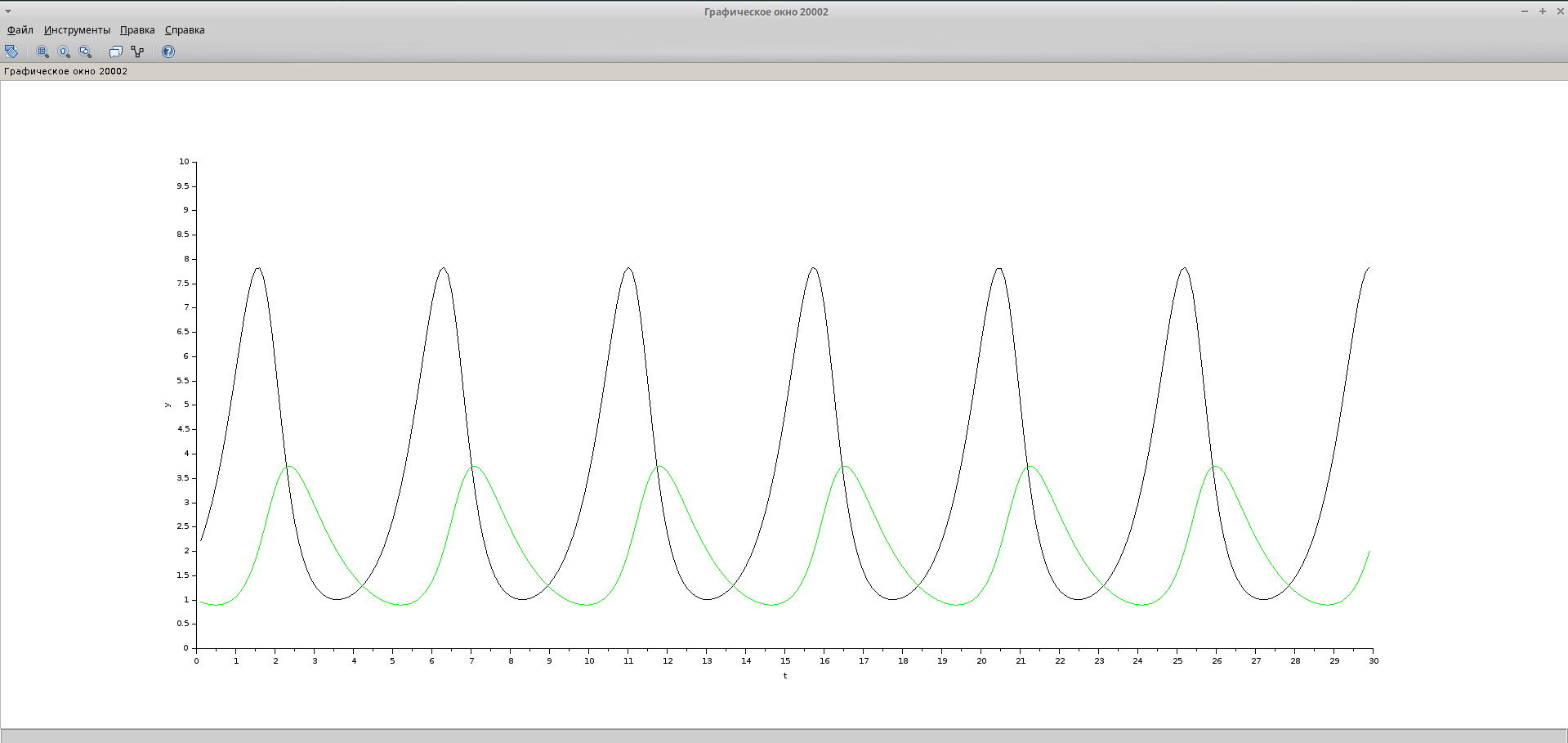


Рис. 37: график изменения численности популяций

## 4.3 Реализация модели в OpenModelica.

Код для реализации данной модели. Задаем начальные коэффициенты и пишем уравнения модели. Задаем конечное время интегрирования. (рис. 38)

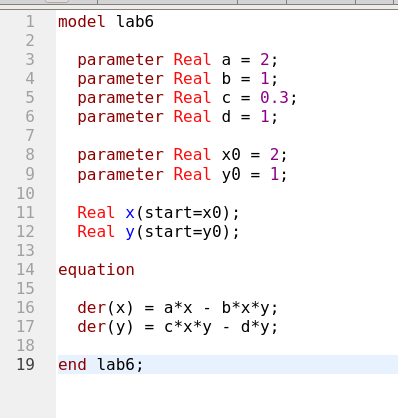


Рис. 38: Реализация модели

Фазовый портрет и график изменения численности популяций. Результат полностью идентичен с предыдущими реализациями. (рис. 39), (рис. 40)

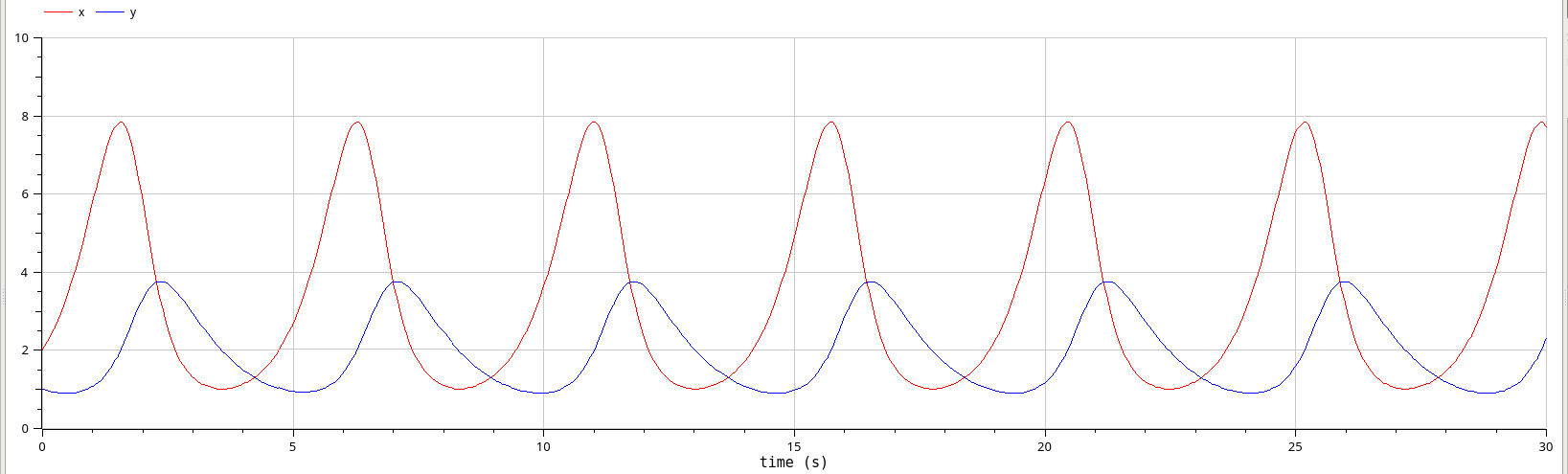


Рис. 39: Фазовый портрет

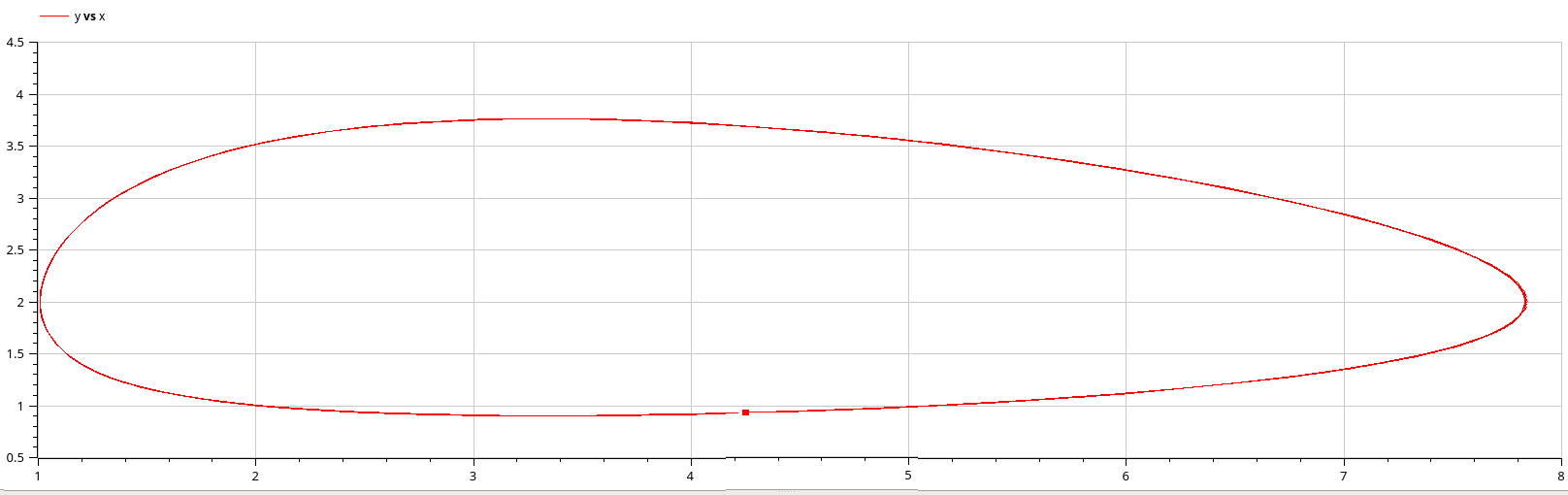


Рис. 40: график изменения численности популяций

# 5 Выполнение лабораторной работы 7

Реализация модели системы массового обслуживания типа . Для начала необходимо указать начальные параметры. (рис. 41).

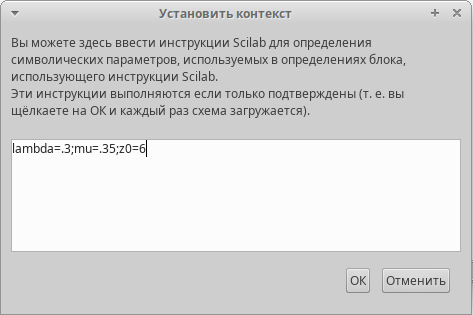


Рис. 41: Установка контекста моделирования

Построение суперблока отвечающего за поступление заявок. (рис. 42)

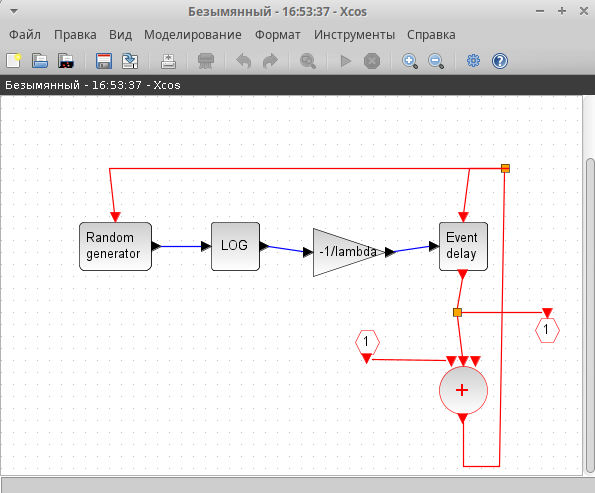


Рис. 42: Суперблок, моделирующий поступление заявок

Построение суперблока отвечающего за обработку заявок. (рис. 43)

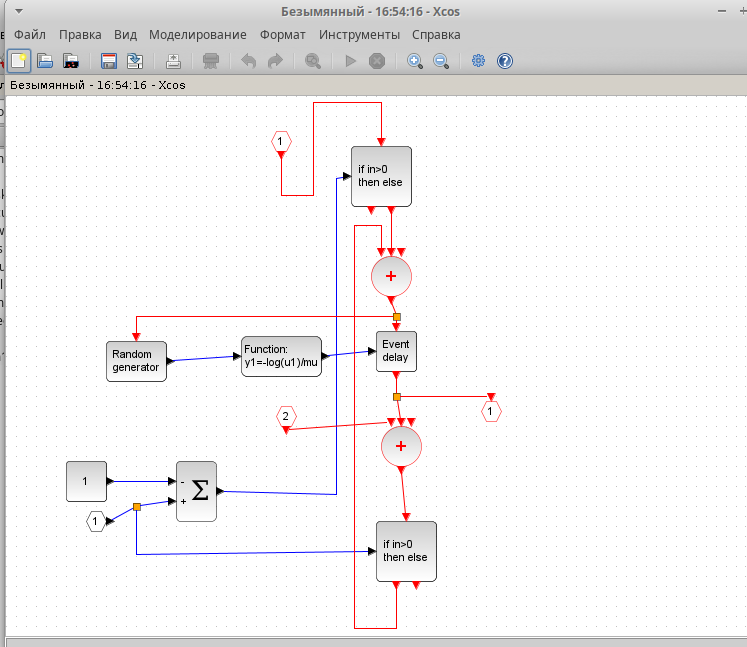


Рис. 43: Суперблок, моделирующий обработку заявок

Модель (рис. 44)

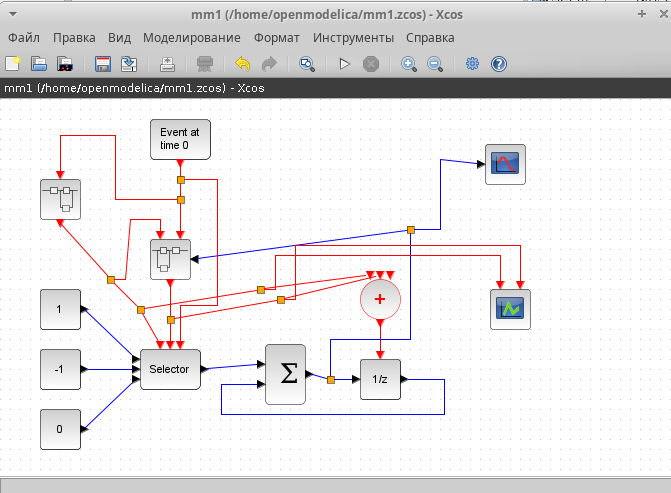


Рис. 44:

График поступления и обработки заявок (рис. 45)

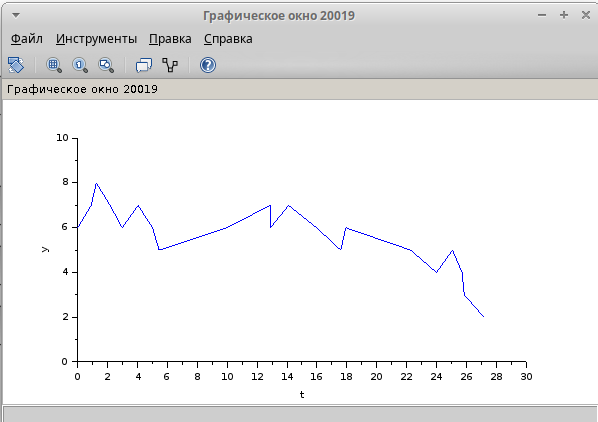


Рис. 45: График поступления и обработки заявок

График динамики размера очереди (рис. 46)

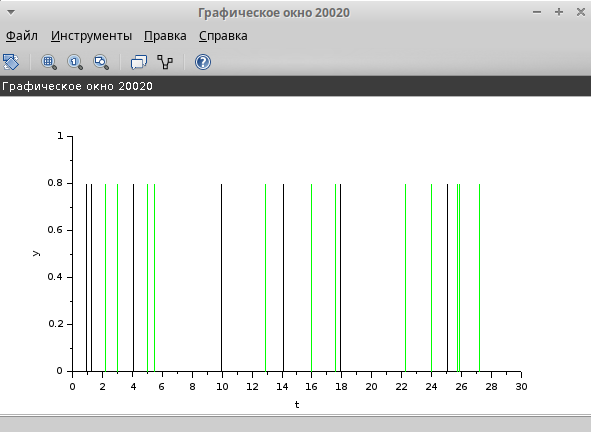


Рис. 46: График динамики размера очереди

# 6 Выполнение лабораторной работы 8

## 6.1 Реализация в xcos

Построим схему xcos, моделирующую нашу систему, с начальными значениями параметров . Для этого сначала зададим переменные окружения (рис. 47).

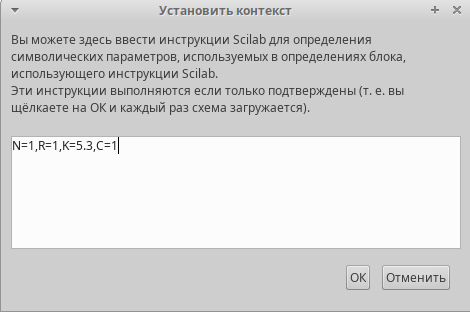


Рис. 47: Установка контекста

Затем реализуем модель TCP/AQM, разместив блоки интегрирования, суммирования, произведения, констант, а также регистрирующие устройства (рис. 48):

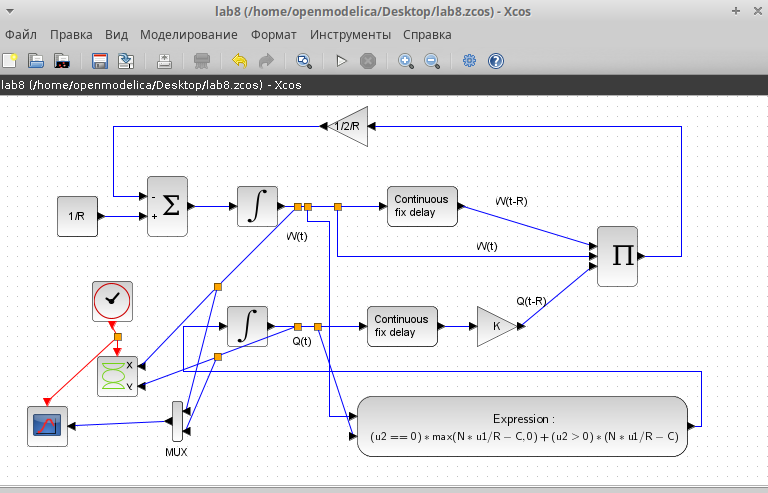


Рис. 48: Модель TCP/AQM в xcos

В результате получим динамику изменения размера TCP окна W(t) (зеленая линия) и размера очереди Q(t) (черная линия), а также фазовый портрет, который показывает наличие автоколебаний параметров системы — фазовая траектория осциллирует вокруг своей стационарной точки (рис. 49, 50):

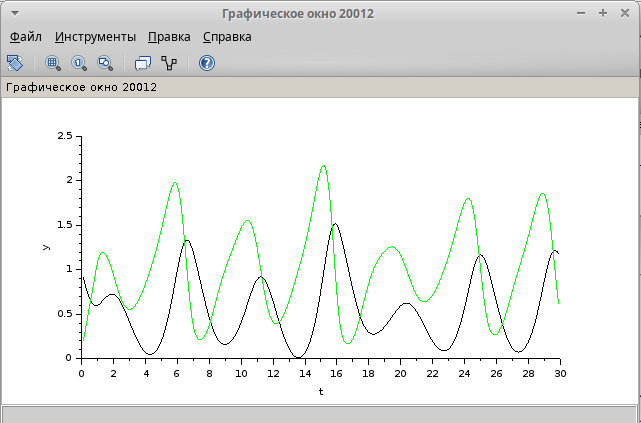


Рис. 49: Динамика изменения размера TCP окна W (t) и размера очереди Q(t)

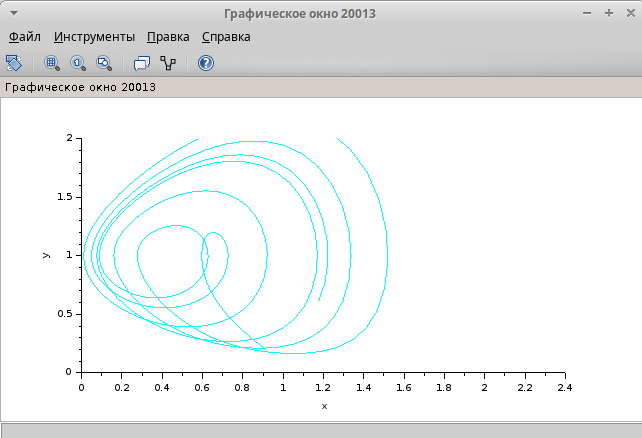


Рис. 50: Фазовый портрет (W, Q)

Уменьшив скорость обработки пакетов до увидим, что автоколебания стали более выраженными (рис. 51, 52).

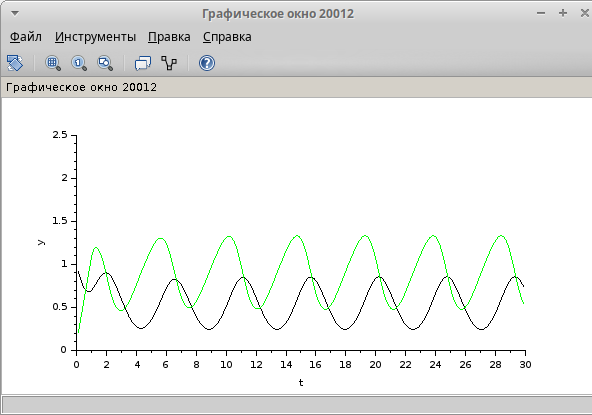


Рис. 51: Динамика изменения размера TCP окна W (t) и размера очереди Q(t) при С = 0.9

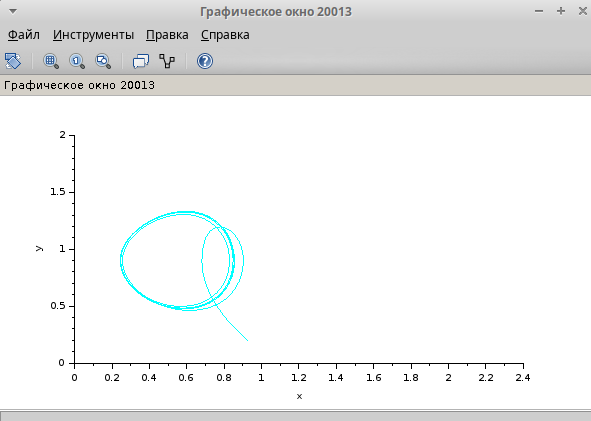


Рис. 52: Фазовый портрет (W, Q) при С = 0.9

## 6.2 Реализация модели в OpenModelica

Перейдем к реализации модели в OpenModelica. Зададим параметры, начальные значения и систему уравнений.

parameter Real N=1;  
parameter Real R=1;  
parameter Real K=5.3;  
parameter Real C=1;  
  
Real W(start=0.1);  
Real Q(start=1);  
  
equation  
  
der(W)= 1/R - W\*delay(W, R)/(2\*R)\*K\*delay(Q, R);  
der(Q)= if (Q==0) then max(N\*W/R-C,0) else (N\*W/R-C);

Выполнив симуляцию, получим динамику изменения размера TCP окна W(t)(зеленая линия) и размера очереди Q(t)(черная линия), а также фазовый портрет, который показывает наличие автоколебаний параметров системы — фазовая траектория осциллирует вокруг своей стационарной точки (рис. 53, 54).

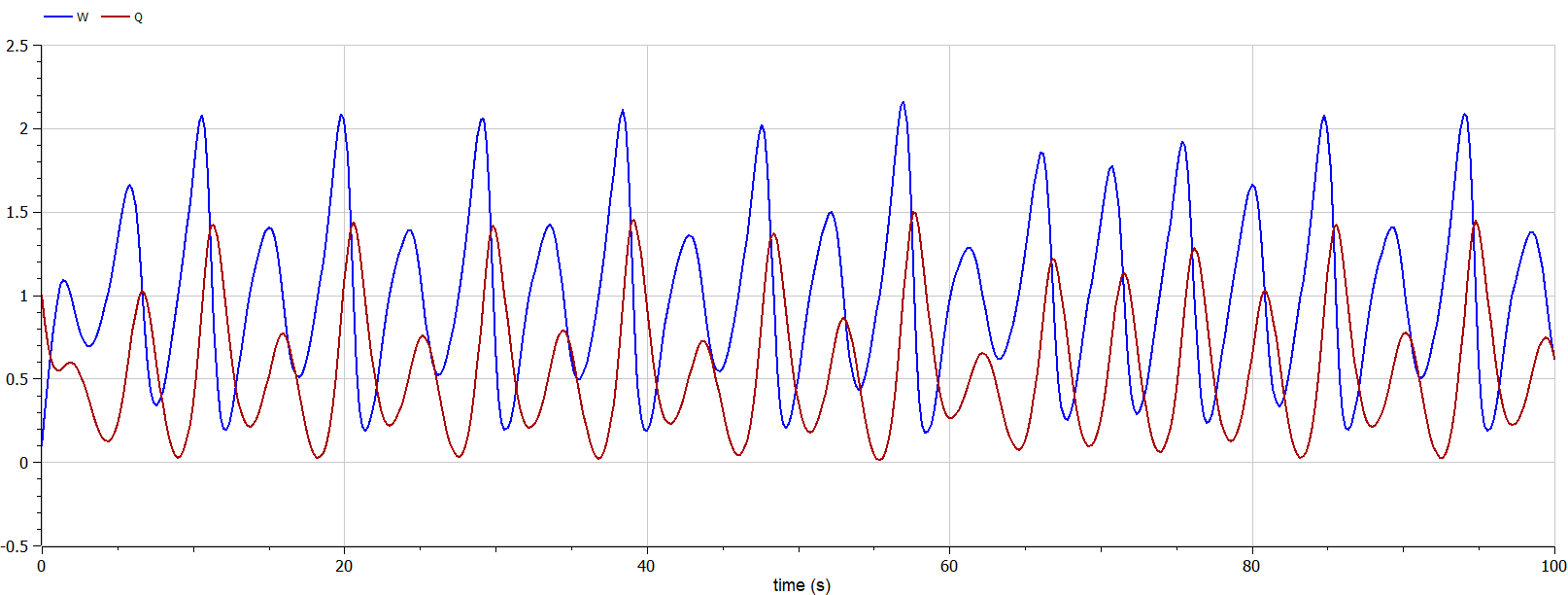


Рис. 53: Динамика изменения размера TCP окна W (t) и размера очереди Q(t). OpenModelica

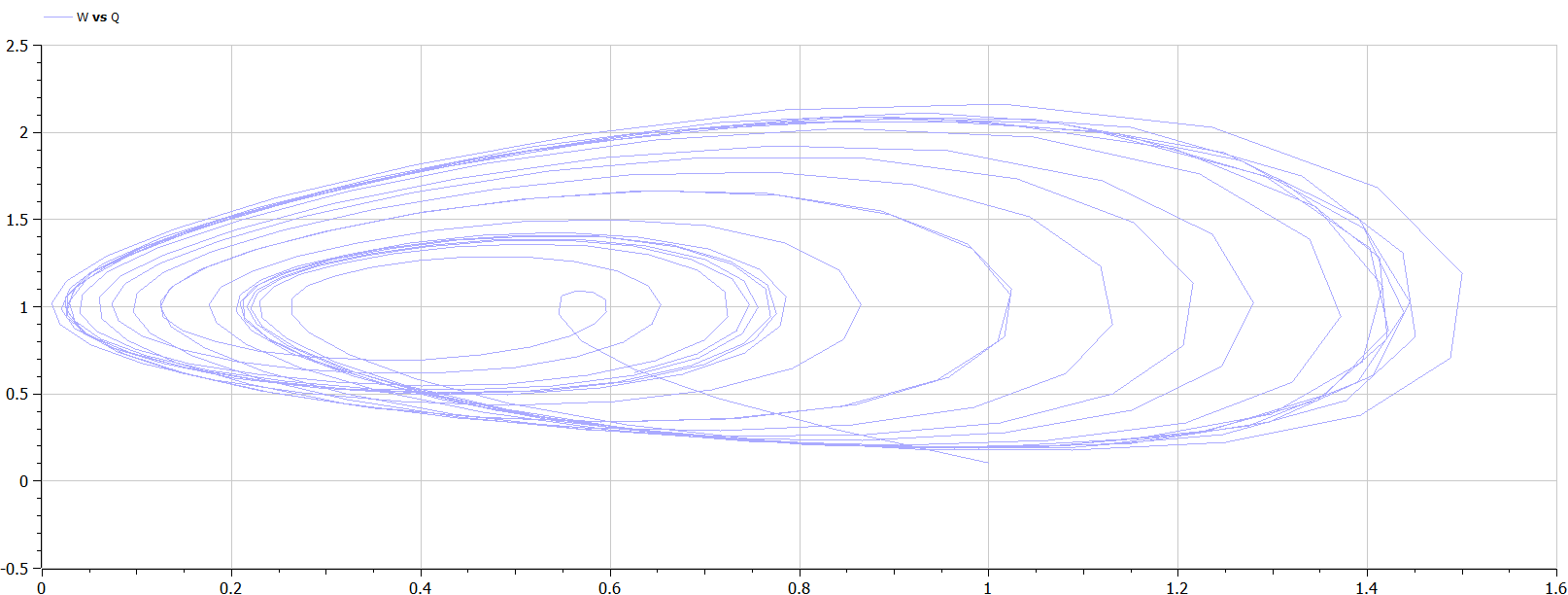


Рис. 54: Фазовый портрет (W, Q). OpenModelica

# 7 Выводы

В процессе выполнения данной лабораторной работы я реализовал модель TCP/AQM в xcos и OpenModelica.