

Лабораторная работа № 8

ТСР/АQM

Королёв Иван

Содержание

| | | |
|----------|---|-----------|
| 1 | Цель работы | 6 |
| 2 | Задание | 7 |
| 3 | Выполнение лабораторной работы 5 | 8 |
| 3.1 | Реализация модели эпидемии в xcos | 8 |
| 3.2 | Реализация модели с помощью блока Modelica в xcos | 12 |
| 3.3 | Выполнение упражнения построения модели эпидемии в OpenModelica | 15 |
| 3.4 | Задание для самостоятельного выполнения. Реализация с помощью xcos | 16 |
| 3.5 | Задание для самостоятельного выполнения. Реализация с помощью блока Modelica в xcos | 18 |
| 3.6 | Задание для самостоятельного выполнения. Реализация в OpenModelica | 21 |
| 3.7 | Результаты на различных параметрах. | 23 |
| 4 | Выполнение лабораторной работы 6 | 26 |
| 4.1 | Реализация модели в xcos | 26 |
| 4.2 | Реализация модели с помощью блока Modelica в xcos | 29 |
| 4.3 | Реализация модели в OpenModelica. | 32 |
| 5 | Выполнение лабораторной работы 7 | 35 |
| 6 | Выполнение лабораторной работы 8 | 39 |
| 6.1 | Реализация в xcos | 39 |
| 6.2 | Реализация модели в OpenModelica | 42 |
| 7 | Выводы | 44 |

Список иллюстраций

| | | |
|------|---|----|
| 3.1 | beta, nu | 8 |
| 3.2 | Реализованная модель эпидемии | 9 |
| 3.3 | Начальные значения для верхнего блока интегрирования | 10 |
| 3.4 | Начальные значения для среднего блока интегрирования | 10 |
| 3.5 | Конечное время интегрирования | 11 |
| 3.6 | Модель эпидемии при beta=1, nu=0.3 | 11 |
| 3.7 | Модель эпидемии | 12 |
| 3.8 | Параметры блока реализации | 13 |
| 3.9 | Параметры блока реализации | 14 |
| 3.10 | Модель эпидемии Modelica | 14 |
| 3.11 | Реализация модели эпидемии в OpenModelica | 15 |
| 3.12 | Модель эпидемии в OpenModelica | 16 |
| 3.13 | Переменные окружения | 17 |
| 3.14 | Реализация модели эпидемии с учетом процесса рождения / гибели особей с помощью xsos | 17 |
| 3.15 | Модель эпидемии при beta=1, nu=0.1, mu=0.1 | 18 |
| 3.16 | Реализация модели эпидемии с учетом процесса рождения / гибели особей с помощью блока Modelica в xsos | 18 |
| 3.17 | Параметры блока реализации | 19 |
| 3.18 | Параметры блока реализации | 20 |
| 3.19 | Модель эпидемии при beta=1, nu=0.1, mu=0.1 | 21 |
| 3.20 | Реализация модели с учетом процесса рождения / гибели особей эпидемии в OpenModelica | 22 |
| 3.21 | Модель эпидемии с учетом процесса рождения / гибели особей в OpenModelica | 23 |
| 3.22 | Результаты на различных параметрах. | 23 |
| 3.23 | Результаты на различных параметрах. | 24 |
| 3.24 | Результаты на различных параметрах. | 24 |
| 3.25 | Результаты на различных параметрах. | 24 |
| 4.1 | Константы | 26 |
| 4.2 | Реализация модели | 27 |
| 4.3 | Начальные значения | 27 |
| 4.4 | Начальные значения | 28 |
| 4.5 | конечное время интегрирования | 28 |
| 4.6 | Фазовый портрет. | 29 |
| 4.7 | Динамика изменения численности хищников и жертв | 29 |

| | | |
|------|--|----|
| 4.8 | Реализация модели | 30 |
| 4.9 | Параметры блока моделирования | 30 |
| 4.10 | Параметры блока моделирования | 31 |
| 4.11 | Фазовый портрет | 31 |
| 4.12 | график изменения численности популяций | 32 |
| 4.13 | Реализация модели | 33 |
| 4.14 | Фазовый портрет | 33 |
| 4.15 | график изменения численности популяций | 34 |
| 5.1 | Установка контекста моделирования | 35 |
| 5.2 | Суперблок, моделирующий поступление заявок | 36 |
| 5.3 | Суперблок, моделирующий обработку заявок | 36 |
| 5.4 | $M M 1 \infty$ | 37 |
| 5.5 | График поступления и обработки заявок | 37 |
| 5.6 | График динамики размера очереди | 38 |
| 6.1 | Установка контекста | 39 |
| 6.2 | Модель TCP/AQM в xcos | 40 |
| 6.3 | Динамика изменения размера TCP окна $W(t)$ и размера очереди $Q(t)$ | 40 |
| 6.4 | Фазовый портрет (W, Q) | 41 |
| 6.5 | Динамика изменения размера TCP окна $W(t)$ и размера очереди $Q(t)$ при $C = 0.9$ | 41 |
| 6.6 | Фазовый портрет (W, Q) при $C = 0.9$ | 42 |
| 6.7 | Динамика изменения размера TCP окна $W(t)$ и размера очереди $Q(t)$. OpenModelica | 43 |
| 6.8 | Фазовый портрет (W, Q) . OpenModelica | 43 |

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 $W(t)$ и размера очереди $Q(t)$;
3. Построить модель TCP/AQM в OpenModelica;

3 Выполнение лабораторной работы 5

3.1 Реализация модели эпидемии в xcos

Зафиксируем начальные данные: $\beta = 1$, $\nu = 0,3$, $s(0) = 0,999$, $i(0) = 0,001$, $r(0) = 0$.

В меню моделирования устанавливаем переменные окружения (рис. 3.1)

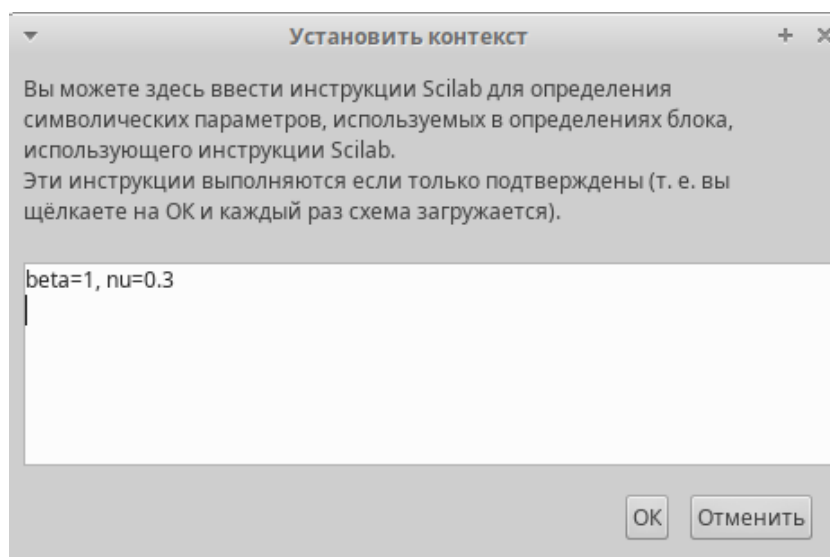


Рис. 3.1: beta, nu

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:

$$\begin{cases} \dot{s} = -\beta s(t)i(t); \\ \dot{i} = \beta s(t)i(t) - \nu i(t); \\ \dot{r} = \nu i(t), \end{cases}$$

где β – скорость заражения, ν – скорость выздоровления.

Реализованная модель эпидемии. Выходы трёх блоков интегрирования соединяем с мультиплексором.(рис. 3.2)

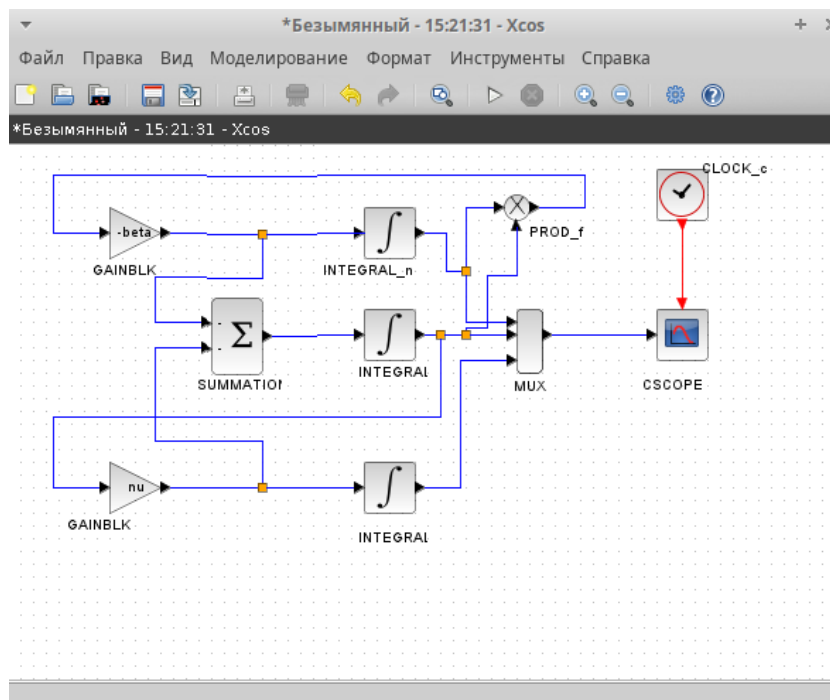


Рис. 3.2: Реализованная модель эпидемии

В параметрах верхнего блока интегрирования задаем значения $s(0) = 0, 999$, который отвечает за здоровых особей. (рис. 3.3)

Рис. 3.3: Начальные значения для верхнего блока интегрирования

В параметрах среднего блока интегрирования задаем значения $i(0) = 0,001$, который отвечает за переносчиков болезни. (рис. 3.4)

Рис. 3.4: 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. 3.5)

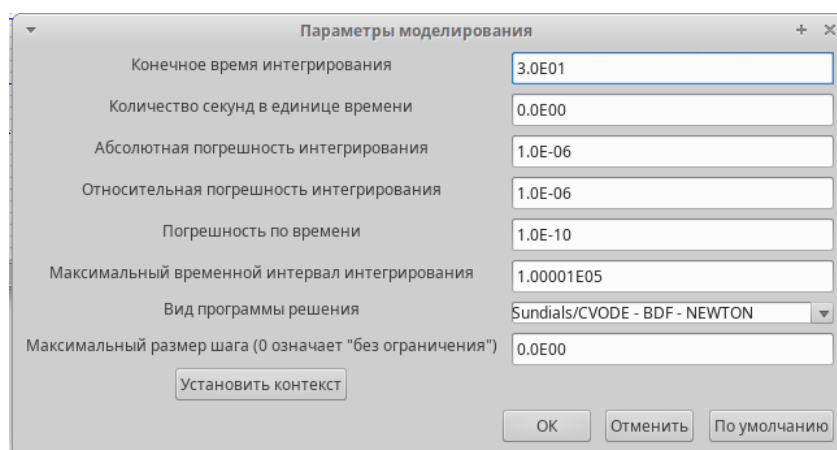


Рис. 3.5: Конечное время интегрирования

Результат моделирования представлен на (рис. 3.6), где черной линией обозначен график $s(t)$ (динамика численности уязвимых к болезни особей), красная линия $g(t)$ — динамику численности выздоровевших особей, наконец, зеленая линия $i(t)$ — динамику численности заражённых особей. Пересечение трёх линий определяет порог эпидемии.

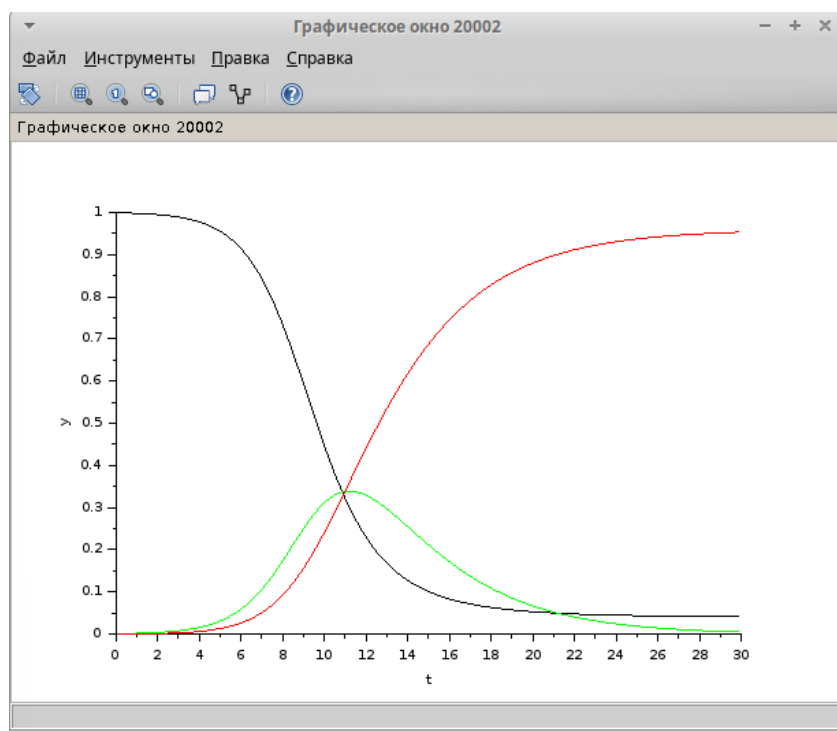


Рис. 3.6: Модель эпидемии при $\beta=1$, $\nu=0.3$

3.2 Реализация модели с помощью блока Modelica в xcoss

В данном задании необходимо было реализовать такую же модель эпидемии при $\beta=1$, $\nu=0.3$, только с помощью блока Modelica в xcoss. Для начала добавляем новый блок констант и блок реализации кода на Modelica. Таким образом выглядит наша модель (рис. 3.7)

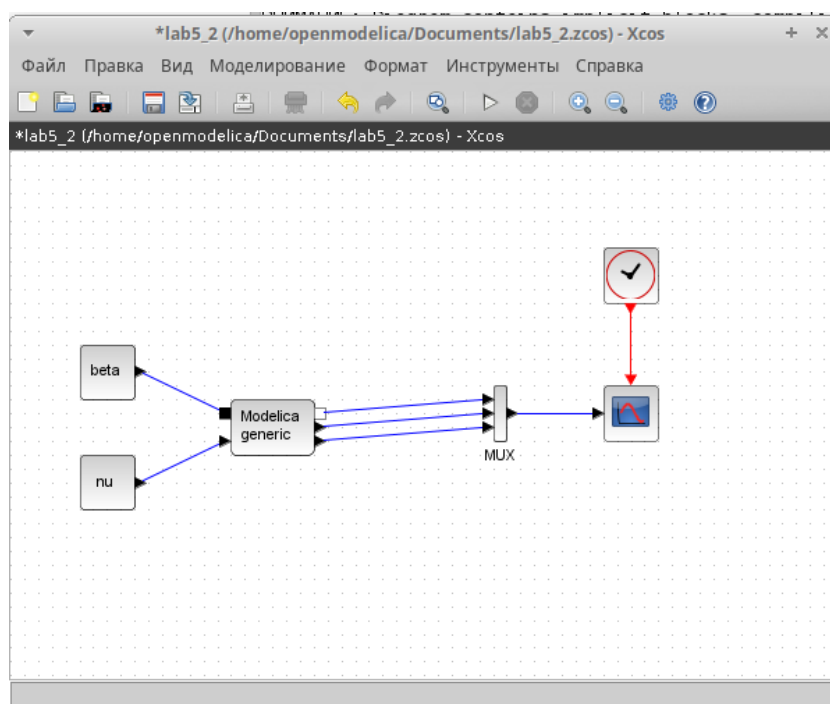


Рис. 3.7: Модель эпидемии

Указываем параметры для блока реализации. Переменные на входе (“beta”, “nu”) и выходе (“s”, “i”, “r”) блока заданы как внешние (“E”). (рис. 3.8)

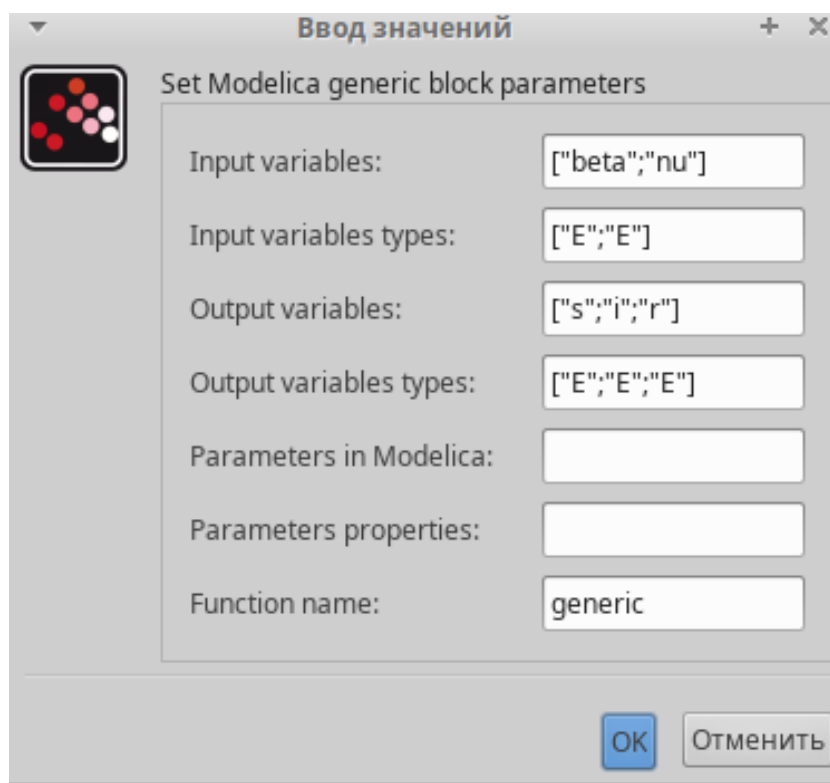


Рис. 3.8: Параметры блока реализации

Код на языке Modelica. Задаем переменные beta, nu. Указываем начальные значения для s, i, r и пишем систему уравнения. (рис. 3.9)

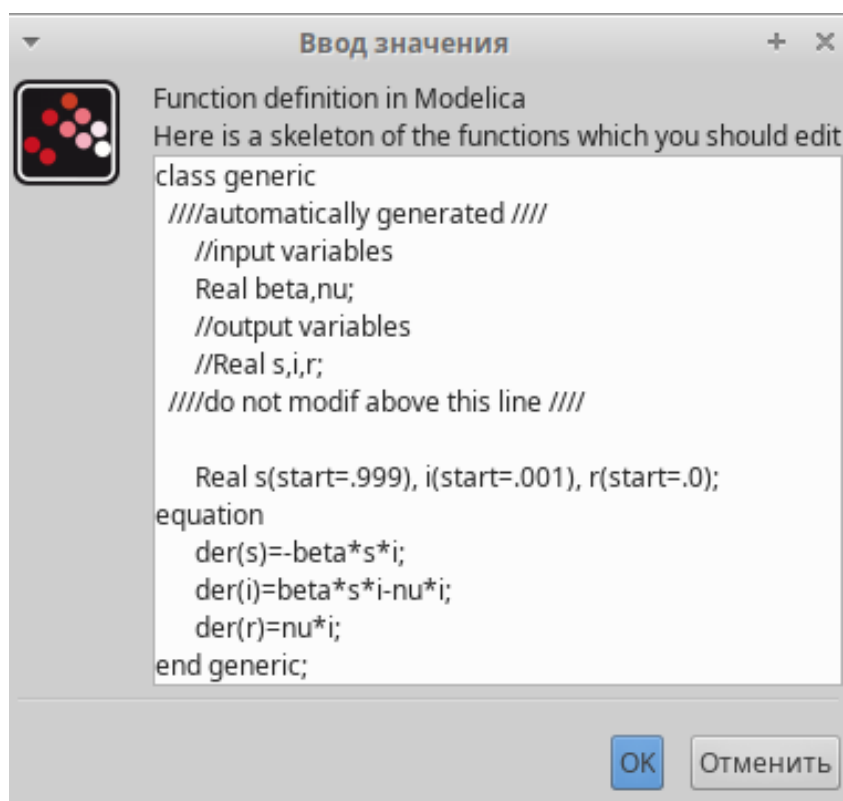


Рис. 3.9: Параметры блока реализации

Результат работы модели. Он идентичен с реализацией в xcos. (рис. 3.10)

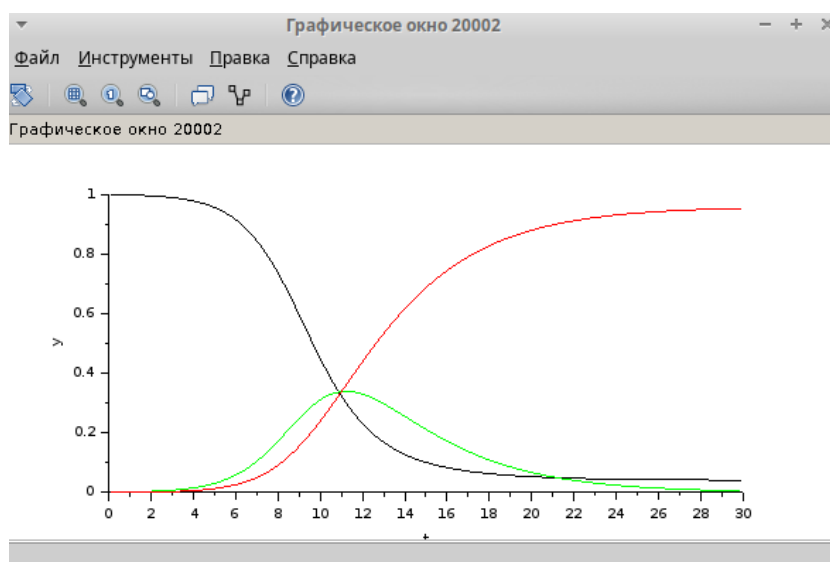
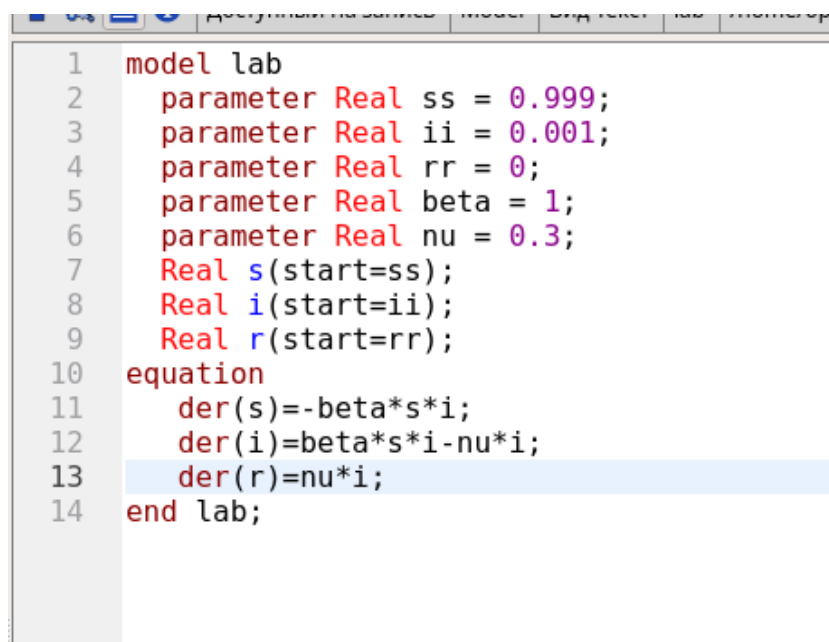


Рис. 3.10: Модель эпидемии Modelica

3.3 Выполнение упражнения 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. 3.11)



```
1 model lab
2   parameter Real ss = 0.999;
3   parameter Real ii = 0.001;
4   parameter Real rr = 0;
5   parameter Real beta = 1;
6   parameter Real nu = 0.3;
7   Real s(start=ss);
8   Real i(start=ii);
9   Real r(start=rr);
10  equation
11    der(s)=-beta*s*i;
12    der(i)=beta*s*i-nu*i;
13    der(r)=nu*i;
14  end lab;
```

Рис. 3.11: Реализация модели эпидемии в OpenModelica

Результат модели. Результат идентичен с построением с помощью других способов, значит все выполнено правильно. (рис. 3.12)

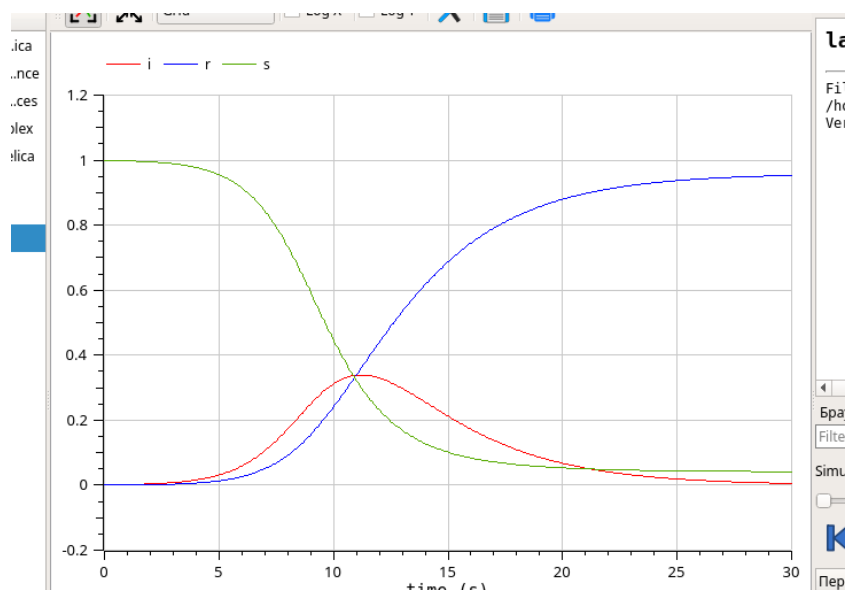


Рис. 3.12: Модель эпидемии в OpenModelica

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:

$$\begin{cases} \dot{s} = -\beta s(t)i(t) + \mu(N - s(t)); \\ \dot{i} = \beta s(t)i(t) - \nu i(t) - \mu i(t); \\ \dot{r} = \nu i(t) - \mu r(t), \end{cases}$$

где μ — константа, которая равна коэффициенту смертности и рождаемости.

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

В меню моделирования устанавливаем переменные окружения. (рис. 3.13)
Реализация с помощью xcos. (рис. 3.14)

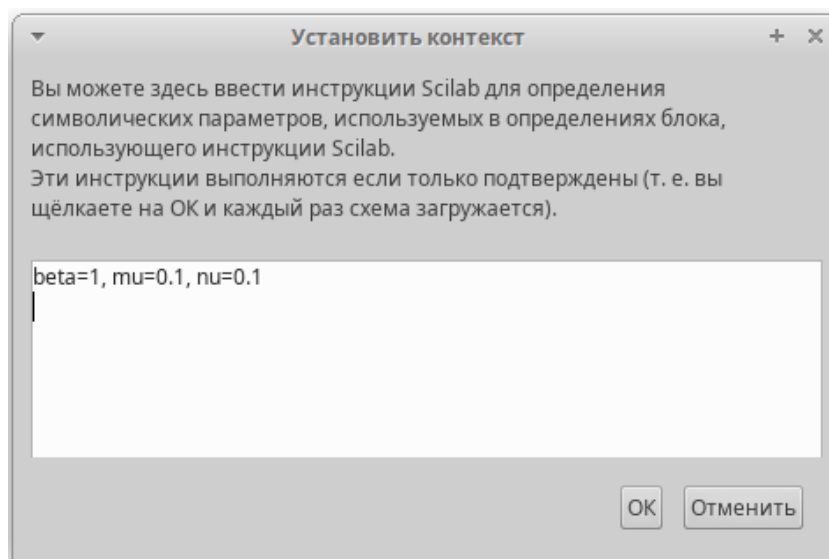


Рис. 3.13: Переменные окружения

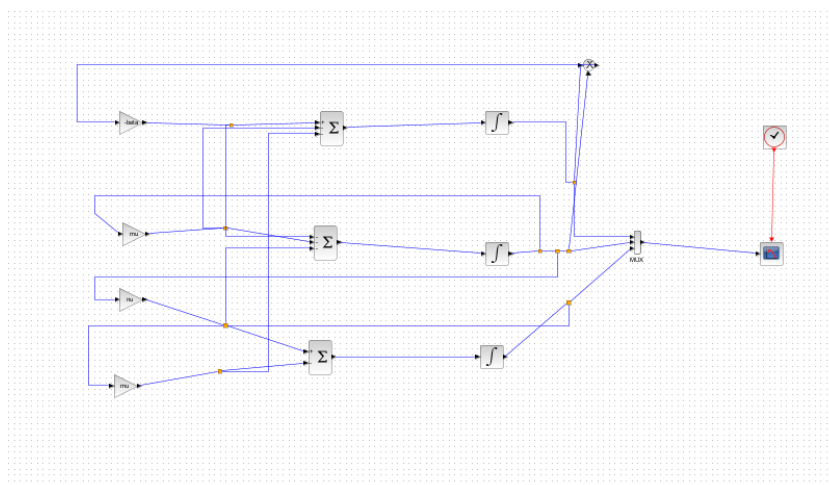


Рис. 3.14: Реализация модели эпидемии с учетом процесса рождения / гибели особей с помощью xcos

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

Результат моделирования представлен на (рис. 3.15), где черной линией обозначен график $s(t)$ (динамика численности уязвимых к болезни особей), красная линия $g(t)$ — динамику численности выздоровевших особей, наконец, зеленая линия $i(t)$ — динамику численности заражённых особей. Пересечение трёх линий

определяет порог эпидемии.

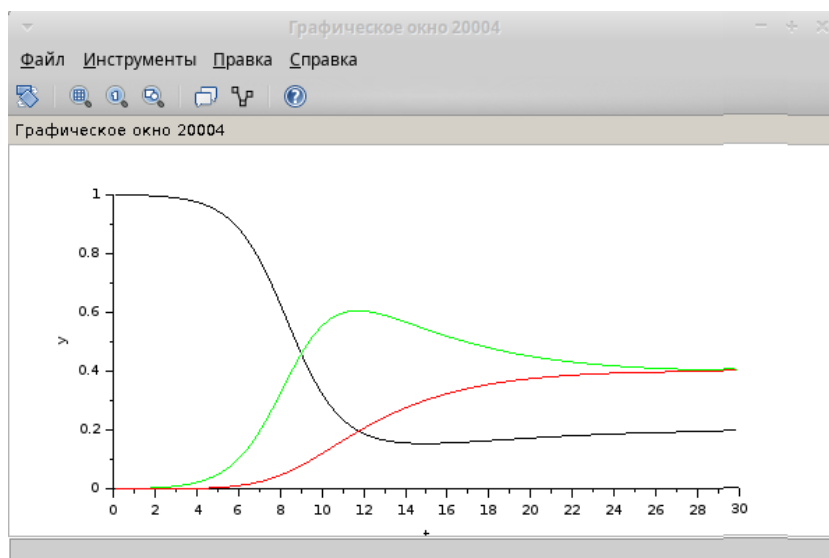


Рис. 3.15: Модель эпидемии при $\beta=1$, $\nu=0.1$, $\mu=0.1$

3.5 Задание для самостоятельного выполнения.

Реализация с помощью блока Modelica в xcos

Реализация с помощью блока Modelica в xcos. (рис. 3.16)

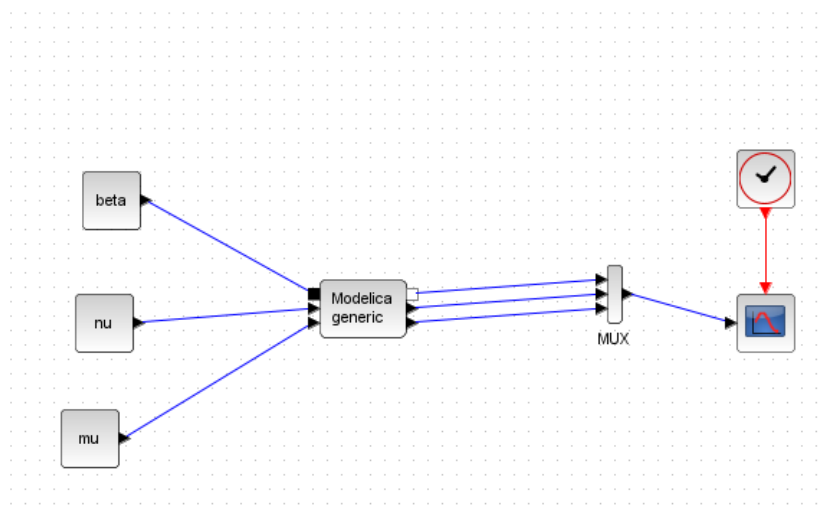


Рис. 3.16: Реализация модели эпидемии с учетом процесса рождения / гибели особей с помощью блока Modelica в xcos

Указываем параметры для блока реализации. Переменные на входе (“beta”, “nu”, “mu”) и выходе (“s”, “i”, “r”) блока заданы как внешние (“E”). (рис. 3.17)

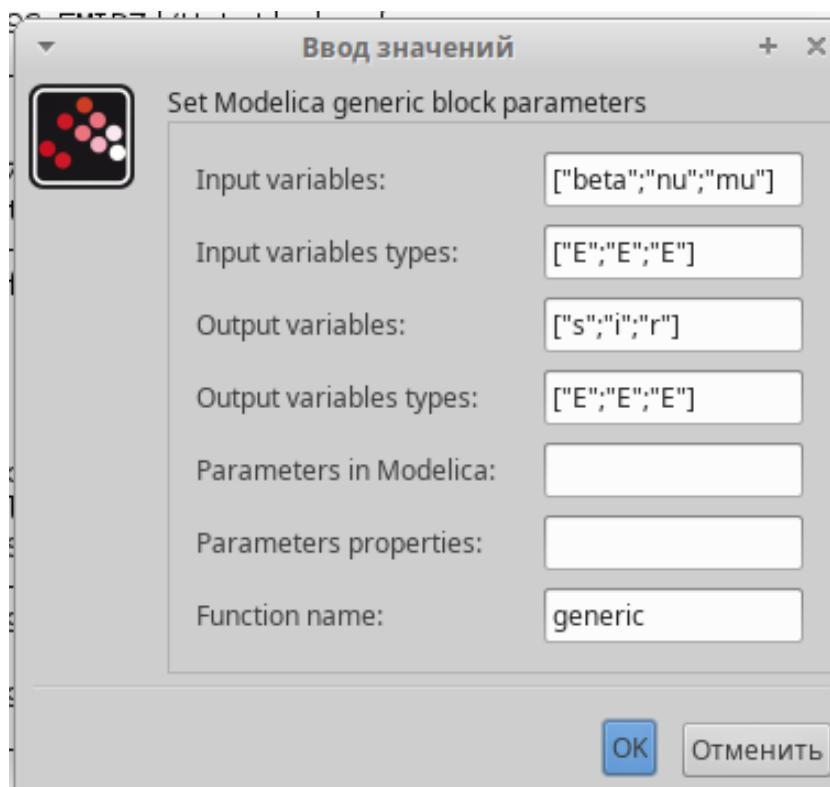


Рис. 3.17: Параметры блока реализации

Код на языке Modelica. Задаем переменные beta, nu, mu. Указываем начальные значения для s, i, r и пишем систему уравнения. (рис. 3.18)

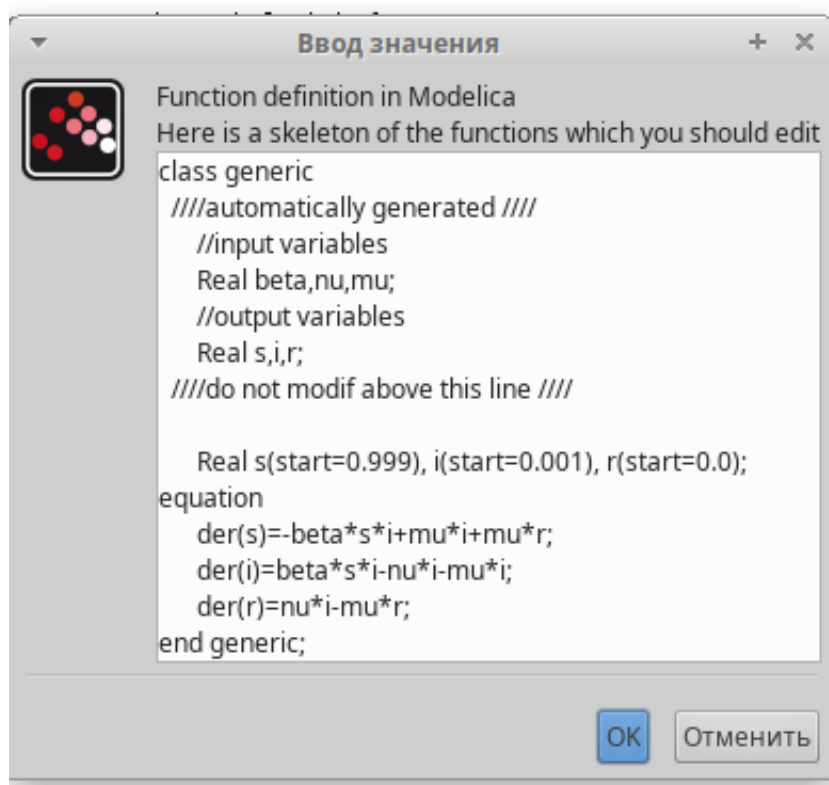


Рис. 3.18: Параметры блока реализации

Результат моделирования представлен на (рис. 3.19), 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.

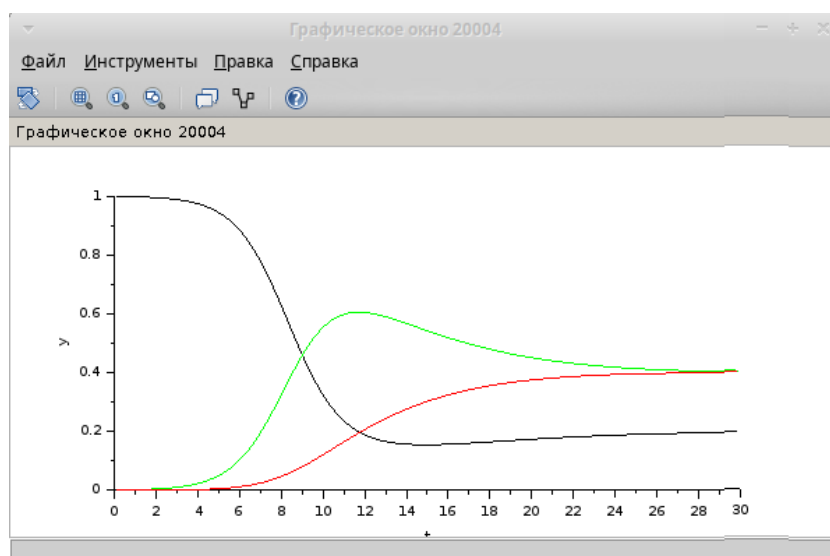


Рис. 3.19: Модель эпидемии при $\beta=1$, $\nu=0.1$, $\mu=0.1$

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. 3.20)

```

1  model lab
2    parameter Real ss = 0.999;
3    parameter Real ii = 0.001;
4    parameter Real rr = 0;
5    parameter Real beta = 1;
6    parameter Real nu = 0.1;
7    parameter Real mu = 0.1;
8    Real s(start=ss);
9    Real i(start=ii);
10   Real r(start=rr);
11   equation
12     der(s)=-beta*s*i+mu*i+mu*r;
13     der(i)=beta*s*i-nu*i-mu*i;
14     der(r)=nu*i-mu*r;
15   end lab;

```

Рис. 3.20: 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. 3.21)

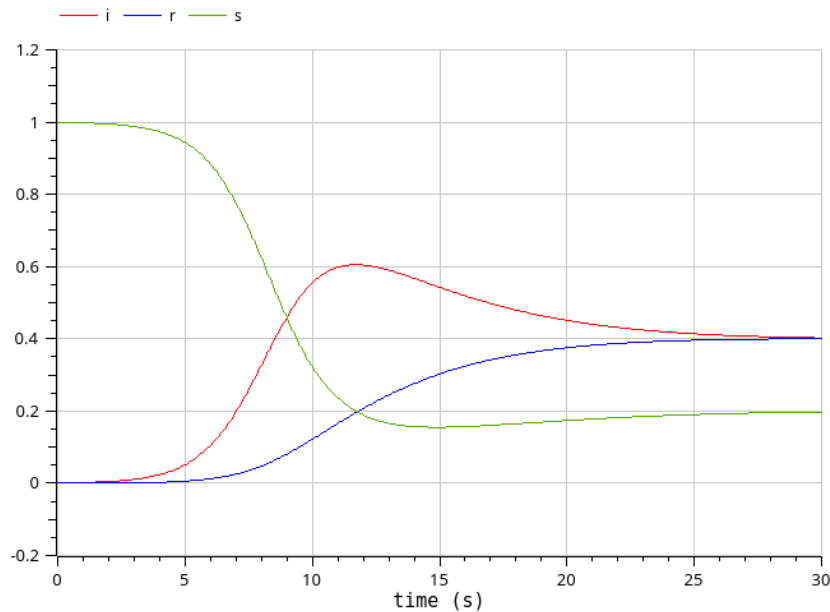


Рис. 3.21: Модель эпидемии с учетом процесса рождения / гибели особей в OpenModelica

3.7 Результаты на различных параметрах.

При $\mu=0.6$, $\nu=0.1$, $\beta=1$ (рис. 3.22), (рис. 3.23)

```

1  model lab
2    parameter Real ss = 0.999;
3    parameter Real ii = 0.001;
4    parameter Real rr = 0;
5    parameter Real beta = 1;
6    parameter Real nu = 0.1;
7    parameter Real mu = 0.6;
8    Real s(start=ss);
9    Real i(start=ii);
10   Real r(start=rr);
11   equation
12     der(s)=-beta*s*i+mu*i+mu*r;
13     der(i)=beta*s*i-nu*i-mu*i;
14     der(r)=nu*i-mu*r;
15   end lab;

```

Рис. 3.22: Результаты на различных параметрах.

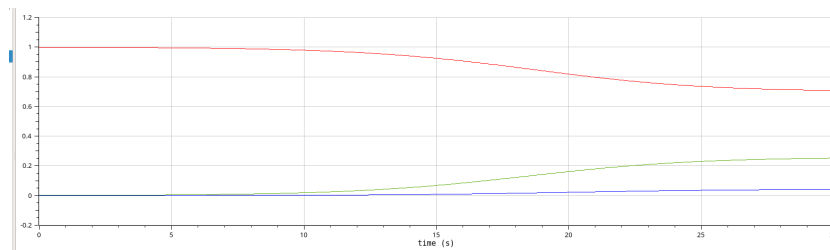


Рис. 3.23: Результаты на различных параметрах.

При $\mu=0.6$, $\nu=0.6$, $\beta=1$ (рис. 3.24), (рис. 3.25)

```

1  model lab
2    parameter Real ss = 0.999;
3    parameter Real ii = 0.001;
4    parameter Real rr = 0;
5    parameter Real beta = 1;
6    parameter Real nu = 0.6;
7    parameter Real mu = 0.6;
8    Real s(start=ss);
9    Real i(start=ii);
10   Real r(start=rr);
11   equation
12     der(s)=-beta*s*i+mu*i+mu*r;
13     der(i)=beta*s*i-nu*i-mu*i;
14     der(r)=nu*i-mu*r;
15   end lab;

```

Рис. 3.24: Результаты на различных параметрах.

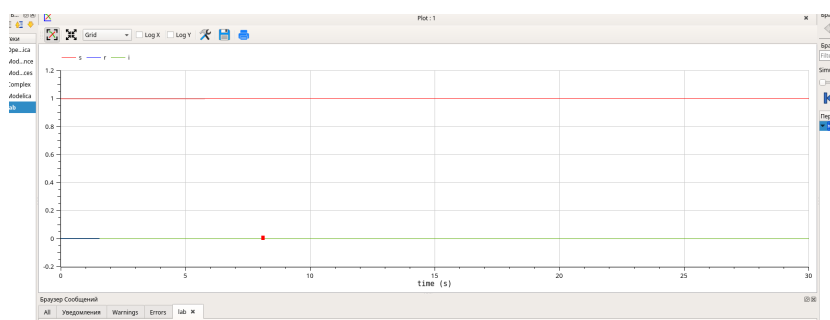


Рис. 3.25: Результаты на различных параметрах.

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. 4.1).

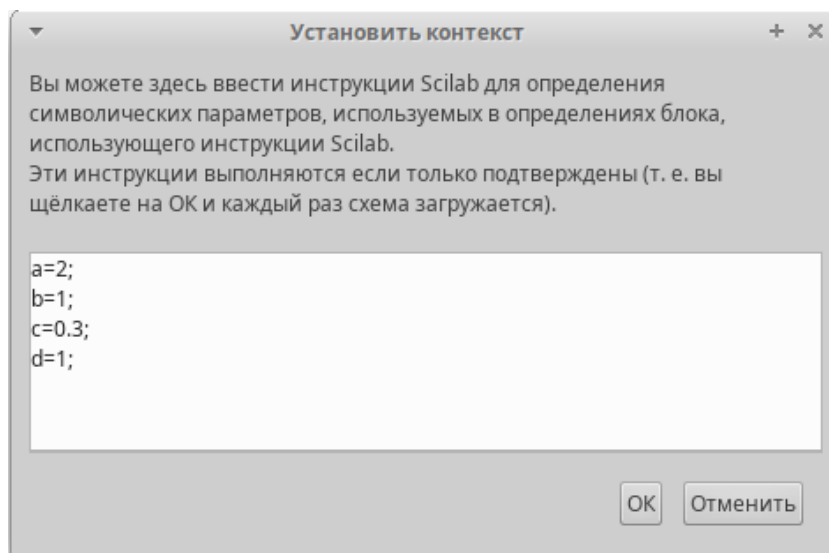


Рис. 4.1: Константы

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. 6.2).

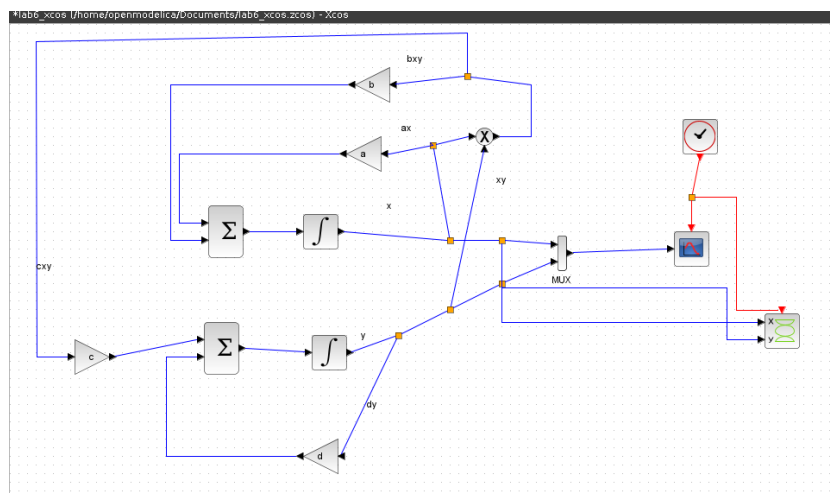


Рис. 4.2: Реализация модели

Задаем начальные значения для x и y в параметрах блоков интегрирования.
(рис. 4.3), (рис. 4.4)

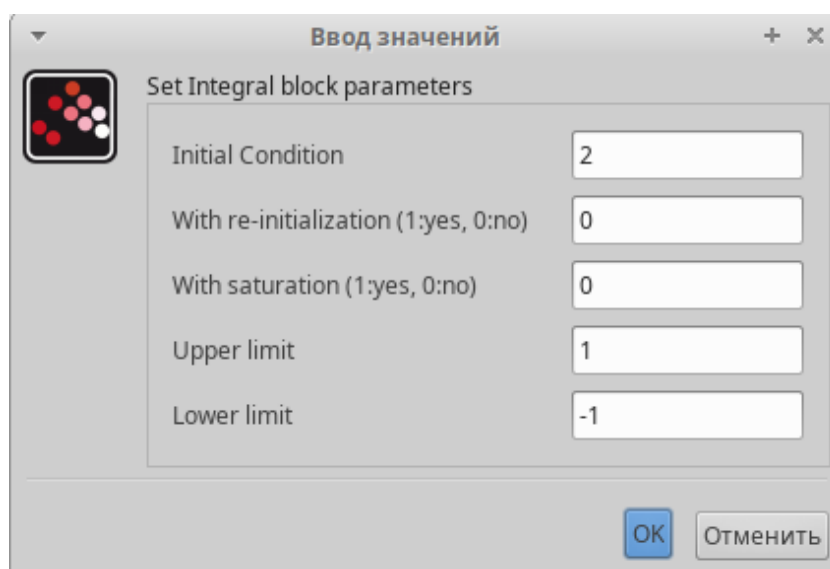


Рис. 4.3: Начальные значения

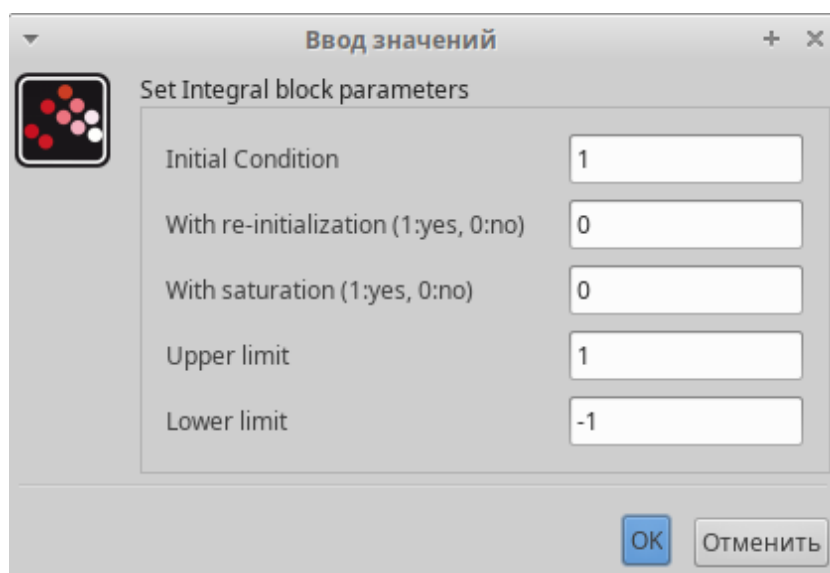


Рис. 4.4: Начальные значения

Устанавливаем конечное время интегрирования 30. (рис. 4.5)

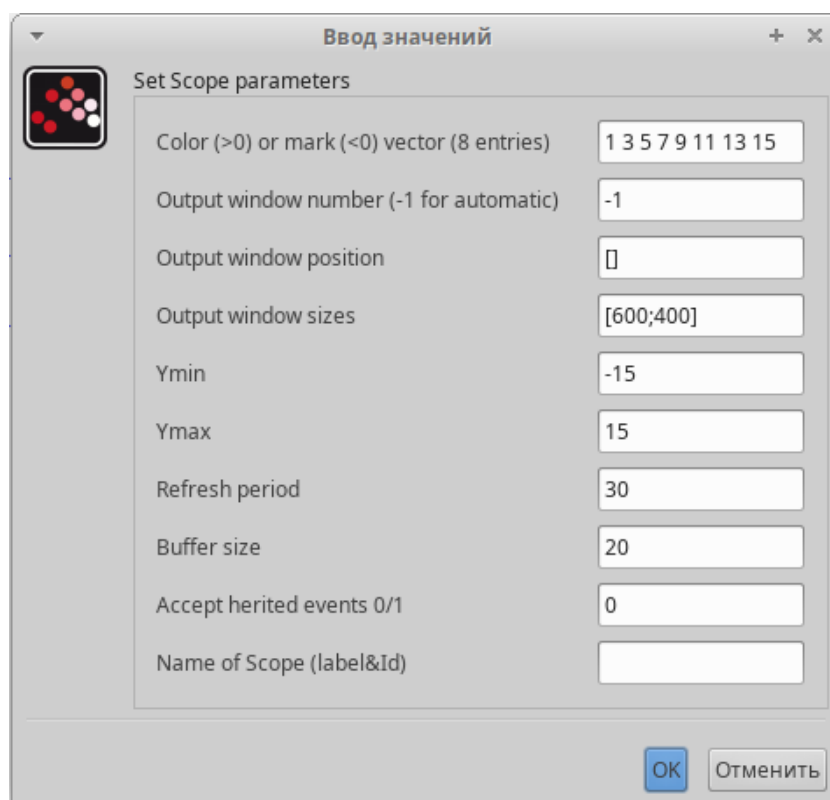


Рис. 4.5: конечное время интегрирования

Фазовый портрет. (рис. 4.6)

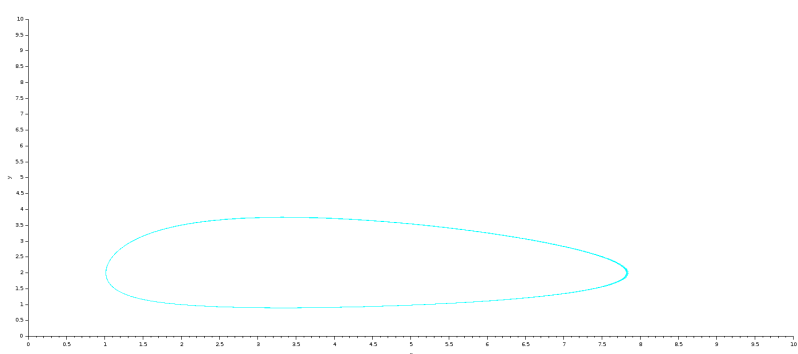


Рис. 4.6: Фазовый портрет.

Динамика изменения численности хищников и жертв. Черной линией обозначена динамика численности жертв. Зеленой линией обозначена динамика численности хищников. (рис. 4.7)

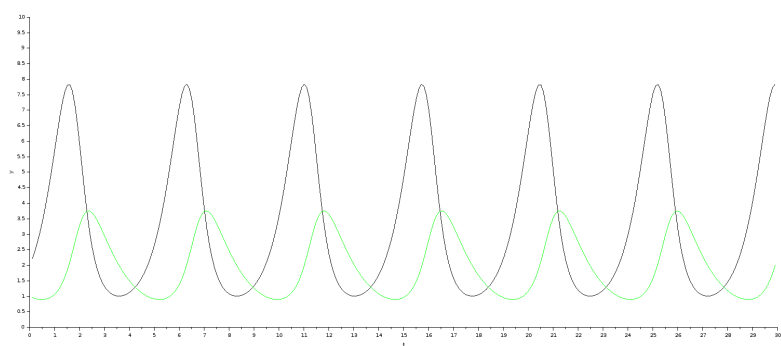


Рис. 4.7: Динамика изменения численности хищников и жертв

4.2 Реализация модели с помощью блока Modelica в xcoss

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. 4.8)

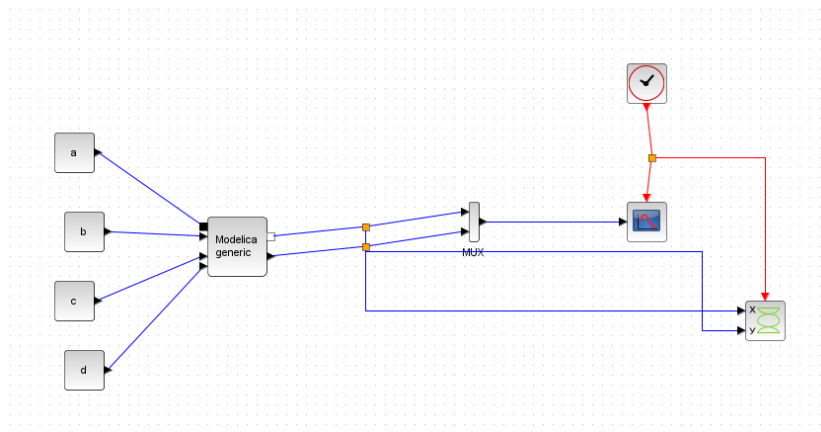


Рис. 4.8: Реализация модели

Параметры блока моделирования и программный код (рис. 4.9), (рис. 4.10)

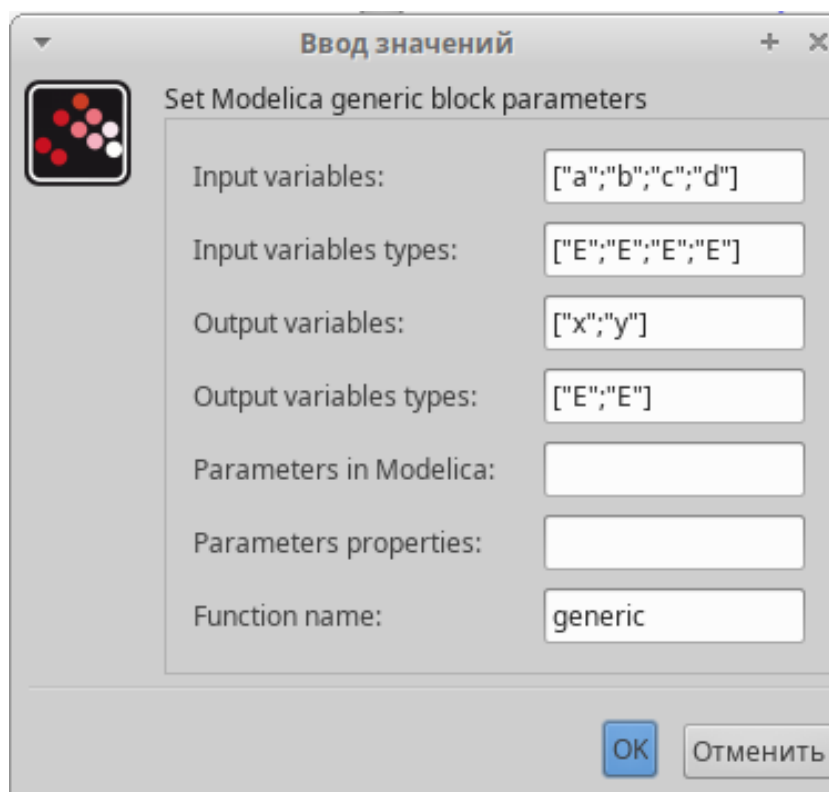


Рис. 4.9: Параметры блока моделирования

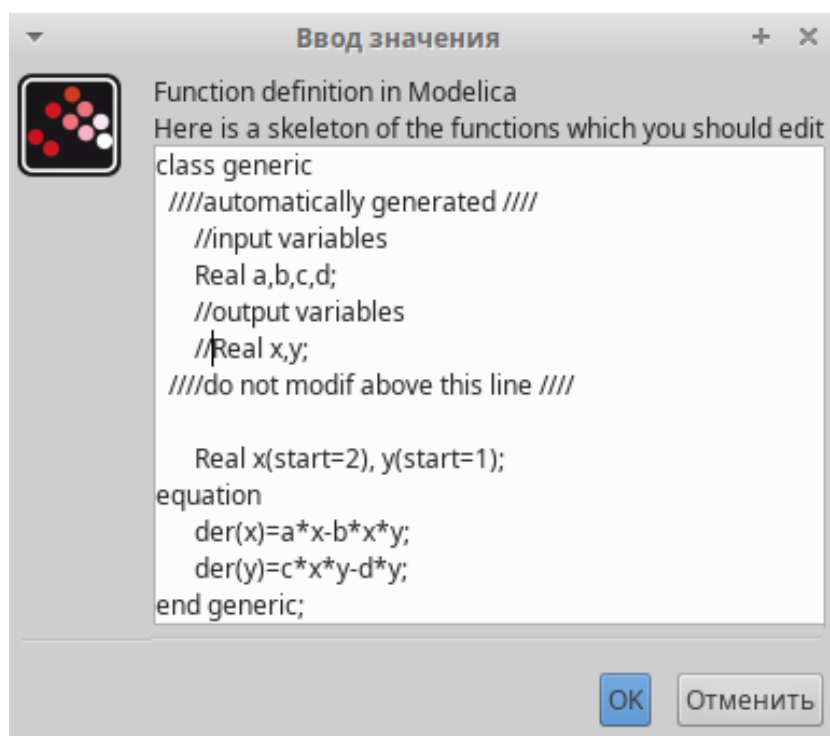


Рис. 4.10: Параметры блока моделирования

Фазовый портрет и график изменения численности популяций. Результат полностью идентичен с xcos. (рис. 4.11), (рис. 4.12)

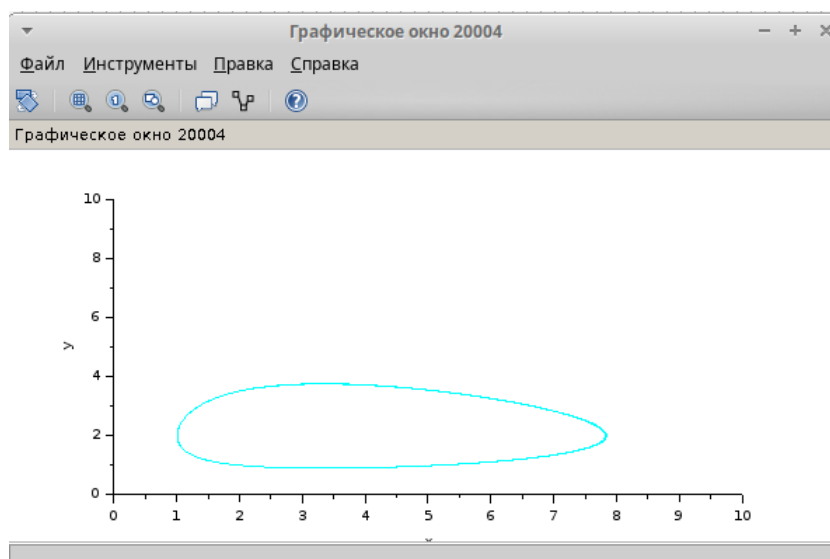


Рис. 4.11: Фазовый портрет

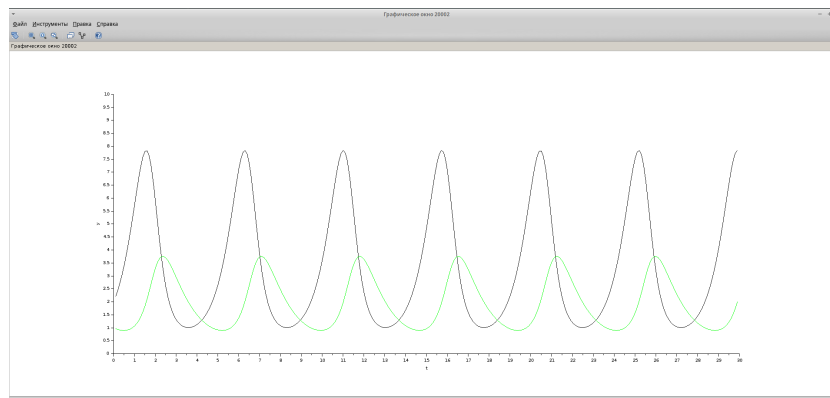


Рис. 4.12: график изменения численности популяций

4.3 Реализация модели в OpenModelica.

Код для реализации данной модели. Задаем начальные коэффициенты и пишем уравнения модели. Задаем конечное время интегрирования. (рис. 4.13)


```

1  model lab6
2
3      parameter Real a = 2;
4      parameter Real b = 1;
5      parameter Real c = 0.3;
6      parameter Real d = 1;
7
8      parameter Real x0 = 2;
9      parameter Real y0 = 1;
10
11     Real x(start=x0);
12     Real y(start=y0);
13
14     equation
15
16         der(x) = a*x - b*x*y;
17         der(y) = c*x*y - d*y;
18
19     end lab6;

```

Рис. 4.13: Реализация модели

Фазовый портрет и график изменения численности популяций. Результат полностью идентичен с предыдущими реализациями. (рис. 4.14), (рис. 4.15)

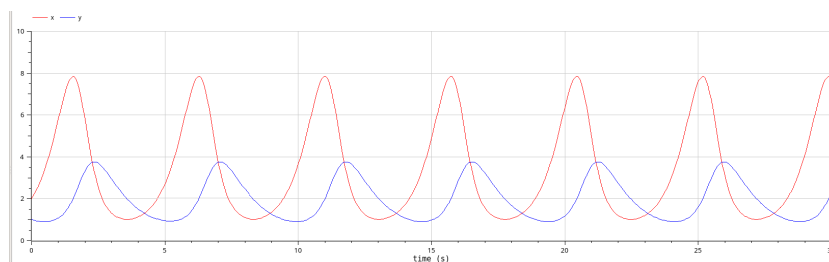


Рис. 4.14: Фазовый портрет

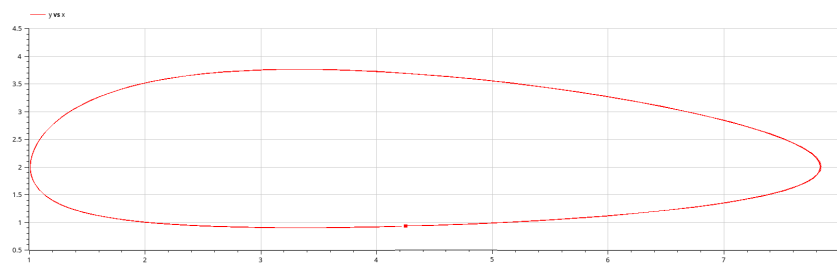


Рис. 4.15: график изменения численности популяций

5 Выполнение лабораторной работы 7

Реализация модели системы массового обслуживания типа $M|M|1|\infty$. Для начала необходимо указать начальные параметры. (рис. 5.1).

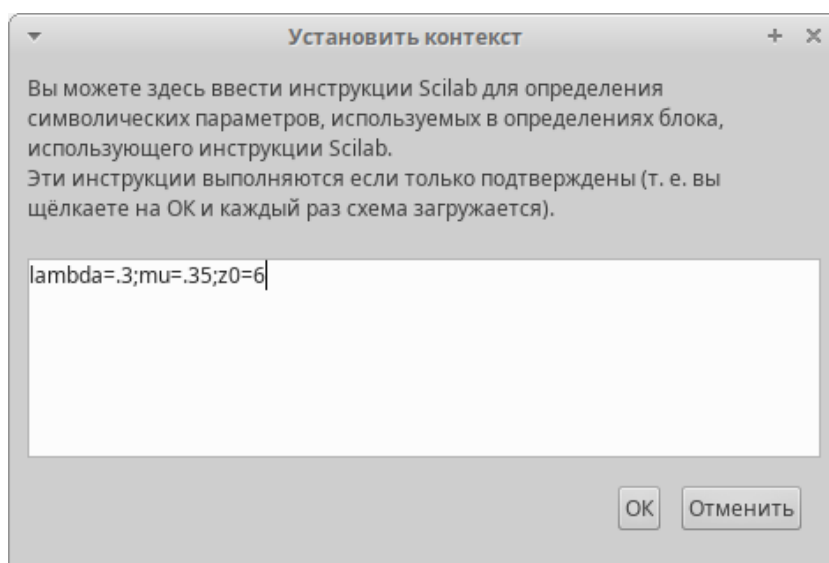


Рис. 5.1: Установка контекста моделирования

Построение суперблока отвечающего за поступление заявок. (рис. 5.2)

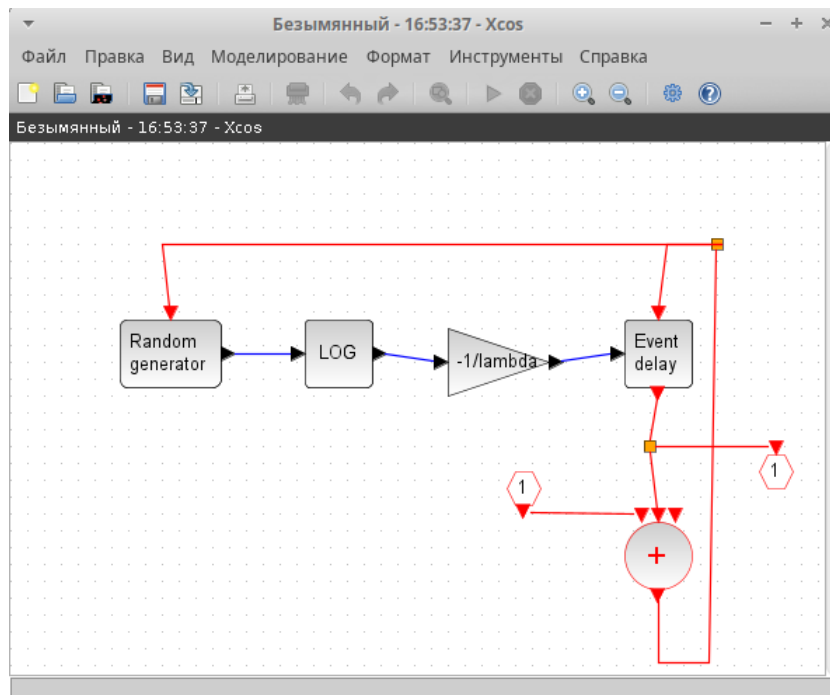


Рис. 5.2: Суперблок, моделирующий поступление заявок

Построение суперблока отвечающего за обработку заявок. (рис. 5.3)

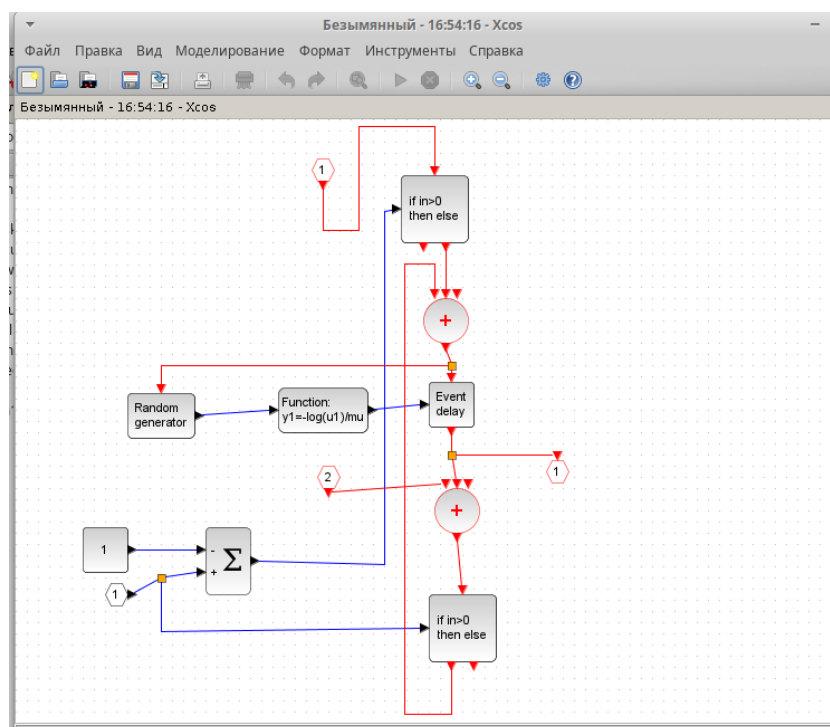


Рис. 5.3: Суперблок, моделирующий обработку заявок

Модель $M|M|1|\infty$ (рис. 5.4)

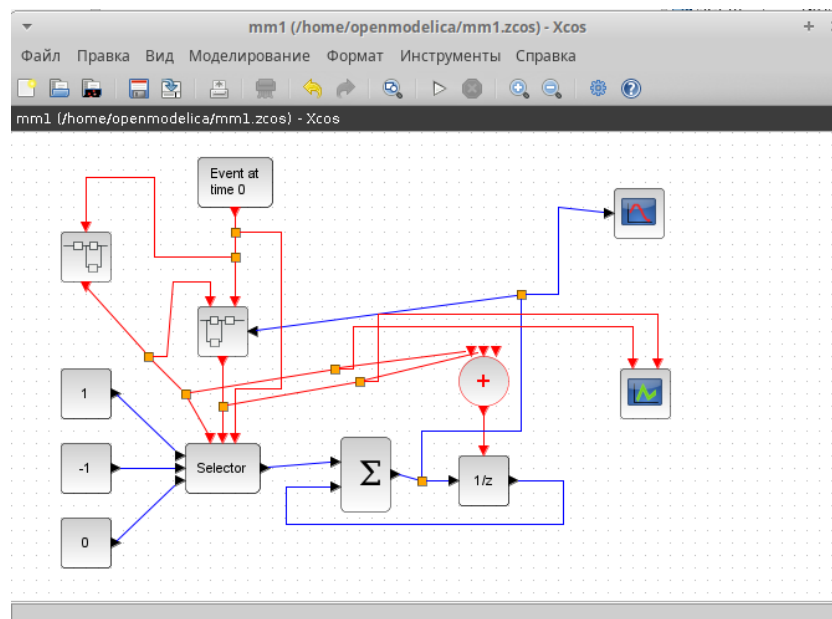


Рис. 5.4: $M|M|1|\infty$

График поступления и обработки заявок (рис. 5.5)

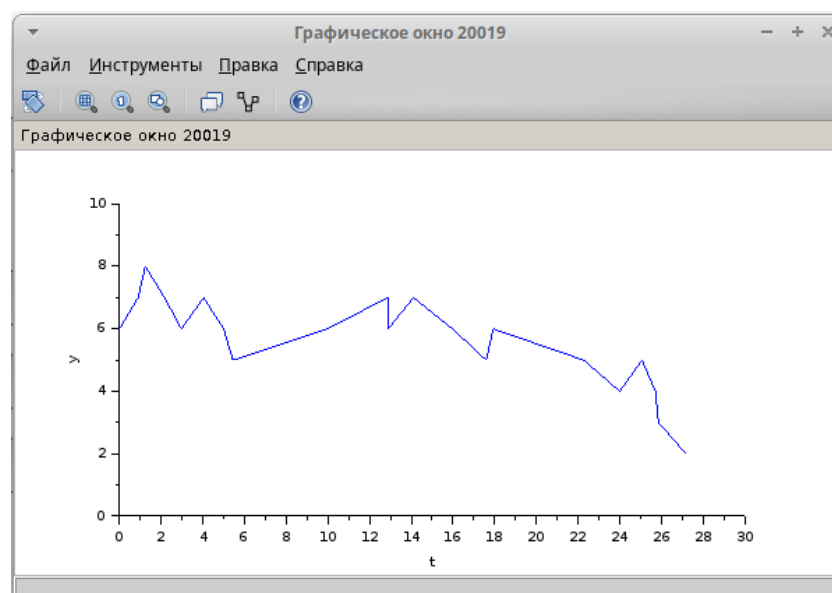


Рис. 5.5: График поступления и обработки заявок

График динамики размера очереди (рис. 5.6)

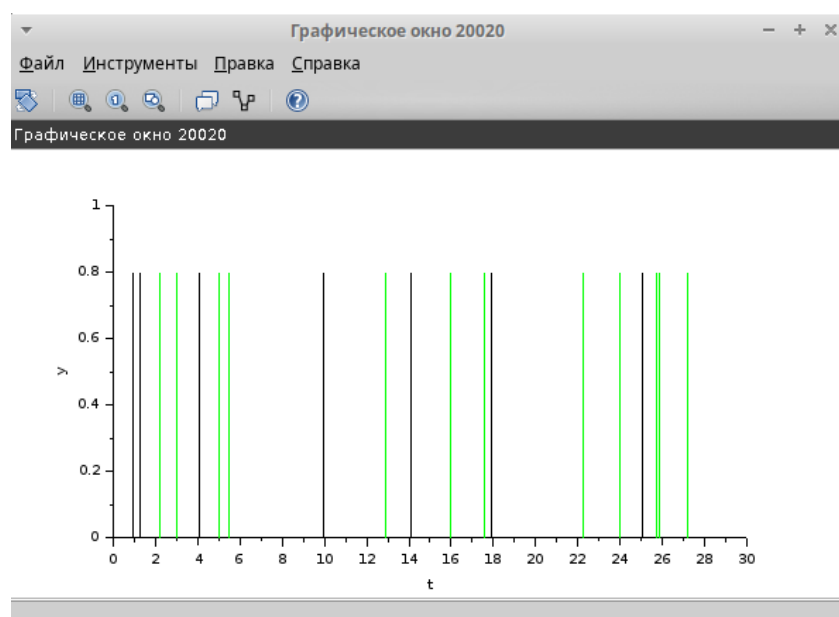


Рис. 5.6: График динамики размера очереди

6 Выполнение лабораторной работы 8

6.1 Реализация в xcos

Построим схему xcos, моделирующую нашу систему, с начальными значениями параметров $N = 1, R = 1, K = 5.3, C = 1, W(0) = 0.1, Q(0) = 1$. Для этого сначала зададим переменные окружения (рис. 6.1).

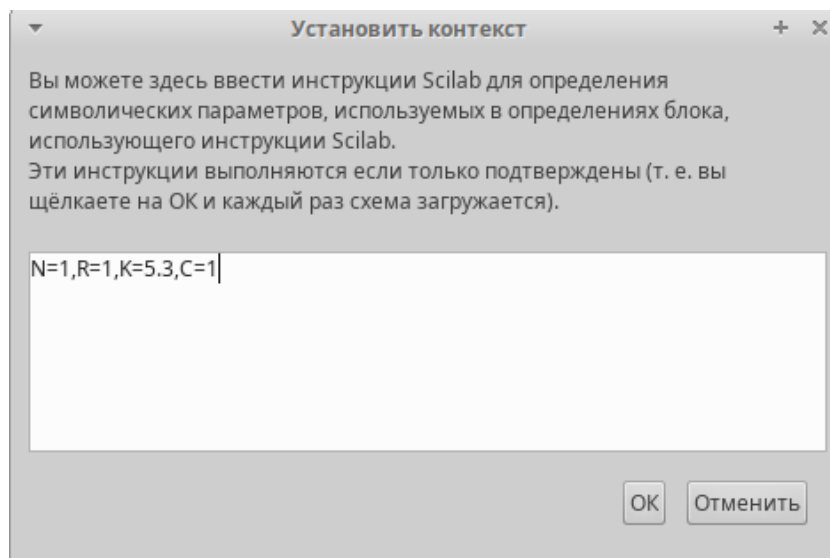


Рис. 6.1: Установка контекста

Затем реализуем модель TCP/AQM, разместив блоки интегрирования, суммирования, произведения, констант, а также регистрирующие устройства (рис. 6.2):

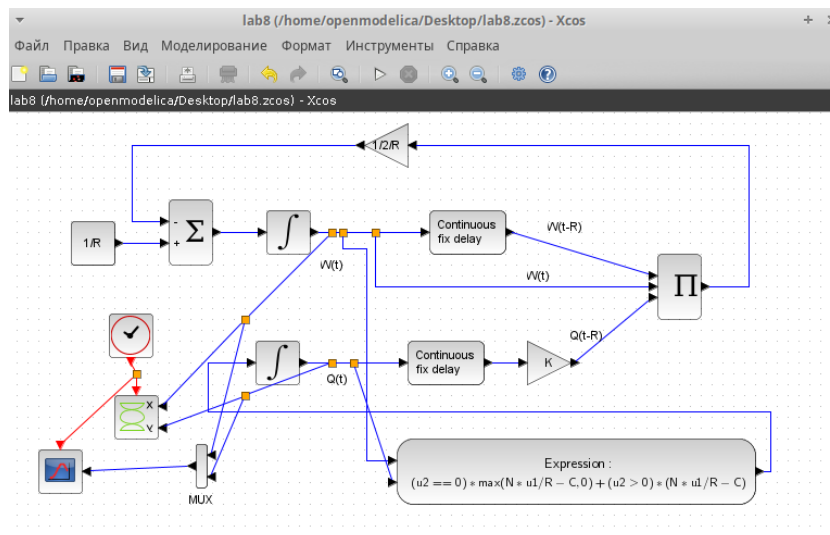


Рис. 6.2: Модель TCP/AQM в xcos

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. 6.3, 6.4):

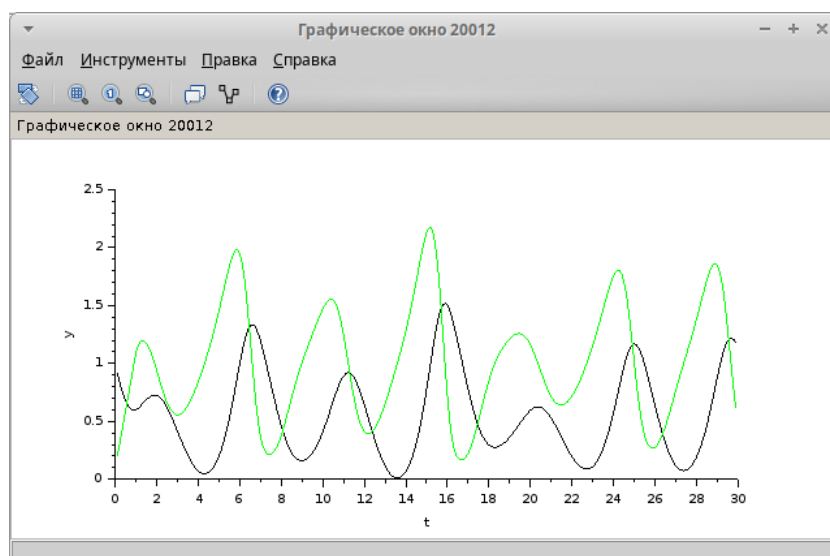


Рис. 6.3: Динамика изменения размера TCP окна $W(t)$ и размера очереди $Q(t)$

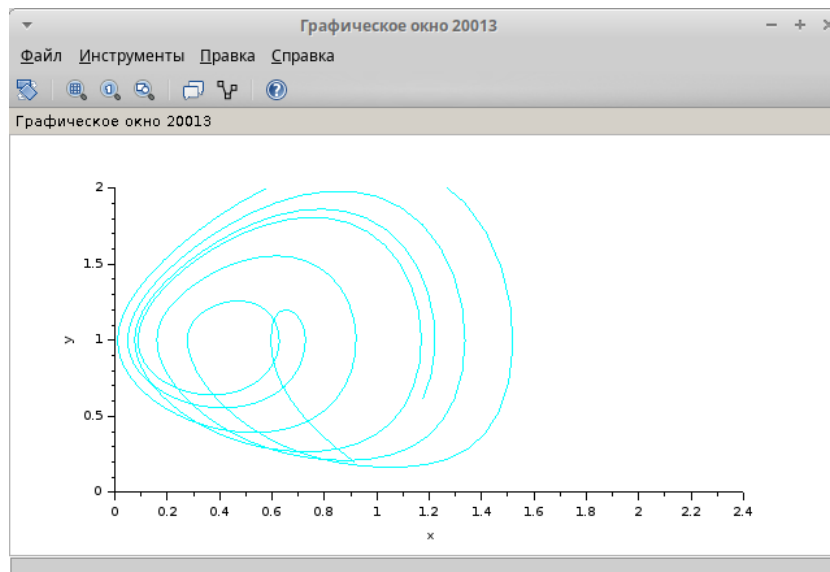


Рис. 6.4: Фазовый портрет (W, Q)

Уменьшив скорость обработки пакетов C до 0.9 увидим, что автоколебания стали более выраженными (рис. 6.5, 6.6).

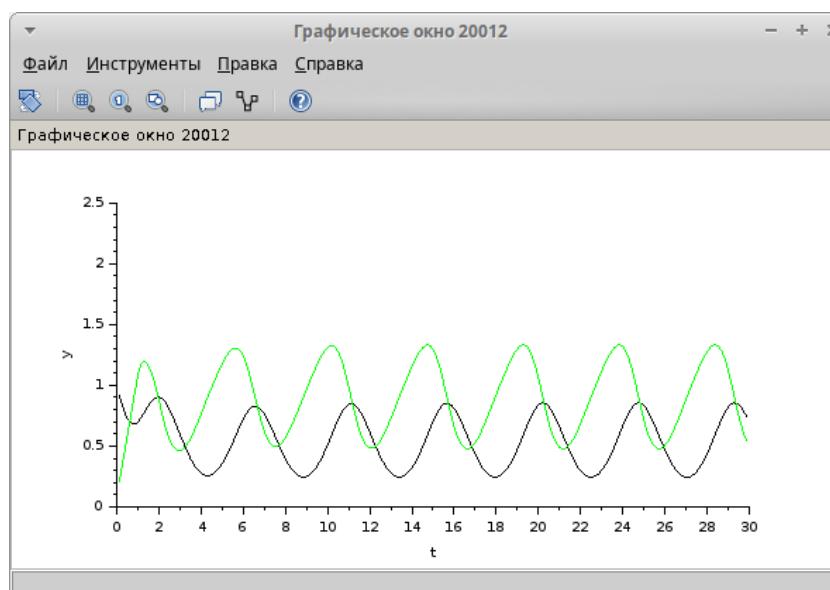


Рис. 6.5: Динамика изменения размера TCP окна $W(t)$ и размера очереди $Q(t)$ при $C = 0.9$

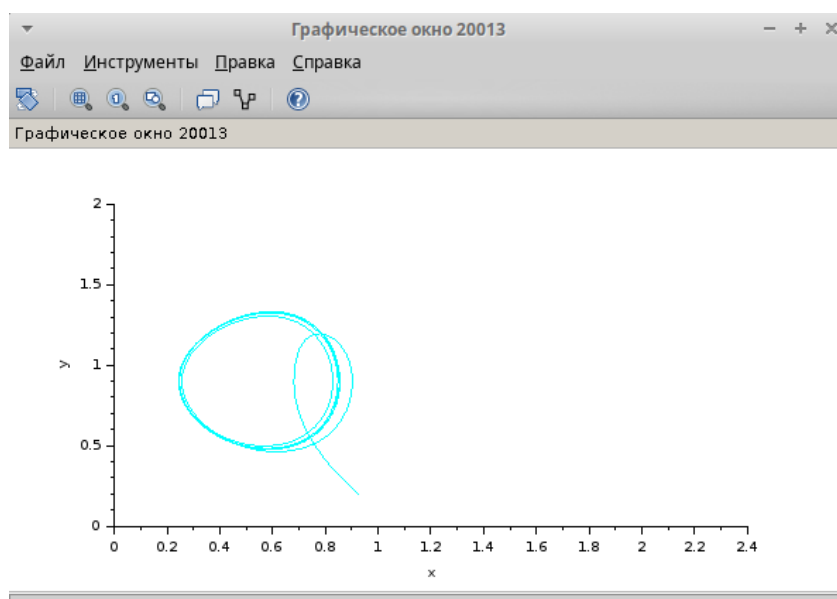


Рис. 6.6: Фазовый портрет (W, Q) при $C = 0.9$

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. 6.7, 6.8).

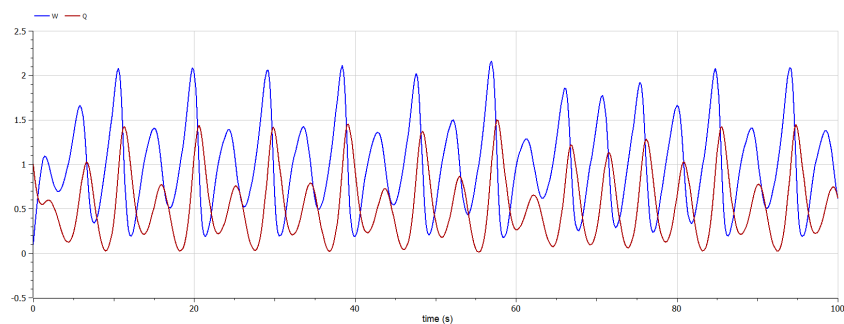


Рис. 6.7: Динамика изменения размера TCP окна $W(t)$ и размера очереди $Q(t)$. OpenModelica

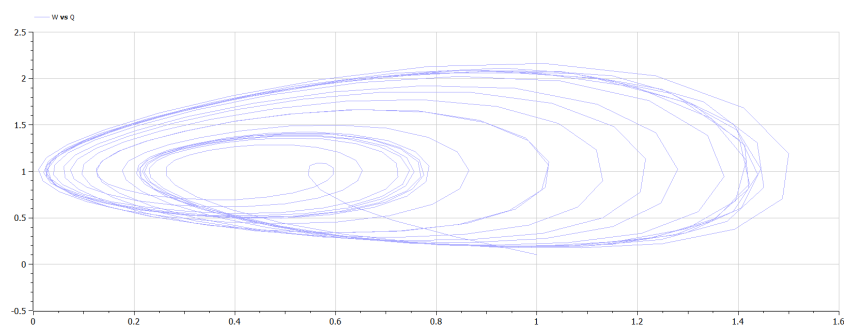


Рис. 6.8: Фазовый портрет (W, Q) . OpenModelica

7 Выводы

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