ТЕХНИЧЕСКОЕ ЗАДАНИЕ 1

Дана однокамерная фармококинетическая модель с всасыванием (Рис. 1):

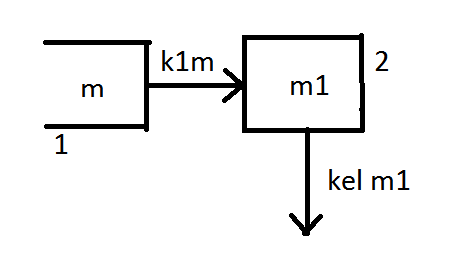


Рис. 1

где

1 – место введение лекарственного препарата

2 – камера.

Камера представляет собой ограниченноый объем жидкости (ткани), неизменяемый с течением времени. Дан определенный объем лекарственного препарата, который всасывается в камеру пропорционально своей массе в соответствии с уравнением:

(1)

где

m – масса лекарственного препарата в месте введения 1,

– константа скорости поступления препарата в камеру (константа скорости всасывания).

Пусть

Преполагается, что масса лекарственного препарата в 1 в начальный момент времени равна M=30мг, причем в самой камере в начальный момент времени препарата нет. Тогда масса лекарственного препарата в камере изменяется в соответствии с следующим уравнением:

(2)

где

– масса лекарственного препарата в камере,

– константа элиминации (выведения) лекарственного препарата из камеры.

Пусть .

Как только масса лекарственного препарата в месте введения становится меньше порогового значения, засекается отрезок времени , по истечении которого в месте введения уничтожаются все остатки прежней дозы препарата и вводится новая доза m=M.

Существует более сложная система, называемая двухкамерной фармококинетической моделью с всасыванием (рис. 2):

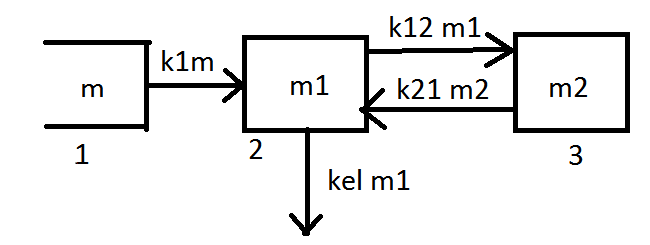


Рис. 2

В ней лекарственный препарат вводится из аналогичного ранее описанному места введения 1 и выводится из камеры 2, но существует еще камера 3, подсоединенная к камере 2. Между камерами 2 и 3 может циркулировать лекарственный препарат.

В этой модели масса лекарственного препарата вместе введения описывается также уравнением (1) с теми же начальными условиями. Массы же в камерах 2 и 3 описываются уравнениями (3) и (4) соответственно:

(3*)*

(4)

где

– масса лекарственного препарата в камере 3,

– константа скорости поступления препарата из камеры 2 в камеру 3.

– константа выведения лекарственного препарата из камеры 3 в камеру 2.

Пусть , .

Как и в первой системе, в начальный момент времени в камерах 2 и 3 лекарственный препарат отсутствует. Как и в первой системе, как только масса лекарственного препарта в месте введения становится меньше порогового значения е, засекается отрезок времени Time, по истечении которого в месте введения уничтожаются все остатки прежней дозы препарата и вводится новая доза m=M.

Построить модель системы, содержащей обе фармококинетические модели, несвязанные друг с другом с различными местами введения лекарственного препарата. Также построить модель системы, состоящей из обеих фармакокинетических моделей, имеющих общее место введения препарата. Скорость всасывания лекарственного препарата в обеих моделях одинаковая, отрезок времени Time и пороговое значение e для места введения препарата совпадают со значениями, принятыми для однокамерной фармококинетической модели с всасыванием.

СОЗДАНИЕ МОДЕЛИ

Для того чтобы работать с пакетом Simulink необходимо запустить систему MATLAB: Пуск⟶Программы⟶MATLAB, в результате запуститься программа MATLAB (рис. 3).

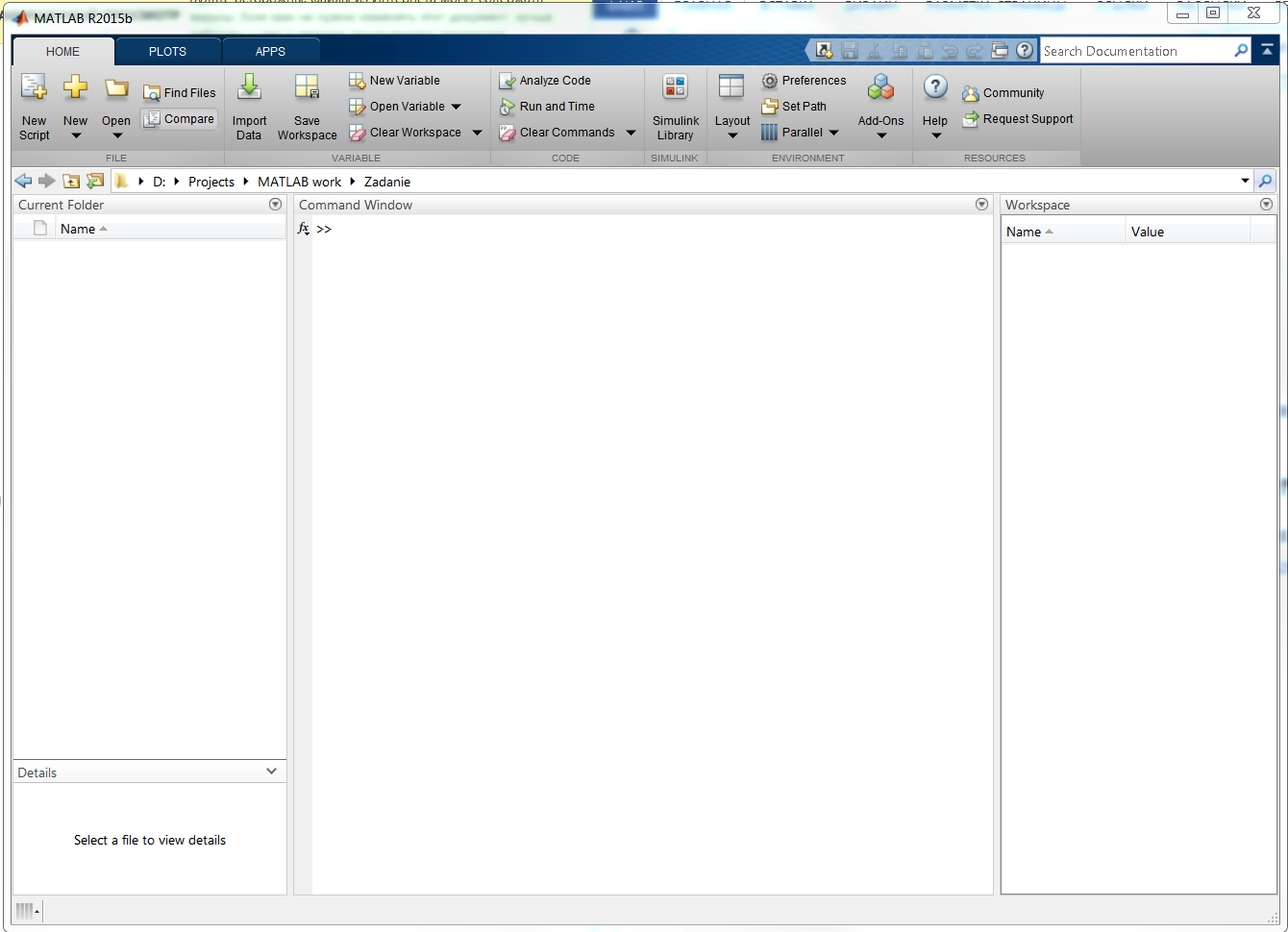


Рис. 3

Для запуска Simulink нужно в Comand Window написать >>Simulink. Откроется библиотека элементов Simulink Library Browser. Для построения модели необходимо создать окно модели нажав на иконку New Model (Рис. 4)

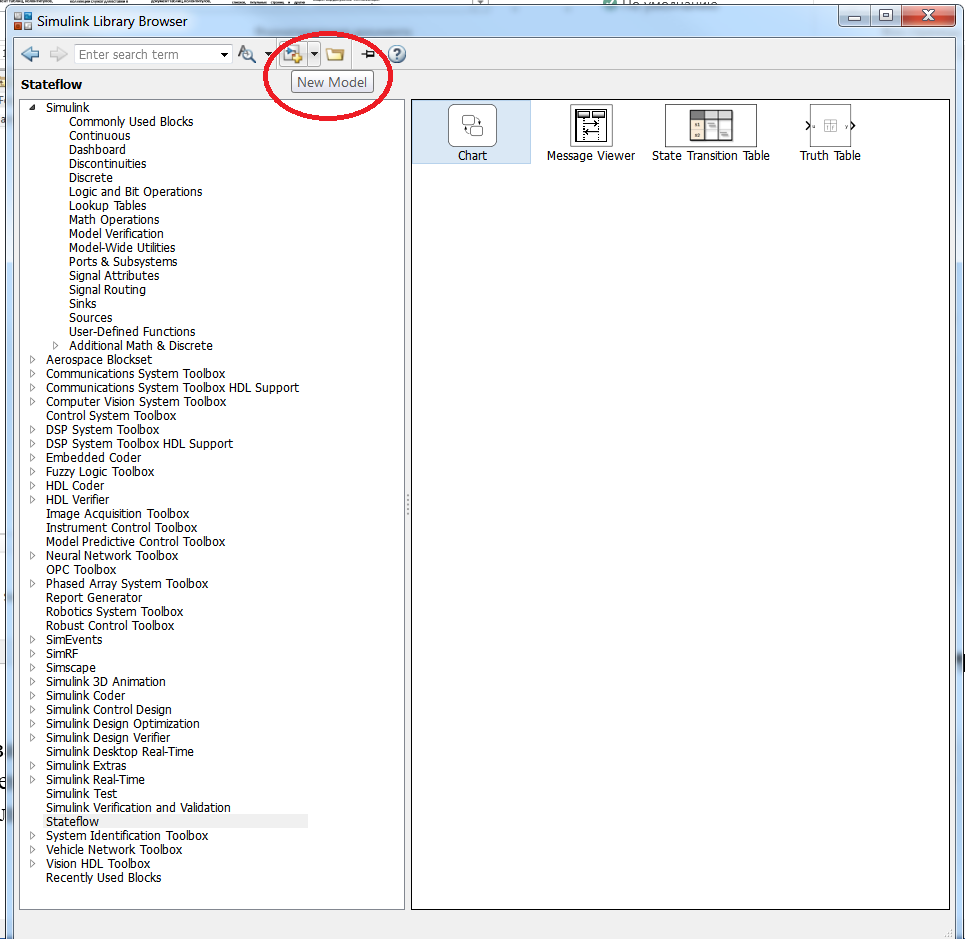


Рис. 4

В открывшемся окне модели (рис. 5) нажимаем на иконку Save и сохраняем модель с именем farmokinetic.slx

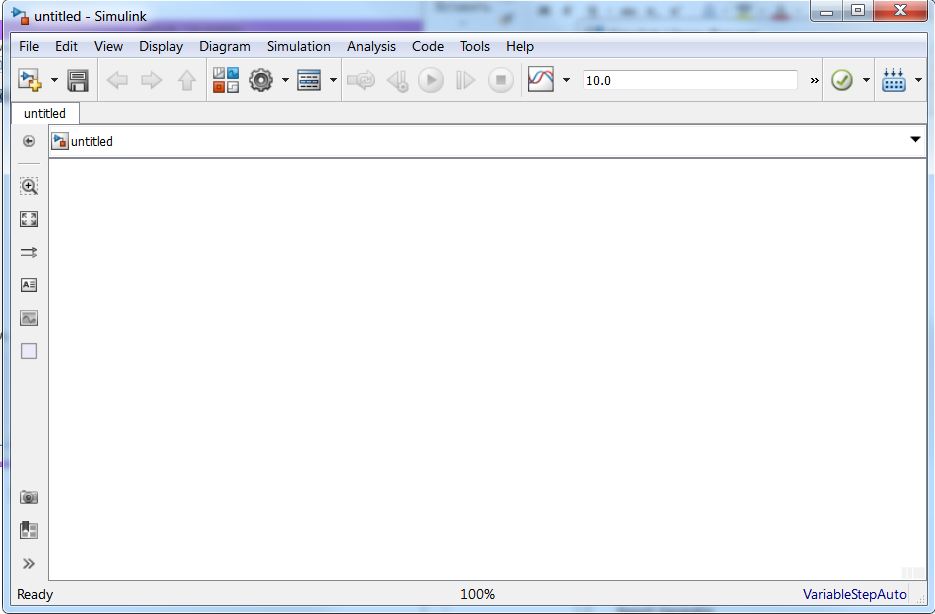


Рис. 5

Для работы модели необходимо задать параметры моделирования. Это можно сделать через Comand Window в MATLAB (рис. 6)

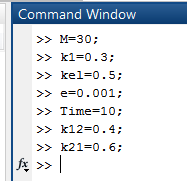


Рис. 6

Далее находим в Simulink Library Browser в разделе Simulink-Ports & Subsystems блок Subsystem. Переносим этот блок в окно модели. Изменяем его название на “1-kamernaya model”(рис. 7)

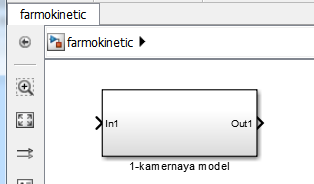


Рис. 7

Открываем блок, удаляем входной порт In1, создаем еще 2 копии порта Out1 и переименовываем их как на рис. 8

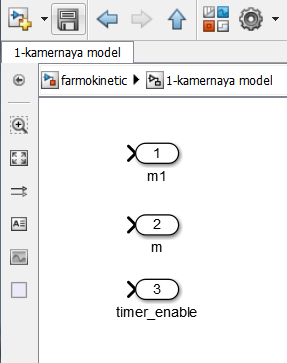


Рис. 8

Далее добавляем в подсистему нужные блоки (рис. 9)

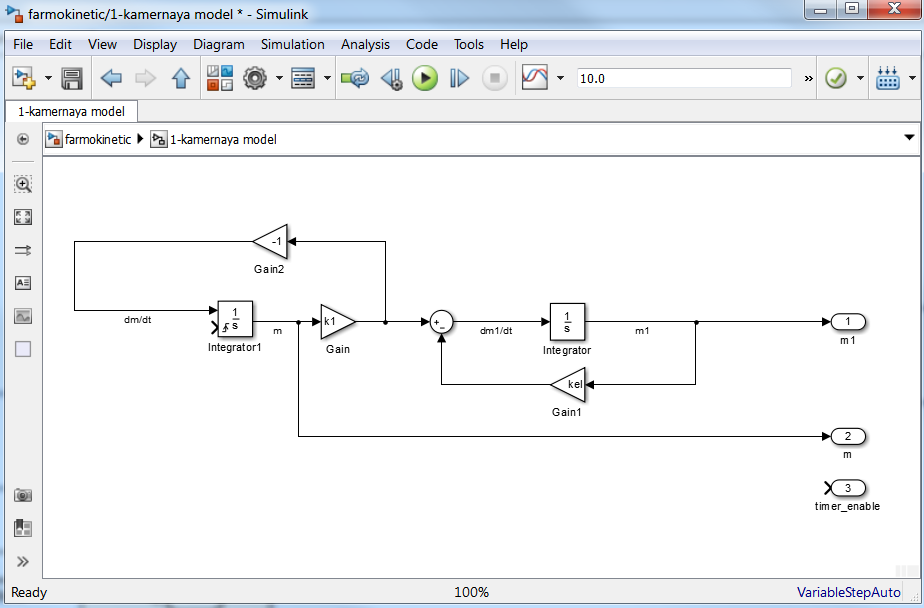


Рис. 9

Блок Integrator осуществляет интегрирование уравнения (2), а Integrator1 решает дифференциальное уравнение (1). Настройки блока integrator оставляем по умолчанию, а integrator1 выставляем как на рис. 10:

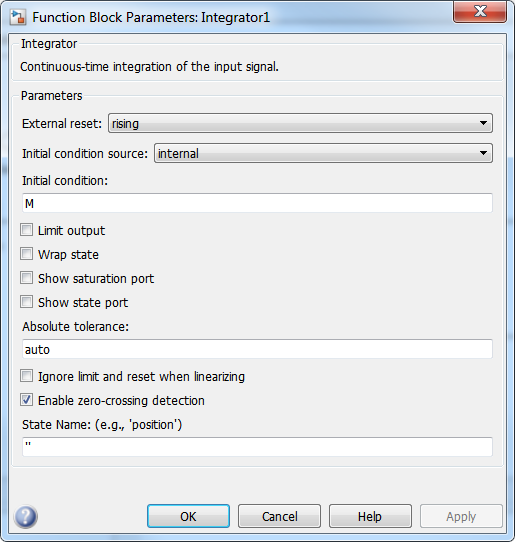


Рис. 10

Initial condition – начальное условие моделирование, в начале интегрирования в месте ввода лекарства находится M мг препарата.

External reset/Rising – сброс интегратора к начальным условиям осуществляется по возрастающему фронту внешнего сигнала. Это нужно для того, чтобы выполнить уловие задание по “в месте введения уничтожаются все остатки прежней дозы препарата и вводится новая доза m=M”

Схему сброса по таймеру осуществим с помощью блока Chart из раздела библиотеки Stateflow. Этот блок реализует диаграмму состояний и переходов между ними.

Перенесем Chart в подсистему и откроем его. Создадим три состояния (рис. 11)

Wait – ожидание пока масса препарата в точки введения не станет меньше пороговой

Count – отсчет промежутка времени Time

Reset – генерация сигнала сброса

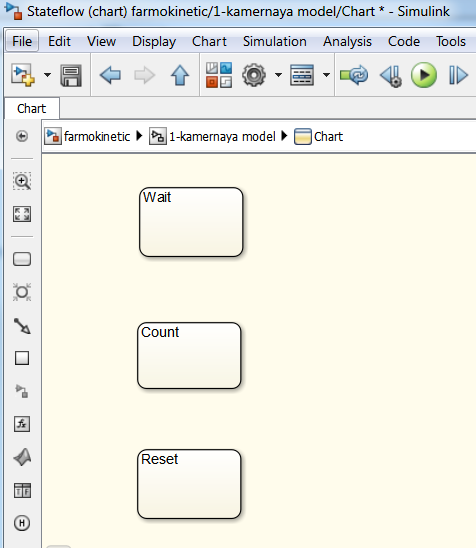


Рис. 11

В состоянии Wait установим “reset=0;” -сигнал сброса выключен, “timer\_enable=0;” - таймер не считает.

В состоянии Count установим “timer\_enable=1;” – таймер начинает отсчет.

В состоянии Reset установим “reset=1;” – сигнал сброса включен.

Установим Default transition на состояние Wait, т.е изначально мы находимся в состоянии ожидания. Установим условия других переходов как на рис. 12.

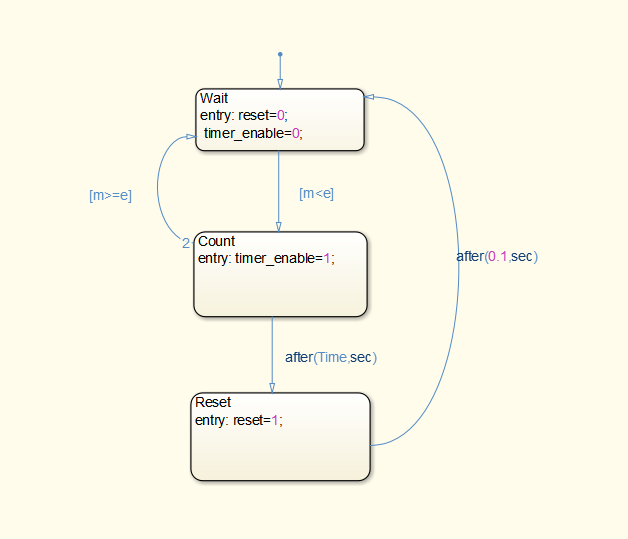


Рис. 12

Нажмем на кнопку run. Simulink предложит создать входные и выходные порты для блока Chart. Нажмем ОК (рис. 13)

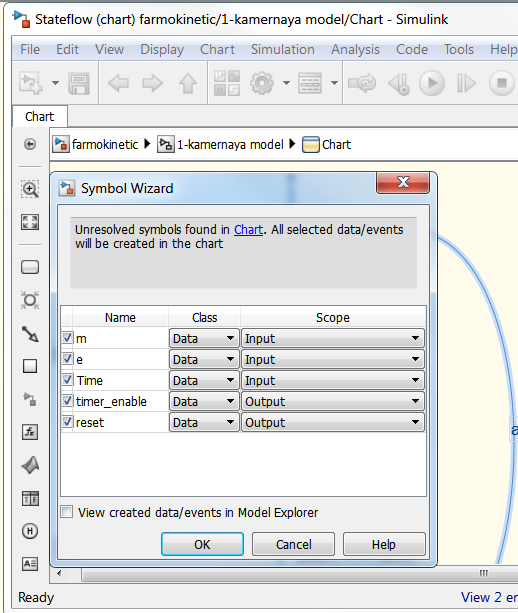


Рис. 13

Соеденим все как на рис. 14, добавив в модель два блока Constant со значениями e и Time

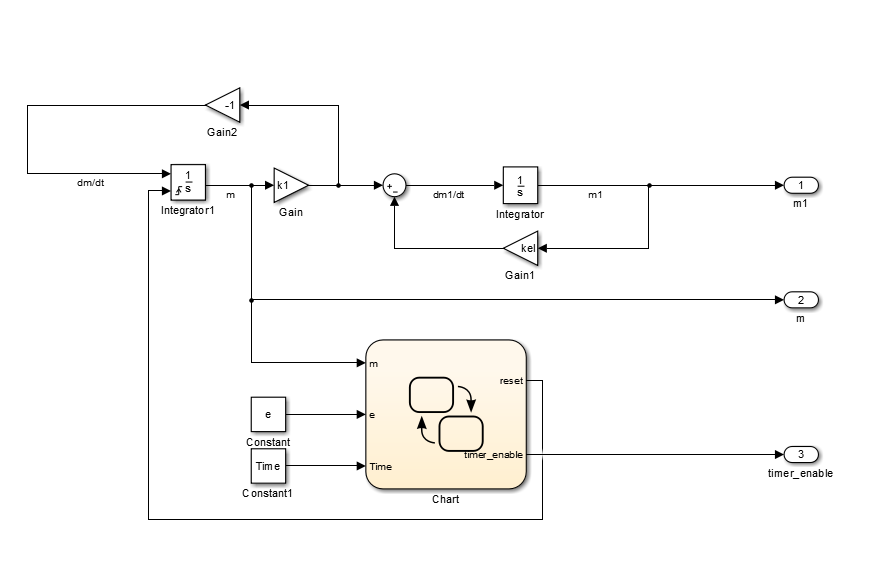


Рис. 14

Блок подсистемы через блок mux подключаем к блоку Scope (рис. 15)

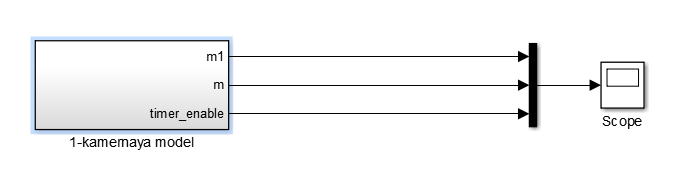


Рис. 15

В настройках блока Scope на вкладке Display установим галку на Show legend, чтобы отобразить подписи к линиям. (рис. 16)

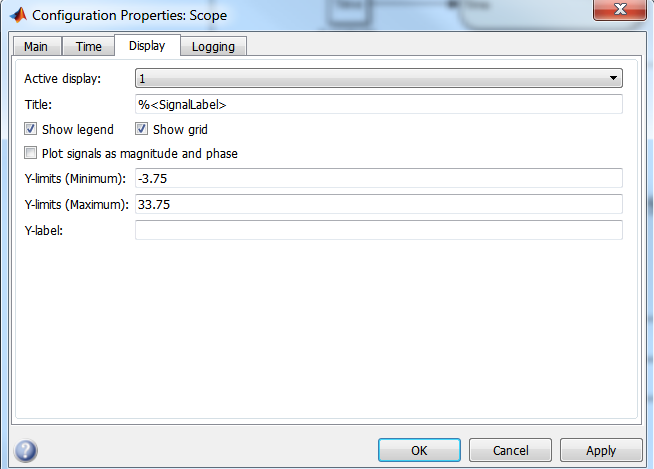


Рис. 16

В настройках модели установим Stop time = 100 – моделировать 100 секунд, Solver – ode45 – метод Рунге-Кутты 4 порядка, Max step size 0.001, чтобы моделирование было более точным. (рис. 17)

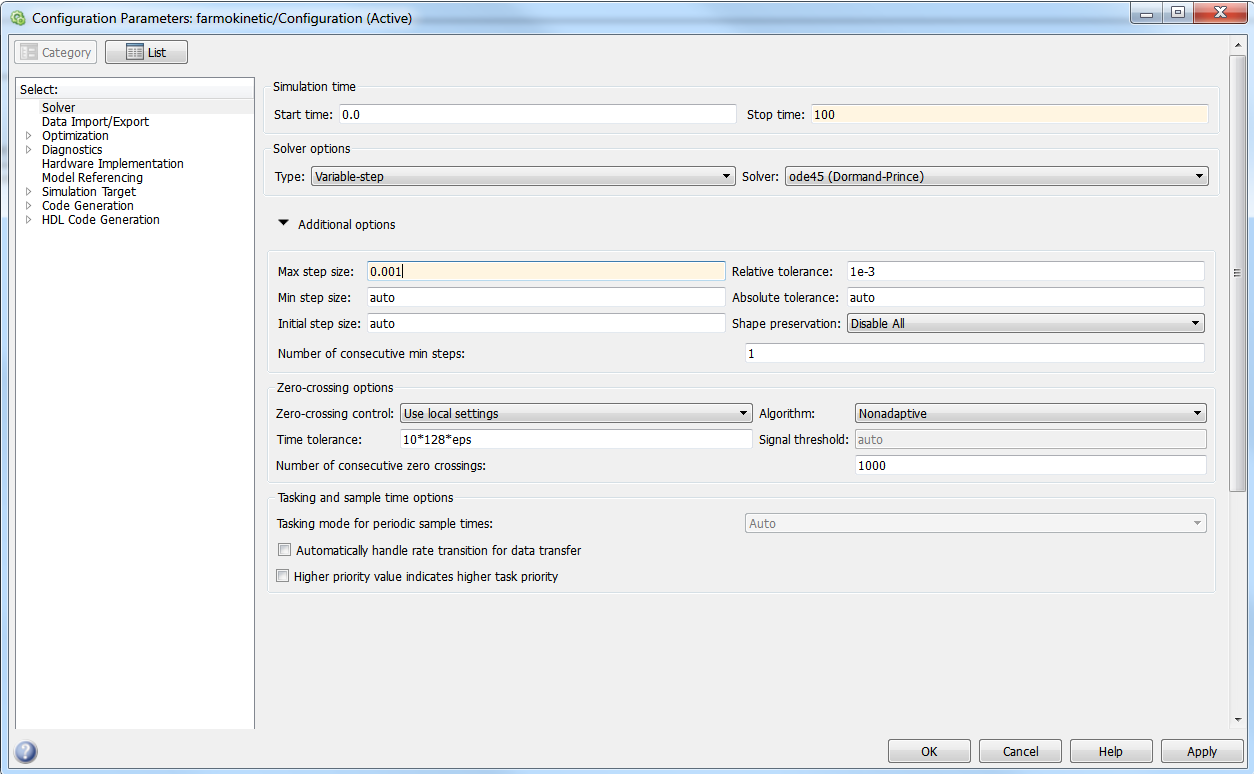


Рис. 17

Запустим модель на выполнение, нажав Run и откроем Scope, чтобы посмотреть график. На рис. 18 видим результат работы. Синяя линия –масса лекарства в точке ввода, желтая – масса препарата в камере, оранжевая – показывает момент отсчета периода времени Time таймером. По завершении отсчета происходит сброс остатков препарата в точке ввода и начинается новый цикл.

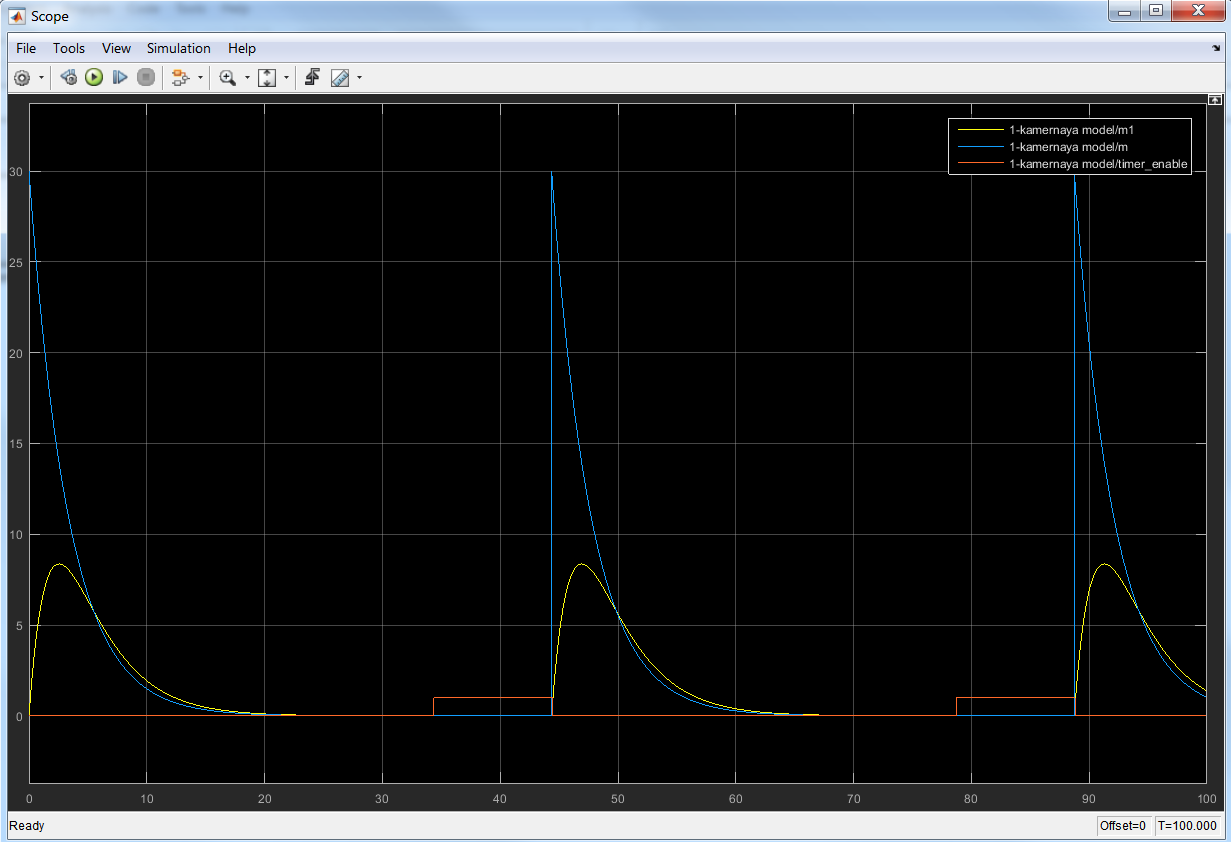


Рис. 18

Далее создадим двухкамерную модель. Так как она лишь незначительно отличается от однокамерной, то скопируем подсистему с однокамерной моделью вместе с блоками mux и scope, переименуем скопированную подсистему как "2-kamernaya model“ (рис. 19)

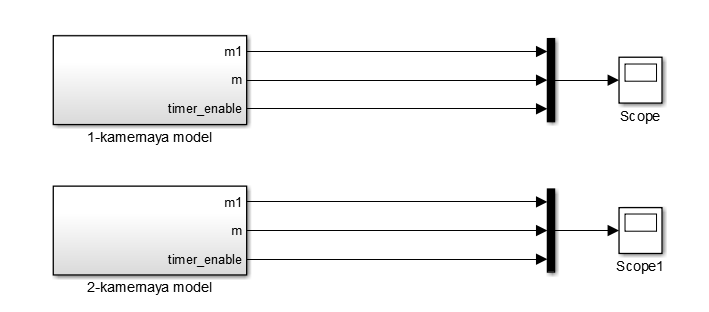


Рис. 19

В подсистеме двухкамерной модели добавим вторую камеру (рис. 20) Настройки integrator2 оставим по умолчанию, настройки самого левого блока сумматора List of signals «||++-»

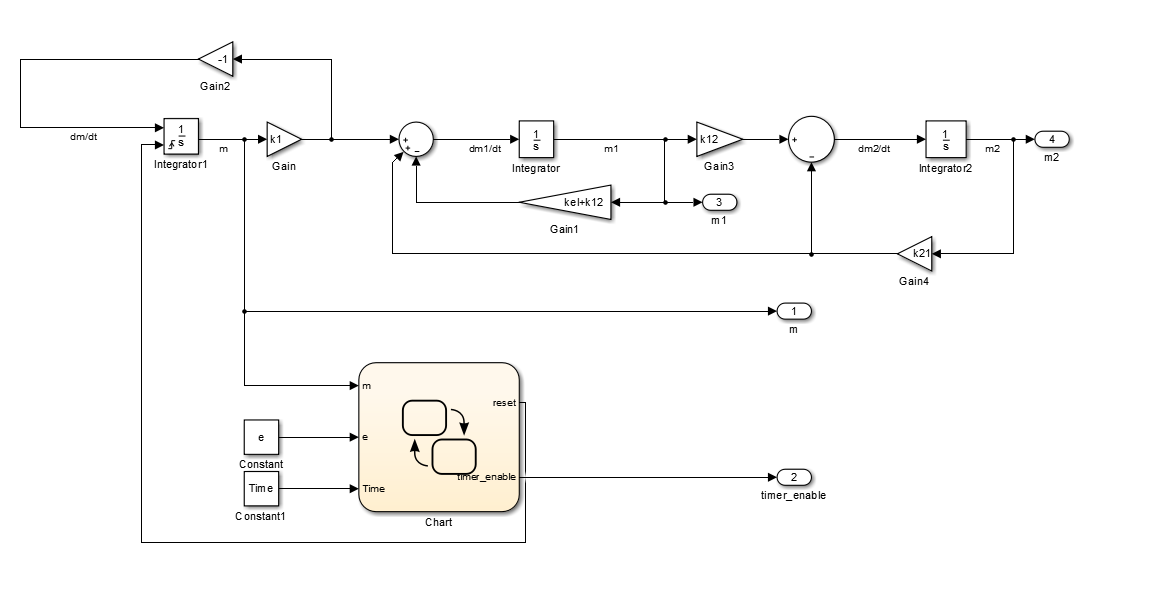


Рис. 20

Подключаем подсистему через mux(изменив в настройках 3 порта на 4) к scope (рис. 21)

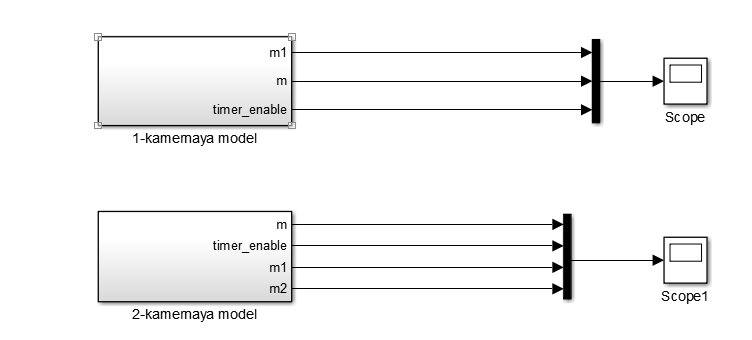


Рис. 21

Запускаем моделирование и открываем Scope1 (рис. 22). Видим что характер поведения линии m1 слегка изменился, так как часть лекарства переместилась в камеру 3 (масса m2 – зеленая линия)

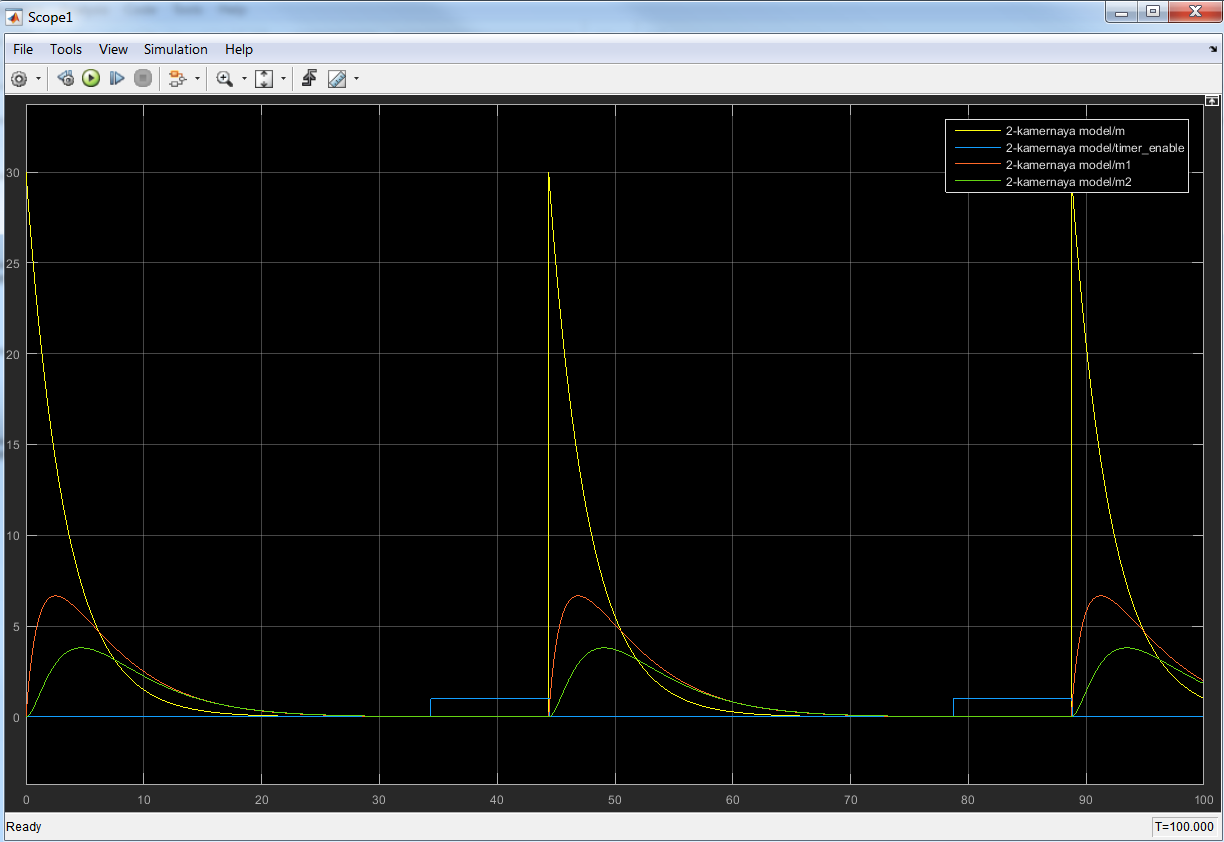


Рис. 22

Осталось сделать третью подсистему, в которой в обе модели препарат вводится из одной точки. Снова копируем подсистему 2-kamernaya model, называем Obe modeli. Копируем из 1-kamernaya model участок показанный на рис. 23 и вставляем в Obe modeli

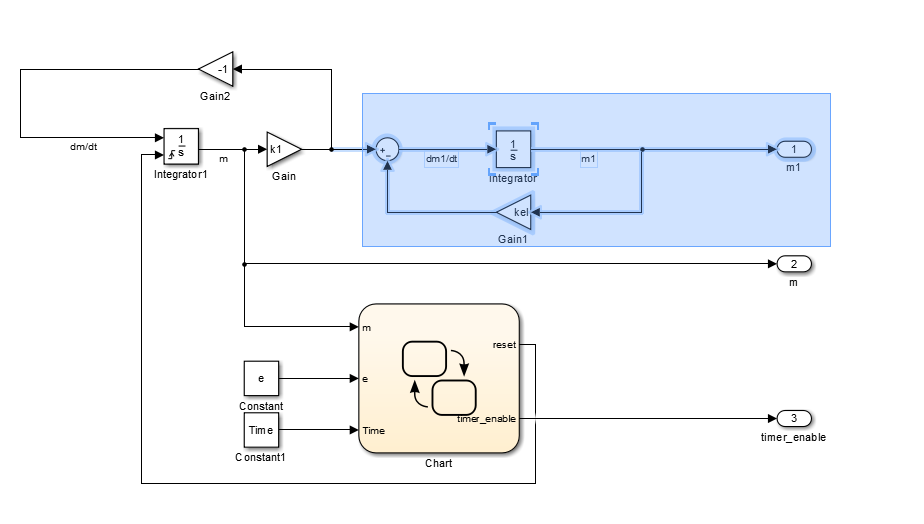


Рис. 23

Соединяем все как на рис. 24. Не забываем изменить значение Gain на 2\*k1 так как теперь расход стал в 2 раза больше, а также добавить блок Gain8 со значением 0,5, чтобы на каждую модель пришлось по половине общего потока. Также переименовываем порты, чтобы отличать выходы 1-камерной и 2-камерной моделей.

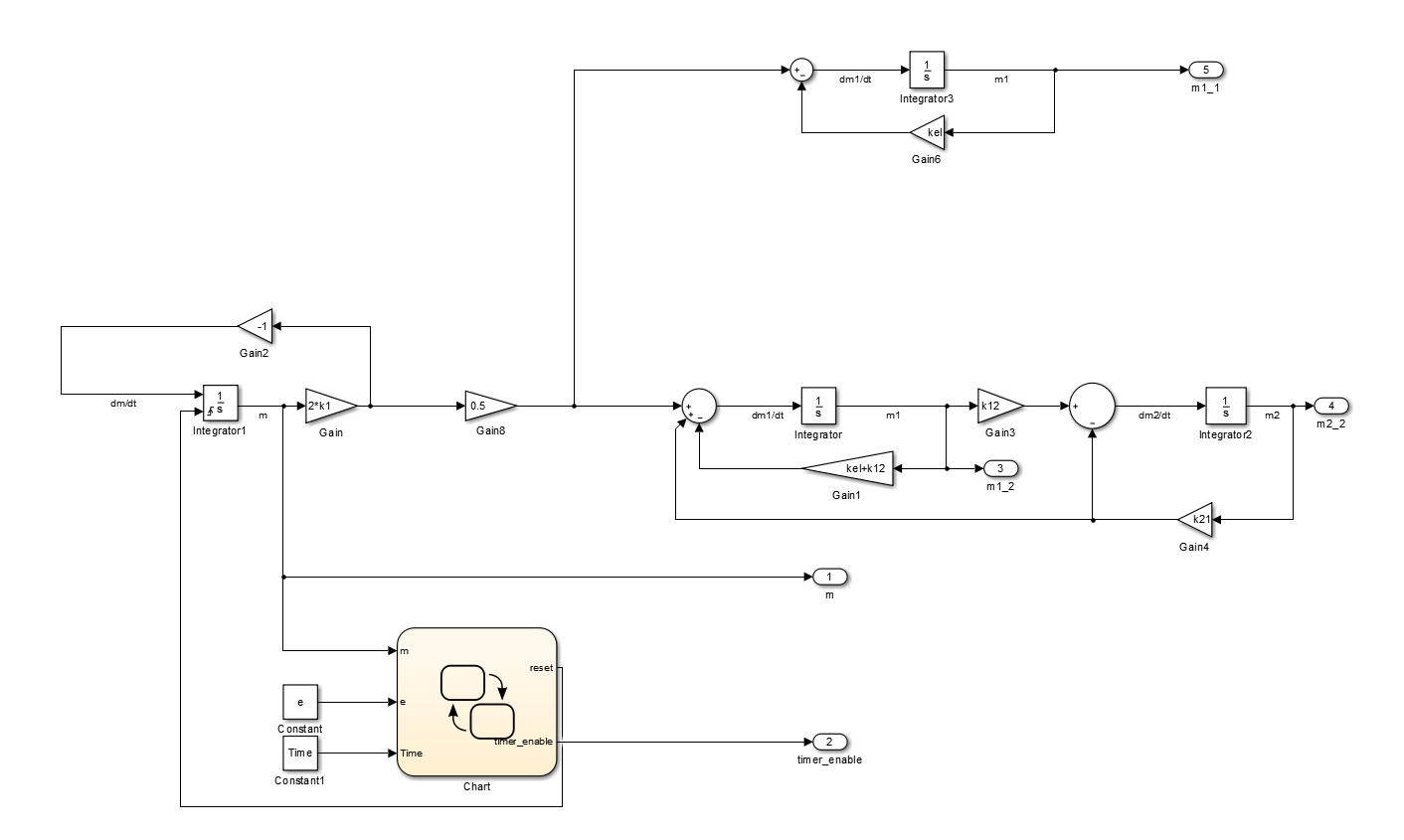


Рис. 24

Подключим все как на рис. 25

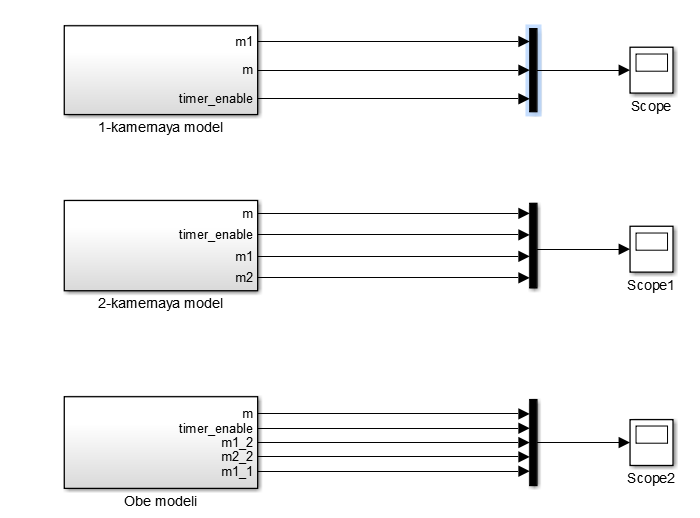
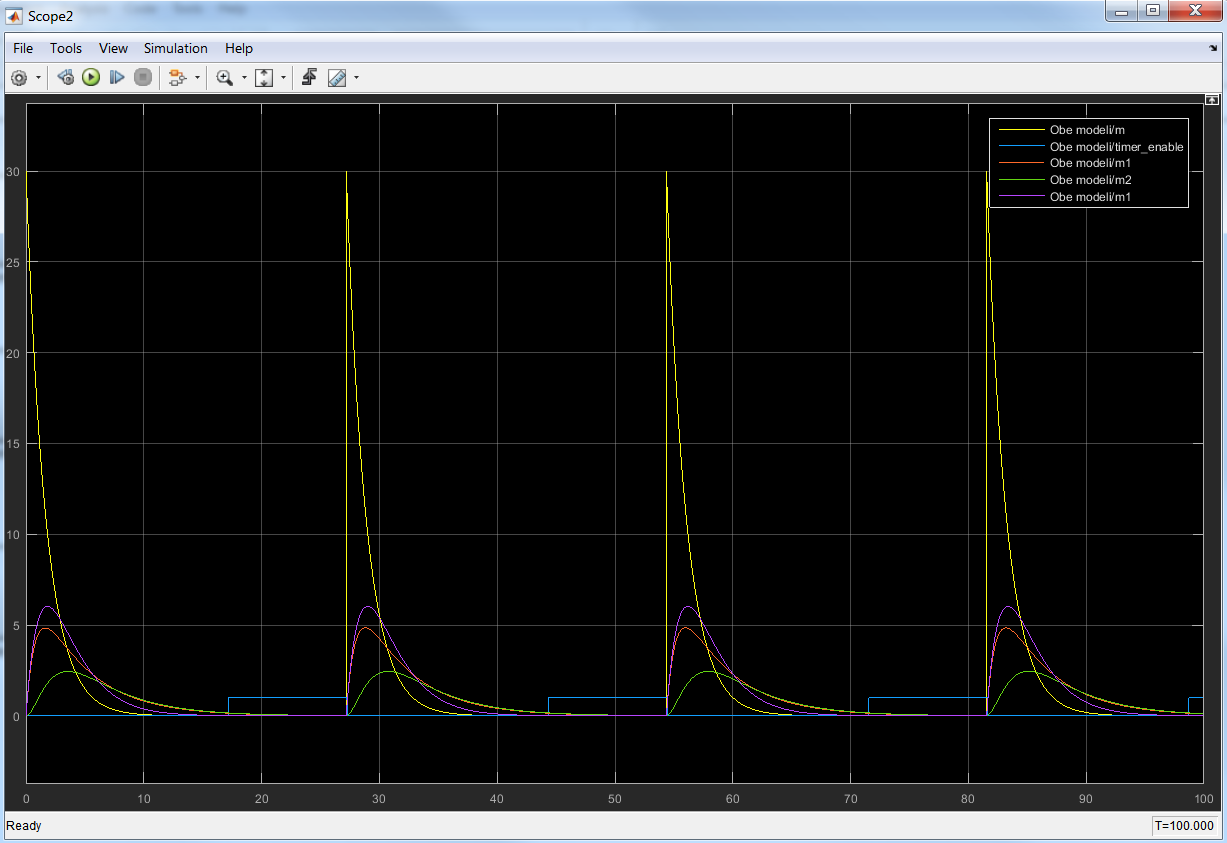


Рис. 25

Запустим моделирование и откроем блок scope2 (рис. 25) Видим, что теперь расход препарата происходит в 2 раза быстрее, следовательно сбросы происходят почти в 2 раза чаще (не точно в 2 раза из-за таймера в 10 секунд).

Рис. 26

Теперь сделаем так, чтобы в подсистемах 2-kamernaya model и Obe modeli после того, как m достигает порогового значения е, по истечении времени Time препарат не выводился полностью и потом туда же подавалась новая доза.

Для этого в подсистеме 2-kamernaya model нужно изменить настройки Integrator1, а имеенно установить Initial condition source = External (рис. 27) – наружный ввод начального значения. Снаружи интегратора начальное значение каждой следующей дозы формируется как сумма исходной дозы M и того, что не було введено до конца, т.е. m. Запоминание недовведенного остатка осуществляется через блок Memory с настройками по цмолчанию. (рис. 28)

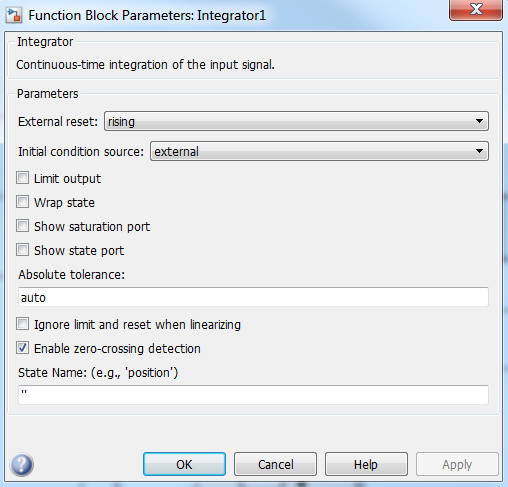


Рис. 27

.

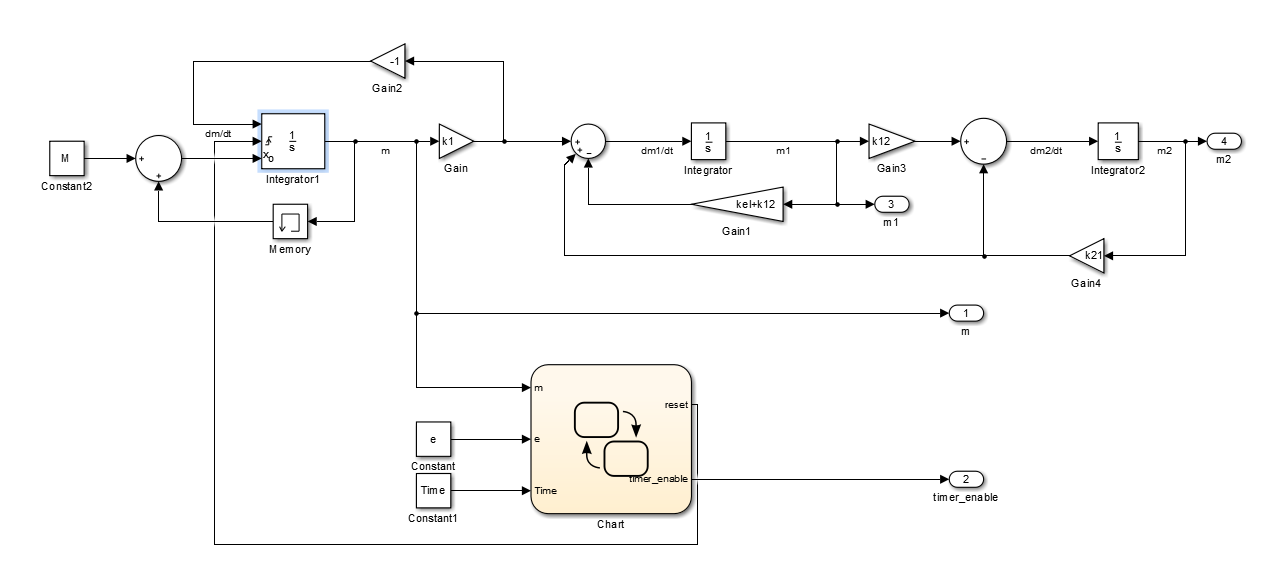


Рис. 28

Проделаем все то же самое с подсистемой Obe modeli

При текущих настройках изменения в поведении систем почти не заметны. Изменим пороговое значение e с 0.001 до 10, а также уменьшим Time с 10 до 5, набрав в Comand window MATLABa >>e=10;Time=5;

Запустим моделирование и посмотрим результат в Scope1 и Scope2. Видим, что после сброса значение m теперь не 20, а 20+δ, где δ остаток от предыдущего цикла введения препарата. (рис. 29, 30)

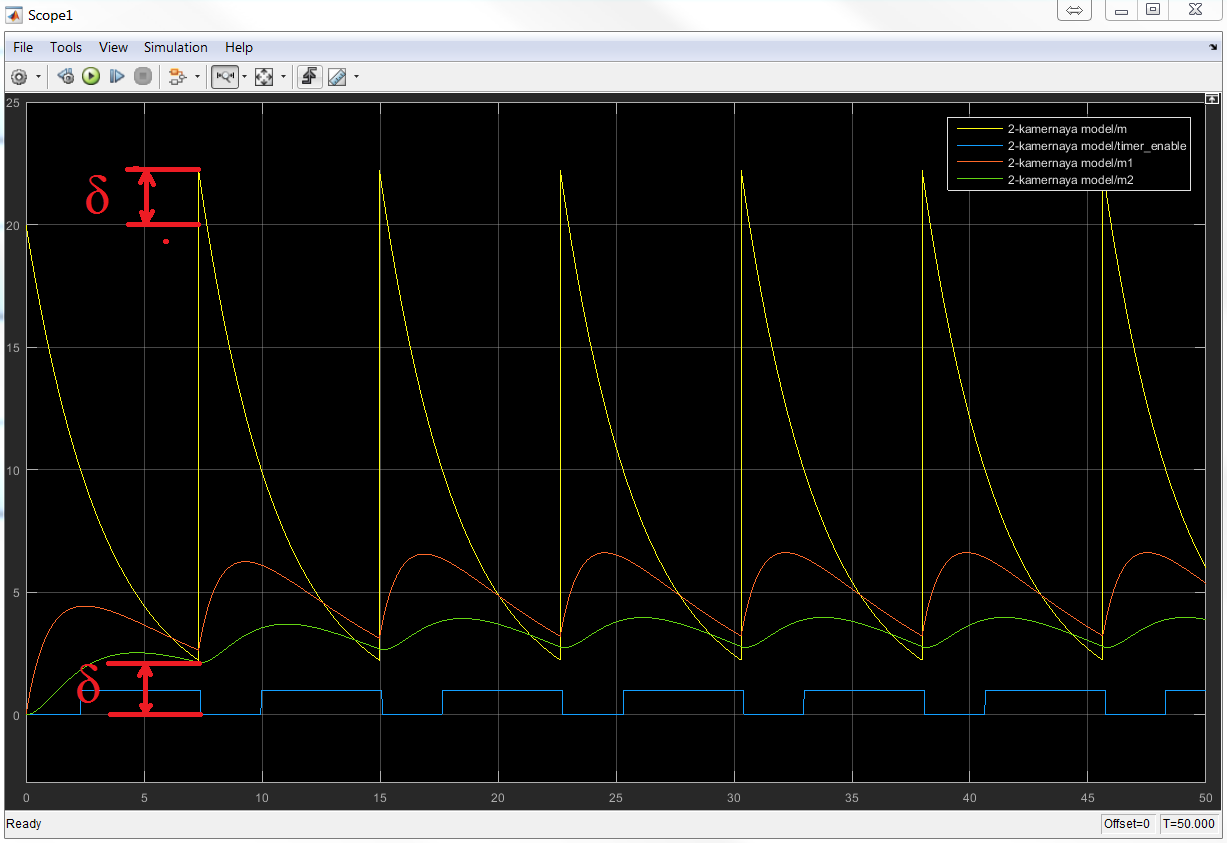


Рис. 29

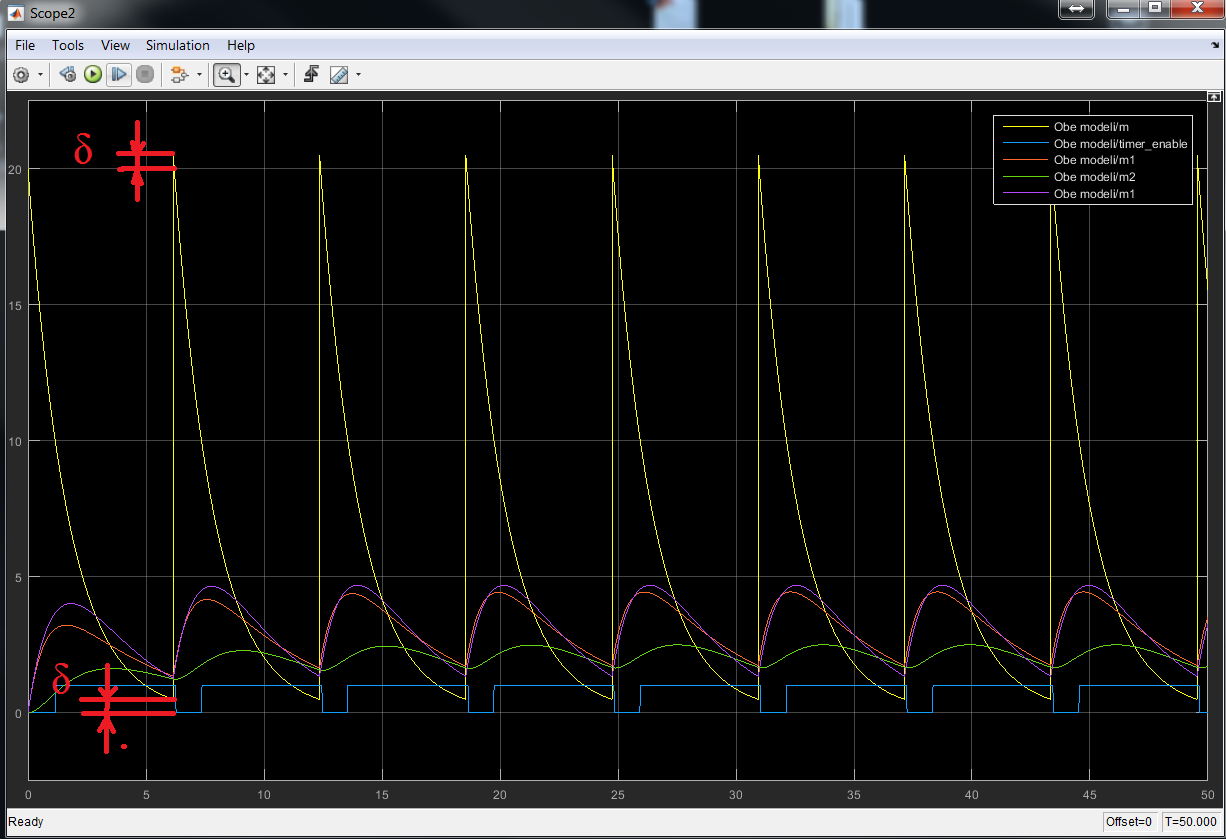


Рис. 30

В этом случае ввиду того, что порог включения таймера в каждом цикле одинаковый, то и по истечении таймера Time в точке ввода препарата остается одно и то же значение препарата δ, поэтому в каждом цикле кроме первого будет вводиться M+δ препарата, что мы и видим на рис. 29 и 30.

Это можно показать математически. Решение дифференциального уравнения (1) на первом участке:

Здесь е это экспонента, а не порог включения таймера. Обозначим порог сброса E, чтобы не путаться. Тогда в момент включения таймера t1: , а в момент t1+Time: . Отсюда видно, что если E и Time не меняются от цикла к циклу, то не меняется и δ. Значит в каждом цикле кроме первого будет вводиться M+δ препарата.

Если мы хотим, чтобы δ возрастала в каждом цикле, то нам нужно либо изменять порог сброса Е, либо изменять значение Time.

Построим такую схему, где E(n)=n\*E, здесь n –номер цикла (n=1,2,3…). Тогда

Для осуществления описанного выше изменим подсистемы 2-kamernaya model и Obe modeli. Добавим счетчик, который будет считать сигналы сброса (блок Counter с настройками на рис. 32). Умножим значение счетчика на E, таким образом в каждом последующем цикле таймер быдет запускаться по достижениий сигналом m значения E, 2E, 3E и т.д. (Рис. 31). Также установим значения Time=5 и e=1, чтобы лучше видеть результат.

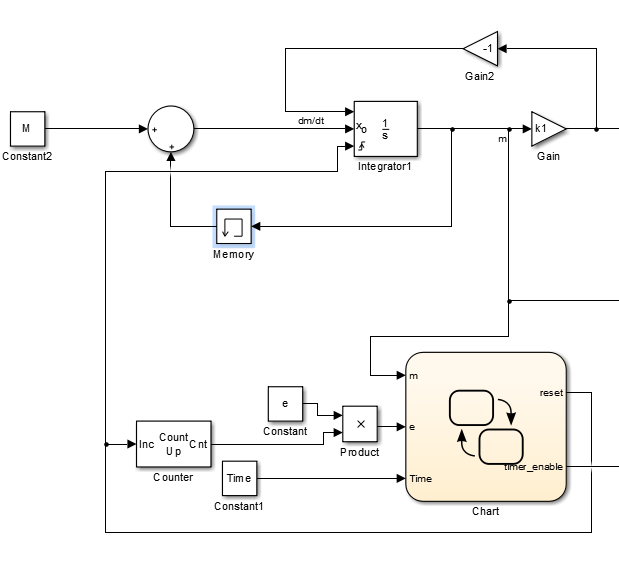


Рис.

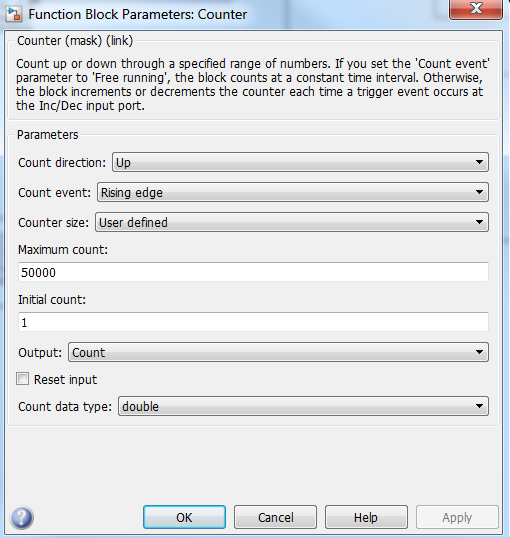


Рис.

Результаты моделирование на рис. 33 и 34

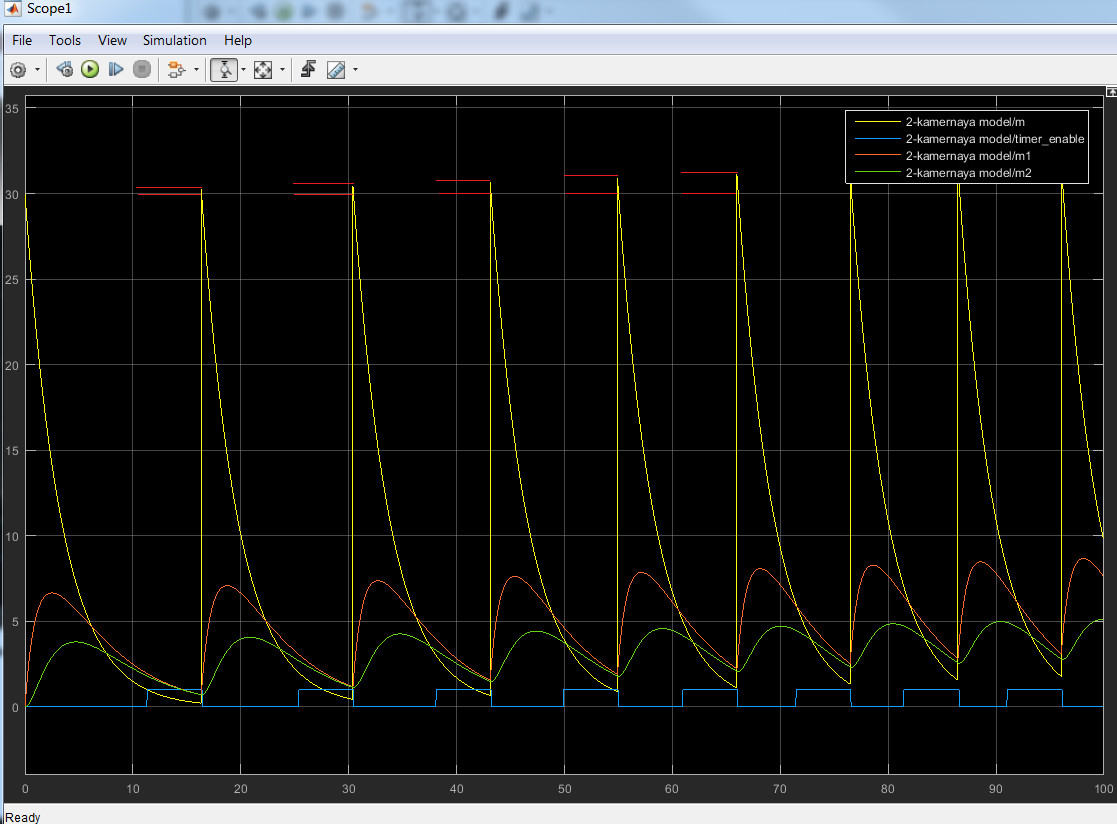


Рис.

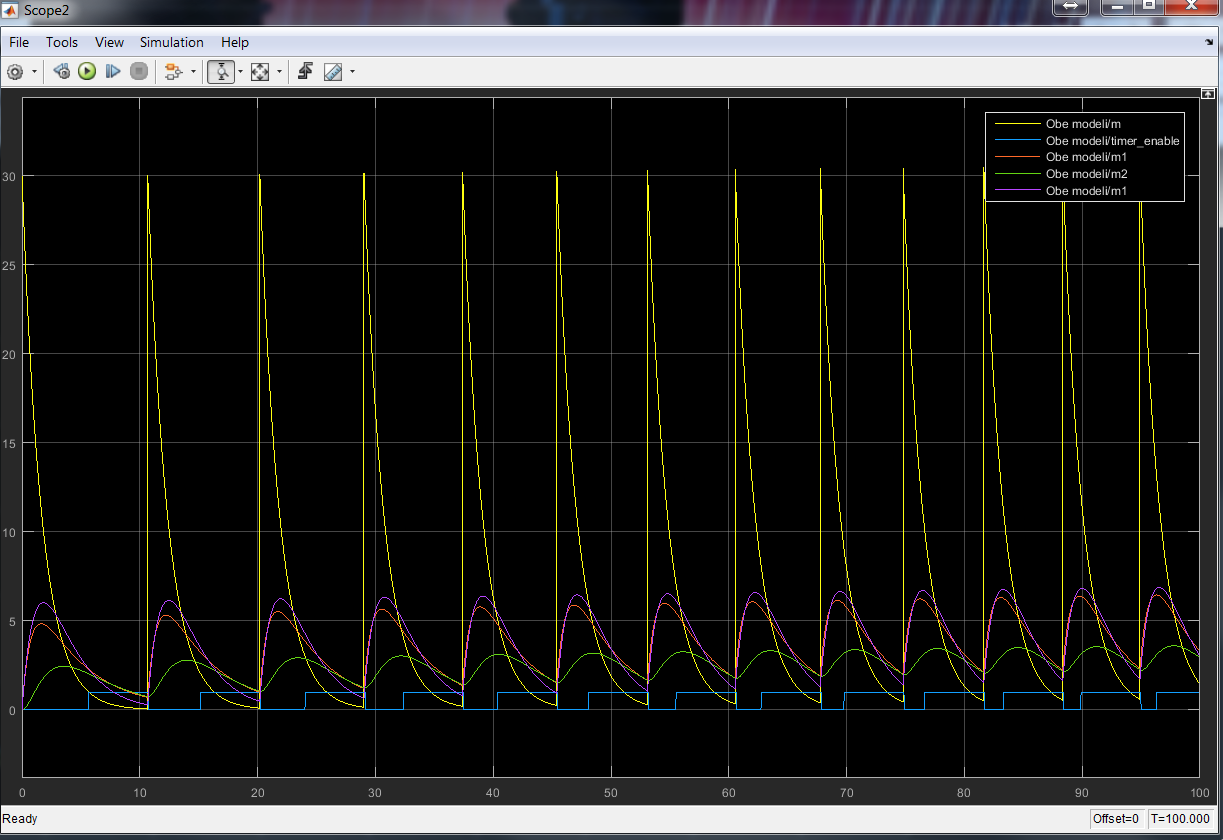


Рис.