Dr Bob Davidov

# Stateflow моделирование процессов

***Цель работы:*** Изучить правила моделирования stateflow процессов в среде MATLAB.

***Задача работы:*** Построение stateflow моделей.

***Приборы и принадлежности:*** Персональный компьютер, MATLAB R2015.

## *Введение*

Среда Stateflow (MATLAB Simulink) предназначена для моделирования логики принятия решений, основанных на машинах состояний и блок-схемах, включая моделирование логики диспетчерского управления, планирования задач и диагностического тестирования для выявления некорректных условий, переходов и недостижимых состояний.

Диаграмму состояний Stateflow можно транслировать в код PLC, C, C++ и HDL.

В этой работе рассматриваются особенности технологии Stateflow MATLAB на примерах построения Широтно-Импульсных Преобразователей (ШИП)

### ОБЩИЕ СВЕДЕНИЯ

Stateflow применяется для моделирования событийных систем; машин состояний; графов; конечных автоматов Мура и Мили. В Мили модели действия связаны с переходами, в то время как в Мура модели они связаны с состояниями.

ПРИМЕЧАНИЕ. Для построения простых событийных систем можно обойтись стандартными средствами if else, switch case.

Stateflow диаграмма (Chart) должна быть частью подсистемы Simulink. Chart состоит из набора

- графических (состояния, переходы, соединения (узлы), хронологические соединения) объектов,
- неграфических (события, данные, программные коды) объектов,
- отношений между этими объектами.

Компоненты Stateflow могут запускаться на выполнение с использованием логики по времени или по условию. Компоненты могут включать функции MATLAB и Simulink, собственный C-код, таблицы истинности.

## Графические объекты

Stateflow диаграмма (chart) может включать следующие графические объекты.

- Состояние (State)
- Ящик (Box)
- Подключаемое соединение (Connective junction)
- Хронологическое соединение (History junction)
- Переход (Transition)
- Переход по умолчанию (Default transition)
- Графическая функция (Graphical function)

### Состояние (State)

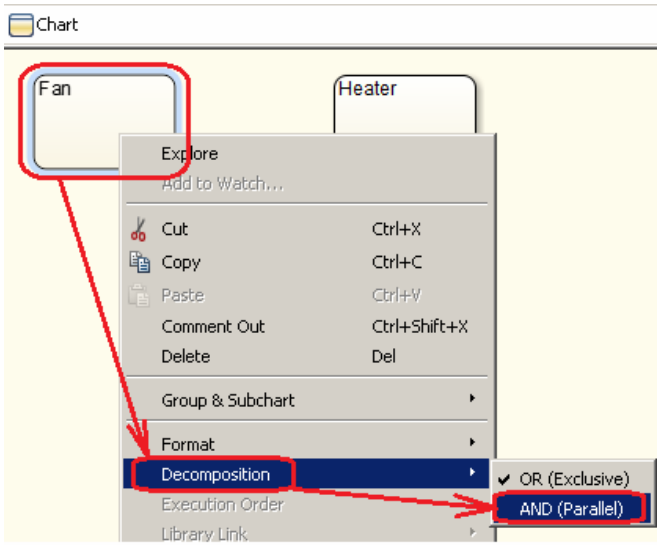
Состояния процессов моделируют графические объекты State. В меню диаграммы Chart ‘состояние’ обозначается прямоугольником значком с скругленными углами. В диаграмме состояние представлено объектом


, в котором знак вопроса заменяется именем состояния и метками.

Имя состояния может заканчиваться символом /. Состояния могут включать последовательность команд (например, pwm = 0; tooth = tooth + 1; и др.), которые выполняются вначале (entry), во время (during), или в конце выполнения (exit) состояния, в зависимости от используемых ключевых слов (действий):

- **en**try (на входе состояния),
- **du**ring (в течение состояния),
- **ex**it (на выходе состояния) и
- on event_name (в случае возникновения события с именем).

Stateflow обеспечивает два типа состояний: параллельный (И), когда состояния выполняются одновременно, и исключительный (ИЛИ), когда состояния выполняются последовательно. Границы параллельных (И) состояний отображаются штриховыми линиями. Пример задания типа состояния И (AND) приведен на рисунке ниже. В примере, подсостояния (внутренние блоки) состояния Fan устанавливаются на параллельное выполнение (И).



### Ящик (Box)

Ящик используется для выделения части Stateflow диаграммы. В меню диаграммы Chart ‘ящик’ обозначается прямоугольником. На диаграмме ящик изображается объектом


, в котором знак вопроса заменяется именем ящика.

### Подключаемое соединение (Connective junction)

Соединения обозначают точки принятия решений. В меню диаграммы Chart подключаемое соединение обозначается  . На диаграмме -  .

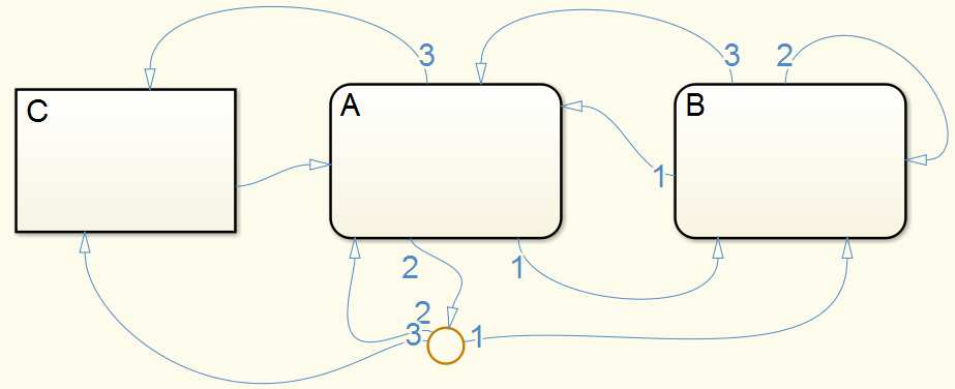
### Хронологическое соединение (History junction)

Хронология указывает, что при новом входе в состояние с хронологическим соединением активным становится то подсостояние, которое было активным последним при завершении состояния с хронологическим соединением.

В меню хронологическое соединение обозначается  . На диаграмме -  .

### Переход (Transition)

Переход моделирует изменение состояния при наступлении события. Переход может включать условия выполнения перехода. На диаграмме переход обозначается линией со стрелкой, которая связывает объекты. На диаграмме переходы рисуются нажатой мышкой, перемещаемой между границами соединяемых объектов. Переход может замыкаться на объект ‘состояние’ из которого выходит. Пример линий переходов без использования условий показан ниже.




Переход может иметь метки, которые описывают условия выполнения перехода. Метки перехода имеют следующий основной формат:

событие[условие]{действие условия}/действие перехода
event[condition]{condition_action}/transition_action,

например,

switch_on[p>20]{func1}
event2/data1=5
switch_off[d2]{func2}/light_off

ВНИМАНИЕ! Переходы, которые заканчиваются в соединениях, могут иметь только действия условий (не допускаются действия переходов).

Условия заключаются в квадратные скобки []. Для выполнения перехода условие должно быть истинным. Соединения с условиями имеют приоритет над соединениями без условий.

Иерархия переходов согласуется с иерархией состояний.

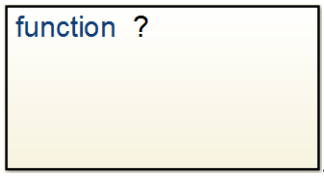
## Переход по умолчанию (Default transition)

Переход по умолчанию используется для указания на подсостояние, которое должно стать активным в момент активации состояния включающего подсостояния.

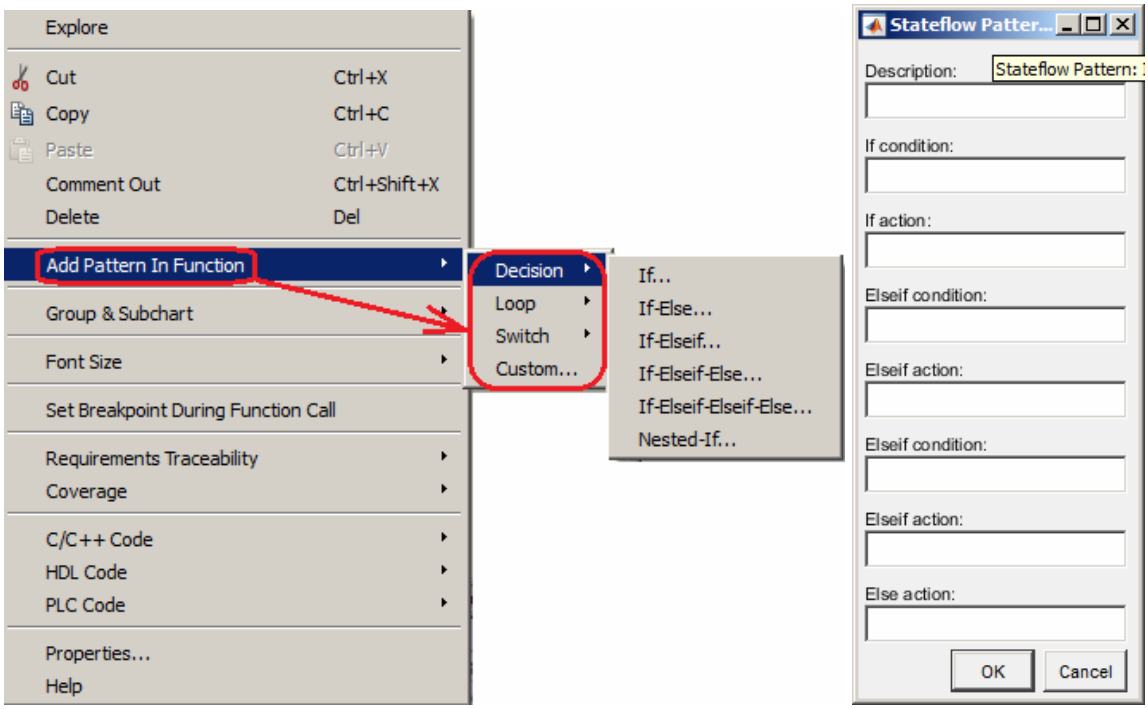
ВНИМАНИЕ! В параллельных (И) состояниях переходы по умолчанию всегда должны присутствовать.

В меню переход по умолчанию обозначается  . На диаграмме – стрелкой, исходящей из точки.

## Графическая функция (Graphical function)




В меню графическая функция обозначается  . На диаграмме - , в которой знак вопроса заменяется именем функции. Графическую функцию можно создать при помощи Stateflow Pettern, который можно открыть нажатием правой клавиши мыши по прямоугольнику функции:



## Неграфические Объекты

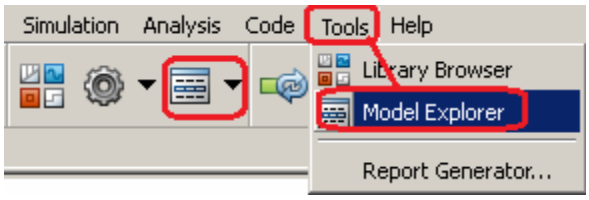
Stateflow включает следующие неграфические объекты.

- События
- Данные
- Коды

### События (Events)

События управляют выполнением диаграммы Stateflow. Все события диаграммы должны быть определены. Наступление события активирует состояние, запускает переход или действие. События наступают по-нисходящей, начиная от родителя события в иерархии.

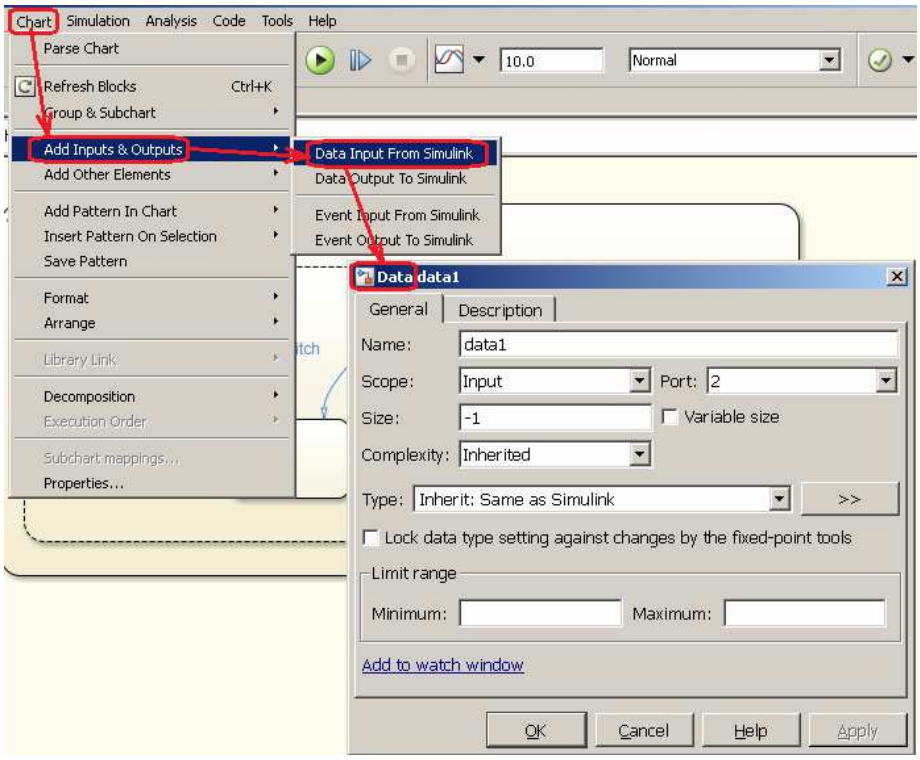
События создаются и изменяются при помощи Stateflow проводника (Stateflow Explorer):

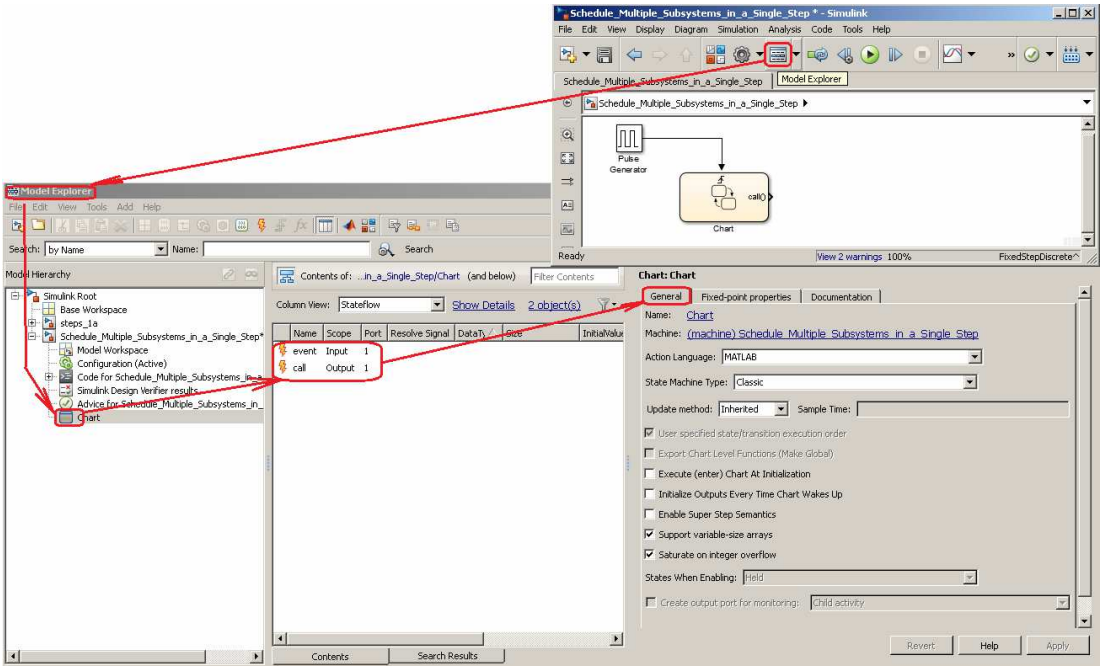



События могут быть созданы на любом уровне иерархии. События разделяются на

- Локальные (внутри диаграммы Stateflow)
- Вход в Stateflow диаграмму от модели Simulink
- Выход из Stateflow диаграммы в модель Simulink
- Экспортируемое в код, внешний к Stateflow диаграмме и модели Simulink
- Импортируемое из источника кода, внешнего к Stateflow и Simulink

Ниже показана последовательность создания входа данных в Stateflow из Simulink и проверка созданного входа события Event (значок ).






## Данные (Data)

Объекты-данные создаются и изменяются в Stateflow Explorer. Данные имеют такое свойство, как видимость. Видимость определяет для объектов-данных одну из следующих возможностей.

- Быть локальными для диаграммы Stateflow
- Поступать в Stateflow диаграмму от модели Simulink
- Выходить из Stateflow диаграммы в модель Simulink
- Быть временными данными
- Быть определенными в рабочем пространстве MATLAB
- Быть Константами
- Экспортироваться в код, внешний к Stateflow диаграмме и модели Simulink
- Импортироваться из источника кода, внешнего к Stateflow и Simulink

## Моделирование логики по времени

Темпоральные операторы (temporal logic операторы) позволяют задавать логику переходов между состояниями по числу событий или прошедшему времени без использования таймеров и счетчиков.

Stateflow поддерживает работу следующих операторов.

- after – после, например, after(3,sec)], [after(5,tick)]
- before – до
- at – в момент
- every - каждый

## Вызов функций MATLAB

Для вызова функций MATLAB в действии условия перехода необходимо указать имя функции матлаб, например, для выхода функции expo с аргументом Input: {expo(Input)}.

## Генерация HDL кода из Stateflow и Simulink

HDL Coder MATLAB может генерировать полноценный VHDL и Verilog код из блоков Stateflow и Simulink. Этот код расширяет возможности разработки аппаратных устройств в среде MATLAB, он позволяет создавать аппаратные устройства на основе программируемых логических интегральных схем (ПЛИС/FPGA) и специализированных интегральных схем (СИС /ASIC), разрабатывать прототипы, тестировать и проверять HDL-реализации Simulink-моделей. HDL Coder генерирует код без аппаратной привязки к конкретной платформе.

## Загрузка stateflow

Редактор диаграммы Stateflow chart находится в соответствующем разделе библиотеки Simulink (Рисунок 1).

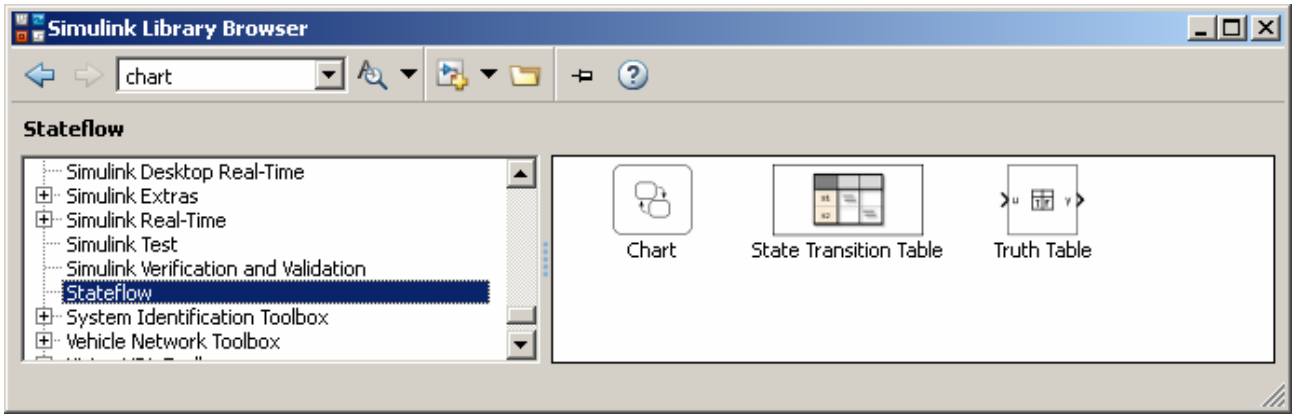


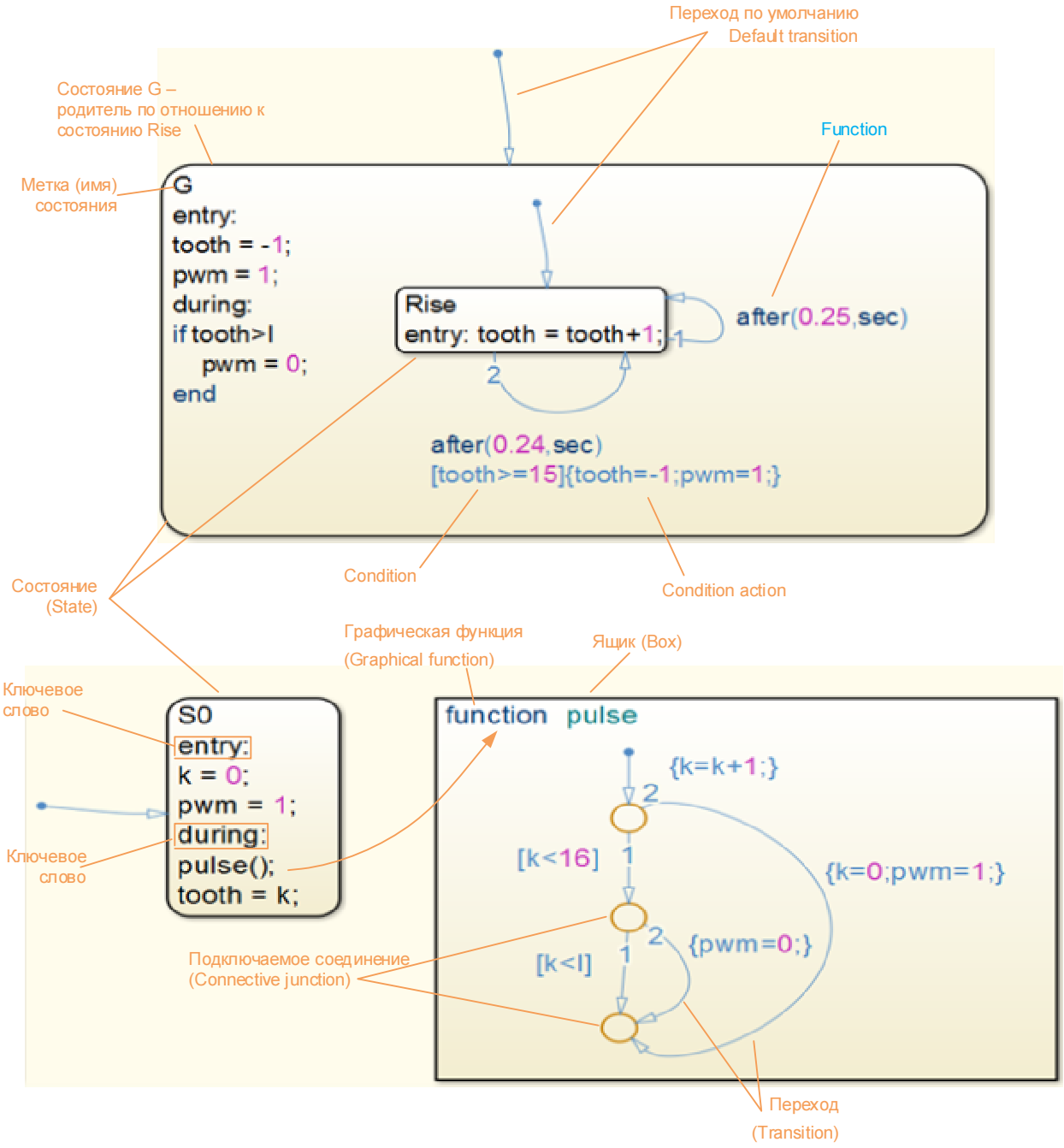

**Рисунок 1.** Содержимое раздела Stateflow библиотеки Simulink.

Раздел Stateflow библиотеки Simulink можно открыть и командой

>>stateflow

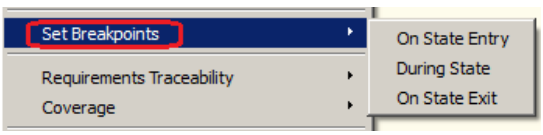
## Пример Stateflow модели

Пример Stateflow диаграммы Chart с обозначением объектов показан на рисунке ниже.




## Отладка Stateflow модели

Запускать Stateflow модель можно из любого сохраненного состояния как в непрерывном  , так и в пошаговом  режимах с использованием точек останова:




Отладчик Stateflow определяет ошибки времени выполнения, включая несогласованность состояний, нарушения диапазонов данных и потенциальные бесконечные циклы.

## ПРИМЕРЫ ПОЛУЧЕНИЯ ПРОВЕРЕННЫХ РЕЗУЛЬТАТОВ И ВАРИАНТЫ ДЛЯ САМОКОНТРОЛЯ

Технология Stateflow рассматривается на примерах построения различными средствами MATLAB 4-х разрядного Широтно-Импульсного Преобразователя (ШИП, PWM) с периодом 4 сек. Шаг моделирования 0.01 сек.

**Задание 1.** Построение ШИП средствами Simulink без применения Stateflow. Опорный (сравнительный) вариант.

1. Соберите Simulink модель, показанную на Рисунок 2.

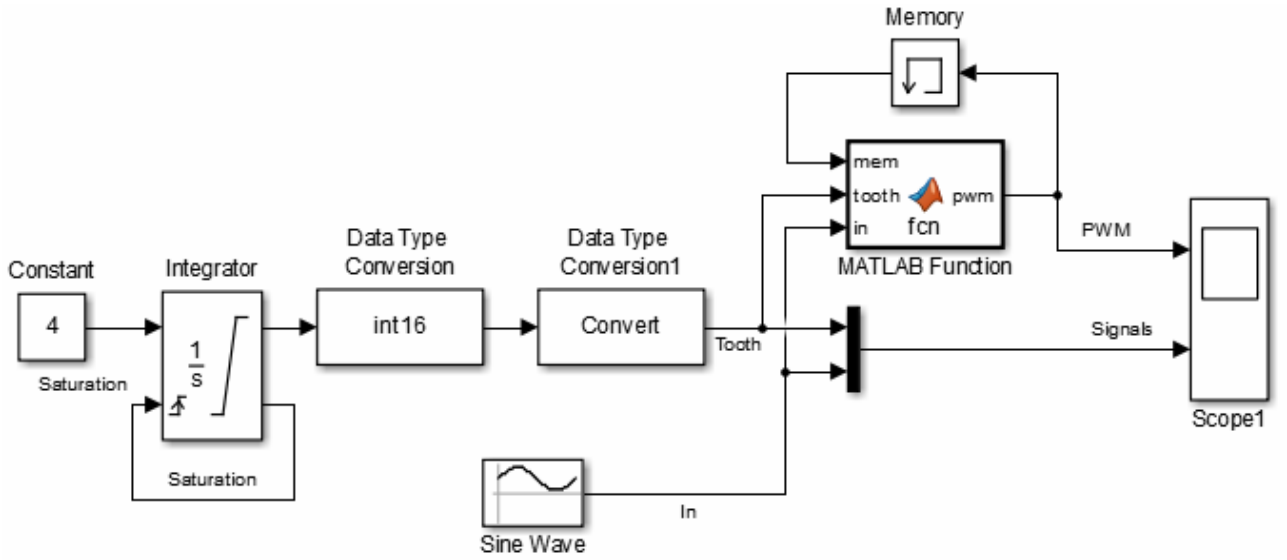
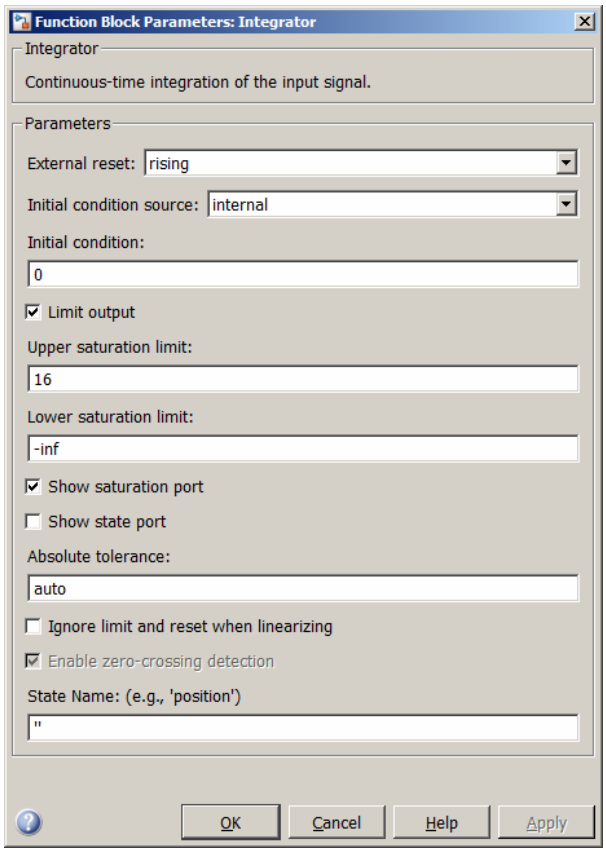
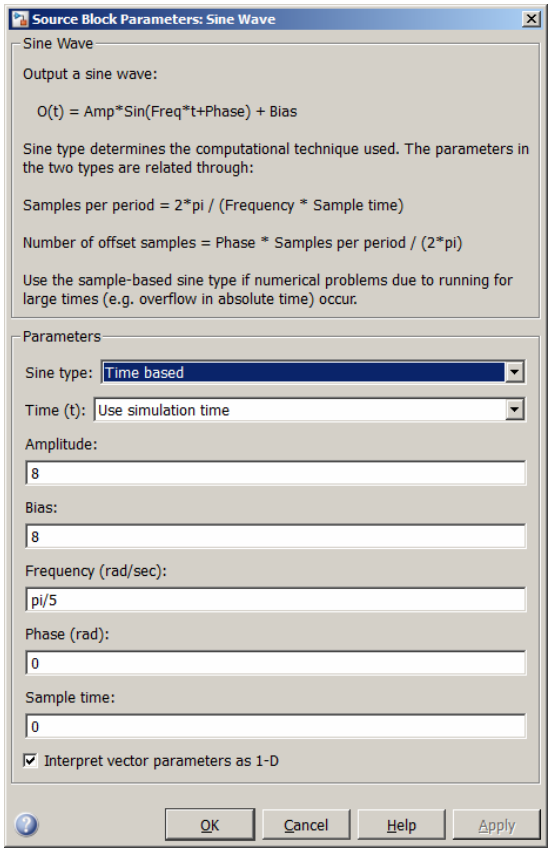
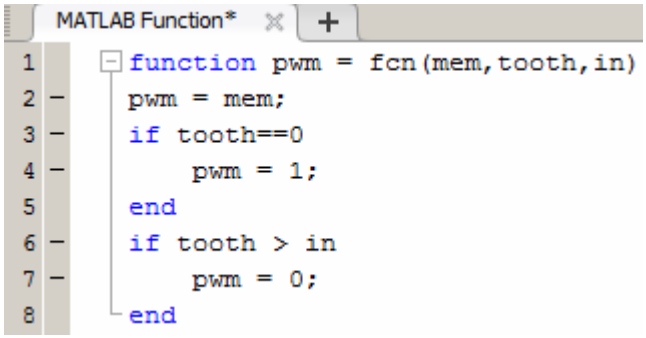


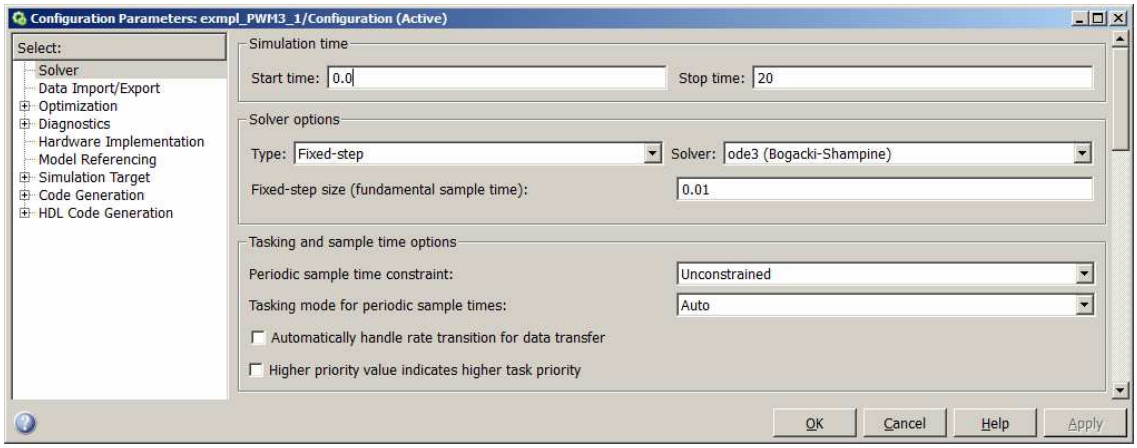

**Рисунок 2.** Сравнительный вариант модели ШИП (PWM). Пилообразный сигнал tooth формируется интегратором со сбросом. Функция MATLAB устанавливает ШИП в единицу в начале периода пилообразного сигнала и сбрасывает в ноль в момент превышения пилообразным сигналом Tooth входного сигнала In, формируемого генератором синусоидального сигнала Sine Wave.

2. Установите следующие параметры модели и режима моделирования.



3. Запустите модель на выполнение. Постройте графики сигналов PWM, Tooth и In.

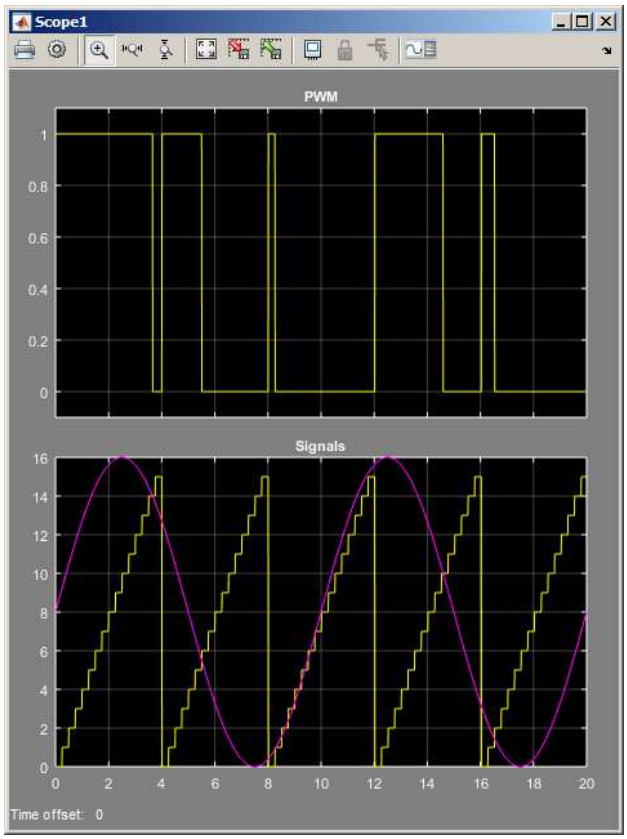
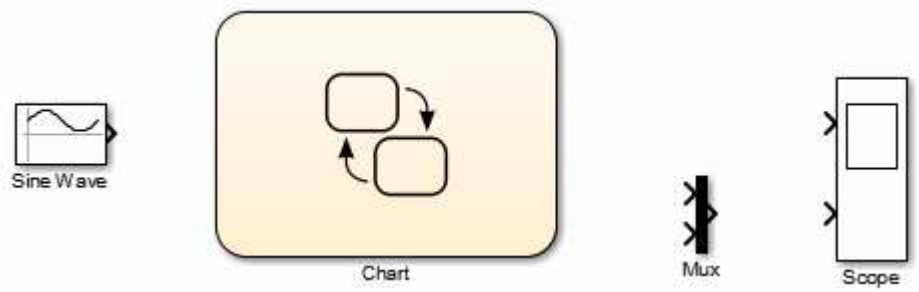



**Рисунок 3.** Сигналы модели формирования ШИП (PWM). Шаг моделирования 0.01 сек. Период пилообразного 16-ти уровневого (0:1:15) сигнала Tooth (выделен желтым на нижнем графике) равен 4 сек. Синусоидальный входной сигнал In показан малиновой линией. Выходной сигнал ШИП, получаемый сравнением входного и пилообразного сигналов, показан на верхнем графике. Длительность ШИП пропорциональна амплитуде входного сигнала.

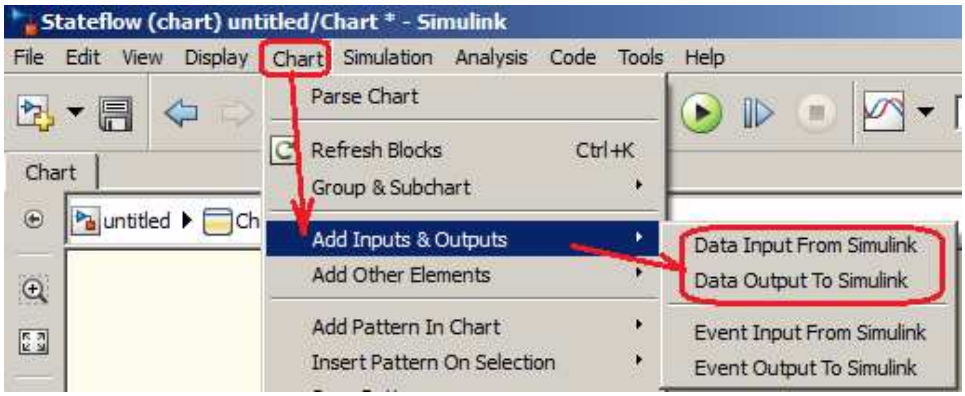
**Задание 2.** Построение ШИП с применением Stateflow моделирования логики по времени.

1. Откройте редактор построения Simulink моделей: MATLAB R2015 > Меню > New > Simulink Model.
2. Из библиотеки Simulink перетащите в окно моделирования следующие блоки.

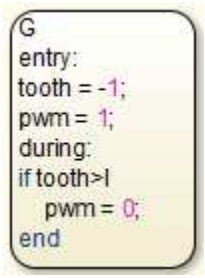



3. Установите режим моделирования и параметры Sine Wave как и в предыдущем задании (задание 1, п. 2).

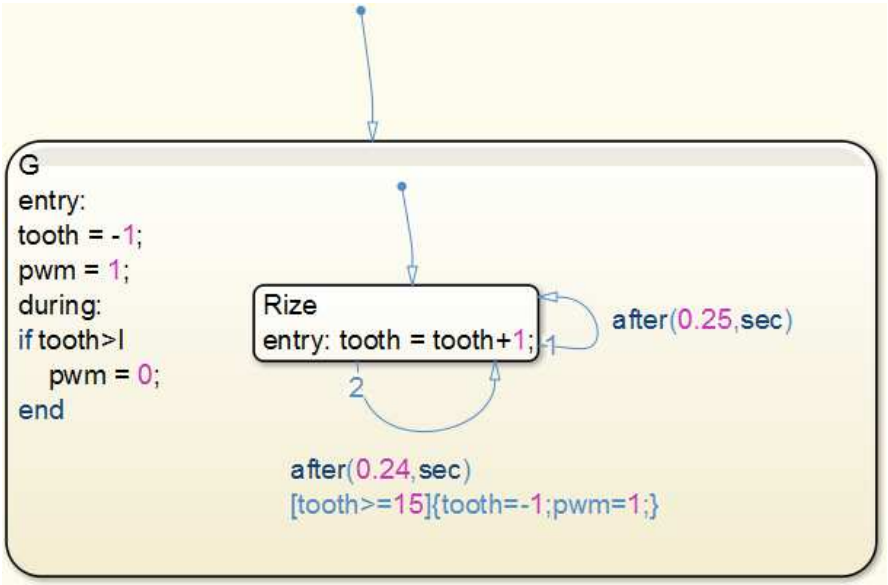
4. Откройте блок Chart - stateflow диаграмму.
5. Для связи диаграммы Chart с блоками Simulink установите один порт ввода (с именем I) и два порта вывода (pwm и tooth).



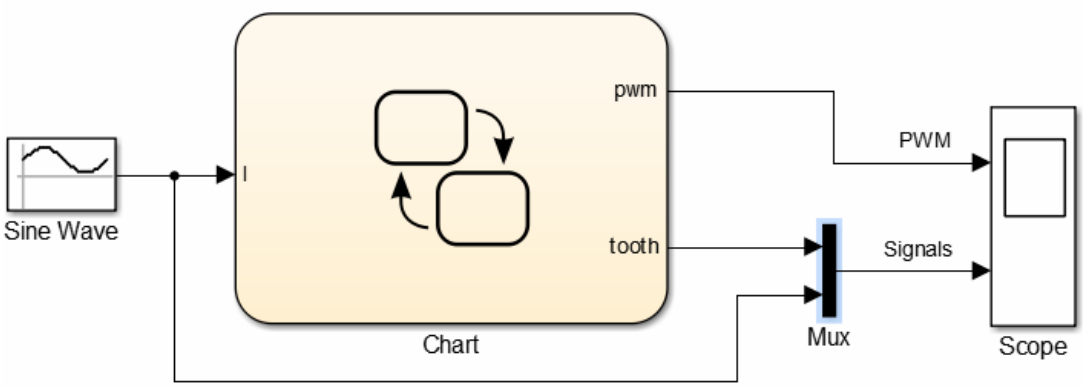

6. Введите в диаграмму Chart блок Состояние (State). Присвойте блоку имя, например, G, установите начальные (entry:) значения переменных tooth = -1; pwm = 1; и сброс pwm = 0 в продолжении работы (during:) блока G в моменты, когда tooth > I:



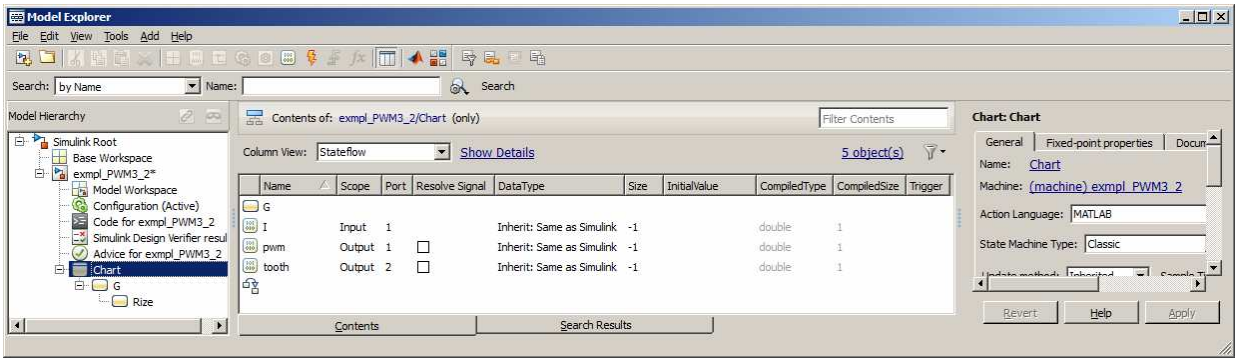

7. Введите в состояние G состояние с именем Rize, формирующим 16-ти уровневый пилообразный сигнал tooth. Включите в диаграмму переходы и установите их метки, которые описывают задержки и другие обстоятельства выполнения перехода. В квадратных скобках указываются условия выполнения перехода, в фигурных – значение переменных при выполнении перехода.



8. Перейдите на уровень выше stateflow диаграммы  .
9. Подключите к портам диаграммы Chart (I, pwm, tooth) блоки Simulink.



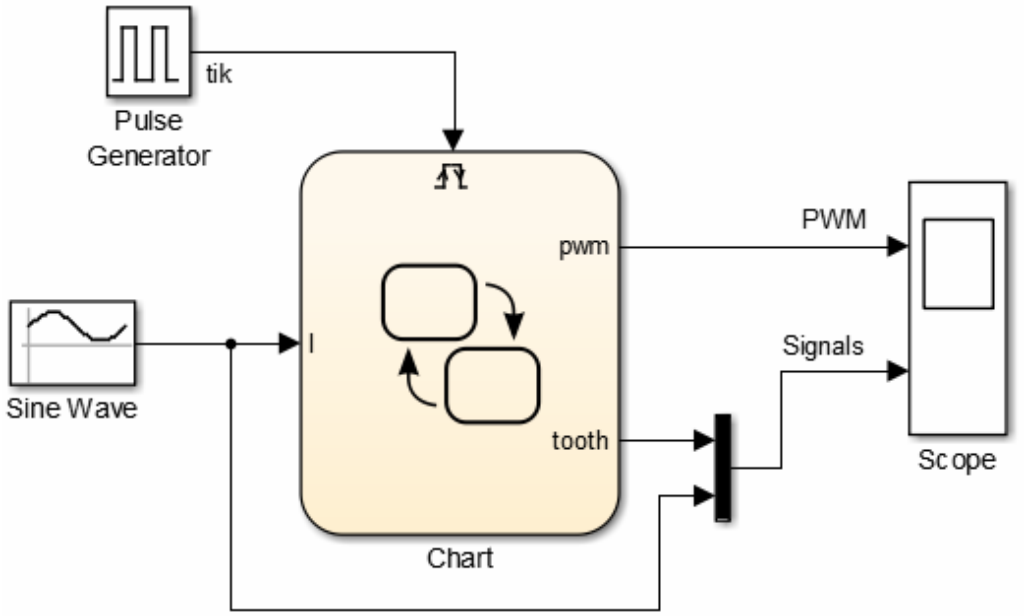

10. Проверьте состав модели, типы переменных и ее параметры при помощи проводника  Model Explorer.

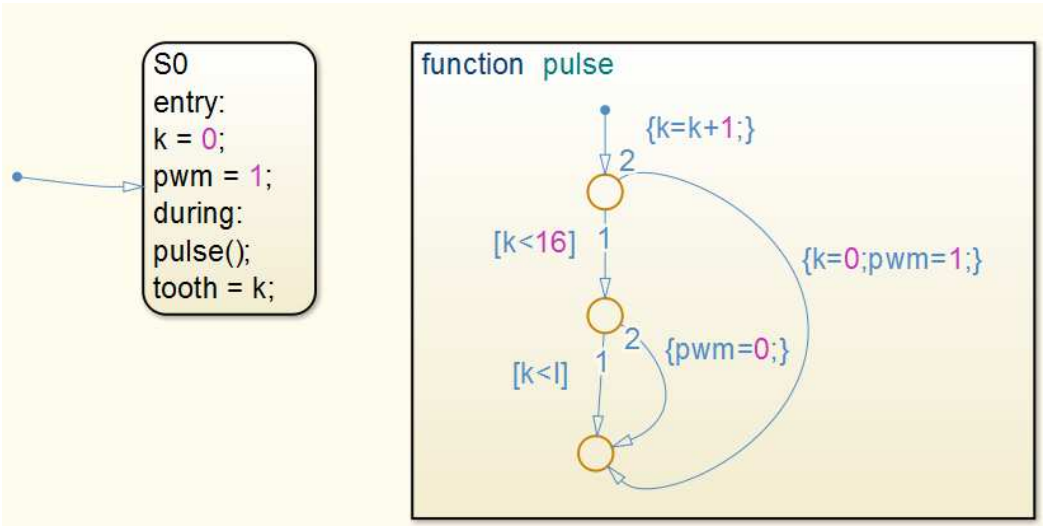


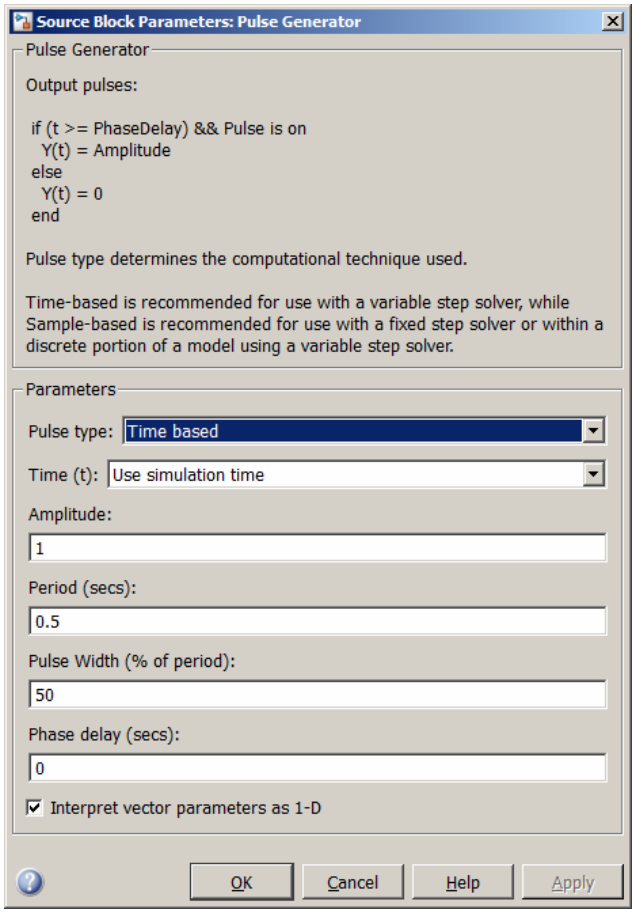

11. Запустите модель на выполнение. Сравните сигналы графопостроителя Scope с соответствующими сигналами модели задания 1.

**Задание 3.** Построение ШИП с применением графической функции Stateflow и тактированием от Simulink генератора.

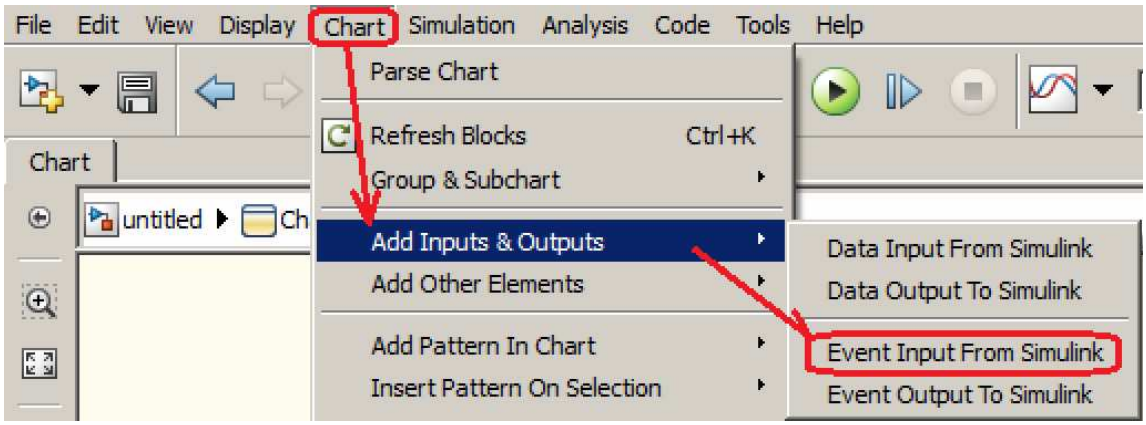
1. Используя параметры моделирования задания 1, параметры блоков и приёмы построения моделей предыдущих заданий постройте следующую модель ШИП.



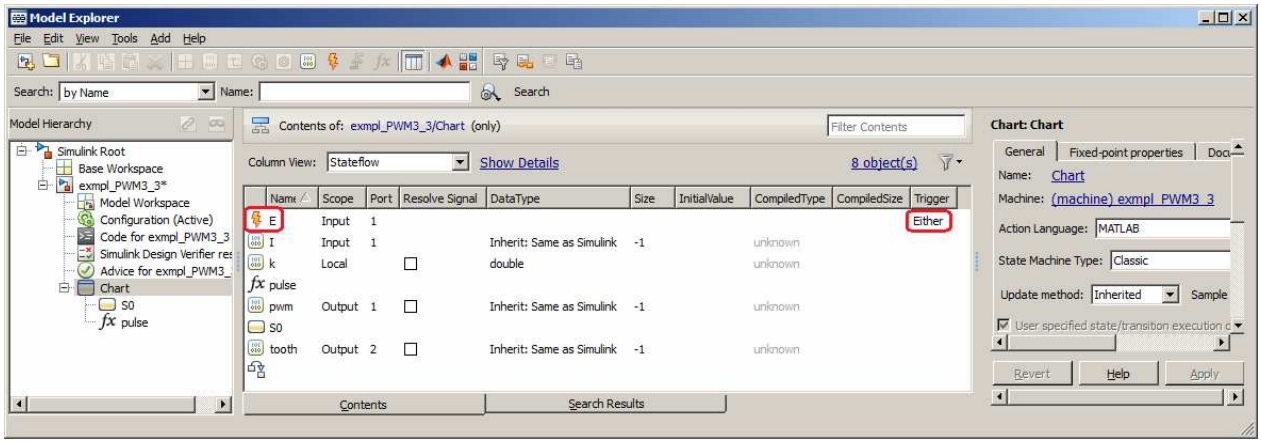






Порт подключения генератора импульсов (включение в диаграмму Chart события (Event) ) установите, как показано ниже.



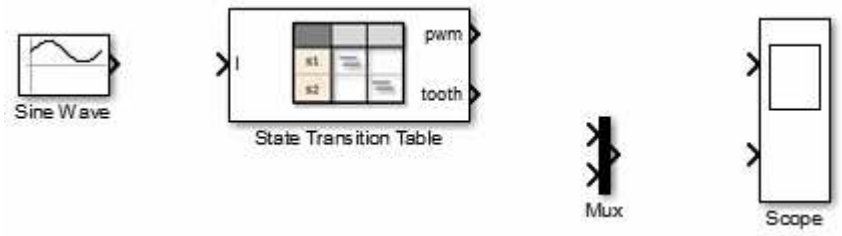

2. Проверьте состав модели, типы переменных и ее параметры при помощи проводника Model Explorer.



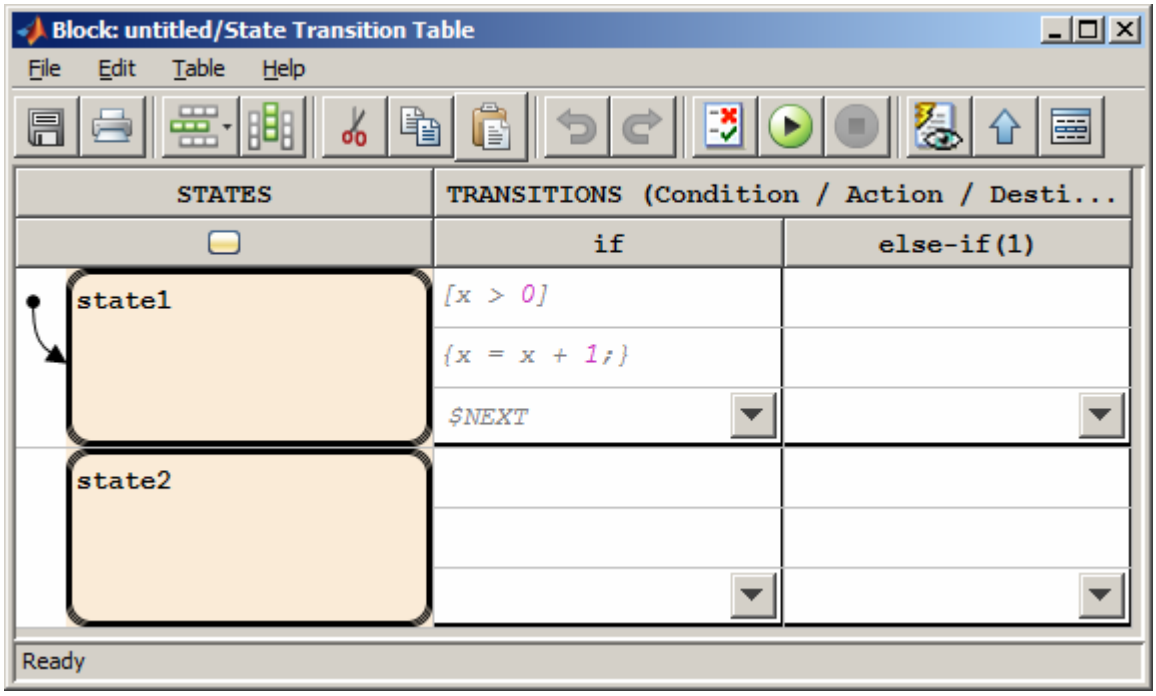

3. Запустите модель на выполнение. Сравните сигналы графопостроителя Scope с соответствующими сигналами модели задания 1 и задания 2.

**Задание 4.** Построение ШИП с применением таблицы переходов состояний (State Transition Table) Stateflow.

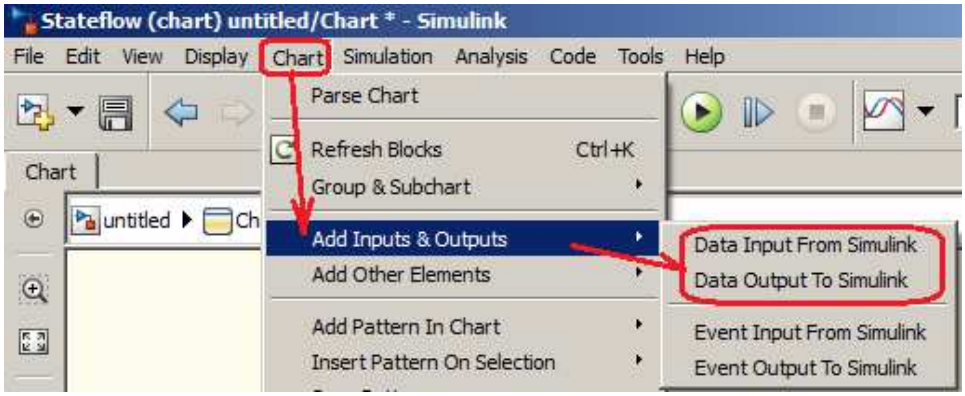
1. Откройте редактор построения Simulink моделей: MATLAB R2015 > Меню > New > Simulink Model.
2. Из библиотеки Simulink перетащите в окно моделирования следующие блоки.



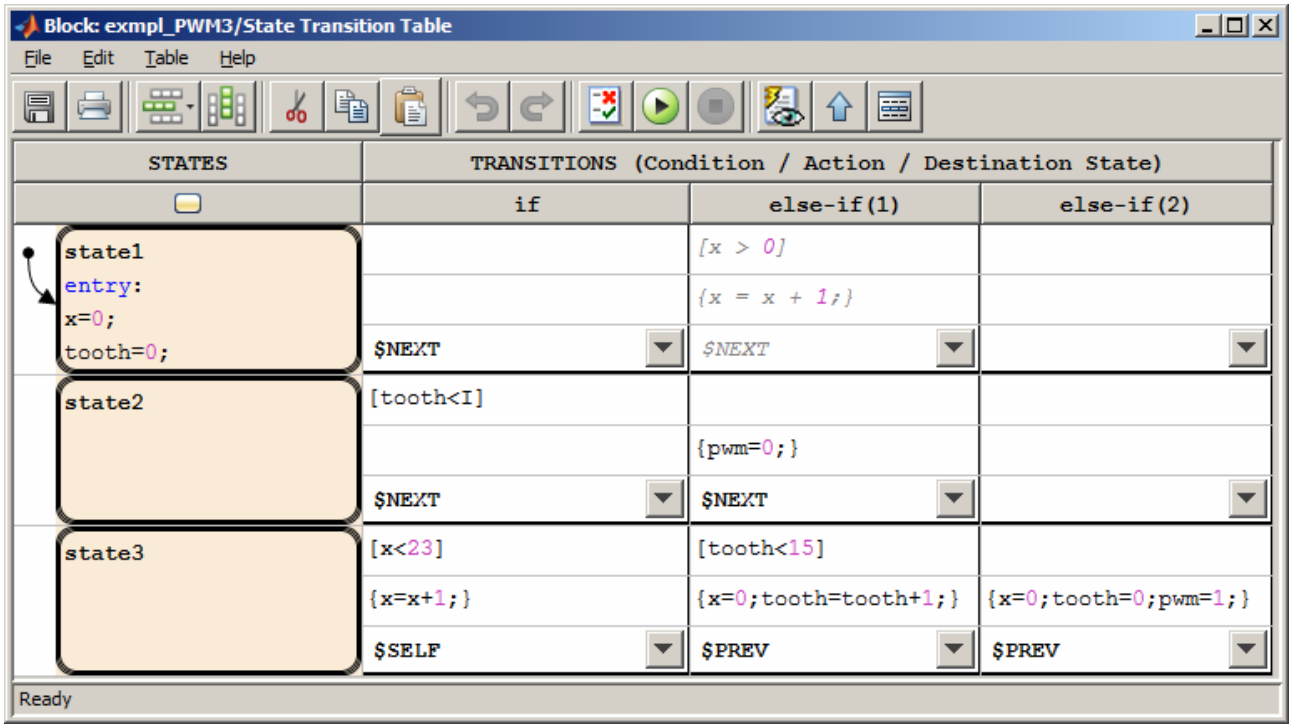

3. Установите режим моделирования и параметры Sine Wave такими же как в задании 1.
4. Откройте блок State Transition Table.




5. Откройте окно редактора диаграммы состояний кнопкой меню  и создайте порт ввода данных Simulink модели: I, и  два порта вывода: pwm и tooth.

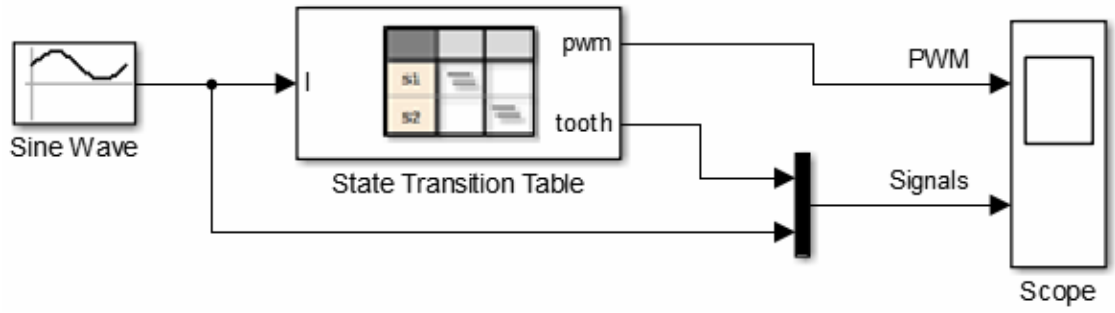



6. Добавляя в таблицу (п. 4) строки состояний  и колонки переходов  заполните ее функциональными связями, реализующими модель ШИП, например, как показано ниже.




7. Раскройте диаграмму состояния, соответствующую содержанию таблицы, нажав кнопку меню  .

8. Перейдите на уровень выше (  ) диаграммы состояний и соедините порты таблицы с соответствующими блоками Simulink модели.




9. Проверьте состав модели, типы переменных и ее параметры при помощи проводника  Model Explorer.

10. Запустите модель на выполнение. Сравните сигналы графопостроителя Scope с соответствующими сигналами модели задания 1, задания 2, и задания 3.

## КОНТРОЛЬНЫЕ ВОПРОСЫ

1. В чем отличие реализаций логических функций в Stateflow (задание 2, задание 3, задание 4) и Simulink (задание 1)?
2. Какими преимуществами обладает технология Stateflow?
3. Какими функциями MATLAB Simulink можно заменить графические объекты Stateflow?

## БИБЛИОГРАФИЧЕСКИЙ СПИСОК


1. Help MATLAB.
2. Dr. Bob Davidov. http://portalnp.ru/author/bobdavidov