



Тема 7. моделирование нелинейных систем (пакет SimuLink)

7.1. Общая характеристика пакета SimuLink

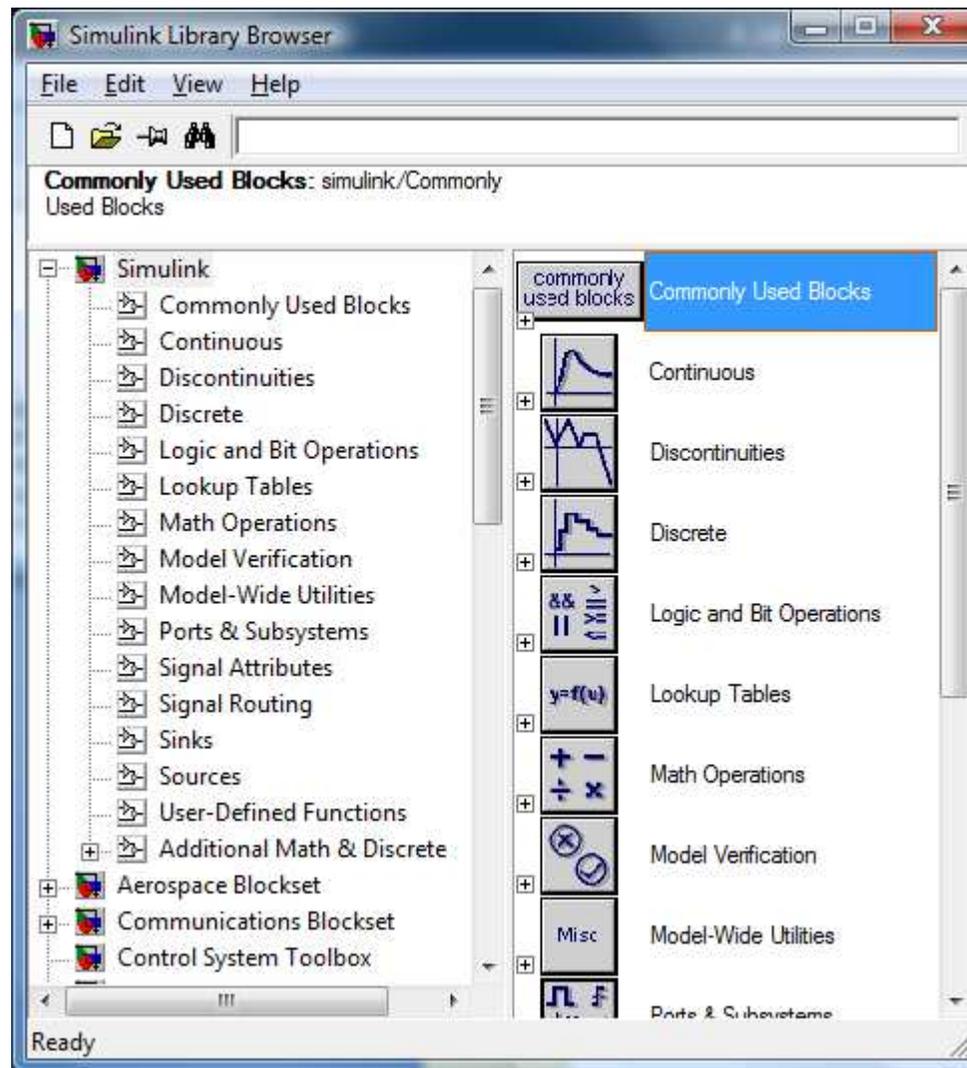
SimuLink – пакет визуального программирования.

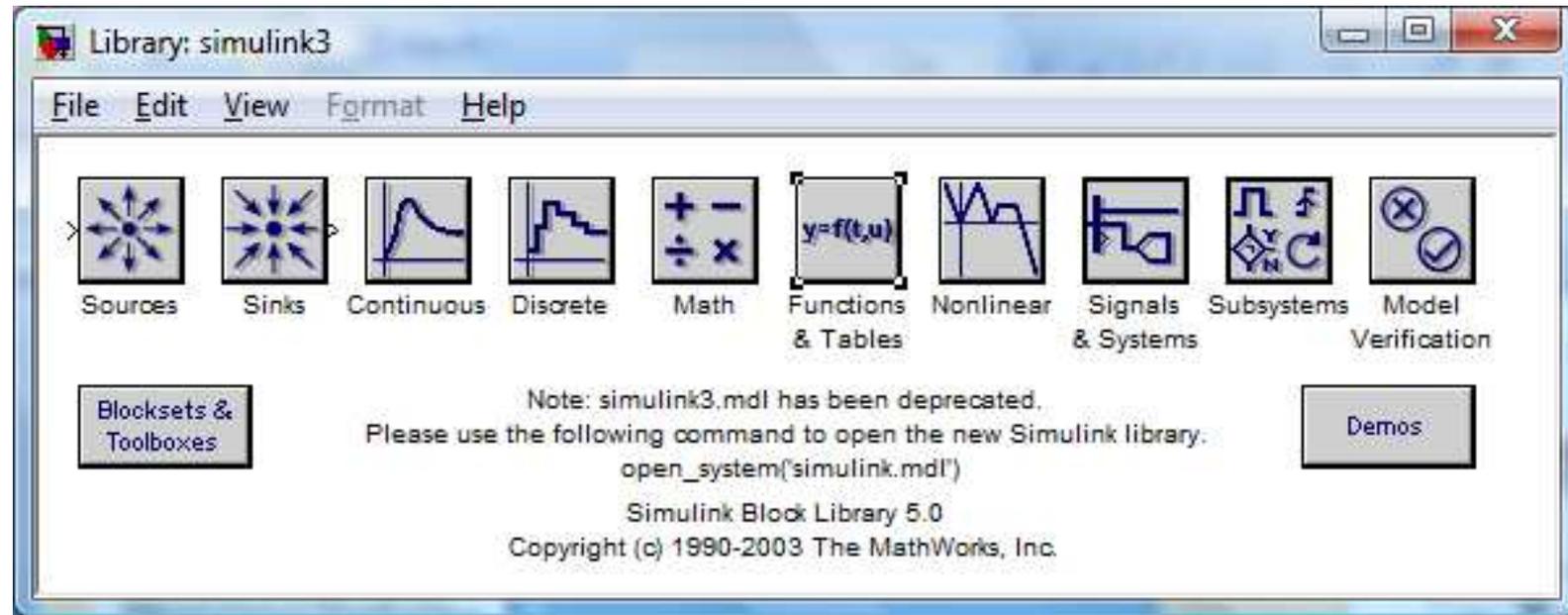
Позволяет осуществлять моделирование во времени поведения динамических нелинейных систем, причем введение характеристик исследуемых систем осуществляется путем *графической сборки схемы соединений элементарных звеньев*. В результате такого составления получается так называемая *S-модель*. Которая сохраняется в файле с расширением **.mdl**.

В качестве *элементарных звеньев* при построении *S-модели* используются визуальные блоки хранящиеся в библиотеках *SimuLink*. *S-модель* может иметь иерархическую структуру, т. е. состоять из моделей более низкого уровня, причем количество уровней иерархии практически не ограничено.

7.2. Библиотека блоков

Запустить *SimuLink* можно набрав команду *simulink* либо *simulink3* в командном окне MATLAB.

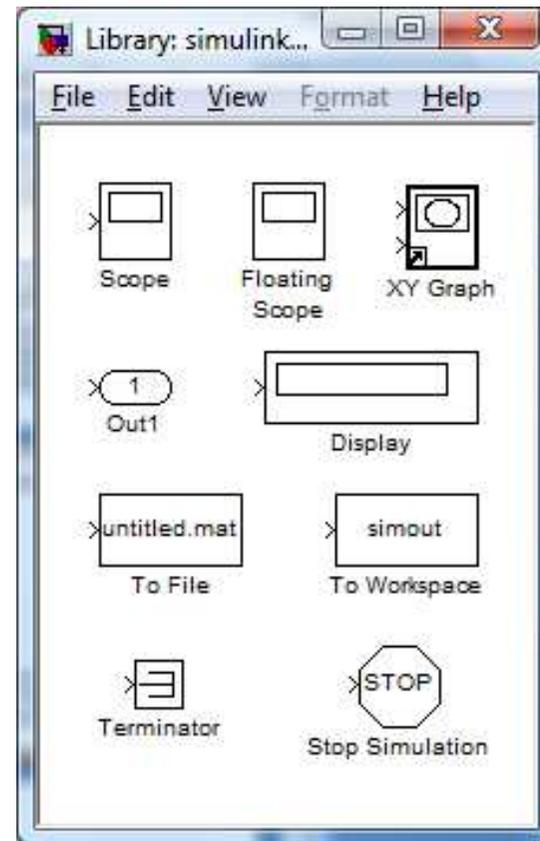




Эта библиотека состоит из 10 разделов. Восемь из них являются главными и не могут изменяться пользователем:

- Sources (Источники);**
- Sinks (Приемники);**
- Continuous (Непрерывные элементы);**
- Discrete (Дискретные элементы);**
- Math (Математические блоки);**
- Functions & Tables (Функции и таблицы)**
- Nonlinear (Нелинейные элементы);**
- Signals & Systems (Сигналы и системы).**

7.2.1. Раздел **Sinks** (приемники)



В этом разделе размещены три группы блоков, которые не имеют выходов, а ТОЛЬКО ВХОДЫ:



1) блоки, которые при моделировании играют роль обзорных окон:

- блок **Scope** с одним входом, который выводит в графическое окно график зависимости величины, подаваемой на его вход, от модельного времени;
- блок **XYGraph** с двумя входами – обеспечивает построение графика зависимости одной моделируемой величины (второй сверху вход) от другой (первый вход);
- блок **Display** с одним входом – отображает численные значения входной величины;

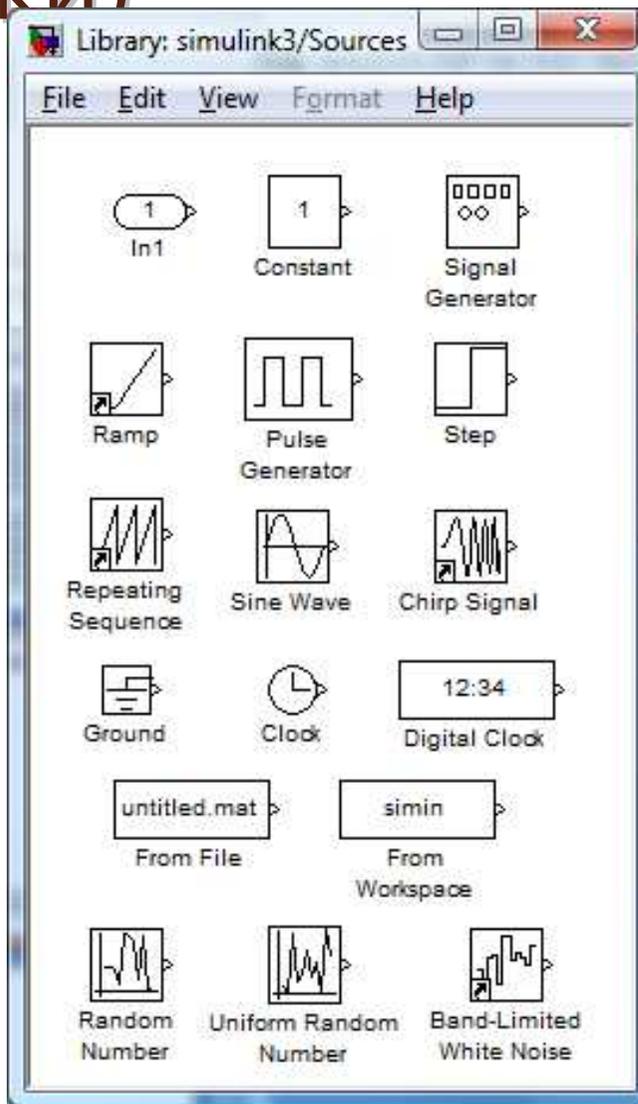
2) блоки для сохранения результатов:

- блок **To File**, который обеспечивает сохранение результатов моделирования на диске в MAT-файле (с расширением **.mat**);
- блок **To Workspace**, который сохраняет результаты в рабочем пространстве;

3) блоки управления моделированием:

- **Stop Simulation** – прерывает моделирование при выполнении тех или иных условий; блок срабатывает в том случае, когда на его вход поступает ненулевой сигнал.
- Блок **Terminator (Ограничитель)** может использоваться как "заглушка" для тех портов, которые по какой-либо причине оказались не подсоединенными к другим блокам S-модели. При этом блок **Terminator** используется как заглушка для выходных портов.

7.2.2. Раздел **Sources** (ИСТОЧНИКИ)



В этом разделе в качестве источников сигналов предусмотрены такие блоки:



Constant – формирует постоянную величину (скаляр, вектор или матрицу);

Signal Generator - создает (генерирует) непрерывный колебательный сигнал одной из волновых форм на выбор – синусоидальный, прямоугольный, треугольный или случайный;

Step - генерирует сигнал в виде одиночной ступеньки (ступенчатый сигнал) с заданными параметрами (начала ступеньки и ее высоты);

Ramp - создает линейно восходящий (или нисходящий) сигнал;

Sine Wave - генерирует гармонический сигнал;

Repeating Sequence - генерирует периодическую последовательность;

Pulse Generator - генератор непрерывных прямоугольных импульсов;

Chirp Signal - генератор гармонических колебаний с частотой, которая линейно изменяется с течением времени;

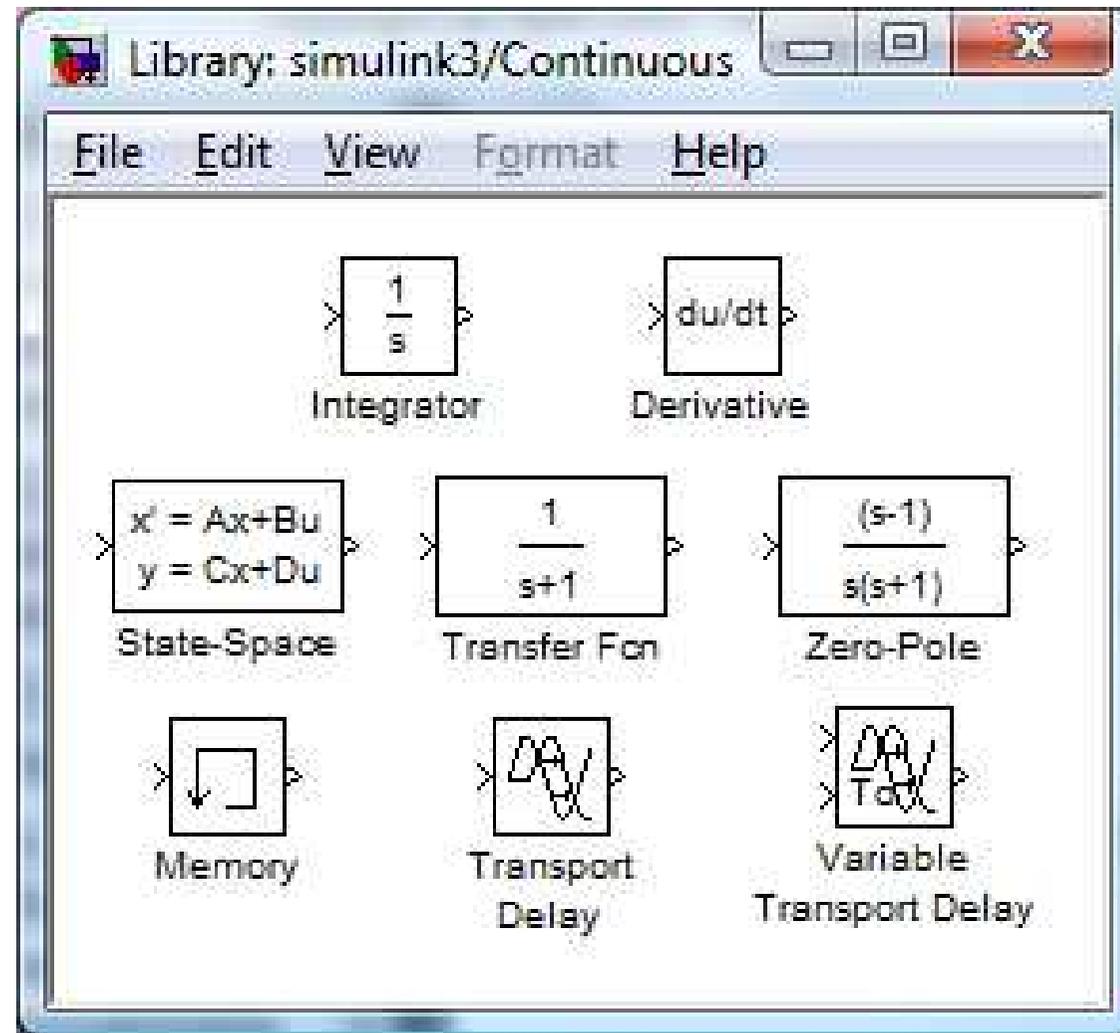
Clock (Часы) - источник непрерывного сигнала, пропорционального модельному времени;

Digital clock (Цифровые часы) - формирует дискретный сигнал, пропорциональный времени;

Random Number - источник дискретного сигнала, значения которого являются случайной величиной, распределенной по нормальному (гауссовому) закону;

Uniform Random Number - источник дискретного сигнала, значения которого являются случайной равномерно распределенной величиной;

7.2.3. Раздел Continuous





Этот раздел библиотеки содержит такие блоки:

Integrator - идеальное интегрирующее звено (интегратор);

Derivative - идеальное дифференцирующее звено;

State-Space - определение линейного звена через задание четырех матриц его пространства состояний;

Transfer Fcn - определение линейного звена через задание его передаточной функции;

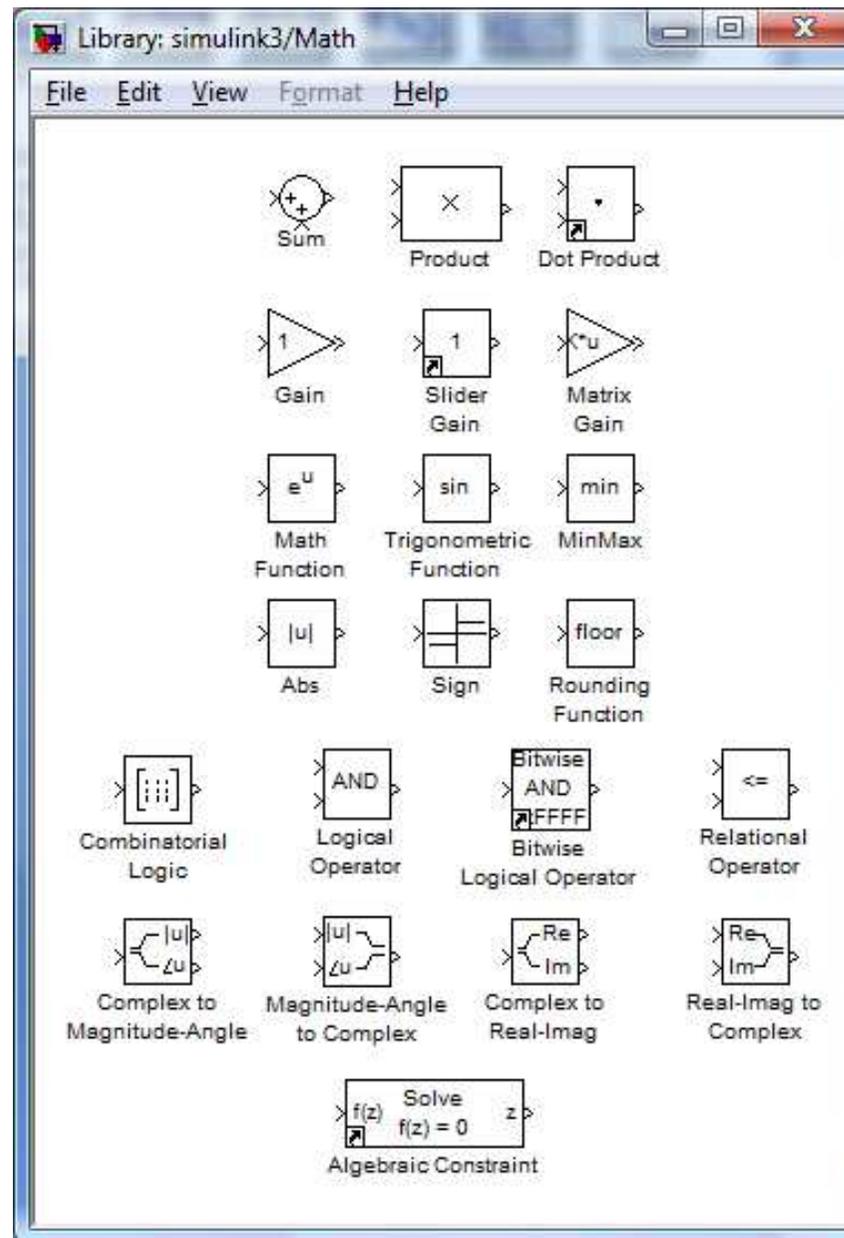
Zero-Pole - задание звена через указание векторов значений его полюсов и нулей, а также значения коэффициента передачи;

Memory (Память) выполняет задержку сигнала на один шаг модельного времени;

Transport Delay обеспечивает задержку сигнала на заданное количество шагов модельного времени, причем необязательно целое;

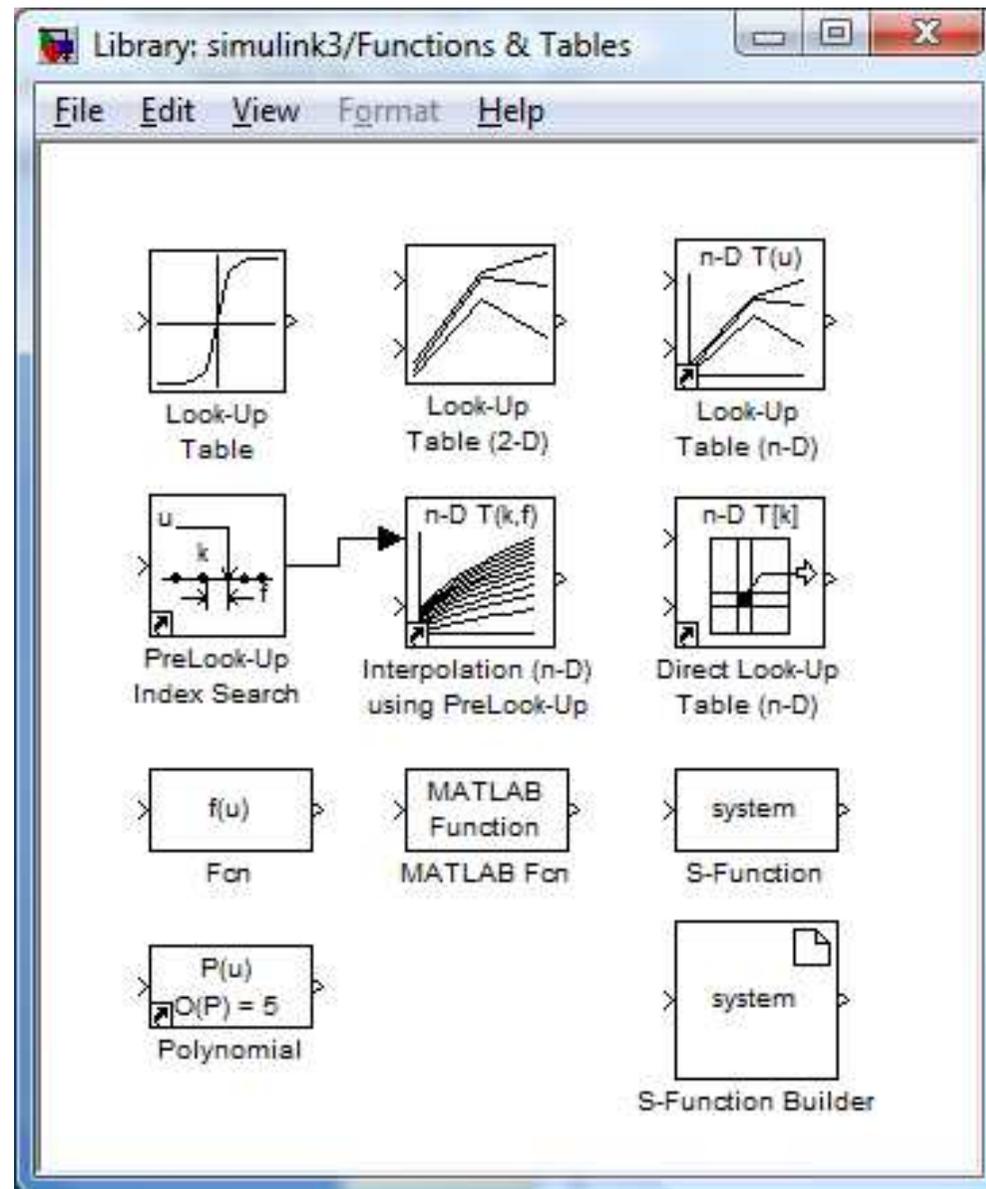
Variable Transport Delay позволяет задавать управляемую извне величину задержки.

7.2.4. Раздел Math



- 
- **Sum** – осуществляет суммирование сигналов, поступающих в него;
 - **Product** – выполняющий умножение или деление входных сигналов;
 - **Dot Product** – блок, осуществляющий перемножение двух входных величин, если они являются скалярами, или определяющий сумму поэлементных произведений элементов двух входных векторов (одинаковой длины);
 - **Gain** – линейное усилительное звено;
 - **Slider Gain** – звено интерактивного изменения коэффициента усиления;
 - **Matrix Gain** - матричное усилительное звено для многомерной системы;
 - шесть блоков математических стандартных операций (**MathFunction, TrigonometricFunction, MinMax, Abs, Sign u Rounding function**);
 - три блока логических операций (**Combinatorial Logic, LogicalOperator** и **Relation Operator**);
 - четыре блока трансформирования комплексных сигналов в действительные и наоборот (**Complex to Magnitude-Angle, Complex to Real-Imag, Magnitude-Angle to Complex u Real-Imag to Complex**);
 - **AlgebraicConstraint** – блок решения алгебраических уравнений.

7.2.5. Раздел Functions & Tables

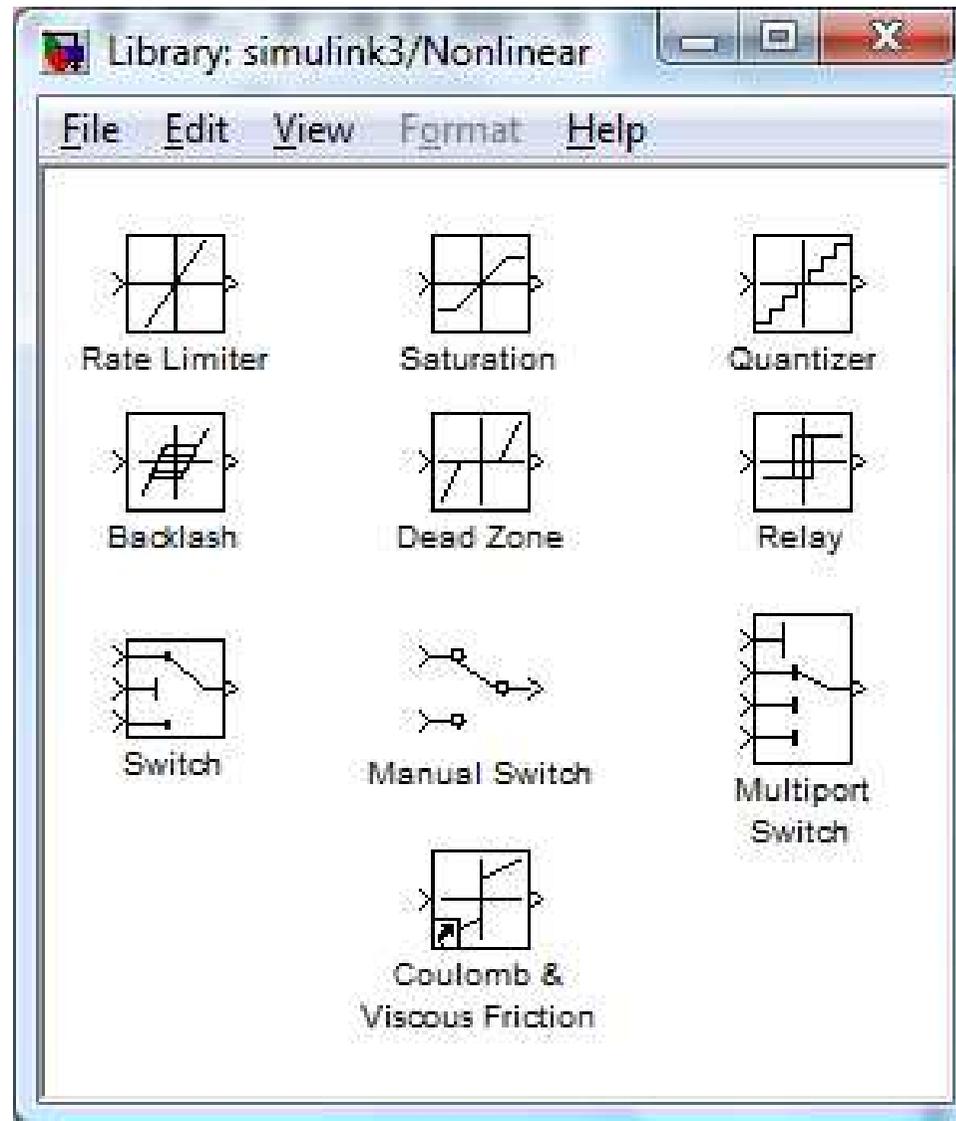




В этот раздел входят две группы блоков:

- шесть блоков, осуществляющие линейную интерполяцию значений выходного сигнала по значениям входного сигнала, которые поступают в блок, в соответствии с заданной в окне настраивания таблицы соответствий;
- четыре блока, которые являются заготовками, дающими пользователю возможность разработать собственные блоки программы, выполняющие необходимые ему преобразования сигналов.

7.2.6. Раздел **Nonlinear**



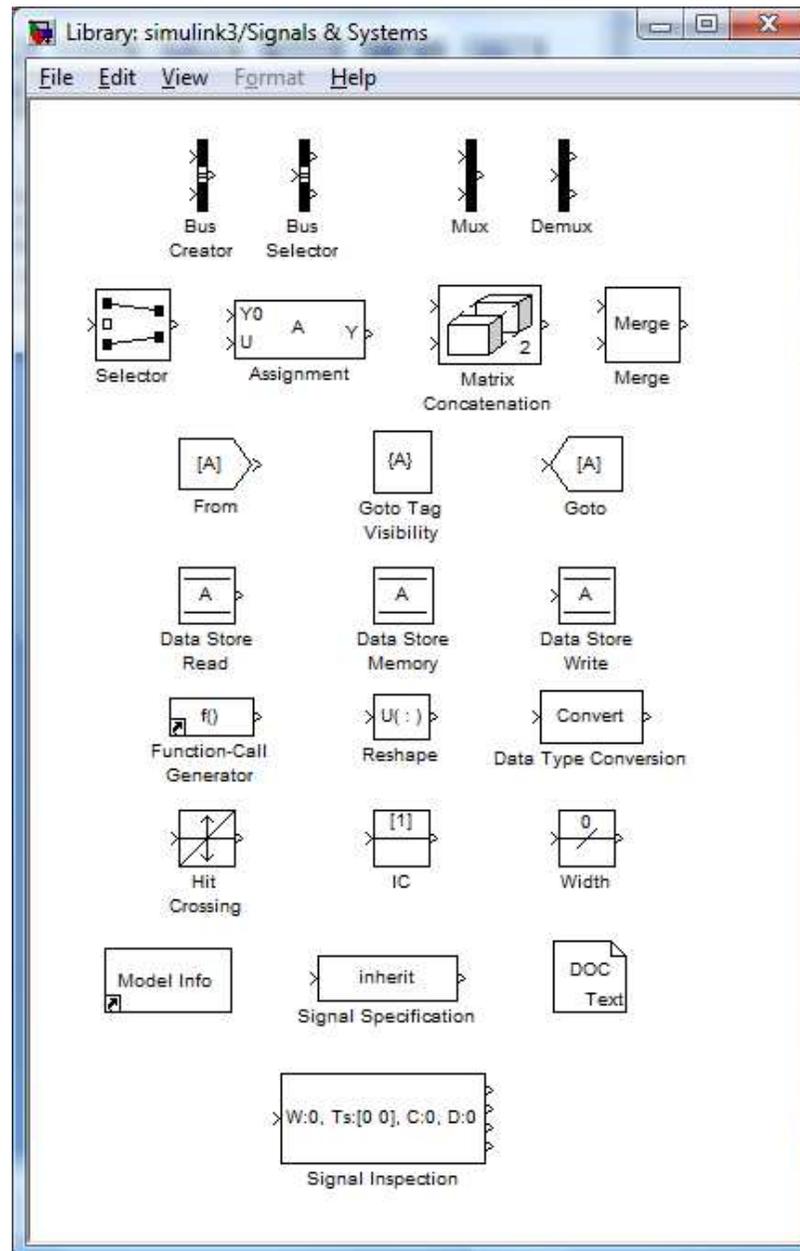


Здесь расположены 6 блоков, которые реализуют некоторые типовые нелинейные (кусочно-линейные) зависимости выходной величины от входной и 3 блока-переключателя.

- Блок ***Rate Limiter (Ограничитель скорости)*** обеспечивает ограничивание сверху и снизу скорости изменения сигнала, проходящего через него.
- Блок ***Saturation (Насыщение)*** реализует линейную зависимость с насыщением (ограничением).
- Блок ***Quantizer (Квантователь)*** осуществляет дискретизацию входного сигнала по величине.
- Блок ***BackLash (Люфт)*** реализует нелинейность типа зазора (люфта). В нем предусмотрено два параметра настраивания: *Deadband width* - величина люфта и *Initial output* - начальное значение выходной величины.
- Блок ***Dead Zone (Мертвая зона)*** реализует нелинейность типа зоны нечувствительности.
- Блок ***Relay (Реле)*** работает по аналогии с обычным реле: если входной сигнал превышает некоторое предельное значение, то на выходе блока формируется некоторый постоянный сигнал.

- 
- Блок ***Coulomb & Viscous Friction*** (Кулоново и вязкое трение) реализует нелинейную зависимость типа «сухого трения». Если вход положителен, то выход пропорционален входу с коэффициентом пропорциональности - "коэффициентом вязкого трения" - и увеличен на величину «сухого трения». Если вход отрицателен, то выход также пропорционален входу (с тем же коэффициентом пропорциональности) за вычетом величины "сухого трения". При входе равном нулю выход тоже равняется нулю.

7.2.7. Раздел Signals & Systems





Раздел ***Signals & Systems (сигналы и системы)*** предназначен для построения сложных S-моделей, которые состоят из других моделей более низкого уровня.

- Блоки ***In (Входной порт)*** и ***Out (Исходный порт)*** обеспечивают информационную связь между подсистемами модели и с рабочим пространством системы MatLAB.
- Блок ***Mux (Мультиплексор)*** выполняет объединение входных величин в единый выходной вектор. При этом входные величины могут быть как скалярными, так и векторными. Длина результирующего вектора равняется сумме длин всех векторов. Порядок элементов в векторе выхода определяется порядком входов (сверху вниз) и порядком расположения элементов внутри каждого входа.
- Блок ***Demux (Разделитель, Демультимплексор)*** выполняет обратную функцию - разделяет входной вектор на заданное количество компонентов. Он имеет единственный параметр настраивания *Number of outputs* (Количество выходов). В случае, когда указанное число выходов (N) задается меньшим длины входного вектора (M), блок формирует исходные векторы следующим образом. Первые (N-1) выходов будут векторами одинаковой длины, равной целой части отношения $M/(N-1)$. Последний выход будет иметь длину, равную остатку от деления.

- 
- Блоки ***From*** (***Принять от***) и ***Goto*** (***Передать к***) используются совместно и предназначены для обмена данными между разнообразными частями S-модели с учетом досягаемости (видимости) этих данных.
 - Блоки ***Data Store Read*** (***Чтение данных***), ***Data Store Memory*** (***Запоминание данных***) и ***Data Store Write*** (***Запись данных***) также используются совместно и обеспечивают не только передачу данных, но и их сохранение на протяжении моделирования.
 - Блок ***Selector*** выбирает в входном векторе и передает на выход только те элементы, номера которых указаны в одном из параметров настраивания блока.
 - Блок ***IC*** (***Initial Condition - начальное условие***) позволяет установить произвольное начальное значение входного сигнала.

- 
- Блок ***Width (Размер)*** определяет размерность сигнала, который поступает на его вход. Значения размерности выводятся непосредственно на изображении блока. Параметров настраивания блок не имеет.
 - Блок ***Merge (Слияния)*** выполняет объединение сигналов, которые поступают в его входы, в единый сигнал.
 - Блок ***Hit Crossing (Обнаружить пересечение)*** позволяет зафиксировать состояние, когда входной сигнал "пересекает" некоторое значение. При возникновении такой ситуации на выходе блока формируется единичный сигнал.

7.3. Создание подсистем

Использование подсистем дает следующие преимущества:

- сокращения количества блоков, которые выводятся в окне модели;
- объединение в единую группу (подсистему) функционально связанных блоков;
- возможность создания иерархических блок-схем.

Существуют две возможности создания подсистем:

- добавить блок ***Subsystem*** в модель, потом войти в этот блок и создать подсистему в возникшем окне подсистемы;
- выделить часть блок-схемы модели и объединить ее в подсистему.

Создание подсистемы через добавление блока **Subsystem**

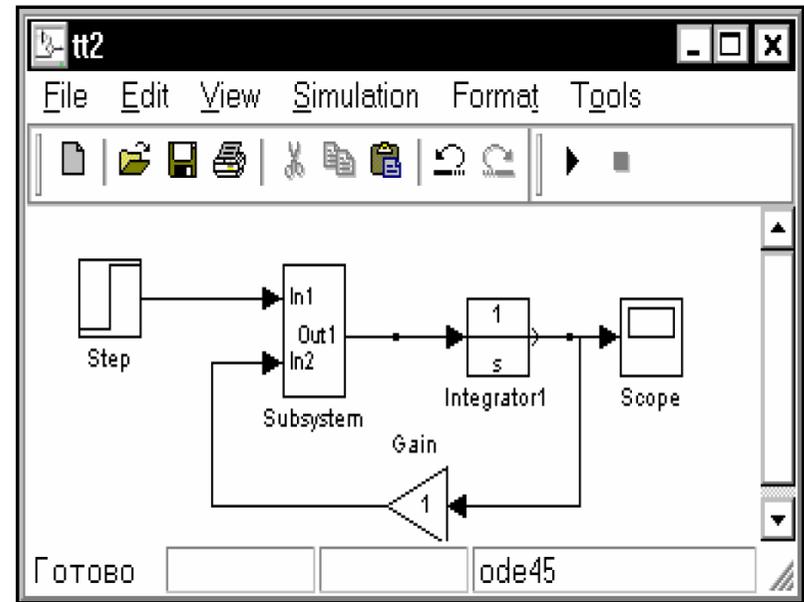
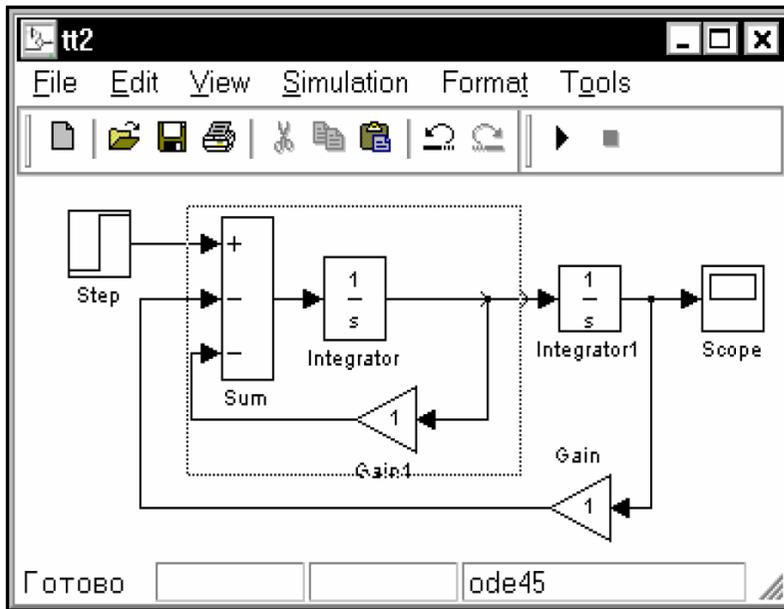
В этом случае следует поступить так:

- скопировать блок **Subsystem** в окно модели, перетянув его из библиотеки;
- раскрыть окно блока **Subsystem**, дважды щелкнув на изображении блока в блок-схеме;
- в пустом окне модели создать подсистему, используя блоки **In** и **Out** для создания входов и выходов подсистемы.

Создание подсистемы путем группирования существующих блоков

Если блок-схема уже содержит блоки, которые нужно объединить в подсистему, то последнюю можно образовать таким образом:

- выделить объединяющим боксом блоки и соединительные блоки, которые нужно включить в состав подсистемы;
- избрать команду *Create Subsystem* из меню *Edit* окна модели; при этом *SimuLINK* заменит выделенные блоки одним блоком *Subsystem*.



7.4. Процесс моделирования в MatLab

Моделирование состоит из двух фаз - *инициализации* и собственно *моделирования*.

В фазе *инициализации* осуществляются такие действия:

- блочные параметры передаются в MatLAB для оценивания (вычисления);
- каждая не условно выполняемая подсистема заменяется блоками, из которых она складывается;
- блоки сортируются в порядке, в котором их нужно изменять; на этом шаге выявляются алгебраические циклы;
- проверяются связи между блоками, чтобы гарантировать, что длина вектора выхода каждого блока совпадает с ожидаемой длиной векторов входа блоков, которые управляются ним.



Собственно моделирование осуществляется путем численного интегрирования. Каждый из имеющихся в наличии методов интегрирования (ODE) зависит от способности модели определять производные ее непрерывных состояний.

Вкладка *Solvers* содержит в верхней половине четыре окна:

Start time – в котором устанавливается начальное значение времени;

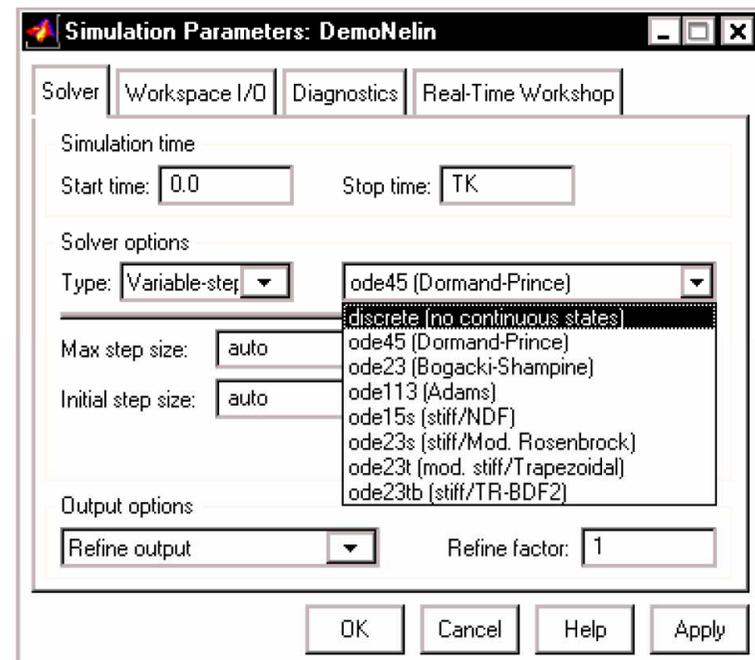
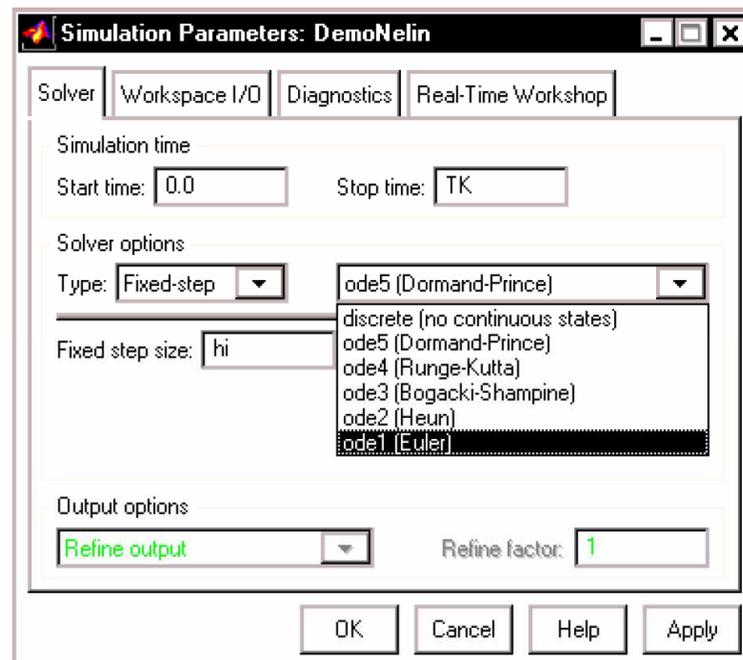
Stop time – где записывается конечное значение времени;

Type – где выбирается вид решателей;

Solver options – в котором выбирается конкретный решатель.

Если в окне *Type* выбран вид решателей *Fixed step* (с фиксированным шагом), в окне *Solver options* появляется такой набор решателей с фиксированным шагом:

- *discrete (no continuous states)* - дискретный (не непрерывные состояния);
- *ode5* – метод Дормана-Пренса (пятого порядка);
- *ode4* – метод Рунге-Кутты (четвертого порядка);
- *ode3* – метод Богацкого-Шампена (третьего порядка);
- *ode2* – метод Хойне (второго порядка);
- *ode1* – метод Ейлера (первого порядка).





Если же выбрать тип решателя *Variable step*, в окне *Solver options* возникнет другая подборка интеграторов (методов численного интегрирования):

- *ode45* – метод Дормана-Пренса с переменным шагом;
- *ode23* – метод Багацкого-Шампена с переменным шагом;
- *ode113* – метод Адамса с переменным шагом;
- *ode15s* – жесткий метод NDF с переменным шагом;
- *ode23s* – жесткий метод Розенброка с переменным шагом;
- *ode23t* – жесткий метод трапеций с переменным шагом;
- *ode23tb* – жесткий метод TR/BDF2 с переменным шагом.

В этом случае в нижней половине вкладки возникают другие окна:

- *Max step size* – максимальная величина шага;
- *Initial step size* - начальная величина шага;
- *Relative tolerance* – допустимая относительная погрешность;
- *Absolute tolerance* - допустимая абсолютная погрешность.