Simulink - интерактивный инструмент для моделирования, имитации и анализа динамических систем. Он дает возможность строить графические блок-диаграммы, имитировать динамические системы, исследовать работоспособность систем и совершенствовать проекты. Simulink полностью интегрирован с MATLAB, обеспечивая немедленным доступом к широкому спектру инструментов анализа и проектирования. Simulink также интегрируется с Stateflow для моделирования поведения, вызванного событиями. Эти преимущества делают Simulink наиболее популярным инструментом для проектирования систем управления и коммуникации, цифровой обработки и других приложений моделирования.

Simulink-функции (S-функции, S-functions) являются описанием блока на одном из языков программирования: MATLAB, C, C++, Ada, или Fortran. Набор стандартных блоков Simulink, достаточно обширен, однако в практике моделирования встречаются ситуации, когда нужного блока нет, либо структурное моделирование делает модель слишком сложной. В этом случае необходимо использовать технологию S-функций для создания нужного блока. С помощью языков программирования пользователь может создать описание сколь угодно сложного блока и подключить его к Simulink-модели, при этом с точки зрения взаимодействия пользователя с моделью, блок на основе S-функции ничем не отличается от стандартного библиотечного блока Simulink. Создаваемые блоки могут быть непрерывными, дискретными или гибридными. S-функции, созданные на C, C++, Ada или Fortran компилируются в исполняемые (\*.dll) файлы, за счет чего обеспечивается повышенная скорость выполнения таких блоков. Такие S-функции обладают еще и дополнительными возможностями, которые включают работу с разными типами данных (целыми, действительными и комплексными числами различной степени точности), использование матриц в качестве входных и выходных переменных (MATLAB S-функции могут оперировать только векторами в качестве входных и выходных переменных), а также больший набор внутренних функций (сallback-методов).

Чаще всего S-функции используются при создании новых библиотечных блоков, блоков, обеспечивающих взаимодействие Simulink с аппаратными средствами компьютера, при создании блоков на основе математических уравнений, блоков реализующих анимационные возможности MATLAB, а также при подключении к модели Simulink существующего программного кода языков высокого уровня.

Этапы моделирования

Процесс расчета модели выполняется Simulink в несколько этапов. На первом этапе выполняется инициализация модели: подключение библиотечных блоков к модели, определение размерностей сигналов, типов данных, величин шагов модельного времени, оценка параметров блоков, а также определяется порядок выполнения блоков и выполняется выделение памяти для проведения расчета. Затем Simulink начинает выполнять цикл моделирования. На каждом цикле моделирования (временном шаге) происходит расчет блоков в порядке, определенном на этапе инициализации. Для каждого блока, Simulink вызывает функции, которые вычисляют переменные состояния блока x, производные переменных состояния, и выходы y в течение текущего шага модельного времени. Этот процесс продолжается, пока моделирование не будет завершено. На рис. 16.3 показана диаграмма, иллюстрирующая этот процесс.

Основные переменные

Для того, чтобы создать **S-функцию**правильно необходимо определить основные понятия, используемые в технологии создания **S-функций**.К эти понятиям относятся:

* **Direct feedthrough –**Прямой проход. Проход входных сигналов на выход. Прямой проход реализуется в **S-функций**, если в выражениях для выходных переменных присутствуют входные переменные, либо при расчете времени следующего срабатывания блока также используются входные переменные. Установка правильного значения параметра **Direct feedthrough**очень важна, поскольку именно с помощь него **Simulink** определяет наличие в модели замкнутых алгебраических контуров.
* **Dynamically sized inputs –** Динамическая размерность входов. **S-функция**может быть написана таким образом, чтобы обеспечить произвольную размерность векторов входных и выходных переменных, а также векторов состояния непрерывной и(или) дискретной части системы. В этом случае фактическая размерность переменных определяется в самом начале процесса моделирования и устанавливается равной размерности входных сигналов. Чтобы задать динамическую размерность какой-либо переменной, нужно задать значение размерности для этой переменной равное -**1** (минус один) в соответствующем поле структуры **sizes** (см. приведенный ниже шаблон **S-функции**).
* **Setting sample times and offsets -**Установка шагов модельного времени и смещений. **S-функция** может задавать время срабатывания достаточно гибко. **Simulink** обеспечивает следующие варианты задания шага модельного времени:

1. **Continuous sample time**– Непрерывное модельное время. Задается для систем имеющих непрерывные переменные состояния. Для этого типа **S-функций** выходные переменные вычисляются на внутреннем шаге моделирования.
2. **•Continuous but fixed in minor time step sample time –**Непрерывное модельное время с фиксированным шагом во внутреннем цикле. Задается для **S-функций**, выходные переменные которых должны изменяться только в соответствии с внешним шагом моделирования, но должны быть неизменными на внутреннем.
3. **Discrete sample time –**Дискретное модельное время. Задается для дискретной системы (дискретной части системы). Пользователь должен задать шаг модельного времени **sample time** и смещение (задержку) **offset**, чтобы определить моменты времени, в которые **Simulink** должен вызвать на выполнение данный блок. Величина смещения не может превышать величину шага модельного времени. Время срабатывания блока определяется выражением:

**TimeHit = (n \* sample\_time) + offset**,где **n** – целое число шагов расчета.Если задано дискретное модельное время, то **Simulink** обращается к методам **mdlUpdate**и **mdlOutputs** на каждом внешнем шаге моделирования.

1. **Variable sample time** – Дискретный переменный шаг расчета. Модельное время дискретное, но интервалы времени между срабатываниями блока могут быть разными. В начале каждого шага моделирования **S-функция** должна определить значение времени следующего срабатывания. Для этого используется **mdlGetTimeOfNextVarHit**метод.
2. **Inherited sample time** – Наследуемый шаг расчета. В некоторых случаях работа блока не зависит от выбора варианта задания шага модельного времени. На пример, для блока **Gain** не имеет значения, какой шаг модельного времени реализован – блок выполняет усиление входного сигнала для любого варианта **sample time**. В подобных случаях параметр **sample time**может быть унаследован от предыдущего или последующего блока, либо от блока, имеющего наименьший шаг расчета.