DR. BOB DAVIDOV

Математическое моделирование систем управления в среде Simulink

Цель работы: освоение правил моделирования систем в среде Simulink

Задачи работы: простейшее моделирование объектов и систем управления в Simulink

Приборы и принадлежности: Персональный компьютер, среда МатЛАБ с пакетом Simulink.

ОБЩИЕ СВЕДЕНИЯ

СОЗДАНИЕ МОДЕЛЕЙ В SIMULINK

Simulink является приложением к пакету МатЛАБ и предназначен для моделирования линейных и нелинейных, дискретных и непрерывных систем которые могут работать как в режиме вычисления так и в режиме реального времени. В Simulink реализован принцип визуального программирования в соответствии с которым вся модель строится из стандартных библиотечных блоков и блоков модернизированных или разработанных пользователем. Каждый блок реализует свою математическую функцию. Блоки имеют входы и выходы и соединяются в модели линиями связи по которым аргументы поступают на вход функций.

Кроме секций с типовыми блоками библиотека Simulink имеет дополнительные разделы с блоками для разных областей применения, например, для моделирования электромеханических устройств, каналов передачи данных, и т.д.

Многие функции (команды m-файлов) и инструменты МатЛАБ могут использоваться в среде Simulink.

Для запуска Simulink необходимо предварительно запустить МатЛАБ, а затем нажать на кнопку **(a)** или ввести в командной строке Simulink:

AMATLAB R2012a	
File Edit Debug Parallel Desktop Window Help	
: 🛅 📁 🔏 ங 🛍 🤊 (* 🚺 🗊 🗐 🖉 Current Fo	older: C:\tmp_MatLAB_Optimisation 🛛 🖌 🗈
Shortcuts 🗷 How to Add 🛛 What's New	
Current Folder	Command Window
E Optimisation	New to MATLAB? Watch this <u>Video</u> , see <u>Demos</u> , or read <u>Getting Started</u> ,
Name 🔺	>> simulink

Для создания новой модели необходимо

- 1. открыть окно модели,
- 2. перенести в окно необходимые блоки из Simulink библиотеки и
- 3. соединить блоки линиями связи.

Окно новой модели открывается:

		AMATLAB R2012a	
		File) Edit Debug Parallel Desktop Window	Help
Simulials Library Province		New Scrip	ipt Ctrl+N
Simulink Library browser		Open Ctrl+O Fun	nction
File) Edit View Help	🐱 Simulink Library Browser	Close Command Window Ctrl+W Clas	55
		Import Data Enu	Imeration
Model Ctrl+N	File Edit View Help	Save Workspace Ctrl+S Syst	tem Object
🚰 Open Ctrl+O Library		Figu	ure
Close Ctrl+W	🕻 🗋 🚘 🛛 🚿 Enter search term	Set Path Vari	iable
		Preferences (Mod	del
Preferences ИЛИ	Libraries ИЛИ	Page Setup	I

Для переноса в окно модели необходимого блока необходимо найти блок в библиотеке нажать на блок левой кнопкой мыши (ЛКМ) и перетащить блок в окно модели:

🖬 untitled * 🛛 🗖 🔀	🗑 🤅 🖬 🐨	
File Edit View Simulation Format Tools Help	File Edit View Help	
D 😂 🖬 🚳 👗 🖻 🖻 🗇 ↔ ☆ 🗠 으 🕒 🕨 🔳 10.0 Normal 📃	🗋 😅 \Rightarrow Enter search term 💽 🚧 📺	
	Libraries	Library: Simulink/Sources Search Results: (none) Most F
-	Commonly Used Blocks	Band-Limited Chirp Signal White Noise
Step	Discontinuities Discrete Logic and Bit Operations	Counter Free- Running Counter Limited
	Lookup Tables	IN THE Simile From Workspace
Ready 100% ode45	Model vernication Model-Vikide Utilities Ports & Subsystems	Pulse Generator Ramp
	Signal Attributes Signal Routing Sinke	Repeating Seq. Repeating uence Interpol Sequence Stair
	-Sources User-Defined Functions	Sine Wave Step
	B - Additional Math & Discrete B - M Aerospace Blockset M Arduino IO Library	

Чтобы скопировать блок (или выделенные блоки) надо перетащить его при нажатой правой кнопке мыши (ПКМ).

Для изменения параметров блока надо дважды щелкнуть на блоке и в открывшемся диалоговом окне изменить соответствующие параметры, например, параметры блока Step:

📓 Source Block Parameters: Step 🛛 🔀
Step
Output a step.
Parameters
Step time:
Initial value:
0
Final value:
1
Sample time:
0
☑ Interpret vector parameters as 1-D
Enable zero-crossing detection
OK <u>C</u> ancel <u>H</u> elp Apply

Чтобы изменить название блока надо щелкнуть по названию блока ЛКМ и отредактировать текст.

С выделенными блоками можно выполнять следующие действия

- изменить положение и размеры блока,
- изменить цвет контура и самого блока,
- изменить ориентацию блока,
- выровнять положение блоков,
- создать из нескольких блоков подгруппу (subsystem) которая отображается одним блоком.

Примечание: Команды для работы с блоками находятся в разделе меню > Format окна модели

Для соединения блоков можно протянуть мышкой линию связи между нужными выходом (входом) и входом (выходом) или щелкнуть ЛКМ по блоку с нужным выходом, затем нажать на клавишу клавиатуры Ptrl и щелкнуть ЛКМ по блоку с нужным входом.

Чтобы подать один сигнал на два блока (сделать «развилку»), надо сначала создать одну линию обычным способом. Для создания второй линии, следует нажать ПКМ на точку предполагаемой развилки и протащить линию ко второму блоку или ЛКМ протянуть линию от входа нужного блока к точке развилки.



Для выделения объекта модели (блока или соединительной линии) надо щелкнуть по нему ЛКМ. Для удаления объекта или группы необходимо сначала их выделить щелчком или оконтуриванием, а затем удалить командой Delete - связанной с объектом или группой объектов.

Пользователь может подобрать оптимальные для конкретной модели численные методы и шаг интегрирования (Рис. 1).

Select:	Simulation time								
Solver	Start time: 0.0	Start time: 0.0							
Data Import/Export Optimization Diagnostics									
-Hardware Implementation	Type: Va	riable-step 🛛 😽	Solver:	ode45 (Dormand-Prince)	~				
Simulation Target	Max step size: au	to	Relative tolerance:	1e-3					
Code Generation	Min step size: au	to	Absolute tolerance:	auto					
±-HDL Code Generation	Initial step size: au	to	Shape preservation:	Disable All	~				
	Number of consecut	Number of consecutive min steps: 1.							
	Tasking mode for pe	riodic sample times: ndle rate transition for data trans lue indicates higher task priority	Auto						
		15							
	Zero-crossing option								
	Zero-crossing option	1: Use local settings	Algorithm:	Nonadaptive	~				
	Zero-crossing option Zero-crossing contro Time tolerance:	il: Use local settings 10*128*eps	Algorithm: Signal threshold	Nonadaptive	~				
	Zero-crossing option Zero-crossing contro Time tolerance: Number of consecut	ve zero crossings:	Algorithm: Signal threshold	Nonadaptive auto 1000					

Рис. 1. Окно выбора параметров моделирования: шага интегрирования, метода моделирования, допусков и т.д. Путь: меню модели > Simulation > Ponfiguration Parameters (Ptrl+E) > **Solver**.

Входные сигналы (раздел библиотеки Sources) и переменные состояния модели можно наблюдать в процессе моделирования при помощи блоков раздела (Sinks). Например, блок Scope в зависимости от его настроек показывает графики изменения входных сигналов как в одном окне, так и в нескольких, отображая заданное число точек. Масштабы графиков можно изменять, а их данные - сохранять в рабочей области МатЛАБ (workspace).

Время	моделирования	задается	в	окне	⊇ ⊆)	= (10.0	Normal	Для	установки
бесконе	чного времени м	моделирова	ния	в соот	гветствующее	окно	необходимо	ввести	значение
inf									

Для запуска моделирования необходимо щелкнуть по кнопке Start simulation на панели инструментов окна модели. Кнопки и останавливают моделирование.

Копирование модели в буфер обмена виде растрового рисунка выполняется командой меню окна модели > Edit – P opy Model to Plipboard. Предварительно лучше уменьшить размеры окна модели до минимальных, чтобы не было белых полей:



Рис. 2. Пример простейшей модели и ее реакция. Модель включает источник ступенчатого воздействия, сумматор, интегратор охваченный отрицательной обратной связью и осциллограф для наблюдения за реакцией системы.

Библиотека Simulink содержит следующие основные разделы:

Libraries 🗄 😈 Simulink Commonly Used Blocks Continuous -Discontinuities -Discrete "Logic and Bit Operations -Lookup Tables -Math Operations -Model Verification Model-Wide Utilities Ports & Subsystems Signal Attributes Signal Routing Sinks Sources User-Defined Functions É-Additional Math & Discrete 🔙 Aerospace Blockset 🔂 Arduino IO Library 🙀 Arduino Target 🗄 👦 Communications System Toolbox 🗄 👦 Computer Vision System Toolbox 🙀 Control System Toolbox 🗄 - 🔂 DSP System Toolbox 🖬 Data Acquisition Toolbox 🗄 🛺 Embedded Coder 🗄 🐻 Fuzzy Logic Toolbox 🗄 🐻 Gauges Blockset 🗄 - 🚺 HDL Verifier 🙀 Image Acquisition Toolbox 🔂 Instrument Control Toolbox Model Predictive Control Toolbox 🗄 - 🔂 Neural Network Toolbox • 🖬 OPC Toolbox 뒂 Real-Time Windows Target 🔂 Report Generator 🔂 Robust Control Toolbox 🗄 🐻 SimEvents 🗄 🙀 SimRF - 🙀 Simscape 🖬 Simulink 3D Animation - 🖬 Simulink Coder 🔙 Simulink Control Design 🙀 Simulink Design Optimization 🗄 🐻 Simulink Design Verifier 🗄 - 🔂 Simulink Extras 📓 Simulink Verification and Validation 😼 Stateflow 駴 System Identification Toolbox 🔂 Vehicle Network Toolbox 🚺 xPC Target

- Pommonly Used Blocks наиболее часто используемые блоки разных разделов библиотеки (воздействия, сумматор, интегратор, усилитель, осциллограф и т.д.)
- Pontinuous библиотека непрерывных элементов (интегратор, дифференциатор, линейная система ОДУ и т.д.);
- Discontinuities нелинейные элементы (насыщение, реле, переключатель и т.д.);
- Discrete библиотека дискретных элементов (интегратор, фильтр, задержка и т.д.);
- Logic and Bit Operations логические операции (сравнение, сдвиг, инверсия, AND и т.д.);
- Lookup Tables блоки задания в табличной форме функции переменных с использованием линейной интерполяции (функции одной переменной, двух переменных и т.д.);
- Math Operations математические функции (абсолютное значение, комбинаторная логика, выделение вещественной и мнимой составляющей комплексного числа и т.д.);
- Model Verification блоки проверки проектируемой системы на соответствие требованиям в частотной и временной областях (границы значения сигнала, ограничения по амплитуде ЛАЧХ, ограничения на отклик на единичное ступенчатое воздействие, запасы устойчивости по амплитуде и фазе и т.д.);
- Model-Wide Utilities библиотека дополнительных утилит (описание модели, информация о типе данных используемыми блоками Simulink и т.д.);
- Ports & Subsystems блоки подсистем (порты, синхронизация подсистем, вызов функций и т.д.);
- Signal Routing сигналы и системы (составной блок, входной сигнал, выходной сигнал, мультиплексор, демультиплексор и т.д.);
- Sinks средства отображения (временная диаграмма, вывод результатов в файл, остановка выполнения модели и т.д.);
- Sources источники сигналов (генератор импульсных/синусоидальных сигналов, генератор

случайных чисел, генератор пилообразных сигналов, часы и т.д.).

- User-Defined Function функции пользователя (S-функции, m-функции, табличные функции и т.д.);
- Additional Math & Discrete дополнительные математические и дискретные функции (задержки, инкрементирование и т.д.);

Помимо Simulink имеется целый ряд специальных библиотек для расчетов, моделирования систем, обмена данными, управления ресурсами компьютера и др. (см. рис. выше).

ОФОРМЛЕНИЕ ГРАФИКОВ В МАТЛАБ

Для графического отображения переменных состояния Simulink модели сохраненных в рабочей области (workspace) можно использовать следующие функции МатЛАБ.

Команда, параметр	Описание
figure; figure(N рисунка: 1,2,);	Открытие окна графика
plot(y);	Построение графика переменной у
plot(t, y);	Построение графика переменной у(t)
plot(t, y, 'xb', i, z, 'g', 'linewidth',2);	Построение голубого ('b') графика у(t) из х точек и зеленого ('g') графика z(i) линией двойной толщины (, 'linewidth',2) в одной зоне окна.
	Цветовое обозначение линий графиков:
	b синий (blue)
	g зеленый (green)
	г красный (red)
	с голубой (cyan)
	т фиолетовый (magenta)
	у желтый (yellow)
	k черный (black)
	Обозначение типа линии:
	– сплошная
	: точечная
	штрих-пунктирная
	штриховая
	х – обозначение точек массива

	Обозначение толщины линии:
	'linewidth',3
	'linewidth',2
subplot(N строки, N столбца, N активного графика);	Активизация одной из зон окна для построения в ней графика
title('xxx')	ххх заголовок графика
xlabel('xxx')	ххх название оси абсцисс
ylabel('xxx')	ххх название оси ординат
legend('название1',' название2',' название3')	вывод типа линий графиков и их названий в отдельный список окна графиков.

Вот пример кода т-файла МатЛАБ и его график.

```
if 1
    figure
    plot(time, depth, 'b', 'linewidth',2);
    hold on
    plot(time, set_depth, 'c', 'linewidth',2);
    hold on
    plot(time, valves, 'm', 'linewidth',1);
    grid on
    legend('depth','target depth','valves');
    xlabel('Time, sec');
    ylabel('Dept [m], Target dept [m], Valves [bit]' );
    title('Depth and Valve against Time');
end
```



Рис. 3. Пример построения графиков средствами МатЛАБ.

ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

Задание 1. Построение зависимости выхода от входа стандартного блока библиотеки Simulink.

- 1. Запустите МатЛАБ
- 2. Настройте МатЛАБ на свой рабочий каталог, например, tmp_Simulink.

📣 MATLAB R2012a		
File Edit View Debug F	arallel Desktop Window	Help
፡ 🛅 🗃 👗 🐂 🛍 🦻	0 🔁 🛅 🐔 🤊 🕈	Current Folder: D:\tmp_Simulink
Shortcuts 💽 How to Add	🛃 What's New	
Current Folder	s 🗆 🕂	× Command Window
🛅 ► D: ► tmp_Simulink	- 🔎 🖻 🤅)- >>

3. Откройте новое окно (файл модели) для построения модели.



 Командой меню > File > Save As переименуйте файл модели untitled, например, в Example_1.



5. Клавишей 🖬 окна модели откройте библиотеку Simulink.



6. Перенесите из библиотеки Simulink в окно модели следующие блоки.



Примечание: нужный блок в библиотеке можно найти введя имя блока в окне поиска, например,

🐱 Simulink Library Browser									
<u>File E</u> dit <u>V</u> iew <u>H</u> elp									
🗅 🖻	~	M 📺							
Libraries		Library: Simulink	/Commonly Use	d Blocks	(Found: 'mux')	Most Frequent	ly Used Blocks		
🕂 🙀 Simulink	<u>^</u> :	Simulink			4				
···Commonly Used Blocks ···Continuous ···Discontinuities		₹ •	emux	, ,	Mux	×	Demux	}	Mux
Discrete									

7. Соедините блоки как показано ниже.



8. Откройте блок Scope

🛃 Scope				
860]≗ ⊉			Р
5				
0				
- T				
-5 [ii		 10
U		4	ь	TU
Time offset:)			

9. Запустите модель. Наблюдайте результат моделирования в окне Scope.



10. Командой 🖾 блока Scope разместите графики во всей зоне окна.



- 11. Командой 🌇 блока Scope зафиксируйте размеры окна для будущих графиков.
- 12. Открыв окно с параметрами реле установите требуемые параметры, например, как показано ниже.

🐱 Function Block Parameters: Relay	×
Relay	
Output the specified 'on' or 'off' value by comparing the input to the specified thresholds. The on/off state of the relay is not affected by input between the upper and lower limits.	
Main Signal Attributes	
Switch on point:	
0.5	
Switch off point:	
-0.5	
Output when on:	
1	
Output when off:	
-1	
Input processing: Elements as channels (sample based)	~
Enable zero-crossing detection	
Sample time (-1 for inherited):	
-1	
OK Cancel Help Apply	

13. Задайте параметры выходного сигнала блока развертки (Repeating Sequence) так, чтобы он перекрывал точки переключения реле (-0.5 и 0.5).

🖬 Source Block Parameters: Repeating Sequence 🛛 🔀				
Repeating table (mask) (link)				
Output a repeating sequence of numbers specified in a table of time- value pairs. Values of time should be monotonically increasing.				
Parameters				
Time values:				
[0 2 4]				
Output values:				
[-1 1 -1]				
<u>QK</u> <u>C</u> ancel <u>H</u> elp <u>A</u> pply				

14. Постройте графики модели.



Графики показывают, что реле переключается в 1 когда входной сигнал превышает 0.5 и переключается в -1 когда входной сигнал становится меньше -0.5.

- 15. Для уменьшения искажений входного сигнала (пик желтого сигнала отображается полочкой) замените (в окне меню модели > Simulation > Solver) переменный шаг моделирования "по умолчанию" Туре: Variable-step и на постоянный : Туре: Fixed-step
- 16. Запустите модель и наблюдайте изменения на графиках сигнала реле.



Графики показывают, что исчезло искажение пика входного сигнала, но теперь сместились точки переключения реле.

- 17. Для уменьшения искажений в точках переключения реле замените шаг моделирования Fixed-step size (fundamental sample time): Auto (в окне меню модели > Simulation > Solver), например, на 0.01
- 18. Постройте графики реле с новыми значениями параметров моделирования.



Задание 2. Построение модели скачущего мяча.

Описание процесса.

На заданной высоте, например, 1м отпускают мяч весом 100 г. который летит до поверхности, и отскакивает от нее на высоту ~80% от начальной точки. Необходимо построить модель полета и отскоков мяча и определить основные параметры модели.

Составим список отношений которые, как нам кажется, влияют на поведение мяча.

- Сила гравитации равно произведению массы тела на ускорение свободного падения
- Ускорение тела равно отношению сил действующих на тело к массе тела
- Изменение скорости равно интегралу от ускорения.
- Изменение расстояния равно интегралу от скорости.
- 1. В новом окне наберите модель полета мяча состоящую из блоков реализующих исходные отношения.



 Выберите следующие параметры моделирования (меню модели > Simulation > Ponfiguration Parameters (Ptrl+E) > Solver)

Solver options			
Type: Fixed-step	Y So	olver: ode8 (Dormand-Prince)	/
Fixed-step size (fundamental sample time):	(0.001	

3. Запустите модель. Рассмотрите полет мяча. Постройте его график.



Видно, что полет до поверхности с высоты одного метра под действием гравитации занимает 0.45 секунд. Далее мяч пролетает поверхность без отскока. Необходимо включить в модель описание отскока. Добавим в список отношений новое:

• Сила сжатия тела равна произведению сжатия тела на коэффициент упругости.

Отношение начинает действовать когда мяч достиг поверхности, а высота меняет знак.

4. Дополните модель функциями описывающими отскок и проверьте работу доработанной модели. Подберите коэффициент упругости который соответствует сжатию мяча на 4.5 см.



Видно, что полет с отскоками не затухает. Процесс не теряет энергию. Эта модель не соответствует часто наблюдаемым скачкам мяча. Добавим в список новое отношение:

- Сила трения мяча о воздух пропорциональна квадрату скорости мяча.
- Дополните модель блоками описывающими трение мяча о воздух и проверьте работу доработанной модели. Подберите коэффициент трения при котором мяч взлетает на 80% исходной высоты.



Этот полет мяча наиболее полно соответствует условиям задачи. В дальнейшем, параметры модели можно уточнять по экспериментальным данным.

Задание 3. Построение модели управления стабилизацией амплитуды отскока мяча.

Для поддержания заданной амплитуды отскока мяча можно организовать управление смещением поверхности отскока перпендикулярно падению мяча в зависимости от амплитуды отскока.



Рис. 4. Полет меча с отскоком (слева) и контур управления амплитудой отскока (справа).

- 1. Сохраните модель движения мяча предыдущего задания под новым именем.
- 2. Включите в модель сумматор и канал дополнительной скорости:



3. Выделите оконтуриванием блоки модели не включая блок Scope и канал дополнительной скорости (красная штрих-пунктирная линия) и создайте подсистему командой ПКМ Preate Subsystem.



Должен получиться следующий блок подсистемы (Sybsystem):



4. Добавьте в подсистему порт для вывода информации о контакте мяча с поверхностью, измените названия подсистемы и портов и цветовую окраску портов.



5. Изменяя относительную скорость меча во время контакта (вход Add rate подсистемы отскок (Rebound)) убедитесь, что амплитуда отскока зависит от входного воздействия подсистемы и через такое воздействие можно управлять отскоком.

Рис. 5. Зависимость отскока от дополнительной скорости контакта мяча при -0.3,-0.23, 0 и 0.3 м/сек. Через дополнительную скорость контакта мяча можно увеличивать и уменьшать амплитуду отскоков мяча.

6. Постройте контур управления и подберите структуру и коэффициенты регулятора обеспечивающего устойчивость контура управления амплитудой отскоков мяча.

Рис. 6. Результат управления амплитудой отскоков мяча.

Задание 4. Отображение сигналов модели Simulink средствами МатЛАБ.

- 1. Откройте окно графика блока Scope модели предыдущего задания.
- 2. Нажатием на кнопку 🔤 меню откройте окно параметров блока Scope.
- 3. Установите соответствующие параметры закладки History:

4 'Scope' pa r	ameters 🚺 🗖 🔀				
General History Style					
imit data points to last: 5000					
Save data to workspace					
Variable name:	ScopeData				
Format:	Structure with time				
	OK Cancel Help Apply				

4. Запустите модель и найдите наблюдаемые параметры (структура ScopeData) в рабочей

	Workspace	
	1 1 2 2	Stack: Base 🗹
	Name 🔺	Value
области МатЛАБ (workspace)	🗄 ScopeData	<1×1 struct>

- 5. Сохраните данные моделирования в mat файле с именем, например, ball.
- > save ('ball','ScopeData')
- 6. Очистите workspace: >clear all
- 7. Откройте редактор m-файла: меню МатЛАБ > File > New > Script
- 8. Создайте программу отображения данных моделирования хранящиеся в ball.mat файле. filename = 'ball'; load(filename); time = ScopeData.time; amplitude = ScopeData.signals(1).values(:,1); peak = ScopeData.signals(1).values(:,2); error = ScopeData.signals(2).values; figure plot(time, amplitude, 'b', 'linewidth',2); hold on plot(time, peak, 'r', 'linewidth',2); hold on plot(time, error, 'm', 'linewidth',1); grid on legend('Height', 'Peak', 'Error'); xlabel('Time, sec'); ylabel('Height [m], Peak [m], Error [m]'); title('Ball control parameters against Time');

9. Запустите скрипт на выполнение командой 돈. Наблюдайте построенные графики.

Рис. 7. Результат моделирования управления амплитудой отскока мяча.

контрольные вопросы

- 1. Как запустить пакет Simulink?
- 2. Как создать модель?
- 3. Как передать результаты моделирования в рабочую область МатЛАБ?
- 4. Как задать параметры и метод моделирования?
- 5. Какое расширение имеют файлы Simulink модели?

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Help МатЛАБ.