

Школа MATLAB.

Моделирование устройств силовой электроники. Урок 8. Создание виртуальных лабораторий в среде Matlab – Simulink

Предлагаемая статья продолжает цикл публикаций [2–8]. Перед изучением данного материала рекомендуется ознакомиться с основными разделами уроков 1–7. Модели, использованные в настоящей и предыдущей статьях, могут быть переданы автором по запросу через Интернет.

**Сергей Герман–Галкин,
д. т. н.**

ggsg@yandex.ru

Цель этого урока — рассмотрение возможностей среды Matlab – Simulink в создании масштабированных (виртуальных) блоков и имитационных лабораторных стендов, представляющих собой удобный и наглядный интерфейс для исследования разнообразных технических устройств.

Введение

Модель отражает те или иные свойства реального физического объекта. Модели бывают физическими, аналоговыми и математическими.

Математическая модель является математическим описанием реального физического объекта. Это описание базируется на физических законах, определяющих поведение объекта, и служит для исследования свойств объекта, необходимых инженеру. При наличии математического описания (математической модели) исследование свойств объекта может быть реализовано аналитическими методами, либо с использованием компьютерного моделирования.

Аналитические методы имеют существенные ограничения. Они позволяют в полной мере исследовать системы, которые описываются дифференциальными уравнениями первого и второго порядка. Системы, описываемые уравнениями третьего и четвертого порядка, поддаются аналитическому решению, но влияние параметров системы приходится исследовать уже численными методами. Системы более высоких порядков исследуются только численными методами.

Численные методы базируются на использовании компьютерного моделирования.

Компьютерная модель — это программная реализация математической модели, дополненная различными служебными программами (например, рисующими и изменяющими графические образы во времени).

На заре компьютерного моделирования программы создавались на языке машинных кодов (1100101...). Следующим шагом стал язык Ассемблера. В дальнейшем появились языки программирования высокого уровня (Алгол, Бейсик, Фортран, Паскаль, С++ и др.). Применявшаяся в те годы технология программиро-

вания требовала очень много времени на создание моделей. Трудозатраты на создание простой, с современной точки зрения, компьютерной модели оценивались в несколько человеко-месяцев.

В настоящее время положение кардинально изменилось. С 1970-х годов было разработано множество прикладных пакетов моделирования, автоматизирующих, прежде всего, этап представления математической модели для компьютера. Среди них лидирующее положение для технических систем занимают пакеты Matlab и Simulink фирмы Math Work.

Matlab имеет собственный язык программирования более высокого уровня, чем перечисленные ранее. Однако разработчики пакета пошли дальше. Был создан пакет Simulink для исследования и проектирования динамических систем, который, по сути, является очень удобным интерфейсом для пользователей — специалистов в области теории управления. Продемонстрируем сказанное примером.

Пример 1. Динамическая система описывается дифференциальным уравнением четвертого порядка

$$a_4 \frac{d^4 x}{dt^4} + a_3 \frac{d^3 x}{dt^3} + a_2 \frac{d^2 x}{dt^2} + a_1 \frac{dx}{dt} + a_0 x = u, \quad (1)$$

где x — выходной сигнал, u — входной сигнал. Для решения уравнения (1) в пакете Matlab его необходимо представить в нормальной форме Коши

$$\dot{X} = A_1 X, \quad (2)$$

где

$$\dot{X} = \begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{u} \end{pmatrix} \quad \text{— расширенная матрица производных от переменных состояния,}$$

$$X = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ u \end{pmatrix} \text{ — расширенная матрица переменных состояния,}$$

$$A_1 = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ \hline a_0 & a_1 & a_2 & a_3 & 1 \\ a_4 & a_4 & a_4 & a_4 & a_4 \\ \hline 0 & 0 & 0 & 0 & 0 \end{pmatrix} \text{ — расширенная матрица состояния.}$$

Рассмотрим расчет переходного процесса в системе со следующими данными:

$$a_0 = 1, a_1 = 0.8, a_2 = 0.54, a_3 = 0.18, a_4 = 0.025, u = 1.$$

Процедура интегрирования системы дифференциальных уравнений (2) включает создание специальной функции, в которой задаются начальные условия и выводятся результаты и создание подфункции, в которой вычисляются правые части уравнения (2). Программа решения представлена листингом 1.

```

Листинг 1
% Создание специальной функции
% для вычисления начальных условий
% и вывода результатов интегрирования
function solvden
X0=[0;0;0;0;1];
[T, X]=ode45(@sgg,[0 15], X0);
%Вывод результатов решения на график
plot(T, X(:,1), 'r.-');
hold on;
plot(T, X(:,2), 'k.-');
xlabel('Time (s)');
ylabel('x1, dx1/dt');
grid on;
hold off;
% Подфункция вычисления правых частей уравнения.
function F=sgg(t, x)
A=[0,1,0,0;0,0,1,0;0,0,0,1;0,-40,-32,-21.6;-7.2,40;0,0,0,0];
X=[x(1); x(2); x(3); x(4); x(5)];
F=A*X;
    
```

Результат выполнения программы показан на рис. 1, где построены зависимости выходной координаты системы и ее производной от времени (четвертая строчка программы).

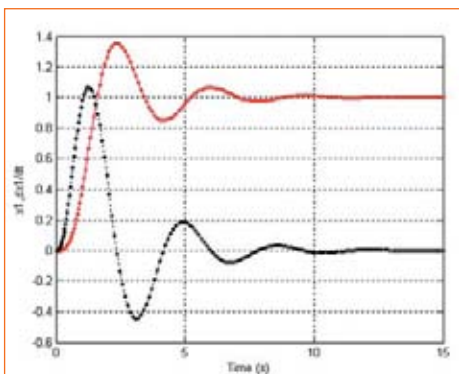


Рис. 1. Зависимости выходной координаты системы и ее производной от времени

Дифференциальному уравнению (1) в пакете Simulink соответствует операторная передаточная функция (блок Transfer Fcn), пиктограмма и окно настройки параметров которой представлены на рис. 2.

Если на вход этого блока подключить входной порт Input из библиотеки Source, а на выход подключить выходной порт Output из библиотеки Sink, то можно исследовать динамические характеристики в системе, используя пакет расширения Control System Toolbox [8]. В качестве таких характеристик на рис. 3 представлены переходная и амплитудно-частотная характеристики рассматриваемого динамического объекта.

Специалисту в области теории управления гораздо удобнее иметь дело с передаточной функцией, при этом ему совершенно не требуется глубоко изучать язык Matlab. Блок Transfer Fcn пакета Simulink связан с основным ядром пакета Matlab, где написаны и встроены соответствующие программы.

Дальнейшее совершенствование возможностей среды Matlab – Simulink связано с разработкой так называемых программ расширения (Toolboxes – Blocksets).

Библиотеки Blocksets пакета Simulink в большинстве случаев представляют собой маскированные блоки этого пакета. Маскированные блоки пакетов расширения, интегрируя в себе несколько блоков пакета Simulink, еще в большей степени упрощают работу специалиста.

Создание маскированных (виртуальных) блоков

Маскированные (виртуальные) блоки особенно полезны при реализации виртуальных лабораторий для дистанционного обучения. Эти лаборатории создаются путем маскирования моделей пакета Simulink, из них легко формируются отдельные (специализированные) библиотеки.

Пример 2. Рассмотрим способ создания виртуального ПИД-регулятора с реальным дифференцирующим звеном, реализованного методом прямого программирования.

Представим передаточную функцию ПИД-регулятора с реальным дифференцирующим звеном в виде

$$W_R(s) = \frac{x}{u} = k_p + \frac{k_I}{s} + \frac{k_D s}{T_D s + 1} = \frac{b_2 + b_1 s^{-1} + b_0 s^{-2}}{1 + a_1 s^{-1}}, \quad (3)$$

$$\text{где } a_1 = \frac{1}{T_D}, b_0 = \frac{k_I}{T_D}, b_1 = \frac{k_p + T_D k_I}{T_D}, b_2 = \frac{k_D + T_D k_p}{T_D}$$

k_p, k_I, k_D — коэффициенты передачи пропорционального, интегрального и дифференци-

Рис. 2. Пиктограмма и окно настройки блока Transfer Fcn

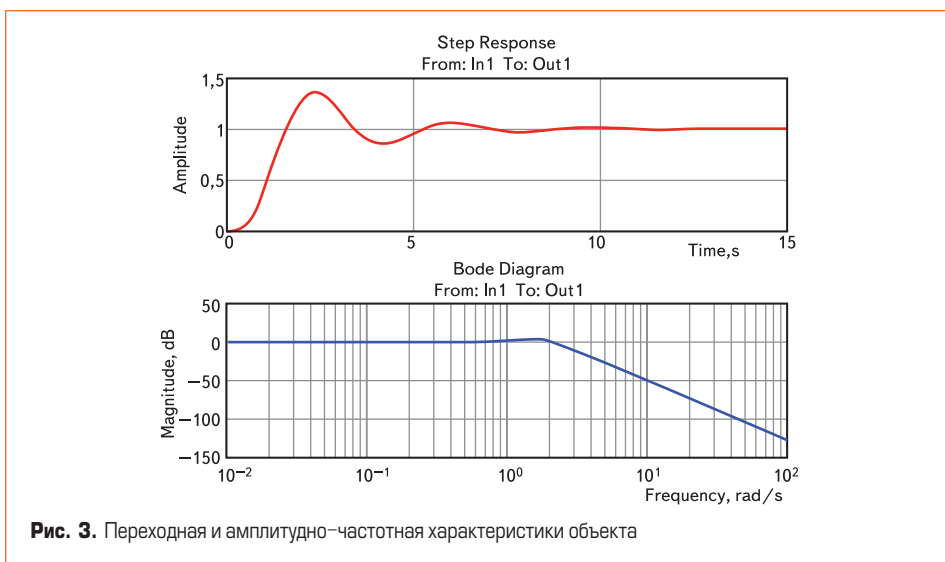


Рис. 3. Переходная и амплитудно-частотная характеристики объекта

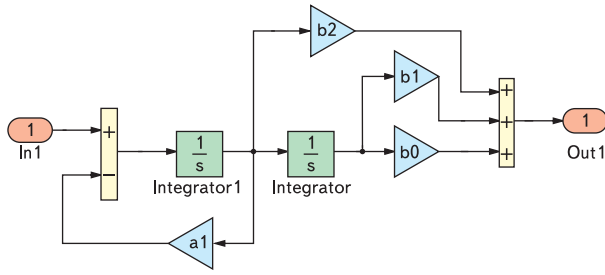


Рис. 4. Непрерывный ПИД-регулятор, реализованный методом прямого программирования в пакете Simulink

ального звеньев соответственно, T_D — постоянная времени аperiodической части дифференцирующего звена. Из уравнения (3) получим:

$$x = (b_2 + b_1s^{-1} + b_0s^{-2}) \cdot e, \text{ где } e = u / (1 + a_1s^{-1}), e = u - a_1e s^{-1} \quad (4)$$

По уравнениям (4) на рис. 4 построена модель ПИД-регулятора с реальным дифференцирующим звеном в пакете Simulink.

Коэффициенты передачи блоков в модели вычисляются по выражениям (3) при задании четырех основных параметров регулятора k_p , k_i , k_d и T_D .

Маскирование модели осуществляется в определенной последовательности:

1. Заключаем всю модель в рамку, удерживая левую кнопку мыши.
2. В поле Edit основного меню выбираем опцию Create Subsystem, при этом схема модели примет вид одного блока Subsystem.
3. Отметим этот блок и в поле Edit основного меню выберем опцию Mask Subsystem — при этом открывается редактор маски.

Редактор маски содержит четыре основные вкладки (*Icon*, *Parameter*, *Initialization*, *Documentation*), позволяющие задать все свойства маскированной системы. В нижней части окна редактора расположены четыре кнопки *OK*, *Cancel*, *Help*, *Apply*, которые являются общепринятыми для всех блоков пакета Simulink, а также кнопка *Unmask*, позволяющая демаскировать модель.

Окно редактора при нажатой кнопке *Icon* показано на рис. 5. В этом случае создается внешний вид маскируемой подсистемы. На па-

нели *Icon option* имеются четыре разворачивающихся списка — *Frame*, *Transparency*, *Rotation*, *Units*, которые позволяют установить способ представления иконки маскируемого блока. В таблице 1 представлены способы представления иконки блока.

Таблица 1. Способы представления иконки блока

Вкладка	Опции разворачивающегося меню	Внешний вид иконки маскированного блока
Frame	Visible	Рамка иконки выводится
	Invisible	Рамка иконки невидима
Transparency	Opaque	Изображение на иконке непрозрачное
	Transparent	Изображение на иконке прозрачное
Rotation	Rotates	Изображение поворачивается с иконкой
	Fixed	Изображение на иконке фиксировано
Units	Autoscale	Автомасштабирование иконки блока
	Normalized	Масштаб иконки блока постоянный

На панели *Drawing commands* вводятся команды графического представления текста, графика или рисунка, отображаемого на иконке. В примере на рис. 5 введена команда на языке Matlab, помещающая текст **PID-control** в центр иконки.

Вид окна редактора при активной вкладке *Parameters* показан на рис. 6. В этом окне задаются параметры маскируемого блока. В левой

части окна помещены кнопки, позволяющие добавлять, убирать и перемещать параметр. На панели *Dialog parameters* имеются четыре поля: *Prompt*, *Variable*, *Type*, *Evaluate*, *Tunable*. Поле *Prompt* (пояснение) служит для ввода названия параметра. В это поле введены названия основных параметров ПИД-регулятора (уравнение 3). В поле *Variable* (переменная) задаются обозначения параметров, под которыми они записываются в рабочую область (Mask Workspace) маскированной системы.

В поле *Type* в выпадающем списке устанавливается способ задания для каждого параметра:

- *Edit* — с помощью строки редактирования;
- *Checkbox* — с помощью переключателя;
- *Popup* — с помощью разворачивающегося меню.

При выборе разворачивающегося меню становится доступным поле *Popups* (one per line), куда вводятся значения соответствующего параметра. Для того чтобы видеть и редактировать параметры маскированного блока, в окнах *Show parameter*, *Enable parameter* необходимо проставить галочки.

В поле *Dialog callback* помещаются команды MATLAB, которые выполняются при открытии блока.

Вид окна редактора с активной вкладкой *Initialization* показан на рис. 7. При открытии этого окна в поле *Dialog variables* высвечиваются обозначения параметров, заданных на предыдущей вкладке *Variable*.

Область ввода *Initialization commands* (Команды инициализации) является аналогом командного окна MATLAB, область действия которого ограничена рабочей областью маскированной системы. В эту область помещаются операторы и М-функции, которые загружаются в рабочую область маскированной системы при закрытии окна редактора маски.

Вид окна редактора с активной вкладкой *Documentation* показан на рис. 8. В полях этого окна вводятся комментарии, которые помогают пользователю работать с маскированным блоком. В поле *Mask type* вводится название блока. В поле *Mask description* вводится краткий поясняющий текст. В поле *Mask help* вводится текст, который появляется при переходе на вкладку *Help* в окне параметров блока.

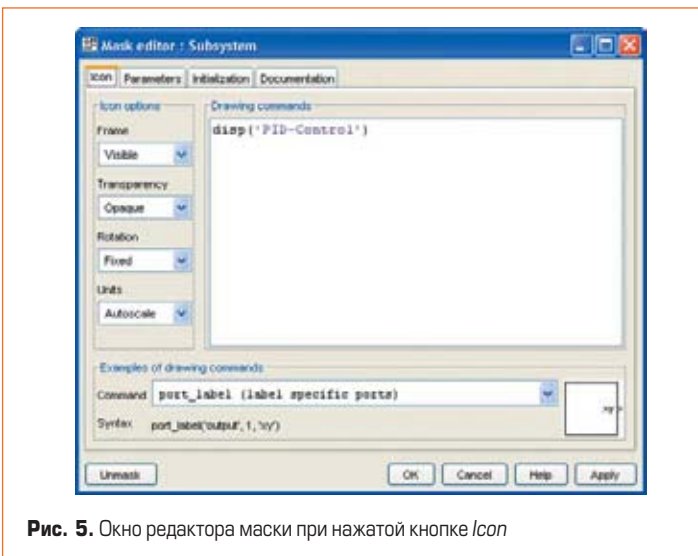


Рис. 5. Окно редактора маски при нажатой кнопке *Icon*

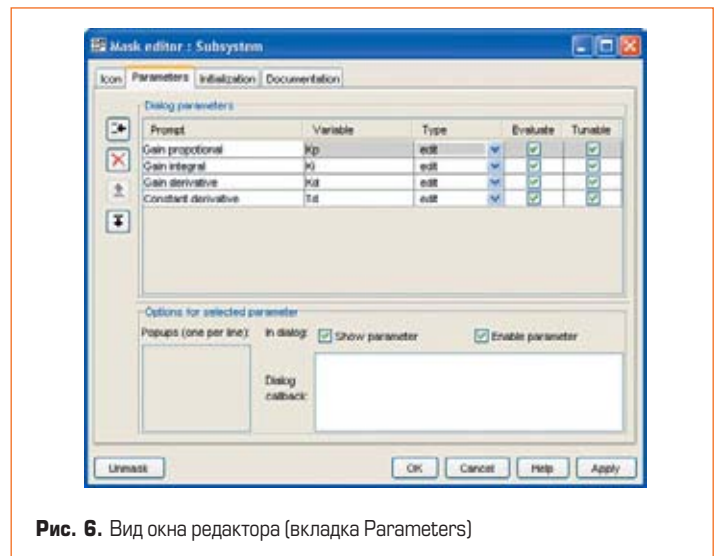


Рис. 6. Вид окна редактора (вкладка *Parameters*)

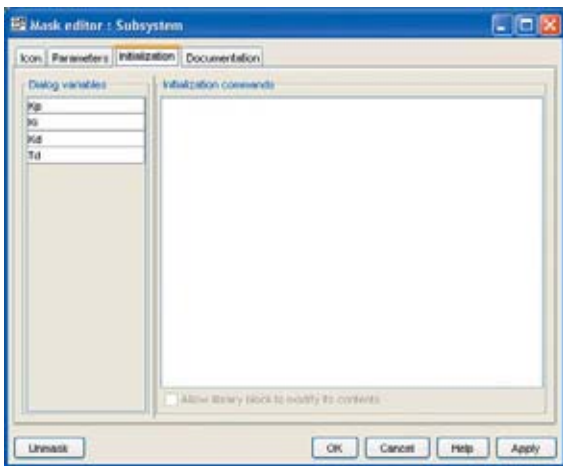


Рис. 7. Вид окна редактора (вкладка Initialization)

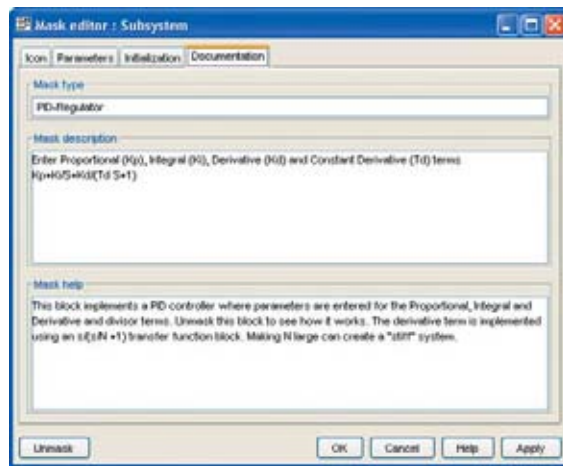


Рис. 8. Вид окна редактора (вкладка Documentation)

После задания всех параметров маскированной системы ее пиктограмма и окно настройки параметров принимают вид, показанный на рис. 9.

Создание имитационных лабораторных стендов (GUI)

Пакет Matlab предоставляет разработчику чрезвычайно удобные графические средства для создания графических интерфейсов пользователя — GUI (Graphical User Interface). Назначение разрабатываемых GUI может быть самым различным. В частности, эти GUI могут выполнять функции имитационных лабораторных стендов, предназначенных для изучения различных технических устройств и, в частности, устройств силовой электроники. Подробное описание вопросов проектирования GUI читатель найдет в руководстве [1].

Далее представлен результат проектирования имитационного лабораторного стенда, предназначенного для изучения и исследования широтно-импульсного преобразователя (ШИП).

Предварительно коротко рассмотрим принципы построения и физические основы работы широтно-импульсного преобразователя.

Одноплечевой ШИП с симметричным законом управления

Широтно-импульсные преобразователи (ШИП) используются, как правило, в системах постоянного тока. По построению ШИП можно разделить на одноплечевые и мосто-

вые (двухплечевые). Питание первых осуществляется от источника со средней точкой.

Для управления ШИП используются в основном три способа (алгоритма) переключения полупроводниковых ключей:

- симметричный;
- несимметричный;
- поочередный.

Два последних способа используются в мостовых ШИП.

Схема одноплечевого транзисторного ШИП представлена на рис. 10а.

При симметричном способе управления транзисторы плеча переключаются в противофазе (рис. 10г).

Рассмотрим последовательность образования коммутационных интервалов и электромагнитные процессы в ШИП при симметричном управлении для общего случая R, L нагрузки с противо-ЭДС. При включении верхнего по схеме транзистора VT1 и выключении нижнего VT2 образуется цепь +U_n, VT1, R, L, E, -U_n (рис. 10б) для протекания тока нагрузки. К нагрузке в этом интервале прикладывается напряжение питания U_d/2, а ток увеличивается от минимального значения I_{min} до максимального I_{max} (рис. 10г). Напряжение на транзисторе VT1 на этом интервале равно нулю, а ток равен току нагрузки.

На транзисторе VT2 напряжение равно U_d, это напряжение является отрицательным для диода D2, и ток через него равен нулю. При

включении нижнего по схеме транзистора VT2 и выключении верхнего VT1 ток, поддерживаемый индуктивностью нагрузки, продолжает протекать в том же направлении. При этом образуется цепь, показанная на рис. 10в, в которой ток нагрузки протекает навстречу ЭДС. E и источнику питания. На этом интервале напряжение на нагрузке изменяет знак, а ток уменьшается (рис. 10г).

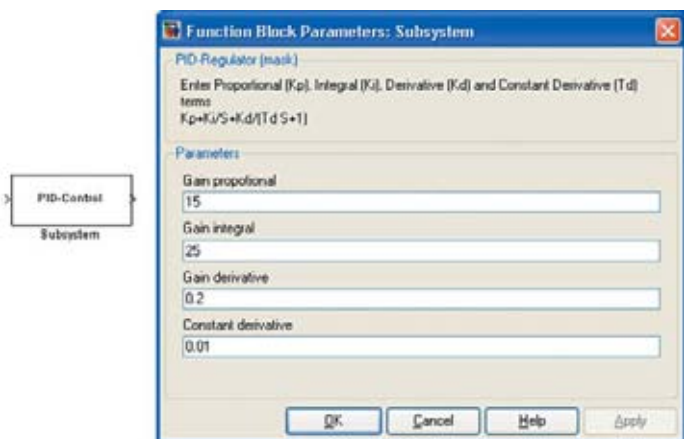


Рис. 9. Пиктограмма и окно настройки параметров ПИД-регулятора

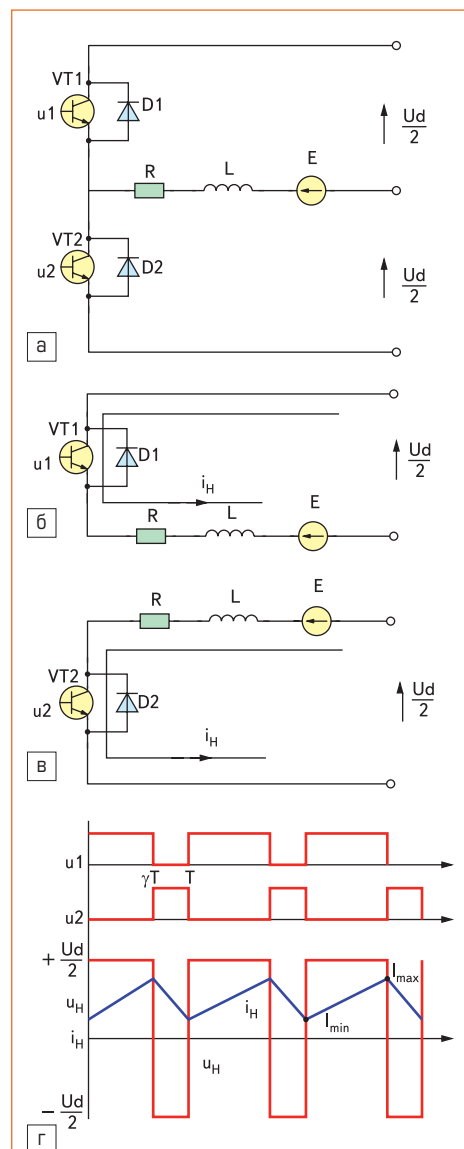


Рис. 10. Одноплечевой транзисторный ШИП

К транзистору VT1 прикладывается напряжение U_d , D2 открыт, напряжение на нем равно нулю, а ток равен току нагрузки (рис. 10в).

Если к началу рассматриваемого интервала в индуктивности нагрузки накоплена достаточная энергия, то такое состояние цепи сохраняется до очередного переключения транзисторов плеча. Если эта энергия недостаточна, то ток может упасть до нуля, а затем изменить направление под действием напряжения $U_d/2$ и ЭДС. В этом случае он переходит с диода D2 на транзистор VT2.

Поскольку диод и транзистор во включенном состоянии считаются идеально замкнутыми ключами, то на анализе электромагнитных процессов переход тока с диода на транзистор не сказывается. При этом реализуется два состояния силовой части ШИП: одно — когда VT1 включен, а VT2 выключен; второе — когда VT1 выключен, а VT2 включен. Длительности этих состояний соответственно равны γT и $(1 - \gamma) T$, где γ — относительная длительность первого состояния, изменяющаяся от 0 до 1 в зависимости от величины сигнала управления.

Поэтому можно считать, что в схеме на периоде образуются два коммутационных интервала, схемы замещения на которых показаны на рис. 10б, в. При симметричном законе формируется знакопеременное напряжение на нагрузке, а среднее значение этого напряжения определяется из выражения:

$$U_H = \frac{1}{T} \int_0^{\gamma T} \frac{U_d}{2} dt - \frac{1}{T} \int_{\gamma T}^T \frac{U_d}{2} dt = (2\gamma - 1) \frac{U_d}{2} \quad (5)$$

Оно равно нулю при $\gamma_0 = 0,5$, при $\gamma > 0,5$ среднее напряжение на нагрузке положительное, при $\gamma < 0,5$ — отрицательное.

Мостовой широтно-импульсный преобразователь

Принципиальная схема мостового ШИП представлена на рис. 11а. Она содержит четыре транзисторных ключа VT1–VT4 с обратными диодами D1–D4. В диагональ моста, образованную транзисторными ключами, включена нагрузка.

Питание ШИП осуществляется от источника постоянного тока.

При симметричном способе управления в состоянии переключения находятся все четыре транзисторных ключа моста, а напряжение на выходе ШИП представляет собой знакопеременные импульсы, длительность которых регулируется входным сигналом.

Временные диаграммы ШИП при симметричном способе управления приведены на рис. 11б. По сравнению с рассмотренной выше одноплечевой схемой здесь нет никаких особенностей.

Симметричный способ управления обычно используется в маломощных приводах постоянного тока. Его преимуществом является простота реализации и отсутствие зоны нечувствительности в регулировочной характеристике. Недостатком ШИП с симметричным управлением является двуполярное напряже-

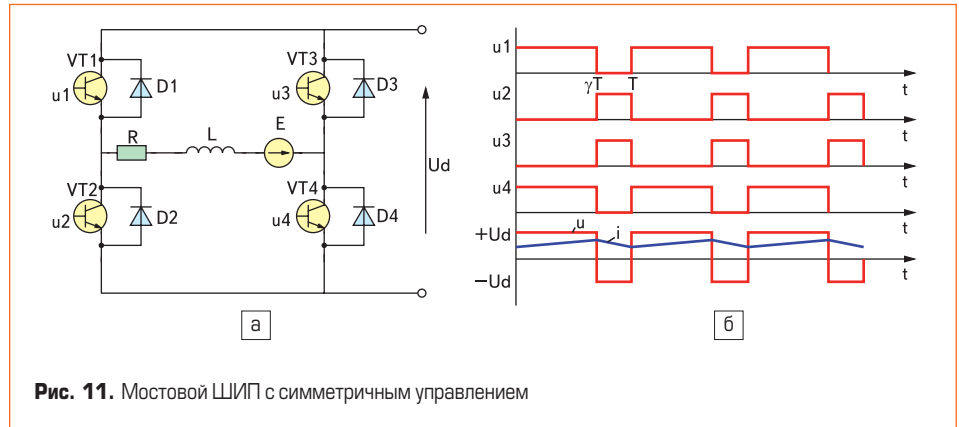


Рис. 11. Мостовой ШИП с симметричным управлением

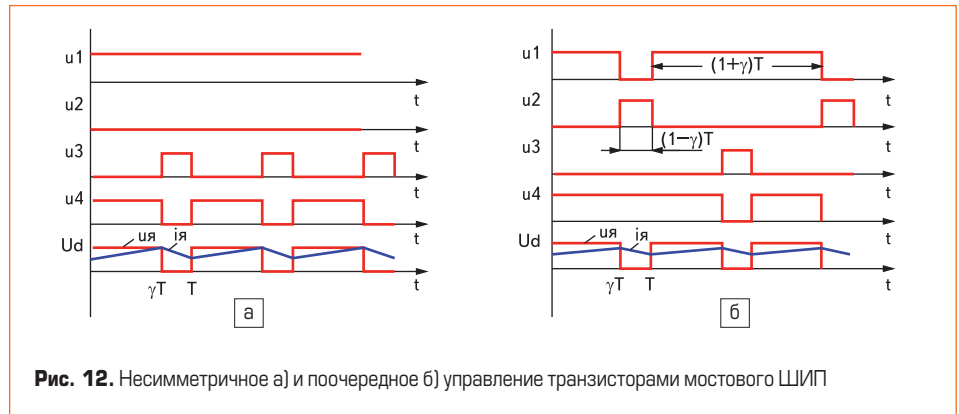


Рис. 12. Несимметричное а) и поочередное б) управление транзисторами мостового ШИП

ние на нагрузке и, в связи с этим, повышенные пульсации тока в нагрузке.

Стремление исключить этот недостаток привело к разработке способов, обеспечивающих однополярное напряжение на выходе ШИП. Простейшим из них является несимметричный.

Электромагнитные процессы в ШИП при несимметричном управлении представлены на рис. 12а. В этом случае переключаются транзисторные ключи фазной группы VT3 и VT4 (ключи VT1 и VT2 при противоположной полярности входного сигнала), транзисторный ключ VT1 постоянно открыт и насыщен, а ключ VT2 постоянно закрыт.

Транзисторные ключи VT3 и VT4 переключаются в противофазе.

При этом на выходе ШИП формируются однополярные импульсы и среднее напряжение на выходе равно нулю, когда относительная продолжительность включения одного из нижних по схеме транзисторов $\gamma_0 = 0$.

Недостатком рассмотренного способа управления является то, что верхние по схеме транзисторные ключи (VT1, VT3) по току нагружены больше, чем нижние. Этот недостаток устраняется при поочередном управлении, временные диаграммы которого изображены на рис. 12б.

Здесь при любом знаке входного сигнала в состоянии переключения находятся все четыре транзисторных ключа моста, при этом частота переключения каждого из них в два раза меньше частоты напряжения на выходе. Управляющие напряжения транзисторных ключей одной фазы моста VT1, VT2 и VT3, VT4 постоянно находятся в противофазе; при этом ключи переключаются через период выходного напряжения T . Этим достигаются одинаковые условия работы полупроводниковых приборов в мостовой схеме.

При некотором знаке входного сигнала управляющие импульсы u_1, u_4 длительностью $t_1 = (1 + \gamma) T$ подаются на диагонально расположенные транзисторные ключи (рис. 12б) со сдвигом на полпериода, а управляющие импульсы u_2, u_3 длительностью $t_2 = (1 - \gamma) T$, также со сдвигом на полпериода, подаются на транзисторы противоположной диагонали (VT2, VT3). В этом случае на интервале γT нагрузка подключена к источнику питания с помощью диагонально расположенных ключей, а на интервале $(1 - \gamma) T$ нагрузка закорочена с помощью верхних или нижних транзисторных ключей. При изменении знака входного сигнала порядок управления диагональными ключами изменяется на противоположный. При несимметричном и поочередном управлении на нагрузке формируются однополярные импульсы длительностью γT , пропорциональной сигналу на входе, среднее значение напряжения на нагрузке определяется из выражения:

$$U_H = \frac{1}{T} \int_0^{\gamma T} U_H dt = \gamma U_H. \quad (6)$$

Процесс создания имитационного стенда разделяется на два этапа — **проектирование и реализация**.

На этапе проектирования разработчик стенда должен четко представить себе конечную задачу, которую должен выполнять разработанный лабораторный стенд.

На втором этапе, по сути, осуществляется программная реализация проекта. Вся разработка является процессом итерационным, поскольку общая концепция проекта влияет на программную реализацию, а практическое программирование часто может подсказать новые, интересные решения в самом проекте.

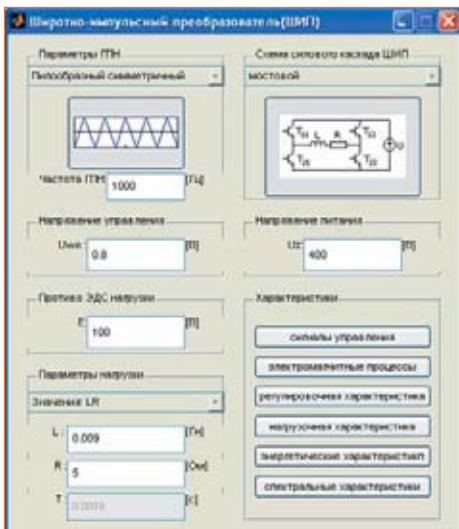


Рис. 13. Проект главного окна GUI для исследования транзисторного ШИП

Базой для расчета и построения характеристик исследуемого устройства являются Simulink-модели, данные для которых задает пользователь в главном окне GUI. Программы, реализующие GUI, собирают эти данные, передают их соответствующим Simulink-моделям, запускают эти модели и представляют результаты моделирования.

Проект главного окна GUI для исследования транзисторного ШИП представлен на рис. 13.

Все исходные данные исследуемого устройства и все полученные при исследовании характеристики доступны для пользователя в главном окне GUI, создавая впечатление реальной лабораторной установки.

Главное окно содержит несколько блоков, каждый из которых выполняет определенную функцию по сбору данных, вводимых пользователем, или по представлению результатов моделирования.

Первый блок (верхний слева) позволяет задать параметры генератора пилообразного напряжения несущей частоты. Здесь в раскрывающемся списке задается и отражается форма пилообразного напряжения.

В поле «Частота ГПН» задается частота генератора в Герцах.

Три последующие блока содержат поля, в которые вводятся:

- напряжение на входе;
- противо-ЭДС нагрузки;
- параметры (R, L, T) нагрузки.

В шестом блоке (верхний справа) из разворачивающегося списка выбирается схема ШИП мостовая или полумостовая (одноплечевая). Эта схема отражается в графическом поле блока. Далее задается напряжение питания ШИП.

Последний блок содержит кнопки, открывающие соответствующие графические окна, в которых учащийся может наблюдать электромагнитные процессы и характеристики.

На рис. 14 показаны основные файлы директории, содержащей весь проект.

Как следует из рис. 14, GUI содержит mdl-файлы (Simulink-модели), графические bmp-файлы и m-файлы. Simulink-модели предназначены для получения квазиустановившихся



Рис. 14. Основные файлы проекта

ся процессов и статических характеристик. Подробное описание получения этих характеристик было представлено ранее [3]. В графических файлах содержатся рисунки, которые выводятся в два верхних блока GUI (рис. 13). Внешний вид GUI и взаимодействие между всеми файлами проекта описаны в m-файлах.

Главный блок проекта, представленный листингом 1, определяет тип переменных, размер главного графического окна и размещение основных панелей в этом окне.

Листинг 1

```
01 % pr0
02 global data simin_ster;
03 data.gui.L=0.009;
04 data.gui.R=5.0;
05 data.gui.T=0.0018;
06 data.gui.Hz=500.0;
07 data.gui.HzType=1;
08 data.gui.Uz=240.0;
09 data.gui.Ster=0.5;
10 data.gui.simType=1;
11 data.gui.Sem=0;
12
13 h=pr0_init ([16 8 92 37]);
14
15 pr2(h,[4 23 40 12]);
16 pr5(h,[48 23 40 12]);
17 pr4(h,[4 16 40 5]);
18 pr3(h,[48 16 40 5]);
19 pr1(h,[4 2 40 12]);
20 prB (h,[48 2 40 12]);
21 set (h, 'Color', get (findobj (h, 'Tag', 'pr1'), 'BackgroundColor'));
```

M-файлы `pr1` – `pr6` инициализируют отдельные панели. В результате в программу вводятся задаваемые пользователем данные и запускаются функции обратного вызова (callback-функции).

Далее, в листинге 2, представлен код m-функции `pr1`, отвечающей за панель ввода параметров нагрузки.

В строках 23 и 40 записана функция обратного вызова `pr1_callback`. Код этой функции приведен в листинге 3.

Инициализация остальных панелей и их функций обратного вызова программируются аналогично.

Программа m-файла `prB` (листинг 4) обеспечивает запуск Simulink-моделей и построение соответствующих характеристик с использованием функций обратных вызовов.

Программа позволяет построить следующие квазиустановившиеся процессы и характеристики:

- Сигналы управления (функция `prB1_callback`).
- Электромагнитные процессы (функция `prB2_callback`).

Листинг 2

```
01 % pr1
02 function h=pr1 (parent, pos);
03 global combochrg editchrg data grayColor;
04 h=uipanel ('Parent', parent);
05 set (h, 'Units', 'characters');
06 set (h, 'Position', pos);
07 set (h, 'Title', 'Parameter obciazenia');
08 set (h, 'Tag', 'pr1');
09 grayColor=get (h, 'BackgroundColor');
10 str={'dane: LR','dane: LT','dane: RT'};
11 combochrg=icontrol (h);
12 set (combochrg, 'Style', 'PopupMenu', 'Units', 'normalized', 'Position', [0 0.66 1 0.18]);
13 set (combochrg, 'String', str);
14
15 strdata={'L:', 'R:', 'T:', '[H]', '[Ohm]', '[sek]'};
16 for ii=1:3;
17 editchrg (ii)=icontrol (h);
18 set (editchrg (ii), 'Style', 'Edit');
19 set (editchrg (ii), 'HorizontalAlign', 'left');
20 set (editchrg (ii), 'Units', 'normalized');
21 set (editchrg (ii), 'Position', [0.2 (3-ii)*0.22 0.6 0.18]);
22 set (editchrg (ii), 'BackgroundColor', 'white');
23 set (editchrg (ii), 'Callback', 'pr1_callback (i)');
24 for jj=1:2;
25 t (ii, jj)=icontrol (h);
26 set (t (ii, jj), 'Style', 'Text');
27 set (t (ii, jj), 'String', strdata (ii+3*(jj-1)));
28 set (t (ii, jj), 'Units', 'normalized');
29 set (t (ii, jj), 'Position', [(jj-1)*0.8 (3-ii)*0.22 0.2 0.18]);
30 if (jj==1);
31 set (t (ii, jj), 'HorizontalAlign', 'right');
32 else
33 set (t (ii, jj), 'HorizontalAlign', 'left');
34 end;
35 end;
36 end;
37 set (editchrg (1), 'String', data.gui.L);
38 set (editchrg (2), 'String', data.gui.R);
39 set (editchrg (3), 'String', data.gui.T);
40 set (combochrg, 'Callback', 'pr1_callback');
41 pr1_callback ();
```

Листинг 3

```
01 % pr1_callback
02 function pr1_callback ();
03 global combochrg editchrg grayColor data;
04 w=get (combochrg, 'Value');
05 for ii=1:3;
06 if (w==ii)
07 set (editchrg (4-ii), 'Enable', 'off');
08 set (editchrg (4-ii), 'BackgroundColor', grayColor);
09 else
10 set (editchrg (4-ii), 'Enable', 'on');
11 set (editchrg (4-ii), 'BackgroundColor', 'white');
12 end;
13 end;
14 switch w;
15 case 1;
16 data.gui.R=pr_edit (editchrg (2));
17 data.gui.L=pr_edit (editchrg (1));
18 data.gui.T=data.gui.L/data.gui.R;
19 set (editchrg (3), 'String', data.gui.T);
20 case 2;
21 data.gui.L=pr_edit (editchrg (1));
22 data.gui.T=pr_edit (editchrg (3));
23 data.gui.R=data.gui.L/data.gui.T;
24 set (editchrg (2), 'String', data.gui.R);
25 case 3;
26 data.gui.R=pr_edit (editchrg (2));
27 data.gui.T=pr_edit (editchrg (3));
28 data.gui.L=data.gui.T*data.gui.R;
29 set (editchrg (1), 'String', data.gui.L);
30 end;
```

Листинг 4

```
01 % prB
...
19 set (buttons (1), 'Callback', 'prB1_callback (i)');
20 set (buttons (2), 'Callback', 'prB2_callback (i)');
21 set (buttons (3), 'Callback', 'prB3_callback (i)');
22 set (buttons (4), 'Callback', 'prB4_callback (i)');
23 set (buttons (5), 'Callback', 'prB5_callback (i)');
24 set (buttons (6), 'Callback', 'prB6_callback (i)');
```

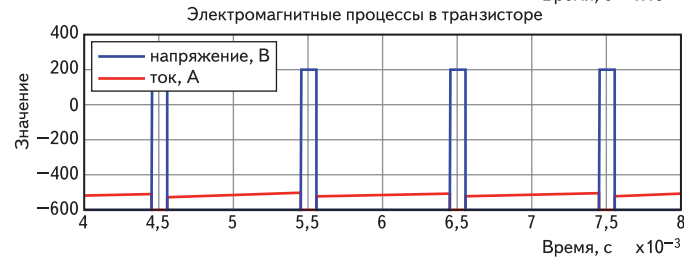
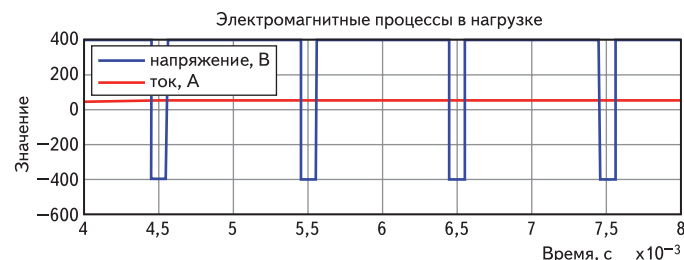
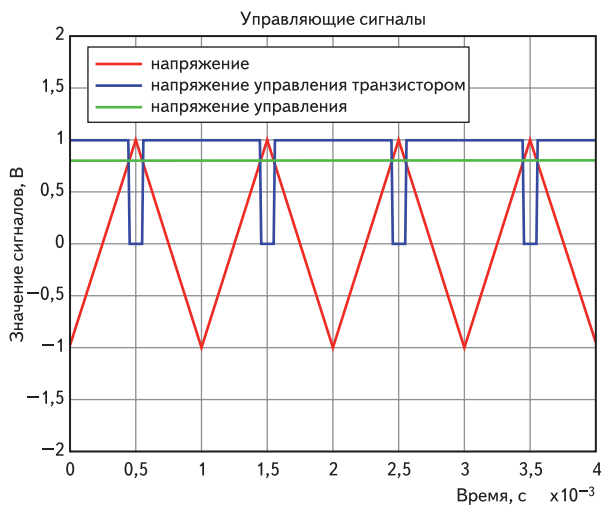


Рис. 15. Сигналы управления на входе и электромагнитные процессы на выходе ШИП

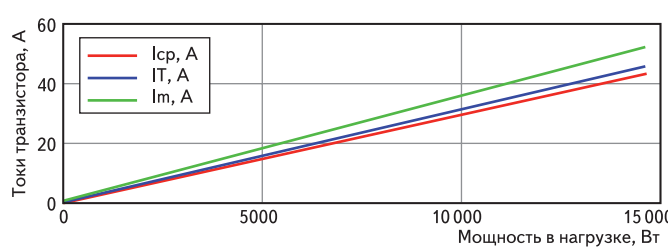
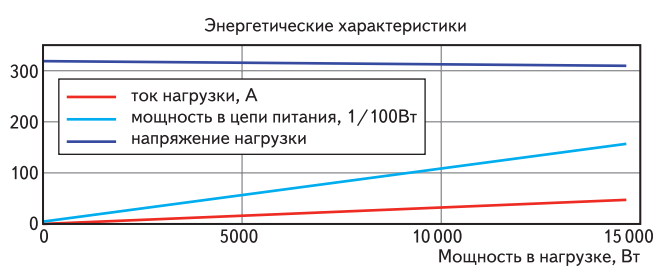


Рис. 16. Основные электромагнитные и энергетические характеристики ШИП

- Регулировочная характеристика (функция `prB3_callback`).
 - Нагрузочная (внешняя) характеристика (функция `prB4_callback`).
 - Энергетические характеристики (функция `prB5_callback`).
 - Спектральная характеристика выходного напряжения, функция `prB6_callback`.
- Способ построения перечисленных процессов и характеристик рассмотрим на примере построения квазиустановившихся процессов управления ШИП, который демонстрирует m-файл, представленный в листинге 5.

- напряжение генератора пилообразного напряжения (ГПН) (цвет красный);
- напряжение управления транзистором (цвет голубой);
- сигнал управления (цвет зеленый).

Сигналы управления на входе и электромагнитные процессы на выходе ШИП, а также основные электромагнитные и энергетические характеристики ШИП, полученные на имитационном лабораторном стенде, представлены на рис. 15, 16.

Реализация GUI в среде Matlab-Simulink может осуществляться с использованием инструментального средства GUIDE, значительно облегчающего разработку проекта [1].

Литература

1. Ануфриев И., Смирнов А., Смирнова Е. MATLAB 7. Наиболее полное руководство. СПб: БХВ-Петербург, 2005.
2. Герман-Галкин С. Г. Школа MATLAB. Урок 6. Программные и инструментальные средства представления результатов моделирования // Силовая электроника. 2007. № 4.
3. Герман-Галкин С. Г. Школа MATLAB. Урок 7. Модельное исследование основных характеристик силовых полупроводниковых преобразователей // Силовая электроника. 2008. № 1.
4. Худяков В. Моделирование устройств силовой электроники. Урок 1. Основные инструменты Simulink // Силовая электроника. 2005. № 1.
5. Худяков В. Школа MATLAB. Урок 2. Библиотека SimPower System // Силовая электроника. 2005. № 2.
6. Худяков В. Школа MATLAB. Урок 3. Построение SPS моделей с полупроводниковыми элементами // Силовая электроника. 2005. № 3.
7. Худяков В. Школа MATLAB. Урок 4. Анализ динамических свойств устройств силовой электроники во временной области // Силовая электроника. 2005. № 4.
8. Худяков В. Школа MATLAB. Моделирование устройств силовой электроники. Урок 5. Анализ устройств силовой электроники в частотной области // Силовая электроника. 2006. № 1.
9. Черных В. Simulink среда создания инженерных приложений. М.: ДИАЛОГ-МИФИ, 2004.

Закключение

Маскированные (виртуальные) блоки и имитационные лабораторные стенды (графические интерфейсы пользователя) разрабатываются для того, чтобы пользователь, не искушенный в вопросах программирования, мог в конкретной технической области провести модельный эксперимент, задав понятные ему параметры технического устройства. Такие модели выполняют роль лабораторного стенда, который позволяет задавать и исследовать любые режимы и характеристики. Эти стенды, разработанные в среде Matlab-Simulink, позволяют реализовать сетевые версии и, тем самым, обеспечить дистанционное исследование различных технических устройств, в том числе, устройств силовой электроники.

Листинг 5

```

01 % prB1_callback
02 function prB1_callback ();
03 global data simin_ster;
04 f=prBpre;
05 repeatX=4;
06 [m maxstep]=pr_m (repeatX);
07 pr2_calc (m/repeatX);
08 opt=simset ('MaxStep',maxstep);
09 [t x y]=sim ('c1', m, opt);
10
11 ax=axes ('Parent', f);
12 pl=plot (ax, t, simout_gen.signals.values, 'r', t,
simout_sim.signals.values, 'b', t, simout_ster.signals.values, 'g');
13 legend (ax, 'напряжение ГПН',
'напряжение управления транзистором', 'сигнал управления', 2);
14 title (ax, 'Управляющие сигналы');
15 xlabel (ax, 'время [с]');
16 ylabel (ax, 'значение сигналов (В)');
17 set (ax, 'YLim', [-2 2]);
18 set (ax, 'XLim', [0 m]);
19 set (ax, 'XGrid', 'on');
20 set (ax, 'YGrid', 'on');
21 prB_activ ('on');
    
```

Обратим внимание, что в строке 12 записано построение трех характеристик: