

УДК 621.311. 078.

**В.И. Гребенщиков, А.И. Чивенков, В.В. Севастьянов, И.А. Захаров**

## **РАЗРАБОТКА СИСТЕМ УПРАВЛЕНИЯ ИНТЕЛЛЕКТУАЛЬНЫХ РЕГУЛЯТОРОВ ПАРАМЕТРОВ КАЧЕСТВА ЭЛЕКТРИЧЕСКОЙ ЭНЕРГИИ**

Нижегородский государственный технический университет им. Р.Е. Алексеева

Рассмотрены вопросы использования универсального инструментария для разработки прототипов систем управления интеллектуальными устройствами силовой электроники с применением пакета программирования нелинейных динамических систем *Simulink/SimPowerSystems* в режиме реального времени по схеме программно-аппаратного тестирования “*Hardware in the Loop*”.

*Ключевые слова:* прототип системы управления, силовая электроника, MATLAB-SIMULINK, SimPowerSystems, Real-Time Windows Target, xPC Target, Hardware in the Loop.

В настоящее время уже существует понятие цифровой силовой электроники. Современные установки регулирования параметров качества электрической энергии на базе импульсных преобразователей находят всё большее применение в промышленных сетях. Появляются многофункциональные устройства, объединяющие в себе свойства регуляторов, компенсаторов, активных фильтров. Оборудование становится всё более сложным с точки зрения аппаратной и информационной частей систем управления.

Многофункциональность силовых устройств регулирования параметров качества электрической энергии позволяет на сегодняшний день создавать интеллектуальные электрические сети, одной из подсистем которых являются гибкие активно-адаптивные устройства передачи, преобразования и регулирования потоков мощности (Flexible Alternative Current Transmission Systems – гибкие системы передачи переменного тока FACTS и Unified Power Flow Controller – унифицированная система управления энергопотоками UPFC). UPFC воплощает мечту комплексного регулирования потоков активной и реактивной мощностей в реальность. Это наиболее сложная из систем. Она позволяет осуществлять следующие функции: непосредственное управление напряжением, сложение или вычитание напряжений, фазированных узлом поперечной компенсации; поперечный компенсатор – путем управления поперечным преобразователем с переводом последнего в режим поглощения или возврата реактивной мощности (напряжение должно поддерживаться постоянным); продольный компенсатор: путем добавления последовательного напряжения, со сдвигом на  $90^\circ$  по отношению к току связи (при этом необходимо управлять выходным напряжением и реактивной мощностью на выходе); фазосдвигающее устройство: если величина и фаза продольного напряжения таковы, что при поддержании такого же напряжения на выходе модуля оно уменьшает величину сдвига фаз по отношению ко входу. Это имеет важное значение при управлении передаваемой активной мощностью.

Или, наконец, одновременное использование всех функций – число степеней свободы системы позволяет это делать, – когда необходимо управлять и реактивной, и активной мощностью.

Один из возможных вариантов реализации UPFC представлен на функциональной схеме, отображенной на рис. 1. Рассмотрена ситуация, при которой питающая сеть 2 является проблемной в плане дефицита электроэнергии и параметров качества напряжения.

При решении комплексной задачи управления параметрами перетоков электрической энергии из одной питающей сети в другую с учётом возможной функциональной взаимозаменяемости подчинённого и ведущего инверторов возникает сложная задача по обработке информационного потока большого количества аналоговых, дискретных каналов измерения

и управления. Можно выделить основные этапы разработки: техническое предложение; разработка и изготовление экспериментального образца; разработка программного обеспечения; разработка и изготовление опытного образца.

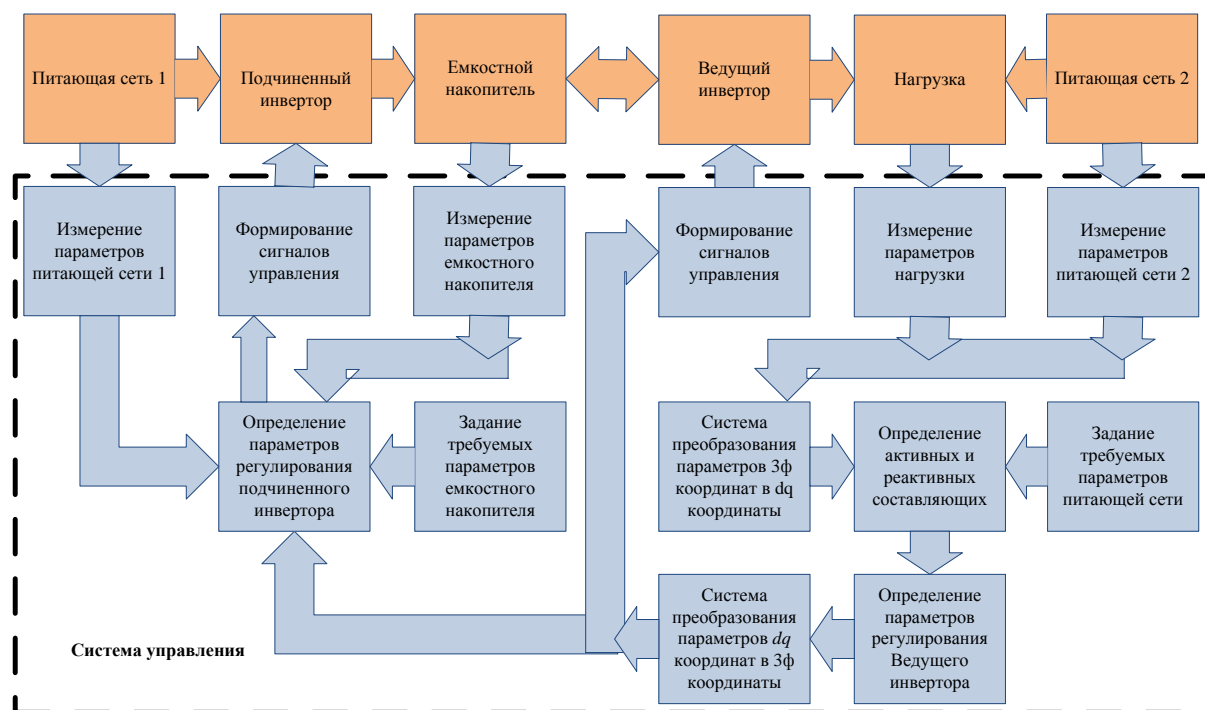


Рис. 1. Структурная схема силовой части UPFC и её системы управления

Длительным и требующим дополнительных финансовых расходов может оказаться второй из перечисленных этапов при разработке и изготовлении физического экспериментального образца из-за многократных переделок и корректировок, включающий в себя:

- разработку схем силовой части установки;
- определение алгоритмов управления активными элементами силовой части;
- разработку схем системы управления, защиты и контроля, в общем случае – системы управления (СУ);
- изготовление макетов силовой части и системы управления;
- исследование работоспособности установки на макете;
- выявление сбоев и возможных аварийных режимов работы;
- корректировку алгоритмов и принципиальных электрических схем установки;
- разработку экспериментального образца установки;
- испытания экспериментального образца.

Возрастающая сложность проектов, а также необходимость совместной работы различных групп проектировщиков, заставляют использовать альтернативные методы, ориентированные на использование новых инструментов проектирования, позволяющих сократить время и стоимость разработки.

Проблема при традиционном подходе к проектированию систем управления состоит в том, что физические прототипы разрабатываемых систем управления являются несовершенными и дорогими. Кроме того, при ручной разработке алгоритма работы системы обычно возникают ошибки в программировании. Дальнейшее тестирование системы управления приводит к обнаружению указанных ошибок, но часто лишь на конечной стадии отладки.

В настоящее время вариант с непосредственным программированием приемлем лишь для простых систем или небольших фрагментов большой системы, для которых нет стан-

дартных решений. Именно поэтому в большинстве случаев представляется целесообразным использовать универсальный инструментарий для разработки управляющего ПО [1].

Предлагаемый метод позволил объединить в непрерывный интерактивный процесс различные этапы разработки системы: имитационное моделирование; разработка системы управления; отладка и тестирование (рис. 2).

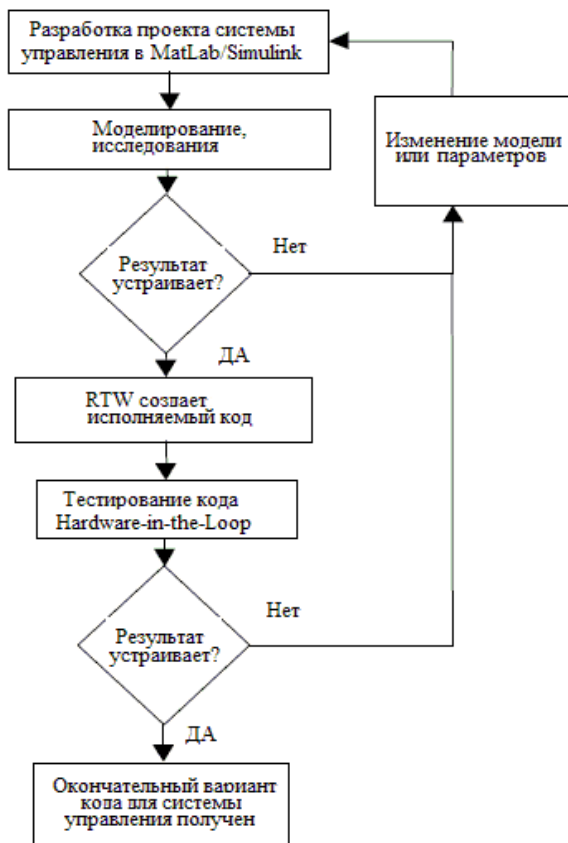
Метод разработан на базе системы реального времени *Real-Time Windows Target*, позволяющей выполнять автоматическую генерацию исполняемого кода и гарантировать отсутствие

ошибок ручного программирования, обеспечивая широкие возможности для последующего циклического тестирования [2].

Для работы на начальном этапе без использования физических объектов была разработана имитационная модель силовой части UPFC на платформе *Matlab/Simulink*, в дальнейшем называемая исполняемой моделью. Для отладки разрабатываемой системы управления преобразователем могут применяться: программное моделирование, процессорно-программное моделирование, аппаратно-программное моделирование.

На рис. 3 показана структурная схема программного моделирования. При таком решении откомпилированный и сгенерированный С-код для локальной операционной системы впоследствии выполняется в имитационной модели. Например, создается *Windows Dynamic Linked Library (DLL)*.

Для программного моделирования используется *Embedded Coder*, который автоматически формирует соответствующую оболочку, вызывающую код. Данная схема тестирования предназначается для отладки системы управления, работающей в режиме реального времени с представленной в *Simulink* моделью управляемого объекта.



**Рис. 2. Интерактивный процесс разработки прототипа системы управления**

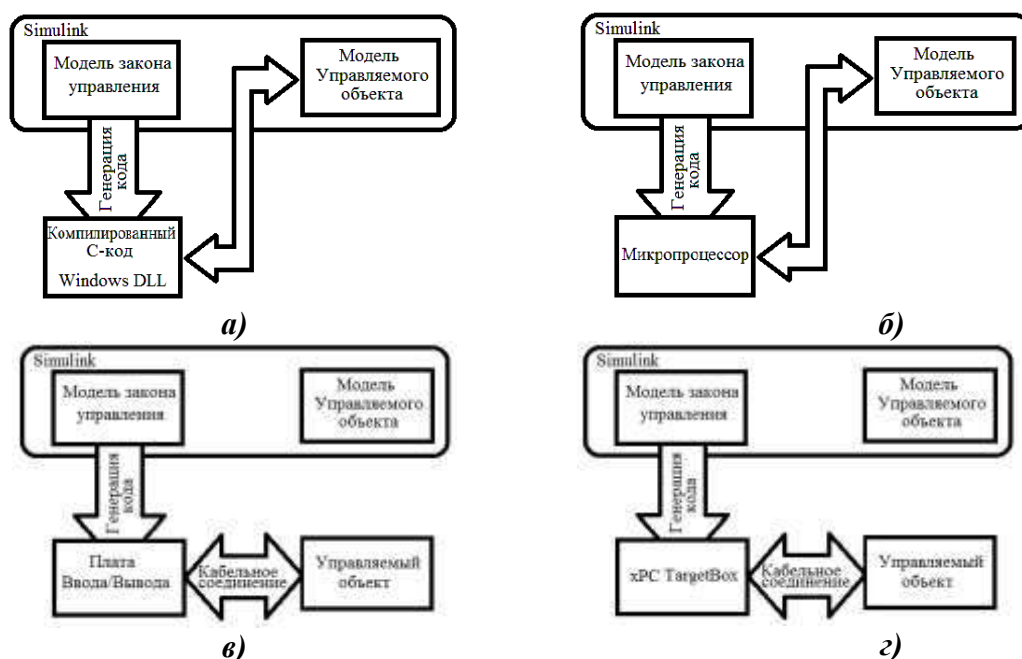
Схема процессорно-программного моделирования приведена на рис. 4. При этом используются: персональный компьютер с созданными моделями закона управления и управляемого объекта; внешний микропроцессор, требующий программирования.

Процессорно-программное моделирование является методом, предназначенным для оценки эффективности тестируемой системы управления. В *Simulink* такая возможность предоставляется пакетом *Embedded Target*. Откомпилированный сгенерированный С-код загружается в выбранный микропроцессор, после чего он выполняется на данном микропроцессоре совместно с моделью управляемого объекта, созданной в *Simulink*.

Схема программно-аппаратного моделирования показана на рис. 5. В этом случае основными компонентами являются: компьютер с моделью закона управления; плата ввода/вывода; управляемый физический объект.

Использование данной схемы является в настоящее время одним из наиболее простых и доступных решений для создания прототипов системы управления. Аппаратно-программное моделирование за рубежом часто называют системой "все в одном" для быстрой разработки прототипов ("*one-box rapid prototyping system*"), поскольку и *Simulink*, и сгенерированный *Real-Time Windows Target* код выполняются на одном и том же компьютере.

В составе *Matlab/Simulink* имеется библиотека *Real Time Windows Target Blokset*. В ней содержатся типовые блоки ввода-вывода “*Analog Input*”, “*Digital Input*” и “*Digital Output*”. Каждый такой блок можно назначить соответствующему каналу платы ввода-вывода.



**Рис. 3. Структура программного моделирования:**

*а* - программное моделирование; *б* - процессорно-программное моделирование; *в* - программно-аппаратное тестирование; *г* - использование *xPC Target*

После размещения соответствующего блока ввода-вывода в модели следует выбрать в его настройках “*Install New Board*”, после чего из выпадающего списка производителей выбрать плату.

Под многозадачной системой *Windows* порой не удается использовать все возможности платы ввода-вывода при высокой частоте ее опроса. В этом случае следует воспользоваться пакетом *xPC Target*. В данном случае создается модель системы управления в *Simulink*, но соответствующие блоки ввода-вывода выбираются из библиотеки *xPC Target*.

Целевой компьютер в дальнейшем будет выполнять задачу в реальном времени и передавать данные на хост-компьютер по *TCP/IP* или *RS232*. Мониторинг работы управляемого объекта и настройка параметров работы системы управления производится на хост-компьютере (рис. 6).

Следует отметить, что пакет *xPC Target* накладывает более жесткие ограничения на число поддерживаемых им плат ввода-вывода по сравнению *Real-Time Windows Target* и использование этого инструмента представляется целесообразным лишь в случае, когда пакет *Real-Time Windows Target* не позволяет получить желаемый результат по быстродействию выполняемых операций из-за недостаточной производительности используемого компьютера.

Алгоритм, иллюстрирующий работу пакета *Real-Time Windows Target (RTW)*, приведен на рис. 4.

Пакет расширения *RTW* позволяет создавать файлы на языке C/C++, а также исполняемые *exe*-файлы для программ системы *Matlab/Simulink*. При этом обеспечивается автоматический подбор необходимых библиотек и компоновку программных модулей с их последующей компиляцией.

На рис. 5 показана последовательность создания требуемого исполняемого файла с *RTW*.

Вначале в *Simulink* подготавливается файл некоторой модели, например, *model.mdl*. Данный файл вместе с системным файлом *system.tmf* поступает в утилиту *Build* пакета *RTW*.

Полученный файл вместе с необходимыми для компиляции системными и библиотечными файлами передается в компилятор *Target Language Compiler (TLS)*. В результате компиляции в указанной в качестве рабочей папке создается набор файлов, в состав которого входит файл *model.c*, задающий точку входа в программу. Все сформированные файлы поступают на вход утилиты *Make*, результатом работы которой является исполняемый файл *model.exe*. Полученный файл может быть вызван как средствами *Matlab*, так и средствами операционной системы.

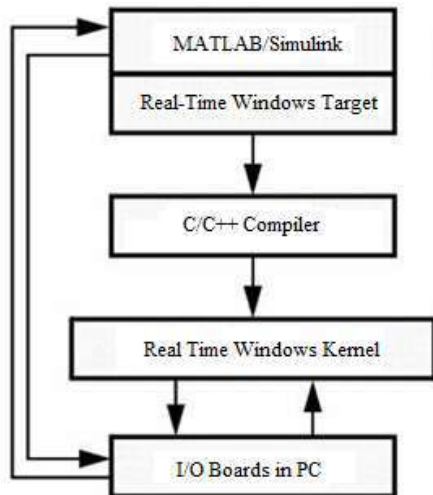


Рис. 4. Алгоритм работы *Real-Time Windows Target*

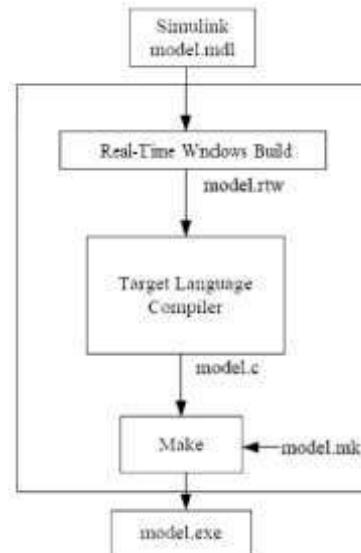


Рис. 5. Алгоритм работы пакета *Real-Time Windows Target*

Файл *model.exe* может быть использован для работы с платами расширения, имеющими встроенное программное обеспечение. Таким образом создается замкнутая система управления внешними устройствами. Режим внешнего управления позволяет осуществлять динамическую настройку параметров исполнения, созданных пакетом *RTW* кодов за счет применения *Application Program Interface (API)* операционной системы *Windows*. В результате получается среда разработки программного кода, обычно называемая средой быстрого макетирования (*rapid prototyping program framework*).

Процесс создания исполняемых кодов моделей имеет некоторые ограничения. Для того чтобы модель системы управления экспортировать в *dll*-библиотеку, необходимо исключить из нее компоненты, производящие какие-либо операции ввода/вывода, также компоненты, для которых в системе *Matlab/Simulink* отсутствуют исходные коды или компиляция которых невозможна. Например, в состав компилируемых моделей нельзя включать *S*-функции. Обязательным условием работы модели в режиме реального времени является исключение из ее структуры всех алгебраических петель. Под алгебраической петлей понимается необходимость подачи сигнала в систему на текущем шаге интегрирования из последующего. Необходимо, чтобы модель системы управления в *Simulink* обязательно была дискретной и ее следует запускать в режиме *External* при заданном дискретном шаге опроса.

Работа над созданием прототипа системы управления *UPFC* состояла из следующих основных этапов:

- моделирование систем и подсистем *UPFC* в пакете *Matlab/Simulink*;
- проектирование закона управления *СУ* и имитация его работы на модели преобразователя;
- реализация *СУ* на базе промышленного компьютера с использованием пакета реального времени *Real-Time Windows Target*;
- проверка и тестирование разработанной системы управления в режиме реального времени по схеме программно-аппаратного тестирования на промышленном компью-

тере *iROBO-5000-93i3* с установленной операционной системой *Windows 7 Professional 32bit*.

По окончании проверочных и отладочных работ функционирование системы управления было испытано на реальном опытном образце UPFC мощностью 20 кВА.

Рассмотренный пример по разработке системы управления UPFC может быть успешно применён при создании практически любых устройств преобразования параметров качества напряжения электрических сетей. Таким образом, при разработке прототипов и алгоритмов СУ преобразователями наиболее целесообразным представляется их реализация и настройка на имитационной модели, а не на физической аппаратной установке системы управления.

Такой подход позволяет: во-первых создать законченную работоспособную СУ; во-вторых - непосредственно перейти к управлению реальным объектом сразу же после окончания исследований, исключая дополнительные затраты на адаптацию при внедрении разработанной системы. Кроме того, он открывает большие возможности по созданию новых алгоритмов СУ и их отладки в лабораторных условиях без конструктивных изменений.

#### Библиографический список

1. <http://old.lvk.cs.msu.su/index.php/articles/710>, "A Hardware-in-the-Loop Simulation Environment for Real-Time Systems Development and Architecture Evaluation"
2. <http://www.mathworks.com, products, xpctarget>, "xPC Target Getting Started Guide R2012A"

*Дата поступления  
в редакцию 15.12.2014*

**V.I. Grebenshikov, A.I. Chivenkov, V.V. Sevastyanov, I.A. Zakharov**

#### **DEVELOPMENT OF CONTROL SYSTEMS OF INTELLIGENT CONTROLLERS ELECTRICAL ENERGY QUALITY PARAMETERS**

Nizhny Novgorod state technical university n.a. R.E. Alexeev

**Purpose:** development of control systems of intelligent controllers

**Desing/methodology/approach:** questions of use of the universal tools for prototyping control systems of power electronics using with a packet of programming of non-linear dynamic systems Simulink/SimPowerSystems in real time according to the scheme of hardware-software testing of "Hardware in the Loop" are considered.

**Findings:** the approach allows to create a complete functional SU b can go directly to the management of the real object immediately after the end of studies.

**Ressearch limitations/implications:** control systems of intelligent controllers electrical energy quality parameters.

**Originality/value:** the approach opens up great opportunities to create new algorithms su and debug them in laboratory conditions without structural changes of the physical installation

*Key words:* The prototype control system, power electronics, MATLAB-SIMULINK, SimPowerSystems, Real-Time Windows Target, xPC Target, Hardware in the Loop.