электронный журнал
МОЛОДЕЖНЫЙ НАУЧНО-ТЕХНИЧЕСКИЙ ВЕСТНИК
Издатель ФГБОУ ВПО "МГТУ им. Н.Э. Баумана". Эл No. ФС77-51038.

УДК 621.396.621

Разработка программного обеспечения для стенда полунатурного моделирования бортовой информационно управляющей системы

***Ерченко К.А.**, специалист*
Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана,
кафедра «Автономные информационные и управляющие системы»

Научный руководитель: Кондрашов К.С., ассистент
Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана
bauman@bmstu.ru

Бортовая информационная и управляющая система высокомобильной модульной платформы (БИУС ВМП) представляет собой сложную вычислительную систему реального времени. Разработка такой системы включает в себя этапы создания алгоритмов, разработки аппаратной части, этап комплексирования систем, проверку и тестирование. Ввиду сложности данной системы, ошибки, возникающие на данных этапах не всегда возможно выявить проверкой отдельных модулей или отдельных алгоритмов. Зачастую ошибки возникают на этапе комплексного взаимодействия всех узлов. Однако, тестирование системы на самой платформе связано с определенными проблемами: организационными, экономическими, проблемами безопасности, к тому же не всегда возможно воспроизвести для проверки критические условия эксплуатации. Для отработки алгоритмов БИУС, отладки программного обеспечения БИУС, тестирования аппаратных модулей разрабатывается комплексный стенд полунатурного моделирования БИУС. Он должен моделировать работу агрегатов ВМП и её динамики, учитывая изменение внешних условий (тип дорожного покрытия и т.д.). На рисунке 1 изображена обобщенная функциональная схема отладочного стенда с указанием основных компонентов программного обеспечения. Базовая среда моделирования обеспечивает подключение стенда к цифровым и аналоговым линиям связи, а также общее управления программными модулями, моделирующими работу отдельных агрегатов и подсистем ВМП. Удельный вес программных модулей, реализующих пользовательский интерфейс, и модулей, предназначенных для моделирования управляющих воздействий и управления параметрами моделирования сравнительно невелик, поэтому основной проблемой является создание программных модулей имитации работы собственно агрегатов и

подсистем ВМП. Разработка таких программных модулей «с нуля» чрезвычайно трудоемка, поскольку для обеспечения адекватной имитации поведения ВМП в различных режимах требуется использовать достаточно сложные модели, описываемые нелинейными дифференциальными и разностными уравнениями с большим числом переменных состояния и взаимных связей. Эти модели реализованы в среде *Matlab/Simulink*, поэтому существенного упрощения работы по созданию программного обеспечения стенда моделирования БИУС ВМП можно добиться путем создания технологии автоматической или полуавтоматической генерации кода на C++ – языке реализации программного обеспечения стенда из таких моделей в среде Matlab/Simulink.

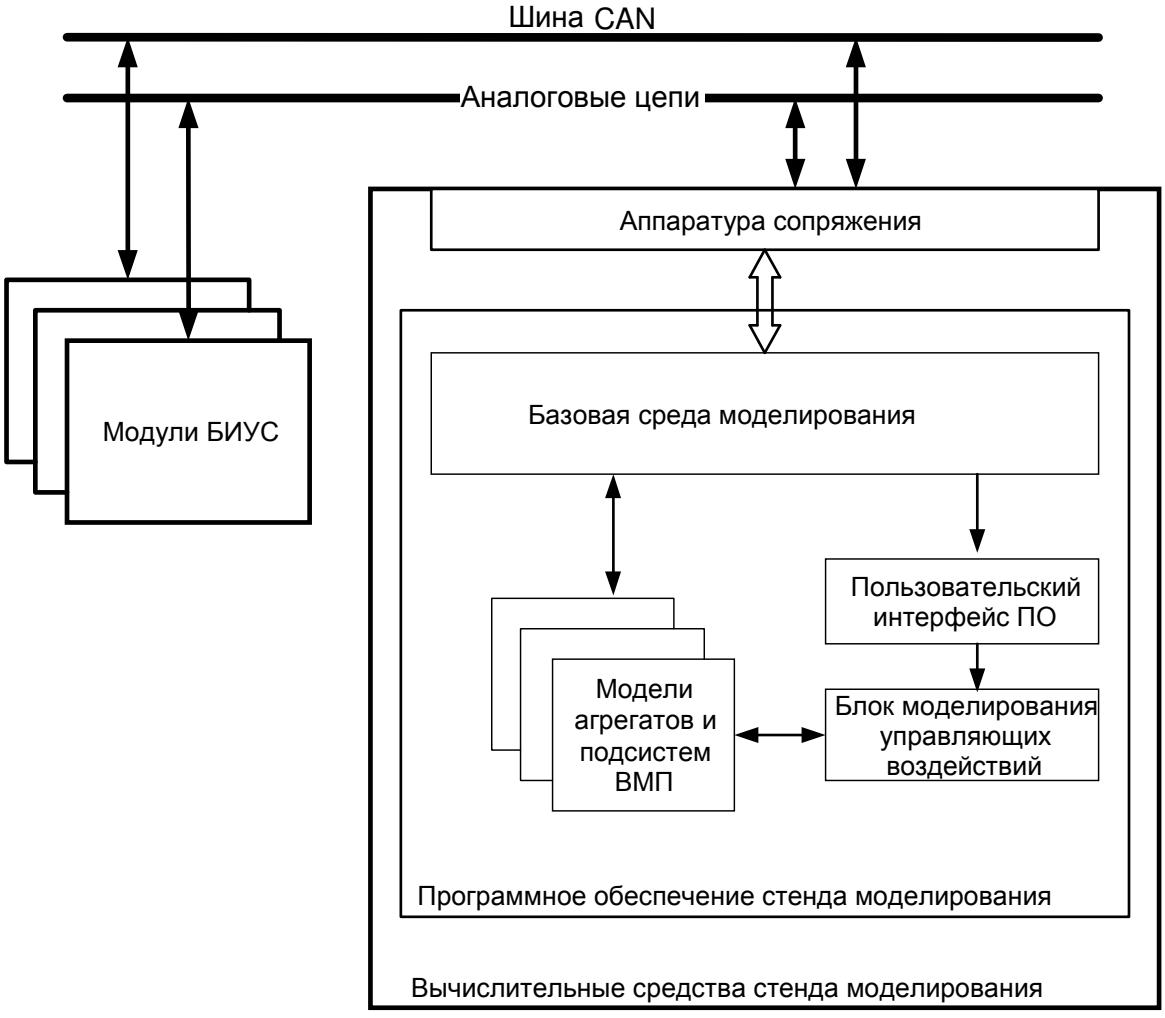

Рис. 1. Структура стенда полунатурного моделирования.

Далее рассматривается программная организация отладочного стенда в соответствии со структурой, изображенной на рис. 1. Модули отладочного стенда, реализующие функциональность моделей агрегатов и окружающей среды, представляют

собой внутренние *COM*-серверы в формате dll-библиотек. . Базовая среда моделирования вызывает исполнение кода *COM*-компонента в определенные моменты временив соответствии с заданными входными воздействиями и принимает данные с его выходов. Также она осуществляет обмен информацией по устройствам сопряжения с модулями БИУС. Оператор имеет возможность управлять процессом моделирования и анализировать результаты работы модели. Работа с *COM*-компонентом организована при помощи нескольких функций.

Метод *InitializeModel* реализует процедуру инициализации модели и возвращает указатели на внутренние буферы ввода-вывода и параметров в аргументах. Метод *Operate* запускает один цикл имитации. Метод *Reset* перезапускает инициализацию модели, что приводит, в частности, к сбросу параметров к начальным значениям. Служебные методы *Initialized* и *GetName* предназначены для возврата признака «проинициализирована» и имени модели соответственно. Методы *GetExternalStep* и *SetExternalStep* предназначены для установки размера шага, с которым предполагается вызывать модель. При каждом вызове функции *Operate* в ее теле будет выполняться ровно столько шагов интегрирования, сколько необходимо для корректности данных на выходе блоков с непрерывными состояниями на момент очередного окончания *внешнего* шага. Таким образом, внешний шаг должен быть кратен внутреннему постоянному шагу интегрирования.

Связь формата *COM* и моделей в *MATLAB/Simulink*.

COM-сервер и приведённые выше методы работы с ним выступают лишь в роли удобного средства реализации программного модуля. Сам же функционал моделируемого окружения БИУС сгенерирован в виде кода на языке *C/C++*. MATLAB/Simulink позволяет получать из моделей Simulink исходный код на выбранном языке (среди прочих, поддерживаются C и C++) и автоматически компилировать программы и библиотеки для различного окружения посредством программного пакета Simulink Coder. Ниже рассмотрены функции из сгенерированного таким образом кода, используемые *COM*-компонентом.

Ниже приводится сводка *C*-функций, представленных в сгенерированном посредством *Simulink Coder* коде, который предназначен для ускоренного моделирования (как уже было упомянуто, именно данный код используется в модулях отладочного стенда):

- *Model*() – функция регистрации модели. Она отвечает за инициализацию рабочих областей (распределение памяти и сопоставление указателей различным

структурам данных), необходимых для модели. Данная функция вызывает *MdlInitializeSizes* и *MdlInitializeSampleTimes*.

- *MdlStart(void)* – запуск модели на вычисление. Вызывается один раз. Имеет следующие секции:
  - Код для инициализации состояний каждого блока, рекурсивно вызываемый из корня каждой подсистемы (и модели в целом), которая имеет состояния.
  - Код, соответствующий старту работы каждого из блоков.
  - Код, разрешающий выполнение блоков в корне модели, которые имеют метод *enable*, а также для таких блоков, включенных в состав триггерных или функциональных подсистем в составе корневой модели.
  - Код для обновления выходов каждого блока, имеющего постоянный шаг по времени.
- *MdlOutputs(int_T tid)* – отвечает за обновление состояния блоков в нужное время. *tid* (идентификатор задачи) позволяет различать, к какому из параллельных потоков (задач) относится данный блок. Данная процедура вызывается драйвером выполнения модели на старших и младших шагах по времени.
- *MdlUpdate(int_T tid)* – отвечает за обновление дискретных состояний и информации в рабочих векторах (которые присутствуют для каждого экземпляра блоков, подсистем и проч.). *tid* выполняет ту же роль, что и в *MdlOutputs*. Кроме того, данная функция должна вызываться после нее.
- *MdlDerivatives(void)* – отвечает за расчет производных в блоках с непрерывным временем. Вызывается на младших шагах интегратором дифференциальных уравнений. Каждый блок, имеющий непрерывные состояния, предполагает наличие одинакового количества производных, которые необходимо проинтегрировать. Данная функция необходима для корректной работы интегратора дифференциальных уравнений.
- *MdlTerminate(void)* – вызывает любой код, включенный в блоки модели, который должен быть выполнен в конце его работы.

Выполнение модели начинается с фазы инициализации. Далее модель выполняется итеративно, инкрементируя модельное время на размер главного шага на каждой итерации. В процессе выполнения сначала запускается *ModelOutputs* в момент *t*, затем осуществляется журналирование, а затем *ModelUpdates* обновляет дискретные состояния. Далее, если модель содержит блоки, работающие в непрерывном времени, *ModelDerivatives* осуществляет обновление производных, чтобы получить состояние

блоков на момент $t' = t + h$, где h – длина шага. При этом модельное время приобретает значение t', и процесс повторяется.

Работа с модельным кодом, сгенерированном посредством Simulink Coder, может быть описана диаграммой, представленной на рисунке 2. Таким образом, код, сгенерированный посредством *Simulink Coder*, содержит все процедуры, необходимые для реализации функциональности COM-компонента стенда и могут напрямую вызываться в соответствующих функциях.

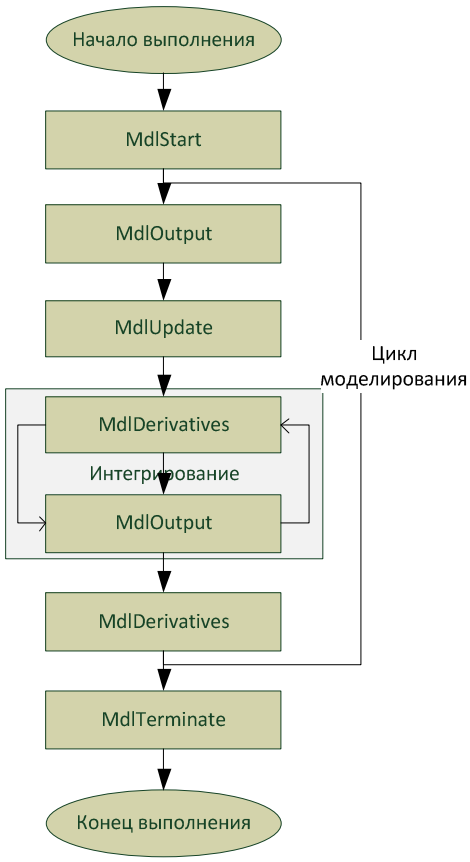

Рис. 2. Диаграмма выполнения кода ускоренной имитации

Simulink Coder , используемый для генерации C/C++-кода, описанного выше, включает набор сценариев для программы TLC (Target Language Compiler), определяющих процесс генерации кода для каждого блока Simulink. А именно, данные сцнеарии (.tlc-файлы) содержат правила, по которым совершается обход древовидных структур, описанных в .rtw-файле, создаваемом для каждой модели в Simulink автоматически. В процессе работы TLC выполняет директивы, содержащиеся в .tlc-файлах и выводит результат в поток вывода, который в данный момент подключен. Таким

образом, перенаправляя поток вывода в файл, предоставляется возможность генерировать исходный текст на любом языке или просто служебные файлы. Код, генерируемый в Simulink Coder не является кодом COM-компонентов непосредственно. Для его использования требуется совершить ряд действий «вручную», что неудобно в случае, если моделей много и их нужно поддерживать в актуальном состоянии. Для автоматизации таких действий для *Simulink Coder* была написана программа, позволяющая из среды MATLAB автоматически получать *COM*-компоненты в формате внутренних *COM*-серверов, а также библиотеки статического связывания (для технологических нужд).

Рассмотрит пример использования описанной технологии

Рассмотрим пример использования описанной технологии. Для демонстрации была использована простейшая модель электродвигателя без модели контроллера управления. *Simulink* модель представлена на рис. 3.

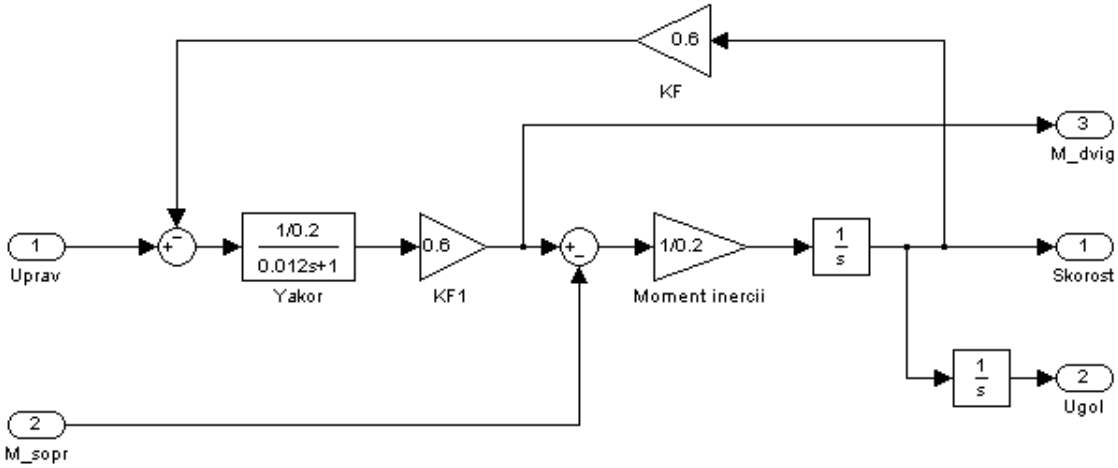

Рис. 3. Модель электродвигателя

Шаг по времени выбран 0.001 с. Численное интегрирование производится методом Рунге-Кутты 4 порядка.

Полученный в результате процедуры кодогенерации компонент представлен на рис. 4. Данный модуль идентичен по интерфейсу исходной модели. На вход модели поступает сигнал напряжения на обмотке и момент сопротивления. Выходами модели служат сигналы вращающего момента двигателя, угловой скорости, угла поворота.

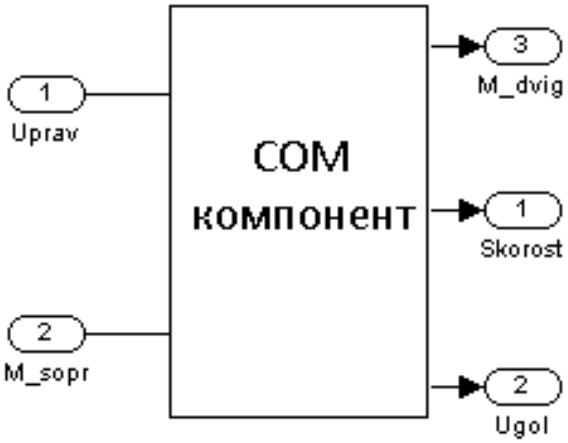

Рис. 4. *COM* компонент исходной модели.

Моделирование проводилось в течение 5 секунд. На 1 секунде на обмотку было подано напряжение 7В. Момент сопротивления оставался нулевым на протяжении всего моделирования. Результаты моделирования по соответствующим выходам представлены на рис. 5.

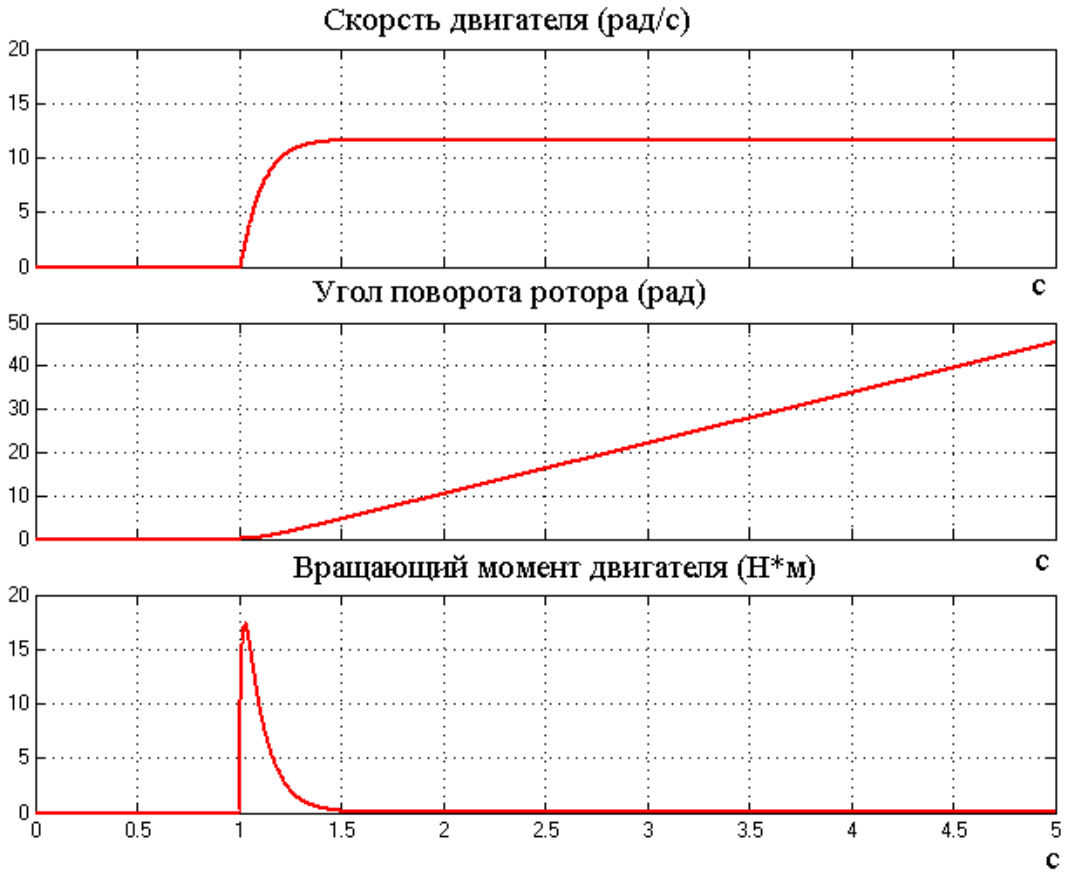

Рис. 5. Результаты моделирования

Результаты работы программного модуля совпадают с результатами моделирования *Simulink* модели.

Получен программный модуль, реализующий функционал агрегатов ВМП. Он функционирует на платформе Windows и может использоваться для внедрения в системы моделирования, реализованные на различных языках программирования. Данный файл динамический библиотеки можно добавить на компьютер, выполняющий функции стенда полунатурного моделирования. При необходимости редактирования модели агрегатов и подсистем БИУС можно использовать данную технологию к изменённой Simulink модели и получить новый *COM* компонент без лишних трудозатрат разработчика. В стенде необходимо заменить dll файл.

Сгенерированный при использовании данной технологии набор программных модулей позволяет производить моделирование в реальном времени, если выполнение кода на каждом итерационном шаге будет выполняться с привязкой к временной сетке. Те же самые *модули* будут функционировать и в модельном времени, при этом изменение его кода и повторная генерация не требуется. Достаточно лишь выполнять моделирование под управлением соответствующей среды, которая уже реализована. Данный режим позволяет в зависимости от сложности системы существенно сократить время моделирования по сравнению с исходной моделью в MATLAB/Simulink.

Имея подобные наработки можно расширить возможности стенда под операционные системы реального времени, так как использование Windows не всегда гарантирует моделирование в реальном времени.

Список литературы

1. Эванс Э. Предметно-ориентированное проектирование (DDD): структуризация сложных программных систем. М.: Вильямс. 2011. 448 с.
2. Васин В.А., Власов И.Б., Егоров Ю.М. Информационные технологии в радиотехнических системах: учеб. пособие / под ред. И.Б. Федорова. М.: Изд-во МГТУ им Н.Э.Баумана, 2003. 672 с.: ил. (Сер. Информатика в техническом университете.)
3. Дьяконов В.П. MATLAB и SIMULINK для радиоинженеров. М: ДМК Пресс, 2011. 976 с.