

УДК 681.51

МОДЕЛЬНО-ОРИЕНТИРОВАННЫЙ ПОДХОД ДЛЯ АВТОМАТИЗАЦИИ ГЕНЕРАЦИИ ПРОГРАММНОГО С-КОДА ДЛЯ ВСТРАИВАЕМЫХ СИСТЕМ ИЗ МОДЕЛИ MATLAB/SIMULINK

¹Оськин Д.А., ²Громашева О.С., ¹Дьяченко М.Е.

¹Морской государственный университет имени адмирала Г.И. Невельского, Владивосток, e-mail: daoskin@mail.ru;

²Тихоокеанский океанологический институт им. В.И. Ильичева ДВО РАН, Владивосток, e-mail: gromasheva@poi.dvo.ru

В работе предложено использование модельно-ориентированного подхода для автоматизации проектирования встраиваемых систем. Вычислительные устройства, используемые в подобных системах, реализованы на микроконтроллерах или микропроцессорах. Разработка программного обеспечения для встраиваемых систем отличается сложностью. Это связано в первую очередь с существенным многообразием современных микроконтроллеров и различием их архитектурных особенностей. Также в процессе разработки существенное влияние оказывает человеческий фактор. Автоматизация проектирования встраиваемых систем предлагает использование модельно-ориентированного подхода для генерации программного кода, используемого на целевых процессорах (контроллерах). Для приведенного подхода рассмотрены уровни конфигураций тестирования, а именно «модель в контуре», «программа в контуре», «процессор в контуре» и «аппаратное обеспечение в контуре». Приведена и описана методика генерации С-кода для встраиваемых систем. В качестве демонстрации приведена пошаговая реализация модельно-ориентированного подхода для модели, реализованной в среде Matlab/Simulink. В среде моделирования реализована модель замкнутой системы управления одномерным линейным объектом. Для нелинейного устройства управления, реализующего алгоритм «super-twisting», приведено пошаговое исполнение модельно-ориентированного подхода. Предложенный подход позволяет снизить количество ошибок, допускаемых при проектировании, программировании и тестировании подобных систем.

Ключевые слова: модельно-ориентированный подход, встраиваемая система, «модель в контуре», «программа в контуре», «процессор в контуре», «аппаратное обеспечение в контуре», «super-twisting» алгоритм

MODEL-ORIENTED APPROACH FOR THE AUTOMATION OF THE SOFTWARE C-CODE GENERATION FOR EMBEDDED SYSTEMS USING MATLAB / SIMULINK MODEL

¹Oskin D.A., ²Gromasheva O.S., ¹Dyachenko M.E.

¹Admiral Nevelskoy Maritime State University, Vladivostok, e-mail: daoskin@mail.ru;

²V.I. Il'ichev Pacific Oceanological Institute Far East Branch of Russian Academy of Science, POI FEB RAS, Vladivostok, e-mail: gromasheva@poi.dvo.ru

The paper suggests the use of a model-oriented approach for automating the design of embedded systems. The computing devices used in such systems are implemented on microcontrollers or microprocessors. The development of software for embedded systems is difficult. This is due, first of all, to the significant diversity of modern microcontrollers and the difference in their architectural features. Also, in the development process, the human factor has a significant impact. Automation of embedded system design offers the use of a model-oriented approach for generating program code used on target processors (controllers). For the above approach, the levels of test configurations are considered, namely the «model in the loop», the «program in the loop», the «processor in the loop» and the «hardware in the loop». The method of generating C-code for embedded systems is described and described. As a demonstration, a step-by-step implementation of the model-oriented approach for the model implemented in the Matlab / Simulink environment is presented. In the simulation environment, a closed-loop control model for a one-dimensional linear object is implemented. For a non-linear control device implementing the «super-twisting» algorithm, a step-by-step execution of the model-oriented approach is presented. The proposed approach allows reducing the number of errors allowed in the design, programming and testing of such systems.

Keywords: embedded system, model-oriented approach, «model in the loop», «program in the loop», «processor in the loop», «hardware in the loop», «super-twisting» algorithm

Разработка современных систем управления сложными динамическими объектами (мехатронными, робототехническими и др.) напрямую связана с задачами моделирования, проектирования, тестирования и создания рабочих прототипов управляющих систем.

Система, которая имеет в своем составе устройство управления, выполнен-

ное на основе вычислительного устройства, называется встраиваемой системой (англ. Embedded System). Вычислительные устройства могут быть выполнены на базе микропроцессора (МП) или микроконтроллера (МК).

При проектировании и реализации МК-систем управления сложными динамическими объектами одной из актуальных про-

блем является автоматизация разработки управляющих программ, реализующих синтезированные алгоритмы управления.

Традиционно процесс проектирования систем управления можно разделить на несколько этапов: создание математической модели; разработка алгоритмов непрерывного и логико-программного управления; подготовка соответствующих технических требований и спецификации для конструкторов, разрабатывающих электрические схемы и конструкции встраиваемого устройства; разработка программного обеспечения для встраиваемых систем.

Компания MathWorks предлагает новый подход для проектирования – модельно-ориентированное проектирование (МОП), реализуемое в среде MATLAB/Simulink совместно с пакетом Simulink Coder [1]. Необходимо отметить, что полученный с помощью МОП программный код может быть использован для решения задач в приложениях реального времени, делает более эффективным сам процесс моделирования, позволяет производить быстрое прототипирование, оптимизировать аппаратно-программное тестирование разрабатываемых систем.

Целью настоящей статьи является обобщение принципов модельно-ориентированного подхода при проектировании встраиваемых систем. Для проведения практических исследований в работе предлагается использовать методы моделирования и проектирования в среде Matlab с использованием пакета Simulink Coder.

Модельно-ориентированное проектирование

МОП – эффективный и простой способ разработки МП-систем управления. Основой МОП является модель управляемого объекта или процесса. Соответственно, использование в качестве платформы разработки пакета Simulink Coder дает возможность проводить имитационное моделирование как на уровне отдельных элементов, так и на интеграционном уровне для всей системы управления. Автоматическая генерация программного кода ведет к уменьшению ошибок, связанных с человеческим фактором и, что не менее важно, приводит к сокращению времени разработки. Такие необходимые и достаточно сложные этапы разработки, как тестирование и верификация, проводятся также на основе технических требований, но начинаются с уже процесса генерации программного кода. Тестирование в МОП практически осуществляется уже в процессе моделирования, при создании прототипов программ, то есть на всех этапах, вплоть до уровня аппаратной реализации [1].

Применение МОП для микроконтроллерных систем

МОП также используется для сокращения времени разработки встраиваемых систем. Данная методология разработки позволяет производить надежный продукт за короткое время разработки. МОП в основном используется для тестирования и известен как «в контуре» (In-The-Loop). Рассмотрим четыре уровня конфигураций тестирования:

- Model-In-the-Loop (MIL) – «модель в контуре»,
- Software-In-the-Loop (SIL) – «программа в контуре»,
- Processor-In-the-Loop (PIL) – «процессор в контуре»,
- Hardware-In-the-Loop (HIL) – «аппаратное обеспечение в контуре».

Каждый из уровней конфигурации сокращает время в процессе разработки, которая начинается с построения математической модели и заканчивается запуском программного обеспечения на автономной встраиваемой платформе [2].

МОП начинается с этапа MIL, при этом реализуется имитационная модель системы управления и управляемой модели объекта (процесса) управления. Основной целью MIL является тестирование и проверка математической модели.

На этапе SIL имитационная модель, синтезированная на предыдущем этапе, заменяется исполняемым кодом, запущенным на той же аппаратной платформе, где используются вычисления с фиксированной запятой. В среде Matlab для программного уровня тестирования (SIL) создается блок S-функции, что позволяет перейти к использованию в Simulink модели, производить подбор и настройку необходимых параметров модели. При этом происходит наследование: выполнение сгенерированного и откомпилированного C-кода на данном этапе выполняется в имитационной Simulink модели.

Следующий уровень (PIL) определяет наиболее ответственный в практическом применении уровень тестирования. Этот этап предназначен для оценки параметров эффективности разрабатываемой системы управления на уже конкретном, выбранном для проекта целевом процессоре. Производится загрузка сгенерированного программного кода на целевую отладочную плату. Имитационная модель, запущенная на ПК, обменивается данными с загруженным на аппаратную платформу программным обеспечением, как правило, с помощью последовательных линий связи. Таким образом,

обеспечивается обмен тестовыми сигналами, генерируемыми в имитационной модели, и программным обеспечением, установленным в процессоре тестовой платы. Тестирование программного обеспечения в реальном времени не может быть выполнено на этапе PIL, этот шаг выполняется на этапе HIL.

Преимущества PIL заключаются в низкой стоимости по сравнению с HIL; возможностью проводить произвольные испытания; в высокой точности при тестировании встраиваемых алгоритмов; в отсутствии ограничений по сложности модели оборудования (по сравнению с HIL). Кроме того, на этапе PIL можно вести учёт многопоточности контроллера. Недостатком этого этапа является отсутствие теста охватывать только программное обеспечение и процессор.

Последним этапом разработки встраиваемой системы является HIL. На данном этапе выполняется эмуляция датчиков и исполнительных механизмов в режиме реального времени для целевой платформы до того, как к контроллеру будут подключены реальные датчики и исполнительные механизмы. Процесс эмуляции используется для взаимодействия модели оборудования

и интегрированной системы при испытании всех их на одной платформе.

В настоящее время наиболее известны компании Mathworks и National Instruments, разработавшие инструменты реализации модельно-ориентированных платформ, которые позволяют выполнять рассмотренные этапы моделирования контроллера.

Как иностранные, так и российские производители микроконтроллерных систем уже активно используют МОП. Компания ABB применила продукты корпорации MathWorks для проектирования разработки и проверки программного обеспечения для электронной системы управления силовых преобразовательных элементов [1]. В учебно-научном центре «Интеллектуальные системы» при МГТУ им. Н.Э. Баумана занимаются разработкой и внедрением промышленных систем управления на основе отечественных и зарубежных контроллеров и SCADA [3]. Российской компанией «Центр Инженерных Технологий и Моделирования «Экспонента» для микроконтроллеров-аналогов STM CortexM3 выпущен пакет целевой поддержки контроллеров Миландр для Simulink и Embedded Coder, обеспечивающий поддержку МОП на основе платформы MATLAB/Simulink [4].

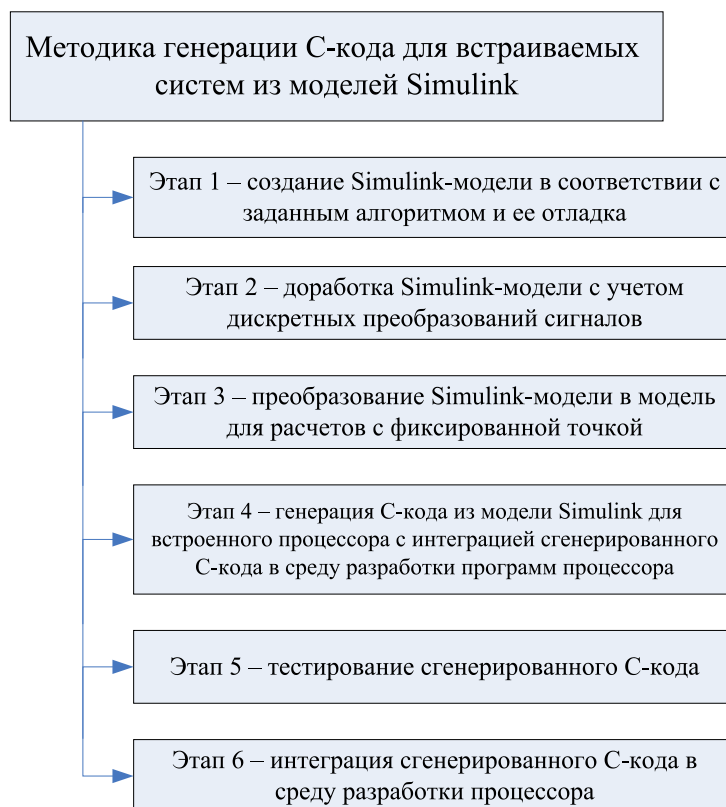


Рис. 1. Методика генерации С-кода для микроконтроллерных систем

Модельно-ориентированное проектирование в MATLAB

Основой для МОП является пакет прикладных программ MATLAB и его Simulink-приложение, предназначенное для визуального моделирования динамических систем. Simulink предназначен для имитационного моделирования и анализа различных динамических систем [5].

МОП отличаются от традиционной методики проектирования управляющих систем. Основным принципом данной методики является использование стандартных функциональных блоков для оптимизации характеристик имитационной модели. Применяемые стандартные блоки могут быть выбраны для режимов как для непрерывного, так и для дискретного времени. Главное преимущество МОП состоит в том, что данный подход обеспечивает взаимодействие разработчиков в процессе анализа данных и проверки системы. На ранних стадиях проектирования системы управления можно обнаружить и свести к минимуму ошибки. Это приводит к минимизации временных затрат и последствий изменения системы. МОП способствует повторному использованию моделей для модификации системы и создания систем с расширенными возможностями [6].

Методика генерации программного С-кода из модели Simulink включает следующие этапы (рис. 1) [7–9].

Рассмотрим возможность генерирования программного кода на примере нелинейной системы управления одномерным динамическим объектом.

Пример генерации программного кода

На рис. 2 представлена структурная схема системы, на рис. 3 – структурная схема управляющего контроллера. Она включает в себя объект управления (блоки TransferFcn и Integrator) и контроллер (Controller).

Математическое описание блока Controller (формирующего управляющий сигнал) представляется системой нелинейных уравнений, реализующих один из релейных алгоритмов управления «super-twisting» [10]:

$$e = K3 * (qd - q) - w,$$

$$u = K1 * \sqrt{|e|} * \text{sign}(e) + v,$$

$$\dot{v} = K2 * \text{sign}(e).$$

На начальном этапе по математическому описанию модели в MATLAB/Simulink создается структурная схема непрерывной модели системы управления.

Следующий этап – переход к цифровой модели регуляторов. Построение цифровой системы требует корректировки модели системы управления. Значение времени дискретизации необходимо внести в поля «Sample time» блоков модели. Цифровая система управления приведена на рис. 4.

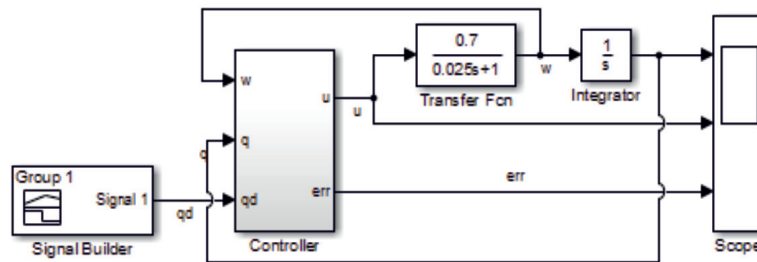


Рис. 2. Пример Simulink системы управления

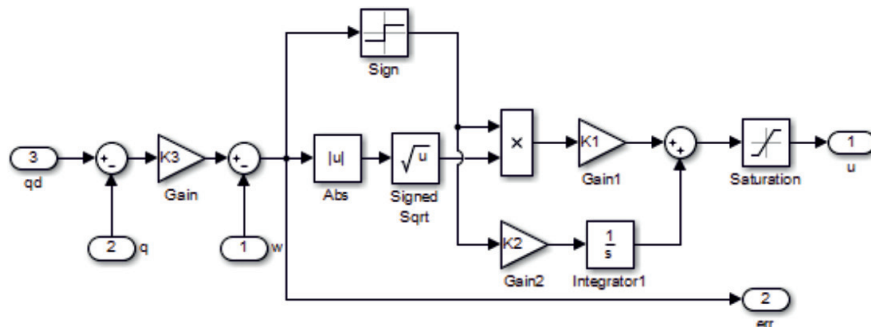


Рис. 3. Структура блока Controller

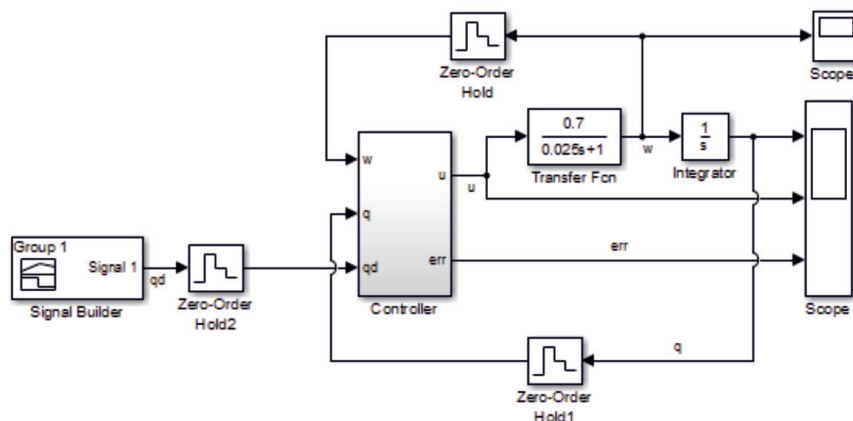


Рис. 4. Цифровая САУ

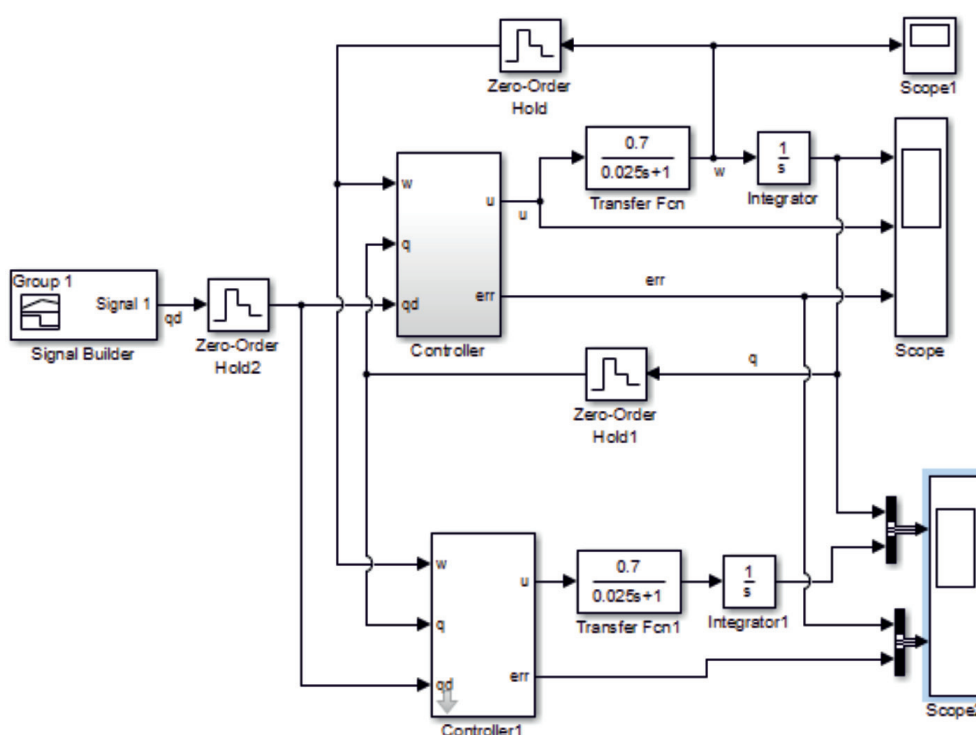


Рис. 5. Структурная схема цифровой САУ (сверху) и САУ с SIL (снизу)

После оценки качества переходных процессов приведенной модели в цифровую форму этап «MIL» считаем завершенным и переходим к тестированию в режиме SIL, генерируем программный С-код регуляторов для конкретной аппаратной платформы.

Тестирование SIL. Генерация С-кода разработанных регуляторов

Проведем настройки целевой аппаратной платформы. Воспользуемся возможностью настройки параметров модели Simulink *Simulink Preferences*. Во вкладке

Hardware Implementation необходимо выбрать целевую платформу и ознакомиться с размером типов данных на выбранной платформе.

Во вкладке *Code Generation* выберем утилиту для получения программного кода и язык программирования, на котором этот код будет написан.

Для проведения тестирования в режиме SIL выберем блок *Controller*, выберем вкладку *C/C++ Code* и далее *Generate S-Function*. S-функция как раз является описанием блока на выбранном языке программирования.

Выберем коэффициенты настроенных регуляторов, также поставим галочку в нижней строке для создания блока S-функции в Simulink. Результатом работы программы является блок S-функции разработанного регулятора. Структурная схема для сравнения цифрового регулятора с SIL блоком регулятора приведена на рис. 5.

После успешной генерации кода получаем 5 файлов с программным кодом: *Controller_sf.c*, *Controller_sf.h*, *Controller_sf_private.h*, *Controller_sf_types.h*, *Controller_sid.h*.

После завершения этапов MIL и SIL переходим к разработке виртуальной модели САУ в любой системе автоматизированного проектирования, позволяющей провести имитацию контроллера и управляемого объекта (Proteus и др.).

Заключение

Опыт разработки программного обеспечения для встраиваемых систем с помощью пакета MATLAB/Simulink совместно с Simulink Coder показал преимущества использования модельно-ориентированного подхода для генерации программного кода, используемого на целевых процессорах (контроллерах). Заложенные в методе уровни конфигурации тестирования позволяют снизить количество ошибок, связанных с человеческим фактором, допускаемых при проектировании, программировании и тестировании подобных систем, а также снизить погрешности, неизбежно возникающие при переходе от модели к аппаратной реализации. МОП обеспечивает минимизацию временных затрат на разработку и тестирование системы, способствует повторному использованию моделей для модификации системы и создания систем с расширенными возможностями.

Таким образом, за счет автоматизации процесса генерирования и тестирования

программного обеспечения для МК встроенных систем, МОП дает возможность сделать процесс разработки более эффективным, оперативно изменять параметры модели, что особенно важно для подбора оптимальных вариантов реализации систем управления.

Список литературы

1. Деменков Н.П. Модельно ориентированное проектирование систем управления [Электронный ресурс]. URL: http://is.ifmo.ru/misc2/_matlab_simulink.pdf (дата обращения: 04.09.2018).
2. Processor-in-the-Loop Simulations Applied to the Design and Evaluation of a Satellite Attitude Control [Электронный ресурс]. URL: <http://dx.doi.org/10.5772/57219> (дата обращения: 04.09.2018).
3. Сайт учебно-научного центра при МГТУ им. Н.Э. Баумана «Интеллектуальные системы» [Электронный ресурс]. URL: <http://iu1.bmstu.ru/isyst/intelsyst.htm> (дата обращения: 05.09.2018).
4. Пакет целевой поддержки процессоров Миландр для Simulink и Embedded coder [Электронный ресурс]. URL: <https://exponenta.ru/products/milandr/> (дата обращения: 05.09.2018).
5. Моделирование и симуляции динамических систем [Электронный ресурс]. URL: <https://matlab.ru/solutions/mbd/simulation-model-design> (дата обращения: 04.09.2018).
6. Модельно-ориентированное проектирование [Электронный ресурс]. URL: <https://matlab.ru/solutions/mbd/mbd> (дата обращения: 04.09.2018).
7. Понятский В.М. Генерация из модели MATLAB C-кода для микропроцессора, реализующего блок управления привода // Известия ТулГУ. Технические науки. 2011. Вып. 5. Ч. 2. С. 240–247.
8. Понятский В.М., Кушников Д.В., Федорищева В.Г. Автоматизированная технология генерации программы в среде Matlab для реализации алгоритмов управления рулевого привода // Известия ТулГУ. Технические науки. 2011. № 2. С. 357–362.
9. Понятский В.М., Фимушкин В.С., Кушников Д.В., Федорищева В.Г., Петрушин А.В., Шидловский Д.Ю. Разработка и отладка средствами Matlab управляющих программ рулевого привода // Электронный журнал «Труды МАИ». Выпуск № 62. URL: <http://trudymai.ru/upload/iblock/62a/razrabotka-i-otladka-sredstvami-matlab-upravlyayushchikh-programm-rulevogo-privoda.pdf> (дата обращения: 04.09.2018).
10. Arie Levant. Sliding order and sliding accuracy in sliding mode control. International Journal of Control. 1993. № 58(6). P. 1247–1263.