**Лектор: доц. каф. САУ, Никоза Александр Владимирович**

**e-mail: nikoza@yandex.ru**

**tel. mob. +7 911 949 18 41**

**Лекция 1. ВВЕДЕНИЕ В МОДЕЛЬНО-ОРИЕНТИРОВАННОЕ ПРОЕКТИРОВАНИЕ**

Процесс проектирования систем управления можно разделить на несколько этапов:

* создание математической модели;
* разработка алгоритмов непрерывного, дискретного, дискретно-непрерывного и логико-программного управления;
* разработка программного обеспечения для встраиваемых систем;
* подготовка соответствующих технических требований и спецификации для конструкторов, разрабатывающих электрические схемы и конструкции встраиваемого устройства.

В среде MATLAB/Simulink совместно с пакетом Simulink Coder компания MathWorks предлагает новый подход для проектирования – модельно-ориентированное проектирование (МОП). МОП — это математический и визуальный метод решения задач, связанных с проектированием систем управления и обработки сигналов. МОП определяет общую структуру взаимодействия в процессе проектирования.

МОП является методологией, применяемой при разработке встроенного программного обеспечения.

В МОП систем управления разработка происходит в 4 этапа:

* построение модели объекта управления;
* анализ и построение регулятора;
* моделирование объекта и системы управления;
* как результат — реализация системы управления на объекте.

Принципы МОП существенно отличаются от традиционной методологии проектирования. Вместо создания сложных программных кодов разработчики могут применять стандартные функциональные блоки с непрерывным и дискретным временем. Построенные таким образом модели вместе с использованием инструментов для моделирования могут быстро привести к созданию прототипа системы управления, тестированию и верификации программного обеспечения. В некоторых случаях аппаратно-программное моделирование может быть использовано в качестве инструмента проектирования для более быстрого и эффективного тестирования динамических воздействий на систему, в отличие от традиционного метода проектирования.

Некоторые из наиболее заметных преимуществ МОП в сравнении с традиционным подходом:

* МОП предоставляет **общую среду разработки**, что способствует взаимодействию группы разработчиков в процессе анализа данных и проверки системы;
* инженеры могут **найти и исправить ошибки на ранних стадиях** проектирования системы, когда время и финансовые последствия изменения системы сводятся к минимуму;
* МОП способствует **повторному использованию модели** для улучшения системы и создания производных систем с расширенными возможностями.

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 лекции является рассмотрение общих принципов модельно-ориентированного подхода при проектировании встраиваемых систем управления. Для проведения практических исследований используются методы моделирования и проектирования в среде MATLAB/Simulink 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 MATLAB/Simulink и 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 в пакете Simulink Coder 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 на ранних стадиях проектирования с 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.

**Применение МОП для микропроцессорных 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). Можно выделить 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) - «аппаратура в контуре».

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

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

На этапе SIL имитационная модель регулятора, синтезированная на предыдущем этапе, заменяется исполняемым кодом, запущенным *на той же аппаратной платформе*, где используются вычисления либо с плавающей, либо с фиксированной запятой для ускорения вычислений. В среде MATLAB для программного уровня тестирования (SIL) создается блок S-функции, что позволяет перейти к использованию в Simulink программной модели, производить подбор и настройку ее параметров. При этом происходит выполнение сгенерированного и откомпилированного С-кода 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 данного 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 сигналами, генерируемыми в имитационной модели, и программным обеспечением, установленным в целевом процессоре тестовой платы.

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 использовать возможности многопоточности контроллера.

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. Так, например, компания ABB (шведско-швейцарская транснациональная корпорация) применила продукты корпорации MathWorks для проектирования разработки и проверки программного обеспечения для электронной системы управления силовых преобразовательных элементов. 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. Российской компанией «Центр Инженерных Технологий и Моделирования «Экспонента» для микроконтроллеров-аналогов STM32 ARM CortexM3 выпущен пакет целевой поддержки контроллеров Миландр для Simulink и Simulink Coder, обеспечивающий поддержку МОП на основе платформы MATLAB/Simulink. Особо следует выделить отечественную компанию ООО "3В Сервис" ([www.3v-services.com](http://www.3v-services.com)), являющуюся разработчиком среды динамического моделирования динамических систем SimIn Tech. По своей идеологии построения и основной сущности среду SimIn Tech можно считать альтернативой платформе Simulink. В ее составе имеются все необходимые компоненты для поддержки технологии МОП. Эта среда совместима с ОС Windows и ОС "Гослинукс" (платформа Linux).

Следует также упомянуть набирающую популярность платформу Engee − это российская платформа для разработки сложных технических систем с применением методологии модельно-ориентированного проектирования.

**Модельно-ориентированное проектирование в MATLAB**

Основой для МОП является пакет прикладных программ MATLAB и его Simulink-приложение, предназначенное для визуального моделирования динамических систем.

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.

Методика генерации программного С-кода из модели Simulink включает следующие этапы (рис. 1).

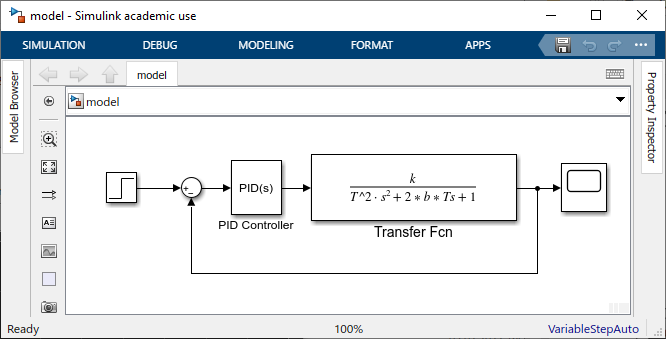
*Рис. 1.* Методика генерации С-кода для микроконтроллерных систем

*Примечание: Выполнение этапа 3 (см. рис. 1) подразумевает ориентацию на микроконтроллеры малой разрядности (8, 16), в которых отсутствуют встроенные средства выполнения вычислений с плавающей запятой (плавающей точкой). В качестве примера можно привести платформу Arduino.*

Рассмотрим возможность генерации программного кода на примере простой системы управления одномерным динамическим объектом.

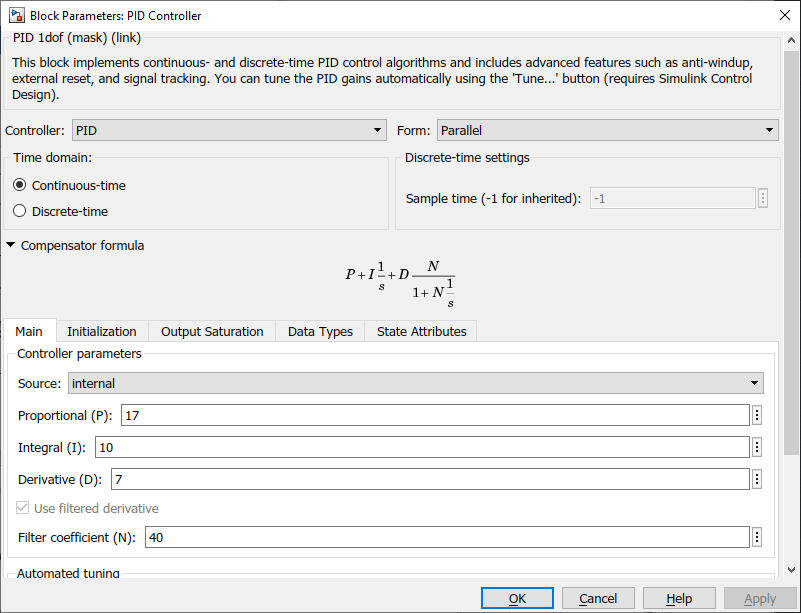
**Пример генерации программного кода (практическая работа 1)**

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 (PID Controller).



*Рис. 2.* Пример Simulink-модели системы управления, где k=1, T=1, b=0,5

(файлы data\_model.m и model.slx)



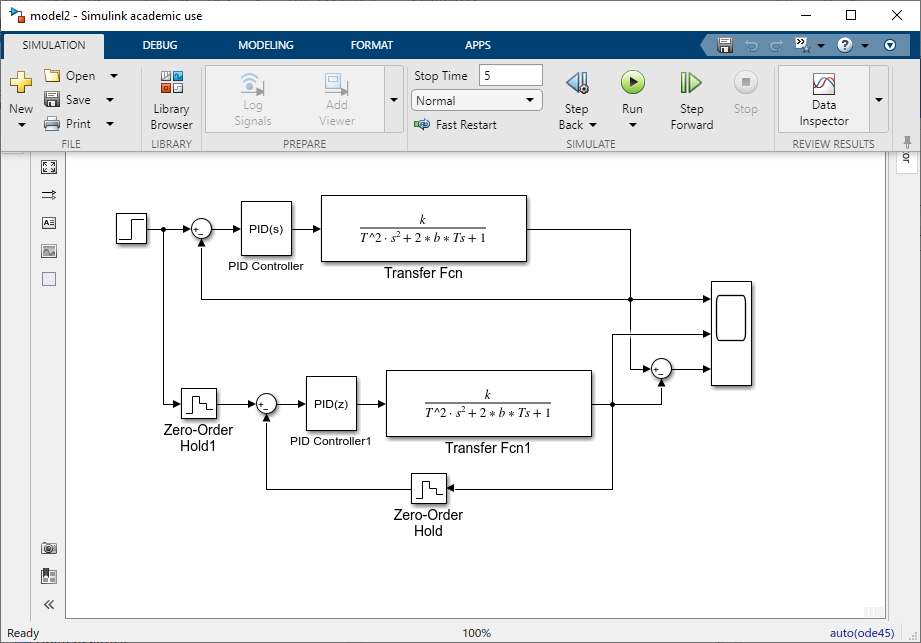
*Рис. 3.* Окно параметров блока PID Controller с настроенными коэффициентами

Математическое описание блока PID Controller (формирующего управляющий сигнал) представляется уравнением, реализующим закон PID-управления:

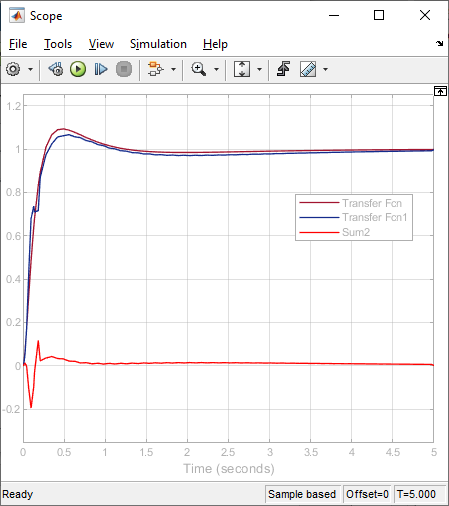
.

На начальном этапе по математическому описанию модели в MATLAB/Simulink создается структурная схема непрерывной модели системы управления (рис. 2) и производится настройка регулятора.

Следующий этап – переход к цифровой модели регулятора. Для перехода к цифровой модели регулятора в поле Time domain выбираем Discrete-time. Построение цифровой системы (точнее, цифроаналоговой системы в нашем случае) требует корректировки модели системы управления. Значение времени дискретизации (в нашем случае 0.07 с) необходимо внести в поля «Sample time» блоков модели PID(z) и Zero-Order Hold. Цифровая система управления совместно с непрерывной системой (файл model2.slx) приведена на рис. 4, а результаты моделирования показаны на рис. 5.



*Рис. 4.* Непрерывная (сверху) и цифровая (снизу) САУ

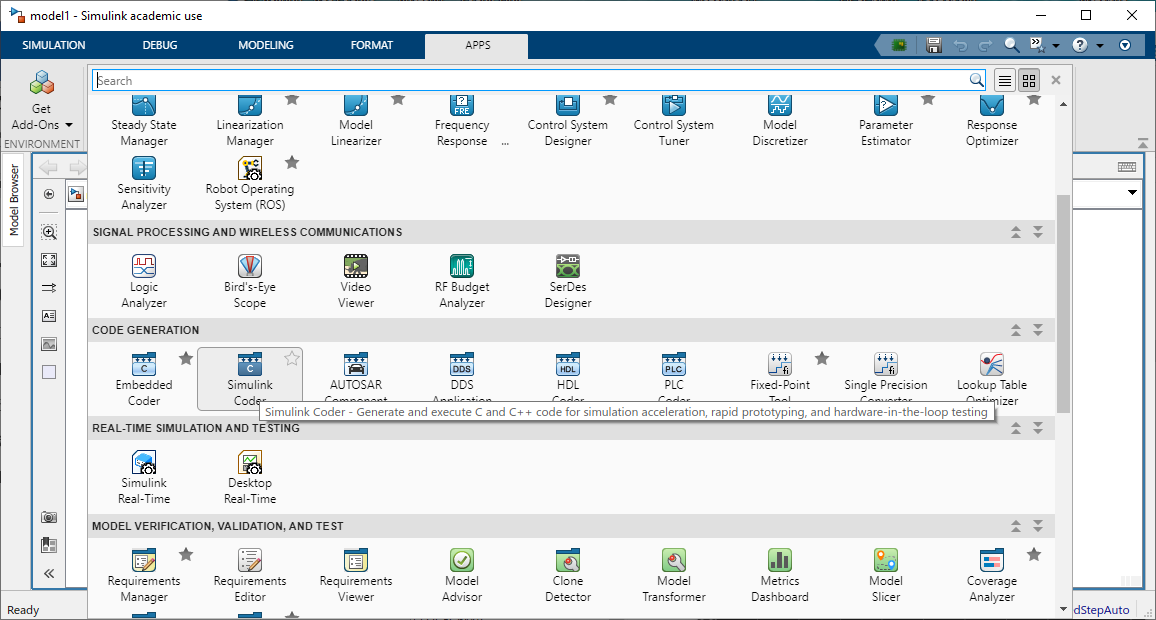


*Рис. 5.* Результаты моделирования

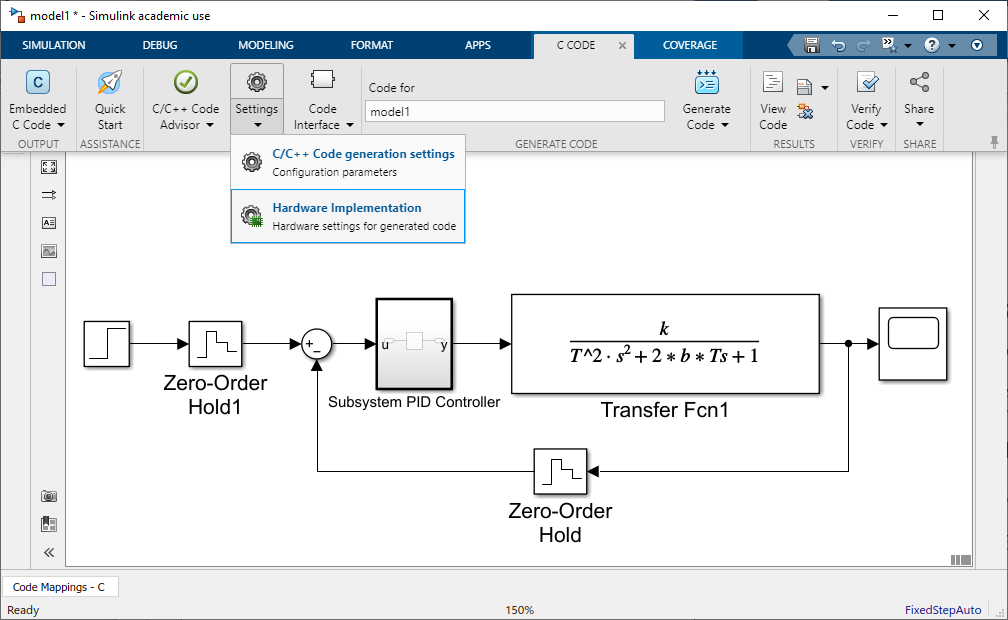
После оценки качества переходных процессов приведенной модели в цифровую форму (см. график ошибки Sum2 на рис. 5) этап «MIL» считаем завершенным и переходим к разработке кода регулятора и тестированию системы управления в режиме SIL.

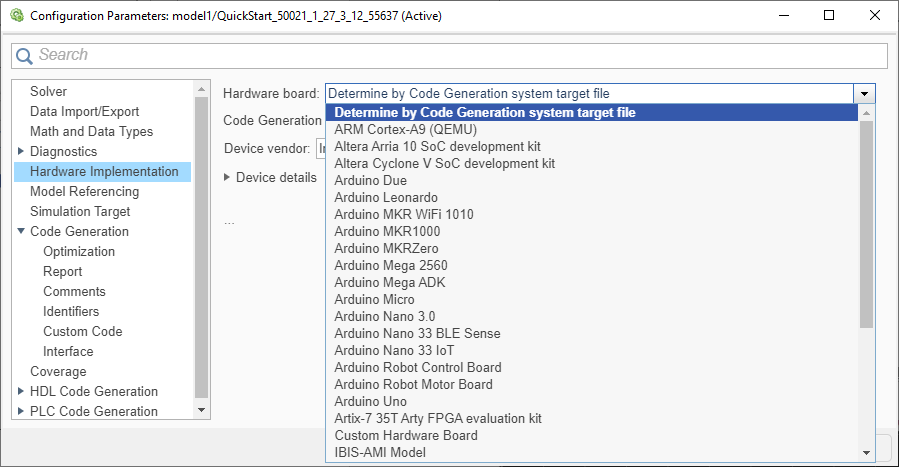
**Тестирование SIL. Генерация С-кода разработанных регуляторов**

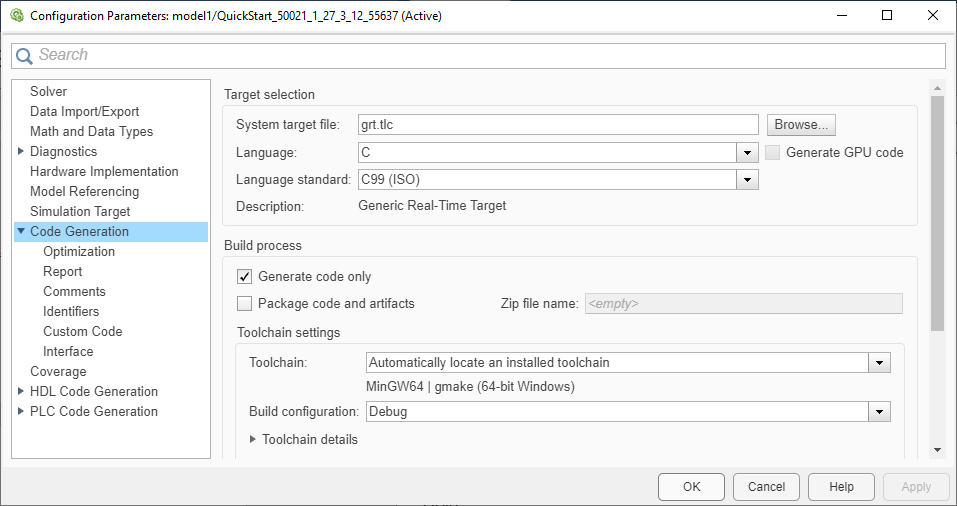
Предварительно преобразуем блок PID Controller (см. файл model1.slx) в подсистему (Subsystem). Тип подсистемы должен быть **монолитным** (Atomic Subsystem). Запустим приложение APPS/Simulink Coder (рис. 6). Далее проведем настройки целевой аппаратной платформы. Воспользуемся возможностью настройки параметров модели Simulink. Откроем вкладку C CODE Settings. Во вкладке Hardware Implementation (см. рис. 7) необходимо выбрать целевую платформу и ознакомиться с размером и типами данных на выбранной платформе.



*Рис. 6.* Запуск Simulink Coder





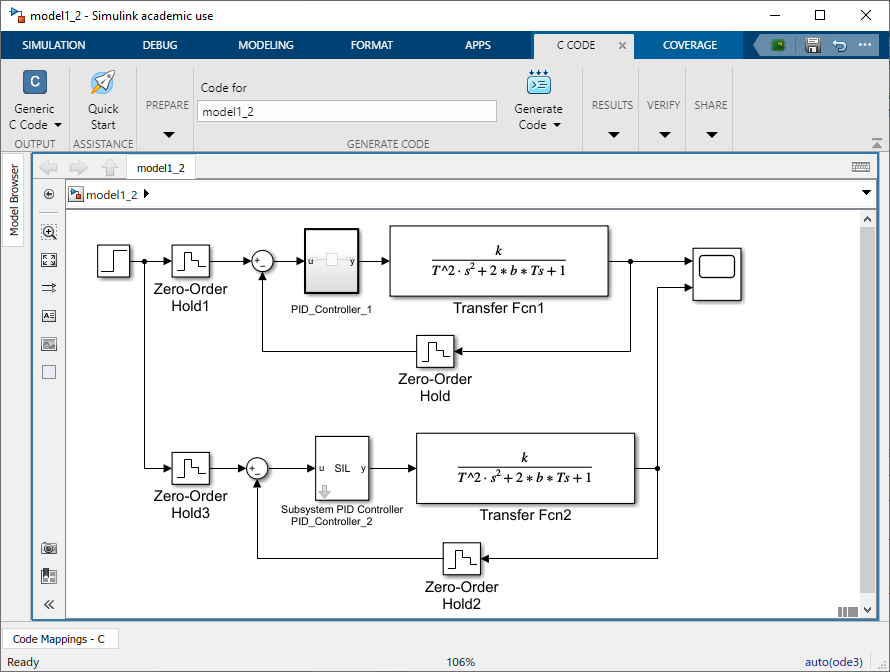


*Рис. 7.* Подготовка к генерации C-кода

Затем во вкладке C/C++ Code Generation settings (см. рис. 7) выберем утилиту для получения программного кода и язык программирования, на котором этот код будет написан (C или C++).

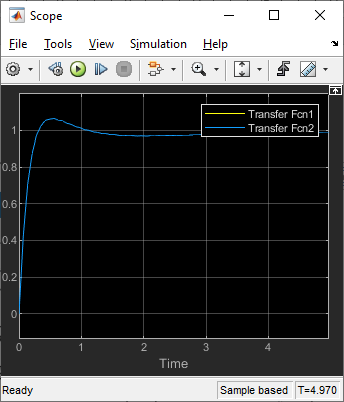
Для проведения моделирования и тестирования в режиме SIL выберем блок Subsystem PID Controller (файл model1.slx), выберем из контекстного меню блока команду C/C++ Code и далее Generate S-Function. S-функция как раз является описанием блока на выбранном языке программирования. Поставим галочку в нижней строке для создания блока S-функции в Simulink (Create Software-in-the-Loop (SIL) block) и нажмем клавишу Build. Результатом работы программы является создание блока S-функции разработанного регулятора. Этот блок будет обозначен как SIL-блок (файл SIL\_block.slx). Его следует скопировать в Simulink-модель, предназначенную для SIL-моделирования.

Структурная схема для сравнения цифрового регулятора с созданным в результате компиляции блоком регулятора SIL PID Controller2 приведена на рис. 8. На рис. 9 приведены результаты моделирования в режиме SIL.



*Рис. 8.* Структурная схема цифровой САУ (сверху) и САУ с SIL (снизу)

(файл model1\_2.slx)



*Рис. 9.* Результат моделирования в режиме SIL

В качестве примера на рис. 10 приведено окно настройки параметров блока PID\_Controller\_1, а далее приведен С-код, полученный для SIL-блока PID\_Controller\_2 (файл Subsystem.c).

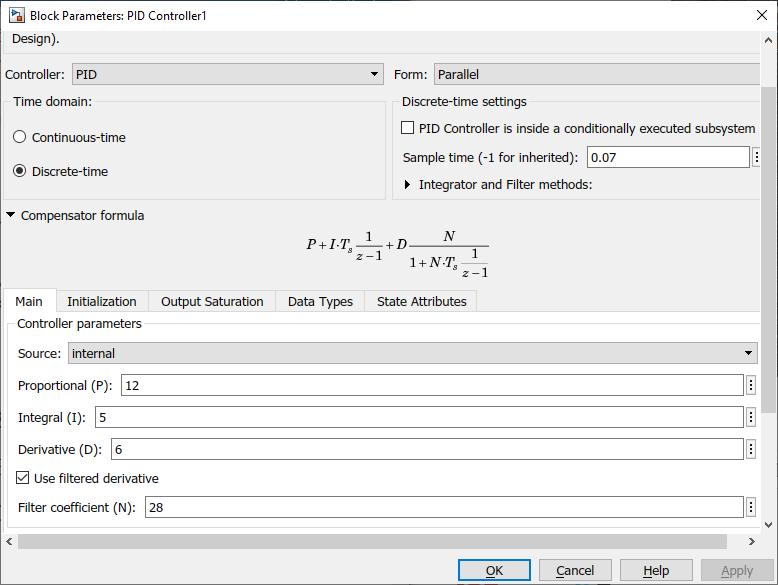


Рис. 10. Окно настройки параметров блока PID\_Controller\_1

Исходный С-код для SIL-блока PID\_Controller\_2 (файл Subsystem\_ert\_rtw\Subsystem.c без комментариев)

void Subsystem\_step(void)

{

real\_T rtb\_FilterCoefficient;

rtb\_FilterCoefficient = (6.0 \* Subsystem\_U.u - Subsystem\_DW.Filter\_DSTATE) \*28.0;

Subsystem\_Y.y = (12.0 \* Subsystem\_U.u + Subsystem\_DW.Integrator\_DSTATE) +

rtb\_FilterCoefficient;

Subsystem\_DW.Filter\_DSTATE += 0.07 \* rtb\_FilterCoefficient;

Subsystem\_DW.Integrator\_DSTATE += 5.0 \* Subsystem\_U.u \* 0.07;

}

После завершения этапов MIL и SIL моделирования можно переходить к разработке виртуальной модели САУ в любой системе автоматизированного проектирования, позволяющей провести имитацию контроллера и управляемого объекта (например, в среде Proteus), либо использовать среды разработки микропроцессоров и соответствующие прикладные программы и библиотеки (Arduino, Waijung и др.).

**Краткие выводы**

Опыт разработки программного обеспечения для встраиваемых систем с помощью пакета MATLAB/Simulink совместно с Simulink Coder показывает 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 созданию 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.

В данной лекции были кратко рассмотрены основные принципы МОП и методология проектирования. Более подробно мы будем рассматривать эти вопросы в процессе последующих лекций, на практических и лабораторных работах, а также при выполнении индивидуальных заданий.

План изучения дисциплины:

|  |  |
| --- | --- |
| **Темы занятий** | **Вид контроля** |
| Введение в модельно-­ориентированное проектирование. Способы создания моделей в MATLAB/Simulink. | Тест 1 |
| Проектирование систем на основе заданного расположения полюсов. Модальное управление с использованием наблюдателей состояния. | Тест 2 |
| Модели систем цифрового управления непрерывными объектами. Управление в непрерывной и дискретной системе. | Тест 3 |
| Непрерывный и дискретный ПИД-­регулятор. Работа с Simulink Coder. Разработка программ для программируемых логических контроллеров и микропроцессоров. |  |
| Работа Simulink-­моделей в режиме реального времени. Разработка моделей в среде Simulink/Simscape | Тест 4 |

***Задание к практической работе 1***

1. ***Установите на личный компьютер систему MATLAB/Simulink. Установка MATLAB описана на сайте университета.***

***Желательно установить/переустановить версию MATLAB R2021b, так как именно эта версия используется в лаборатории 7310.***

***Для выполнения компиляции на языки C/C++ дополнительно требуется установка компилятора MinGW-w64 C / C ++ для Windows. При установке MATLAB (в конце установки) также предлагается установить компилятор. Воспользуйтесь этой возможностью.***

***Все установочные файлы MATLAB R2021b имеются в лаборатории 7310. Для их скачивания нужна флешка 32 гб.***

***Все файлы моделей, сопровождающие лекции и лабораторные работы, созданы в среде MATLAB R2021b. Их работа в более ранних версиях невозможна!***

***При работе с Simulink-моделями в более поздних версиях (R2022, R2023) сохраните файлы моделей в формате версии R2021b для их корректной работы в лаборатории.***

1. ***В системах MATLAB различных годов выпуска имеются различия в интерфейсах. Могут отсутствовать некоторые вкладки, иное расположение пунктов меню и т. п.***
2. ***При затруднениях используйте справочную систему Help и сайт*** [***https://exponenta.ru/matlab***](https://exponenta.ru/matlab) ***(требуется регистрация).***
3. ***Зарегистрируйтесь также на сайте разработчика системы MATLAB*** [***www.mathworks.com***](http://www.mathworks.com)***.***
4. ***Выполните на своем компьютере описанный в данной лекции Пример генерации программного кода, изменив значения исходных данных k, T, b в небольших пределах по своему выбору. Также самостоятельно подберите период дискретизации.***
5. ***Составьте отчет по п. 5.***