СТЕНДЫ ПОЛУНАТУРНОГО МОДЕЛИРОВАНИЯ НА БАЗЕ ТЕХНОЛОГИЙ МАТЛАБ ДЛЯ РАЗРАБОТКИ СИСТЕМ УПРАВЛЕНИЯ НА ОСНОВЕ ТЕХНОЛОГИЙ МОДЕЛЬНО-ОРИЕНТИРОВАННОГО ПРОЕКТИРОВАНИЯ

А.А. ЕГОРОВ (ООО “АВИАТЭКС”)

Статья посвящена разработке систем управления на основе модельно-ориентированного проектирования с использованием математического и полунатурного моделирования. Рассмотрены вопросы модельно-ориентированного проектирования систем управления реального времени и их тестирования.

Тестирование систем управления в реальном времени представлено на базе технологий МАТЛАБ с использованием моделей Simulink, ОСРВ Simulink Real-Time и КПМ “РИТМ”. Обсуждаются возможности продуктов Simulink Real-Time и комплекса полунатурного моделирования КПМ “РИТМ”. Приводится пример успешного применения КПМ “РИТМ” с ОСРВ Simulink Real-Time на Уральском заводе гражданской авиации. В заключение рассмотрены мотивации использования математического и полунатурного моделирования при разработке сложных систем автоматического и интеллектуального управления.

Ключевые слова: *системы управления; модельно-ориентированное проектирование; стенды полунатурного моделирования; тестирование систем управления; МАТЛАБ; ОСРВ; Simulink Real-Time; КПМ “РИТМ”.*

Разработка в сжатые сроки современных систем управления сложными дорогостоящими объектами проводится на основе методологий модельно-ориентированного проектирования.

Модельно-ориентированное проектирование (МОП) представляет собой особую методологию проектирования встраиваемых систем реального времени. Проектирование систем управления выполняется с помощью методологий, основанных на систематическом использовании математических моделей. При этом математические модели развиваются, уточняются и эффективно используются на всех этапах жизненного цикла систем управления. Основной целью применения МОП является гарантия того, что при разработке системы управления в динамике учтено взаимное влияние компонентов системы управления, параметры ее настройки оптимальны для выполнения поставленных задач, испытания будут проведены во всех возможных режимах эксплуатации систем управления с объектом управления и компания разработчик уложилась в бюджет при достижении нужного качества систем управления в приемлемые сроки. Эта методология хорошо зарекомендовала себя при разработке различных сложных систем управления, в частности встраиваемых систем управления, цифровой обработки сигналов, компьютерного зрения, радиолокации и связи [1].

Создание любой системы управления начинается с описания компонентов и определения связей между ними. Однако разработка ее архитектуры представляет собой непростой и итерационный процесс, который основывается на требованиях или тактико-техническом задании на систему управления. Понятие “Техническое задание” подразумевает вполне конкретный документ со структурой, определенной действующими стандартами. Причем, как показывает опыт специалистов [2], использование этого документа в большинстве случаев не позволяет добиться одной из основных целей, ради которой разрабатываются требования – сформировать единое представление системы управления как разработчиками, так и заказчиком.

Рассмотрим кратко место и роль документа требований в общем процессе разработки систем управления. Прежде чем разбирать подробно непосредственно сам документ требований к системе управления коротко рассмотрим место и роль этого документа во всем процессе разработки программного-аппаратного обеспечения (ПАО) системы управления (СУ).

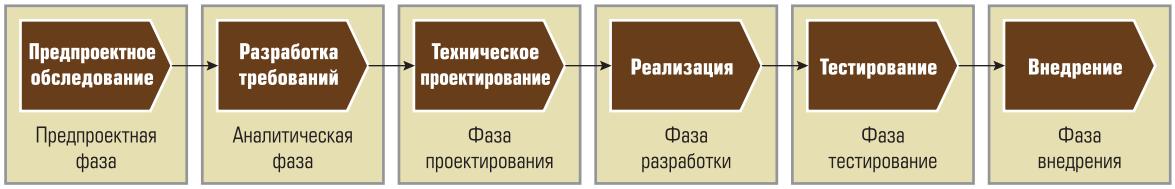

Рис. 1. Упрощенное представление жизненного цикла разработки ПАО СУ

Жизненный цикл разработки ПАО СУ в упрощенном виде выглядит следующим образом (рис. 1).

Жизненный цикл (ЖЦ) включает в себя фазы создания ПАО, каждая из которых может выполняться итерационно. Кроме того, сам жизненный цикл не является строго линейным, поскольку существует довольно большое количество обратных связей, влияющих на весь ход проекта разработки СУ.

Кроме основных процессов ЖЦ, представленных на рис. 1, существуют так называемые процессы управления, поддерживающие и сопровождающие весь цикл разработки ПАО: управление требованиями; управление проектом в целом; конфигурационное управление; управление рисками и т. д. Однако эти процессы находятся за пределами рассматриваемой в данной статье темы, поэтому их сознательно игнорируем.

Из рис. 1 явно видно, что разработка требований является отправной точкой всего процесса создания ПАО СУ. Действительно, имея на входе концептуальное представление о продукте, на выходе аналитической фазы проектная команда получает документ требований, описывающий продукт настолько детально, как это необходимо для дальнейшего проектирования ПАО СУ.

Исходя из этого можно утверждать, что качественно разработанный документ требований позволит избежать ненужных переделок уже готовой системы, снизить стоимость и скорость непосредственно разработки и тестирования, не допустить расползания границ проекта и, наконец, добиться большей удовлетворенности заказчиков систем управления.

Верхнеуровневая структура документа требований (рис. 2) определяется структурой требований, выявление которых является необходимым условием успешного проектирования автоматизированной системы.

Основой для начала определения требований являются цели проекта. Однозначная трактовка целей позволяет сформировать однозначное видение Заказчиком и Разработчиком того эффекта, который будет получен бизнесом Заказчика в результате создания и внедрения проектируемого ПАО СУ. Цели проекта служат своего рода точкой зрения заинтересованных сторон на достигнутые в ходе реализации проекта результаты.

Следует особо отметить, что цели не должны восприниматься как банальная отписка, служащая введением в документ. Цели создания продукта являются основой для определения бизнес-требований к системе (Business requirements). Это самый первый, самый верхний уровень документа требований. Бизнес-требования описывают, каким образом разрабатываемая система управления связана с достижением бизнес-целей организации и что она должна для этого делать. Можно сказать, что бизнес-требования являются своего рода задачами, которые должна решать ПАО СУ для достижения целей своего создания.

Как видно из схемы, представленной на рис. 2, бизнес-требования определяют функциональные и нефункциональные требования.

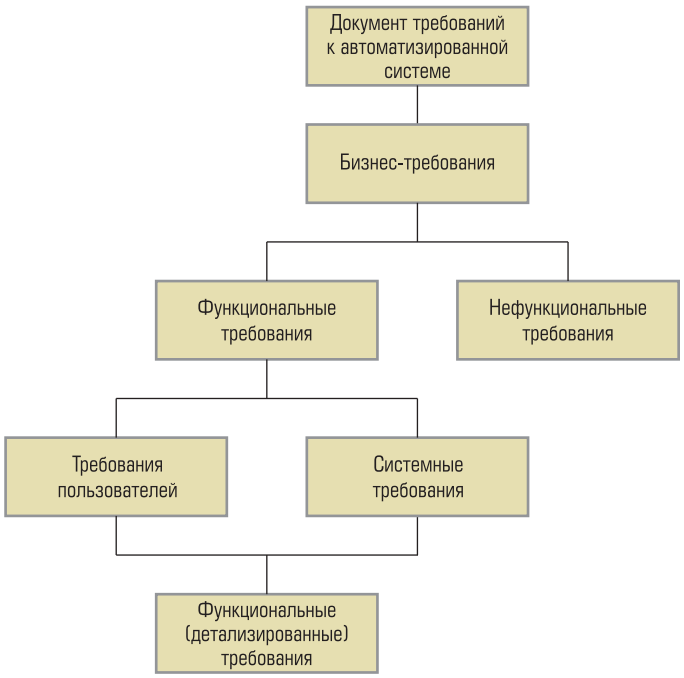

Рис. 2. Верхнеуровневая структура документа требований к АС

Важно отметить, что на практике может получиться так, что функциональные и нефункциональные требования могут “заходить” за границы бизнес-требований, однако они никогда не должны им противоречить.

К функциональным требованиям относится все то, что описывает функциональность системы управления, т.е. что, как и когда делает система управления, а также, кто принимает в этом участие.

Нефункциональные требования представляют собой характеристики системы управления, которые не имеют прямого отношения к автоматизируемым процессам, но, тем не менее, без которых работать с системой управления было бы, мягко говоря, проблематично. В качестве примера, сюда можно отнести требования по удобству в использовании, производительности, масштабируемости, требования к документированию, к организационному и методическому обеспечению, к интеграции с внешними системами и т.п.

Функциональные требования включают в себя, так называемые, требования пользователей и системные требования. Требования пользователей (Customer requirements) представляют собой те требования верхнего уровня, которые предъявляют к системе управления будущие бизнес-пользователи системы. Другими словами, те задачи, которые система должна решать с точки зрения бизнеса Заказчика. Еще раз необходимо подчеркнуть, что требования пользователей описывают верхнеуровневую функциональность системы управления, не вдаваясь в подробности ее технической реализации.

Системные требования (System requirements) имеют ровно такой же смысл, что и пользовательские требования, с той разницей, что в большей степени продиктованы не потребностями бизнеса, а особенностями IT-инфраструктуры Заказчика систем управления. Другими словами, это требования, которые определяют регламент взаимодействия создаваемого ПАО с внешними системами, регламентируют программно-аппаратные платформы функционирования продукта и т.п. Например, “Система управления должна иметь возможность осуществлять автоматический обмен данными с системой мониторинга”.

В то время как требования пользователей и системные требования, как правило, отвечают на вопрос: что должна делать система управления, то детализированные функциональные требования (Functional requirements) отвечают на вопросы: кто, как и когда. Требования этого типа определяются на основе анализа и детального описания процессов, озвученных в требованиях верхнего уровня – пользовательских и системных. Со всеми соответствующими атрибутами – функциями, условиями, событиями, исполнителями, входами и выходами. Т.е., фактически, представляют собой детальное описание реализуемых системой потоков задач и данных.

Теперь попробуем спроецировать участников процесса разработки требований на приведенную выше верхнеуровневую структуру и, тем самым, получив матрицу соответствия, понять зоны ответственности каждого из участников (таблица 1) [2].

Эта матрица необходима при подробном рассмотрении соответствующих разделов документа требований к системе управления.

Как показано выше, пользователям и разработчикам ПАО СУ приходится сталкиваться с необходимостью проработки уточнённых требований, которые естественным образом появляются в процессе продумывания поведения и сценариев работы системы управления. Модельно-ориентированное проектирование предоставляет гибкие инструменты для создания и проектирования архитектуры системы в виде модели, описания компонентов, их свойств, интерфейсов между

Таблица 1

№	Тип требований	Ответственные	Возможные участники
1	Бизнес-требования	Менеджмент заказчика	Эксперты предметной области. Аналитик исполнителя. Прочие специалисты исполнителя
2	Требования пользователей	Бизнес-пользователи системы со стороны заказчика. IT-подразделение заказчика. Аналитик исполнителя	Менеджмент заказчика. Эксперты предметной области. Прочие специалисты исполнителя
3	Системные требования	IT-подразделение заказчика. Аналитик исполнителя	Бизнес-пользователи системы со стороны заказчика. Прочие специалисты исполнителя
4	Функциональные требования	Бизнес-пользователи системы со стороны заказчика. Аналитик исполнителя	Менеджмент заказчика. IT-подразделение заказчика. Прочие специалисты исполнителя
5	Нефункциональные требования	IT-подразделение заказчика. Аналитик исполнителя. Разработчик исполнителя	Бизнес-пользователи системы со стороны заказчика

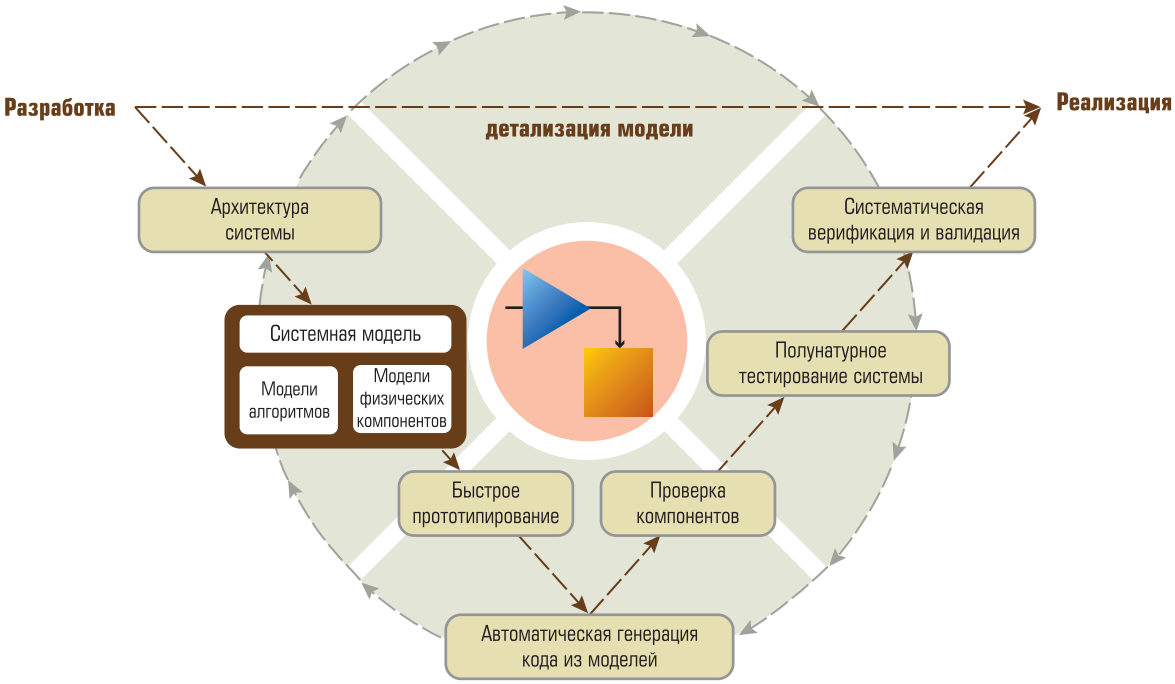

Рис. 3. Модельно-ориентированное проектирование предоставляет гибкие инструменты для создания и проектирования архитектуры системы управления [1]

компонентами, а также анализа свойств системы (рис. 3).

Рассмотрим далее методологию МОП на примере технологий, предоставляемых пакетом прикладных программ МАТЛАБ [3]. Важным продуктом этих технологий является Simulink, который представляет собой среду для моделирования и разработки динамических и встраиваемых систем управления. Эта среда позволяет при помощи блок-схем моделировать линейные и нелинейные дискретные и непрерывные, гибридные и многочастотные системы. Сегодня Simulink предоставляет решения для систем управления, обработки сигналов, систем связи и других областей с применением системной инженерии. Кроме того, Simulink является основой для модельно-ориентированного проектирования в том числе и сложных систем управления. Что же такое модельно-ориентированное проектирование, которое уже несколько раз было упомянуто выше – это современная технология разработки больших сложных систем управления. В центре процесса разработки находятся математические модели объекта и системы управления. Эта модель выполняет разные функции на разных фазах развития проекта (рис. 4).

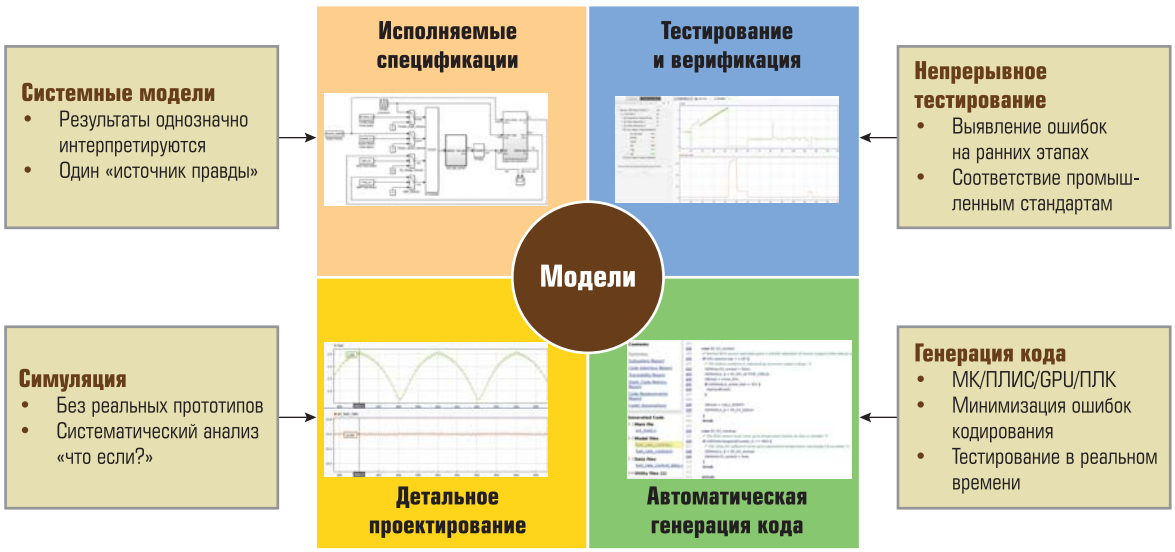

Рис. 4. Модельно-ориентированное проектирование систем [1]

- **Исследование функционирования компонента или полной встраиваемой системы в реальных рабочих условиях:**
  - детерминизм выполнения задач;
  - учет частоты дискретизации;
  - синхронизация задач;
  - учет работы интерфейсов.
- **Распространенные виды тестирования:**
  - **быстрое прототипирование** алгоритмов управления (Rapid Prototyping);
  - **полунатурное моделирование** объекта управления (Hardware-in-the-loop, HIL);
  - сбор данных и обработка сигналов.

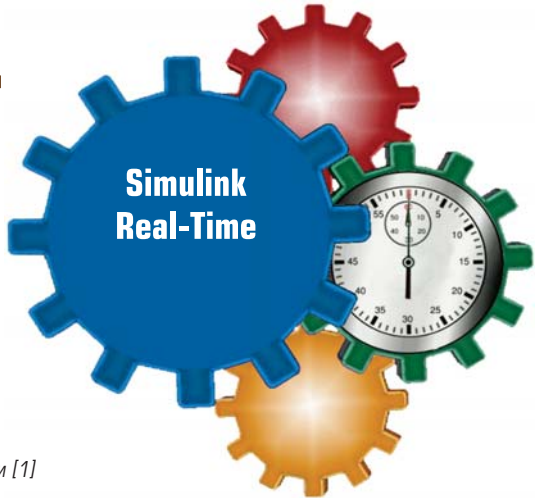

Рис. 5. Задачи тестирования разрабатываемой СУ в реальном времени [1]

Начнем описание МОП на базе технологий МАТЛАБ с исполняемых спецификаций. Они называются так потому, что в этой модели спецификации отражают высокоуровневые требования к проектируемой системе, а исполняемые, потому что можно нажать кнопку и запустить моделирование (симуляцию). Такие исполняемые спецификации являются однозначными и не предполагают разночтений. Формат исполняемых спецификаций и возможность запуска их на моделирование позволяет разработчику проводить валидацию системных требований, например, осуществлять поиск пропущенных требований.

Разработчик может запускать математическое моделирование (симуляцию) на самых ранних стадиях разработки проекта, что позволяет ему найти ошибки гораздо раньше, чем при традиционном процессе разработки. Исполняемая спецификация, которая является часто системной моделью, постоянно детализируется и уточняется разработчиками. Это позволяет уточнять, детализировать новые требования, проводить моделирование (симуляцию) без реальных прототипов оборудования, которое может быть ещё недоступно или слишком дорого. Моделирование также позволяет проводить систематический анализ, например, что будет, если изменить тот или иной параметр алгоритма. Что будет, если изменить сам алгоритм тем или иным образом?

Автоматическая генерация кода – это важная часть модельно-ориентированного проектирования систем. При традиционном подходе, по сути, разработчик получает две составляющие разрабатываемого проекта – это программный код и математическая модель. Если разработчику необходимо внести изменения, возникает вопрос – нужно ли это сделать на уровне программного кода или на уровне математической модели и как синхронизировать эти изменения? Если разработчик обнаружил ошибку, то возникает вопрос: ошибка возникла при моделировании или при ручном кодировании. Фактически автоматическая генерация кода позволяет избегать этих и многих других проблем, минимизируя ошибки кодирования, устанавливая единообразность исходного программного кода и помогает разрабатывать действительно большие и сложные системы управления.

Задачи тестирования представлены на рис. 5.

Это процесс, который охватывает все этапы жизненного цикла проекта разработки и внедрения СУ. Модельно-ориентированное проектирование предоставляет возможности непрерывного тестирования и верификации на всех стадиях жизненного цикла проекта и, что очень важно, позволяет выявлять ошибки на самых ранних стадиях разработки проекта. Причем чем раньше была обнаружена ошибка, тем проще и дешевле ее исправить. Модельно-ориентирное проектирование позволяет объединить разные группы инженеров, предоставляя им единую среду разработки и верификации. Системные инженеры, разработчики, программисты и тестировщики могут разговаривать на одном языке, могут свободно обмениваться информацией друг с другом.

Тестированию на всех этапах разработки систем управления должно уделяться достаточное внимание. Тестирование модуля в составе системы может оказаться неполным из-за сложности расчета спектра воздействий на систему, достаточного для тестирования модуля, а также ограниченных средств и времени тестирования.

Тестирование аппаратных и программных модулей в имитационной среде путем подачи тестовых последовательностей и анализа реги-

- Каждый шаг расчета начинается в гарантированный момент времени.
- Каждый шаг расчета гарантированно заканчивается до начала следующего шага.
- Пример временнóй диаграммы: каждая задача гарантированно завершается за время, отведенное ей.

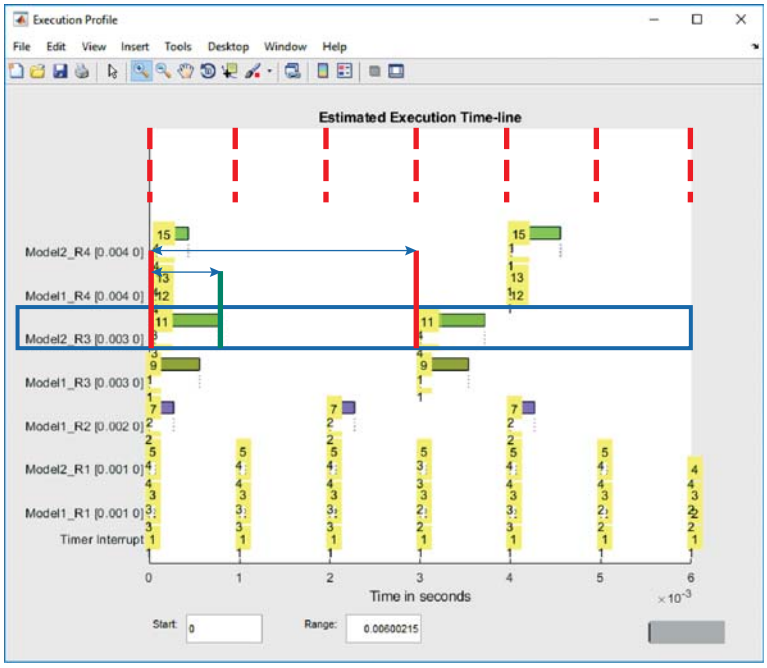

Рис. 6. Реализация жесткого реального времени в многозадачном режиме [3]

стрируемых откликов позволяет существенно упростить процедуру тестирования.

Тестирование в реальном времени представляет собой исследование функционирования системы или ее отдельных компонентов в реальных рабочих условиях при заданной частоте дискретизации с эффектами, возникающими при работе с реальным оборудованием и интерфейсами, с учетом синхронизации различных выполняющихся задач. Тестирование в реальном времени применяется в технологиях быстрого прототипирования алгоритма управления, полунатурного моделирования объекта управления, также известная как Hardware-in-the-loop (HIL) [3].

Основой работы систем управления реального времени является понятие детерминизма, то есть вызова расчета алгоритма управления строго в определенные моменты времени. Одним из ключевых компонентов является целевой вычислитель с операционной системой реального времени, который позволяет выполнять требуемые задачи и расчеты в темпе тестируемого реального объекта (рис. 6).

Системы жёсткого реального времени не допускают задержек реакции системы, так как это может привести к потере актуальности результатов, большим финансовым потерям или даже авариям и катастрофам. Ситуация, в которой обработка событий происходит за время, большее предусмотренного, в системе жёсткого реального времени считается фатальной ошибкой. При возникновении такой ситуации операционная система прерывает операцию и блокирует её, чтобы, насколько возможно, не пострадала надёжность и готовность остальной части системы. Примерами систем жёсткого реального времени могут быть бортовые системы управления (на самолёте, космическом аппарате, корабле и пр.), системы аварийной защиты, регистраторы аварийных событий.

Жесткое реальное время означает, что каждый шаг расчета начинается в строго определенный гарантированный момент времени. При этом каждый шаг расчета гарантированно завершается до начала следующего шага (рис. 6).

Здесь, на рис. 6 (справа) приведен пример временной диаграммы реального алгоритма управления с помощью приложения реального времени, полученного при помощи процедуры профилирования. Профилирование является способом измерить время исполнения кода и позволяет идентифицировать, где MATLAB® проводит большую часть времени при вычислениях. После того, как разработчик идентифицирует, какие функции используют большую часть времени, он может оценить их для возможности повышения производительности. Также можно про-

- **Алгоритмы придуманы, но их не на чем проверять:**
  – Целевая платформа (блок управления, контроллер) еще недоступна.
- **Алгоритмы придуманы, но еще не реализованы программно:**
  – Программист еще только планирует когда-нибудь реализовать алгоритм.

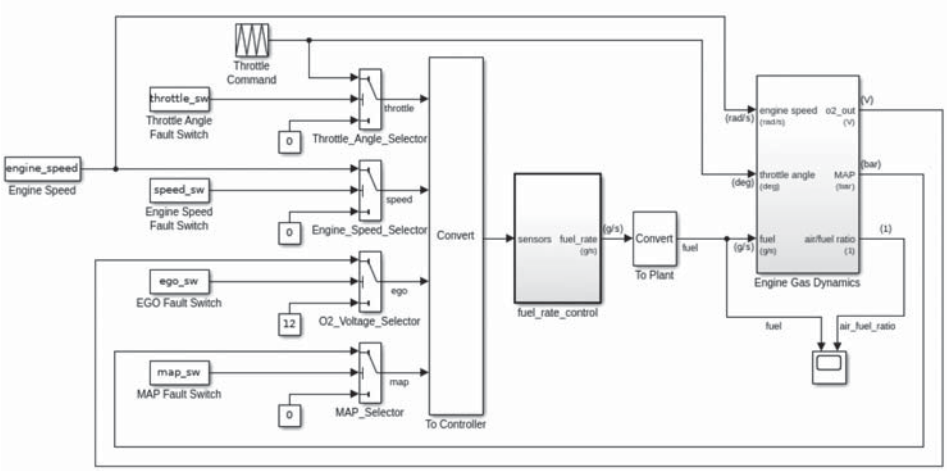

Рис. 7.
Проблема 1:
Целевой вычислитель недоступен для проверки разрабатываемых алгоритмов управления [3]

филировать код, чтобы определить, какие строки кода не запускаются. Определение, которое не запускает строки кода, полезно при разработке тестов для кода, или как средство отладки, чтобы помочь изолировать проблему в коде.

Прерывание системного таймера (рис. 6) приходит в этом примере каждую миллисекунду. В рассматриваемом примере всего 7 различных задач, которые работают с разным шагом расчета, т.е. с разной частотой дискретизации. Например, задача modl2_R3 считается 3 мсек. Видно, что планировщик операционной системы реального времени вызывает эту задачу точно каждые 3 мсек. Также видно, что время выполнения этой задачи чуть меньше одной мсек и укладывается в шаг расчета 3 мсек. Планировщик операционной системы реального времени обычно предоставляет пользователю возможность настроить, что будет происходить, если время выполнения задачи не уложилось в шаг расчета.

Операционные системы общего назначения, такие как Windows или Linux, не являются операционными системами реального времени и не могут гарантировать выполнение алгоритма в жестком реальном времени.

Далее остановимся на разных подходах к тестированию в реальном времени, которые могут применяться при разработке системы. Давайте остановимся на следующих проблемах, возникающих при разработке программного обеспечения (рис. 7).

Что будет делать разработчик, если у него уже есть разработанные алгоритмы, но нет оборудования, на котором их можно проверить, например, отсутствует целевая платформа, то есть блок управления или контроллер ещё недоступны? Что делать, если у разработчика есть алгоритмы, но ему приходится ждать, когда программист реализует этот алгоритм. Решение этих проблем лежит в области современной технологии быстрого прототипирования алгоритмов управления (рис. 8).

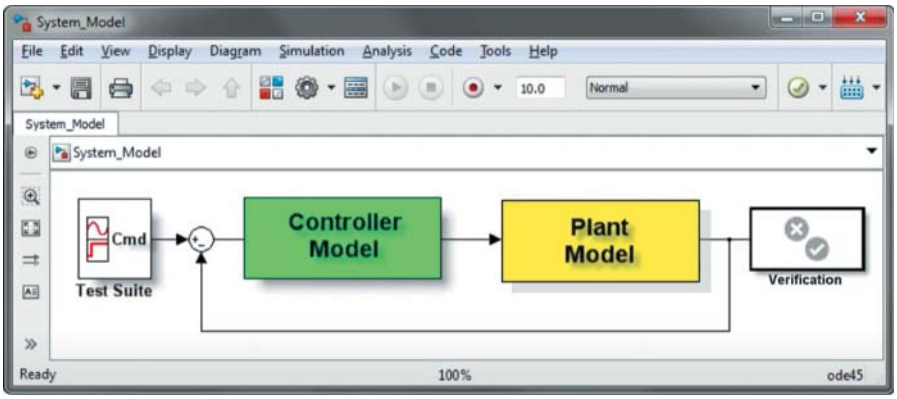

Рис. 8.
Решение:
прототипирование алгоритмов управления [3]

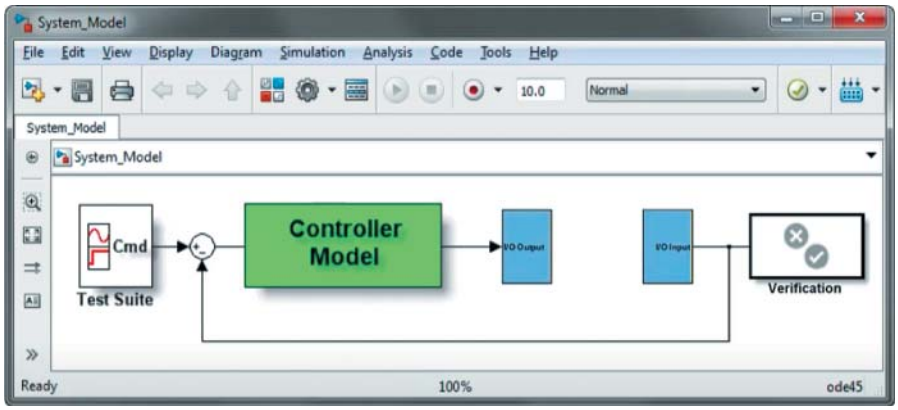

Рис. 9. ***Решение:*** *прототипирование алгоритмов управления [3]*

Быстрое прототипирование – это процедура, позволяющая вносить изменения в алгоритм управления разработчиком и исследовать работу алгоритма с использованием реального оборудования. Быстрое прототипирование означает, что разработчик берет разработанный алгоритм управления, загружает его в систему реального времени и подключает к реальному объекту управления. Модель объекта управления заменяется реальным объектом управления (рис. 9).

В модели управления используются периферийные блоки (I/O Output и I/O Input) или драйверы для подключения алгоритма к реальному объекту управления. Из модели, содержащей алгоритм управления, автоматически генерируется код, который переносится затем на компьютер реального времени. Входные данные при этом могут генерироваться самой моделью. Компьютер реального времени с работающим алгоритмом управления подключен жгутами проводов к реальному объекту управления. Такой метод называется быстрым прототипированием потому, что от фазы внесения изменений в алгоритм, разработчик сразу переходит к фазе проверки алгоритма на реальном оборудовании (например, объекте управления) и обратно, совершая таким образом быстрые итерации.

Этот метод тестирования также полезен, когда целевого процессора у разработчика еще нет. Или разработчик ещё не решил, какой процессор выбрать, но он может разрабатывать и проверять алгоритм управления, не дожидаясь, когда целевая платформа станет доступна для него. У разработчика есть компьютер реального времени со всеми необходимыми интерфейсами (в том числе каналами ввода/вывода аналоговых и цифровых каналов), и разработанные алгоритмы из модели в среде Simulink он их загружает туда простым нажатием кнопки.

Рассмотрим использование машин реального времени, которые обычно являются достаточно мощными и не накладывают ограничения, например, на использование памяти или процессорного времени. Стенд для быстрого прототипирования является универсальным и легко переконфигурируется под другой или следующий проект. Быстрое прототипирование алгоритмов управления существенно сокращает цикл разработки систем в целом.

Проанализируем другие проблемы, которые могут возникать (рис. 10).

Пусть у разработчика системы управления есть уже готовый блок управления. Он может оказаться в ситуации, когда использование

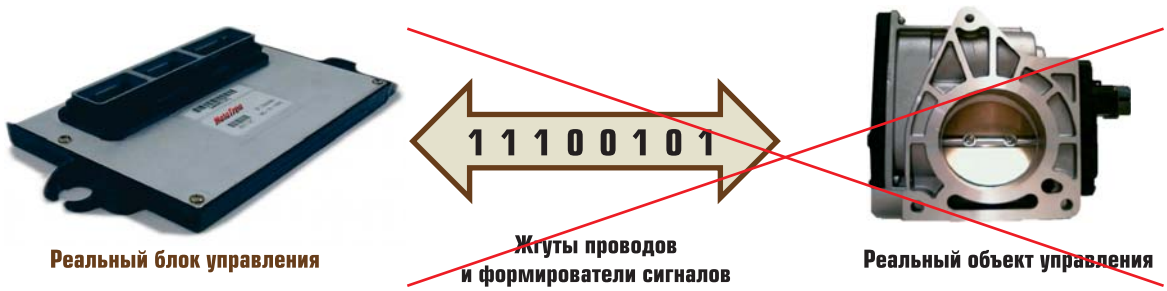

Рис. 10. Отладка алгоритмов управления в отсутствии реального объекта управления [3]

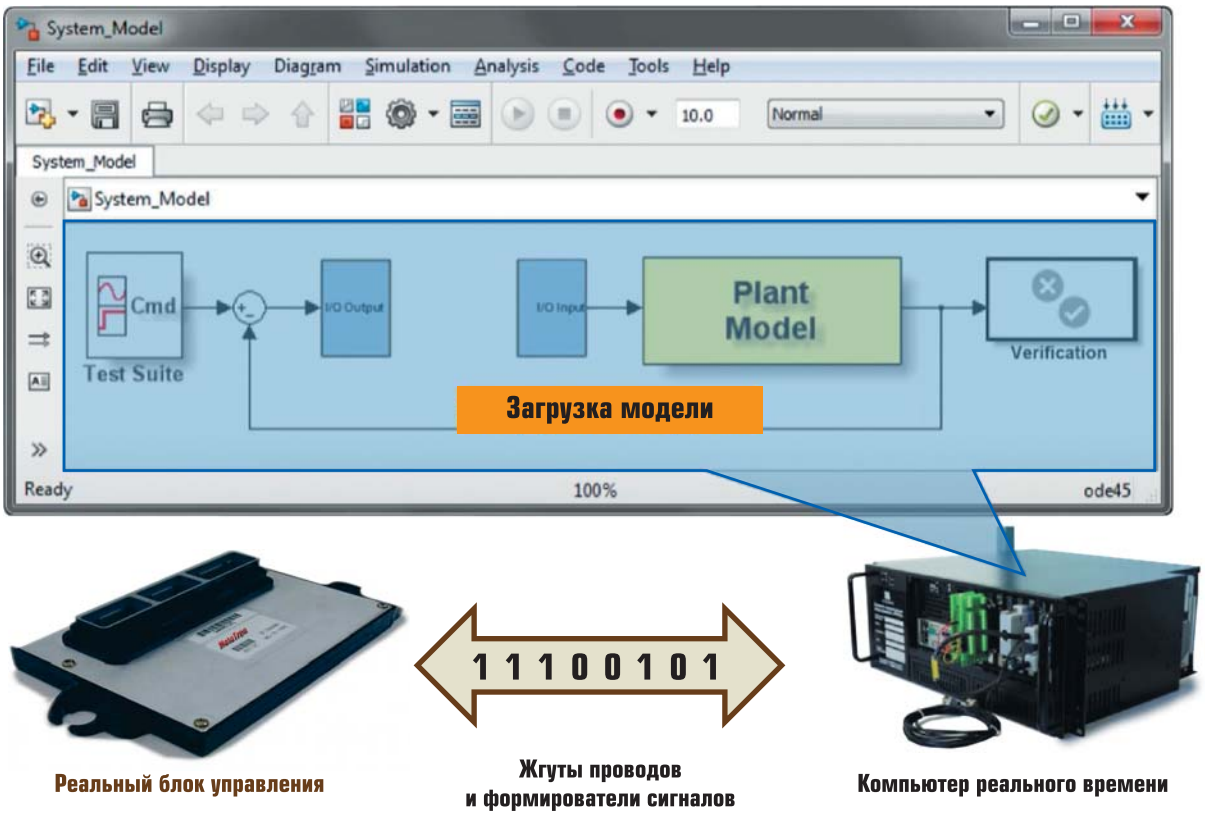

Рис. 11. Тестирование алгоритмов на реальном блоке управления с имитатором объекта управления [3]

реального объекта управления невозможно. Например, ещё нет реального объекта, или проводить натурные испытания на объекте крайне дорого, или нецелесообразно, например, алгоритм управления недостаточно отлажен, чтобы подключать реальный дорогостоящий объект управления. Поскольку это может быть опасно или может привести к разрушению объекта. Решением этих проблем является технология тестирования в рамках полунатурного моделирования. Разработчик может использовать специальный компьютер с операционной системой реального времени, который будет имитировать реальный объект. При этом алгоритм управления работает на целевом вычислителе или электронном блоке управления, подключенным к симулятору посредством интерфейса ввода вывода (рис. 11).

Модель может быть реализована как на одной системе реального времени, так и на нескольких системах, связанных интерфейсами. При этом разработчик сам решает, как разбить его систему на отдельные компоненты. Например, для систем управления силовым агрегатом автомобиля на одной машине может быть реализована модель коробки передач на другой – модель двигателя или одной из подсистем двигателя на третей модель динамики автомобиля. Следует отметить, что разные варианты применения тестирования в реальном времени могут применяться совместно, а также модифицироваться и даже называться немного иначе в зависимости от конкретной области применения.

Какую же пользу приносит полунатурное моделирование? Полунатурное моделирование позволяет запускать модель объекта управления в реальном времени с подключением их к реальным исполнительным механизмам, датчикам, блокам управления. Такое решение позволяет получить полноценную имитацию объекта управления. Полунатурное моделирование может применяться в широком спектре задач, включая создание симуляторов различных условий эксплуатации тренажеров для обучения персонала, например, пилотов или технологов, а также при создании полунатурных испытательных стендов для отладки систем управления. Полунатурное моделирование полезно, когда реальные объекты еще не доступны, например, только проектируются или находятся

в процессе производства, алгоритмы управления ещё недостаточно проверены и могут привести к разрушению объекта или представляют опасность для оператора. Время на стенде с реальных объектов может стоить дорого или сам стенд может находиться далеко в таких ситуациях полунатурное моделирование помогает сокращать издержки на испытание. Это также полезно для создания воспроизводимых повторяемых тестов, независящих, например, от износа объекта управления или внешних факторов. Или тогда, когда разработчик хочет проверить алгоритм управления в условиях, которые трудно достижимы или вообще недостижимы с реальных объектов, например, различные отказные состояния и аварийные режимы. Все это позволяет осуществлять систематическое полноценное тестирование системы. Полунатурное моделирование существенно сокращает цикла разработки и испытаний систем и повышают надежность разрабатываемой системы.

Но каким образом можно перейти от моделей, разработанных в МАТЛАБ и Simulink к тестированию в реальном времени? Разработчикам требуется запускать, тестировать и доказывать корректность функционирования алгоритмов управления в Simulink в реальных рабочих условиях. Как организовать такое тестирование в реальном времени? Для этого, с успехом, используется продукт под названием Simulink Real Time.

Simulink Real Time представляет собой программно-аппаратное решение, состоящее из трех компонентов.

Первый компонент – это операционная система жесткого реального времени с функцией многозадачности. Многозадачность – понятие из теории операционных систем, под которым подразумевается обеспечение возможности параллельной (или псевдопараллельной) обработки нескольких процессов. Однако идеальная многозадачность ОС возможна только в распределённых вычислительных системах или на многоядерных процессорах. Другими словами, многозадачностью можно назвать способ исполнения нескольких задач в один промежуток времени. При этом задачи распределяют между собой общие ресурсы (resources sharing), а также выполняют планирование (scheduling) задачи в очереди исполнения.

Второй компонент – это расширение для Simulink, позволяющее автоматически запускать модели в реальном времени под управлением операционной системы Simulink Real Time. Simulink Real-Time позволяет создавать приложения реального времени из моделей Simulink и запускать их на выделенном целевом компьютере, подключенном к физической системе. Он поддерживает симуляцию и тестирование в режиме реального времени, включая быстрое прототипирование алгоритмов управления, прототипирование систем ЦОС и машинного зрения и полунатурного моделирования (HIL).

С помощью Simulink Real-Time разработчик может расширить свои модели Simulink блоками драйверов, автоматически создавать приложения реального времени, управлять записью сигналов и выполнять интерактивные или автоматические запуски на выделенном целевом компьютере, оснащенном ядром реального времени, многоядерным процессором и интерфейсами ввода/вывода с поддержкой протоколов и ПЛИС.

Третий компонент – это специализированное оборудование, обеспечивающее работу системы Simulink Real Time и интерфейса с внешним миром. Итак, Simulink Real Time это расширение для Simulink, которое позволяет запускать модели в реальном времени на отдельном компьютере, работающем под управлением ядра Simulink Real Time. Данные моделирования (симуляции) могут записываться на жесткий диск машины реального времени и передаваться на хост компьютера для дальнейшего анализа и обработки. Разработчик может создавать графический интерфейс пользователя для управления работой модели и создания интерактивной визуализации. Создавать графические интерфейсы можно при помощи App Designer, а также встроенных инструментальных панелей. App Designer является богатой средой разработки, которая обеспечивает размещение и представление кода в полностью интегрированную версию редактора MATLAB® и большой набор интерактивных компонентов. Можно группировать файл установщика приложения непосредственно от панели инструментов App Designer, или можно создать автономное настольное приложение, или веб-приложение (требует MATLAB Compiler™).

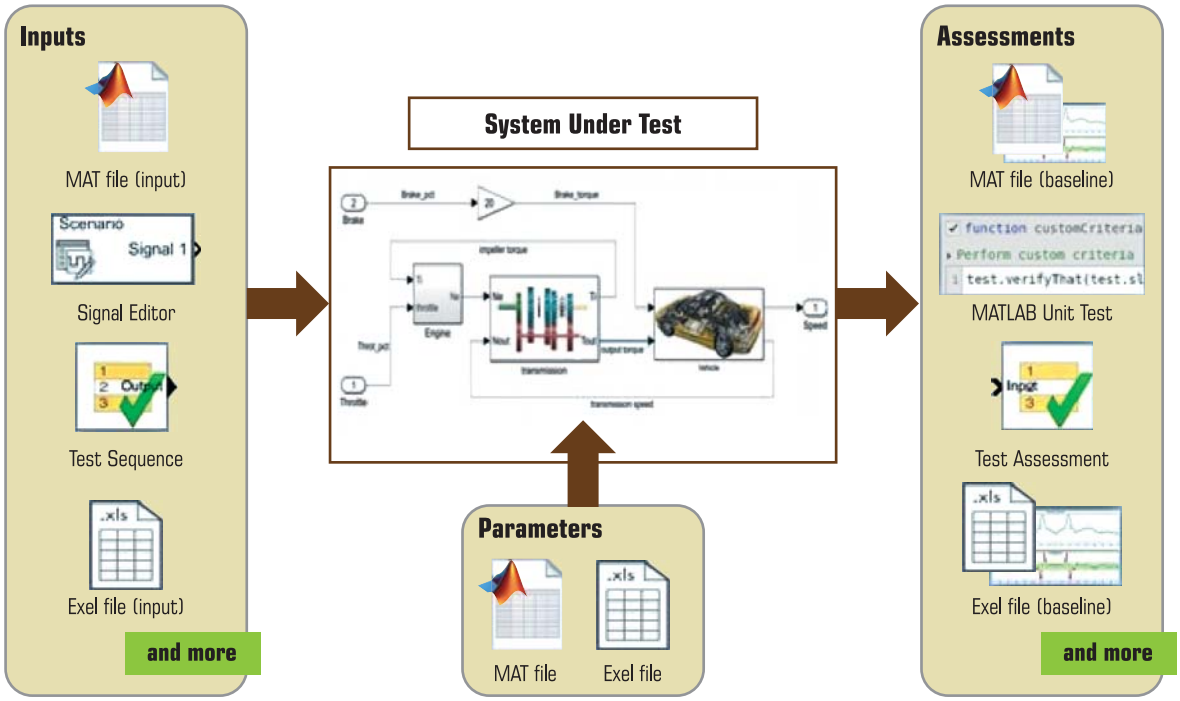

Рис. 12. Использование различных форматов для определения входов, выходов и параметров тестовых векторов [4]

На рис. 12 продемонстрировано использование различных форматов для определения входов, выходов и параметров тестовых векторов.

ПРОГРАММНО-АППАРАТНОЕ РЕШЕНИЕ ДЛЯ ОТЛАДКИ АЛГОРИТМОВ И СИСТЕМ В РЕАЛЬНОМ ВРЕМЕНИ КПМ "РИТМ"

ООО "РИТМ" производит в России комплексы полунатурного моделирования, на основе которых поставляет готовые стендовые решения для быстрого прототипирования или полунатурного тестирования встраиваемых систем. Комплексы поддерживают совместное использование с системой моделирования Simulink Real Time методологии модельно ориентированного проектирования [5].

КПМ "РИТМ" и Simulink Real Time представляют собой программно-аппаратное решение под ключ, обеспечивающее возможность тестирования систем управления в реальном времени. КПМ "РИТМ" может использоваться как для быстрого прототипирования, так и для полунатурного моделирования (рис. 13). РИТМ является Российской разработкой и поддерживается российскими инженерами.

РИТМ «Мобильный»
- ЦПУ: 2.8 ГГц
- ОЗУ: 4 Гб
- Eth: 3 шт
- Габариты: 150×200×225

РИТМ «Стандартный»
- ЦПУ: 3.3 ГГц
- ОЗУ: 4 Гб
- Eth: 2 шт
- Габариты: 482×177×450 (4U)

РИТМ «Производительный»
- ЦПУ: 4.2 ГГц
- ОЗУ: 4 Гб
- Eth: 2 шт
- Габариты: 482×177×450 (4U)

Рис. 13. Варианты исполнения стендов КПМ "РИТМ"

- **Аналого-цифровые модули общего назначения:**
  - АЦП, ЦАП, цифровые входы/выходы, ШИМ, квадратурный энкодер/декодер.
- **Специализированные интерфейсы:**
  - ARINC 429;
  - MIL-STD-1553;
  - RS-232/RS-422/RS-485;
  - CAN, J1939, XCP;
  - Fibre Channel (FC-AE-ASM);
  - Платы на базе ПЛИС.
- **Список постоянно пополняется...**

Рис. 14. Модули ввода вывода КПМ "РИТМ"

Рассмотрим дополнительную информацию о комплексе полунатурного моделирования.

Технические характеристики КПМ "РИТМ":

Корпус: ..... **19" 4U алюминиевый корпус**
Центральный процессор: ..... **Intel Xeon 3.5 ГГц, 4 ядра (или Intel Core i7)**
Память: ..... 4096 MB DDR3 (расширяется)
Диск: ..... 200 Гб SSD (расширяется)
Видео интерфейсы: 1xDVI-D, 1xVGA
Разрешение окна осциллографа до 1280×1024
Порты USB: ..... 4xUSB 3.0 и 2xUSB 2.0 на передней панели
Порты Ethernet: ..... 1xWG82579LM (GbE Phy), 1xWG82574L (PCIe GbE).
Разъемы PCI(e): ..... **3xPCI, 2xPCIe (x16), 2xPCIe (x4)**
Последовательные порты: 1xRS232/RS422/RS485
Температурный диапазон: ..... от 0 до 50 °C
Вес: ..... 6,5 кг (без модулей ввода/вывода)
Питание: ..... 400 Вт
Операционная система реального времени с бесшовной интеграцией с Simulink

Технические характеристики могут изменяться в зависимости от необходимости решения конкретных задач разработчиков. Комплексы КПМ "РИТМ" комплектуются различными модулями ввода/вывода в зависимости от задач конкретного разработчика. Поддерживается широкий спектр модулей и протоколов, включая аналого-цифровые платы общего назначения АЦП и ЦАП, цифровые входы и выходы, ШИМ и другими цифровыми интерфейсами до специализированных, таких как ARINC 429 и др.(рис. 14).

Особо следует отметить, что среди поддерживаемых плат расширения есть, в том числе, настраиваемые платы на базе ПЛИС, которые разработчики систем управления имеют возможность самостоятельно программировать под требуемый функционал. Этот список не является полным и постоянно пополняется в том числе драйверами для заказных модулей ввода/вывода, сделанных под требования конкретного заказчика.

Модули ввода/вывода поставляются вместе с драйверами, библиотеками Simulink, кабелями и разъемами.

На рис. 15 представлен пример конфигурирования АЦП и генерации ШИМ-сигналов из библиотеки блоков Simulink.

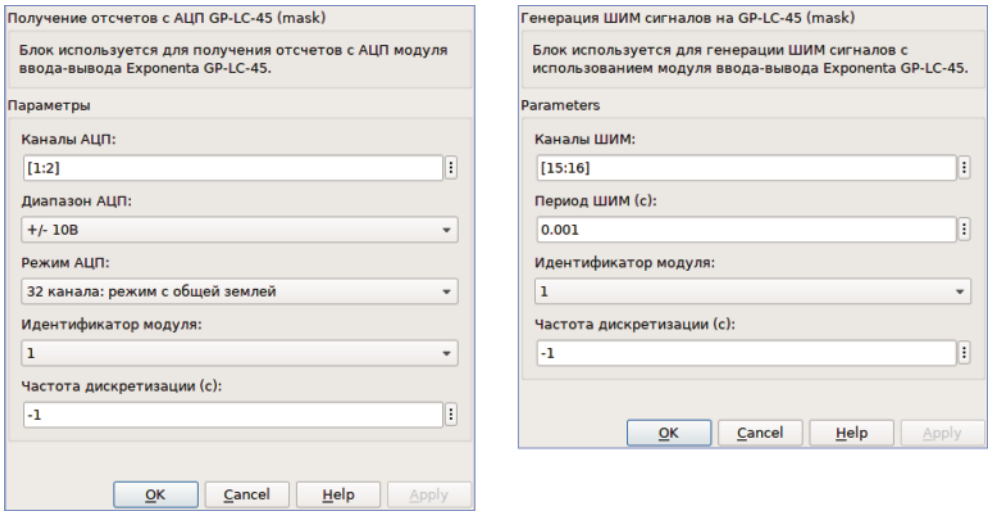

Рис. 15. Пример конфигурирования модулей АЦП и генерации ШИМ-сигналов из библиотеки блоков Simulink

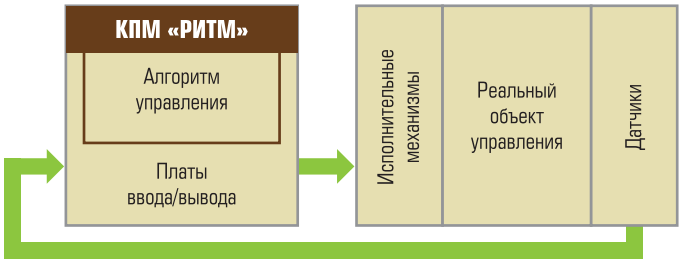

Рис. 16. Структурная схема: Быстрое прототипирование алгоритмов управления в реальном времени

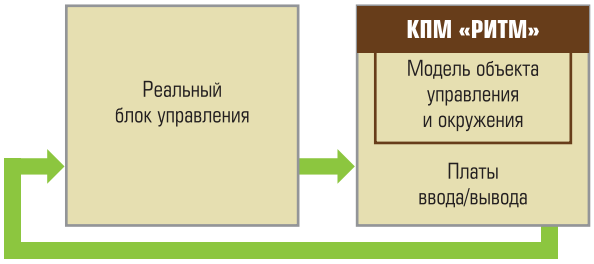

Рис. 17. Испытание блоков управления, бортовых вычислителей и контроллеров на полунатурных моделях

Эти блоки легко настраиваются из Simulink и общаются с настоящим “железом” в режиме реального времени. Вместе с машинами поставляются также тестовые модели, подтверждающие корректную работу как самой машины, так и интерфейсов ввода/вывода. Вместе с комплексом КПМ “РИТМ” разработчик получает гарантию, полную документацию, техподдержку и своевременные обновления программного обеспечения. Понятно, что один из самых важных вопросов, а сколько это стоит? Чтобы рассчитать стоимость, специалистам “РИТМ” нужно получить необходимые входные данные. Основная стоимость заключается в платах ввода/вывода в зависимости от типов интерфейсов, которые необходимы, и их количества.

Типичные применения конфигурации КПМ “РИТМ” включают полевые испытания, мелкосерийные партии блоков управления, лабораторные испытания. Разработчики имеют возможность разворачивать свои приложения реального времени на нескольких КПМ “РИТМ” без дополнительных лицензий.

ВАРИАНТЫ ИСПОЛЬЗОВАНИЯ КПМ “РИТМ” [5]

Вариант 1: ***Быстрое прототипирование алгоритмов управления в реальном времени (рис. 16).***

Постановка задачи

Необходимо протестировать придуманный алгоритм управления, когда он написан на “бумаге” или в виде модели, но не перенесен на электронный блок управления по причине недоступности самого вычислителя или долгих итераций при реализации алгоритма на вычислителе программистами.

Решение

Необходимо произвести подключение КПМ “РИТМ” к испытываемой системе вместо электронного блока управления, который недоступен, и произвести запуск модели алгоритма управления в режиме жесткого реального времени буквально по нажатию одной кнопки. Таким образом можно убедиться, что выбранный алгоритм управления работает корректно, если же нет, то производится коррекция и повторный запуск модели.

Такой подход позволяет проводить быстрые итерации при разработке алгоритмов управления в реальном окружении, не обладая профессиональными навыками программиста.

Испытание блоков управления, бортовых вычислителей и контроллеров на полунатурных моделях.

Вариант 2: ***Испытание блоков управления, бортовых вычислителей и контроллеров на полунатурных моделях (рис. 17).***

Постановка задачи

Необходимо провести испытания готового блока управления, однако этот процесс может быть очень дорог, требовать специальных и труднодостижимых условий, а также быть небезопасным для здоровья людей или сохранности испытываемого изделия.

Решение

Необходимо подключить разработанный блок управления к КПМ “РИТМ”, на котором запущена математическая модель объекта управления или окружения в режиме жесткого реального времени. Такой подход позволяет проводить систематические комплексные испытания в любых режимах эксплуатации без риска нанесения вреда изделию или испытателям в лабораторных условиях.

К такому стенду можно подключать реальные устройства, которые уже доступны, или постепенно заменять математические модели реальными компонентами, входящими в состав изделия.

КПМ “РИТМ” – это отечественное серийное решение, которое призвано освободить ресурсы компаний, и вместо построения стендов сфокусироваться непосредственно на проектировании систем. Это достигается за счет масштабируемой, легко настраиваемой, реконфигурируемой платформы, в которую входят универсальные аппаратные и унифицированные программные средства для математического моделирования и полунатурного моделирования (соответственно ММ и ПНМ).

Рассмотрим пример применения комплекса КПМ “РИТМ”. На Уральском заводе гражданской авиации была отработана бортовая система управления беспилотным летательным аппаратом с использованием комплекса КПМ “РИТМ”.

Основная задача была в том, чтобы убедиться в корректности разработанных алгоритмов для испытания бортовой системы управления (БСУ) беспилотника, разработанного сторонним предприятием [5].

Проведение натурных испытаний БСУ могло привести к потере прототипа БПЛА, что достаточно опасно и финансово затратно. Поэтому было принято решение сначала провести полунатурные испытания БСУ в лабораторных условиях. Для этого была создана модель объекта управления, а в качестве стенда для запуска данной модели в режиме реального времени был выбран КПМ “РИТМ”.

Электронный блок управления подключается к КПМ “РИТМ” по тем же интерфейсам, что и к самому БПЛА, этим обеспечивается полное соответствие между обменом данными на стенде и на самом объекте управления.

Имея точную математическую модель БПЛА, исполняемую в режиме жесткого реального времени, была получена возможность проверить БСУ БПЛА на различные сценарии и нештатные ситуации.

Уральский завод гражданской авиации – крупнейшее частное авиапромышленное предприятие в России, занимающееся производством и ремонтом авиационной техники и ее компонентов.

Новый БПЛА “Форпост-Р” представляет собой переработанный вариант базового “Форпоста”. В новом проекте предусматривается переработка конструкции и внутреннего оснащения с целью исключения зависимости от иностранных комплектующих. Кроме того, удалось получить прирост некоторых характеристик:

- продолжительность полета до 16-18 ч;
- максимальная скорость – 200 км/ч;
- дальность действия увеличена до 500 км.

Аппарат несет оптико-электронную аппаратуру, радиолокатор или иные средства наблюдения и обнаружения. Данные с таких средств передаются оператору в режиме реального времени.

В заключение обсудим вопрос мотивации использования математического и полунатурного моделирования. Сложность разрабатываемых систем управления растет, практически, в геометрической прогрессии. Это приводит к увеличивающимся требованиям при испытании систем управления. Проводить натурные испытания систем управления безусловно необходимо, но крайне дорого.

Промышленность ожидает следующее соотношение между различными типами испытаний [6]:

- Снижение доли натурных и летных испытаний с 85 % до 50 %.
- Увеличение доли испытаний на стендах ММ с 5 % до 20 %.
- Увеличение доли испытаний на стендах ПНМ с 10 % до 30 %.

В связи с этим разработчики сложных систем управления, в том числе авиационной техники и вооружений, стоят перед задачей сокращения объема натурных, в том числе, летных испытаний, за счет широкого использования математического моделирования и полунатурного моделирования (ММ и ПНМ соответственно).

ЦИТМ ЭКСПОНЕНТА [7]

Центр Инженерных Технологий и Моделирования создан для содействия русскоязычным инженерам в освоении передовых инструментов и технологий разработки и отладки сложных технических систем, алгоритмов и инженерного анализа данных.

ЦИТМ Экспонента дает импульс развитию и ускорению внутренних инженерных процессов заказчика, передает практические компетенции по технологиям модельно-ориентированного проектирования и Engineering Data Analytics на живых совместных проектах, через тренинги и семинары, консультационную поддержку, образовательные видео, интернет-ресурсы и примеры в исходных кодах или моделях.

Наша цель – постоянное укрепление инженерных команд заказчиков, которые знают “что делать”, нашим пониманием возможностей и технологий того “как это делать быстрее и дешевле”, а также помощь в вопросах поиска и развития кадров.

Опыт и решения ЦИТМ Экспонента помогают сотням учебных заведений усиливать практико-ориентированность учебного процесса и последовательно формировать вычислительные компетенции будущих инженеров и ученых. Мы хорошо понимаем организационные и управленческие особенности наших заказчиков, умеем приносить пользу и развитие в рамках ограничений и специфики ВУЗа или корпорации.

Список литературы

1. Электронный ресурс: *https://exponenta.ru/mbd*
2. Электронный ресурс: *www.betec.ru*
3. Электронный ресурс: *https://exponenta.ru/events/sozdanie-stendov-polunaturnogo-modelirovaniya*
4. Электронный ресурс: *https://exponenta.ru/SLRP#media_pc*
5. Электронный ресурс: *https://kpm-ritm.ru*
6. Электронный ресурс: *https://kpm-ritm.ru/m/207438917-Kompleks-polunaturnogo-modelirovaniya-ritm.html*
7. Электронный ресурс: *https://exponenta.ru*

Егоров Александр Александрович *– канд. техн. наук, доцент кафедры 301 “Системы автоматического и интеллектуального управления” Института №3 “Системы управления, информатика и электроэнергетика” МАИ, генеральный директор ООО “АВИАТЭКС”.*