

любые нештатные ситуации. Новые элементы внедряются в структурную часть модели, не затрагивая уже отработанный алгоритм функционирования.

При изучении рассматриваемых систем посредством имитационного моделирования нет возможности собрать данные по временам простоев устройств, входящих в состав АФЛ (так как линия находится на стадии проектирования). Поэтому для выбора распределений времен простоев устройств, входящих в состав АФЛ, воспользуемся следующим методом [4]. Допустим, что случайная величина X , характеризующая время простоя устройства, является непрерывной на интервале $[a, b]$, и ее функция плотности скошена вправо (опыт работы с реальными данными показывает [4, 5], что функции плотности таких случайных величин часто имеют подобную форму). Функции плотности многих теоретических непрерывных распределений вероятностей имеют такую форму (например, бета-, гамма-, Фишера и др.). Предположим, случайная величина X имеет бета-распределение. Известно, что функция плотности вероятности бета-распределения скошена вправо, если выполняется следующее соотношение для параметров этого распределения: $\alpha_2 > \alpha_1 > 1$. Среднее μ и мода c такого распределения задаются формулами [5]:

$$\mu = a + \frac{\alpha_1 \cdot (b-a)}{\alpha_1 + \alpha_2} \quad \text{и} \quad c = a + \frac{(\alpha_1 - 1) \cdot (b-a)}{\alpha_1 + \alpha_2 - 2}.$$

При наличии экспертных оценок μ и c можно решить эту систему уравнений и получить следующие оценки α_1 и α_2 :

$$\hat{\alpha}_1 = \frac{(\mu - b) \cdot (2c - a - b)}{(c - \mu) \cdot (b - a)} \quad \text{и} \quad \hat{\alpha}_2 = a + \frac{\hat{\alpha}_1 \cdot (b - \mu)}{\mu - a}.$$

При задании μ и c необходимо учитывать следующее условие: при $\mu > c$ плотность скошена вправо; при $\mu < c$ – скошена влево.

Аналогичные соотношения можно получить и для других распределений.

Перед проведением эксперимента с имитационной моделью для каждого устройства, входящего в состав АФЛ, ЛПП выбирает распределение вероятностей, характеризующее его время простоя, а затем задает значения μ и c .

Данная методика применяется ЗАО "Литаформ" для анализа альтернативных проектов АФЛ, выбора наилучших структур линий и при разработке алгоритмов управления ими.

Список литературы

1. Ковалев В.В. Введение в финансовый менеджмент. – М.: Финансы и статистика. 2004.
2. Бусленко Н.П. Моделирование сложных систем. – М.: Наука. 1978.
3. Древш Ю.Г., Зенькович М.В., Любченко А.С. Имитационное моделирование автоматизированных формовочных линии для изготовления отливок // Автоматизация в промышленности. 2008. №7.
4. Лоу А., Кельтон В. Имитационное моделирование. – СПб.: Питер. 2004.
5. Jerry Banks., John S. Carson II, Barry L. Nelson, David M. Nicol. Discrete-Event System Simulation, Fourth Edition. Pearson: Prentice Hall. New Jersey. 2004.

*Зенькович Михаил Валерьевич – инженер ЗАО "Литаформ",
Древш Юрий Георгиевич – д-р техн. наук, проф. кафедры "Управляющие интеллектуальные системы"
Национального исследовательского ядерного университета "МИФИ".
Контактный телефон (495) 324-34-54. E-mail: ydrevs@ya.ru zmv@litaform.ru*

ИТЕРАЦИОННАЯ РАЗРАБОТКА УПРАВЛЯЮЩИХ АЛГОРИТМОВ НА ОСНОВЕ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ ОБЪЕКТА УПРАВЛЕНИЯ

В.Е. Зюбин (Институт автоматики и электрометрии СО РАН)

Рассмотрена проблема использования имитационного моделирования при разработке управляющих алгоритмов уровня ПЛК. Предлагается итерационный способ создания программ ПЛК на базе концепции виртуальных объектов управления, позволяющий повысить эффективность и надежность разрабатываемых алгоритмов.

Ключевые слова: имитационное моделирование, виртуальные объекты управления, итерационная модель разработки ПО, программирование ПЛК.

Средства компьютерного моделирования уверенно вошли в практику промышленной автоматизации при проектировании новых производственных комплексов, подготовке операторов и оптимизации существующего, как правило, непрерывного ТП. Ведущие игроки на рынке SCADA-систем и программируемых контроллеров часто предлагают в дополнение к своим основным продуктам многофункциональные моделирующие комплексы. Программные пакеты Unisim (Honeywell), SimSci-Esscor (Invensys), Omega Land (Yokogawa), Mynah – MiMiC (Emerson), HYSYS (AspenTech), Ideas Simulation and Control (Andritz

automation), Tecnomatix Plant Simulation Tool (Siemens), Arena (Rockwell Automation) – лишь верхушка айсберга инструментальных средств моделирования, представленных на рынке. Наблюдаемые сдвиги в области промышленной автоматизации отражают общую тенденцию к тотальной компьютеризации: переход от математического и физического моделирования к моделированию компьютерному.

Несмотря на привлекательность и очевидную необходимость использования компьютерного моделирования на полевом уровне при программировании ПЛК, тестирование управляющих алгоритмов при

разработке средствами программной имитации пока не получило широкого распространения на практике.

В статье рассматриваются особенности существующих подходов к моделированию, в частности, подхода на основе использования программной имитации; обсуждаются проблемы, затрудняющие использование средств компьютерного моделирования при разработке и реализации алгоритмов управления уровня ПЛК; описываются базовые приемы, позволяющие преодолеть возникающие трудности; предлагается итерационный подход к разработке управляющих программ на основе виртуальных объектов управления.

Системы имитационного моделирования.

Базовая функциональность и область использования

Моделирование на протяжении столетий выступало (и выступает) как базовый инструмент научной и технической деятельности. Мы строим модели окружающих предметов и явлений, исследуем эти модели в различных условиях, чтобы затем использовать полученные знания при конструировании искусственных объектов (технических комплексов и инженерных конструкций) и при управлении процессами в физических, химических, биологических, социальных системах.

До недавнего времени исследование моделей было возможно только в двух вариантах. В первом варианте модель объекта задается в виде набора правил (формул, уравнений) и затем исследуется аналитическими методами. При этом исследователь подставляет интересные его начальные значения в правила, описывающие модель, и вычисляет результат либо в виде численных значений, либо в виде функций. Второй вариант — так называемое физическое моделирование — предполагает использование физических имитаторов — реальных объектов со свойствами, схожими со свойствами моделируемого объекта. При физическом моделировании исследователь физически воздействует на объект-имитатор и наблюдает за его поведением.

Физическое моделирование, привлекательное низкими требованиями к квалификации исследователя, сопряжено с принципиальными проблемами, которые резко сужают область его использования. Сложность подбора и конструирования имитатора, проблемы с безопасностью при исследовании в граничных условиях и практическая невозможность модификации — наиболее часто упоминаемые недостатки физического моделирования. В противоположность физическому моделированию математическое описание объекта и аналитическое исследование модели, хотя и может распространяться на неограниченно широкий класс объектов, справедливо связывается с трудоемкостью расчетов, отсутствием наглядности результатов и сложностью их интерпретации.

Компьютеризация современной жизни, в частности, широкое распространение компьютеров в области физико-технических исследований произвела ре-

волюцию в моделировании. Сначала использование ЭВМ практически вытеснило аналитические методы исследования за счет широкого внедрения в практику вычислительных методов, а затем и вовсе привело к смене базового исследовательского подхода: модель объекта задается исключительно программно. Широкое распространение подхода, основанного на идее программной имитации исследуемого объекта, позволяет ряду исследователей заявлять о наступлении "эры компьютерного моделирования" [1].

Инструменты компьютерного моделирования получили широкое распространение в промышленности. Программные пакеты предоставляют автоматизированные средства и библиотеки элементов, позволяя обыгрывать различные варианты размещения оборудования, моделировать материалопотоки, создавать обучающие комплексы для операторов и технологов [2], готовить программирующих специалистов уровня ПЛК [3], оптимизировать ТП [4]. Визуализация — неотъемлемый атрибут программной имитации — радикально упрощает работу исследователя, скрывая многоэтажные формулы математической модели за наглядностью элементов компьютерной графики.

Программная имитация позволяет [1]:

- манипулировать течением модельного времени, ускорять медленные процессы и замедлять быстротекущие, повышая эффективность восприятия исследователем поведения системы;
- изменять параметры запуска модели в широком диапазоне, тем самым предоставляя исследователю мощный эвристический инструмент, чтобы разрабатывать и выдвигать новые гипотезы, модели, теории;
- существенно сократить число натуральных экспериментов, во многих случаях заменяя их численными, что особенно важно в случае рисков (например, аварийный останов производства при исследовании систем со скрытыми связями) или этических последствий (например, при решении задачи оптимизации числа рабочих мест);
- решать педагогические задачи, в частности, сокращая сроки и повышая качество подготовки персонала (операторов, технологов);
- упростить анализ взаимозависимостей, предоставляя возможность настройки, упрощения модели, исключения тривиальных параметров.

В дополнение к этому программный имитатор может быть использован в качестве инструмента маркетинга — для демонстрации, обсуждения и обоснования предлагаемого решения [5]. Принцип "лучше один раз увидеть" позволяет не только сократить объяснения, но и повысить убедительность утверждений.

Первоначально в качестве средств создания программных имитаторов использовались универсальные языки программирования (Algol, Fortran), позже начали появляться специализированные языки программирования (GPSS, Simula). Однако подобные средства не могут претендовать на роль промышленного инструмента, так как сложность освоения, трудоемкость моде-

лирования, слабые средства визуализации результатов сильно ограничивают их применение для решения инженерных задач по сравнению со специализированными пакетами визуального программирования и библиотеками типовых элементов [5].

Средства имитационного моделирования при разработке и реализации алгоритмов управления уровня ПЛК

При создании управляющих алгоритмов уровня ПЛК имитационное моделирование востребовано не меньше, чем при моделировании предприятия в целом. Имитационное моделирование промышленных объектов автоматизации позволяет существенно повысить эффективность подготовки программирующих специалистов: изучить типовые стратегии управления, применимые в различных производствах, обеспечить наглядность решаемых задач и ввести игровой элемент (что также позитивно сказывается на процессе обучения) [3]. Но, несомненно, еще большую значимость имитация объектов управления имеет в реальных проектах. Текущая практика промышленной автоматизации такова, что в подавляющем большинстве случаев пусконаладочные работы на новом объекте превращаются в своеобразный "момент истины": первый запуск ПО фактически означает начало его тестирования на уровне ПЛК. Поэтапный ввод ПО в составе системы управления (автономное тестирование, комплексное тестирование) не обеспечивает (и не может обеспечить) обнаружение ошибок на логическом уровне алгоритма, поскольку по сути сводится к проверке корректности подсоединения сигнальных линий.

Возможность моделирования объектов управления отсутствует и в пакетах разработки. Средства для отладки управляющего ПО, предоставляемые производителями программных пакетов на базе МЭК 61131-3, ограничиваются возможностью "оторвать" сигнальный кабель от датчика и назначить его значение вручную. То есть в качестве имитатора объекта выступает сам программист, что делает невозможным проверить даже логическую корректность сколь угодно сложного алгоритма, не говоря уже о тестировании его устойчивости или динамических характеристик. Другие технологии верификации ПО, такие как статический анализ исходного кода, испытания на автоматизируемом объекте, хотя и обладают огромным потенциалом, но, тем не менее, не в состоянии заменить тестирование как доминирующую технологию [6], которая в случае алгоритмов управления предполагает активное моделирование поведения объекта управления.

Объективная причина, обуславливающая сложившуюся практику, заключается в ограниченности средств МЭК 61131-3 – концептуальной разнородности языков, недостаточности выразительных возможностей, несоответствии специфике автоматизации в части обеспечения необходимых процессуальных и структурных свойств алгоритма [7]:

- событийности (управляемости по событиям);
- синхронизма (возможность работы с временными интервалами);
- средств организации взаимодействия параллельно-исполняемых процессов, в частности, в виде иерархических структур;
- средств абстрагирования и создания программных модулей, понятийно отражающих автоматизируемую технологию.

В результате проверка алгоритма откладывается до этапа пусконаладочных работ на объекте автоматизации. Степень готовности алгоритма до этого момента неизвестна.

Кроме того, поскольку разработка ведется в рамках водопадной модели, то разработчики испытывают серьезные психологические нагрузки – в команде растёт нервозность, вызванная неуверенностью в качестве уже сделанной работы, вероятностью концептуальных ошибок, чреватых "глобальным рефакторингом" проекта (переписыванием "с нуля"). Вполне объяснимое желание избежать стресса и невозможность стороннего контроля качества может привести к тому, что исполнители-программисты начинают имитировать уже свою работу, то есть попросту ее не делать. Однако это не сильно сказывается на сроках ввода объекта в эксплуатацию, поскольку тестирование "сырого" проекта на объекте может привести не только к аварии, требующей длительных восстановительных работ на объекте управления, но и выявить необходимость смены стратегии автоматизации, сопряженной с внесением серьезных изменений в исходные тексты программ.

Риски растут по мере укрупнения выполняемого проекта и сложности управляющего алгоритма.

Практикуемые приемы снижения рисков

Заметно минимизируют свои риски компании, уже имеющие за своими плечами "портфель" выполненных проектов и штат "проверенных в боях" исполнителей. У таких компаний имеется в наличии готовые стратегии автоматизации, библиотеки алгоблоков, методики отладки алгоритмов на объекте автоматизации, то есть опыт "сын ошибок трудных". Именно поэтому начинающие компании в области промышленной автоматизации особо остро испытывают дефицит в клиентах. Этим же обстоятельством во многом объясняется и специализация компаний по отраслям, наблюдаемая на практике.

В продвинутых компаниях используются три подхода для предварительной отладки ПО уровня ПЛК.

Физические имитаторы. ПО ПЛК отлаживается на физических имитаторах, которые воспроизводят автоматизируемый процесс в той или иной степени детализации. Физические имитаторы подсоединяются к ПЛК через имеющиеся датчики и исполнительные органы либо через их аналоги. Этот подход сопряжен с уже описанными проблемами, сопутствующими физическому моделированию: трудностью в выборе

Итерационный метод разработки управляющих алгоритмов на основе виртуальных объектов

имитатора, невозможностью модификации модели, сложностями с использованием тех же датчиков и исполнительных органов, что и на объекте, а также высокой стоимостью поддержания оборудования в работоспособном состоянии [3].

Программно-аппаратные имитаторы. Более продвинутый подход, применяемый на практике, — использование в качестве имитатора отдельного программно-аппаратного комплекса, "комплиментарного" создаваемой системе управления (рис. 1). Такой цифровой имитатор содержит программу, воспроизводящую поведение объекта управления, и формирует входные аналоговые/цифровые сигналы для отлаживаемой системы управления через свои модули выходов на основании управляющих сигналов, считываемых через аналоговые и цифровые модули входов.

В отличие от физического имитатора такая схема позволяет тестировать алгоритмы управления для широкого спектра объектов и изменять поведение моделируемого объекта. В силу упоминавшейся комплиментарности системы управления и объекта подход позволяет также применять уже используемые языки программирования, что дает очевидное преимущество по сравнению со специализированными средствами компьютерной имитации. Еще одно достоинство подхода — простая коммутация алгоритма от имитатора к реальному объекту управления. Однако программно-аппаратное моделирование предполагает дополнительные затраты на технические средства, примерно равные затратам на аппаратуру системы управления.

Встраиваемые программные имитаторы. Иногда применяется подход, при котором для тестирования ПО ПЛК входные сигналы "отрываются" от физических модулей ввода и формируются программно. При этом в алгоритм управления внедряются специализированные программные модули — имитаторы оборудования. Этот подход также позволяет в какой-то степени отладить алгоритм, но тесная интеграция тестируемой и тестирующей части ПО вызывает сложности при последующем удалении тестирующего кода из алгоритма управления. По этой же причине серьезные проблемы возникают и при попытке организовать групповую работу над проектом.

Несомненно, когда дело касается такого критического производства, как атомная электростанция, средств на тестирование и моделирование не жалеют. Хотя анализ случившихся за последнее время техногенных катастроф показывает, что тестирование ПО на моделях проводится явно в недостаточном объеме¹.

¹ Причина аварии не обязательно кроется в программных ошибках. ПО может работать и без сбоев, но авария все равно может произойти, например, при ошибках в спецификации, отсутствии необходимых блокировок на действия оператора или запутанном пользовательском интерфейсе. Разумеется, причина аварий может заключаться и в "чистых" сбоях аппаратных средств, но подавляющее большинство проблем в системе управления выявляются при тестировании ПО на моделях.

Чтобы упростить тестирование ПО ПЛК, сделать его разработку более комфортным, предлагается использовать для отладки ПО уровня ПЛК идею виртуальных лабораторных стендов (ВЛС), развиваемую в Институте автоматизации и электрометрии СО РАН [3].

ВЛС, созданные для обучения студентов программированию управляющих алгоритмов, базируются на концепции виртуального объекта управления (ВОУ) — программного имитатора некоторого объекта управления. Код ВОУ исполняется независимо от алгоритма управления, создаваемого студентом. Обмен данными между ВОУ и алгоритмом управления реализован через массивы, поэтому при изменении алгоритма управления нет необходимости в перекоммутации связей. Алгоритм кодируется на Си-подобном языке Рефлекс [8], ориентированном на описание алгоритмов управления в виде множества независимых процессов. Присущий языку событийный полиморфизм обеспечивает описание поведенческих алгоритмов в широком диапазоне. Код алгоритма, написанный на языке Рефлекс, преобразуется транслятором в текст на языке Python, который затем передается на исполнение интерпретатору языка Python. ВЛС реализованы на базе пакета LabVIEW [9], активно используемом в промышленной автоматизации в качестве аналога так называемых софт-ПЛК, расширенного средствами создания операторского интерфейса и интеграции периферийного оборудования [3]. Выбор LabVIEW позволяет использовать графику и визуализировать поведение ВОУ, скрыть от обучаемого детали реализации и предоставить ВЛС единым приложением с простым интерфейсом. Возможность пакета LabVIEW создавать исполняемые файлы позволяет тиражировать ВЛС без дополнительных лицензионных отчислений.

Поскольку управляющий алгоритм и алгоритм поведения объекта управления комплиментарны по интерфейсам и имеют одинаковую процессную природу, поведение ВОУ также можно описать средствами языка Рефлекс, что используется при создании сложных объектов. При этом ВЛС включает два интерпретатора Python, один из которых реализует алгоритм управления, а другой имитирует объект управления. Схема замыкания входов на выходы алгоритма управления и ВОУ аналогична представленной на рис. 1. При этом для отладки алгоритма необходим только ПК. Другой важный момент — если не требуется из-

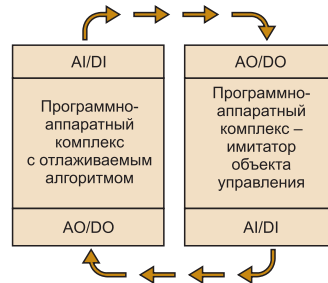


Рис. 1. Схема отладки алгоритма управления с использованием программно-аппаратного имитатора (AI — аналоговые входы, DI — цифровые входы, AO — аналоговые выходы, DO — цифровые выходы)

менения в части графики, то изменения в системе касаются только описаний алгоритма управления и поведения управляемого объекта.

Возможность отделять коды алгоритма и программного имитатора объекта управления позволяют не только тестировать алгоритм управления, но и использовать при разработке алгоритма итерационную схему (рис. 2). Итерационная разработка алгоритма управления начинается с фиксации частичной модели объекта управления, для которой пишется программный имитатор и алгоритм управления. После этого проверяется взаимное соответствие алгоритма и программного имитатора объекта. Обнаружение несоответствия означает наличие ошибок либо в алгоритме, либо в программной модели объекта, которые устраняются. В случае успешного прохождения теста частичная модель объекта управления усложняется. Цикл повторяется до реализации полноразмерного имитатора объекта управления.

Поскольку создание модели и алгоритма ведется независимо, радикально снижается вероятность не выявленных до начала пусконаладочных работ ошибок, которая может быть оценена как произведение вероятностей ошибок в алгоритме управления и ВОУ:

$$P_{\text{остаточная}} \approx P_{\text{ошибки АУ}} \cdot P_{\text{ошибки ВОУ}}$$

Для проведения работ требуются только ПК, а само ПО представлено тремя независимыми частями: 1) базовой функциональностью с визуализацией, оформляемой в виде исполняемого файла; 2) текстовым описанием поведения объекта; 3) текстовым описанием алгоритма управления. Поэтому, во-первых, программирование может вестись автономно от разработки аппаратуры, во-вторых, разработка алгоритма может вестись разными группами и, возможно, даже разными организациями, в-третьих, к процессу разработки могут привлекаться представители заказчика. Последнее особенно важно, поскольку коммуникативные проблемы между исполнителем и заказчиком (например, разночтения в техническом задании, ошибки в нем, вариативность реализаций интерфейса оператора), обусловленные математическим подходом при формулировке технического задания, могут стать причиной затягивания окончания работ (хотя бы потому, что обычно "всплывают" только на этапе пуско-наладки).

Появляется возможность оперативно согласовать с заказчиком текущие вопросы по интерфейсу оператора, предметно обсудить разночтения и устранить неоднозначность технического задания. В конечном итоге разработка ведется без потенциально опасной неопределенности, что позволяет избежать психологических нагрузок, свойственных водопадной модели разработки. Появляется возможность контролировать выполнение проекта и использовать для оценки степени готовности численные показатели, например, число обнаруженных ошибок как функцию времени [6].

Особый интерес описанный подход представляет в случае применения для программирования пакета

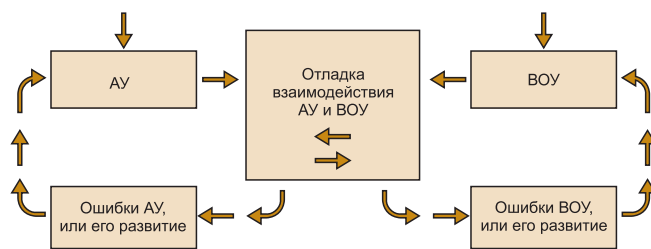


Рис. 2. Итерационная модель разработки алгоритма управления (АУ) с тестированием на виртуальном объекте управления (ВОУ)

LabVIEW, расширенного языком Рефлекс, так как при этом сохраняется исходная однородность программных средств и концепций. В этом случае вовлечение в процесс разработки представителей эксплуатирующей организации позволяет не только отработать эргономичность пользовательского интерфейса, но и заблаговременно ознакомить с ним будущих операторов и инструкторов.

Заключение

Для снижения рисков при вводе систем управления в эксплуатацию выдвинута концепция итерационной разработки управляющих программ на основе ВОУ. Предложен метод реализации концепции в среде LabVIEW, расширенной Си-подобным языком процесс-ориентированного программирования Рефлекс.

Использование метода в реальных проектах по автоматизации позволяет:

- тестировать создаваемые алгоритмы, начиная с самых ранних стадий разработки, внедрить итерационную модель разработки для случая промышленной автоматизации;
- обеспечить контроль процесса создания управляющих алгоритмов и снизить психологическую нагрузку на коллектив разработчиков;
- сократить время выполнения проекта и имеющиеся риски этапа пуско-наладки;
- гибко расширять круг лиц, участвующих в процессе разработки, в частности, чтобы своевременно выявлять и устранять ошибки в техническом задании.

Предложенный подход применен в проекте по созданию системы тестирования радиоэлектронной аппаратуры. Проект реализовывался в среде LabVIEW. В рамках работы были созданы программные модели контролируемого объекта и плат коммутации контрольных точек объекта к мультиметру. Собранное приложение, включающее алгоритм управления, интерфейс оператора и виртуальный объект контроля, обеспечило возможность проверки базовой функциональности системы исключительно на ПК. К разработке, которая носила итерационный характер, активно привлекался заказчик, находящийся в другом городе. Обмен приложением обеспечило оперативную проверку функциональности и сбор замечаний со стороны заказчика, которые позволили скорректировать техническое задание. Организация работ на

основе предложенных принципов привела к заметному снижению сроков разработки (по оценкам, примерно в 1,5 раза) и времени проведения пусконаладочных работ (приблизительно в 5 раз).

Список литературы

1. Hartmann S. The World as a Process: Simulations in the Natural and Social Sciences // in: R. Hegselmann et al. (eds.), Modelling and Simulation in the Social Sciences from the Philosophy of Science Point of View, Theory and Decision Library. Dordrecht: Kluwer 1996. <http://philsci-archive.pitt.edu/archive/00002412/01/Simulations.pdf>
2. Дозорцев В.М., Крейдлин Е.Ю. Современные автоматизированные системы моделирования ТП // Автоматизация в промышленности. 2009. № 6.
3. Зюбин В.Е., Калугин А.А. Виртуальные лабораторные стенды: обучение программированию задач промышленной автоматизации // Промышленные АСУ и контроллеры. 2009. № 2.
4. Гурко Н.С., Федоров В.И., Лисицын Н.В. Имитационное моделирование и оптимизация процесса дегидрирования n-парафинов в моноолефины // Автоматизация в промышленности. 2009. № 9. С. 11-14.
5. Медведев В. Имитационное моделирование в промышленности // Plm news. Инновации в промышленности. Май 2008. http://www.plm.automation.siemens.com/ru_ru/Images/Plant%20Simulation_tcm802-92342.pdf
6. Мейер Б. Семь принципов тестирования программ // Открытые системы. 2008. №7.
7. Зюбин В.Е. Пути расширения языка ST из состава МЭК 61131-3 для задач промышленной автоматизации // Приборы и системы. 2009. №3.
8. Зюбин В.Е. "Си с процессами": язык программирования логических контроллеров // Мехатроника. 2006. № 12.
9. Бутырин П.А., Васьяковская Т.А., Каратаева В.В., Материкин С.В. Автоматизация физических исследований и эксперимента: компьютерные измерения и виртуальные приборы на основе LabVIEW 7. М.: ДМК Пресс, 2005.

Зюбин Владимир Евгеньевич — канд. техн. наук, руководитель тематической группы "Языки технологического программирования" Института автоматики и электрометрии СО РАН, доцент каф. "Информационно-измерительные системы" Новосибирского государственного университета. Контактный телефон (383) 330-71-62. E-mail: zubin@iae.nsk.su

ПРОЕКТИРОВАНИЕ НА ОСНОВЕ МОДЕЛЕЙ И АВТОМАТИЧЕСКОЙ ГЕНЕРАЦИИ КОДА

Компания B&R Industrial Automation

Представлен программный продукт B&R Automation Studio Target for Simulink, реализующий взаимодействие между программными пакетами имитационного моделирования от компании The MathWorks и средой Automation Studio от компании B&R, в которой осуществляется программирование контроллеров. Указаны преимущества от совместного использования инструментов моделирования и проектирования.

Ключевые слова: программирование, имитационное моделирование, промышленные контроллеры.

Мир машиностроения постоянно претерпевает изменения. Происходит разработка новых и оптимизация имеющихся контроллеров, механического и электронного оборудования. Но, если взглянуть на используемые методы разработки, то покажется, что время остановилось. "Устойчивое развитие" и "возможность повторного использования" — эти модные слова редко звучат применительно к ведению разработок в области машиностроения. Очень часто серийно выпускаемые машины проектируются с нуля, хотя в новых проектах можно было бы использовать большую часть уже готовых решений.

Однако оптимизация методов проектирования сулит огромный потенциал экономии для машиностроителей. С одной стороны, повторное использование существующих схем в новом оборудовании может существенно сократить процесс разработки, с другой — использование методов математического моделирования и симуляции позволяет отобразить функционирование выпускаемых машин максимально точно, что облегчает процесс их обслуживания.

Хотя сейчас наиболее распространенным является проектно-ориентированный принцип разработки, методы проектирования на основе моделирования постепенно проникают в сферу промышленной автоматизации. Несмотря на начальные затраты, связанные с анализом системы и разработкой модели, инвестиции машиностроите-

ля быстро окупаются. Например, созданные имитационные модели можно сравнительно недорого обновлять для разработки оборудования будущих поколений. Имитационная модель позволяет в четко документированном виде аккумулировать опыт и "ноу-хау", накопленные в различных отделах компании. Кроме того, имитация помогает обнаруживать концептуальные ошибки на ранних этапах разработки, позволяя обходиться без изготовления дорогостоящих прототипов (рис. 1).

Усовершенствованные алгоритмы управления повышают не только качество выпускаемой продукции, но и производительность машин.

B&R Automation Studio Target for Simulink — инструмент, объединяющий два мира

Новые методы разработки требуют нового инструмента, полностью совместимого с проектированием на основе имитационного моделирования и автоматической генерации кода. Программные средства компаний B&R и The MathWorks дают современному разработчику возможность с максимальной выгодой использовать преимущества устойчивого развития проектов, основанных на моделях.

Программная среда разработки от B&R — Automation Studio — предназначена для конфигурирования и программирования контроллеров, создания проектов визуализации, а также реализации логики в зада-