

СОЗДАНИЕ ИМИТАЦИОННЫХ МОДЕЛЕЙ ТЕХНОЛОГИЧЕСКИХ ПРОЦЕССОВ ДЛЯ ОТЛАДКИ ПРОГРАММ ПЛК И ПРОЕКТОВ SCADA/HMI

Н.Н. Луцкая, А.Н. Пупена, С.Н. Швед (Национальный университет
пищевой промышленности)

На этапе предварительной отладки программного обеспечения ПЛК и SCADA/HMI в системах управления сложными объектами существует возможность использования имитационных моделей. Предложены практические подходы к созданию имитационных моделей с использованием ресурсов ПЛК на языках МЭК 61131.

Ключевые слова: имитационное моделирование, отладка, ПЛК, SCADA/HMI.

Актуальность

Одним из наиболее ответственных этапов жизненного цикла АСУТП является отладка ПО ПЛК а также SCADA/HMI. В большинстве случаев отладку проводят в несколько этапов:

- после написания функционально законченной части программы, до внедрения на объекте (грубая отладка);
- тонкая отладка программы на объекте во время пуско-наладочных работ;
- доотладка программы в режиме эксплуатации.

На первом этапе разработчик пытается максимально приблизить программу к рабочему состоянию еще во время ее создания. Отсутствие технологического объекта требует от разработчика ПО ПЛК имитировать сигналы от датчиков или изменять соответствующие входные переменные в соответствии с алгоритмом функционирования объекта. Разработчик должен четко представлять себе, как работает технологический объект, чтобы имитировать сигналы в правильной последовательности и в нужных диапазонах, а также внимательно проверять реакцию программы на эти действия. Для сложного объекта отладить программу путем перебора заданных комбинаций значений переменных процесса довольно проблематично. В таких ситуациях могут помочь имитационные модели объектов управления.

Использование имитационных моделей способствует также быстрой отладке проектов SCADA/HMI. Кроме того, современные подходы в системной и программной инженерии требуют периодического участия представителей заказчика в оценке работы ПО верхнего уровня, что без наличия связи с реальным объектом требует имитации переменных процесса.

С развитием компьютерной техники имитационное моделирование набирает все большую популярность как для проведения исследований, так и для построения тренажеров [1]. В настоящее время существует много программных пакетов, предназначенных для имитационного моделирования в РВ. Использование таких пакетов для задач отладки программ ПЛК и проектов SCADA/HMI ограничено следующими факторами:

- 1) дополнительные затраты времени на изучение среды моделирования;
- 2) неоправданная дороговизна для объектов средней сложности;

3) излишняя избыточность и сложность применительно к поставленным задачам;

4) наличие аппаратных ресурсов (входов/выходов реального контроллера) или программного интерфейса (например, OPC).

Первые три фактора создают серьезный барьер для любого внедряемого ПО, будь то САПР или среда имитационного моделирования. Кроме того, современные среды разработки ПЛК средней и большой информационной сложности поставляются со встроенными имитаторами исполнительной системы (ИС) контроллера, способными также работать совместно с имитаторами ИС SCADA/HMI. Для проверки работы программ на таких ИС не нужно оборудование, а его отладка возможна с использованием только программных имитационных моделей.

Исходя из вышесказанного, для задач отладки наиболее практичным способом имитационного моделирования является использование программных ресурсов контроллера, в котором реализованы проверяемые алгоритмы. Такой подход не нов, и используется многими разработчиками прикладного ПО для ПЛК и SCADA/HMI. Рассмотрим практические рекомендации по созданию и использованию таких имитационных моделей с использованием языков МЭК 61131-3.

Имитационная модель в структуре программы ПЛК

Имитационная модель в данном контексте является частью исполнительного проекта ПЛК. То есть, это часть программы ПЛК (включая все дополнительные переменные), имитирующая работу объекта. Модель может быть выполнена как программная секция, функция или функциональный блок, написанные на тех же языках МЭК 61131-3, что и основная программа управления (далее по тексту эта часть будет называться программой имитации). Таким образом, исполнительный проект ПЛК во время отладки будет состоять из двух типов программных элементов: программ управления и программ имитации.

Следует отметить, что создание имитационной модели имеет смысл только для объектов средней и большой алгоритмической сложности. ПЛК, а также среды разработки для них, используемые в такого уровня системах управления, можно охарактеризовать следующими показателями:

- наличие некоторой ресурсной избыточности касательно решаемых задач (например, объем памяти,

максимальное число программных элементов, таймеров и т. п.);

- разделение программ на части (POU (Program Organization Unit), секции, функции, функциональные блоки);

- кроме адресного обращения к переменным имеется возможность символьной адресации;

- наличие имитатора исполнительных систем ПЛК в среде разработки;

- наличие развитых средств поиска перекрестных ссылок в среде разработки;

- наличие элементов объектно-ориентированного программирования и/или возможности создания пользовательских функций и типов функциональных блоков.

В качестве примера можно привести ПЛК Premium/Quantum/M340 (Schneider Electric, среда Unity PRO); S7-1200/S7-1500/S7-300/S7-400 (Siemens, среда Step7); FX/Q (Mitsubishi Electric, среда GX IEC Development), среда ISAGraph (ICS Triplex), среда CodeSys (3S-Smart Software Solutions) и т. п.

В работе [1] указаны некоторые сложности, возникающие при встраивании программ имитации в структуру программы управления. Учитывая приведенные выше показатели ПЛК и сред разработки, эти проблемы могут быть решены, с помощью ряда приемов и рекомендаций:

- программа имитации не должна быть частью программы управления, а выделена в отдельный структурный элемент (секция, функция, функциональный блок);

- переменные, константы и остальные объекты языка ПЛК (таймеры, счетчики, экземпляры функциональных блоков), созданные исключительно для имитации, нужно идентифицировать и организовывать таким образом, чтоб их можно было быстро найти и при необходимости удалить (например, использовать специфические префиксы в именах переменных);

- программа имитации должна вызываться в контексте той же задачи, что и программа управления;

- программа имитации должна выполняться в задаче перед вызовом программы управления;

- следует учитывать время выполнения программы имитации, то есть длительность выполнения задачи (Task) ПЛК будет большей, чем без программы имитации;

- рекомендуется создавать сначала программу имитации объекта, а затем программу управления;

- рекомендуется использовать графические средства отладки или ресурсы SCADA/HMI.

Внедрение программы имитации в структуру проекта может производиться различным способом, в зависимости от особенностей платформы и удобства разработчика. Возможно внедрение секции имитации в самом начале циклической задачи, тогда появляется возможность программной имитации входных переменных. В других случаях можно разместить секцию имитации непосредственно перед программой управ-

ления. Перед пуско-наладочными работами на объекте секция имитации удаляется или отключается.

Особенности построения имитационных моделей, используемых для отладки программ ПЛК и SCADA/HMI

В настоящее время существует разнообразный математический аппарат создания имитационных моделей ТП, например, методы статистического анализа, нейронные сети, теория нечетких множеств и т. д. В отличие от многих подходов, аналитические имитационные модели, построенные на фундаментальном моделировании процессов (кинетики, тепло- и массообмена, гидравлики и пр.), не требуют массивов исторических данных о переменных объекта, а строятся на основании физико-химических знаний о ТП. Для разработчика алгоритмов управления эти знания являются обязательными. Поэтому построение аналитической имитационной модели ТП не займет много времени, если выработать методику и не усложнять модель ненужными для отладки деталями. Выделим некоторые особенности построения имитационных моделей:

- 1) модель должна отображать сущность ТП, то есть быть адекватной сути исследованию, однако требование к точности могут быть занижены;

- 2) модель должна быть устойчивой, что предусматривает заранее заданный диапазон изменения переменных имитационной модели;

- 3) модель должна предусматривать инициализацию, то есть переход в начальное состояние в любой момент времени;

- 4) учитывая использование ресурсов ПЛК, модель должна быть максимально простой и описывать только те элементы, которые непосредственно участвуют в управлении или существенно влияют на него;

- 5) модель должна иметь возможность работать в реальном, ускоренном или замедленном масштабе времени.

Многие среды разработки включают библиотеки с реализацией типовых звеньев, которые могут быть использованы для построения имитационной модели процесса. Однако эти блоки, как правило, являются линейными и использование их в удалении от рабочей точки и в переходных режимах ведет не только к большим отклонениям переменных имитационной модели и объекта управления, но и к неправильному воспроизведению функционирования объекта. Кроме того, не все ТП описываются линейными моделями. Поэтому, целесообразным будет применять в имитационных моделях нелинейные звенья.

Технологические установки состоят, как правило, из типового оборудования. Это является предпосылкой для блочного построения имитационной модели, при котором программа имитации состоит из набора связанных экземпляров типовых функциональных блоков. Таким образом, с каждым новым объектом разработчик постепенно накапливает библиотеку собственных функциональных блоков, которые могут быть использованы для моделирования последующих объектов.

Учитывая использования для написания таких блоков языков МЭК 61131-3, они являются портативными, то есть могут быть использованы в других средах программирования ПЛК. Например, в качестве таких функциональных блоков могут быть модели теплообменников, емкостей смешивания, дозаторов, centrifуг, насосов и т.д. Кроме того, в системе управления используются различные датчики, исполнительные механизмы и регулирующие органы, для моделирования которых тоже необходимо разработать функциональные блоки. Ниже приводятся подходы при создании таких блоков, а также пример их реализации.

Моделирование переменных состояния объекта

Для динамического моделирования переменных состояний технологического объекта предлагается использовать системы дифференциальных уравнений. При этом, численное решение системы обыкновенных дифференциальных уравнений можно получить, используя конечные разностные схемы, например, явный метод Эйлера [2]:

$$y_{n+1} = y_n + \Delta t \cdot f(t_n, y_n), \quad (1)$$

где y_{n+1} , y_n — переменные состояний объекта управления на текущем и предыдущем шаге; Δt — время шага моделирования; $f()$ — вектор-функция процесса; t_n — время на текущем шаге моделирования.

Явный метод Эйлера имеет существенные недостатки, в частности, малую точность и систематическое накопление ошибок. Однако они компенсируются небольшим временем расчета и невысокими требованиями к точности имитационной модели. Особенностью реализации приведенных формул в ПЛК является совпадение циклов переисчисления с вызовом задачи, то есть отсутствуют итерационные расчеты на каждом цикле, что было бы невозможно при неявных методах.

Например, для разработки блока имитации теплообменника, параметрическая схема которого приведена на рис. 1, можно воспользоваться системой дифференциальных уравнений:

$$\frac{dT(t)}{dt} = \frac{F(t)}{V} \cdot (T_{in}(t) - T(t)) - \frac{k \cdot S}{\rho \cdot c \cdot V} \cdot (T_a(t) - T(t)), \quad (2)$$

$$\frac{dT_a(t)}{dt} = \frac{F_a(t)}{V_a} \cdot (T_{ain}(t) - T_a(t)) - \frac{k \cdot S}{\rho_a \cdot c_a \cdot V_a} \cdot (T_a(t) - T(t)); \quad (3)$$

с начальными условиями:

$$\begin{aligned} T(t_0) &= T_0, \\ T_a(t_0) &= T_{a0}; \end{aligned} \quad (4)$$

где V , V_a — объем в теплообменнике рабочей жидкости и теплоносителя соответственно, m^3 ; ρ , ρ_a — плотность рабочей жидкости и теплоносителя соответственно, kg/m^3 ; c , c_a — теплоемкость рабочей жидкости и теплоносителя соответственно, $kJ/(kg \cdot K)$; k — коэффициент передачи через поверхность теплообмена, $W/(m^2 \cdot ^\circ C)$; S — поверхность теплообмена, m^2 ; t — время моделирования, s .

В математической модели, описанной уравнениями (2)–(4), сделан ряд допущений. Однако ее можно использовать в качестве первого приближения, если не требуется отладка сложных контуров регулирования (например, каскадных схем).

На рис. 2 показана структура и программа пользовательского функционального блока, имитирующего работу теплообменника. Параметры модели (2)–(3) и начальные значения (4) задаются как глобальные переменные блока. Входные значения модели являются входными переменными блока, а координаты состояния — выходными. В случае отсутствия в среде разработки глобальных параметров функциональных блоков как таковых, можно использовать входные.

На примере этого типа функционального блока, прокомментируем использование общих подходов, описанных выше. Вход INIT используется для инициализации модели. То есть в любой момент времени отладчик разработчик может перевести модель в начальное состояние. Глобальный параметр d_t вызывает алгоритм блока, с какой периодичностью он вызывается. При таком подходе сам блок должен вызываться периодически, для чего можно использовать неявный вход EN (использование

EN/ENO описан в стандарте МЭК 61131-3). Задав параметр d_t равным периодичности вызова, наладчик выставляет имитационной модели режим реального времени, больше периодичности вызова — ускоренный режим, меньше — замедленный.

Моделирование возмущений и шумов измерения

Моделирование возмущений играет значительную роль при проверке работы программы ПЛК. В зави-

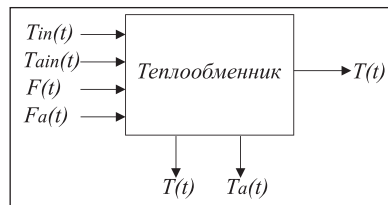


Рис. 1. Параметрическая схема теплообменника (T_{in} , T — температура рабочей жидкости на входе и выходе теплообменника соответственно, $^\circ C$; T_{ain} , T_a — температура теплоносителя на входе и выходе теплообменника соответственно, $^\circ C$; F , F_a — расход рабочей жидкости и теплоносителя, соответственно, kg/c)

Name	no.	Type	Value	Comment
HeatExch		<DFB>		теплообменник
INIT	1	BOOL		команда инициализации (начальные условия)
F	2	REAL	0	расход жидкости, м3/с
Fa	3	REAL	0	расход теплоносителя, м3/с
Tin	5	REAL	24	температура жидкости на входе, град С
Tain	6	REAL	80	температура теплоносителя на входе, м3/с
T	3	REAL		температура жидкости, град С
Ta	4	REAL		температура теплоносителя, град С
d_t		REAL	0.1	периодичность вызова, с
T0		REAL	24	нач. температура жидкости, м3
Ta0		REAL	80	нач. температура теплоносителя, град С
S		REAL	45	площадь теплообмена, м2
C		REAL	3.9	теплоемкость жидкости, кДж/(кг*К)
Ca		REAL	4.19	теплоемкость теплоносителя, кДж/(кг*К)
k		REAL	2	коэффициент теплопередачи, кВт/(м2*градС)
rho		REAL	1027	плотность жидкости, кг/м3
rhoa		REAL	1000	плотность теплоносителя, кг/м3
V		REAL	0.5	объем жидкости, м3
Va		REAL	0.2	объем теплоносителя, м3
<pre> IF Init THEN T:=T0; Ta:=Ta0; END_IF; T:=T+(d_t/V)*(F*(Tin-T)+k*S*(Ta-T)/(rho*C)); Ta:=Ta+(d_t/Va)*(Fa*(Tain-Ta)-k*S*(Ta-T)/(rhoa*Ca)); </pre>				

Рис. 2. Реализация пользовательского типа функционального блока для имитации пластинчатого теплообменника

симости от решаемой задачи, при моделировании возмущений и шумов измерения можно использовать различные закономерности изменения сигналов: ступенчатый, импульсный, случайный, синусоидальный, кусочно-линейный и т.д. Тут ограничимся только моделированием случайной составляющей сигнала, путем генерирования псевдослучайных чисел. Для этого, например, можно использовать генератор случайных чисел в диапазоне $[0, M]$, описываемый линейным конгруэнтным методом:

$$r_{i+1} = \text{mod}(k \cdot r_i + b, M), \quad (5)$$

где r_i — значение случайной величины на новом цикле пересчета; M, k, b — коэффициенты, r_0 — начальное значение, MOD — остаток деления.

На рис. 3 показан пример реализации функционального блока для генерации случайных чисел с использованием формулы (5). Коэффициенты M, k, b и начальное значение задаются как глобальные переменные блока, внутренняя переменная r_i предназначена для запоминания предыдущего значения.

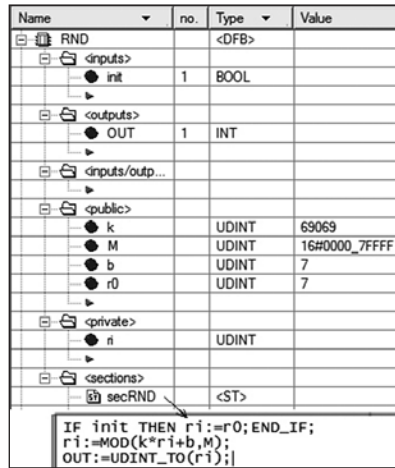


Рис. 3. Реализация пользовательского типа функционального блока для генерации случайных чисел

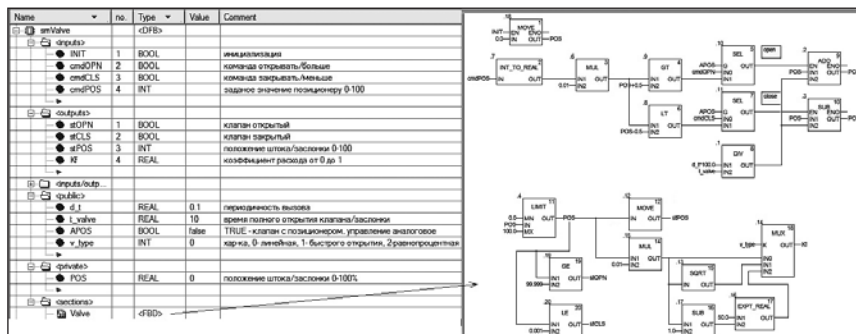


Рис. 4. Реализация пользовательского типа функционального блока для имитации исполнительного механизма на языке FBD

Моделирование исполнительных механизмов и регулирующих органов

Тип исполнительного механизма (ИМ) и регулирующего органа (РО) вносит свои коррективы в алгоритм управления, поэтому его поведение также необходимо учитывать при проверке работы программы. Для запорно-регулирующей трубопроводной арматуры с электро- или пневмоприводом, можно разработать универсальный функциональный блок, имитирующий всю сборку. Такой блок должен учитывать следующие особенности:

- время полного хода (открытия) РО с заданным типом ИМ при рабо-

чих условиях (давление пневмопитания т.п.);

- управление двумя входными сигналами типа BOOL: открыть (больше) и закрыть (меньше); в случае управления одним сигналом (ИМ с возвратом) на другой вход должна подаваться его инвертированная копия;

- возможность аналогового управления с позиционером, то есть блок должен иметь входной сигнал в диапазоне 0...100%, задающий позицию РО;

- выходы для датчиков: положения штока/заслонки (0...100%), сигнализаторы положения "открыт" и "закрыт";

- тип характеристики регулирующего органа: линейный, равнопроцентный, быстрого открытия.

На рис. 4, 5 показаны примеры реализации такого функционального блока. Параметр t_valve определяет время полного хода РО для выбранного ИМ, $APOS$ — определяет тип управления (аналоговое или дискретное), v_type — указывает на тип характеристики РО. Входы $cmdOPN$ и $cmdCLS$ предназначены для сигналов управления соответственно типа "открыть"

и "закрыть" при дискретном управлении (при $APOS=0$), а $cmdPOS$ (0-100%) — для аналогового управления (при $APOS=1$). Выход блока Kf — является коэффициентом расхода в диапазоне 0...1. Выходы $stOPN$ и $stCLS$ имитируют работу конечных датчиков положений, а $stPOS$ — текущей позиции РО. Таким образом, один и тот же тип функционального блока может быть использован для разных типов ИМ и трубопроводной арматуры.

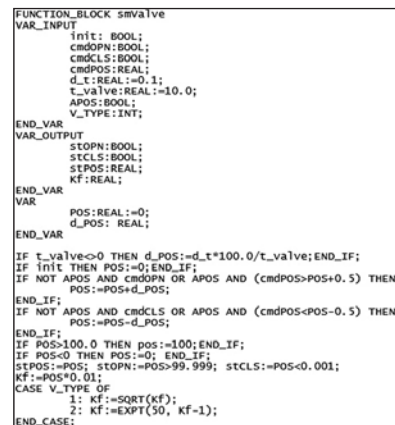


Рис. 5. Реализация пользовательского типа функционального блока для имитации исполнительного механизма на языке ST

Апробация

Приемы, описанные в данной статье, апробированы при предварительной наладке программ ПЛК для процессов молочного производства. В частности, моделировался процесс последовательного набора танков приготовления цельномолочных продуктов с пастеризационно-охладительной установки, и переходом между танками по показателям счетчиков и датчиков проводимости. Предварительная наладка на имитационной модели дала возможность отловить большинство ошибок в программе управления до этапа внедрения, сократить время внедрения

и пуско-наладки, а также избежать потерь продукта на этапе пуско-наладочных работ.

Авторами также разработан ряд имитационных моделей, используемых в лабораторных работах в качестве объекта управления при изучении дисциплин: "Контроллеры и их программное обеспечение", "Программно-технические комплексы и промышленные контроллеры".

Следует также отметить, что имитационное моделирование также используется студентами при курсовом проектировании в этих дисциплинах.

Для демонстрации описанных подходов, ниже приводится фрагмент учебного примера для лабораторной работы по изучению библиотеки регулирования UNITY_PRO (Schneider Electric). На рис. 6 показан операторский экран для контура стабилизации температуры на выходе теплообменника (ТТ101), путем изменения расхода теплоносителя клапаном с ИМ типа МЭО (TV101). ИМ управляется двумя сигналами TV101_OPN (больше) и TV101_CLS (меньше). Для контроля положения РО используются контакты ИМ конечного положения (GSH101 и GSL101).

Задача лабораторной работы сводится к созданию программной части контура регулирования в различных операционных режимах и проверке ее работоспособности. Один из вариантов реализации программной части контура показан на рис. 7. Регулятор реализован в виде фрагмента FBD-программы, состоящей из взаимосвязанных функциональных блоков: TC101 (PI_B) — реализующего ПИ закон регулирования, и TS101 (SERVO) — для управления МЭО. На рис. 8 не показаны фрагменты масштабирования реальных входов/выходов, так как эту часть программы можно отладить на самом объекте при проверке каналов измерения и управления.

Для проверки работоспособности программной части контура составлена имитационная модель в виде программы на FBD (рис. 8). Все дополнительные переменные и экземпляры функциональных блоков имеют префикс *sm*, что позволяет удалить их перед использованием реального объекта. Блок *smTV101* имитирует работу исполнительного механизма, *smHeat* — температуры теплообменника. Таким образом, контур из рис. 7 замыкается через имитатор объекта управления, показанный на рис. 8.

Луцкая Наталья Николаевна — канд. техн. наук, доцент кафедры «Автоматизация процессов управления»,
Пупен Александр Николаевич — канд. техн. наук, доцент, *Швед Сергей Николаевич* — ассистент кафедры
«Интегрированные АСУ» Национального университета пищевой промышленности.

Контактный телефон 38(044) 287-97-90

E-mail: lutskaya@yandex.ru pupena_san@ukr.net

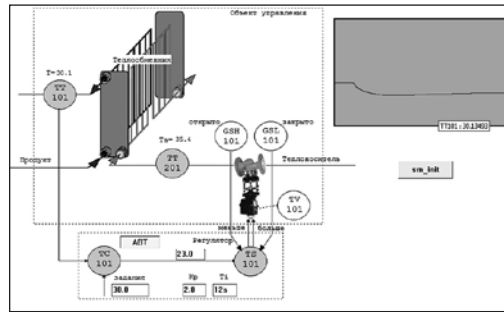


Рис. 6. Пример системы управления на операторском экране (учебный вариант)

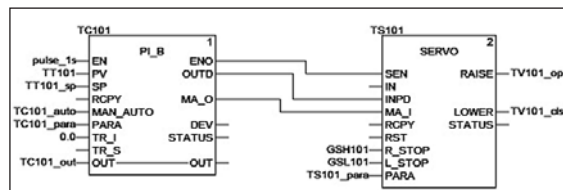


Рис. 7. Фрагмент программы контура регулирования

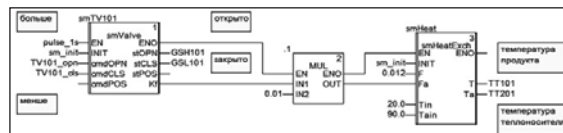


Рис. 8. Программа имитации пластинчатого теплообменника

Выводы
Идея использования ресурсов ПЛК для имитационного моделирования ТП не является новой, однако известные практические приемы в лучшем случае сводятся к использованию блоков типовых линейных звеньев [3]. Приведенные выше рекомендации, прошедшие апробацию в производственных и учебных проектах, имеют практическую

ценность. В наработке имеется библиотека функциональных блоков типовых объектов, описание которых не вписывается в рамки данной статьи. Постоянно ведется работа над расширением библиотеки. Использование языков МЭК 61131-3 делает эти блоки универсальными (вендор-независимыми) и понятными всем разработчикам программного обеспечения ПЛК.

Однако следует выделить границы использования этих имитационных моделей. Использование явного метода Эйлера не рекомендуется для точного моделирования, а, значит, такая модель не может быть

применена в контуре регулирования в качестве эталонной модели технологического процесса. Перед созданием программы имитации следует оценить время на её разработку и верификацию, и соотнести общее время, затраченное на тестирование с моделью и без неё. Для типовых проектов наличие модели оправдано, так как ее адаптация к конкретному объекту не занимает много времени.

В качестве весомых аргументов в пользу разработки имитационной модели является уменьшение рисков производственных затрат а также сокращение времени на этапе пусконаладочных работ. Также есть смысл разработки модели для верификации проектов SCADA/HMI на разных этапах их жизненного цикла.

Список литературы

1. *Зубин В.Е.* Итерационная разработка управляющих алгоритмов на основе имитационного моделирования объектов управления // Автоматизация в промышленности. 2010. № 11.
2. *Гартман Т.Н., Клушин Д.В.* Основы компьютерного моделирования химико-технологических процессов / Уч. пособие для вузов. М.: ИКЦ «Академкнига». 2006.
3. *Деменков Н.П.* Языки программирования промышленных контроллеров: Учебное пособие / Под. ред. К.А. Пупкова. — М.: Изд-во МГТУ им. Н.Э. Баумана, 2004. — 172 с.: ил.