

**МЕТОДИКА СОЗДАНИЯ ВИРТУАЛЬНЫХ ЛАБОРАТОРНЫХ СТЕНДОВ
В ОБЛАСТИ ПРОГРАММИРОВАНИЯ УПРАВЛЯЮЩИХ СИСТЕМ****А. В. Носенко, В. Е. Зюбин (Новосибирск)****Введение**

При изучении языков программирования, ориентированных на создание управляющих алгоритмов, теоретические знания должны подкрепляться лабораторно-практическими занятиями по решению типовых задач из области промышленной автоматизации на стендах-тренажерах. Однако такие стенды громоздки, требуют серьезных финансовых затрат на создание и поддержку их в работоспособном состоянии, а ошибки, допускаемые студентами-практикантами во время работы, часто приводят к выходу стендов из строя.

В этих условиях наиболее перспективный способ организации практических занятий по курсам промышленной автоматизации должен быть основан на использовании программных имитаторов – виртуальных лабораторных стендов (ВЛС). В Институте автоматизации и электрометрии СО РАН было предложено создавать ВЛС на базе пакета LabVIEW и языка Рефлекс. Однако на настоящее время создано ограниченное число ВЛС, а процесс создания ВЛС унифицирован частично.

В этих условиях унификация всего процесса разработки ВЛС позволила бы просто и быстро создать набор ВЛС, обеспечить его пополнение и сопровождение.

В статье приводится концепция виртуального лабораторного стенда, формулируются требования к методике создания и работе со стендом, приводятся несколько выбранных сценариев, моделирующих типовые ситуации, возникающие при решении задач промышленной автоматизации, и рассматривается реализация этих сценариев в ВЛС.

Концепция ВЛС

Концепция ВЛС основана на идее подмены реального объекта управления или его физической имитации на виртуальный объект управления (ВОУ), реализуемый исключительно программно. ВЛС представляет собой законченное приложение, включающее визуальную и поведенческую имитацию технологического объекта, а также средства создания управляющего алгоритма с возможностью запуска его на исполнение.

В качестве базовой среды программирования для создания ВЛС используется пакет LabVIEW [1], популярный и широко используемый в обучающем процессе, в частности для создания обучающих лабораторных стендов.

В качестве языка программирования использован язык Рефлекс (также известный как «Си с процессами») [2], ориентированный на описание алгоритмов управления и реализующий процессо-ориентированный стиль программирования.

Исполнение управляющих алгоритмов реализовано на основе интерпретатора языка Python [3]. Совместно с возможностью среды LabVIEW создавать исполняемые модули это позволяет оформлять ВЛС в виде законченного приложения, тиражировать создаваемые стенды и исполнять их автономно от среды разработки.

При предлагаемом подходе алгоритм на языке Рефлекс, создаваемый обучаемым, преобразуется в текст на языке Python, который затем запускается на исполнение интерпретатором языка Python и взаимодействует с имитатором объекта (рис. 1).

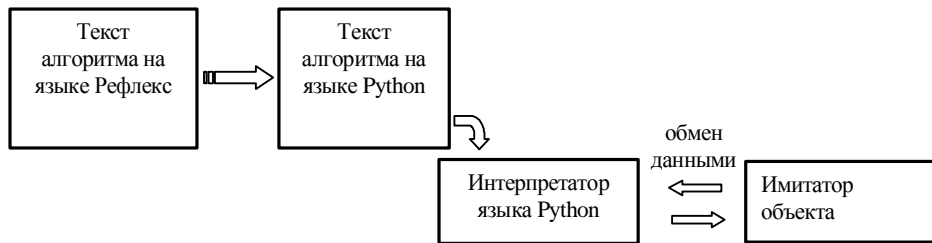


Рис. 1. Схема получения управляющего алгоритма для взаимодействия с имитатором объекта

Эквивалентная запись алгоритма на языке Python получается транслятором, созданным на базе существующего транслятора языка Рефлекс в язык Си. Интерпретатор языка Python интегрирован в среду LabVIEW через механизм ActiveX. При реализации взаимодействия управляющего алгоритма и имитатора объекта использована штатная программная модель языка Рефлекс – модель гиперпроцесса [4].

Работа студента-практиканта с виртуальным стендом в рамках предлагаемой концепции происходит следующим образом (рис. 2). Сначала создается текстовое описание алгоритма на языке Рефлекс, затем запускается трансляция текста на языке Python. При обнаружении синтаксических ошибок во время трансляции происходит возврат к редактированию. После успешной трансляции алгоритм управления, представленный уже в тексте языка Python, запускается на исполнение и тестируется на корректность путем создания различных ситуаций, предусмотренных сценарием задачи. В случае обнаружения некорректного поведения алгоритма, происходит возврат к начальному этапу редактирования текста алгоритма управления.

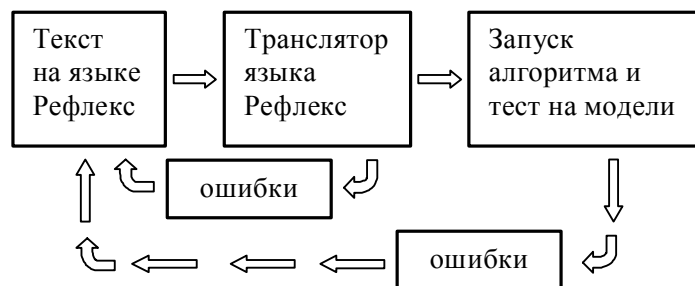


Рис. 2. Схема работы по созданию и отладке управляющего алгоритма

Выбор типовых сценариев для ВЛС

При отборе сценариев была просмотрена литература и статьи по тематике программирования информационно-управляющих, дискретных, реагирующих (reactive) систем и систем так называемого реального времени, из которых были выбраны сценарии, предусматривающие типовые ситуации из области автоматизации: обработку логических сигналов, регулирование, работу с временными интервалами (паузами, таймаутами, задержками) и параллелизм. Всего было выделено пять сценариев: электросу-

шилка для рук, регулирование потоков транспорта на перекрёстке, линия розлива бутылок, линия закупорки бутылок, автопогрузка гравия [5].

Методика реализации ВЛС

В результате анализа последовательности действий, выполняемых при создании виртуального лабораторного стенда, было установлено, что оптимальным является следующий порядок.

1. Выбор сценария для ВЛС.

Выбирается объект управления, для реализации в виде ВОУ.

2. Формулировка условия задачи.

Для каждого из отобранных сценариев формулируется условие задачи, которую будет решать обучаемый.

3. Определение входных/выходных сигналов ВОУ.

Для ВОУ определяются входные и выходные сигналы (датчики, исполнительные устройства, входные и выходные сигналы, через которые алгоритм управления воздействует на ВОУ).

4. Определение управляющих воздействий и визуальных эффектов, описание алгоритмического фильтра.

Определяются управляющие воздействия, которые может осуществить пользователь на ВОУ, и условия, при которых воздействия можно осуществить. Определяется внешний вид объекта управления, какие визуальные эффекты будут использованы при его реализации.

5. Абстрактный синтез. Определение алгоритма поведения ВОУ.

Определяется внутренняя структура виртуального объекта управления. Описывается алгоритм его работы: очерёдность визуальных эффектов, процессы, происходящие при оказании управляющего воздействия на объект управления. Определяется способ реализации датчиков.

6. Структурный синтез. Определение способа реализации ВОУ.

В этом процессе определяются структуры данных, функции для реализации визуальных эффектов, вспомогательные функции, которые потребуются для реализации ВОУ.

7. Реализация визуальных эффектов и алгоритмов.

После выполнения предыдущих шагов происходит реализация визуальных эффектов и алгоритмов. Отладка ВОУ с ручным заданием управляющих воздействий и анализ их выполнения. Создаются нестандартные ситуации, анализируется работа с различными алгоритмами управления.

8. Интеграция в ВЛС ВЛС.

Производится интеграция ВОУ в ВЛС, оформление стандартного пользовательского интерфейса.

Как видно из описанного порядка действий, методику создания виртуального лабораторного стенда удалось свести к созданию виртуального объекта управления.

Реализация ВЛС

Методика создания ВЛС была опробована при разработке ВЛС «Электросушилка» и ВЛС «Регулируемый перекресток». Имитационные модели создавались штатными средствами LabVIEW для работы с изображениями путем наложения графических примитивов. Срабатывание датчиков определялось путем анализа цвета пикселей в заданных координатах сцены. Анимация обеспечивалась циклическим перебором изображений в зависимости от условий.

Пользовательский интерфейс приложения обеспечивает возможность ввода алгоритма (закладка «Алгоритм») и просмотра результатов трансляции (закладки «Ошибки» и «Py-Code»). Краткое описание условия задачи доступно через закладку «Помощь». Выход из приложения осуществляется через закладку «КОНЕЦ РАБОТЫ».

В процессе реализации было предложено выделять в ВЛС входные сигналы диагностики состояния, выходные сигналы управления моделью объекта (формируемые в зависимости от режима программно или вручную), сигналы оперативного воздействия, формируемые только пользователем, которые позволяют отработать поведение алгоритма в различных ситуациях и используемые визуальные эффекты. Это позволяет унифицировать процесс создания ВЛС. Таким образом, в каждом стенде реализованы два режима управления: ручной и автоматический. В ручном режиме имеется возможность непосредственного управления. В автоматическом режиме возможность ручного управления блокируется, а команды управления формируются алгоритмом, созданным студентом-практикантом по описанной выше схеме. В обоих режимах имеется возможность произвести внешнее воздействие на систему. В специальном окне отображаются числовые идентификаторы текущих функций-состояний процессов.

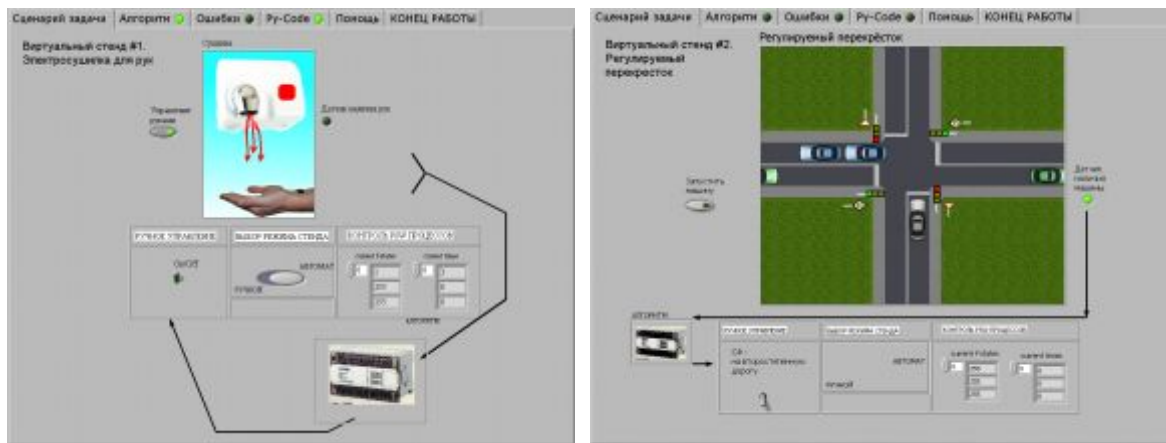


Рис. 3. Виртуальные лабораторные стенды «Электросушилка» и «Регулируемый перекрёсток»

Были реализованы виртуальные лабораторные стенды для задач «Электросушилка» и «Регулируемый перекрёсток» (рис. 3).

Заключение

В работе исследовалась методика создания ВЛС, ориентированных на практическое освоение основ программирования задач промышленной автоматизации. Предлагаемый подход может быть с успехом применён в процессе обучения и использован компаниями, специализирующихся на создании средств разработки управляющего программного обеспечения: в качестве демо-версий может поставляться замкнутое приложение с включенным в него ВОУ. Получая такую демо-версию, потенциальный клиент в состоянии сформировать более полное представление о продукте, попробовав его возможности на задаче автоматизации ВОУ.

Для достижения цели были собраны задачи по созданию управляющих алгоритмов из открытых источников. Из этого набора сформирован пакет типовых сценариев, предполагающих параллельную (независимую) обработку дискретных сигналов, регу-

лирование физических параметров, обработку событий, работу с временными интервалами.

В настоящее время создается библиотека типовых модулей отображения визуальных эффектов (заполнение объема жидкостью, сыпучими материалами, движение, наложение графических примитивов и др.)

В процессе исследования был сделан вывод, что концепция ВЛС применима при создании управляющих алгоритмов, поскольку позволяет использовать в процессе разработки test-driven технологию, которая не только облегчает разработку сложных алгоритмов, но и повышает качество создаваемых программ. Особую важность подобная возможность может иметь при автоматизации критических производств.

Литература

1. **Бутырин П. А., Васьковская 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. 264 с.
2. **Зюбин В. Е.** «Си с процессами»: язык программирования логических контроллеров // Мехатроника. 2006. № 12. С. 31–35.
3. **Сузи Р. А.** Язык программирования Python. Изд-ва: Интернет-университет информационных технологий, Бином. Лаборатория знаний, 2006, 328 с.
4. **Зюбин В. Е.** Программирование информационно-управляющих систем на основе конечных автоматов: Учеб.-метод. пособие. Новосибирск: Новосиб. гос. ун-т, 2006. 96 с. [<http://reflex-language.narod.ru/index.html>].
5. **Зюбин В. Е., Носенко А. В.** Создание набора виртуальных лабораторных стендов для обучения программированию управляющих систем // Proceedings of 7th International Conference “Perspectives of System Informatics” 15–19 June 2009, Novosibirsk, Russia. P. 51–56.