Энергетика. Автоматика

УДК 681.518, 519.876.2, 519.876.5

А.Н. Рыбалев

ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ АСУ ТП ПРИ ПРОЕКТИРОВАНИИ И В УЧЕБНОМ ПРОЦЕССЕ. ЧАСТЬ 2

Во второй части статьи сделан обзор типовых решений по имитационному моделированию ключевых компонентов АСУ ТП: систем регулирования и программно-логического управления.

Ключевые слова: Simulink-модель, виртуальный контроллер, SCADA-система.

IMITATIVE MODELING OF APCS IN DESIGNING AND IN THE EDUCATIONAL PROCESS. PART 2

In the second part of the article, there is an overview of typical solutions for simulating the key components of the process control system: regulation and software-logical control systems.

Key words: Simulink-model, virtual controller, SCADA-system.

Введение

В первой части статьи обоснована необходимость имитационного моделирования АСУ ТП как на практике, так и в учебном процессе по направлению подготовки 15.03.04 «Автоматизация технологических процессов и производств». Предложен вариант построения имитационной модели системы управления на базе модели объекта управления в Matlab Simulink, виртуального контроллера CoDeSys PLC WinNT и SCADA-системы Trace Mode. Обмен между программными компонентами осуществляется по технологии OPC. Кратко рассмотрены разработки, сделанные на кафедре автоматизации производственных процессов и электротехники АмГУ. Здесь будут рассмотрены типовые решения (прототипы подсистем регулирования и программно-логического управления), которые могут быть положены в основу «больших» моделей АСУ ТП.

Системы автоматического регулирования

Системы с типовыми законами регулирования

На первом этапе был создан прототип системы автоматического регулирования одного технологического параметра с непрерывным управлением простейшим динамическим объектом. В Simulink-модели (рис. 1) сравнивалось поведение системы с «внутренним» ПИ-регулятором, представленным экземпляром библиотечного блока PID Controller, и системы с «внешним» регулятором, реализованном в виртуальном контроллере PLC WinNT (также на базе стандартного функционального блока CoDeSys), связь с которым осуществлялась по OPC.

Результаты моделирования показали достаточно хорошее совпадение откликов, что позволило продолжить работу по разработке модели подсистемы АСУ ТП.

Сразу же следует отметить, что сам контур автоматического регулирования, являясь, безусловно, важнейшей частью этой подсистемы, в плане реализации никаких трудностей не представляет.

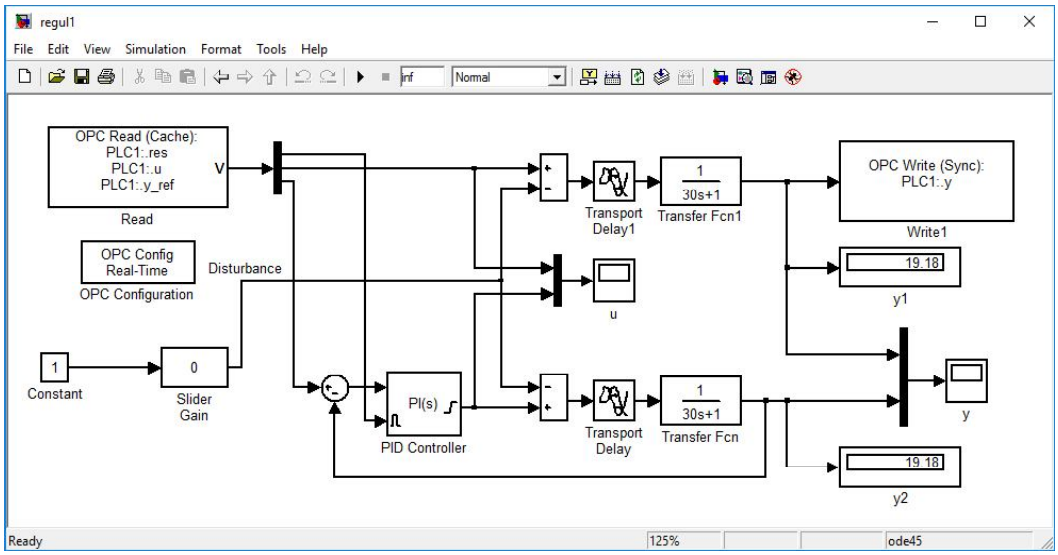

Рис. 1. Simulink-модель.

Гораздо сложнее решить многочисленные «вспомогательные» задачи, которые возлагаются на подсистему: введение задания, переключение режимов управления, сигнализация, архивирование и т.д.

В нашем прототипе предусмотрена возможность формирования управляющего сигнала тремя «действующими лицами»: оператором панели управления на щите (режим ручного управления), программой автоматического управления контроллера (режим автоматического управления) и оператором станции – пользователем SCADA-системы (дистанционное управления). Панель управления имитируется экраном визуализации CoDeSys (рис. 2), который может быть запущен и вне самой среды CoDeSys под управлением программы CoDeSys HMI. Экран SCADA-системы показан на рис. 3.

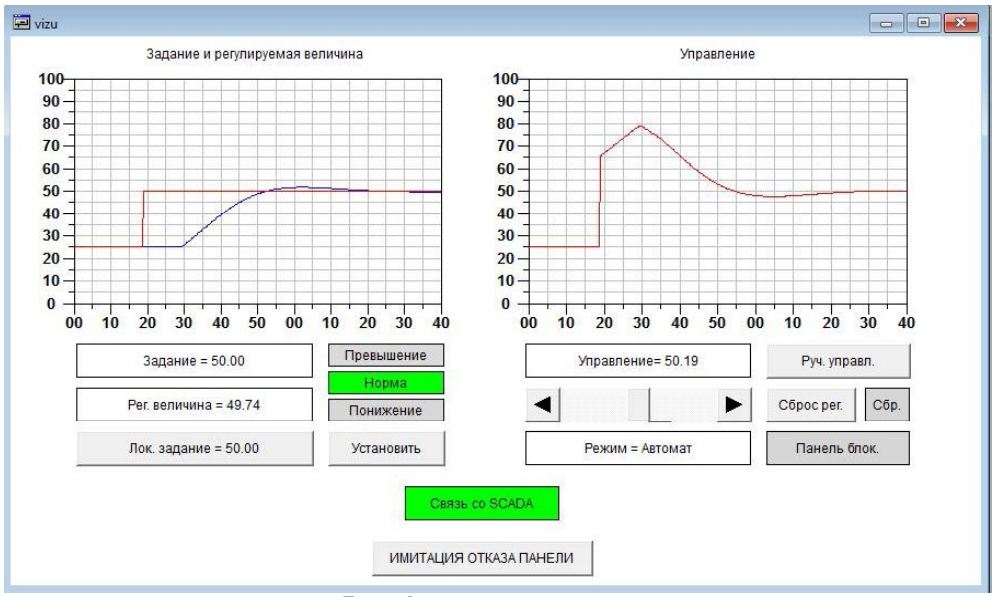

Рис. 2. Панель оператора.

Разработанный программный комплекс включает 1) Simulink-модель, 2) три программы ПЛК (одна из них эмулирует программное обеспечение панели оператора), 3) визуализацию CoDeSys для панели, 4) экран, архивы и три программы SCADA. Комплекс решает основные задачи подсистемы АСУ ТП:

автоматическое регулирование технологического параметра с возможностью сброса регулятора командами операторской панели и SCADA;

ввод задания с панели оператора и экрана SCADA;

контроль связи между ПЛК, операторской панелью и SCADA. «Обрыв связи» можно имитировать с помощью специальной «фиктивной» кнопки на панели и отключением профайлера Trace Mode соответственно;

переключение режимов управления (ручной/ автоматический/ дистанционный). При наличии связи со всеми компонентами приоритет отдан дистанционному режиму: в любой момент панель оператора может быть заблокирована с экрана SCADA. Однако при потере связи с операторской станцией панель автоматически разблокируется;

графическое и цифровое отображение значений сигналов задания, управления и регулируемой величины на панели оператора и экране SCADA;

оповещение оператора SCADA о таких событиях как выход регулируемой величины за пределы разрешенного диапазона и возвращение ее обратно, опасных отклонениях, изменениях режима и т.п. Все эти события фиксируются также в специальном текстовом файле – «архиве тревог»;

запись в СПАД (структурированный промышленный архив данных, или англ. SIAD, Structured Industry Archive of Data) значений сигналов задания, управления и регулируемой величины, просмотр архива и его импорт в документы форматов TXT, XML и HTML.

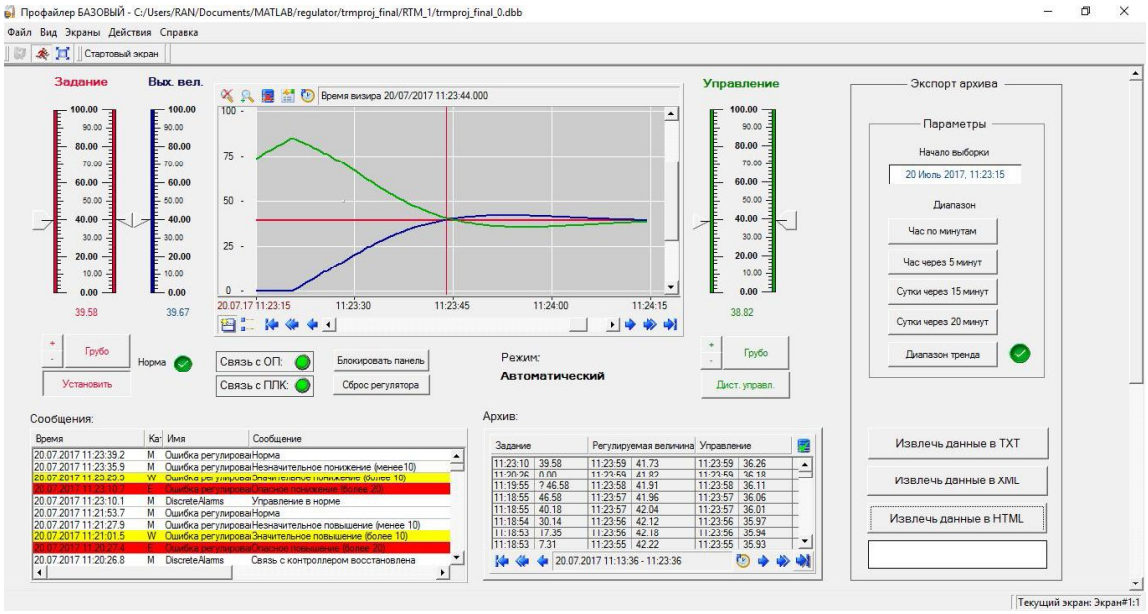

Рис. 3. Экран SCADA-системы.

На следующем этапе разработки созданы имитационные модели систем регулирования, в которых управляющее воздействие формируется исполнительным механизмом постоянной скорости. В промышленности такие системы составляют большинство, и в них в качестве исполнительного механизма выступает электропривод дросселирующего органа: задвижки, заслонки, клапаны и т.д.

Существуют два отличающихся друг от друга варианта построения таких систем: без измерения положения механизма и с измерением.

В первом случае исполнительный механизм интегрирующего типа входит в состав регулятора (возможен вариант внесения механизма в объект, но такой вариант не рассматривался), поэтому для реализации ПИ-закона регулирования программа контроллера должна проводить вычисление по ПД-закону. Появляется также необходимость в широтно-импульсном модуляторе, преобразующем непрерывный сигнал, вырабатываемый ПД-алгоритмом, в импульсы управления приводом. Основная сложность именно в программной реализации ШИМ: исполнительный механизм не в состоянии «отработать» как короткие импульсы, так и короткие паузы.

Рассмотрены идеи четырех алгоритмов формирования импульсов управления исполнительным механизмом постоянной скорости:

1) «классический» ШИМ с отбрасыванием коротких импульсов;

2) «классический» ШИМ с переносом коротких импульсов;

3) ШИМ с переменным периодом модуляции. Такой ШИМ при скважности меньшей 50% формирует предельно короткие импульсы, а время паузы устанавливает в соответствии со скважностью. При скважности большей 50% предельно короткой будет, наоборот, пауза, а время импульса рассчитывается;

4) «следящий» ШИМ. Основная идея состоит в использовании программной модели исполнительного механизма, которая, естественно, имеет обратную связь по положению. Зная фактическую скорость перемещения рабочего органа и, следовательно, постоянную времени интегратора, описывающего привод, мы можем рассчитать *приращение положения*, которое имело бы место, если бы на вход интегратора действительно подавался непрерывный сигнал. Далее следует включать привод так, чтобы реализовать это приращение с приемлемой точностью и с ограничением на минимальное время включения.

Все четыре алгоритма сначала опробованы на Simulink-моделях, причем во всех случаях потребовалось основную часть алгоритма переносить в код вызываемой из диаграммы функции Matlab. После апробации все алгоритмы были реализованы как программы и функциональные блоки ПЛК. Небольшим изменениям подверглись другие части программного комплекса: экран операторской панели и SCADA-система.

В случае, когда исполнительный механизм оснащен датчиком положения, появляется возможность построения внутреннего контура регулирования положения с релейным регулятором (рис. 4).

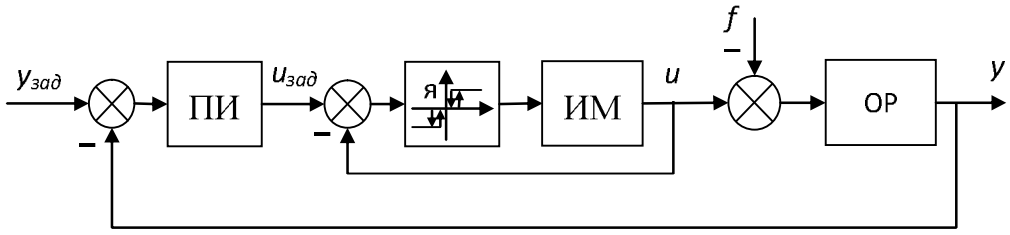

Рис. 4. Структура системы.

Данная система также была программно реализована на всех уровнях.

Системы с нетиповыми законами регулирования

Разработаны прототипы систем с нетиповыми линейными законами регулирования, заданными дробно-рациональными передаточными функциями, а также законами, в состав которых входит транспортное запаздывание.

Реализация регулятора, заданного передаточной функцией, сводится к численному интегрированию соответствующих ей уравнений в пространстве состояний с помощью стандартных функциональных блоков интегрирования, экземпляры которых задействованы в программах на языках ST, FBD, SFC. Следует отметить, что в CoDeSys для правильной работы блоков интегрирования вызывающая их программа должна быть циклически выполняемой, причем период цикла вводится как настроечный параметр интегратора.

Результаты сравнения поведения комбинированной системы с виртуальным контроллером и модели системы регулирования в Simulink показывают, что алгоритмы реализованы правильно. Более подробно результаты части работ изложены в [1].

Программная реализация чистого транспортного запаздывания представляет собой нетривиальную задачу, так как в библиотеках CoDeSys готовых решений нет. Были рассмотрены два подхода: с использованием аппроксимации Паде второго и четвертого порядка и с использованием циклического буфера для запоминания предыдущих отчетов запаздывающего сигнала.

Если задействуется аппроксимация Паде, программированию подлежит дробно-рациональная передаточная функция и задача решается с помощью интегрирования дифференциальных уравнений.

Циклический буфер может быть организован по-разному, в зависимости от того, как он будет читаться и обновляться (после чтения из буфера новое значение всегда заменяет собой старое, если речь не идет о начальном заполнении). Рассмотрено несколько вариантов.

Чтение/обновление буфера по таймеру предполагает периодический запуск таймера на величину шага по времени. Проблема состоит в том, что промежутки времени, необходимые для перезапуска таймера, накапливаются и увеличивают общее запаздывание. Проблему можно было бы решить уменьшением уставки таймера по сравнению с шагом по времени, но определить необходимую разницу для сколько бы то ни было сложных систем практически невозможно.

Чтение/обновление буфера по времени требует наличия в системе часов реального времени и функции для работы с ними. Кроме того, необходима фиксация моментов времени поступления накопленных значений входной величины, т.е, по сути, необходимо задействовать второй массив.

Чтение/обновление буфера с использованием циклически вызываемой задачи предполагает, что алгоритм будет выполняться периодически, через строго определенное время (фактически речь идет о работе по прерыванию).

Все указанные выше методы были программно реализованы для виртуального контроллера и апробированы в комбинированной системе регулирования с предиктором Смита для объекта с большим транспортным запаздыванием. Все варианты оказались вполне работоспособными, но наилучшее приближение к поведению эталона (системы, полностью реализованной в Simulink) показала система с циклическим буфером, обновлявшимся по времени. Однако с практической точки зрения лучше обновлять буфер с использованием циклически вызываемой задачи, поскольку по сравнению с первым вариантом качество процесса управления снижается незначительно, а потребление ресурсов (объем памяти и время выполнения кода) – достаточно сильно. Часть результатов работы приведена в [2].

Системы программно-логического управления

В отличие от систем регулирования для систем программно-логического управления сложно или даже вообще невозможно разработать некие прототипы, поскольку такие системы чрезвычайно разнообразны. Однако можно на примере какой-либо системы показать методы решения типовых задач (таких, например, как блокировка) и в целом продемонстрировать общий подход к программной реализации системы. На наш взгляд, этот общий подход должен базироваться на представлении системы в виде конечного автомата, который может находиться в различных состояниях, переходить по некоторым условиям из одного состояния в другое и в каждом состоянии демонстрировать определенную логику преобразования входных сигналов в выходные. Пусть при этом теория конечных автоматов и не задействуется в полном объеме (например, автомат не оптимизирован по числу состояний), это не так важно, ведь речь идет не об электронной схеме, а о программной реализации, где «лишние» состояния – всего лишь «лишние» фрагменты кода (повторы). Главное, чтобы использовался сам принцип представления программы в виде конечного автомата, иначе создание «правильной» программы для сколько-нибудь сложной системы вообще становится «вопросом случая».

На рис. 5 показана схема технологического процесса дозирования сыпучего материала, который был выбран в качестве образца.

Разработаны принципиальные электрические схемы силовой части и цепей управления на магнитных пускателях и промежуточных реле. К схемам силовой части, как правило, вопросов не бывает, а схемы цепей управления требуют проверки, так как они уже на аппаратном уровне реализуют некоторую, пусть и несложную, логику взаимных блокировок, которая еще и по-разному работает в разных режимах управления (автоматический, ручной, ремонтный).

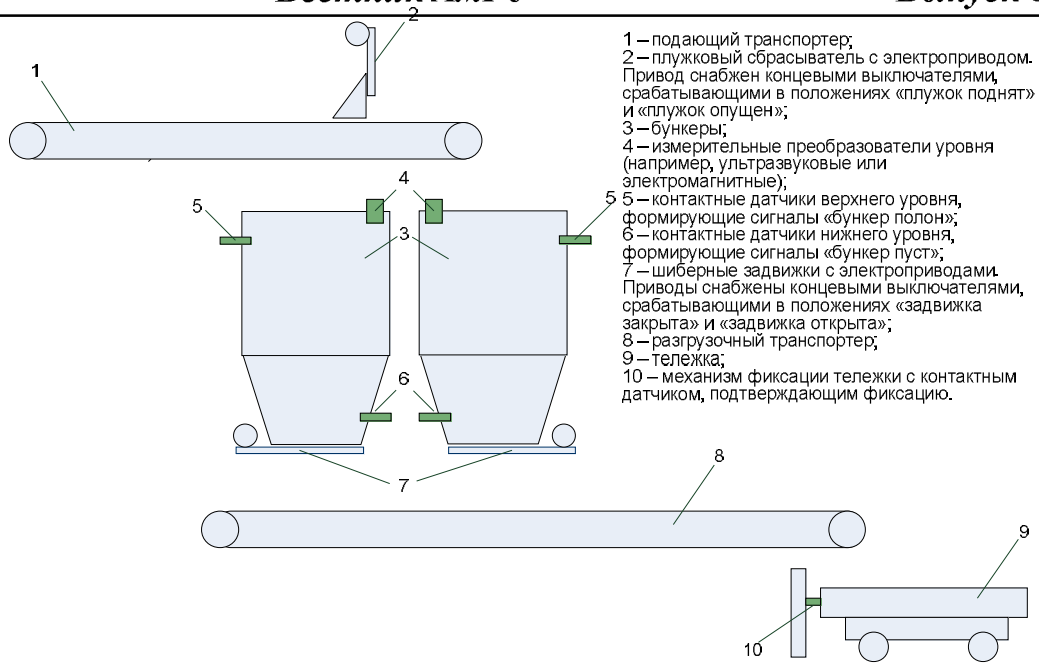

Рис. 5. Системы дозирования.

Структура программного комплекса, разработанного для данной системы, показана на рис. 6. Она состоит из двух частей: Simulink-модели объекта управления, преобразующей все входные воздействия (сигналы включения механизмов) в выходные сигналы датчиков и измерительных преобразователей, и программы CoDeSys. Последняя включает в себя «реальную» часть (то, что в последующем будет без изменения перенесено на целевую платформу – в реальный ПЛК) и «виртуальную» часть (то, что необходимо только для проверки и отладки и в дальнейшем будет реализовано аппаратурой).

Рис. 6. Структура программного комплекса.

В результате разработки получены:

1) принципиальная схема цепей управления, правильность работы которой проверена на модели. На рис. 7 показан небольшой фрагмент модели;

2) полный состав органов управления на щите и схема их размещения (рис. 8);

3) разработанный и опробованный экран операторской панели (рис. 9);

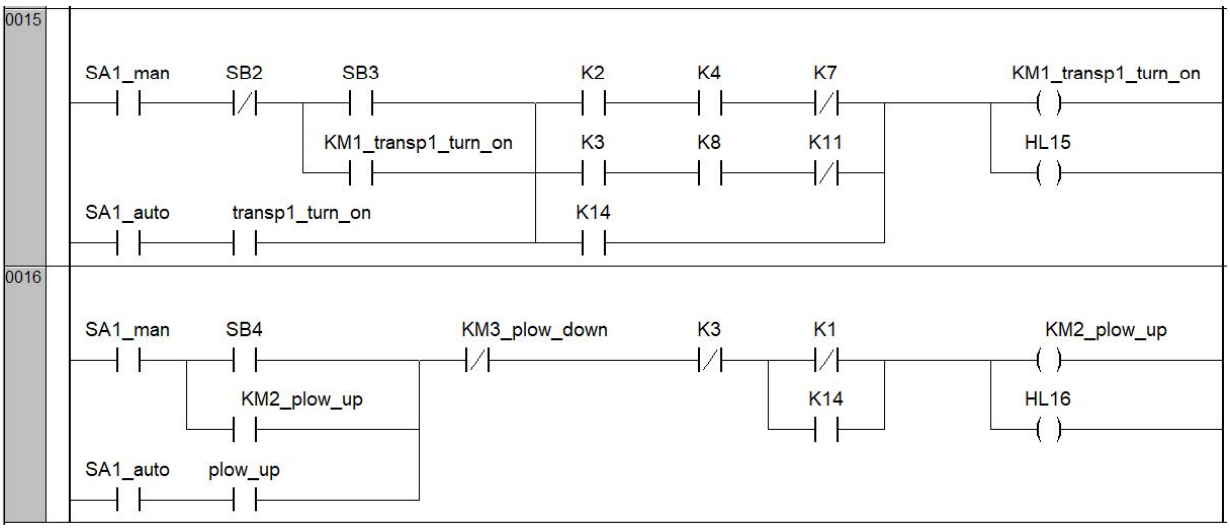

Рис. 7. Фрагмент программной модели принципиальной схемы цепей управления.

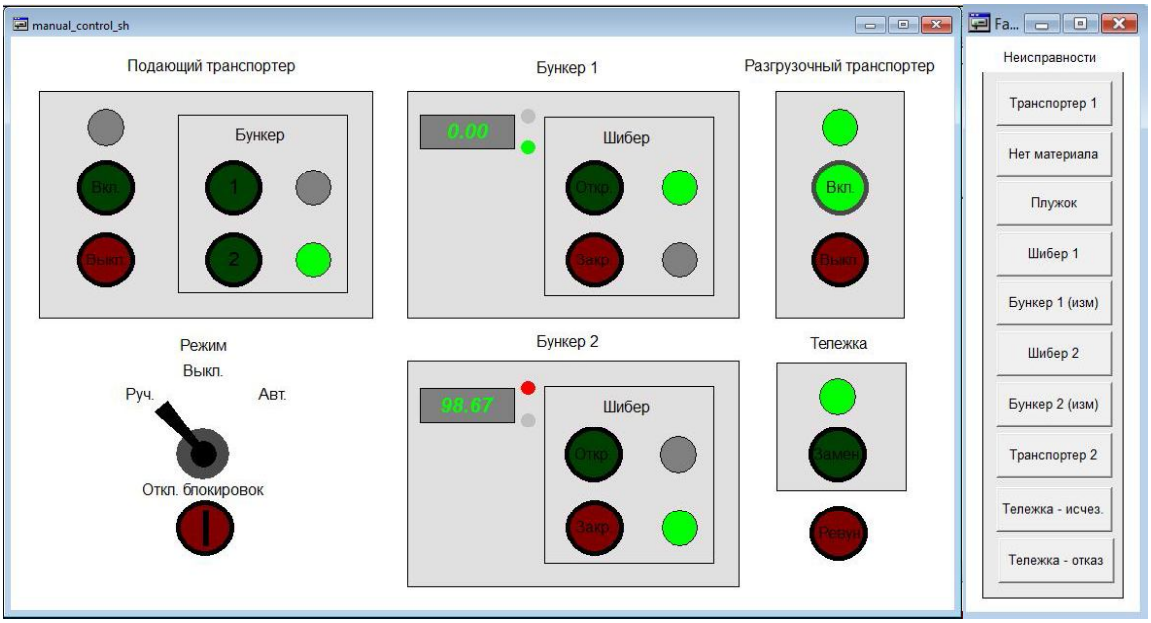

Рис. 8. Экраны визуализации для имитации щита управления и ввода неисправности (фиктивные).

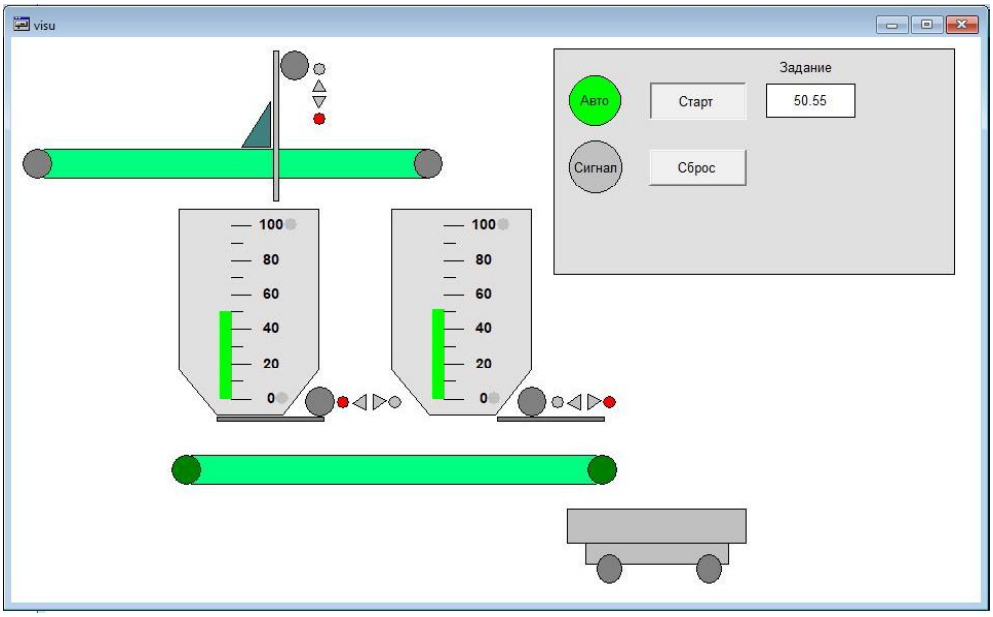

Рис. 9. Экран визуализации операторской панели.

4) отлаженная программа для ПЛК. Структура основной ее части, реализующей автоматическое управление, показана на рис. 10. Программа написана на языке ST и построена на основе управляющей конструкции CASE OF.

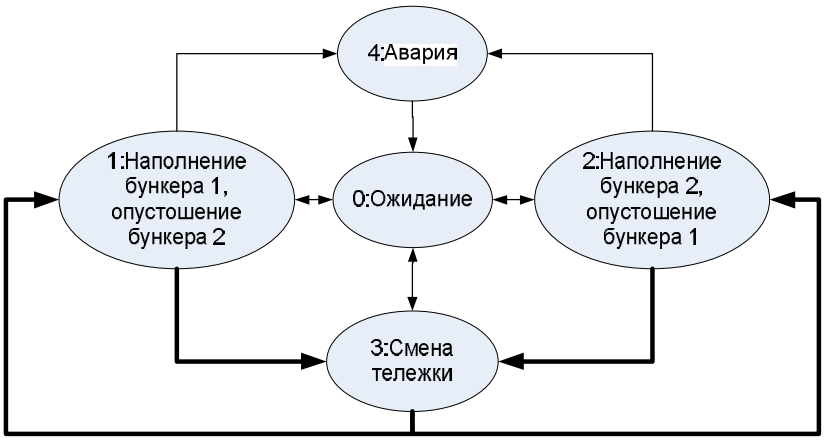

Рис. 10. Сотруктура подпрограммы автоматического управления.

Заключение

Разработаны имитационные модели типовых подсистем АСУТП, решающих задачи автоматического регулирования и программно-логического управления. На моделях отлажены алгоритмы и программные коды управления и средства человеко-машинного интерфейса. Полученные результаты могут использоваться на практике как напрямую, так и в качестве общего подхода к разработке. В учебном процессе описанные технологии проектирования уже успешно применяются по направлению подготовки 150304 «Автоматизация технологических процессов и производств». Выполняя индивидуальные задания, студенты создают имитационные модели АСУТП небольшой сложности. В настоящее время продолжаются работы по таким направлениям как экстремальное регулирование (управление системой позиционирования солнечной батареи), групповое управление с обеспечением равномерного износа агрегатов и др.

1. Рыбалев, А.Н. Компьютерное моделирование нетиповых законов регулирования для программируемых логических контроллеров // Информатика и системы управления.– Благовещенск: АмГУ.– 2016. – Вып. 4(50). – С. 33-43.

2. Рыбалев, А.Н. Реализация и компьютерное моделирование алгоритмов регулирования с транспортным запаздыванием для программируемых логических контроллеров // Информатика и системы управления.– Благовещенск: АмГУ.– 2017. – Вып. 2(52). – С. 12-24.