

предыдущем случае, сдвинутой на единицу аргумента влево.

Из сравнения функции $predict\ y_{p2}(k)$ и практических условно неизвестных $y(k)$ следует, что, хотя по амплитуде переменной функции $predict$ и $y(k)$ отличаются достаточно существенно, точки изменения первой производной (точка разворота цены) функции $predict$ и фактические данные $y(k)$ в большей или меньшей степени совпадают.

Рассмотренные выше примеры применения функции $predict$ показывают на возможность использовать её для экстраполяции экономических данных. По сравнению с обычными методами экстраполяции с помощью аппроксимирующих функций, авторегрессионные методы позволяют учесть большее число особенностей экономического объекта и более тонкие эффекты его функционирования, описанные исходными данными. Функция $predict$, так же как и другие функции, используемые для экстраполяции, могут улучшать своё качество путём настройки. В частности, показано, что используемая настройка функции $predict$ с помощью изменения ширины скользящего окна m может быть дополнена настройкой по аргументу (например, сдвигу по времени) и

при необходимости по величине (введением дополнительно к вычисленной функции $predict$ корректирующих функций или константы). Возможности настройки функции $predict$ существенно расширяют области её использования по сравнению с теми, что приведены в стандартных технических описаниях математической системы *Mathcad*. Помимо прямого предназначения функции $predict$ для предсказания изменения исследуемых переменных путём настройки она может быть использована для исследования управляемости экономического объекта заданием различных вариантов вектора исходных данных и изучением результатов предсказания $predict$ на соответствие полученных результатов желаемым.

Библиографические ссылки

1. Кирьянов Д.В. Самоучитель Mathcad 13. СПб.: БХВ-Петербург, 2006. 528 с.,
2. Макаров Е.Г. Инженерные расчёты в Mathcad 14. СПб.: Питер, 2007. 592 с.,
3. Дьяконов В.П. Mathcad в математике. М.: Горячая линия-Телеком, 2007. 958 с.,
4. Эрлих А.А. Технический анализ товарных и финансовых рынков. М.: Инфра-М, 1996. 176 с.

УДК 621.7.06; 621.9.06

В.К. Шемелин, канд. техн. наук, проф., **Р.А. Нежметдинов**
(Московский государственный технологический университет «Станкин»)

v_shem@intercomm1.ru

ПРИМЕНЕНИЕ ТЕХНОЛОГИИ «КЛИЕНТ-СЕРВЕР» ПРИ ПРОЕКТИРОВАНИИ КОНТРОЛЛЕРА ТИПА SOFT PLC ДЛЯ РЕШЕНИЯ ЛОГИЧЕСКОЙ ЗАДАЧИ В РАМКАХ СИСТЕМ ЧПУ

Предлагается принцип построения программно реализованных контроллеров Soft PLC на основе клиент-серверного подхода. Рассматривается возможность их применения в рамках общего программно-математического обеспечения на станках с числовым программным управлением для решения логической задачи управления (управления электроавтоматикой станка).

Ключевые слова: числовое программное управление, программируемый логический контроллер, клиент-серверная технология, логическая задача ЧПУ.

Construction principle of the programmatically realized controllers Soft PLC from a client-server approach is proposed. Their application possibility within of general mathematical and software framework on the CNC machine tools for control logical task decision (control of the machine tool electrics) is considered.

Key words: CNC, programmable logical controller, client-server technology, CNC logical task.

На очередном витке эволюции программируемых контроллеров появилась и получила заслуженную популярность идея их программной реализации (SoftPLC) как наиболее эффективной альтернативы традиционной аппаратной реализации в виде программируемого логического кон-

троллера (ПЛК). Наибольший эффект реализация Soft PLC даёт в системах числового программного управления (ЧПУ) типа PCNC, где программное обеспечение виртуального контроллера работает в одной операционной среде с общим программным обеспечением ЧПУ [1].

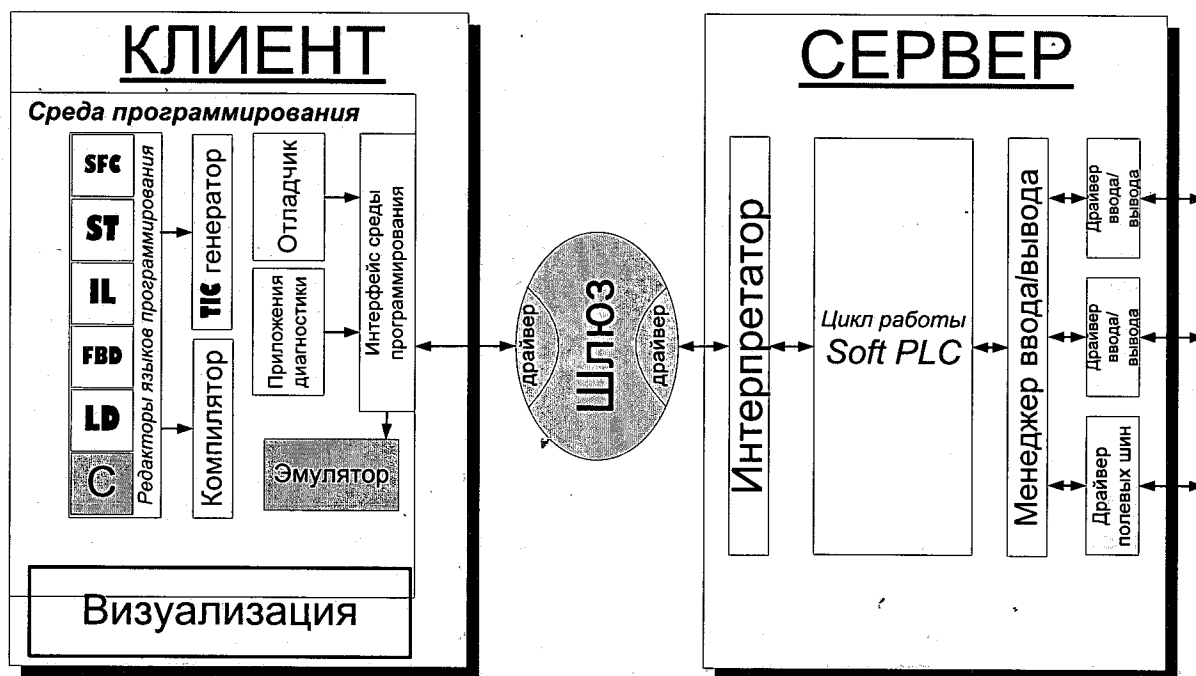


Рис. 1. Клиент-серверная структура Soft PLC

Такой подход позволяет снизить стоимость системы управления при одновременном получении ряда преимуществ. В их числе: унификация общего программного обеспечения, уменьшение ошибок системного программирования, возможность отладки управляющих программ электроавтоматики в рамках самой системы ЧПУ, гибкость конфигурирования электроавтоматики, возможность использования различных коммерческих библиотек [2].

В системах ЧПУ, построенных в соответствии с клиент-серверной моделью, процесс управления условно разделен на две части: клиент-серверная часть, реализующая логическую задачу ЧПУ (управление электроавтоматикой, рис. 1) и клиент-серверная часть, реализующая геометрическую задачу ЧПУ, включающую процесс интерпретации кодов управляющей программы, интерполяцию траектории режущего инструмента и управление следящими приводами подачи. Обе части, как задачи, работают в рамках одной операционной среды и в тесном взаимодействии друг с другом.

Рассмотрим работу данной системы управления в общем случае. На вход клиента подаётся кадр управляющей программы стандарта ISO-7bit, далее происходит анализ кадра и выделение в нём инструкций, которые необходимо подать на условного клиента Soft PLC и условного клиента ЧПУ. В клиентских приложениях ЧПУ и Soft PLC происходит определение управляющих объектов программно-исполнительной части, для которых предназначены инструкции, полученные от управ-

ляющего кадра, а также формирование управляющих взаимодействий на исполнительную часть в соответствии с полученными инструкциями.

Если рассматривать систему управления электроавтоматикой, то необходимо заметить, что размещение клиентской и серверной частей в одном персональном компьютере послужило основой концепции SoftPLC. Клиентская часть является входной частью программно реализованных логических контроллеров, с которой пользователь может ввести управляющую программу и необходимые при вычислениях значения переменных или констант. Многочисленные современные системы программирования придерживаются стандарта IEC 61131-3; при этом ориентированы на общепромышленную электроавтоматику и имеют мощную инструментальную поддержку, в которой доминирует объектный подход. Серверная часть программно реализованного логического контроллера является исполнительным уровнем, на котором согласно управляющему алгоритму происходит изменение переменных и формирование управляющих сигналов. В серверной части отдельные задачи в реальном времени работают циклически, периодически или по событию.

К операционной системе (ОС), под управлением которой должен работать такой виртуальный ПЛК, должны предъявляться достаточно жёсткие требования. Во-первых, система управления должна быть предсказуемой – потеря данных в системе управления абсолютно недопустима. Следовательно, применяемая ОС должна являться систе-

мой жёсткого реального времени с вытесняющей многозадачностью. Во-вторых, необходимо поддерживать большое количество различного оборудования — коммуникационных адаптеров, плат ввода/вывода и т. п., ассортимент которого постоянно меняется. Значит, применяемая ОС должна предоставлять простые механизмы интерфейса с оборудованием, чтобы в случае необходимости можно было быстро разработать нужный драйвер. В-третьих, использовать в качестве виртуальных ПЛК ЭВМ с избытком ресурсов было бы слишком накладно. То есть, применяемая ОС должна легко встраиваться в устройства с ограниченными ресурсами и поддерживать бездисковые конфигурации. И, наконец, в-четвёртых, для обеспечения большей отказоустойчивости необходимо обеспечить безопасность процессов, чтобы они своими действиями не могли навредить как друг другу, так и системе. Это реализуется в ОС, которые предоставляют процессам отдельные защищённые адресные пространства.

Стандартный контроллер предоставляет для размещения прикладной программы память размером всего в несколько килобайт и способен обрабатывать эту программу с тактом в несколько миллисекунд. Совсем другое дело, когда управление осуществляется ПК или в рамках системы ЧПУ, обладающей значительными ресурсами, например: мегабайтами оперативной памяти, гигагерцовым процессором, жёстким диском объёмом в десятки гигабайт, платами расширения с различными интерфейсами, сетью для связи с другими компьютерами, монитором и клавиатурой для взаимодействия с обслуживающим персоналом. Ресурсы системы могут быть использованы также для замены аппаратных средств на функциональные модули в контроллере, которые устанавливаются и поддерживаются на центральном компьютере, откуда могут быть вызваны произвольное число раз. Затраты на аппаратное и программное обеспечение уменьшаются соответствующим образом. Если требуются специальные протоколы, они могут быть написаны в контроллере с помощью стандартных языков. Дополнительный коммуникационный модуль не требуется.

Клиентское приложение программно реализованного логического контроллера имеет компонентную архитектуру и выполняет следующие функции:

- проектирование, разработку (программирование), отладку, тестирование и сопровождение программных продуктов, созданных для реализации алгоритмов работы Soft PLC, входящей в состав систем ЧПУ;

- наладку, проверку и наблюдение за системами мониторинга и контроля;

- конфигурирование пользовательского интерфейса, клиентского и серверного приложений;

- визуализация работы Soft PLC как целевой, так и WEB-визуализации.

Для реализации вышеперечисленных функций архитектура клиентского приложения программно реализованного логического контроллера состоит из следующих компонент:

- среды программирования*, отвечающей за реализацию и сопровождение программ, реализующих алгоритмы работы Soft PLC;

- средств мониторинга и контроля*, встроенных в среду программирования, которые могут быть реализованы с помощью *пользовательских приложений*, использующих открытые интерфейсы программирования;

- менеджера конфигурации*, полностью отвечающего за функционал точной настройки всех компонент системы как на клиентском приложении программно реализованного логического контроллера, так и на его серверной части. Настройку можно осуществлять как непосредственно с помощью средств клиентского приложения Soft PLC, так и удалённо с помощью Internet и сети Ethernet;

- компонента визуализации*, отвечающего за все виды визуального сопровождения работы Soft PLC.

Серверное приложение представляет собой управляющий цикл. При управлении цикл периодически повторяется заданное число миллисекунд; в нём читаются входы, производятся вычисления и записываются выходы. Время цикла может быть до одной миллисекунды. На контроллере может иметься несколько работающих с различными временами циклов, выполняемых с очень высоким приоритетом. Также применяется менеджер ввода/вывода и один или более драйверов для связи с полевым оборудованием. Драйверы работают параллельно, и менеджер ввода/вывода гарантирует качество и скорость сбора данных. Имеется также шлюз, позволяющий полемому оборудованию инициализировать связь и общаться с управляющими циклами, которые могут опрашивать устройства на полевых шинах или опрашиваться подобно обычным PLC.

Как и в случае клиентского приложения программно реализованного логического контроллера, серверное приложение имеет компонентную архитектуру и применяется:

- для интерпретации TIS-кода, полученного на основе одного из языков программирования электроавтоматики в клиентском приложении программно реализованного логического контроллера;

- чтения входных данных, необходимых для выполнения алгоритма работы Soft PLC.

Так как параллельно могут исполняться несколько циклов, реализующих различные задачи по управлению электроавтоматикой, то чтение осуществляется как со входов Soft PLC, так и из других параллельно выполняемых циклов работы:

выполнения алгоритма работы электроавтоматики, заданного программно в клиентском приложении программно реализованного логического контроллера;

записи выходных данных, полученных в процессе работы Soft PLC. Запись осуществляется как на выходы Soft PLC, так и на входы параллельно выполняемых циклов работы;

ведения архива, в котором фиксируется история работы исполнительской части Soft PLC;

Для реализации вышеперечисленных функций архитектура серверного приложения программно реализованного логического контроллера включает в себя следующие компоненты.

Интерпретатор. Это программная компонента, которая допускает в качестве входа исходную программу, записанную на языке, называемом исходным языком, и производит вычисления, предписываемые этой программой. Соответственно в Soft PLC исходным языком выступает сгенерированный ПИС-код.

Компонента, реализующая исполнительную часть Soft PLC. Так же, как и в обычных PLC, программно реализованный логический контроллер использует модель детерминированного цикла сканирования (scan cycle, скан-цикл). Периодически выполняется жёстко заданная последовательность исполнения (опрос входных каналов, просчёт управляющей логики и передача данных в выходные каналы).

Менеджер ввода/вывода. Данная компонента координирует сбор и обработку данных со всех устройств ввода/вывода, осуществляет управленческие драйверами устройств.

Менеджер конфигураций. Отвечает за функционал точной настройки компонент системы на серверной части программно реализованного логического контроллера. Менеджер конфигураций может получать данные от аналогичной компоненты клиентского приложения и перестраивать работу серверного приложения или же осуществлять настройку с помощью средств Internet и сети Ethernet, т. е. удалённо.

База данных реального времени. Получает, обрабатывает и хранит данные о работе Soft PLC.

Так же, как и в обычных PLC, исполнительная часть Soft PLC использует модель детерминированного цикла сканирования (scan cycle): периодически выполняется жёстко заданная последовательность действий (опрос входных каналов, просчёт управляющей логики и передача данных в выходные каналы).

Благодаря возможностям высокопроизводительной многозадачной ОС РВ, а также мощности современных аппаратных средств обеспечивается минимальная длительность цикла работы исполнительской части Soft PLC в пределах одной миллисекунды. Если позволяют ресурсы аппарат-

ной платформы (CPU, DRAM, I/O device,...), то могут одновременно исполняться несколько независимых циклов работы. В Soft PLC типичный цикл сканирования расширен функциями опроса и отправки переменных другим циклам работы. Это позволяет легко использовать совместную информацию в реальном времени.

Программно реализованный логический контроллер, подобно аппаратному ПЛК, должен уметь одновременно выполнять несколько команд и параллельно обрабатывать поступающие внешние сигналы. Внутренняя организация Soft PLC подразумевает, что все процессы контроллера, которым выделены отдельные потоки, выполняются в рамках основного процесса виртуального контроллера, работающего в реальном времени. Процессорное время, выделяемое операционной системой основному процессу, должно быть распределено между потоками.

При реализации исполнительской части Soft PLC необходимо использовать механизм псевдо-многопоточности. Процессорное время должно выделяться потокам отдельными конечными интервалами времени с помощью внутренних механизмов Soft PLC. В каждый момент времени должен выполняться только один поток. Потоки должны быть разделены на группы по приоритетам. Данное разделение потоков на несколько групп в программно реализованном логическом контроллере необходимо для того, чтобы задачи, требующие малого времени реакции на внешнее воздействие, обрабатывались с более высоким приоритетом, чем все остальные.

Высокоприоритетные потоки будут иметь более высокую частоту выделения рабочих интервалов времени. Необходимо также программно реализовать таймеры, которые будут синхронизировать частоту выделения рабочих интервалов времени. Так как реализация исполнительской части Soft PLC подразумевает много приоритетов, то за каждую из частот синхронизации будет отвечать отдельный таймер; каждый из таймеров работает на своей частоте и выделяет рабочие интервалы времени своим потокам.

Алгоритм выполнения программы аппаратного ПЛК в циклическом режиме показан на рис. 2. Содержание отдельных фаз цикла состоит в следующем. Фаза системного анализа отвечает за осуществление мониторинга контроллера (проверяет доступность памяти, обработку запросов со стороны портов программирования и расширения, а также следит за сменой режима RUN/STOP, контролирует системные параметры и др.). В фазе чтения входных данных происходит обновление внутренней памяти данными с входов контроллера. В фазе вычисления выполняется программа, написанная пользователем. В фазе записи выходных данных обновляется состояние выходов контроллера.



Рис. 2. Цикл работы классического ПЛК

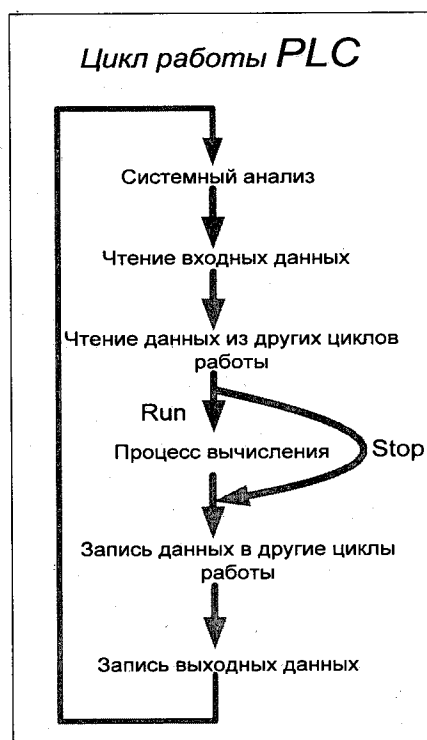


Рис. 3. Цикл работы Soft PLC

Цикл управления включает контроллер (процессор управляет системой, читает входы, выполняет программу и обновляет выходы) или отключает его (процессор лишь управляет системой, читает входы и обновляет таблицу образов выходов; физические выходы не обновляются).

Время цикла контролируется сторожевым таймером и не должно превышать определенного значения, например 150 мс. Иначе возникает ошибка, останавливающая контроллер. Существуют две ситуации:

1. Время цикла сканирования меньше или равно настройке сторожевого таймера (150 мс). Это нормальная ситуация, при которой запускается очередной цикл сканирования.

2. Время сканирования больше настройки сторожевого таймера. Контроллер останавливается, загорается аварийная лампочка.

Данный процесс циклической работы ПЛК полностью реализован в исполнительной части Soft PLC, однако к имевшимся фазам работы в цикле добавились ещё фазы чтения и записи данных в другие циклы работы Soft PLC (они необходимы в случае, если используются одинаковые переменные в параллельно работающих циклах) (рис. 3).

Если были определены межресурсные связи переменных, принимаемые значения связанных переменных обновляются после опроса входных переменных, а выдаваемые другим ресурсам значения переменных посылаются перед обновлением выходных регистров. Данный цикл работы повторяется периодически; если было установлено время цикла, то процесс ожидает истечения этого времени перед началом очередного цикла. Время выполнения цикла работы может варьироваться в зависимости от активных шагов в SFC-программах или выполнения таких инструкций, как Jump, IF и Return и др.

В результате была определена зависимость повышения качества управления электроавтоматикой станка типа PCNC от функционального построения программно реализованных контроллеров SoftPLC в виде компонентной клиент-серверной структуры, встроенной в общую структуру программного обеспечения системы ЧПУ типа PCNC, а также предложена компонентная архитектура клиентской и серверной частей программно реализованного контроллера SoftPLC, которая удовлетворяет всем требованиям, предъявляемым к системе управления электроавтоматикой ЧПУ.

Библиографические ссылки

1. Сосонкин В.Л., Мартинов Г.М. Концепция числового программного управления мехатронными системами: реализация логической задачи // Мехатроника. 2006. № 2. С. 3-5.

2. Шемелин В.К., Нежметдинов Р.А. Повышение качества архитектурных решения систем ЧПУ на основе программно реализованного контроллера типа Soft PLC // Автоматизация и современные технологии. 2008. № 6. С. 33-35.