

Библиографические список

1. Змиева К.А. Повышение энергоэффективности промышленных производств посредством создания автоматизированных программно-управляемых энергосберегающих систем // Вопросы современной науки и практики. Университет им. В.И. Вернадского. 2009. № 9. С. 35 – 40.
2. Змиева К.А. Автоматизированная установка для компенсации реактивной мощности недогруженных электроприводов металлорежущего оборудования // Электро. Электротехника, электроэнергетика, электротехническая промышленность. 2010. № 6. С. 28 – 34.
3. Змиева К.А., Кузнецова Е.В., Шумихина Е.М. Поиск направлений повышения энергоэффективности производственного оборудования // Справочник. Инженерный журнал. 2011. № 12. С. 52 – 55.
4. Григорьев С.Н., Шумихина Е.М., Змиева К.А. Метод повышения энергоэффективности технологий вакуумно-плазменного нанесения покрытий // Вестник МГТУ «СТАНКИН». 2010. № 1. С. 82 – 87.
5. Козлов Д.В., Змиева К.А., Шумихина Е.М. Экспериментальная установка для исследования характеристик двигателя при различных нагрузках и питающих напряжениях ЭП-1 // Электротехнические комплексы и системы управления. 2011. № 1. С. 12 – 18.
6. Григорьев С.Н., Змиева К.А. Методика адаптивного управления энергопотреблением вакуумных насосов // Электротехнические комплексы и системы управления. 2011. № 4. С. 1 – 7.
7. Змиева К.А., Кузнецова Е.В. Методика расчета коэффициента мощности электродвигателя в условиях несинусоидальности тока и напряжения // Вестник МГТУ «СТАНКИН». 2011. Т. 2. № 4. С. 39 – 42.
8. Змиева К.А., Кузнецова Е.В., Козлов Д.В. Разработка инновационной энергосберегающей технологии повышения энергоэффективности машиностроительного оборудования // Автоматизация и современные технологии. 2012. № 7. С. 13 – 20.
9. Змиева К.А. Автоматизированное управление энергопотреблением машиностроительных производств с целью повышения их энергоэффективности: дис. ... канд. техн. наук. М.: МГТУ «СТАНКИН», 2009.
10. Григорьев С.Н., Змиева К.А., Кулагин О.А. и др. К вопросу о методах расчета коэффициента мощности трехфазного асинхронного двигателя // Автоматизация и современные технологии. 2011. № 12. С. 6 – 12.
11. Змиева К.А., Кузнецова Е.В. Интеллектуальные системы управления энергопотреблением промышленного предприятия // Электро. Электротехника, электроэнергетика, электротехническая промышленность. 2012. № 3. С. 11 – 15.

УДК 621.9.08

С.Н. Григорьев, Г.М. Мартинов (ФГБОУ ВПО МГТУ «СТАНКИН», Москва)
E-mail: n.cherkasova@stankin.ru

УПРАВЛЕНИЕ И ДИАГНОСТИКА ЦИФРОВЫХ ПРИВОДОВ СТАНКОВ С ЧПУ

Обоснована необходимость создания единого решения в системе ЧПУ для управления и диагностики цифровых приводов; проанализированы тенденции мировых лидеров в области диагностики следящих приводов. Предложена организация взаимодействия ядра системы ЧПУ с драйверами цифровых приводов, не зависящего от специфики операционной системы; предложена архитектура кроссплатформенного драйвера для сервоприводов, управляемых по протоколам CAN, SERCOS и EtherCAT. Классифицированы функциональные задачи цифрового осциллографа как базового решения для инструментария диагностики цифровых приводов. Представлена компонентная архитектура цифрового осциллографа и приведены примеры практической реализации.

The necessity of creating a common solution in the CNC system for the control and diagnostic of digital drives is explained. The world leaders' trends in the field of servo drives diagnostics are analyzed. The organization of digital servo drives driver interaction with the OS-independent NC kernel is proposed. The architecture of a cross-platform driver for CAN, SERCOS and EtherCAT servo drives is suggested. The functional tasks of digital oscilloscope as a basic solution for digital drives diagnostic tool are classified. The component architecture of digital oscilloscope with examples of practical implementation is presented.

Ключевые слова: система ЧПУ, цифровой осциллоскоп, драйвер приводов, CAN, SERCOS, EtherCAT.

Keywords: CNC system, digital oscilloscope, driver of servo drivers, CAN, SERCOS, EtherCAT.

Цифровые приводы перестали быть эксклюзивным решением для прецизионных и сверхскоростных станков и уверенно вытесняют аналоговых предшественников [1, 2]. До сих пор одним из наиболее распространенных цифровых интерфейсов электроприводов остается интерфейс CAN. Однако

рост вычислительных возможностей систем ЧПУ потребовал применения более быстродействующей промышленной сети. Сегодня на рынке доминирует семейство протоколов на базе шины Ethernet, но больший интерес представляют открытые протоколы SERCOSIII и EtherCAT. Протокол SERCOS

разработан для управления приводами, а в третьей версии ему добавлена возможность работы с входами/выходами электроавтоматики в сети управления приводами. Протокол EtherCAT ориентирован на управление входами/выходами. Для управления приводами используются протоколы CAN (CoE) или SERCOS (SoE) поверх шины EtherCAT, т.е., по сути, шина является только транспортным средством [3].

Управление цифровыми приводами требует некоторой реорганизации системы управления и дополнительного инструментария для их настройки и диагностики. Современные системы числового программного управления располагают определенными свободными ресурсами вычислительных мощностей, которые весьма рационально могут быть использованы для решения диагностических задач. В первую очередь следует диагностировать логическую, геометрическую и технологическую задачи управления [4 – 6].

Диагностирование цифровых приводов в рамках системы ЧПУ нельзя рассматривать отдельно от процесса управления. Проблема заключается в многообразии протоколов цифровых приводов, в многочисленности производителей, которые привносят специфику в свои изделия, а также в необходимости создания отдельных решений для систем управления под конкретные платформы операционной системы.

В работе рассмотрено кроссплатформенное решение для управления цифровыми приводами и построение цифрового осциллоскопа, интегрируемого в систему ЧПУ как одного из самых универсальных инструментов настройки технологического оборудования и исполнительных устройств.

Организация взаимодействия ядра системы управления с драйверами приводов

Программный код ядра системы ЧПУ структурно поделен на две основные части (рис. 1) – платформно-зависимый и платформно-независимый коды. Этот прием обычно применяется при разработке переносимого (portable) программного обеспечения [7].

Основу программного кода составляют платформно-независимые типы данных, структуры и функ-

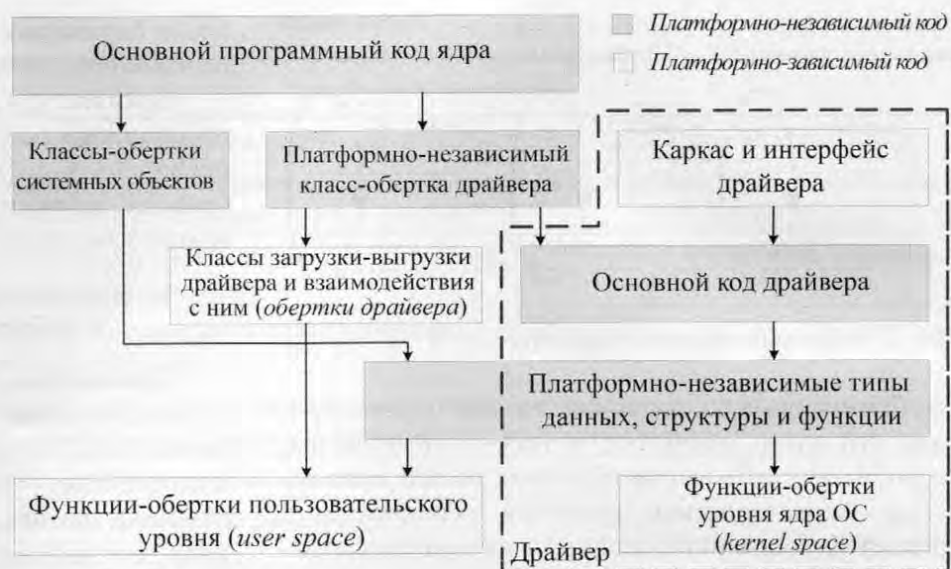


Рис. 1. Структура уровней программного кода ядра системы ЧПУ

ции, которые, в свою очередь, используют соответствующие платформно-зависимые функции-обертки.

Платформно-независимый уровень реализуют функции: C-Runtime (в том числе распределения и освобождения памяти), потоков, определения текущего времени и реализации задержек, синхронизации объектов, совместно используемая память (sharedmemory), доступа к портам ввода-вывода, получения информации о PCI/PCI-E устройствах, работы с прерываниями.

Такие объекты, как обертки драйвера и каркас драйвера, содержатся по одному набору на каждую поддерживаемую платформу и учитывают все особенности конкретной платформы. Обертки драйвера реализуют один и тот же программный интерфейс, а каркасы драйвера – один и тот же набор вызываемых ими платформно-независимых функций. Программно-независимый класс-обертка драйвера при компиляции кода автоматически выбирает и использует ту обертку драйвера, которая соответствует целевой операционной системе.

Архитектура кроссплатформенного драйвера цифрового привода

Драйвер написан на языке C, и его объекты реализованы с использованием платформно-независимых структур. Функционирование программного кода обеспечивается несколькими потоками, в которых передаются указатели на структуры (рис. 2).

Канал драйвера соответствует каналу управления. Система ЧПУ, способная одновременно исполнять N управляющих программ, должна иметь N каналов.



Рис. 2. Основные объекты драйвера

Объект привода содержит данные: о типе привода, его адрес, расчетное и текущее положения, скорость, вид замыкания обратных связей привода и т.д. Характеристики приводов переносятся в драйвер на этапе инициализации из машинных параметров системы управления.

Системные переменные используются для синхронизации работы каналов и приводов, а также для реализации различных станочных циклов.

Поток жесткого реального времени формирует расчетную траекторию движения и передает данные в соответствующие объекты приводов драйвера.

Обработчики прерываний соответствуют количеству SERCOS- либо EtherCAT-мастеров и получают указатель на массив структур приводов. SERCOS- и EtherCAT-приводы управляются по схеме работы с прерываниями; передача заданий приводам выполняется в контексте обработчика прерывания. Приводы по шине CANbus управляются по схеме циклического опроса (pooling), где в

цикле опрашиваются текущие данные с приводов и передаются новые задания на исполнение.

Архитектура предлагаемого решения позволила использовать совместно обе схемы управления приводами в системе ЧПУ. В драйвере цифрового привода заложена базовая функциональность, необходимая для диагностики приводов.

Классификация функциональных задач цифрового осциллографа

Диагностика, выполняемая при запуске в эксплуатацию или при плановом обслуживании технологического оборудования, реализуется «вне реального времени». Это означает, что измерения сохраняются в памяти, а затем анализируются. Подсистема диагностики способна: конфигурировать измерения, считывать измеряемые сигналы, запоминать результаты измерений вместе с результатами конфигурации измерений, распечатывать осциллограммы измерений, считывать файлы с результатами измерений и их конфигураций, выполнять вычислительные операции над измеренными сигналами [8].

В таблице выделены три основные группы прикладных задач. Первая – *управление устройствами и процессом измерения* – определяет процедуру измерения. Вторая – *отображение сигналов* – связана с графическим представлением измеренных данных на экране и возможностью манипулирования сигналами. Третья – *математические преобразования и анализ сигналов* – предназначена для более глубокого анализа диагностируемой системы.

Классификация функциональностей прикладных приложений диагностики

Задачи		Функциональность
Управление устройствами и процессом измерения	Управление устройствами	Подключение и отключение устройства для диагностики
		Параллельная работа с несколькими устройствами, входящими в технологическую систему
		Считывание и изменение параметров устройств
	Конфигурирование измеряемых сигналов	Выбор измеряемых сигналов, определение частоты дискретизации и времени или количества точек для измерения
	Настройка триггеров	Управление группой триггеров, запускающих или останавливающих измерение, определение логики срабатывания группы
		Установка типа триггера (ручной, битовый, сигнальный, программный)
		Установка параметров для битовых, программных и сигнальных триггеров (например, сигнал для триггера, пороговое значение, маска, условие срабатывания)
	Управление процессом измерения	Запуск и останов измерения
		Отображение статуса измерения
		Загрузка измеренных данных из устройства
Сохранение и загрузка в файл данных конфигурации и измерений		

Продолжение табл.

Задачи		Функциональность
Отображение сигналов	Графическое отображение сигналов	Отображение сигналов на экране в декартовой системе или логарифмической шкале
		Задание режима отображения сигналов (непрерывная кривая или точные значения), выбор цвета и стиля линий отрисовки
	Навигация по сигналам	Отображение иерархического списка устройств, измерений и сигналов и возможности навигации по ним
		Операции по скрытию и показу сигналов, удалению и добавлению устройств, измерения и сигналов
	Фильтрация сигналов	Установка фильтра визуализируемых сигналов (например, только измеренные, только вычисленные, только контурные, только битовые и т.д.)
		Задание комбинации фильтров
		Сохранение параметров фильтрации
	Масштабирование	Сдвиг сигнала по осям координатной сетки (Shifting)
		Масштабирование сигнала по осям координатной сетки (Scaling)
		Задание общего масштаба для всех видимых сигналов (Common scaling)
		Задание автоматического масштабирования – каждый сигнал занимает максимально возможную область экрана (Auto scaling)
		Увеличение/уменьшение участка в выбранном прямоугольнике или определенного по позициям курсоров (Zoom)
	Управление курсорами	Поддержка горизонтальных, вертикальных и контурных курсоров
		Отображение значения сигналов в точках пересечения с курсорами
		Связывание курсоров и их синхронное перемещение
Математические преобразования и анализ сигналов	Анализ спектра	Разложение на гармоники с помощью быстрого преобразования Фурье (FFT)
		Построение амплитудно-частотных характеристик (Bodydiagram)
	Логический анализ битовых сигналов	Вычисление битовых сигналов
		Сортировка битовых сигналов на экране по номеру бита или последовательно один за другим
	Анализ контура	Вычисление контурных сигналов
		Вычисление отклонений от траектории (Patherror)
		Реализация теста окружности с алгоритмом масштабирования ошибки
	Арифметические операции над сигналами	Арифметические вычисления с сигналами (сложение, вычитание, умножение, деление, логарифмирование, интегрирование, дифференцирование и т.д.)
	Сервисные функции	Автоматический пересчет вычисленных сигналов при повторе измерения
		Сохранение и загрузка вычисленных сигналов, параметров вычислений

Каждая из этих групп состоит из логически связанных между собой прикладных задач, которые в свою очередь реализуются набором прикладных функциональностей приложений диагностики. Предложенная классификация позволяет определить набор программных компонентов универсального осциллографа и закрепить прикладные функциональности за ними.

Компонентная архитектура цифрового осциллоскопа

Ведущие производители систем ЧПУ – Fanuc, Siemens, BoschRexroth, Mitsubishi, Fagor и др. –

в свои системы управления включают осциллограф в виде встроенного решения [9 – 11]. Другие производители, например Renishaw и Heidenhain, предлагают внешние измерительные системы ballbar и KGM соответственно.

Внешние измерительные системы позволяют комплексно диагностировать станок, включая все элементы и обратные связи, но не позволяют проводить измерения в процессе обработки, тогда как в современных системах ЧПУ есть вся информация об измерениях или при необходимости ее можно получить [12]. В системах управления используются



Рис. 3. Открытость базового решения осциллоскопа

разные приложения диагностики и настройки, хотя большинство из них могло быть реализовано на основе некоего общего решения.

Базовое решение цифрового осциллоскопа реализует открытость в отношении поддерживаемого списка устройств для проведения измерения. Каждому физическому устройству сопоставляется виртуальное устройство (virtualdevice) с фиксированным интерфейсом. На сегодняшний день реализована работа с приводами SERCOS (электрическими и гидравлическими), EtherCAT, CAN. Также возможна работа с системами управления ПЛК и ЧПУ, к которым могут быть подключены несколько приводов.

Измеренные сигналы от исполнительных устройств подвергаются математической обработке. Открытость для анализа сигналов позволяет специалистам интегрировать собственные вычислители. В базовом наборе реализованы: арифметические операции, deviation, быстрые преобразования Фурье (FFT), частотный анализ, логический анализатор, оценка погрешности пути, оценка погрешности теста окружности согласно ISO 230-4 и т.д.

На базе набора компонентов ядра строятся специфичные приложения диагностики: осциллоскоп реального времени, FFT Analyzer для спектрального анализа и определения резонансных частот, CyclicTimeAnalyzer для отслеживания использования ресурсов в сложных процессах, ContourDisplay для анализа траектории движения, Frequency Measurement для анализа амплитудно-частотных характеристик, CircleTest для определения отклонения от окружности, Logic Analyzer для анализа сигналов электроавтоматики (рис. 3).

Примеры приложений, построенных на базе набора компонентов ядра цифрового осциллоскопа, проиллюстрированы на рис. 4.

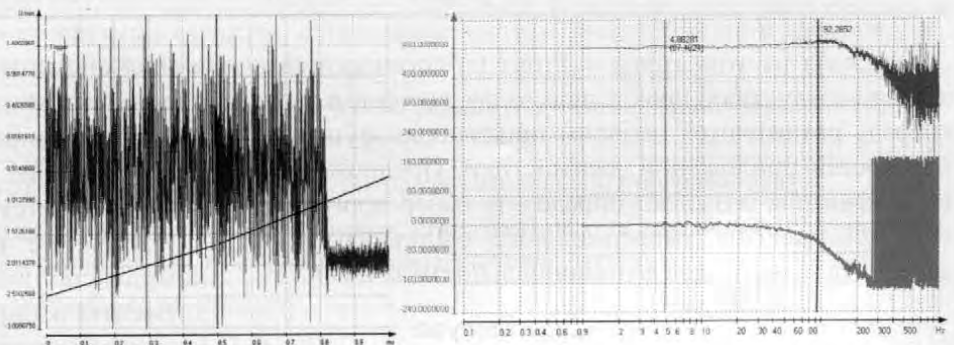


Рис. 4. Скользящие средние (MovingAverage) и FFT/полоса пропускания (Bandwidth)

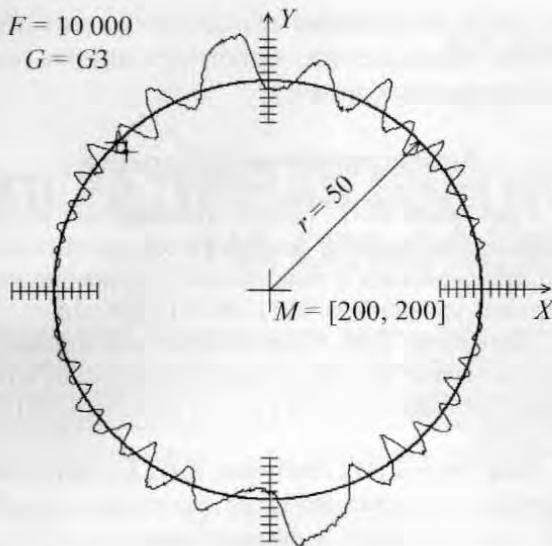


Рис. 5. Приложение для проведения теста окружности, реализованное на базе цифрового осциллоскопа

При создании приложения для отображения результатов измерений часто требуются специальные графические символы и метки на экране. Реализованный механизм открытости пользовательского рисования позволяет применять SVG-язык описания графических примитивов. Диаграмма «тест окружности» иллюстрирует применение SVG-примитивов (рис. 5). Пользователь вводит информацию о радиусе окружности, направлении вращения, значении подачи обхода контура; после измерений специальный алгоритм масштабирует ошибку для ее отображения на экране (ее величина в пределах нескольких микрон). Информация об измерении, дополненная координатными осями, отображается серым цветом с помощью SVG.

Использование XML-формата описания сигналов позволяет применять XSLT-трансформацию при генерации отчетов, конвертации форматов и обработки данных.

Компонентный подход обеспечивает открытость интеграции. На базе цифрового осциллографа можно создавать встроенные в систему ЧПУ диагностические приложения или автономный инструментарий.

Практические аспекты использования web-технологии

Благодаря наличию готового инструментария на рынке, легкости использования, независимости от аппаратных и программных платформ, простоте интеграции в пользовательские приложения, возможности описания данных практически любой сложности, язык XML получил широкое распространение. Информация, оформленная в виде XML в отличие от двоичных форматов может обрабатываться не только машинами, но и человеком. Широкое применение XML породило идею его использования для хранения диагностических данных. В формате XML можно сохранять информацию как о конфигурации измерений (точки измерения, использованные каналы, условия начального и конечного триггера, *samplingtime*, ...), так и значения измеренных сигналов и информацию о примененных к ним вычислениях (*computations*), настройки отображения экрана и т.д. Значения измеренных сигналов хранятся в двоичном коде в XML-теге, поскольку количество измеренных точек в сигнале может превышать 500 000 и обработка таких сигналов длилась бы недопустимо долго.

Данные об измерениях, хранящиеся в формате XML, обрабатываются XSLT-скриптами для конвертации форматов данных, генерации отчетов, фильтрации информации и т.д. (рис. 6). Конечный пользователь пишет XSLT-скрипты под свои задачи и применяет их. Для этого достаточно иметь описание формата данных XML или файл схемы *.xsd.

Пример на рис. 7 иллюстрирует создание сигналов SVG-viewer на базе web-браузера. Такое решение хотя и является более медлительным, чем

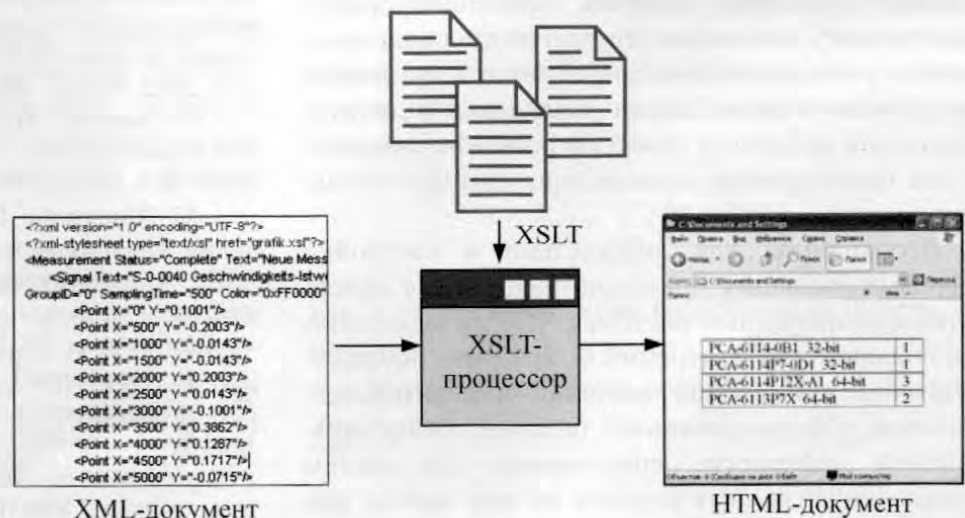


Рис. 6. Генерация отчета с помощью XSLT-трансформации

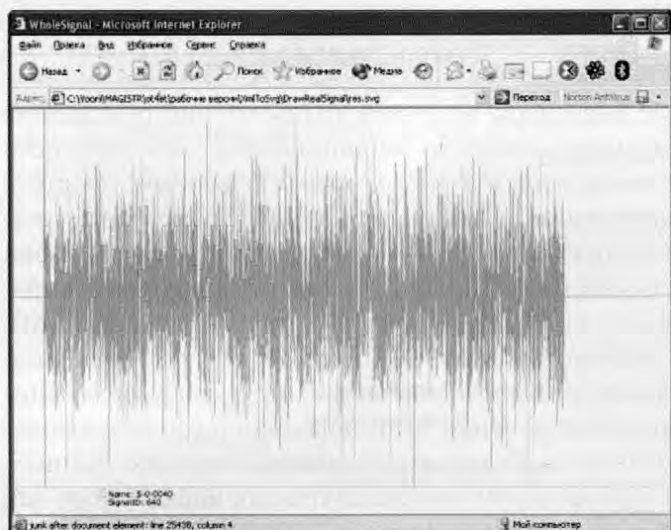


Рис. 7. Визуализация XML-данных с использованием SVG

обычное диагностическое приложение, но позволяет просматривать и анализировать измеренные сигналы на любом персональном компьютере web-браузером без дополнительных ограничений на аппаратуру и программные лицензии.

Заключение

Производители систем ЧПУ ориентируют свою продукцию на конкретный интерфейс цифрового привода и не заинтересованы в продвижении на рынок своих конкурентов, с другой стороны, сложность управляемого технологического оборудования требует возможности совместного применения следящих приводов и исполнительного оборудования от разных производителей. Предложенное решение является кроссплатформенным, поэтому адаптация под другие операционные системы реального времени требует замены только определенного слоя. Задачи управления и диагностирования цифровых приводов решаются совместно при формировании архитектуры драйвера следящего привода.

Необходимая для диагностики и настройки цифровых следящих приводов информация содержится в современных системах ЧПУ, а подавляющее большинство измерений может быть осуществлено без применения внешних измерительных приборов. Сформированный базовый набор компонентов цифрового осциллографа для систем управления позволяет строить на его основе как мощные универсальные инструментари

сти, так и несложные в использовании диагностические приложения, ориентированные на решение конкретных задач.

Библиографический список

1. Григорьев С.Н. Научно-технические проблемы построения современных технологических систем с числовым программным управлением // Мехатроника, автоматизация, управление. 2011. № 4. С. 19 – 26.
2. Мартинов Г.М. Современные тенденции развития компьютерных систем управления технологического оборудования // Вестник МГТУ «СТАНКИН». 2010. № 1. С. 119 – 125.
3. Мартинов Г.М., Любимов А.Б., Бондаренко А.И. и др. Подход к построению мультипротокольной системы ЧПУ // Автоматизация в промышленности. 2012. № 5. С. 8 – 11.
4. Мартинов Г.М., Мартинова Л.И. Современные тенденции в области числового программного управления станочными комплексами // СТИН. 2010. № 7. С. 7 – 11.
5. Сосонкин В.Л., Мартинов Г.М. Концепция числового программного управления мехатронными системами: реализация логической задачи управления // Мехатроника, автоматизация, управление. 2001. № 2. С. 3 – 7.
6. Григорьев С.Н., Мартинов Г.М. Концепция построения базовой системы числового программного управления мехатронными объектами // Информационные технологии в проектировании и производстве. 2011. № 2. С. 21 – 27.
7. Григорьев С.Н., Андреев А.Г., Мартинов Г.М. Перспективы развития кроссплатформенных компьютерных систем числового программного управления высокотехнологичного оборудования // Автоматизация в промышленности. 2011. № 5. С. 3 – 8.
8. Мартинов Г.М., Трофимов Е.С. Модульная компоновка и построение прикладных приложений диагностики систем управления // Приборы и системы. Управление, контроль, диагностика. 2008. № 7. С. 44 – 50.
9. Бретшнайдер Д. Решения для автоматизации металлообрабатывающих станков на основе инновационной системы ЧПУ // Автоматизация в промышленности. 2012. № 5. С. 12 – 15.
10. Брюль Й., Нотнагель Й. Система ЧПУ IndraMotion MTX // Автоматизация в промышленности. 2012. № 5. С. 16 – 18.
11. Мартинов Г.М., Мартинова Л.И., Козак Н.В. и др. Принципы построения распределенной системы ЧПУ технологическими машинами с использованием открытой модульной архитектуры // Справочник. Инженерный журнал. 2011. № 12.
12. Мартинов Г.М., Нежметдинов Р.А., Соколов С.В. Способ построения инструментари систем мониторинга и настройки параметров мехатронного технологического оборудования на основе специализированных программных средств // Мехатроника, автоматизация, управление. 2012. № 7. С. 45 – 50.