

Принципы построения системы ЧПУ с открытой архитектурой

Сформулированы принципы построения системы ЧПУ с открытой архитектурой, в соответствии с которыми система не только должна быть гибкой на всех стадиях своего жизненного цикла (у производителя, станкостроителя, конечного пользователя), но и располагать модульной архитектурой и стандартными структурами внутренних данных. Предложено строить модули в виде гибких командных процессоров. Определены функции отдельных процессоров. Наиболее подробно рассмотрены концепции построения ISO-процессора и "открытого" интерполятора.

The principles are formulated for the numerical programmed control system development. In correspondence with the principles the systems should be flexible during the all stages of its life (development, production, operation) and be assembled as a modular one with the conventional (standart) inside data formats. It is offered to build the modules on the basis of the flexible programmed processors. The separate processor's functions are appointed. The development of the ISO — processor and interpolator are described in more details.

Основное достоинство современной системы ЧПУ — гибкость на всех стадиях ее жизненного цикла: у производителя, станкостроителя, конечного пользователя.

Производители систем ЧПУ заинтересованы в некоторой *структурно гибкой* базовой системе управления, на основе которой можно строить многочисленные модификации соответственно запросам чрезвычайно подвижного рынка систем управления. Яркими примерами структурно гибких могут послужить системы ЧПУ фирмы Bosch (Германия), относящиеся к новому поколению Тур 3 OSA (Open system for automation) [1].

Станкостроители отдают предпочтение *функционально гибкой* системе управления, которая наиболее просто адаптируется к фазовым пространствам станка и рабочего процесса [2, 3]. Можно вспомнить, что на каком-то этапе развития ЧПУ многие станкостроители производили собственные системы ЧПУ, однако сегодня вполне удовлетворяются внесением в структуру математического обеспечения базовой системы своих "ОЕМ-разделов" (Own Equipment Manufacture). Наиболее полно подобная возможность заложена в систему ЧПУ "Andronic 400" фирмы Andron (Германия), а примером станкостроительной компании, использующей эту возможность в полной мере в своих гибких шлифовальных модулях, является фирма Walter (Германия). Дояполни-

тельные резервы функциональной гибкости возникают при использовании цифровых автономных следящих приводов подачи с Sercos-интерфейсом фирмы Indramat (Германия) [4]. Такие приводы, выполненные в виде шестиконтактных узлов, позволяют формировать многомерное (шести — восьмикординатное) рабочее пространство любой степени сложности (в соответствии с концепцией DISC — Digital Integrated Servo Controls).

Конечные пользователи нуждаются в *технологически гибкой* системе ЧПУ, способной учитывать специфику конкретных рабочих процессов собственного производства. Технологическая гибкость предполагает наличие базы данных (БД) для программ, подпрограмм, стандартных и измерительных циклов, параметров; возможность написания параметрических программ на высокоуровневом языке; использование развитой БД инструментов, поддерживающей помимо обычных функций еще и "менеджмент инструментов"; возможность реконфигурации интерфейса пользователя, параллельного воспроизведения нескольких управляющих программ, инсталлирования и использования технологически ориентированных резидентных САМ-систем [5]. Некоторые из наиболее "продвинутых" (advanced) современных систем ЧПУ такими возможностями в той или иной степени располагают.

Обозначенные свойства гибкости необходимы, но недостаточны для построения концепции системы ЧПУ с открытой архитектурой. К другим важным признакам такой системы относятся стандартизация модульной архитектуры и базовых внутренних структур данных и определение модулей в виде гибких командных процессоров. Обе проблемы рассмотрены далее более подробно.

Модульная архитектура и выделение базовых внутренних структур данных (форматов)

Независимо от аппаратной платформы и системной поддержки модульная архитектура может быть представлена как совокупность шести модулей, показанных на рис. 1. Все модули в значительной мере выстроены на базе покупных программно-аппаратных средств; однако используют и мощную программную проблемно ориентированную компоненту ЧПУ.

Терминал на базе персонального компьютера является признаком системы ЧПУ класса PCNC [6, 7]. Его назначение состоит в разнообразном по форме составлении задания и отображении текущего статуса системы. Кроме того, терминал поддерживает работу БД, а также используется в качестве инструментальной системы. Наряду с этим он обеспечивает внешний интерфейс через *Моноканал 1* и в этой связи может выполнять функции и клиента, и сервера.

В соответствии со своим наименованием *Интерпретатор* интерпретирует задание и осуществляет функции расчета сложных видов эквидистантной коррекции и расчета коэффициентов сплайна. Наиболее эф-

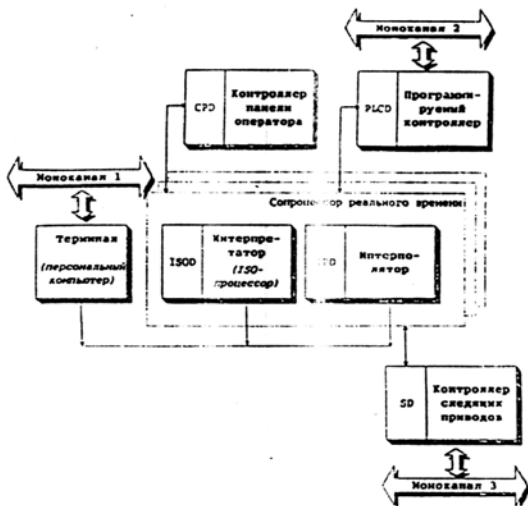


Рис. 1. Модульная архитектура открытой системы ЧПУ

фективный способ реализации интерпретатора — его построение по типу ISO-процессора [8], происхождение наименования которого связано с языком ISO-7bit управляющих программ соответственно стандартам ISO 6983, DIN 66025. Входным форматом интерпретатора (при любом способе подготовки задания) служит формат ISOD (ISO-Data), предусматривающий возможность использования любой версии кода ISO-7bit.

Современный *Интерполятор* включает в себя до шести и более различных алгоритмов интерполяции, оптимизированных соответственно тому или иному варианту входного формата IPD (InterPolator-Data). Выбираемые форматы определяются способом представления данных в управляющей программе, запросами терминала, характером траектории.

Программируемый контроллер может быть как внешней (приобретаемой по выбору станкостроителя) машиной, так и внутренним устройством, встроенным в систему ЧПУ. Для второго варианта могут быть определены (и стандартизованы) входные форматы PLCD (Programmable Logic Controller Data), отображающие имена вызываемых циклов электроавтоматики и другую информацию. Встроенный программируемый контроллер связан с электроавтоматикой станка с помощью *Моноканала* (см. рис. 1), наиболее разумным решением для которого являются стандартные средства Profibus [9].

Контроллер следящих приводов представляет собой упоминавшийся Sercos-интерфейс — ведущий (Master) в оптоволоконной кольцевой сети (*Моноканал 3* на рис. 1). Ведомыми узлами сети (Slaves) являются следящие приводы подачи и главного движения, общее число которых обычно не превышает 8. Сеть работает синхронно с повторяющимися циклами: управлять пятью приводами можно на частоте 1000 Гц, а восьмью приводами — на частоте 500 Гц.

В каждом цикле каждому приводу могут быть переданы заданные скорость подачи и перемещение, а также предельное значение крутящего момента (входные форматы — SD, Sercos-Data); в каждом же цикле от каждого привода собирается информация об истинных значениях скорости, перемещения и момента. На синхронно передаваемую информацию "налагается" асин-

хронная (по запросу ведущего), в качестве которой выступают следующие сообщения: диагностические, на экран дисплея, с указанием значений параметров. На рис. 2 показаны условные три фазы работы сети Sercos в цикле 0,001 с. В фазе *a* ведущий модуль посылает синхросообщение, которое подготавливает ведомые узлы к выдаче собственных сообщений. В фазах *b*, *в* кольцевой контур последовательно размыкается в каждом узле #1—#5 и сообщение соответствующего автономного привода отправляется ведущему, причем каждому узлу выделен свой временной интервал. В фазе *г* сообщение ведущего одновременно поступает ко всем узлам. Все сообщения вместе составляют единый кадр, соответствующий протоколу HDLC.

Входными форматами данных для *контроллера панели оператора* (см. рис. 1) служат CPD — Control Panel Data.

Концепция ISO-процессора

Суть концепции ISO-процессора состоит в использовании *двух принципов*, организации программно-аппаратной среды для подготовки задания к исполнению. Согласно первому принципу ISO-процессор построен таким образом, чтобы он воспринимал операторы языка ISO-7bit как если бы они были машинными инструкциями. Другой принцип заключается в глобальном использовании объектно ориентированного программирования, которое позволяет выстроить ISO-процессор в виде гибкой, перенастраиваемой, синтезируемой структуры. Такая структура представляет собой многопроцессорную систему, реализованную аппаратно, программно или с применением комбинированных решений. Все три процессора (интерпретатор, межмашинный корректор, диспетчер) работают параллельно.

Интерпретация состоит в предварительной подготовке (в реальном времени) кадров управляющей программы: причем эта подготовка тем сложнее, чем шире набор используемых подготовительных функций. Интерпретация включает в себя также предваритель-

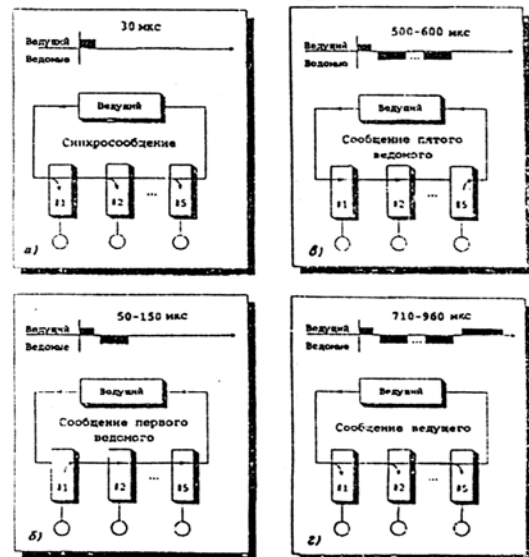


Рис. 2. Фазы работы сети Sercos

ый расчет коэффициентов сплайна. Интерпретатор обрабатывает кадр из очереди входного потока и переставляет готовый кадр в очередь для интерполяции либо для смежной коррекции. Необходимость в последней возникает при расчете эквидистантных контуров и заключается в коррекции и частичном "подавлении" существующих кадров (например, при фрезеровании внутренних контуров) или в синтезе дополнительных кадров (например, при фрезеровании наружных контуров).

Нами предложен "контурный подход" при эквидистантной коррекции, предполагающий выделение полного (из нескольких кадров) корректируемого контура. В задачи ISO-процессора входят отслеживание корректности обработки и выявление необходимости подавления любого элемента контура на всем его протяжении. Число кадров, образующих контур, определяет дополнительное число буферизованных кадров, к которым должны быть добавлены дополнительно отрендерированные кадры. Буферные кадры организованы в виде кольцевого стека, работающего по принципу FIFO (First in First Out). При подаче кадра в стек проводится эквидистантная коррекция смежных кадров, при выталкивании их из стека осуществляется проверка на необходимость подавления элементов контура.

Принципиально для ISO-процессора отношение к информации кадра управляющей программы. Кадр управляющей программы на языке ISO-7bit содержит информацию о заявленных алгоритмах и структурах данных. Алгоритмы представлены подготовительными G- и вспомогательными M-функциями; все же остальные функции образуют структуры данных (т. е. параметры G- и M-функций). Таким образом, G- и M-функции можно рассматривать как инструкции системы команд ISO-процессора. Поскольку M-функции относятся к логической задаче управления электроавтоматикой, ISO-процессор формирует лишь M-вектор активных функций, который поступает затем на вход программируемого контроллера в формате PLCD.

Все команды разбиты на несколько групп, определяющих их функциональное назначение. Установили такие наименования и назначения групп: Condition — организация перехода к следующему кадру; Plane — выбор координатной плоскости и переход к относительной системе координат; Dimension — преобразование размерностей к форме, используемой в алгоритмах интерполяции; Correction — расчет эквидистантной траектории; Acceleration — расчеты траекторий на участках разгонов и торможений; Time-Delay — задание выдержки времени; Interpolation — выбор алгоритмов интерполяции.

Активные в данный момент подготовительные функции (команды) образуют G-вектор, размерность которого определяется числом подгрупп ортогональных (т. е. взаимоисключающих) команд, а следовательно, — версией входного языка системы ЧПУ. Структура системы команд предопределяет структуру процессора интерпретации.

Особенность нашего подхода состоит в том, что функциональное наполнение алгоритмов связано с конкретными подготовительными G-функциями, т. е. сама структура системы команд отображается в ресурсах системы управления. Такая схема обеспечивает гибкость, поскольку функциональное назначение и возможности системы определяются множеством отобранных подготовительных функций. Использование подобной схемы означает, что одна и та же система ЧПУ может работать с различными системами команд.

В соответствии с объектно ориентированным подходом каждой команде ISO-процессора сопоставлен свой объект. В основе иерархии объектов лежит TObject — абстрактный тип, позволяющий порождать объекты и создавать все производные от него объекты в коллекции с тем, чтобы проводить над ними операции как с элементами очереди. Поскольку все инструкции кода ISO-7bit подразделяются на G-функции и их параметры, строятся соответствующие частные объекты. Конкретные экземпляры создаются при каждом появлении конкретных функций (перемещений, подачи и пр.) во входном потоке кадров управляющей программы. Обращение к алгоритмическим процессам с помощью виртуальных методов делает систему команд независимой от реализации процессов в системе.

Встроенные "открытого" интерполятора

Открытый интерполятор допускает свободное написание алгоритмов интерполяции и произвольную их комбинацию при воспроизведении сложных траекторий в многокоординатном пространстве (в том числе с использованием сплайнов). Ключевой момент в построении открытого интерполятора — удачный выбор входных форматов. Далее приведен фрагмент формального описания [10] одной из возможных версий входного формата, в составе которого управляющие структуры — заголовки (HeaderLine) и данные (DataLine):

<заголовок> → <старт интерполяции, подача задана явно> | <старт интерполяции, подача задана неявно> | <вызов подпрограммы> | <конец программы> | <комбинация алгоритмов> | <переопределение скорости подачи> | <относительный номер кадра>;
 <старт интерполяции, подача задана явно> → HL_101 ((идентификатор осей), <подача в мм/мин>;
 <старт интерполяции, подача задана неявно> → HL_102 ((идентификаторы осей), <индекс подачи>;
 <вызов подпрограммы> → HL_104 ((размер подпрограммы в байтах));
 <конец программы> → HL_End;
 <комбинация алгоритмов> → HL_00 ((маска интерполяции));
 <маска интерполяции, 16-разрядное слово> → <признак линейной> | <признак круговой> | <признак сплайновой> | <признак полиномиальной> | <признак алгоритма Безье> | <признак алгоритма Рибенкова> | <резервные биты>;
 <переопределение скорости подачи> → HL_01 ((новая подача));
 <данные> → <линейная интерполяция> | <сплайновая интерполяция, заданы углы входа и выхода> | <сплайновая интерполяция, задан угол выхода> | <круговая интерполяция> | <переопределение плоскости> | <переопределение осей>;
 <линейная интерполяция> → DL_06 ((относительные координаты прямой));
 <сплайновая интерполяция, заданы углы входа и выхода> → DL_07 ((относительные координаты точек), <углы входа-выхода>);
 <сплайновая интерполяция, задан угол выхода> → DL_08 ((относительные координаты точек), <угол выхода>);
 <круговая интерполяция> → DL_11 ((относительные координаты дуг окружности));
 <переопределение плоскости> → DL_16 ((код плоскости));
 <переопределение осей> → DL_17 ((коды осей)).

При выборе форматов определяющими факторами были: соответствие любым инструкциям кода ISO-7bit; отсутствие дублирования определений, неизменных в смежных кадрах.

Заголовок HL_101 инициирует старт при начале интерполяции или переключении быстрого хода. При этом включаются режимы разгона или торможения. Переопределение осей HL_01 проводится, если задействованные в текущем кадре оси интерполяции отличаются от ранее установленных. Каждый кадр генерирует

HL_n05 со своим относительным номером. Кадры, не содержащие информации для интерполятора, только увеличивают относительный номер следующего кадра. Признак "1" в заголовке означает прерывание интерполяции, а признак "л" — отсутствие прерывания.

В таблице показаны примеры исходных текстов в коде ISO-7bit и соответствующие IPD-форматы на входе в интерполятор.

Результаты работы ISO-процессора по формированию IPD-форматов

№	Программа	IPD-форматы
1	N10 % CNC — Test 1	HL_n 05(0)
	N20 * comment	
	N30 G90	
	N40 G00 X20 Y20	HL_n 05(3) HL102(10,0) DL_06(20,20)
	N50 G01 X50 Y50	HL_n 05(1) HL102(10,1) DL_06(30,30)
	N60 X100	HL_n 05(1) DL_17(8) DL_06(50)
	N70 Y100	HL_05(1) DL_17(2) DL_06(50)
	N80 M30	HL_n 05(1) HL_End
2	N10 % CNC — Test 2	HL_n 05(0)
	N20 * comment	
	N30 G91 G00 X20 Y20	HL_n 05(2) HL102(10,0) DL_06(20,20)
	N40 G01 X30 Y10 F2000	HL_n 05(1) HL_101(10,2000)
		DL_06(50,50)
	N50 X 100	HL_n 05(1) DL_17(8) DL_06(100)
	N60 Y100	HL_n 05(1) DL_17(2) DL_06(100)
	N70 G02 X10Y10 I10 J0 F2500	HL_n 05(1) HL_n01(2500) DL_17(10)
N80 M30	HL_n 05(1) HL_End	
3	N10 % CNC — Test 3	HL_n 05(0)
	N20 * comment	
	N30 G91	
	N40 G00 X30 Y30	HL_n 05(3) HL_102(10,0)
		DL_06 (30,30)
	N50 G300 G01 Z50	HL_n 05(1) HL_101(12,2500)
		HL_n 00(3)
	G02 X20 Y20 I0 J20 F2500	DL_17(4) DL_06(50) DL_17(10)
	DL_11(...)	
N60 G01 X100	HL_n 05(1) DL_17(8) DL_06 (100)	
N70 Y100	HL_n 05(1) DL_17(2) DL_06(100)	
N80 M30	HL_n 05(1) HL_End	

Примечание. G300 — вызов комбинированной интерполяции.

На рис. 3 показан пример комбинированного использования алгоритмов при вырезании проволокой сложного фасонного паза на электроэрозионном станке (комбинация линейной, круговой и сплайновой интерполяций).

Выводы

Различные системы ЧПУ различаются дизайном, эргономическими подходами, аппаратной платформой,

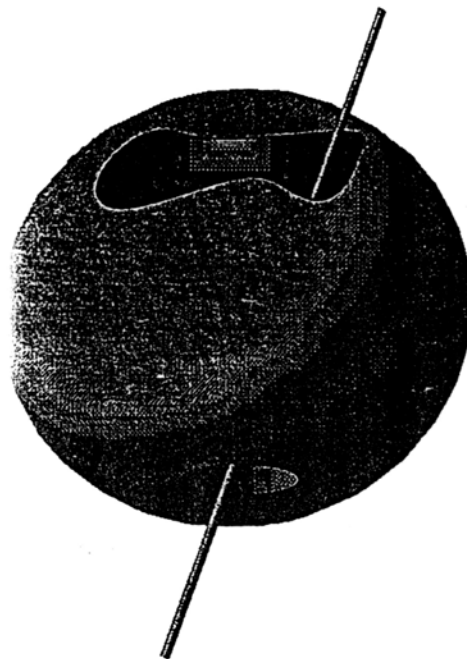


Рис. 3. Использование комбинированной интерполяции при обработке фасонного паза на электроэрозионном станке

операционными средствами, БД, встроенными инструментальными средствами. При этом, однако, было бы разумным стандартизировать модульную структуру, внутренние форматы данных и обеспечить на этой основе открытую архитектуру системы управления. В результате можно ожидать одновременного повышения гибкости системы и снижения ее стоимости.

Список литературы

1. Pritschow G. Automation technology on the way to an open system architecture // Robotics and Computer-Integrated Manufacturing. 1990. Vol. 7. № 1/2.
2. Сосонкин В. Л. Персональный компьютер как архитектурный компонент "персональной системы управления" // Станки и инструмент. 1993. № 5.
3. Сосонкин В. Л. Концепция персональных систем управления в реальном времени // Приборы и системы управления. 1995. № 7.
4. Kiel E., Schierenberg O. Sercos-Interface-Controller Electronics. 1992. № 2.
5. Сосонкин В. Л. Взгляд на предстоящую эволюцию устройств ЧПУ // Станки и инструмент. 1992. № 9.
6. Сосонкин В. Л. Концепция системы ЧПУ на основе персонального компьютера // Там же. 1990. № 11.
7. Сосонкин В. Л., Мартинов Г. М. Концепция геометрического ISO-процессора для систем ЧПУ // Там же. 1994. № 7.
8. Open architecture extends capability // Metallworking Production. 1994, April.
9. Ein Profi in Action // Production. 1994. № 1/2.
10. Сосонкин В. Л. Программное управление технологическим оборудованием: Учебник для вузов. М.: Машиностроение, 1991.