

41. Дисперсионная идентификация / Под ред. Н. С. Райбма-на. М.: Наука, 1981.
42. Renyi A. On measures of dependence // Acta Math. Hung. 1959. Vol. 10. № 3—4.
43. Сарманов О. В. Собственные корреляционные функции и их применение в теории стационарных марковских процессов // Доклады АН СССР. 1960. Т. 132. № 4.

44. Сарманов О. В., Братосла З. Н. Вероятностные свойства билинейных разложений по полиномам Эрмита // Теория вероятностей и ее применения. 1967. Т. XII. Вып. 3.
45. Пашенко Ф. Ф., Чернышев К. Р. Применение метода функциональных преобразований и идентификации нелинейных систем // Автоматика и телемеханика. 1992. № 4.
46. Сарманов О. В. Максимальные коэффициенты множественной корреляции // Доклады АН СССР. 1960. Т. 130. № 2.

УДК 681.3

В. Л. СОСОНКИН, д-р техн. наук, проф., Г. М. МАРТИНОВ, канд. техн. наук

## 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 Тип 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). Дополни-

тельные резервы функциональной гибкости возникают при использовании цифровых автономных следящих приводов подачи с Sercos-интерфейсом фирмы Indramat (Германия) [4]. Такие приводы, выполненные в виде мехатронных узлов, позволяют формировать многомерное (шести — восьмикоординатное) рабочее пространство любой степени сложности (в соответствии о 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-



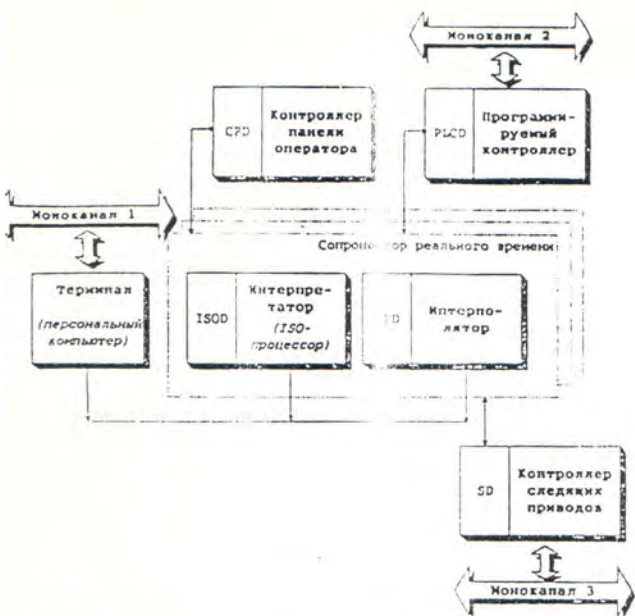


рис. 1. Модульная архитектура открытой системы ЧПУ

активный 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 десятки 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, запроса 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 мена 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*, *c* 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-

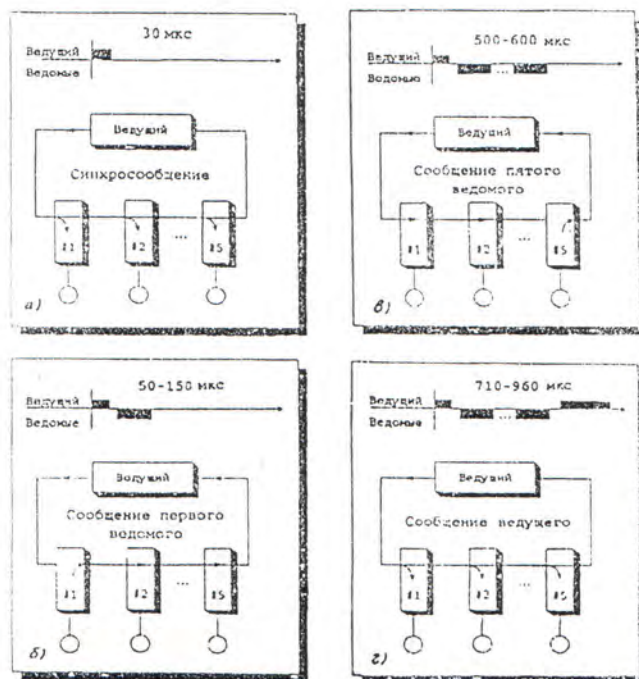


рис. 2. Фазы работы сети Sercos



ли расчет коэффициентов сплайна. Интерпретатор обрабатывает кадр из очереди входного потока и пересылает готовый кадр в очередь для интерполяции либо в 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 подго- (из нескольких кадров) корректируемого контура. задачи ISO-процессора входят отслеживание коррек-ности обработки и выявление необходимости подав-ления любого элемента контура на всем его протяже-нии. Число кадров, образующих контур, определяет до-лнительное число буферизованных кадров, к ко-рым должны быть добавлены дополнительно отге-рированные кадры. Буферные кадры организованы в де кольцевого стека, работающего по принципу FO (First in First Out). При поминании кадра в стек оводится эквидистантная коррекция смежных кадров, при выталкивании их из стека осуществляется проверка на необходимость подавления элементов кон-тра.

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

Все команды разбиты на несколько групп, определя-ющих их функциональное назначение. Установили та-кие наименования и назначения групп: Condition — анализация перехода к следующему кадру; Plane — выбор координатной плоскости и переход к относи-тельной системе координат; Dimension — преобразова-ние размерностей к форме, используемой в алгоритмах интерполяции; Cogection — расчет эквидистантной траектории; Acceleration — расчеты траекторий на участках разгонов и торможений, Time-Delay — зада-ние выдержки времени; Interpolation — выбор алгорит-ма интерполяции.

Активные в данный момент подготовительные функ-ции (команды) образуют G-вектор, размерность ко-торого определяется числом подгрупп ортогональных (взаимоисключающих) команд, а следовательно, — числом входного языка системы ЧПУ. Структура сис-темы команд предопределяет структуру процессора ин-терпретации.

Особенность нашего подхода состоит в том, что фун-циональное наполнение алгоритмов связано с кон-кретными подготовительными G-функциями, т. е. сама структура системы команд отображается в ресурсах системы управления. Такая схема обеспечивает гиб-кость, поскольку функциональное назначение и воз-можности системы определяются множеством отобра-женных подготовительных функций. Использование по-добной схемы означает, что одна и та же система ЧПУ может работать с различными системами команд.

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

#### строение "открытого" интерполятора

Открытый интерполятор допускает свободное на-равнение алгоритмов интерполяции и произвольную их комбинацию при воспроизведении сложных траек-терий в многокоординатном пространстве (в том числе с использованием сплайнов). Ключевой момент в строении открытого интерполятора — удачный вы-бор входных форматов. Далее приведен фрагмент фор-мального описания [10] одной из возможных версий одного формата, в составе которого управляющие структуры — заголовки (HeaderLine) и данные (Data-Data):

(заголовок) → (старт интерполяции, подача задана явно) | (старт интерполяции, подача задана неявно) | (вызов подпрограммы) | (ко-д программы) | (комбинация алгоритмов) | (переопределение скорости подачи) | (относительный номер кадра);

(старт интерполяции, подача задана явно) → HL\_101 ((идентифи-кор осей), (подача в мм/мин);

(старт интерполяции, подача задана неявно) → HL\_102 ((идентифи-кор осей), (индекс подачи));

(вызов подпрограммы) → HL\_104 ((размер подпрограммы в бай-т));

(конец программы) → HL\_End;

(комбинация алгоритмов) → HL\_n00 ((маска интерполяции));

(маска интерполяции, 16-разрядное слово) → (признак линей-ной) (признак круговой) (признак сплайновой) (признак полиноми-альной) (признак алгоритма Безье) (признак алгоритма Рябенкова) (верные биты);

(переопределение скорости подачи) → HL\_n01 ((новая подача));

(данные) → (линейная интерполяция) | (сплайновая интерполя-ция, заданы углы входа и выхода) | (сплайновая интерполяция, задан-ый угол выхода) | (круговая интерполяция) | (переопределение плос-кости) | (переопределение осей);

(линейная интерполяция) → DL\_06 ((относительные координаты плоской));

(сплайновая интерполяция, заданы углы входа и выхода) → DL\_07 ((относительные координаты точек), (углы входа-выхода));

(сплайновая интерполяция, задан угол выхода) → DL\_08 ((отно-сительные координаты точек), (угол выхода));

(круговая интерполяция) → DL\_11 ((относительные координаты д-окружности));

(переопределение плоскости) → DL\_16 ((код плоскости));

(переопределение осей) → DL\_17 ((коды осей)).

При выборе форматов определяющими факторами

являются: соответствие любым инструкциям кода ISO-7bit;

отсутствие дублирования определений, неизменных в смежных кадрах.

Заголовок HL\_101 инициирует старт при начале ин-терполяции или переключении быстрого хода. При этом включаются режимы разгона или торможения.

Переопределение осей HL\_n01 проводится, если задей-ствованные в текущем кадре оси интерполяции отлича-ются от ранее установленных. Каждый кадр генерирует



н05 со своим относительным номером. Кадры, не несущие информации для интерполятора, только указывают относительный номер следующего кадра. Знак "1" в заголовке означает прерывание интерполяции, а признак "л" — отсутствие прерывания. В таблице показаны примеры исходных текстов в формате ISO-7bit и соответствующие IPD-форматы на входе в интерполятор.

Результаты работы ISO-процессора при форматировании IPD-форматов

Программа	IPD-форматы
10 % CNC — Test 1 20 * comment 30 G90 40 G00 X20 Y20 50 G01 X50 Y50 60 X100 70 Y100 80 M30	HL_n 05(0)  HL_n 05(3) HL102(10,0) DL_06(20,20) HL_n 05(1) HL102(10,1) DL_06(30,30) HL_n 05(1) DL_17(8) DL_06(50) HL_05(1) DL_17(2) DL_06(50) HL_n 05(1) HL_End
110 % CNC — Test 2 120 * comment 130 G91 G00 X20 Y20 140 G01 X30 Y10 F1000 150 X 100 160 Y100 170 G02 X10Y10 I10 J0 180 M30	HL_n 05(0)  HL_n 05(2) HL102(10,0) DL_06(20,20) HL_n 05(1) HL_101(10,2000)  DL_06(50,50) HL_n 05(1) DL_17(8) DL_06(100) HL_n 05(1) DL_17(2) DL_06(100) HL_n 05(1) HL_n01(2500) DL_17(10)  DL_11(...) HL_n 05(1) HL_End
210 % CNC — Test 3 220 * comment 230 G91 240 G00 X30 Y30 250 G300 G01 Z50 260 X20 Y20 I0 J20 270 F2500 280 G01 X100 290 Y100 300 M30	HL_n 05(0)  HL_n 05(3) HL_192(10,0) DL_06 (30,30) HL_n 05(1) HL_101(12,2500) HL_n 00(3) DL_17(4) DL_06(50) DL_17(10)  DL_11(...) HL_n 05(1) DL_17(8) DL_06 (100) HL_n 05(1) DL_17(2) DL_06(100) 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 интерполяции).

Выводы

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

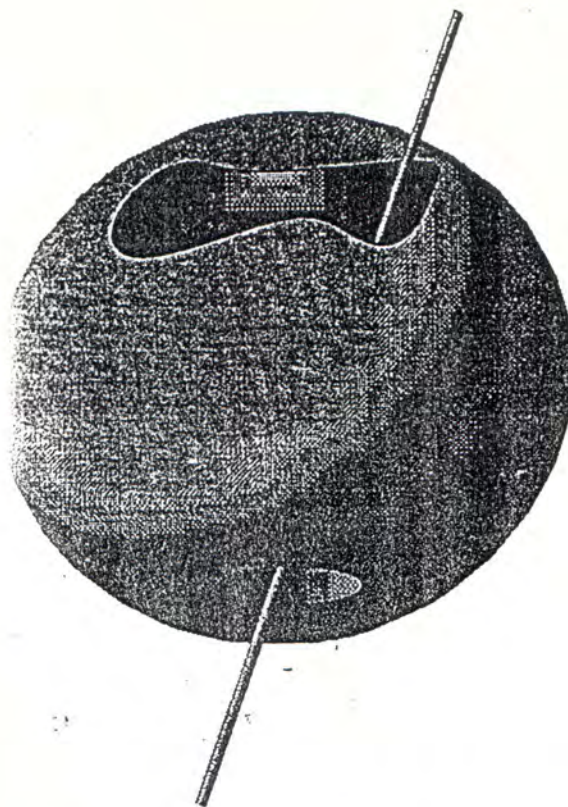


Рис. 3. Использование комбинированной интерполяции при обработке фасонного паза на электроэрозионном станке

операционными средами, 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.