

# Некоторые постпроцессорные премудрости

Уже не раз ваш покорный слуга и мои коллеги из отдела производственного инжиниринга компании ЗАО "СиСофт" рассказывали на страницах журнала о различного рода внедрениях многоосевого оборудования, о решении сложных и нестандартных технологических задач для различных предприятий<sup>1</sup>.

В большинстве своем разговор шел о работе на современных станках, оснащенных, соответственно, современными стойками ЧПУ. Такого рода оборудование отличается тем, что функционал, заложенный в системы ЧПУ данных станков, сам решает многие задачи, которые еще совсем недавно "лежали на плечах" либо САМ-систем, либо (в большинстве случаев) постпроцессоров. Речь, прежде всего, идет о таких проблемах, как ос (поворот) программного нуля, учет вылета инструмента при фиксированной или непрерывной пятиосевой обработке и т.п. Функционал систем ЧПУ динамично развивается, позволяя решать все более широкий круг задач.

Но мы с вами знаем, что любая, даже очень дорогая вещь может приносить определенные неудобства, а иной раз и иметь существенные недостатки. Так, например, даже очень дорогой автомобиль может доставить много хлопот и его владельцу, и окружающим (повышенная вероятность угона, дороговизна даже мелкого ремонта и т.п.)

То же самое происходит и в мире станков, программ и постпроцессоров: любая САМ-система, любая стойка станка с ЧПУ также не лишена тех или иных недостатков. Вот о некоторых примерах их устранения мы и

поговорим. Естественно, все приведенные ниже случаи взяты из нашей практики.

Помимо всего прочего, хотелось бы развеять бытующий в кругу некоторых людей, работающих в сфере машиностроения, миф о том, что постпроцессор — "это такая программка, которая не требует дополнительного внимания, не должна дорого стоить и вообще должна поставляться в комплекте с оборудованием или же шаблоны постпроцессоров должны прилагаться к САМ-системе".

Для начала хотел бы напомнить, что отдел производственного инжиниринга компании ЗАО "СиСофт" уже много лет работает с одной из распространенных систем высокого уровня CAD/CAM/CAE. О том, какая это система, вы уже знаете из предыдущих публикаций. Пусть вас, уважаемые читатели, не удивляют столь туманные формулировки: все дело в том, что правообладатель бренда данного ПО ввел достаточно жесткие правила на его упоминание, поэтому согласование публикации заняло бы довольно много времени.

Помимо этого нашим отделом внедряется и используется VERICUT — программный комплекс для визуализации процесса обработки деталей на станках с ЧПУ, проверки и оптимизации управляющих программ в G- и APT-форматах. Самая главная задача этого ПО — выявление и предотвращение возможных столкновений рабочих органов станка еще до процесса обработки.

О работе в среде данных программных комплексов здесь и пойдет речь. Основное внимание мы уделим постпроцессорам.

Прежде всего остановимся на работе с несложным 3-, а иногда и 2,5-осевым оборудованием. Простота станков не гарантирует отсутствия некоторых аспектов, существенно усложняющих использование "шаблонных" постпроцессоров.

Несмотря на все возрастающую сложность технического оснащения, на многих предприятиях, заводах и фабриках нашей страны по-прежнему используется, мягко говоря, далеко не современное оборудование. В частности — станки с устаревшими стойками ЧПУ, модернизированные станки. И вот тут-то в основном и возникают разного рода проблемы, доставляющие головную боль разработчикам постпроцессоров.

Например, у нас очень распространены станки с устаревшей системой управления H33. И некоторые предприятия, закупающие новое оборудование, одновременно продолжая эксплуатировать имеющееся, при внедрении САМ-систем вполне логично желают применять возможности ПО и для работы с ним. Одно из московских предприятий оказалось в схожей ситуации, и мы помимо нового оборудования не обошли вниманием и старые станки с системой ЧПУ H33.

Многие, кто знаком с этой системой ЧПУ, знают о ее особенностях. Среди них — программирование *только* в относительном (инкрементальном) режиме. Еще одна особенность — координаты задаются не в миллиметрах, как это принято на подавляющем большинстве систем ЧПУ, а в десятках микрометров с указанием предшествующих нулей. Например, при перемещении на 70 мм по X в

<sup>1</sup>"Простые вещи — непростые решения" (CADmaster № 2/2010, с. 26-35); "Unigraphics, VERICUT и станки Hermle" (CADmaster № 2/2008, с. 22-27); "NX + VERICUT в ОАО "ОКБМ Африкантов" при изготовлении импеллеров" (CADmaster, № 2/2010, с. 36-40); "Unigraphics & Renishaw" (CADmaster, № 4/2008, с. 36-38).



Рис. 1

УП должен быть указан кадр G01X+007000E.. Подобные характеристики получаемых УП настраиваются в постпроцессорах довольно легко, не исключение — и постпроцессор, разработанный нами для данной стойки ЧПУ. Гораздо более интересен формат задания подачи F.

Скорость подачи задается словом, содержащим адрес F и следующую за ним функцию подачи, состоящую из четырех цифр (обозначим их f1; f2; f3; f4). Код из трех цифр f2; f3; f4 используется для задания величины подачи. Третья цифра f2 может принимать значения 4,5 или 6 — это некий коэффициент подачи, который указывает, на сколько нужно умножить следующие две цифры f3 и f4, чтобы получить нужную подачу.

Значение 4 соответствует коэффициенту 0.1, 5 — 1, 6 — 10. Первая цифра f1 — режим изменения скорости подачи. Таких режимов два: нормальный (задается цифрой "0") и с торможением до фиксированной скорости (цифра "4"). Немного поясню. Например, для обработки с подачей 500 мм/мин. (без торможения) в УП следует задавать F0650, если же требуется обработка с подачей 50 мм/мин. — F0550 и т.д.

Мы же оперируем в любой САМ-системе минутной подачей. И потому для данной стойки ЧПУ необходимо преобразование в такой вот своеобразный формат. Для меня очевидно, что любой человек, мало-мальски связанный с программированием, скажет, что такая задача элементарна. Согласен. Но ее реализация необходима и в шаблонах постпроцессоров, которые имеют в своем составе специальное ПО для создания и редактирования постпроцессоров.

А там такого формата нет.

Я уж не углубляюсь в проблему первого параметра в подаче. Напомню: это режим изменения подачи — с торможением или без. Нет сомнения, что можно целый научный трактат написать только на основе методики расчета этого параметра. И если одно из условий, при котором следует включать торможение, анализировать довольно легко (речь идет о каждом уменьшении подачи от предыдущего значения), то второй аспект не так прост. Я имею в виду, что при выводе каждого кадра УП нужно анализировать направление движения, а точнее — изменение этого направления. И согласно некоторым критериям, которые, скорее всего, связаны с кинематическими характеристиками конкретного станка, определять: требуется торможение при данном изменении вектора направления движения инструмента или нет.

Вот вам и простенький станочек! Одной научной деятельности сколько можно развести при создании постпроцессора! Более того, ситуация складывается таким образом, что чем старше стойка ЧПУ, чем она дешевле (различного рода урезанные варианты), тем больше потенциальных проблем возникает при создании постпроцессоров.

И еще один пример из нашей практики. У одного из заказчиков имелся в наличии простой станок "Микрос" с одноименной стойкой ЧПУ. Данная система управления, по большому счету, аналогична системам управления типа Fanuc. И, как и в предыдущем случае, одно из ограничений данной стойки потребовало дополнительной работы над постпроцессором, а именно: количество ка-

дров в УП не может превышать 32 000. Очевидно, что анализировать это количество при постпроцессировании — довольно просто. И в принципе можно было бы так и поступить, с одним лишь добавлением: при превышении критического количества кадров УП выдавать соответствующее предупреждающее сообщение. Но для многих из вас очевидно и то, что это отнюдь не оптимальное решение проблемы. Поскольку тогда следует предпринимать шаги в самой САМ-системе по формированию операций с потенциально нужным количеством кадров полученной УП, что может представлять из себя также непростую задачу...

Гораздо более правильным и наиболее приемлемым решением проблемы является автоматическое разбиение программы на несколько по достижении лимита количества кадров УП. Сама по



Рис. 2

себе эта задача опять же совсем не сложна для реализации с точки зрения программирования: нужно всего лишь запомнить координаты XYZ на граничном условии количества кадров, добавить (на заранее оговоренную величину) отскок инструмента, запомнить название текущей операции, номер инструмента и т.п., ну и перенаправить вывод данных в другой файл. И все данные повторить во вновь созданной программе. И так до тех пор, пока все переходы не будут обработаны до конца, сколько бы выходных файлов не получилось. Поскольку система ЧПУ станка подобна Fanuc, то допускаются числовые названия файлов, что упрощает автоматическое переименовывание дополнительно создаваемых файлов УП.

В данном случае мы опять же имеем простой станок, но с ограничениями, которые добавляют работы по созданию (редактированию) постпроцессора.

Следующий пример — уже с оборудованием посложнее.

Многим из вас наверняка известно оборудование под торговой маркой HAAS. Это американские станки, широко распространенные в мире, в том числе и в нашей стране. В России HAAS представлен в основном в виде 3-осевых фрезерных (рис. 1), токарных и токарно-фрезерных станков. Но в последнее время на некоторых отечественных предприя-

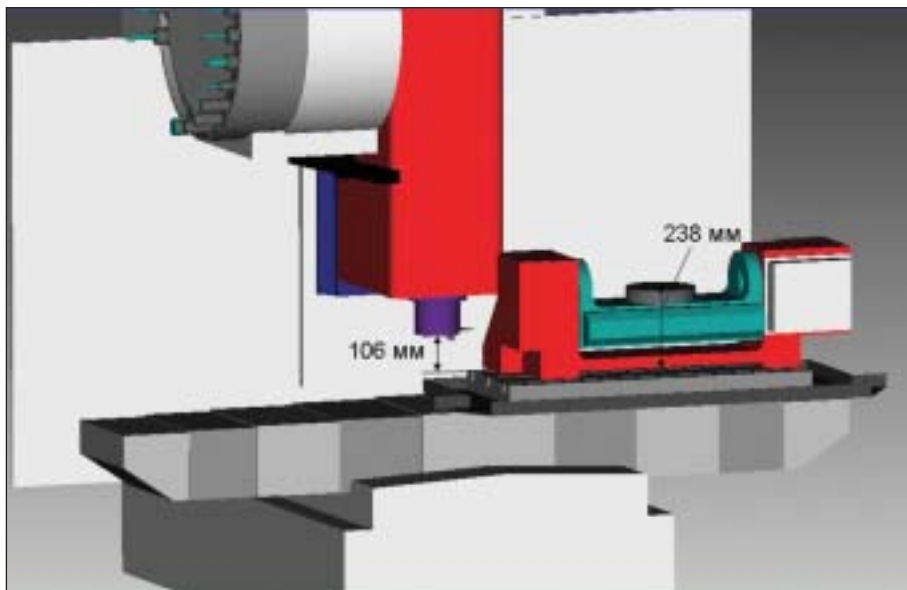


Рис. 3

тиях стали появляться 5-осевые станки, выполненные на базе 3-осевых. Добились этого очень просто: на рабочий стол вертикально-фрезерного станка дополнительно устанавливаются наклонно-поворотные столы (рис. 2). Вполне недорогое и довольно универсальное решение: я могу работать как в 3-осевом режиме, сняв поворотные столы, так и в 5-осевом, установив их.

Но, как это зачастую бывает, при этом возникают два существенных недостатка. Первый — значительное ограничение рабочей зоны станка при установке поворотных столов. Ведь если взять, к примеру, базовый станок марки VF-2, то в технических характеристиках мы имеем ход по оси Z — 508 мм. И если на такой станок установить, например, поворотные столы марки TR-160, то легко посчитать, насколько ограничивается рабочая зона. Так, расстояние от рабочего стола станка, который является основанием для установки поворотных столов, до поверхности планшайбы 5-й оси в не повернутом состоянии — 238 мм. Если учесть, что при минимальном положении шпинделя по оси Z расстояние от торца шпинделя до рабочего стола составляет ~105 (106) мм, то мы имеем чистую потерю хода станка по этой оси  $238 - 106 = 132$  мм, что составляет более четверти от общего хода (рис. 3).

И это мы еще не учитываем высоту боковых колонн поворотных столов, которые заметно выше "зеркала" планшайбы. И, как видно из рис. 3, наиболее вероятными становятся столкновения шпинделя (стойки шпинделя) именно с боковыми колоннами.

Второе ограничение связано с тем, что столы съемные. Дело в том, что для корректного программирования в 5-ти осях в специальные параметры станка вносятся

некоторые геометрические данные для 4 и 5 осей (чуть подробнее о них — ниже). Для традиционных 5-осевых станков эти параметры заносятся единожды на заводе-производителе оборудования, и они остаются неизменными (если, конечно, со станком не происходит каких-либо явлений, физически меняющих его геометрические характеристики, например, удар). На данном же станке из-за того что устройство поворотных столов каждый раз может быть установлено на рабочем столе по-разному (правее, левее на любую величину), возникает необходимость каждый раз перемерять (обкатывать) 4 и 5 оси. Частично проблему можно решить, придумав жесткое базирование устройства поворотных столов на рабочем столе (на штифты или еще как-либо). Но в той поставке оборудования нашему заказчику, с которым мы работали, ничего подобного предусмотрено не было, и организация базирования этого устройства отдавалась "на откуп" пользователю станка. На данный момент нашим заказчиком было принято решение пока не фрезеровать что-либо в устройстве столов, а перемерять нужные геометрические данные каждый раз при установке поворотных столов.

Но вернемся к основной теме — к постпроцессорам. Естественно, описанные выше проблемы отражаются и на постпроцессоре. Например, если еще раз взглянуть на рис. 3, то можно заметить, что устройство поворотных столов занимает большую часть рабочей области и в плоскости XY. И первая же автоматическая смена инструмента при установленных поворотных столах показала, что это действие весьма небезопасно с точки зрения столкновения. То есть при определенных длинах инструмента и текущем

положении стола по X и Y вероятность столкновения инструмента при смене с боковой колонной устройства поворотных столов велика. Это заставило внести в постпроцессор поправки по безопасному перемещению перед сменой инструмента — сначала максимально вверх, затем максимально вправо и от нас. На многих современных станках подобные движения выполняются автоматически при выполнении команды смены инструмента, но в данном станке эти перемещения отсутствуют.

Справедливости ради стоит отметить один из самых главных плюсов оборудования HAAS — его низкую стоимость. Собственно, это обстоятельство во многом и определило такую популярность станков данной марки не только в России, но и во всем мире. Низкая стоимость обусловлена многими факторами, но я бы хотел отметить один из них — это система ЧПУ станка. Дело в том, что стойки станков HAAS (речь идет про фрезерные станки) одинаковы для всех моделей станков. А поскольку в основе своей это станки 3-осевые, то и функционал, заложенный в стойки — минимально необходимый для них. Мы же имеем дело с фактически 5-осевым станком при установленных поворотных столах. И помимо реализации как таковой управления в системе ЧПУ 4-й и 5-й осей, для подобного рода станков необходимы и специальные возможности для корректного, удобного программирования с использованием этих осей. Речь, прежде всего, идет о таких, уже объективно ставших элементарными функциях, как динамическое смещение нулевой точки при фиксированной 5-осевой обработке и контроль кончика инструмента при 5-осевой непрерывной обработке. Как вы уже можете догадаться, ничего подобного в стойках станков HAAS нет. Во всяком случае, на том оборудовании, с которым нам пришлось работать, этого не было, и проведенные консультации со специалистами компании-поставщика этот факт только подтвердили. Соответственно, такой существенный недостаток функционала должен быть каким-либо образом компенсирован. Что и было сделано. Но обо всем по порядку.

Выше я упомянул два вида обработки, и поскольку между ними есть принципиальная разница, влияющая на программирование, остановимся на этом чуть подробнее.

При фиксированной 5-осевой (4-осевой) обработке поворотные оси (столы) поворачиваются один раз, занимая определенную позицию на детали (границы), а далее идет обычная двух- или трехкоординатная обработка. Такая обработка на сленге технологов называется 3+2 обра-



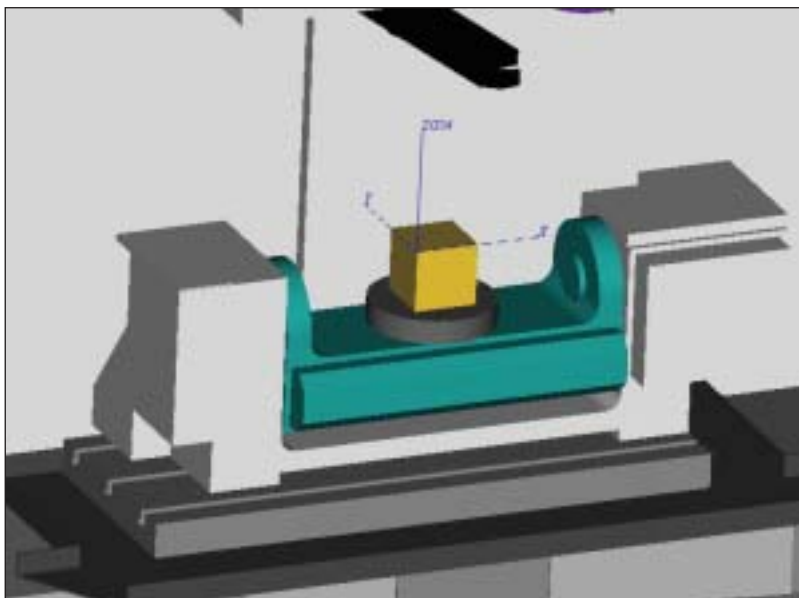


Рис. 4

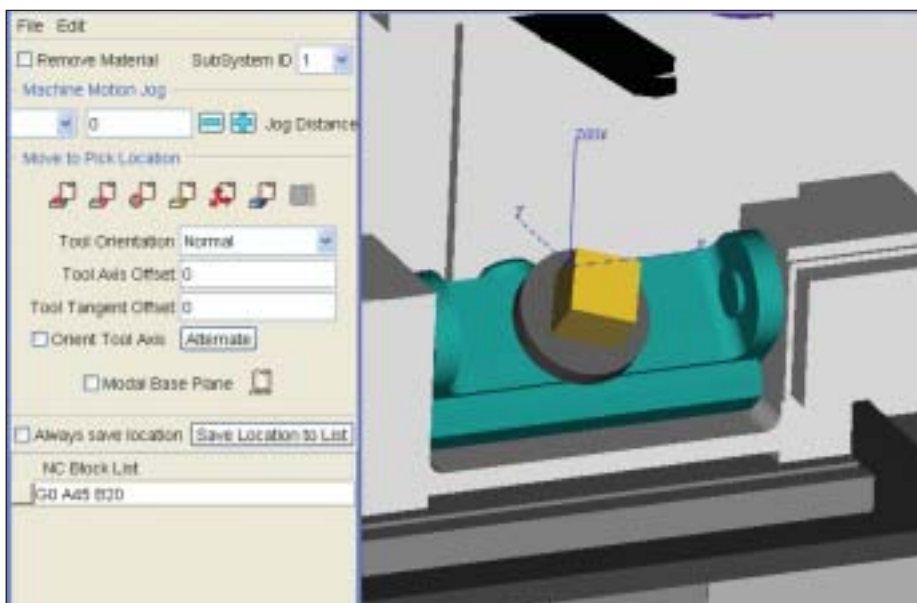


Рис. 5

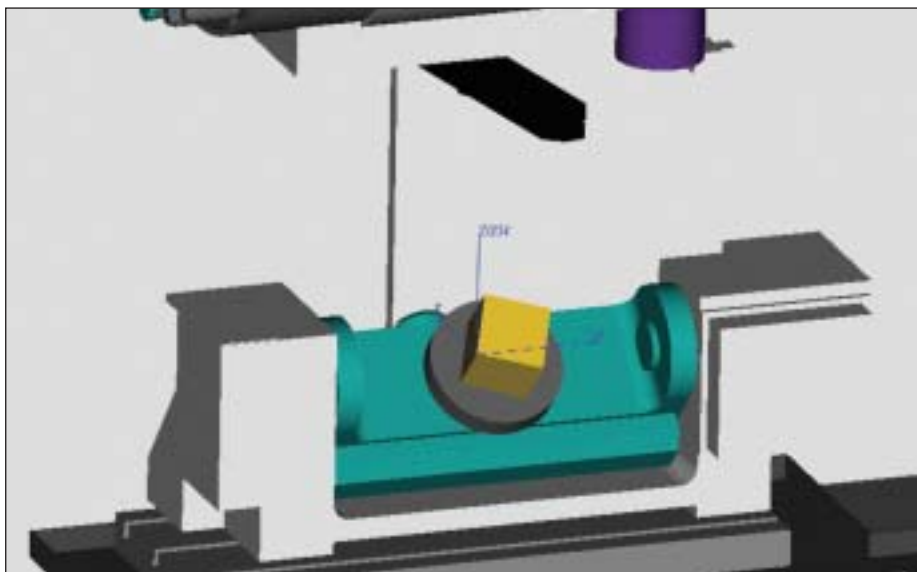


Рис. 6

ботка. Подобные операции характерны для сложных корпусных деталей, штампов или прессформ и т.п.

Идеологически программирование и постпроцессирование таких обработок построено на тезисе, что программный ноль совпадает с точкой пересечения поворотных осей. И все пересчеты при выполнении траекторий на наклонных поверхностях выполняются относительно этой точки. В общем случае это означает одно: программный ноль должен всегда располагаться на пересечении поворотных столов либо на другой характерной точке между этими осями, но при этом смещения (взаимосвязи) необходимо учесть в постпроцессоре. Например, если вернуться к рассматриваемому станку, то такой точкой может быть точка на поверхности планшайбы оси В и в центре ее поворота. На рис. 4 и 5 представлено устройство поворотных столов, где 4-я ось — это ось А (качалка зеленого цвета), вращение происходит вокруг оси Х, 5-я ось — ось В (планшайба темно-серого цвета), вращение происходит вокруг оси Z. Исходя из технических данных, ось А находится выше поверхности планшайбы на ~4 мм. Соответственно, данная величина вносится и в постпроцессоре в специальные разделы описания кинематической схемы станка.

Самый главный недостаток работы такого способа — это жесткая привязка взаимного расположения обрабатываемой заготовки и программного нуля между собой. То есть в САМ-системе программный ноль нужно располагать относительно заготовки так, как они расположены друг относительно друга на станке. И если сдвинуть заготовку, то УП для нее будет уже негодной. В этом случае нужно снова в проекте в САМ-системе сдвинуть либо программный ноль, либо заготовку и заново перепостпроцессировать УП. Частично такое неудобство устраняется путем разработки жесткого базирования заготовок на станке, то есть мероприятиями по соблюдению повторяемости установки заготовок в одно и то же положение (как правило, с помощью дополнительных приспособлений). В некоторых случаях это может потребовать существенных усилий, средств и т.п. Наш заказчик, например, занимается изготовлением различных прессформ. Номенклатура изделий большая, количество каждой позиции не носит массового или даже серийного характера, и потому городить дополнительные приспособления под каждый вид прессформ крайне невыгодно. Поэтому вышеописанный вариант программирования заказчика не устраивал.

Другой же способ работы с такими обработками гораздо более удобен и логичен. Здесь программный ноль привязан не к какой-либо точке станка, а к некой характерной точке на заготовке, где бы она при этом ни располагалась. Жестко базировать заготовку на станке не требуется. Например, если речь идет о пресс-формах, то программный ноль может располагаться на одном из уголков заготовки, а сама заготовка может крепиться в обычные тиски, установленные на планшайбе, то есть совершенно свободно (в разумных, конечно, пределах). Но в этом случае без проведения каких-либо дополнительных процедур мы также будем иметь негодную УП, поскольку (как уже отмечалось выше) постпроцессор выдает пересчитанные координаты, полагая, что программный ноль и точка пересечения поворотных осей совпадают. То есть требуется функция, которая компенсировала бы смещение программного нуля относительно 4-й и 5-й осей станка. Предположим, мы имеем заготовку в виде кубика и программный ноль на ближайшем левом углу (рис. 4.). Если задать команды поворотов 4-й и 5-й осей (рис. 5), то в общем случае повороты как таковые будут выполнены, а программный ноль останется на месте. Чтобы получить корректное перемещение по пересчитанным постпроцессором координатам при данных положениях столов, нужно, чтобы программный ноль по-прежнему находился на том же углу заготовки (рис. 6). То есть условно это выглядит так, будто поворот был выполнен относительно данного угла заготовки. Вопрос в том, кто или что должен рассчитать необходимый перенос программного нуля? Многие, кто еще не очень хорошо знаком с многоосевым оборудованием, могут на этот вопрос резонно заметить, что стойка ЧПУ сама должна рассчитывать все необходимые действия. И они правы. С одним небольшим "но".

Действительно, основные лидеры в разработке систем ЧПУ уже давно создали соответствующий функционал, результатом которого является либо специальный G-код, либо команда. Примеров тому — масса. Так, например, у одной из самых популярных стоек Siemens 840D это специальный цикл — Cycle 800; у другой, не менее популярной системы ЧПУ Heidenhain iTNC 530 (430) — это либо цикл 19, либо (более современный вариант) команда *Plane* и т.п. То есть при наличии таких команд разработчику постпроцессора остается лишь правильно организовать вывод в УП данных команд, а все необходимые пересчеты стойка станка выполнит сама. Как вы можете догадаться, у рассматриваемого нами станка HAAS подобного функционала нет. И,

как уже отмечалось выше, вариант жесткого базирования заготовок нашего заказчика крайне не удовлетворял.

Поэтому бремя выхода из ситуации падало на постпроцессор. В данном случае, правда, не на сам постпроцессор, а на разработку собственной процедуры, восполняющей недостаточность функционала имеющейся системы ЧПУ. Иными словами, речь идет о подпрограмме, в которой производятся необходимые вычисления. На самом деле эта задача не такая сложная, как может показаться на первый взгляд.

Ведь фактически все, что нам нужно, — рассчитать смещение от одной точки к другой в зависимости от углов поворотов наклонных столов. Данные углы нам известны, они формируются постпроцессором, и задача заключается лишь в том, чтобы правильно передать эти значения внутрь подпрограммы пересчета. Такой механизм программирования предусмотрен имеющейся системой ЧПУ. Речь идет про команду вызова подпрограммы с параметрами — G65.

Далее, что нужно нам знать — это геометрические характеристики 4-й и 5-й осей. Речь идет о координатах оси A (это Y и Z) и оси B (это X и Y) относительно машинного нуля. Ну и должна учитываться разница по оси Z между поверхностью планшайбы B и осью A. Эти данные заносятся в начало подпрограммы в так называемые *переменные пользователя* (то есть переменные в стойке ЧПУ, которыми можно пользоваться для своих нужд), и по сути, если устройства поворотных столов не снимать, то эти значения — константы. Но, как опять же отмечалось выше, заказчик будет практиковать частый съем и установку устройства поворотных столов, поэтому каждый раз будет вынужден перемерять эти геометрические характеристики и править значения в подпрограмме пересчета.

Напомню, что машинным нулем является некая точка в станке (а может быть и вне станка), относительно которой выполняются все перемещения и смещения до программных нулей, то есть базовая, начальная система координат станка.

Второй же точкой, до которой требуется рассчитать смещение, является наш программный ноль. В системах ЧПУ HAAS эти данные заносятся в таблицы для нулей от G54 до G59 (как в стойках "Fanuc"). То есть туда заносятся смещения от машинного нуля до требуемой точки, определенной нами как программный ноль. В стойках ЧПУ различных производителей есть возможность считывать и управлять очень широким спектром данных непосредственно во время выполнения УП. Речь идет о *системных переменных*. Описываемая нами

система ЧПУ — не исключение. И такие данные, как, например, смещения по всем координатам от машинного нуля до программного также заносятся в определенные системные переменные. Их можно считывать, передавать в другие (пользовательские) переменные и манипулировать ими на свое усмотрение.

Итак, мы имеем в абсолютных значениях (то есть относительно машинного нуля) координаты поворотных осей, координаты программного нуля и углы поворотов 4-й и 5-й осей. Затем "вступает в бой" наша с вами любимая геометрия (синусы-косинусы, тангенсы, арктангенсы и т.п.), которая и помогает рассчитать требуемые величины смещений по каждой координате. Непосредственно само смещение на рассчитанные величины производится командой G52 внутри той же подпрограммы. Эта процедура позволяет нам использовать программный ноль на любой точке заготовки, где бы она ни находилась в рабочей зоне станка. Хотелось бы отметить, что подобная ситуация с разработкой процедур пересчета положения программного нуля для нас не нова и использовалась уже неоднократно. И не всегда это было продиктовано тем, что мы имели дело с недорогим оборудованием без необходимых опций. Например, подобную подпрограмму пришлось разрабатывать для линейки Variaxis станков Mazak! Отмечу, что кинематически станок не отличается от рассматриваемого нами станка HAAS. Кроме того, хотелось бы подчеркнуть, что станки Mazak — далеко не дешевое оборудование. "В чем же дело? — спросят меня многие. — Неужели у системы ЧПУ данного оборудования (Mazatrol Matrix) нет соответствующего функционала?" Есть! Существует команда G54.2P., которая отвечает за динамический перенос программного нуля. Но есть и ряд "но"... Во-первых, эта команда опциональна, то есть приобретается за отдельные деньги. Но проблема даже не в самом факте опциональности, а в том, что на первых порах некоторые покупатели оборудования не знали о ее назначении (без должных консультаций) и не приобретали при формировании спецификации поставки станка. А докупать подобный функционал потом — гораздо дороже и требует больше времени.

Во-вторых, применение самой команды G54.2P. приносит ряд неудобств. Ну, например, при ее использовании в таблице G54 должна быть не та точка, которая традиционно является программным нулем, а точка пересечения 4-й и 5-й осей, а сам программный ноль задается совсем в других таблицах P. Более того, в эти таблицы данные вносятся не относительно машинного нуля, как это делается в

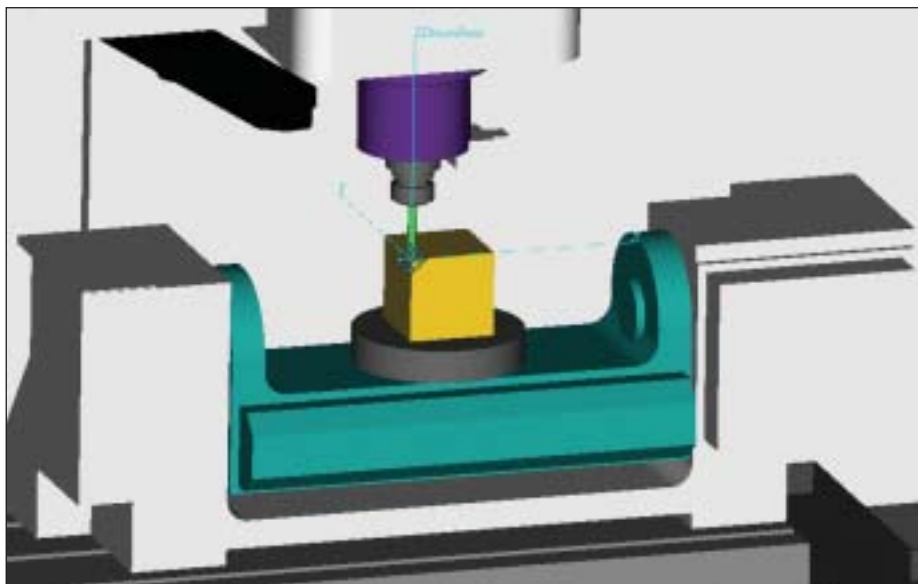


Рис. 7

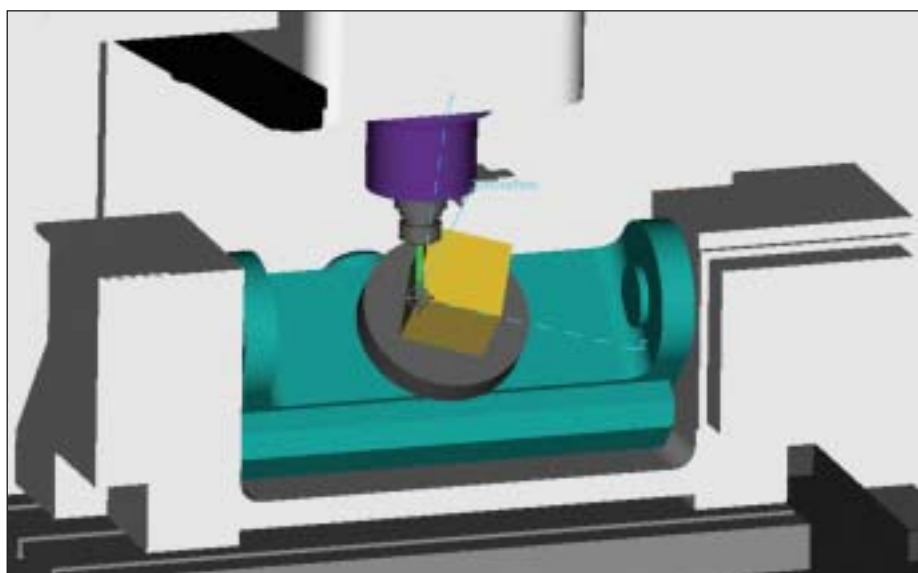


Рис. 8

общем случае, а относительно точки, занесенной в таблицу *G54*!!! И если учесть тот факт, что подобные станки, как правило, приобретаются в комплекте с измерительным шупом (например, Renishaw) и данные при привязке программного нуля автоматически заносятся в таблицы, то мы видим, насколько неудобна эта команда. Ведь все равно приходится вручную пересчитывать и заносить нужные данные в таблицу *P*. Кроме того, команда *G54.2P..* реализована таким образом, что при ее включении станок должен выполнить движение на величину, равную смещению программного нуля относительно точки пересечения 4-й и 5-й осей. И зачастую, если не предпринять компенсирующих мер, это движение приводило к выходу за пределы перемещений (чаще по оси *Z*)!!! Все это и сподвигло на написание собственной процедуры пересчета программного нуля.

Но вернемся к нашему повествованию о станке HAAS. Напомню, что выше мы вели речь о фиксированной 5-осевой обработке. Кроме того, был упомянут другой вид обработок — непрерывная 5-осевая обработка. Это обработка, при которой одновременно (в каждом кадре УП) меняются все 5 (4) координаты. То есть станок совершает довольно сложное движение. Подобные обработки не столь часты, как фиксированные 5-осевые, используются они при изготовлении деталей со сложными геометрическими поверхностями (лопатки турбин, моноколеса, различные авиационные детали). При данном виде траекторий, помимо вышеозвученного положения о программных нулях (с некоторыми отличиями), существует еще один элемент программирования, который уже успел стать базовым. Речь идет о контроле кончика инструмента — RTCP (Rotary

Tool Control Point). Суть этого опционала заключается в том, что при 5-осевом непрерывном движении кончик инструмента движется неотрывно от точки, где находился. Так, на рис. 7 изображен станок в положении, которое возникло после приведения инструмента в точку с координатами *X0Y0Z0* и включения режима RTCP. Если же теперь задать движение только по оси *A* и *B* (к примеру — *G1A45B20F1000*), то инструмент должен будет неотрывно двигаться от той точки, в которой находится, и занять положение, показанное на рис. 8. То есть при выполнении одного кадра УП выполняется комплексное, сложное движение. Таким образом, фактически получается, что при RTCP-режиме происходит наклон инструмента *вокруг* его вершины. А именно такое поведение инструмента и подразумевается в *любой* САМ-системе!!! Если же RTCP-режим не включен, то повороты столов и положение (движение) кончика инструмента не связаны, то есть фактически (с точки зрения траектории УП) повороты происходят относительно поворотных осей станка. И чтобы получить движение, которое я описал чуть выше, то из одного кадра УП (при включенном RTCP-режиме) мы должны иметь много кадров (без RTCP), где наше движение от *A0B0* до *A45B20* должно быть разбито на мелкие перемещения. Более того, от кадра к кадру следовало бы пересчитывать и *X*, *Y*, *Z* с учетом положения программного нуля. А эта задача уже на порядок сложнее, нежели разовый пересчет положения нуля!!!

Стоит ли упоминать, что, к сожалению, ничего подобного на имеющемся оборудовании также нет? И выполнить непрерывную 5-осевую обработку на данном станке нам мешают два существенных ограничения. Во-первых, программный ноль в фиксированной точке станка (на центре планшайбы), о чем мы уже подробно говорили выше. Во-вторых, сами траектории в САМ-системе должны состоять из мелких перемещений. Чем крупнее будет перемещение на том или ином участке обработки, тем большая на нем может возникнуть погрешность (подрез или недорез). Поскольку, еще раз повторю, наклон инструмента происходит не вокруг его кончика, а относительно имеющихся осей поворотов наклонных столов станка. И чем больше на том или ином участке обработки меняется 4-я или 5-я ось, тем большее расхождение будет с тем, что ожидается получить в результате обработки.

Нами была предпринята попытка использовать процедуру пересчета программного нуля для режима работы при непрерывной 5-осевой обработке, чтобы



уйти хотя бы от проблемы фиксированного программного нуля. Выглядело это таким образом, что перед каждым перемещением вызывалась данная процедура, которая динамически смещала программный ноль. Все было бы вполне работоспособно, но нас поджидала другая проблема. Имеющейся системе ЧПУ станка не хватало вычислительной скорости, производительности. Дело в том, что при вполне средних значениях подачи резания (500-1000 мм/мин.) обработка опережала вычисления, и, соответственно, вынужденно ожидала окончания пересчетов в процедуре. Все это выливалось в неравномерный характер движения станка, то есть фактически станок "дергался" почти в каждом кадре УП, что, конечно, недопустимо!

Поэтому для нашего заказчика пришлось формировать 2 постпроцессора, правда, справедливости ради можно отметить, что непрерывные 5-осевые обработки, как ожидается, будут осуществляться крайне редко.

Закончив с разрешением принципиальных моментов при 5-осевой обработке, были выполнены некоторые пожелания попроще. Например, с первых же шагов освоения станка пришло понимание, что мы имеем фактически два разных станка: один — 3-осевой фрезерный, с приличной рабочей зоной, второй — 5-осевой фрезерный с рабочей зоной поменьше. Соответственно, некоторые детали изготавливать при установленных поворотных столах будет невозможно. Что же получается, под каждый режим нужно делать свой постпроцессор? Это опять же нежелательно. Многие могут мне тут же возразить, что постпроцессор, сделанный под общий (5-осевой) режим, должен подойти и для работы под 3-осевой. Разумеется, зачастую это так и есть, но в данном случае, если снять поворотные столы, а в программе задать, например, G0A0B0, система ЧПУ будет выдавать сообщение об ошибке, поскольку столы отключены! А не иметь в УП подобной команды для режима в 5-осевом исполнении станка уже нельзя, поскольку обнуляющие перемещения нужны, как правило, в начале программы, в конце программы и т.п.

Поэтому в постпроцессоре был реализован диалог, в котором пользователь при постпроцессировании выбирал режим работы станка. Соответственно, при выборе 3-осевого режима в УП не попадают команды А и В ни в каком виде. Кроме того, во избежание ошибок в постпроцессоре также реализована проверка на то, чтобы при выборе 3-осевого режима нельзя было постпроцессировать любые 5-осевые операции с выдачей соответствующего сообщения об ошибке (рис. 9).

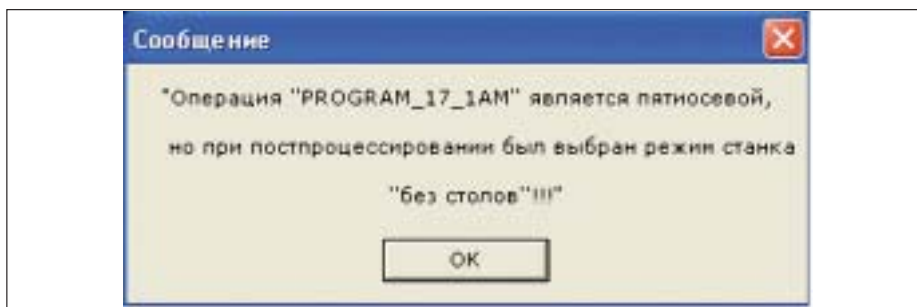


Рис. 9

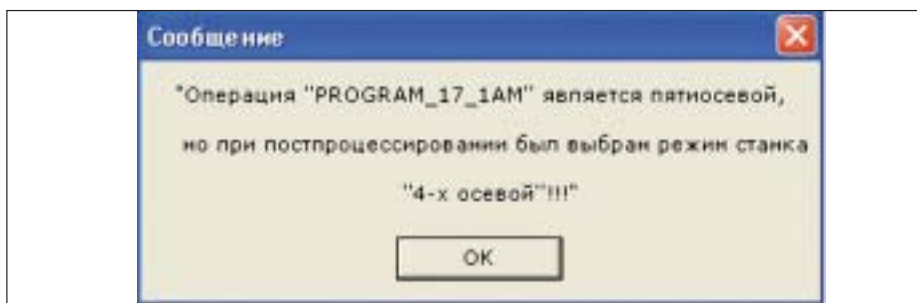


Рис. 10

Более того, почти сразу обнаружилось, что предполагается обрабатывать и изделия, уместающиеся на поворотный стол и требующие обработки с нескольких сторон с использованием только оси А. Но если такое изделие поворачивать по оси В, то мы гарантированно получаем столкновение с боковыми колоннами устройства поворотных столов. Таким образом, понадобился третий, специфичный режим работы станка — режим 4-осевой обработки. Соответственно, в диалог к уже существующим режимам работы был добавлен новый.

При выборе этого режима в УП полностью запрещен вывод команды В в любом виде. Кроме того, при постпроцессировании также анализируется, не требует ли та или иная операция поворота оси В. В этом случае пользователю будет выдано сообщение об ошибке (рис. 10).

Заканчивая рассказ о работе со станком HAAS, упомяну еще про одну мелочь, с которой мы столкнулись. Уже на этапе внедрения постпроцессора технологом была допущена ошибка: в поле задания рабочей подачи вместо 1500 мм/мин. он ввел 15000. В постпроцессорах существует переменная, которая является границей между рабочей подачей (движение по G1) и ускоренной (движение по G0). По умолчанию значение в ней — 10000 мм и указатель на автоматический переход на ускоренное движение также включен. В итоге при обработке получили брак из-за того, что на том участке обработки шла в ускоренном режиме. А, при этом режиме, как и на многих станках, движения осей несогласованны. Поэтому мало того, что движение шло на мак-

симальной скорости, так еще и геометрически было неправильным. Даже если бы в УП было движение по рабочей подаче, но со значением 15000 мм/мин., все равно это негативно сказалось бы как на детали, так и на инструменте. По этой причине в постпроцессор была добавлена возможность анализа величин подачи всех видов движений, при которых осуществляется контакт с обрабатываемой заготовкой — врезание, резание, отвод и т.п. И при превышении значения 10000 будет выдано сообщение об ошибке.

В завершение хотелось бы особо подчеркнуть, что данная публикация ни в коем случае не является попыткой какой бы то ни было антирекламы оборудования компании HAAS, как некоторым может показаться на первый взгляд. Хотелось бы еще раз напомнить о самом главном, на мой взгляд, преимуществе оборудования HAAS — цене!!! Мы лишь стремились объективно осветить некоторые недостатки станков, систем ЧПУ, САМ-систем и путей их устранения для получения приемлемого результата.

**Николай Батарев**  
**ЗАО "CuSoft"**  
**Тел.: (495) 913-2222**  
**E-mail: batarev@csoft.ru**

*Я и мои коллеги из отдела производственного инжиниринга компании ЗАО "CuSoft" традиционно выражаем искреннюю благодарность за совместную работу всем специалистам и коллегам компаний-партнеров. Надеемся, что наше сотрудничество будет продолжаться и расширяться.*