

InventorCAM

для Autodesk Inventor

ЗАОЧНЫЙ МАСТЕР-КЛАСС, ЗАНЯТИЕ 4

Уважаемые читатели, этим материалом мы завершаем цикл статей под рубрикой "заочный мастер-класс"!

Прежде всего обращаю ваше внимание на новое название статьи¹.

Для тех, кто впервые присоединился к нашему заочному обучению, я напоминаю, что на предыдущих трех занятиях был подготовлен токарно-фрезерный проект обработки детали в графической среде Autodesk Inventor. В качестве исходных данных для этого проекта использовался двумерный чертеж в формате DWG. Более подробно об этом вы можете прочитать в предыдущих журналах CADmaster (www.cadmaster.ru/magazin/numbers).

Сегодня мы рассмотрим две темы, которые завершают подготовку управляющих программ обработки на станках с ЧПУ:

- проверка и визуализация процесса обработки;
- настройка постпроцессора для стойки УЧПУ и генерация управляющих программ.

Проверка и визуализация процесса обработки

В одной из своих предыдущих статей² я уже отмечал, что возможность просматривать и контролировать тра-

екторию обработки на экране монитора обеспечивает существенный экономический выигрыш по сравнению с обычным способом отладки управляющей программы непосредственно на станке. "Виртуальная обработка" позволяет не терять драгоценное время оператора, экономнее расходовать дорогой материал заготовки и инструмент, оценить правильность переходов, стратегий и параметров обработки, а также выбора режущего инструмента. Кроме того, можно проверить различного рода "коллизии" и многое другое. В настоящее время для визуализации и контроля обработки InventorCAM предлагает широкий спектр возможностей — от каркасного представления траектории до имитации обработки с учетом полной кинематики станочного оборудования. Конечно же, уважаемые читатели, вы понимаете, что этап визуализации и контроля является динамическим процессом и его "надо видеть, а не читать о нем" в журнальной статье. Поэтому позволю себе остановиться только на описании предоставленных возможностей и кратких советах, не столь очевидных пользователям на первый взгляд.

В InventorCAM для визуализации различных видов обработки (токарной, фрезерной, токарно-фрезерной и электро-

эрозионной) существуют несколько режимов просмотра. Для всех видов обработки поддерживаются режимы *Визуализация на CAD-модели* и *Solid Verify*.

Из файла помощи InventorCAM

Режим *Визуализация на CAD-модели* позволяет просмотреть траекторию обработки непосредственно на модели в окне Autodesk Inventor. В процессе визуализации доступны все опции динамического просмотра моделей Autodesk Inventor: вы можете наблюдать за траекторией инструмента с различных точек, масштабировать и перемещать отображение.

Режим *Solid Verify* позволяет выполнять визуализацию обработки на пространственной модели заготовки. В процессе визуализации InventorCAM посредством булевых операций "вычитает" тело инструмента, движущегося по координатам траектории обработки. Во время визуализации или после нее заготовка является твердотельной моделью, которая может быть динамически повернута или масштабирована. В этом режиме движется инструмент, а не деталь, как это происходит на самом деле (как правило, двигается стол с заготовкой). Следовательно, для всех типов станков визуализация выглядит одинаково.

¹InventorCAM — новое название программного обеспечения SolidCAM, интегрированного в Autodesk Inventor. Впервые InventorCAM был представлен специалистами SolidCAM Ltd. на ежегодной конференции Autodesk University (27-30 ноября 2007 г., Лас-Вегас). В настоящее время InventorCAM сертифицирован компанией Autodesk.

²Андрей Благодаров. Мечты сбываются, или Как сказка стала былью. — CADmaster, № 5/2005.

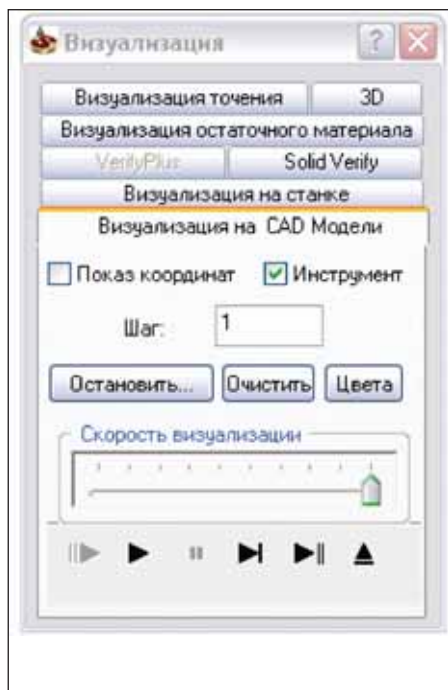


Рис. 1

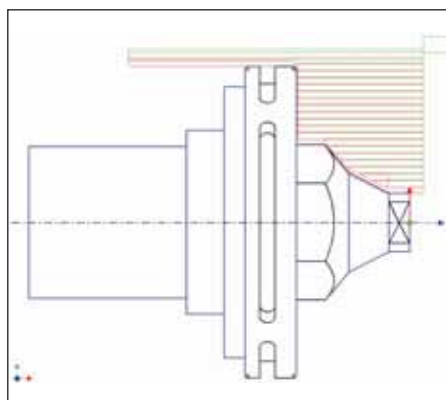


Рис. 3

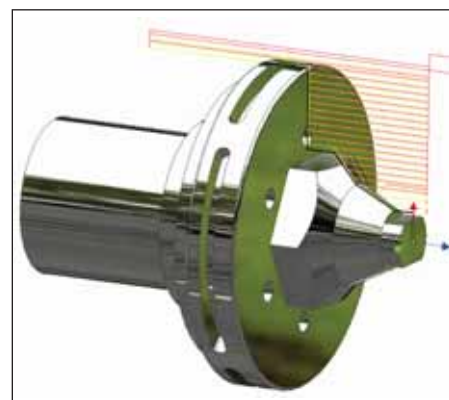


Рис. 4



Рис. 2

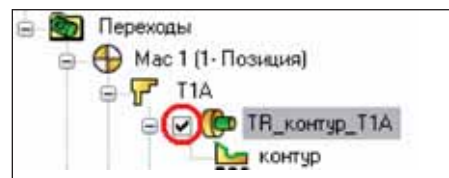


Рис. 5

Шаг 1. Для визуализации всего полученного нами на предыдущих занятиях проекта обработки в InventorCAM Manager (ICM) щелкните правой клавишей мыши (ПрКМ) на разделе *Переходы* → *Визуализация*. Откроется диалоговое окно *Визуализация*, содержащее вкладки с режимами визуализации (рис. 1) и кнопки с управлением процессом визуализации (рис. 2).

Для просмотра первого перехода точения нажмите левой клавишей мыши (ЛКМ) кнопку *Режим отдельного Перехода*.

На экране отобразится траектория, представленная на рис. 3. При обработке 3D-модели детали была бы получена траектория, изображенная на рис. 4. Для завершения процесса визуализации нажмите кнопку *Выход*.

Очень часто в процессе подготовки проекта обработки приходится выполнять визуализацию не всего проекта обработки, а отдельных его переходов. Для таких случаев в InventorCAM предусмотрено несколько возможностей, например просмотр траектории обработки перехода непосредственно из ICM.

Шаг 2. Для просмотра траектории обработки любого перехода активируйте окно просмотра рядом с переходом в ICM (рис. 5).

Если мы хотим использовать различные режимы визуализации отдельного перехода или группы переходов, следует выбрать переход в ICM и в контекстном меню, вызываемом ПрКМ, выполнить команду *Визуализация*. Для примера рассмотрим визуализацию перехода обработки радиальных пазов в режиме *Solid*

Verify. Однако предварительно зададимся вопросами: "Что в этом случае будет отображаться на экране?" и "Что послужит исходным материалом заготовки для данного шага?"

Отвечая на эти вопросы, отметим следующие возможности InventorCAM.

Во-первых, при настройке InventorCAM можно задать параметр *Обновление модели заготовки* (рис. 6) – модель заготовки будет с заданной периодичностью сохраняться во временном файле рабочего каталога.

При этом процесс визуализации существенно ускоряется, поскольку программа не пересчитывает каждый раз состояние модели, однако одновременно увеличивается объем занимаемого места на диске. Рекомендуем использовать периодичность сохранения после каждого перехода. Для экономии места на жестком диске после завершения выполнения проекта все сохраненные временные файлы заготовки можно удалить командой *Удалить файлы обновленной модели*

заготовки в меню настройки *Solid Verify* (рис. 7).

Во-вторых, при настройке InventorCAM можно использовать параметр *Обновление модели заготовки* → *Вручную* (рис. 6). В дальнейшем после выполнения группы переходов возможно сохранять модель заготовки в STL-файле и загружать сохраненный файл перед визуализацией (рис. 8).

Шаг 3. Для визуализации перехода обработки радиальных пазов в режиме

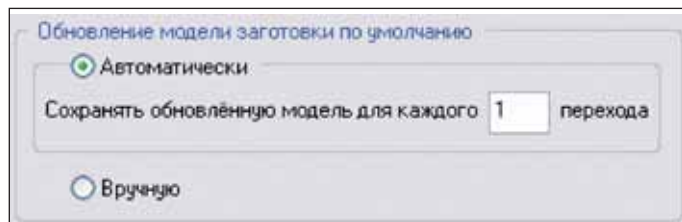


Рис. 6

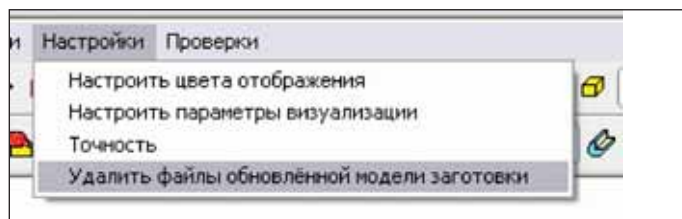


Рис. 7

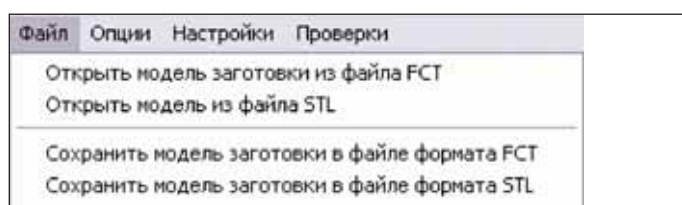


Рис. 8

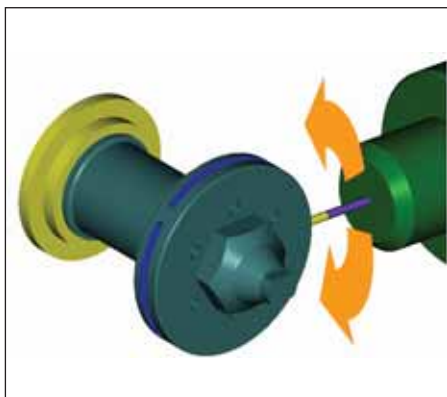


Рис. 9

Solid Verify (в нашем проекте это предпоследний переход) выберем в контекстном меню перехода команду *Визуализация*. Затем перейдем на соответствующую вкладку для выбора режима *Solid Verify* и нажмем кнопку *Start* (рис. 2). На экране отобразится процесс обработки четырех пазов (рис. 9).

Еще раз обращаю ваше внимание, уважаемые читатели: вращается не сама заготовка, а инструмент вокруг нее!

В заключение отметим, что для режима *Solid Verify* очень важна правильность настроек. Они выполняются один раз и, как правило, не требуют изменений.

Находясь в режиме визуализации *Solid Verify*, выберите команду *Настройки* → *Настроить параметры визуализации* (рис. 10).

Рекомендуемые параметры настройки приведены на рис. 11.

Кроме указанных режимов визуализации в *InventorCAM* существуют и режимы, характерные только для фрезерной или только для токарной обработки.

Так, для визуализации токарных переходов предусмотрен режим *Визуализация точения*, позволяющий осуществлять просмотр траекторий движения инструмента в центральном сечении рабочей плоскостью ZX.

Шаг 4. Для визуализации двух токарных переходов обработки выберите ЛКМ оба перехода в ИСМ при нажатой кнопке *Ctrl* и в контекстном меню задайте ко-

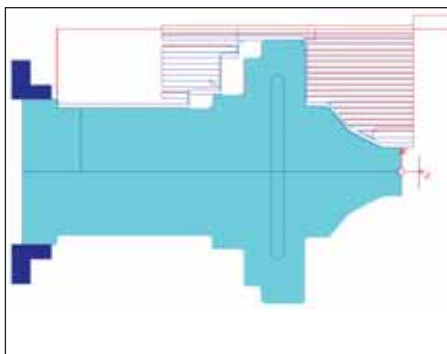


Рис. 13

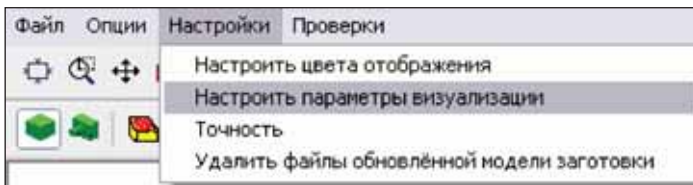


Рис. 10

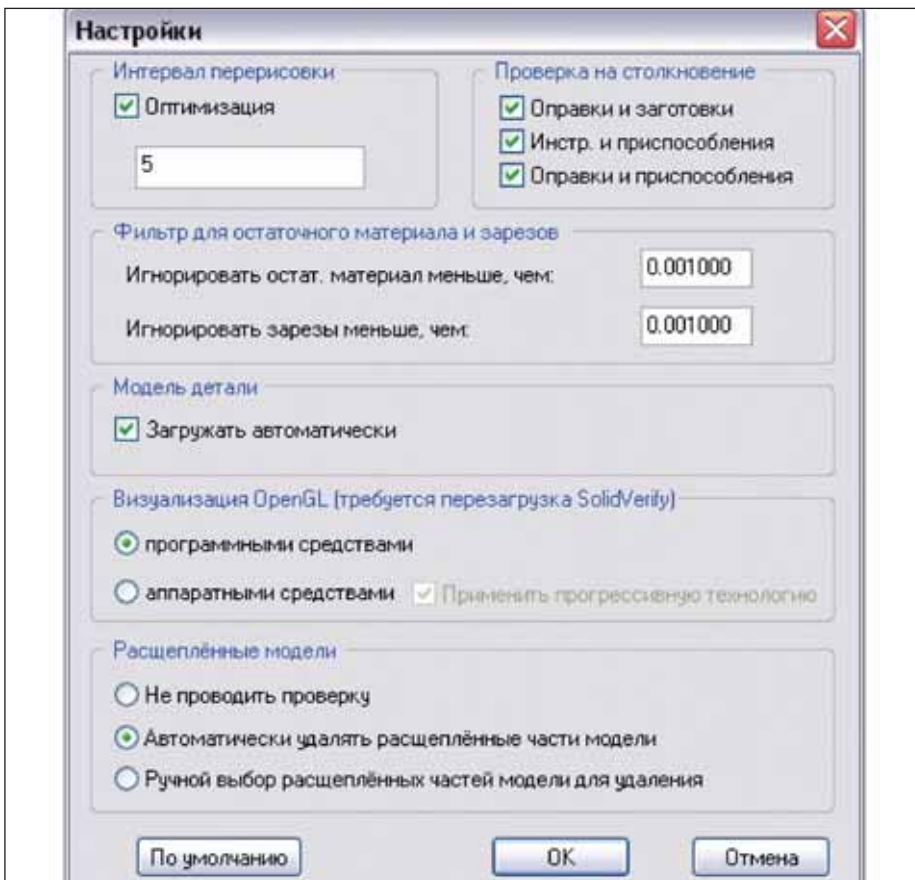


Рис. 11

Из файла помощи *InventorCAM*

Интервал перерисовки — позволяет увеличить скорость визуализации путем определения количества перемещений инструмента до следующей перерисовки изображения на экране.

Проверка на столкновение — в процессе визуализации выявляет возможные коллизии между различными участвующими в обработке компонентами (инструмент, оправка, заготовка, крепежное приспособление). При возникновении коллизий в процессе визуализации на экран будут выводиться соответствующие предупреждения (рис. 12).

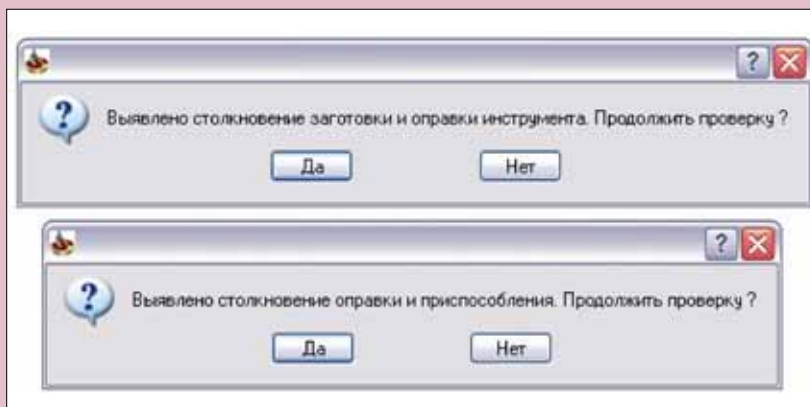


Рис. 12

Фильтр для остаточного материала и резцов — позволяет задавать допуски на визуализацию остаточного материала и областей возможных резцов.

Визуализация OpenGL — обеспечивает увеличение производительности визуализации при использовании различных графических адаптеров посредством переключения между программным/аппаратным ускорением OpenGL.

Расщепленные модели — позволяет автоматически определять и удалять с экрана "повисшие в воздухе" части материала заготовки.

манду *Визуализация*. Затем для выбора режима *Визуализация точения* перейдите на соответствующую вкладку, установите опцию *Все вместе* в рабочем поле *Показать* и нажмите кнопку *Старт*. На экране будет отображен процесс обработки (рис. 13).

Поскольку в нашем проекте мы не имеем возможности показать специфичные режимы визуализации для фрезерной обработки, просто опишем их.

В завершение темы проверки и визуализации процесса обработки отдельно рассмотрим режим *Визуализация на станке*. Очень часто этот режим визуализации воспринимают как проверку кадров управляющей программы.

Уважаемые читатели, я хочу вас огорчить: к сожалению, это не так!

Но чтобы вы не очень расстраивались, сразу же сообщу, что в арсенале нашей компании есть программное обеспечение, позволяющее переводить представленные в текстовом виде кадры управляющей программы в движение инструмента на экране компьютера, а также осуществляет полную проверку процесса обработки с учетом обрабатываемого оборудования — VERICUT! На страницах журнала CADmaster мы неоднократно делились опытом использования этого программного обеспечения. Более подробно ознакомиться с программой вы можете, посетив страничку www.csoft.ru/catalog/soft/version_9948.html или позвонив нашим специалистам.

Однако вернемся к InventorCAM. Зачем же нужен режим *Визуализация на станке* и чем он может помочь на этапе просмотра и проверки проекта обработки?

Ответ очевиден! Он позволяет выполнить проверку траектории обработки с учетом перемещений исполнительных органов оборудования, смены режущего инструмента, линейных и угловых ограничений рабочих зон вашего станка.

Конечно же, прежде чем использовать этот режим визуализации, необходимо выполнить достаточно большой объем ответственной и кропотливой работы, связанной с трехмерным моделированием вашего оборудования в реальном масштабе и с описанием возможностей динамической схемы взаимодействия исполнительных

Из файла помощи InventorCAM

2D — позволяет просмотреть траекторию движения инструмента на проекционном виде, что бывает полезно при обработке призматических деталей.

3D — позволяет просмотреть траекторию обработки на 3D-модели с показом движения 3D-инструмента без оправки. Перед применением данного режима визуализации должна быть определена модель получаемой детали.

Rapid Verify — позволяет просмотреть траекторию движения инструмента фрезерной обработки, которая содержит большое количество строк обработки. Отличие этого режима от *Solid Verify* состоит в том, что визуализация модели заготовки производится в упрощенном представлении, благодаря чему увеличивается скорость визуализации. Однако при этом для данного режима существует ограничение: он не предназначен для просмотра траектории 4- и 5-координатной фрезерной обработки.

органов. Подробный рассказ о подготовительной работе — отдельная тема, выходящая за рамки нашего заочного мастер-класса. Когда-нибудь мы остановимся на ней подробнее, а пока проанализируем этот режим в привязке к нашему проекту обработки. Для удобства возьмем переход, выполненный на шаге 3.

Шаг 5. Для визуализации перехода обработки радиальных пазов зададим в контекстном меню перехода команду *Визуализация* и перейдем для выбора режима *Визуализация на станке* на соответствующую вкладку. На экране отображается модель станка с заготовкой и траекторией обработки, а также деталь (рис. 14).

С помощью колеса мыши увеличим изображение рабочей зоны и выполним пошаговый просмотр траектории обработки, последовательно нажимая кнопку *Step forward* на рабочей панели (рис. 15).

На экране будет отображаться процесс обработки четырех пазов. При этом, что очень важно, движение инструмента по траектории обработки осуществляется в соответствии с динамической схемой взаимодействия исполнительных органов оборудования.

На рисунках 16а, 16б и 16в представлены три положения инструмента при обработке радиального паза. Можно заметить, что в процессе визуализации

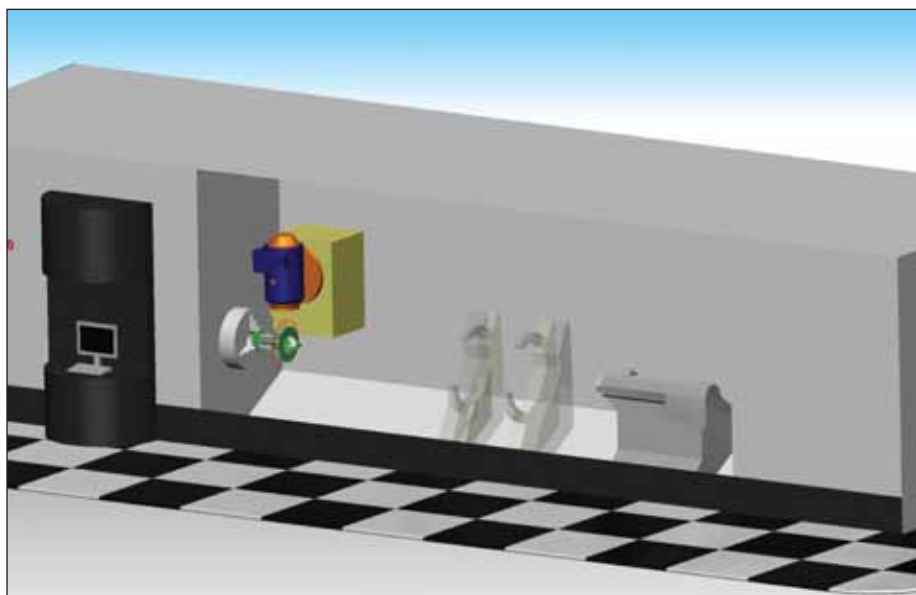


Рис. 14



Рис. 15

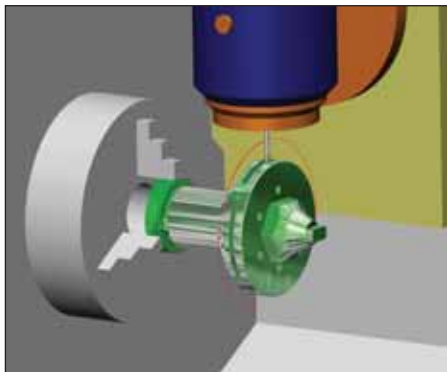


Рис. 16а

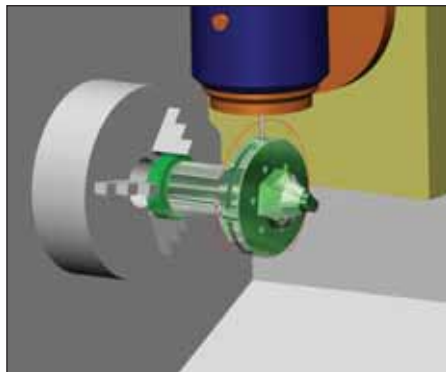


Рис. 16б

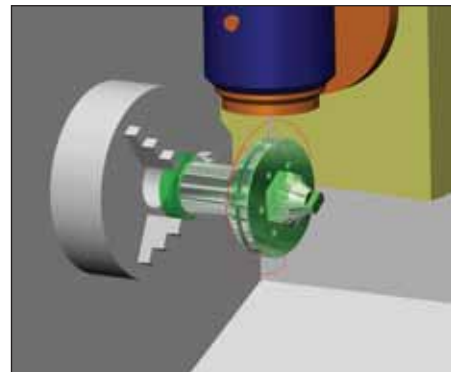


Рис. 16в

происходит вращение кулачков с зажатой в них заготовкой.

Вы, конечно же, уже обратили внимание, что интерфейс данного режима англоязычный. Однако, на наш взгляд, это не мешает вам в работе. Тем более что все найденные коллизии сохраняются в протоколе обработки, демонстрируются на экране и фиксируются в строках с координатами перемещения. Но чтобы вам было проще, переведем несколько сообщений, выдаваемых при возникновении коллизий.

Перевод сообщений об ошибках обработки

Collision between spindle and ... — столкновение между шпинделем и...;

Collision between tool and workpiece, range: more than... — столкновение между инструментом и заготовкой, более чем... (мм);

Out of Limits — выход за лимиты.

Итак, мы в "журнальном варианте" просмотрели процесс обработки детали, подготовленный на предыдущих занятиях заочного мастер-класса, выявили недочеты в нашем проекте обработки, устранив которые можем переходить к завершающему шагу — генерации текста управляющей программы.

Генерация управляющих программ и настройка постпроцессора

Как выяснилось, главная цель обучения в нашем мастер-классе — обыкновенный текстовый файл, файл управляющей программы! Его создание самый легкий шаг в процессе обработки. Как вы уже убедились, в каждом переходе, подготовленном нами, присутствует кнопка **УП**, нажатие которой приводит к генерации текста управляющей программы для данного перехода.

Какими еще способами генерируется управляющая программа?

Можно выделить в ICM переход или группу переходов и в контекстном меню выбрать команду *Создание УП*. В

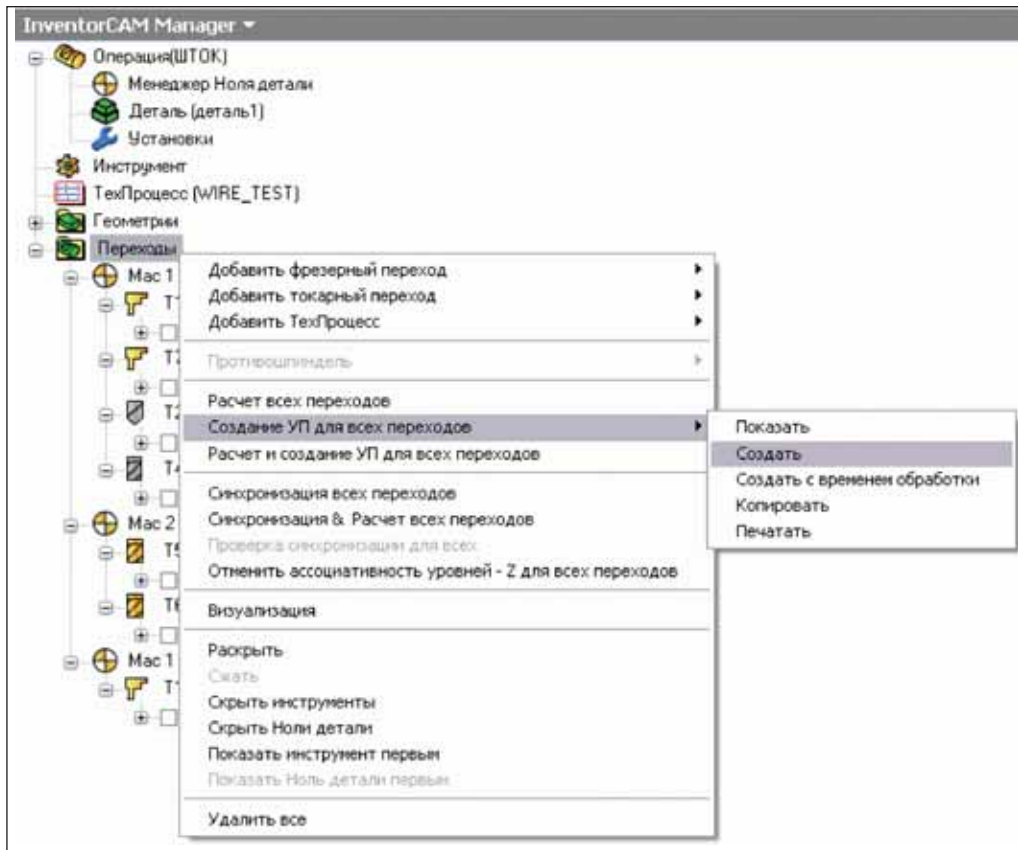


Рис. 17

результате получим управляющую программу для данного перехода или группы переходов.

Шаг 6. Чтобы получить общую управляющую программу для всего проекта обработки, необходимо нажать ПрКМ в ICM на разделе *Переходы* и в контекстном меню выбрать команду *Создание УП для всех переходов* (рис. 17).

Не правда ли, все легко и очень просто?

Почему же тогда на многих предприятиях так сильны страх и ужас перед внедрением новых технологических систем? Почему опытные в своем деле технологи настолько консервативны, что отдают предпочтение устаревшим технологическим системам, а порой прибегают к написанию текста управляющих программ вручную?

Дело в том что за легкостью генерации управляющих программ в InventorCAM скрывается очень ответственный этап — настройка постпроцессора!

На многих предприятиях с большим опытом использования станков с ЧПУ не осталось специалистов, способных выполнить эту работу. А молодые специалисты, "рвущиеся в бой", как правило, натываются на такие "стены неприятия" в лице старых технологов, что им легче научиться работать с устаревшим программным обеспечением, чем реализовать свои знания и возможности.

Какие возможности предлагает InventorCAM для настройки постпроцессора?

Не буду подробно описывать настроенные постпроцессоры, уже имеющиеся в библиотеке, накопленной за годы раз-

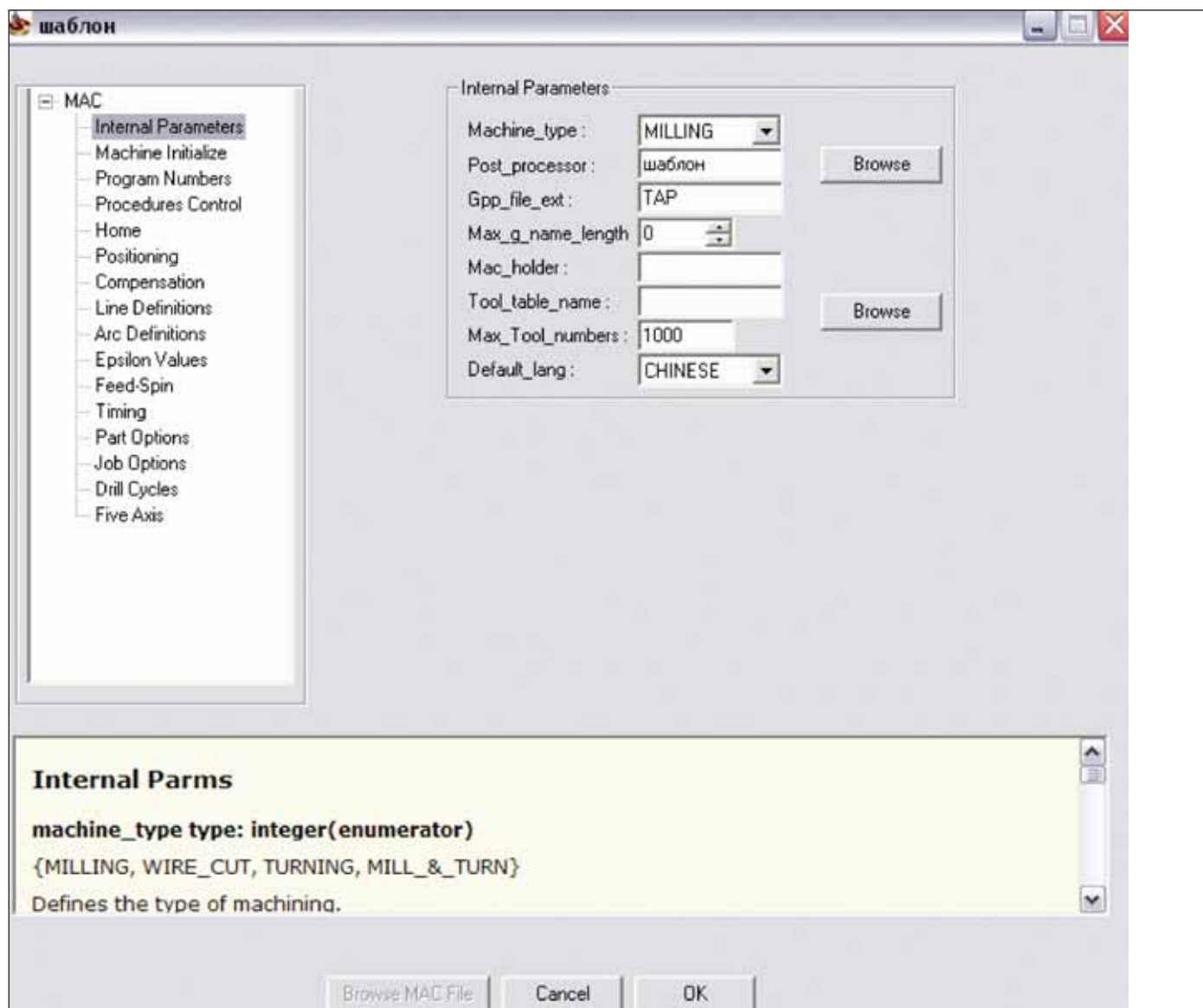


Рис. 18

вития и использования программы SolidCAM/InventorCAM. Отмечу лишь, что такая библиотека существует и она достаточно обширна.

Лучше расскажу о том, как разрабатываются и настраиваются постпроцессоры в программе InventorCAM. Конечно, это не будет перепечаткой файла помощи, потому что любой пользователь программы может с ним ознакомиться самостоятельно. Не станем также подробно рассказывать обо всех этапах настройки, поскольку каждый наш пользователь в любой момент может получить от нас исчерпывающий ответ на любой вопрос. В конце концов, у меня просто нет никакого желания поддерживать и обучать "нелицензионных" пользователей! Поэтому, освещая проблему настройки постпроцессора, я расскажу о принципах и шагах разработки, а также поделюсь опытом решения некоторых проблем.

Нередко приходится слышать, что в той или иной программе настройка

постпроцессора производится с помощью специализированного Мастера настройки, достаточно удобного и легкого: стоит ответить на заданные вопросы — и постпроцессор готов. Хочу предостеречь вас от кажущейся простоты и легкости. Как правило, такой подход возможен только для описания "паспортных" возможностей станка, но никак не для "тонкой" настройки постпроцессора.

В программе InventorCAM настройка нового постпроцессора ведется в двух файлах, в одном из которых содержатся паспортные характеристики и возможности станка (MAC-файл), а во втором — описание синтаксиса кадров готовой управляющей программы (GPP-файл).

Как уже было отмечено выше, для настройки паспортных характеристик можно воспользоваться специализированным Мастером, в котором все процедуры поделены на отдельные разделы и содержат наборы специальных параметров.

На рис. 18 приведено окно первой

процедуры с параметрами (тип станка, расширение файла управляющей программы, имя таблицы инструмента, связанной с данным станком, имя файла постпроцессора, отвечающего за вывод кадров УП и т.д.).

Какие вопросы задаются при настройке данного файла наиболее часто?

Вопрос 1: "У меня есть настроенная таблица применяемого инструмента для конкретного станка. Могу ли я выполнить настройку постпроцессора так, чтобы автоматически подгружать эту таблицу в проект обработки?"

Ответ: "Да, можете. Для этого необходимо использовать параметр *tool_table_name*, значение которого содержит имя таблицы и полный путь к ней".

Вопрос 2: "Могу ли я для уменьшения "длины" файла УП использовать повторяющиеся циклы?"

Ответ: "Да, можете. Для этого следует использовать параметр *loops*. В резуль-

Параметр $arc_3d = Y$	Параметр $arc_3d = N$		
... G3 X165.856 Y40.05 Z0. R10. F100 X171.015 Y22.598 Z-2. R10. G1 X182.271 Y34.185 Z1.852 F100 X181.846 Y35.822 Z1.704 X181.151 Y37.364 Z1.556 X180.207 Y38.767 Z1.407 X179.039 Y39.991 Z1.259 X177.682 Y41. Z1.111 X176.174 Y41.766 Z0.963 X174.559 Y42.267 Z0.815	X172.882 Y42.489 Z0.667 X171.192 Y42.425 Z0.519 X169.537 Y42.077 Z0.37 X167.964 Y41.455 Z0.222 X166.518 Y40.577 Z0.074 X165.241 Y39.468 Z-0.074 X164.17 Y38.159 Z-0.222 X163.334 Y36.689 Z-0.37 X162.758 Y35.099 Z-0.519 X162.458 Y33.434 Z-0.667	X162.443 Y31.743 Z-0.815 X162.713 Y30.074 Z-0.963 X163.261 Y28.473 Z-1.111 X164.07 Y26.988 Z-1.259 X165.118 Y25.661 Z-1.407 X166.375 Y24.529 Z-1.556 X167.805 Y23.626 Z-1.704 X169.366 Y22.976 Z-1.852 X171.015 Y22.598 Z-2. ...

тате можно получить, например, следующие кадры управляющей программы:

```
....
G91 #23 = 0
WHILE [#23 LT 11] DO 3
....
#23 = #23 + 1
....".
```

Вопрос 3: "Могу ли я получить УП с подпрограммой?"

Ответ: "Да, можете. Для этого необходимо использовать параметр $gen_procs = Y$ ".

Вопрос 4: "Могу ли я использовать возможности спиральной обработки на станке?"

Ответ: "Да, можете. Для этого следует применить параметр arc_3d . Если такой параметр не установлен, спиральные дуги разделяются на линейные сегменты в

соответствии с заданной точностью обработки".

Вопрос 5: "Станок умеет обрабатывать дуги только по квадрантам. Можно ли задать разбиение дуг на квадранты?"

Ответ: "Да, можно. Используйте параметр $arc_quadrants = Y$ ".

Вопрос 6: "Могу ли я использовать собственные параметры для определения своих команд в управляющей программе?"

Рис. 19

Ответ: "Да, вы можете использовать до 50 собственных параметров при определении проекта обработки (*Part Options*) и столько же собственных параметров для переходов обработки (*Job Options*)".

Вопрос 7: "В каком файле и разделе описываются параметры циклов обработки отверстий?"

Ответ: "Количество и типы циклов обработки отверстий, как и используемые параметры циклов, описываются в MAC-файле раздела *drill_type*. Правильное формирование кадра вывода цикла обработки зависит от настройки GPP-файла".

Вопрос 8: "Какие параметры необходимо задать для определения многоосевой обработки?"

Ответ: "Параметры, описывающие многоосевую обработку, включая параметры режима *Визуализация на станке*, задаются в разделах *Fourth axis* и *Sim Five axis*".

На рис. 19 приведено окно Мастера настроек для определения параметров раздела *Sim Five axis*.

После описания паспортных данных станка и его функциональных возможностей следует перейти к настройке файла для описания синтаксиса кадров управляющей программы. Для настройки такого файла в InventorCAM не предусмотрено никакого Мастера. Отметим, что оба файла (MAC и GPP) — текстовые и могут быть созданы и отредактированы в любом привычном для вас текстовом редакторе. Язык программирования, используемый при настройке файла, описывающего синтаксис кадров УП, носит название GPPtool.

Описание структуры кадров управляющей программы необходимо начинать с описания форматов вывода параметров (координат, вспомогательных и основных команд и т.п.), которые будут использованы по умолчанию. При необходимости формат можно изменить как при выводе конкретного параметра в строке, так и при выводе параметров процедуры в целом. Пример параметров, описывающих формат, приведен ниже:

- *numeric_def_f*
- *integer_def_f*
- *gcode_f, mcode_f*
- *xpos_f, ypos_f, zpos_f*
- *feed_f*
- *blknum_f*

и т.п.

Для описания форматов вывода существует строковое выражение, определяющее характер вывода параметров строки.

Примером настройки формата вывода координат для нашего проекта может служить строка вывода координаты X:

xpos_f = '5.3(*2)'

Из файла помощи GPPtool

Формат вывода — строковое выражение, определяющее характер вывода элементов.

<sign><leading-zeroes><integer> . <fraction><trailing-zeroes><options>

Формат вывода включает следующие компоненты:

<sign> — управление знаком:

'+'	— введите знак '+' для обозначения положительных чисел;
'-'	— введите знак '-' для обозначения отрицательных чисел;
нет опции	— знак вводится только для отрицательных чисел.
<leading-zeroes>	— управление незначащими нулями перед числом:
''	— перед числом будут выводиться пробелы;
'z'	— перед числом будут выводиться незначащие нули;
нет опции	— ничего не выводится.
<integer>	— управление целой частью числа. Этот компонент может иметь вид 'M' или 'M/m', где:
M	— количество цифр перед десятичной точкой;
/m	— минимальное количество генерируемых цифр (по умолчанию — 1);
.	— десятичная точка.
<fraction>	— управление дробной частью. Этот компонент может иметь вид 'N' или 'N/n', где:
N	— количество цифр после десятичной точки (по умолчанию — 0 для целочисленных выражений и 3 для чисел);
/n	— минимальное количество генерируемых цифр (по умолчанию — 0).
<trailing-zeroes>	— управление незначащими нулями после числа:
''	— добавление пробелов после числа;
'z'	— добавление незначащих нулей после числа;
нет опции	— ничего не добавляется.
<options>	— различные опции форматирования заключаются в скобки, а для удобочитаемости разделяются пробелами. Опции можно задавать как строчными, так и прописными буквами:
*s	— коэффициент масштабирования; перед генерацией число умножается на 's' (где s — любое число, включая нецелочисленные значения);
/s	— коэффициент масштабирования; перед генерацией число делится на 's' (где s — любое число, включая нецелочисленные значения);
d	— не выводить, если ноль (если генерируемое число равно нулю, ничего генерироваться не будет);
n	— без десятичной точки (число генерируется без десятичной точки);
p	— условная десятичная точка (число генерируется без десятичной точки, если у него нет дробной части);
i	— инвертирование знака; знак числа изменяется на противоположный, после чего выполняется его генерация.

■ **отсутствие знака** — соответствует выводу знака только для отрицательных значений;

■ **отсутствие параметров управления незначащими нулями** — обозначает, что такие нули выводиться не будут;

■ цифры 5 и 3 — количество знаков числа, вводимых до и после десятичной точки;

■ опция *2 — обозначает, что расчетные значения координат по оси X должны при выводе умножаться на 2 (вывод координат обработки в метрах).

Продолжим дальше.

Как вы, уважаемые читатели, понимаете, при формировании текста управляющих программ существует множество процедур, которые могут как взаимно исключать друг друга, так и повторяться или постоянно действовать.

Для примера приведу некоторые из них. К взаимоисключающим процедурам, то есть действующим однократно в процессе формирования текста управляющих программ, можно отнести процедуры начала (@start_of_file) и конца (@end_of_file) файла, формирование кадров "шапки" в начале (@start_program) и конце (@end_program) программы. Причем эти процедуры действуют одинаково как для программы отдельного перехода, так и для программы проекта в целом.

К повторяющимся процедурам, то есть действующим периодически по мере необходимости их возникновения, можно отнести процедуры начала (@start_of_job) и завершения (@end_of_job) каждого перехода, смены инструмента (@change_tool, @turn_change_tool), изменения рабочей плоскости обработки (@machine_plane) и т.п.

К постоянно действующим процедурам можно отнести процедуры формирования линейных (@line, @move) и дуговых (@arc) перемещений, компенсации инструмента (@compensation), режимов резания (@feed_spin) и т.д.

Формирование любой процедуры должно начинаться знаком @ и заканчиваться командой *endp*. Таким образом, мы подошли к знакомству с языком программирования GPPtool. Какие процедуры используются при формировании кадров управляющей программы и можно ли добавлять и использовать свои собственные? Основные процедуры, их параметры и порядок выполнения определены разработчиками программного обеспечения. Можно ли изменять этот порядок? Да, можно. Для этого используются операторы условных переходов (*IF, active, change*) и команда вызова процедур (*CALL*).

Внутри каждой процедуры формируются соответствующие ей кадры управляющей программы, в которых используются значения параметров, определенные для данной процедуры. Как же формируются кадры программы и какие параметры в процедурах можно использовать?

Для формирования кадра программы необходимо использовать фигурные скобки {...}, внутри которых заключено выражение для вывода в строке управляющей программы. Это выражение может

Из файла помощи GPPtool

Формат: *trace* <процедуры>: <уровень трассировки>

Описание: при генерации управляющей программы выводится информация о трассировке для процедур, заданных в списке <процедуры>. Значение <уровень трассировки> определяет количество выводимой информации в диапазоне от 1 до 5, где 5 определяет максимальное количество доступной информации.

Примеры:

trace "all": 1 — будет выполнена трассировка всех процедур с минимальным количеством доступной информации;

trace "@line,@arc":5 — будут трассироваться только процедуры '@line' и '@arc', при этом сгенерируется максимальное количество доступной информации.

содержать как неизменяемые значения, например, вывод названия оси перемещения X, Y, Z (эти выражения должны быть заключены в кавычки, например — 'X'), так и изменяющиеся значения, например, координаты по осям X,Y,Z в текущий момент формирования кадра (эти значения равны текущим значениям соответствующих переменных, например, *xpos, ypos, zpos*).

Как уже было сказано выше, в основном имена параметров процедур определены разработчиками, а их значения за-

даются вами во время создания проекта обработки или автоматически рассчитываются системой. Чтобы увидеть, в какой процедуре формируется тот или иной кадр управляющей программы и какие переменные и значения при этом используются, в GPPtool предусмотрена команда *trace*.

Ниже приведен пример трассировки процедур @start_of_file и @arc с различным уровнем трассировки для первого токарного перехода нашего проекта.

Как можно видеть из приведенного примера, в процедуре @start_of_file генерируются два кадра управляющей программы: % и O1001. Оба они выводятся без номера кадра, и ни один из доступных нам параметров (таких как *g_file_name, part_name, clearance_plane* и пр.) не используется. В процедуре @arc генерируется один кадр управляющей программы *N190 G3 X45.8 Z-17.75 R1.45*, в котором используется вывод номера кадра и текущие значения параметров (*zpos, (xpos*2)* и *radius*).

Как было отмечено выше, вы сами определяете используемые параметры. В приведенном выше примере формирование кадра движения по дуге обеспечивается следующим текстом в GPP-файле:

```
@arc
if arc_direction eq CCW then
  gcode = 3
else ; CW
  gcode = 2
endif
{nb, ['G'gcode], 'X'xpos, 'Z'zpos}
if arc_size >= 180 then
  radius = -radius
endif
{' Rradius}
endp
```

- *nb* — команда формирования и вывода номера строки (*N190*);
- условие *if arc_direction eq CCW then* определяет, какую команду (*G2* или *G3*) использовать при выводе в зависимости от направления обработки;
- оператор *{['G'gcode] 'X'xpos 'Z'zpos}* формирует и выводит команду круговой интерполяции и координаты конечной точки дуги, причем команда круговой интерполяции выводится с условием модальности (это определено квадратными скобками [...]);
- оператор *{' Rradius}* выводит величину радиуса дуги.

На первый взгляд, все понятно и просто, однако может возникнуть вопрос: "Почему для формирования одной строки УП в постпроцессоре используются два оператора?" Ответим: "Потому что имеется условие на смену знака величины радиуса, если угловое значение дуги (*arc_size*) строго больше 180 градусов". При отсутствии такого условия оператор

trace "@start_of_file, @arc":3

```
@start_of_file ==> program_number:11 g_file_name:'TR_контур6_T1A.NC'
..> full_g_file_name:'D:\Work_Project\ШТОК\TR_контур6_T1A.NC'
..> part_name:'ШТОК'
..> index_split_file:1
..> rotate_used:false mirror_used:false fourth_axis_used:false
> %
> O1001
```

.....

```
@arc ==> xpos:22.900T ypos:-17.750F zpos:-17.750T feed:0.300F
> N190 G3 X45.8 Z-17.75 R1.45
```

trace "@start_of_file, @arc":5

```
@start_of_file ==> program_number:11 g_file_name:'TR_контур6_T1A.NC'
..> full_g_file_name:'D:\Work_Project\ШТОК\TR_контур6_T1A.NC'
..> part_name:'ШТОК'
..> index_split_file:1
..> rotate_used:false mirror_used:false fourth_axis_used:false
..> first_proc_number:5002 last_procedure_number:5002
..> home_number:1 home_changed:false
..> clearance_plane:0.000 tool_start_plane:0.000
..> work_upper_plane:0.000 zero_plane:0.000
> %
> O1001
```

.....

```
@arc ==> xpos:22.900T ypos:-17.750F zpos:-17.750T feed:0.300F
..> arc_direction:ccw xcenter:-17.750 ycenter:21.450 radius:1.450
..> arc_plane:zx xcenter_rel:-1.342 ycenter_rel:-0.550
..> start_angle:22.291 end_angle:90.000 arc_size:67.709
..> next_direction:0.000F
> N190 G3 X45.8 Z-17.75 R1.45
```

вывода кадра управляющей программы выглядел бы следующим образом:

```
{nb, ['G'gcode], 'X'xpos, 'Z'zpos, 'R'radius}
```

Для многих стоек управления при выводе кадра обработки дуги необходимо определять не значение радиуса, а координаты центра. Причем эти координаты могут задаваться как в абсолютных значениях технологической системы координат, так и в относительных значениях от точки начала дуги. Следовательно, для вывода приведенного выше кадра управляющей программы необходимы операторы (используемые имена и величины параметров приведены в примере с трассировкой):

```
■ {nb, ['G'gcode], 'X'xpos, 'Z'zpos, 'I',  
xcen: xpos_f, 'K', ycen: zpos_f} — для  
абсолютных координат центра дуги  
(кадр N190 G1 X45.8 Z-17.75 I-35.5  
K21.45)
```

```
или  
■ {nb, ['G'gcode], 'X'xpos, 'Z'zpos, 'I', xcen-  
ter_rel: xpos_f, 'K', ycenter_rel: zpos_f} —  
для относительных координат центра  
дуги (кадр N190 G1 X45.8 Z-17.75 I-  
2.683 K-0.55).
```

Аналогичным образом описываются операторы для вывода необходимых кадров управляющей программы в каждой процедуре.

Попробуйте самостоятельно, это интересно и очень просто!!!

Ниже я хотел бы привести несколько советов-примеров описания операторов формирования кадров УП в виде ответов на наиболее часто встречающиеся вопросы (продолжим начатую ранее нумерацию):

Вопрос 9: "Раньше мы писали управляющие программы токарной обработки вручную и при этом применяли токарные циклы. Можно ли в InventorCAM использовать токарные циклы и как их описать в файле настройки постпроцессора?"

Ответ: "Да, можно. Для этого необходимо активировать в токарном переходе опцию *Использование цикла* (рис. 20). Для описания токарных циклов в постпроцессоре используются процедуры @turning, @turn_proc и @turn_endproc.

Пример использования токарных циклов приведен в таблице ниже.

Обращаю ваше внимание, что даже если все строки управляющей программы генерируются без номера кадра, то в описании токарного цикла такая нумерация необходима для кадров, описывающих геометрию контура. Для выполнения таких условий используем параметр *blknum_exist* (вывод нумерации кадров) следующим образом:

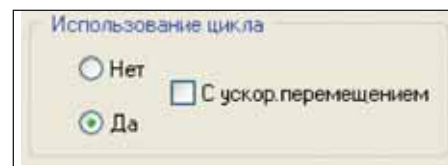


Рис. 20

- 1) для всей программы в процедуре @init_post устанавливаем параметр *blknum_exist = false*;
- 2) для вывода номеров кадров в процедуре @turn_proc устанавливаем параметр *blknum_exist = true*;
- 3) после завершения формирования токарного цикла в процедуре @turn_endproc устанавливаем параметр *blknum_exist = false*.

При такой настройке номера кадров формируются и выводятся только при описании геометрии в цикле, причем в каждом последующем цикле нумерация строк продолжается и, следовательно, в каждом токарном цикле значения параметров *P* (номер первого кадра описания геометрии в цикле) и *Q* (номер последнего кадра описания геометрии в цикле) всегда разные".

Последний вопрос этого занятия я хочу представить в виде задачи.

Опция <i>Использование цикла</i> неактивна				Опция <i>Использование цикла</i> активна
% O1001 G50 S3000 G28 U0 W0 G54 T0101 G97 S600 M4 G0 X54. Z10.3 M8 Z2. X48. G1 Z-41. F0.3 G0 X50.4 Z2. X46. G1 Z-41. G0 X48.4 Z2. X44. G1 Z-16.408 G3 X45.8 Z-17.75 R1.45 G1 Z-41. G0 X46.2 Z2. X42. G1 Z-16.3 G1 X42.9 G3 X44. Z-16.408 R1.45 G1 X44.4 Z-16.208 G0 Z2.	X40. G1 Z-16.3 G1 X42. X42.4 Z-16.1 G0 Z2. X38. G1 Z-16.3 G1 X40. X40.4 Z-16.1 G0 Z2. X36. G1 Z-16.3 G1 X38. X38.4 Z-16.1 G0 Z2. X34. G1 Z-16.3 G1 X36. X36.4 Z-16.1 G0 Z2. X32. G1 Z-16.3 G1 X34. X34.4 Z-16.1 G0 Z2. X30. G1 Z-16.3 G1 X32. X32.4 Z-16.1	G0 Z2. X28. G1 Z-16.3 G1 X30. X30.4 Z-16.1 G0 Z2. X26. G1 Z-16.3 G1 X28. X28.4 Z-16.1 G0 Z2. X24. G1 Z-16.3 G1 X26. X26.4 Z-16.1 G0 Z2. X22. G1 Z-12.215 X22.202 Z-12.3 G3 X22.917 Z-13.066 R1. G1 Z-16.3 G1 X24. X24.4 Z-16.1 G0 Z2. X20. G1 Z-11.376 X22. Z-12.215 X22.4 Z-12.015 G0 Z2.	X18. G1 Z-10.537 X20. Z-11.376 X20.4 Z-11.176 G0 Z2. X16. G1 Z-9.697X18. Z-10.537 X18.4 Z-10.337 G0 Z2. X14. G1 Z-8.663 X14.348 Z-9.004 X16. Z-9.697 X16.4 Z-9.497 G0 Z2. X12. G1 Z-6.7 X14. Z-8.663 X14.4 Z-8.463 G0 Z2. X10. G1 Z-4.737 X12. Z-6.7 X12.4 Z-6.5 G0 Z2. G0X200.Z20. M5 M9 M30 %	% O1001 G50 S3000 G28 U0 W0 G54 T0101 G97 S600 M4 G0 X58.8 Z10.3 M8 Z2. G71 U1 R0.2 G71 P10 Q80 U0 W0 F0.3 N10 G1 X8.8 F0.1 N20 Z-3.56 N30 X14.348 Z-9.004 N40 X22.202 Z-12.3 N50 G3 X22.917 Z-13.066 R1. N60 G1 Z-16.3 N70 G1 X42.9 N80 G3 X45.8 Z-17.75 R1.45 G1 Z-41. G0 Z10.3 G0X200.Z100. M5 M9 M30 %

$$\text{ускорение тела} = \frac{\text{сила}}{\text{масса тела}}$$

$$\text{катет}^2 + \text{катет}^2 = \text{гипотенуза}^2$$

Рис. 21

Вопрос 10: Условие: "Кроме современного оборудования с плавным регулированием оборотов вращения шпинделя мы используем на производстве старые станки с ЧПУ, у которых обороты дополнительно регулируются 4-ступенчатой коробкой передач. Сколько опций, каких и где необходимо установить в Мастере настроек при разработке постпроцессора для такого станка?"

Ответ, скорее всего, будет следующим: "Разработать постпроцессор для такого станка с помощью Мастера настроек невозможно! Чтобы сделать это, следует прибегнуть к программированию!"

Ниже приведем вариант решения такой задачи, выполненный А. Розовым (CSoft Ярославль) при настройке постпроцессора для своих пользователей.

Как оказывается, все просто!

На этом, уважаемые читатели, я заканчиваю последнюю тему нашего занятия в заочном мастер-классе. Как вы понимаете, говорить о разработке и настройке постпроцессоров можно долго. Я рассказал лишь мизерную часть того, о чем можно было бы рассказать. Не предлагаю вам точно следовать шагам, описанным в

этой теме, но выполнять основы программирования на языке GPPtool вам все равно придется. А поскольку у каждого "настройщика" свой стиль и опыт программирования, любые две разработки всегда отличаются друг от друга. Но тем и интересен этот процесс!

Скептикам и консерваторам, которые в тексте описания темы увидели только чуждые им английские слова и символы, я хочу сказать, что любое программирование подчиняется законам математики и логики. И, наверное, никто из вас не хотел бы видеть второй закон Ньютона и теорему Пифагора в таком виде ☺ (рис. 21).

Вот и закончился наш мастер-класс. Подведем некоторые итоги. Прошедшие занятия, как я очень надеюсь, убедили вас в том, что:

- ответственность за выполнение плана механическими участками лежит не только на технологах-программистах, но и на конструкторах, которые обеспечивают графическими данными технологические службы;
- использовать интегрированные CAD/CAM-решения — производительней и удобней;
- в InventorCAM (SolidCAM) можно

легко и быстро разрабатывать сложные процессы обработки не только трехмерных моделей, но и двумерных чертежей;

- InventorCAM (SolidCAM) предоставляет множество возможностей для создания, визуализации и проверки "виртуального" процесса обработки;
- разработка и настройка постпроцессоров на языке GPPtool — не только понятный и простой, но еще и увлекательный процесс.

Все предыдущие занятия я заканчивал тезисным изложением новых, добавленных и улучшенных функциональных возможностей программы, но в этот раз я этого делать не стану. В ближайших номерах CADmaster мы посвятим версии программного обеспечения InventorCAM 2008 полноценную статью. Следите за нашими публикациями! До новых встреч на страницах нашего журнала!

Андрей Благодаров
CSoft
Тел.: (495) 913-2222
E-mail: blag@csoft.ru

```
;-----
@korobka_peredach
;первая ступень
if spin > 0 and spin <= 315 then
mcode=41
endif
;вторая ступень
if spin > 315 and spin <= 630 then
mcode=42
endif
```

```
;третья ступень
if spin > 630 and spin <= 1250 then
mcode=43
endif
;четвертая ступень
if spin > 1250 then
mcode=44
endif
{nb'M'mcode}
endp
```