



ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
УПРАВЛЕНИЕ ДИСТАНЦИОННОГО ОБУЧЕНИЯ И ПОВЫШЕНИЯ
КВАЛИФИКАЦИИ

Кафедра «Металлорежущие станки и инструменты»

Учебное пособие

по дисциплине

«Технологии программирования станков с компьютерным управлением»

Авторы
Лебедев А. Р.,
Исаев А. Н.,
Гавриленко М. Д.

Ростов-на-Дону, 2017

Аннотация

Рассмотрена методика проектирования управляющих программ для станков с ЧПУ НААС. Приведены примеры программ, составлены индивидуальные задания и контрольные вопросы к изучению раздела «Системы числового управления технологическим оборудованием» курса «Управление станками и станочными комплексами». Отображена технология программирования высокого уровня.

Предназначено для бакалавров 4-го курса по направлению подготовки 15.03.05 «Конструкторско-технологическое обеспечение машиностроительных производств» и профилю подготовки «Металлообрабатывающие станки и комплексы», а также для обучающихся 1-го курса по направлению магистратуры 15.04.05 «Конструкторско-технологическое обеспечение машиностроительных производств», программа «Процессы механической и физико-технической обработки, станки и инструмент»

Авторы

к.т.н., доцент, моделирование процессов упрочняюще-калибрующей обработки

Лебедев А.Р.

д.т.н., профессор, технологии упрочняюще-калибрующей обработки

Исаев А.Н.

ст.преподаватель, исследование адаптивных систем автоматического управления в предохранительных функциональных муфтах

Гавриленко М.Д.

Оглавление

Введение	4
ГЛАВА 1. ОСНОВЫ ПРОГРАММИРОВАНИЯ УЧПУ НААС	8
Технические характеристики ЧПУ НААС.....	8
1.2. Структура управляющей программы	9
1.3. Программирование перемещений	11
1.4. Управление механизмами станка	19
1.5. Программные возможности системы ЧПУ.....	21
1.6. Стандартные циклы ЧПУ	26
Задания для самостоятельной работы	33
Контрольные вопросы	35
ГЛАВА 2. ТЕХНОЛОГИЯ ПРОГРАММИРОВАНИЯ ВЫСОКОГО УРОВНЯ ДЛЯ СЧПУ МЕТАЛЛОРЕЖУЩИМИ СТАНКАМИ.....	35
2.1. Элементы векторной геометрии.....	36
2.2. Программирование информации о геометрических элементах	36
2.3. Определение точек начала отсчета	37
2.4. Определение точек.....	38
2.5. Определение прямой линии.....	39
2.6. Определение окружности	40
2.7. Определение профиля.....	41
2.8. Пересечение между элементами	42
2.9. Пример программирования обработки профиля	43
Задание на выполнение самостоятельной работы	47
Контрольные вопросы	47
Заключение	48
Список литературы	48
ПРИЛОЖЕНИЕ 1	49
ПРИЛОЖЕНИЕ 2	53

ВВЕДЕНИЕ

Современные металлорежущие станки с числовым программным управлением – это машины для размерной обработки поверхностей деталей, органически соединенные с компьютером. Управление подобным оборудованием все более напоминает процесс составления компьютерной программы. Поэтому современный специалист в области машиностроительного производства должен несомненно обладать компетенциями в области программирования станком в ЧПУ.

Числовое программное управление (ЧПУ) – компьютеризованная система управления, управляющая приводами технологического оборудования, включая станочную оснастку. Оборудование с ЧПУ представлено: станочным парком, например станками для обработки металлов, дерева, пластмасс, для резки листовых заготовок, для обработки давлением и т.д., приводами асинхронных электродвигателей, использующих векторное управление; системой управления современными промышленными роботами [5].

Несколько станков с ЧПУ можно объединить в гибкую автоматизированную производственную систему (ГПС), которая в свою очередь может быть до-полнена гибким автоматизированным участком (ГАУ) и войти в состав автоматической линии (производства масштаба участка либо цеха), ГАП [2].

Изобретателем первого станка с числовым (программным) управлением (англ. Numerical Control, NC) является Джон Пэрсонс (John T. Parsons), работавший инженером в компании своего отца Parsons Inc, выпускавшей в конце Второй мировой войны пропеллеры для вертолетов. Он впервые предложил использовать для обработки пропеллеров станок, работающий по программе, вводимой с перфокарт.

В 1949 г. ВВС США профинансировали для Parsons Inc разработку станка для контурного фрезерования сложных по форме деталей авиационной техники. Однако компания не смогла самостоятельно выполнить работы и обратилась за помощью в лабораторию сервомеханики Массачусетского технологического института (MIT). Сотрудничество Parsons Inc с MIT продолжалось до 1950 г. В этом же году MIT приобрел компанию по производству фрезерных станков Hydro-Tel и отказался от сотрудничества с Parsons Inc, заключив самостоятельный контракт с ВВС США на создание фрезерного станка с программным управлением.

В сентябре 1952 г. станок был впервые продемонстрирован

публике – про него была напечатана статья в журнале Scientific American. Станок управлялся с помощью перфоленты.

Первый станок с ЧПУ отличался особой сложностью и не мог быть использован в производственных условиях. Первое серийное устройство ЧПУ было создано компанией Bendix Corp. в 1954 г. и с 1955 г. стало устанавливаться на станки. Широкое внедрение станков с ЧПУ шло медленно. Предприниматели с недоверием относились к новой технике. Министерство обороны США вынуждено было на свои средства изготовить 120 станков с ЧПУ, чтобы передать их в аренду частным компаниям.

Первыми отечественными станками с ЧПУ промышленного применения являются токарно-винторезный станок 1К62ПУ и токарно-карусельный 1541П. Эти станки были созданы в первой половине 1960-х годов. Станки работали совместно с управляющими системами типа ПРС-3К и другими. Затем были разработаны вертикально-фрезерные станки с ЧПУ 6Н13 с системой управления «Контур-3П». В последующие годы для токарных станков наибольшее распространение получили системы ЧПУ отечественного производства 2Р22 и Электроника НЦ-31.

Числовое программное управление также характерно для систем управления современными промышленными роботами.

Аббревиатура ЧПУ соответствует двум англоязычным – NC и CNC, – отражающим эволюцию развития систем управления оборудованием.

Системы типа NC (Numerical control), появившиеся первыми, предусматривали использование жестко заданных схем управления обработкой – например, задание программы с помощью штекеров или переключателей, хранение программ на внешних носителях. Каких-либо устройств оперативного хранения данных и управляющих процессоров не предусматривалось.

Более современные системы ЧПУ, называемые CNC (Computer numerical control) – системы управления, позволяющие использовать программные средства для модификации существующих и написания новых программ. Базой для построения CNC служит современный микропроцессор.

Структурно, в состав ЧПУ входят:

- пульт оператора (или консоль ввода/вывода), позволяющий вводить управляющую программу, задавать режимы работы; выполнять операцию вручную. Как правило, внутри шкафа пульта современной компактной ЧПУ, размещаются её остальные части;
- дисплей (или операторская панель) – для визуального

контроля режимов работы и редактируемой управляющей программы/данных; может быть реализован в виде отдельного устройства для дистанционного управления оборудованием;

- контроллер (интерполятор) – компьютеризированное устройство, решающее задачи формирования траектории движения режущего инструмента, технологических команд управления устройствами автоматике станка, общим управлением, редактирования управляющих программ, диагностики и вспомогательных расчетов (траектории движения режущего инструмента, режимов резания);

- ПЗУ – память, предназначенная для долговременного хранения (годы и десятки лет) системных программ и констант; информация из ПЗУ может только считываться;

- ОЗУ – память, предназначенная для временного хранения управляющих программ и системных программ, используемых в данный момент.

Важной характеристикой CNC-контроллера является количество осей (каналов), которые он способен синхронизировать; для этого требуется высокая производительность и соответствующее программное обеспечение.

После того как составлена управляющая программа, оператор при помощи программатора вводит ее в контроллер. Команды управляющей программы размещаются в ОЗУ. В процессе создания или после ввода управляющей программы оператор или программист может отредактировать ее, включив в работу системную программу редактора и выводя на дисплей всю или нужные части управляющей программы и внося в них требуемые изменения. При работе в режиме изготовления детали управляющая программа кадр за кадром поступает на выполнение. В соответствии с командами управляющей программы контроллер вызывает из ПЗУ соответствующие системные подпрограммы, которые заставляют работать подключенное к ЧПУ оборудование в требуемом режиме – результаты работы контроллера в виде электрических сигналов поступают на исполнительное устройство – приводы подач, либо на устройства управления автоматикой станка.

Управляющая система считывает инструкции специализированного языка программирования, который затем интерпретатором системы ЧПУ переводится из входного языка в команды управления главным приводом, приводами подач, контроллерами управления узлов станка. Для определения необходимой траектории движения рабочего органа в целом (инструмен-

та/заготовки) в соответствии с управляющей программой используется интерполятор, рассчитывающий положение промежуточных точек траектории по заданным в программе конечным точкам.

Программа для станка с ЧПУ может быть загружена с внешних носителей, например, магнитной ленты, перфоленты, дискеты или флеш-накопителей в собственную память либо временно, до выключения питания – в оперативную память, либо постоянно – в ПЗУ, карту памяти или другой накопитель: жёсткий диск или твердотельный накопитель. Помимо этого современное оборудование подключается к централизованным системам управления посредством заводских сетей связи. Наиболее распространенный язык программирования ЧПУ для металлорежущего оборудования описан документом ISO 6983 Международного комитета по стандартам и называется «G-код». В отдельных случаях – например, системы управления гравировальными станками – язык управления принципиально отличается от стандарта. Для простых задач, например, раскроя плоских заготовок, система ЧПУ в качестве входной информации может использовать текстовый файл в формате обмена данными, например DXF или HPGL.

Существует три метода программирования обработки станков с ЧПУ [2]:

1. Ручное программирование. Все операторы станков с ЧПУ, технологи-программисты должны иметь хорошее представление о технике ручного программирования. Это как начальные классы в школе, обучение в которых дает базу для последующего образования.

2. Программирование на пульте УЧПУ. Программы создаются и вводятся прямо на стойке ЧПУ с использованием клавиатуры и дисплея. Например, оператор станка может произвести верификацию УП или выбрать требуемый постоянный цикл при помощи специальных пиктограмм и вставить его в код управляющей программы.

3. Программирование при помощи CAD/CAM системы. Программирование при помощи CAD/CAM системы позволяет "поднять" процесс написания программ обработки на более высокий уровень. Работая с CAD/CAM системой, технолог-программист избавляет себя от трудоемких математических расчетов и получает инструменты, значительно повышающие скорость написания управляющих программ.

Цель данного учебного пособия – формирование навыков проектирования управляющих программ для системы ЧПУ HAAS.

ГЛАВА 1. ОСНОВЫ ПРОГРАММИРОВАНИЯ УЧПУ HAAS

Технические характеристики ЧПУ HAAS

Устройство числового программного управления (УЧПУ) HAAS (рис. 1) относится к числу наиболее развитых систем, позволяющих за счет богатых функциональных возможностей создавать гибкие станочные модули, легко встраиваемые в структуру современного автоматизированного производства.



Рис. 1. Внешний вид ЧПУ HAAS

Основные характеристики УЧПУ HAAS [1]:

- Количество управляемых координат – 5, из них по трем одновременно;
- Высокоскоростная обработка;

- Поворот и масштабирование системы координат;
- Программируемое смещение системы отсчета координат;
- Осевая и радиусная коррекция инструмента;
- 1 Мб памяти для программ с возможностью ее расширения;
- Три 32-разрядных процессора;
- Совместимость с G-кодом стандарта ISO;
- Контроль износа инструмента;
- Контроль нагрузки на инструмент;
- Винтовая интерполяция;
- Редактирование в фоновом режиме;
- Многооконный режим просмотра программы
- Калькулятор углов;
- Калькулятор скоростей и подач;
- Калькулятор для расчета дуг;
- Программирование в дюймах или в метрической системе
- Использование переменных;
- Разветвленные и циклические алгоритмы программирования;
- USB-порт;
- Интерфейс Ethernet;
- RS-232 порт - канал связи системы группового ЧПУ;
- Встроенные математические функции;
- Развитая система диагностики;
- Программируемое зеркальное отображение;
- 42 постоянных цикла обработки типовых поверхностей;
- 45 M-команд , 92 G-команды;
- Возможность управления внешними устройствами станка;
- Пробный прогон с графическим представлением;
- Многофункциональный маховичок управления;
- Гравировка;
- Жесткое нарезание резьбы метчиком;
- Управление точностью контурной обработки;
- Кнопочный выключатель для блокировки памяти;
- Всемирная сеть обслуживания.

1.2. Структура управляющей программы

Управляющая программа (УП) начинается с символа «%», далее следует имя программы – набор из 6 любых символов. Структура УП представлена на рисунке 2.

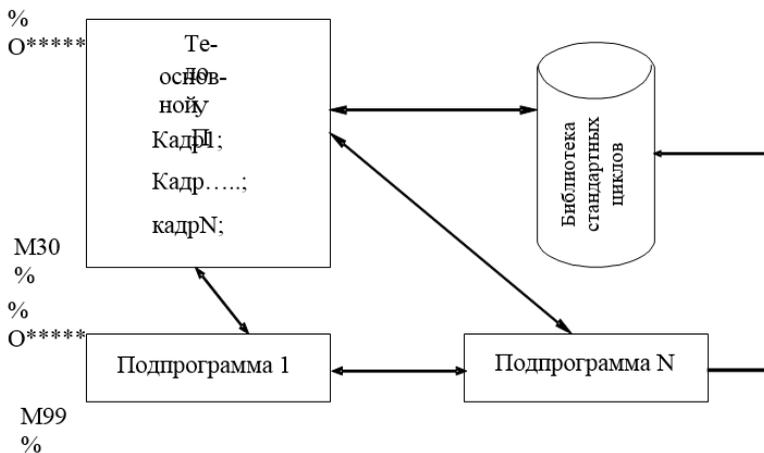


Рис. 2. Структура управляющей программы HAAS

Тело основной УП и подпрограмм представляет собой последовательность кадров. В отличие от многих других систем ЧПУ, нумерация кадров в программе HAAS не требуется. Каждый кадр программы представляет собой последовательность блоков и заканчивается символом «;». Блок кадра состоит из адресной и числовой части. Примеры блоков: X0, M99, G01, H5. В конце программы ставится кадр M30 (или M02) и символ «%».

Кроме основного тела УП, для многократного повторения однотипных операций, можно вводить подпрограммы, циклы, условные переходы, метки. Возможность передачи в подпрограммы управляемых параметров резко расширяет возможности программирования. Каждая подпрограмма начинается символом «O», далее следует имя подпрограммы. Признаком конца подпрограммы является кадр M99 и символ «%». Количество подпрограмм ограничивается памятью устройства ЧПУ.

Для обработки типовых поверхностей (отверстий, карманов, резьбы и т.д.) в ПЗУ системы существует библиотека стандартных циклов: G74, G81, G76 и т.д. Достаточно вызвать такой цикл, передать в него необходимые параметры типовой поверхности, и система выполнит целый комплекс действий по обработке, например глубокого отверстия.

Информация в кадре состоит из блоков, соответствующих командам на перемещения исполнительных органов станка (X,Y,Z,A,B), различные подготовительные команды (**G**) и вспомогательные функции (**M**). Например, **G00 X200Y500 M3**.

В кадре нельзя использовать повторно один и тот же адрес

(кроме «G»). Функции «G» могут повторяться в одном кадре, если они не противоречат логике управления и принадлежат к разным группам. Например, в кадре «G01G02X50Y100...» устанавливается режим линейной и круговой интерполяции одновременно.

Числовая часть адреса может быть представлена либо набором цифр, либо адресом #(*M*), указывающим на номер переменной в памяти устройства, в котором содержится число. Номера доступных для программирования переменных устройства #1 - #33. Например, кадры X1000Z-400; и X#30Z(#31- #32+#33); работают одинаково, если перед ними поместить следующие строки: #30=1000; #31=500; #32=-800; #33=-1700;

Естественный порядок отработки кадров (сверху вниз) в устройстве может быть изменен с помощью условных переходов, циклов, обращений к подпрограммам или стандартным циклам.

1.3. Программирование перемещений

1.3.1. Системы координат станка и детали

Программирование любого станка с ЧПУ начинается с изучения расположения и количества координатных осей конкретного станка. Например, в токарном или горизонтально-фрезерном станке оси будут расположены по-разному. Однако есть несколько общих правил, знание которых необходимо любому специалисту в этой области:

1. Положительное направление линейных координат всегда соответствует движению заготовки к инструменту.
2. Положительное направление угловых координат – по часовой стрелке.
3. Различают одну **систему координат станка** (все исполнительные органы находятся в исходном положении – «ноль станка») и **систему координат детали** («ноль детали»), определяющее положение заготовки, установленной в приспособление в рабочей зоне станка. Систем координат детали может быть несколько, что особенно удобно при обработке сложной детали или при использовании многоместного приспособления.

Расположение координатных осей для вертикально-фрезерного станка HAAS показано на рис. 3.

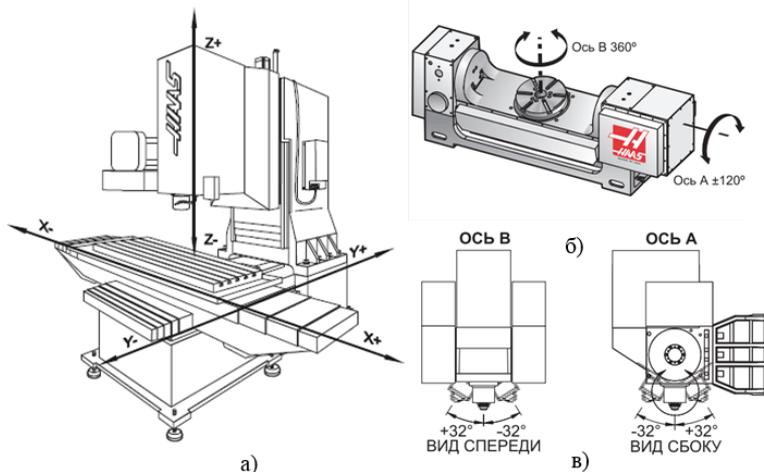


Рис. 3. Система координат фрезерного станка модели HAAS: а) – основные линейные координаты; б) – реализация угловых перемещений с применением поворотного стола HAAS; в) – угловые перемещения шпиндельной головки в некоторых моделях HAAS

Одна из основных функций наладчика станка с ЧПУ – установка нуля детали. При этом он совершает ряд манипуляций с кнопками пульта и приводами станка с целью выставить инструмент в желаемое нулевое положение и занести текущие значения координат в память системы ЧПУ (область параметров). При использовании измерительного щупа и специальных макрокоманд системы HAAS возможен поиск нулевой точки детали в автоматическом режиме (G35, G36, G37).

Для программиста системы HAAS важно помнить команды G54–G59. Причем в системе по умолчанию (не требует специального указания в программе) действует код G54 – первая система координат детали. Параметры каждой системы координат – значения линейных и (или) круговых координат, хранящиеся в специальной области памяти ЧПУ. Возврат к системе координат станка осуществляется указанием команды G53.

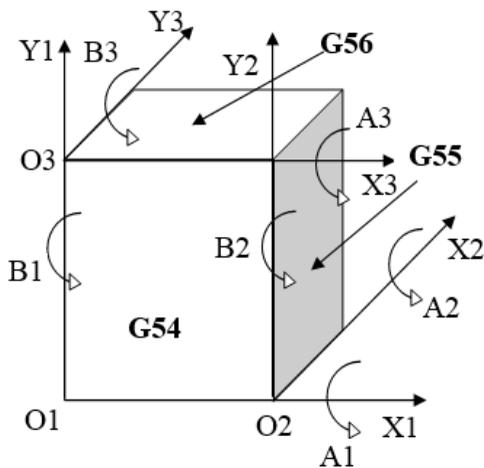


Рис. 4. Установка систем координат для заготовки типа куб

Рассмотрим пример применения команд установки нулей детали (рис. 4). Пусть в начале управляющей программы действует режим G54. Тогда отсчет всех координатных перемещений производится относительно точки O1 и мы работаем в плоскости X1Y1. Если оператор заблаговременно при наладке станка сообщил системе координаты и положение осей системы X2O2Y2, то переход к новой системе координат в программе задается указанием команды G55. Попробуйте определить, какие параметры X, Y, A, B должен ввести оператор, для того чтобы установить систему координат X3O3Y3 и какую команду для этого должен применить проектировщик управляющей программы.

Итак, система координат детали известна. Далее в этой системе необходимо перемещать инструмент или заготовку для реализации движений формообразования. Но какую часть перемещать? Введем в рассмотрение важное понятие. Программируемая точка (ПТ) – точка рабочей части инструмента, которую необходимо перемещать по заданным координатам для обработки поверхностей детали.

Перемещения могут быть заданы в абсолютной (G90) или относительной (G91) системах координат. В режиме G90 координаты ПТ указываются относительно установленного нуля детали. В режиме G91 координаты ПТ в текущем кадре указываются относительно ее положения в предыдущем кадре.

1.3.2. Быстрые перемещения

Движения позиционирования применяются для быстрого выхода инструмента в точку начала обработки. В кадр УП включается функция G00, действующая до отмены (включения G01, G02, G03).

Величины линейных координатных перемещений программируются в миллиметрах. Адреса перемещений: X, Y, Z.

Величины круговых координатных перемещений программируются в градусах. Адреса перемещений: A, B, C.

Скорость перемещений определяется параметром, установленным в настройках станка и, как правило, составляет 10000-20000 мм/мин. В режиме G00 при перемещении по нескольким координатам траектория ПТ не отслеживается.

1.3.3. Плоскости интерполяции

Интерполяция – способ приближенного вычисления значения величины, находящегося между двумя известными значениями. В системах ЧПУ применяется для вычисления линейных и круговых траекторий движения ПТ.

Плоскость интерполяции – плоскость, в которой происходит совместное движение программируемой точки по двум координатам. Различают линейную (G01) и круговую (G02, G03) интерполяцию. В координатном пространстве станка в текущий момент может быть установлена только одна из трех возможных плоскостей интерполяции. По умолчанию в системе действует режим G17 – плоскость XOY. Программист может сменить эту плоскость, указав G18 – плоскость XOZ или G19 плоскость YOZ. Определите ошибку программиста в следующем фрагменте управляющей программы: G54 G18 G01 X50 Y100.

1.3.4. Рабочие линейные перемещения

В режиме G01 ПТ совершает прямолинейное движение из исходной в заданную точку со скоростью рабочей подачи F. Система ЧПУ с помощью интерполяции гарантирует движение вдоль прямой линии с точностью, определяемой дискретностью системы ЧПУ (± 1 мкм). В кадр УП включается функция G01, действующая до отмены (включения G00, G02, G03).

Скорость подачи в режиме G94 (по умолчанию) программируется в мм/мин. В режиме G95 подача задается в мм/об.

1.3.5. Рабочие перемещения по дуге окружности

Круговая интерполяция обеспечивает перемещение по окружности и задается функциями G02 (по часовой стрелке) и G03 (против). Возможна в любой из трех координатных плоскостей XY, XZ, YZ. Кадр, описывающий перемещение по окружности, включает в себя:

- Указание плоскости интерполяции;
- Режим круговой интерполяции;
- Координаты конечной точки дуги;
- Координаты центра окружности (I, J, K) или радиус (R).

Дуга полностью определена в пространстве, если известны координаты двух ее точек, радиус или координаты центра. В случае задания координат центра дуги вводятся дополнительные адреса координат I, J, K. Правило их применения: I соответствует координате X, J – Y и K –Z. В случае применения команды R важно помнить, что для дуг менее 180° применяется положительное значение радиуса, а для дуг больше 180° - отрицательное. Для обхода полной окружности команда R неприменима.

1.3.6. Примеры перемещений

Рассмотрим задачу перемещения ПТ по точкам ИП-1-2-3-4-5-6--ИП, координаты которых известны (рис. 5).

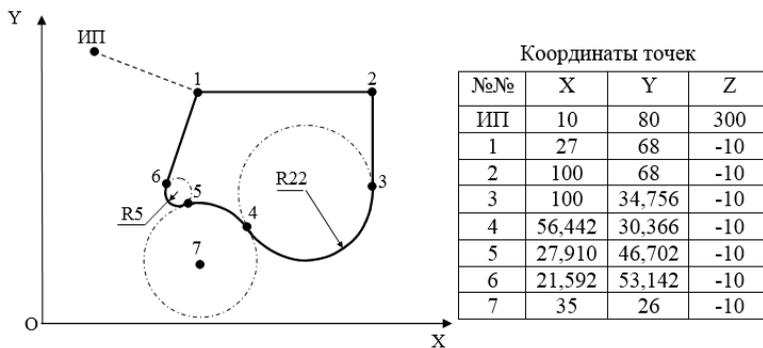


Рис. 5. Пример перемещения инструмента в плоскости XOY

Управляющая программа для данной задачи с пояснениями к каждому кадру приведена в таблице 1.

Таблица 1

Управляющая программа для примера на рис. 5

Кадр УП	Комментарии	Смена режимов по умолчанию G00, G90, G17, G94
1	2	3
O00001;	Заголовок программы	-
T01M06**;	Установка инструмента T1	
G54*G90*G94* F100 S500**;	Абсолютная система координат. Подача в мм/мин. Подача 100 мм/мин. Вращение шпинделя 500 об/мин	
G00* X27 Y68 Z-10;	Быстрый выход в точку 1	
G17*G01X100;	Плоскость интерполяции XOY. Линейная интерполяция. Выход в точку 2 на рабочей подаче	G01
Y34.756;	Выход в точку 3 на рабочей подаче	-
G02 X56.442 Y30.366 R22;	Круговая интерполяция по часовой стрелке. XY координаты точки 4. Радиус 22 мм.	G02
G03 X27.910 Y46.702 I35 J26;	Круговая интерполяция против часовой стрелки. XY координаты точки 4. Радиус 22 мм. I – XY координаты точки 7.	G03

Продолжение табл. 1

1	2	3
G02 X21.592 Y53.142 R5;	Круговая интерполяция по часовой стрелке. XY координаты точки 4. Радиус 5 мм.	G02
G01 X27 Y68;	Линейная интерполяция движения в точку 1.	G01
G00 X10 Y80 Z300;	Быстрый выход в ИП	G00
M5;	Остановка шпинделя	-
M30;	Конец программы	-

Примечание * - Команда может быть опущена, так как установлена по умолчанию

** - Информацию по командам приведена ниже

1.3.7. Коррекция на длину и радиус инструмента

Реальный инструмент – это не программируемая точка ПТ. Каждый технолог при оформлении эскиза операции ЧПУ обязан указать положение ПТ для конкретного инструмента. Коррекция решает задачу построения реальной (**эквидистантной**) траектории перемещения инструмента.

Радиусная коррекция (рис. 6) используется для учета радиуса инструмента при построении эквидистантной линии реального перемещения инструмента. Позволяет программировать контур обработки без учета диаметра осевого инструмента.

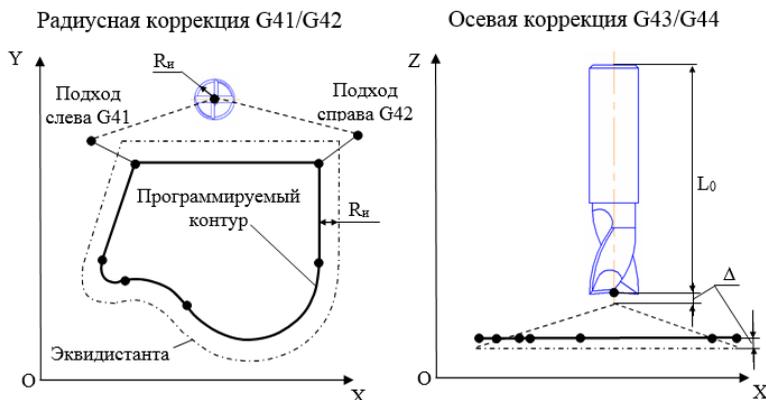


Рис. 6. Коррекция, ее виды и правила применения

Функция *G41* означает коррекцию радиуса инструмента при движении слева от контура, а *G42* – при движении справа от контура. Адрес *D* указывает на номер параметра, в котором содержится значение диаметра инструмента.

Функции *G41* или *G42* действуют до отмены. Отменяет радиусную коррекцию функция *G40*; Коррекцию *G41* или *G42* можно вводить только в кадрах с линейным перемещением.

Осевая коррекция используется для исправления величин перемещений параллельно одной из координатных осей. Значение коррекции (величина и знак) вводится в специально отведенную область памяти устройства. Функция *G43* означает ввод коррекции со своим знаком, а *G44* – с противоположным. Номер корректора задается в кадре под адресом *H*. Коррекция вводится перед адресом перемещения. Функции *G43* или *G44* действуют до отмены. Отмена коррекции по функции *G49*. Пример изображен на рис.6.

В отличие от предыдущего примера (см. табл.1, рис. 5), когда ставилась задача перемещения ПТ по заданным точкам, при обработке контура реальным инструментом необходимо осуществлять дополнительное движение на рабочей подаче при подходе к контуру. В следующем примере обрабатываем контур (см. рис. 5) с учетом коррекции на инструмент.

Пример. Радиусная и осевая коррекция (координаты точек на рис. 5).

O00002;	(Заголовок программы)
T01M06;	(установка инструмента в шпиндель)
F100 S500;	(режимы фрезерования)

X17 Y78; (быстрый выход в точку возле контура)
G43H1 Z-10; (осевая коррекция и выход в точку по Z)
G01G41D1 X27 Y68; (радиусная коррекция и выход на контур по XY)
X100; (линейное перемещение в точку 1)
Y34.756; (линейное перемещение в точку 2)
G02 X56.442 Y30.366 R22; (круговое движение в точку 3)
G03 X27.910 Y46.702 I35 J26; (круговое движение в точку 4)
G02 X21.592 Y53.142 R5; (круговое движение в точку 5)
G01 X27 Y68; (линейное перемещение в точку 6)
G40 G49 X17 Y78; (выключение коррекций и выход в точку возле контура)
G00 X10 Y80 Z300; (отвод инструмента)
M5; (выключение приводов)
M30; (конец программы)

1.3.8. Дополнительные команды по перемещению инструмента

Режущий инструмент в конечной точке траектории упруго смещается, что приводит к потере точности. Для предотвращения динамических ударов в кадре можно указать функцию точной остановки *G09*, действующей только в одном кадре. Перемещения станка замедлятся до заданной точки прежде, чем произойдет обработка следующей команды.

Функция *G04* используется для задержки или паузы в программе. Кадр, содержащий *G04*, сделает задержку на время, указанное в коде P. Например, *G04 P10.0*. Это вызовет паузу в программе длительностью 10 секунд. Обратите внимание на использование десятичной точки - *G04 P10*. означает задержку в 10 секунд; *G04 P10* означает задержку в 10 мс.

Функция *G28* используется для выведения в нуль станка всех (если не указаны оси) или только указанных осей станка. *G28* отменяет коррекцию на длину инструмента для последующих строк программы.

Функция *G51* используется для масштабирования траектории перемещений. Формат команды: *G51 [X...] [Y...] [Z...] [P...]*, где X, Y, Z – координаты центра масштабирования; если не указаны, то масштабирование осуществляется от последнего заданного положения инструмента; P - коэффициент масштабирования для всех осей (диапазон от 0.001 до 8383.0).

При задании масштабирования (*G51*) все значения X, Y, Z, I, J, K или R, относящиеся к перемещениям станка, умножаются на коэффициент масштабирования и смещаются относительно цен-

тра масштабирования. Масштабирование не влияет на значения коррекций на инструмент. Масштабирование не влияет на перемещение по оси Z в стандартных циклах, такие, как плоскость безопасного отвода и значения приращений. Функция G50 отменяет масштабирование.

Функция G68. используется для поворота траектории перемещений.

Формат команды: G68 [A...] [B...] [R...], где A, B – координаты центра поворота; R – угол поворота в градусах. Перед командой G68 необходимо использовать G17, G18 или G19, чтобы установить плоскость поворота.

Например, G17 G68 A...B... R50, A и B соответствуют осям текущей плоскости; для примера с G17 A – это ось X, а B – ось Y.

Система управления всегда использует центр поворота для определения значений положения, передаваемых системе управления после поворота. Если не задана ни одна из осей поворота, в качестве центра поворота используется текущее положение. При задании поворота (G68) все значения X, Y, Z, I, J и K поворачиваются на заданный угол R по отношению к центру поворота.

Функция G68 действует до отмены, то есть применяется ко всем командам перемещения в последующих блоках. При задании положительного значения (угла) адреса R поворот производится против часовой стрелки. Угол поворота устанавливается в ноль в начале программы, или же может быть установлен в конкретное значение с помощью G68 в режиме G90. Функция G69 отменяет режим поворота.

1.4. Управление механизмами станка

1.4.1. Управление скоростью перемещения рабочих органов станка

Скорость шпинделя программируется с помощью адреса S и числа оборотов в минуту. Вращение по часовой стрелке задается положительным числом. Код S запоминается и действует до нового кода.

Скорость движения подачи программируется адресом F и числом. Необходимо учитывать вид координаты и режим подачи. Например, F5 означает:

- В режиме G94 для линейной координаты 5 мм/мин;
- В режиме G95 для линейной координаты 5 мм/об;
- В режиме G94 для круговой координаты 5 град/мин;
- В режиме G95 для круговой координаты 5 град/об.

Запрограммированная подача запоминается и дей-

ствует до нового кода *F*.

1.4.2. Управление станком с помощью М-кодов

Формат М-кода – это буква “М”, за которой следуют две цифры, например M03. *В каждом кадре может быть запрограммирован только один М-код.* Все М-коды действуют в конце выполнения кадра. Перечислим некоторые важные М-коды.

M00 - остановка программы. Он останавливает оси, шпиндель, выключает СОЖ. При нажатии кнопки «Запуск цикла» выполнение программы продолжится с выделенного кадра.

M02 - конец программы.

M03 - включает движение шпинделя в направлении вперед.

M04 - включает движение шпинделя в направлении назад.

M05 - останавливает шпиндель.

M06 - смена инструмента. Например, M06 T12 устанавливает в шпиндель инструмент 12. Если шпиндель работает, то шпиндель и СОЖ будут остановлены командой M06.

M08 - включение СОЖ.

M09 - выключение СОЖ.

M19 - ориентация шпинделя. Используется для установки шпинделя в определенное положение. Без дополнительной функции ориентации шпинделя M19 шпиндель только будет ориентирован в нулевое положение. Дополнительная функция Orient Spindle (Ориентация шпинделя) делает возможным использование адресных кодов P и R. Например, M19 P270 ориентирует шпиндель в положение 270 градусов. Величина R позволяет программисту указать до четырех десятичных разрядов, например, M19 R123.4567.

M30 - конец программы и перемотка. Код останавливает шпиндель и выключает СОЖ, и курсор возвращается к началу программы. M30 отменяет коррекции на длину инструмента.

M31 - запускает двигатель конвейера уборки стружки. Конвейер не будет работать, если открыта дверь. Рекомендуется использовать конвейер стружки с перерывами. Непрерывная работа приведет к перегреву двигателя.

M33 - останавливает движение конвейера уборки стружки.

Команды управления паллетами. M36 - готовность детали для смены паллеты. Этот М-код задерживает смену паллеты до отпускания клавиши Part Ready (Деталь готова). Смена паллеты по команде M50 происходит при отпуске клавиши готовности к смене паллеты (и закрытых дверях). Например:

Onnnnn (номер программы)

M36 (Мерцает лампочка "Part Ready", ожидает нажатия клавиши)

M50 (Выполняет смену паллеты после нажатия клавиши «Part Ready»).

M30

M80/M81 - Открывание / закрывание автоматической двери. M80 открывает автоматическую дверь, а M81 закрывает автоматическую дверь. Подвесная клавиатура управления будет издавать звуковой сигнал, когда дверь будет двигаться.

1.5. Программные возможности системы ЧПУ

1.5.1. Переменные

Переменные – это десятичные значения, представленные в виде чисел с плавающей точкой. Существует три вида переменных: *системные, глобальные и локальные.*

Локальные переменные находятся в диапазоне от #1 до #33. Набор локальных переменных доступен в текущем структурном блоке управляющей программы (тело программы, подпрограмма).

Глобальными называются переменные, которые доступны всегда. Каждая глобальная переменная существует всего в одном экземпляре. Глобальные переменные находятся в трёх диапазонах: #100-#199, #500-#699 и #800-#999. При выключении питания глобальные переменные сохраняются в памяти.

Системные переменные дают программисту возможность полного управления системой ЧПУ. Для ознакомления с этими возможностями обратитесь к руководству по программированию ЧПУ [1].

Переменные можно использовать вместо адресов G-кодов, когда "адреса" обращаются к буквам A-Z. В блоке N1 G0 G90 X1.0 Y0; переменным можно присвоить следующие значения: #7=0; #11=90; #1=1.0; #2=0.0; и заменить блок на: N1 G#7 G#11 X#1 Y#2;

Значения переменных во время выполнения программы используются как значения адресов. Косвенные ссылки на переменные можно задавать, заключая номер переменной в скобки. Выражение, вычисляется и его результат становится номером переменной. Например: #1=3; #[#1]=3.5 + #1;. Это выражение присваивает переменной #3 значение 6.5.

1.5.2. Операторы, функции, выражения

Арифметические операторы. К ним относятся:
Сложение +; Вычитание - ; Умножение *; Деление /; Остаток MOD.

Логические операторы. К ним относятся:
OR - логическое ИЛИ; XOR - исключающее ИЛИ; AND - логическое И;

Булевы операторы. К ним относятся:
EQ – равно; NE - не равно; GT - больше чем; LT - меньше чем; GE - больше чем или равно; LE - меньше чем или равно.

Функции – это встроенные программы, которыми может пользоваться программист. Все функции имеют следующую форму: <имя_функции> [аргумент], и возвращают десятичные значения с плавающей точкой. В систему управления HAAS включены следующие функции: SIN[число в градусах] - синус; COS[] косинус; TAN[] – тангенс; ATAN[] – арктангенс; SQRT[число] квадратный корень числа; ABS[] - модуль числа; ROUND[] – округление до большего целого; FIX[] целая часть числа; ACOS[] - арккосинус; ASIN[]- арксинус; DPRNT[]. Текст в кодировке ASCII.

Выражение (логическое или арифметическое) строится из переменных и операторов. Попробуйте вычислить результат выполнения следующего арифметического выражения:

```
#1=2;#2=3;  
#30=ROUND[ABS[#1+#2]/(COS[#1]*SIN[#2])];
```

Примеры логических выражений приведены в п. 1.5.3.

1.5.3. Условные и безусловные переходы

Для реализации разветвленных алгоритмов управления применяются операторы перехода.

Безусловный переход всегда осуществляется к указанной метке блока. Формат метки – несколько символов латинского алфавита. Команда GOTO15 выполняет переход к блоку, имеющему метку N15. В системе управления Haas команда GOTO может использоваться в одной строке с другими G-кодами.

Условный переход позволяет передавать управление другому фрагменту кода в структурном блоке программы. Формат оператора: IF [<условное выражение>] THEN [оператор].

В примере

```
#1=5.0; #2=3.0; IF [#1 GT 3.0] AND [#2 LT 10] GOTO1
```

управление перейдет к метке 1, поскольку #1 GT 3.0 даёт в результате 1.0 (ИСТИНА), а #2 LT 10 даёт 1.0 (ИСТИНА), в результате 1.0 AND 1.0 равно 1.0 (ИСТИНА), значит, выполняется команда перехода GOTO 1.

1.5.4. Циклы

Важной характеристикой любой системы программирования является способность исполнять последовательности операторов заданное число раз или выполнять циклы операторов до достижения некоторого условия. Формат цикла:

WHILE [*<условное выражение>*] *DO*n; *<операторы>*;
*END*n;

Выполняются операторы, заключённые между *DO*n и *END*n, пока условное выражение остаётся истинным. Если выражение становится ложным, то следующим выполняется блок, следующий за *END*n. Вместо *WHILE* можно использовать краткое написание *WH*. Команды *DO*n-*END*n являются парными. Значение n равно 1-3. Это значит, что в подпрограмме может быть не более трёх вложенных циклов.

Пример вложенного построения циклов *WHILE* для определения матрицы:

#101= 3; #102= 4; (количество строк и столбцов матрицы)

#103= 40; #104= 60 (шаг строк по X, шаг строк по Y)

#105= 50; #106= 100 (исходная точка матрицы)

G00 X#105 Y#106 F25; (выход в исходную точку матрицы)

WH [#101 GT 0] DO1; (повторять пока #101 больше нуля)

#102= 4; #104=#106; (возврат счетчика столбцов, выход в исходный Y)

WH [#102 GT 0] DO2; (повторять пока #102 больше нуля)

G81 X#103 Y#104 Z-0.5; (Цикл сверления в текущей точке)

#104= #104+#104 (приращение по координате Y)

#102= #102 - 1; (уменьшение счетчика столбцов)

END2; (метка конца внутреннего цикла)

#103=#103+#103 (приращение по координате X)

#101= #101 - 1; (уменьшение счетчика строк)

END1; (метка конца внешнего цикла)

M30; (конец программы)

Эта управляющая программа осуществляет сверление матрицы отверстий 3х4 с расстояниями между строками 40 мм, между столбцами 60 мм.

Для выхода из фрагмента, ограниченного *DO*-*END*, можно использовать *GOTO*, но с помощью *GOTO* нельзя перейти внутрь цикла. Переход в пределах цикла, ограниченного *DO*-*END*, с помощью *GOTO* разрешается.

1.5.5. Подпрограммы

Подпрограмма – это специальным образом оформленная последовательность кадров, которая может быть вызвана на обработку из любого места УП. Ее применение становится эффективным тогда, когда требуется выполнить несколько раз, но с разными условиями, одну и ту же последовательность действий. Например, фрезеровать несколько отверстий разного диаметра или обработать несколько деталей в многоместном приспособлении.

Локальная подпрограмма может быть организована внутри основной программы. Вызывается следующим кодом:

M97 P(номер строки) L (число повторений).

Подпрограмма помещается в конце программы и обязательно отделяется признаком конца программы M30 или M02. Подпрограмма должна заканчиваться M99. В рассмотренном ниже примере подпрограмма фрезерования отверстия вызывается 3 раза. Положение центра отверстия задается сменой локальной системы координат G54, G55, G56. Управляемый параметр – глобальная переменная #105.

Пример вызова подпрограммы с кодом M97

O000001	(заголовок основной программы)
T01 M06;	(загрузка инструмента в шпиндель станка)
#105=53;	(присвоение значения 53 переменной #105)
M97 P1000 L3;	(подпрограмма N1000 выполняется 3 раза)
M30;	(конец основной программы)
	(текст локальной подпрограммы)
N1000 #105=#105+1;	(метка строки 1000, увеличение на 1 переменной)
G00 G90 G[#105] X0 Y0;	(выход в 0 по X и Y новой координатной системы)
S500 F100.M03;	(вращение шпинделя 500 мин ⁻¹ , подача 100 мм/мин)
G43 H01 Z5;	(ввод осевой коррекции на инструмент)
G01 Z-5	(заглубление фрезы на требуемый уровень по Z)
G41 X50;	(ввод радиусной коррекции, выход на контур)

G03 X50 Y50 I0 J0; (по кругу относительно нулевой точки)

G01 X0 G40; (отход по X, выключение радиусной коррекций)

G28 Z0; (отвод в 0 станка по оси Z, отмена осевой коррекции)

M99; (конец подпрограммы)

Внешняя подпрограмма по коду M98 вызывается из основной программы. Формат команды: M98 Pnnnn (Pnnnn - это номер вызываемой программы). Подпрограмма должна быть в списке программ, и она должна содержать M99 для возврата в главную программу. На одной строке с M98 можно поставить счетчик Lnn, что приведет к вызову подпрограммы nn раз перед переходом к следующему блоку.

O0001 (номер главной программы)

M98 P100 L4; (вызов подпрограммы, номер подпрограммы, цикл 4 раза)

M30 (конец программы)

O0100 (номер подпрограммы)

....

M99

Команда G65 вызывает подпрограмму и может передавать ей аргументы. Используется следующий формат:

G65 Pnnnn [Lnnnn] [аргументы];

Всё, что выделено курсивом в квадратных скобках, является необязательным. Команда G65 требует P-адреса, соответствующего номеру программы, находящейся в текущий момент в памяти блока управления. При использовании L-адреса обращение к макрокоманде повторяется заданное число раз.

Вызовы G65 близки к вызовам M98, но не идентичны им. Вызовы G65 могут иметь вложенность до 4 уровней, это значит, что программа 1 может вызвать программу 2, которая может вызвать программу 3, а та в свою очередь – вызвать программу 4.

Пример: G65 P1000; (вызов подпрограммы 1000 как макропрограммы)

M30; (останов программы)

O1000; (макроподпрограмма)

...

M99; (Возврат из макроподпрограммы)

При передаче параметров можно применять алфавитную адресацию переменных (табл. 2). В следующем примере подпрограмма 9010 предназначена для сверления последовательности

отверстий вдоль линии, наклон которой определяется аргументами X и Y, которые ей передаются в строке команды G65.

Таблица 2

Алфавитная адресация переменных

Адрес Переменная:	A 1	B 2	C 3	D 7	E 8	F 9	G -	H 11	I 4	J 5	K 6	L -	M 13
Адрес: Переменная:	N -	O -	P -	Q 17	R 18	S 19	T 20	U 21	V 22	W 23	X 24	Y 25	Z 26

Глубина сверления Z передается как Z, скорость подачи передается как F, а количество отверстий передается как T. Линия отверстий сверлится, начиная от текущего положения инструмента в момент вызова макроподпрограммы.

G00 G90 X10 Y10 Z0 S1000 M03; (расположение инструмента)

G65 P9010 X10 Y30 Z-5 F20. T10; (вызов 9010 с параметрами X,Y,Z,F,T)

G28; (выход в 0 станка)

M30; (конец основной программы)

(согласно табл.2 #9=20; #20=10; #24=10; #25=30; #26=-

5);

O9010; (внешняя подпрограмма O9010)

F#9; (F=Скорость подачи, #9=20)

WHILE [#20 GT 0] DO1; (цикл повторять #20=10 раз)

G91 G81 Z#26; (сверлить до глубины Z=-5)

#20=#20-1; (обратный отсчёт)

IF [#20 EQ 0] GOTO5; (все отверстия просверлены ?)

G00 X#24 Y#25; (двигаться вдоль наклона)

N5 END1; (метка конца цикла)

M99; (возврат к месту вызова)

1.6. Стандартные циклы ЧПУ

Стандартные циклы – это фактически готовые внешние подпрограммы, встроенные в структуру математического обеспечения устройства ЧПУ. Приведем перечень таких циклов:

G35 - автоматическое измерение диаметра инструмента;

G36 - автоматическое измерение центра смещения детали;

G37 - автоматическое измерение смещения инструмента;

G47 - гравировка текста;

G70 - круг болтовых отверстий;

G71 - дуга болтовых отверстий;

G72 - болтовые отверстия вдоль угла;

G73 - цикл скоростного сверления со ступенчатой подачей;

- G74** - цикл реверсного нарезания резьбы;
- G76** - цикл чистовой расточки;
- G77** - цикл растачивания в обратном направлении;
- G80** - отмена стандартного цикла;
- G81** - цикл сверления;
- G82** - цикл засверливания;
- G83** - цикл обычного сверления со ступенчатой подачей;
- G84** - цикл нарезания резьбы;
- G85** - цикл расточки;
- G86** - стандартный цикл расточки и останова;
- G87** - стандартный цикл расточки и ручного отвода;
- G88** - стандартный цикл расточки, паузы и ручного отвода;
- G89** - стандартный цикл расточки, паузы и отвода.

Выбранный стандартный цикл действует до отмены его кодами G80, G00 или G01. Рассмотрим некоторые интересные циклы системы ЧПУ.

1.6.1. Циклы сверления G81 и G83

Если отверстие не глубокое ($L < 3D$), то можно применить цикл **G81** со следующими параметрами (рис. 7): F - скорость подачи; L - количество просверливаемых отверстий при использовании G91 (режим приращений); R - положение плоскости R (расстояние от поверхности детали); X - команда перемещения по оси X; Y - команда перемещения по оси Y; Z - координата оси Z на дне отверстия.

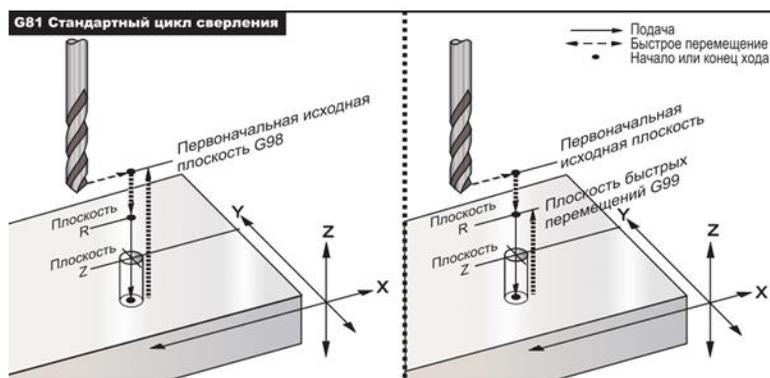


Рис. 7. Стандартный цикл сверления G81

Рассмотрим пример программы сверления пяти отверстий в пластине

T1 M06 - установка сверла в шпиндель;

G00 G90 G54 X10 Y20 - выход в точку над первым отверстием;

S4500 M03 - вращение шпинделя 4500 мин⁻¹;

G43 H01 Z5 - осевая коррекция на сверло;

G81 Z-5 R5 F200. - цикл сверления в точке X10 Y20 на глубину Z-5;

X20 - цикл сверления в точке X20 Y20 на глубину Z-5;

X30 Y40 - цикл сверления в точке X30 Y40 на глубину Z-5;

X45 Y50 - цикл сверления в точке X45 Y50 на глубину Z-5;

X50 Y60 - цикл сверления в точке X50 Y60 на глубину Z-5;

G80 G00 Z5 - отмена цикла, быстрый выход в Z 5 мм;

G28 - выход в 0 станка;

M30 - конец программы.

Для обработки глубокого отверстия применяется цикл **G83** с расширенным набором параметров: F – скорость подачи; I – глубина заглабления сверла на первом проходе; J – величина уменьшения глубины резания на каждый проход; K – минимальная глубина резания; L – количество отверстий при использовании G91 (режим приращений); P – пауза в конце последнего прохода (в секундах); Q – глубина резания (всегда относительная); R – положение плоскости R (расстояние от поверхности детали); X, Y – координаты центра отверстия; Z – координата оси Z на дне отверстия (рис. 8).

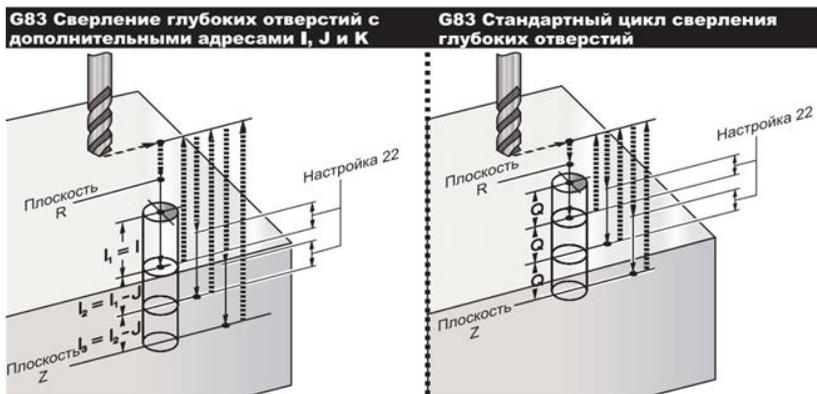


Рис. 8. Стандартный цикл сверления G83

При указании адресов I, J и K в процессе первого прохода происходит сверление на глубину I, каждый последующий проход сокращается на величину J, а минимальная глубина резания равна K. В следующем примере

G83 X10 Y20 Z-64 F20. R5 I20 J4 K4 P1.5

сверло на первом проходе пройдет 20 мм, на последующих проходах соответственно 16, 12, 8, 4 мм.

При указании адреса Q величины проходов одинаковы и не меняются:

G83 X10 Y20 Z-155 F20. R5 Q10 P1.5.

Последний проход рассчитывается автоматически. Попробуйте рассчитать количество проходов и величину последнего прохода для вышеприведенного примера.

В случае указания адреса R инструмент будет задерживаться на дне отверстия на указанное время. В приведенных примерах сверление выполняется за несколько проходов с паузой 2 с. Для всех последующих блоков не надо указывать продолжительность задержки, поскольку она повторяется.

1.6.2. Контактное измерение

Контактное измерение значительно расширяет возможности станка. Вот несколько примеров:

- 1) Профилирование детали с целью определения неизвестных размеров для последующей обработки;
- 2) Калибровка инструментов для учёта значений коррекции и износа;
- 3) Обследование перед обработкой для определения размера снимаемого припуска;
- 4) Обследование после обработки для определения параметров параллельности и плоскостности, а также положения поверхностей.

Положение измерительного щупа определяется значениями координат X, Y, Z, A и B. Если щуп касается чего-либо в точке, то система ЧПУ вырабатывает «сигнал пропуска». Таким образом, наш станок с ЧПУ превращается в координатно-измерительную машину. Следует помнить номера ячеек памяти, в которых хранятся следующие сведения:

- #5021-#5025 - *текущие координаты станка*. Текущее положение координат станка можно получить с помощью #5021-#5025, X, Y, Z, A и B, соответственно. Эти значения НЕЛЬЗЯ прочитать, когда станок находится в движении. Значение #5023 (Z) даётся с учётом коррекции на длину инструмента.
- #5041-#5045 - *текущие рабочие координаты*. Текущее положение в текущих рабочих координатах можно получить с помощью #5041-#5045, X, Y, Z, A и B, соответственно. Эти значения НЕЛЬЗЯ прочитать, когда станок

находится в движении. Значение #5043 (Z) даётся с учётом коррекции на длину инструмента.

- #5061-#5065 Текущая позиция сигнала пропуска. *Позиция, в которой срабатывал последний сигнал пропуска, можно получить с помощью #5061-#5065 для X, Y, Z, A и B, соответственно.* Значения даются в текущей рабочей системе координат и могут использоваться, когда станок находится в движении. Значение #5063 (Z) даётся с учётом коррекции на длину инструмента.

При измерении задается специальная команда G31 – подача до пропуска со следующими параметрами: F- скорость подачи; X, Y, Z, A, B – команды перемещения по осям. Этот G-код перемещает оси в заданную позицию. Перемещение начинается и продолжается до достижения заданной позиции, или до получения сигнала датчика касания. По завершении перемещения выдается звуковой сигнал.

В качестве примера рассмотрим задачу определения фактического припуска при обработке наружного контура (рис. 9). Для этого измерительный щуп последовательно выводим в точки P1-P8 и направляем к точкам контура.

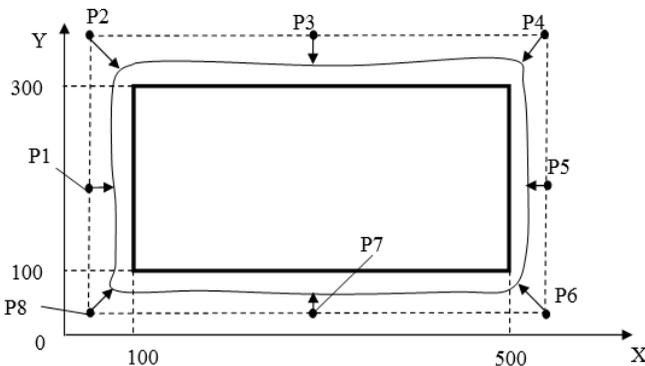


Рис. 9. Определение максимального припуска при фрезеровании контура

После касания щупа с заготовкой вычисляем фактический припуск. Из полученных значений припусков выбираем наибольшее значение. Значение печатается на экран пользователя (команда DPRINT).

T10 M06 (установка датчика касания в шпиндель)
G00 G90 G54 X50 Y200 (выход в точку P1)
Z10 (установка датчика по высоте контура)
M53 (включение датчика касания)
#20=0 (наибольший припуск равен нулю)

```

G31 F400 X100 (движение до касания из точки P1)
G04 P100 (выдержка времени)
#20=100-#5061 (фактический припуск в точке P1)
G00 X50 Y350 (выход в точку P2)
G31 F400 X100 Y300 (движение до касания из точки P2)
G04 P100 (выдержка времени)
#21=[100-#5061]+[#5062-300] (фактический припуск в
точке P2)
IF [#21 GT #20] #20=#21 (определяем наибольший
припуск)
.....
#21=[100-#5061]+[100-#5062] (фактический при-
пуск в точке P8)
IF [#21 GT #20] #20=#21 (определяем наибольший
припуск)
DPRINT[<Наибольший припуск для данного конту-
ра><#20[33]>]
M63 (выключение датчика касания)
G28 (выход в 0 станка)
M30 (конец программы)

```

1.6.3. Гравировка текста G47

Для использования G47 программа должна использовать режим G90 (абсолютный): E – скорость погружения (ед./мин.); F – скорость гравирования (ед./мин.); I – угол поворота (от -360° до +360°), по умолчанию 0; J – высота текста в дюймах (минимум = 0,001 дюйма), по умолчанию 1,0 дюйм; P0 для гравировки строки текста; P1 для гравировки последовательного серийного номера; P32-P126 для символов ASCII; R – плоскость возврата; X – начало гравировки по оси X; Y – начало гравировки по оси Y; Z – глубина резания.

Гравировка последовательного серийного номера используется для гравировки номеров на серии деталей, с увеличением номера каждый раз на единицу. Символ # используется для выбора количества цифр в серийном номере, например: G47 P1 (###) ограничивает серийный номер четырьмя цифрами.

Начальный серийный номер может быть задан как программно, так и вручную. К примеру, если серийный номер задан программно: G47 P1 (1234) устанавливает начальный серийный номер "1234".

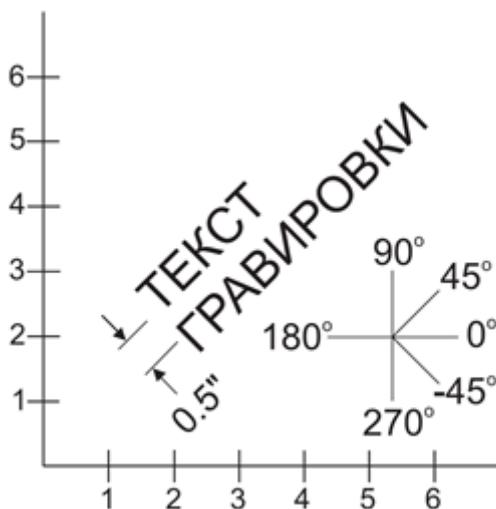


Рис. 10. Пример гравировки

Начальный серийный номер может также быть задан вручную через макропеременную. Для этого не обязательно должна быть включена функция макрокоманд. Значение начального серийного номера для гравировки содержится в макропеременной #599. Например, если макропеременная #599 установлена в "1234", то G47 P1 (####) выдаст: 1234

Гравировка строки текста применяется для гравировки на детали требуемой строки текста. Текст должен идти в виде комментария в той же строке, что и оператор P0. Данный пример создаст рисунок (рис. 10).

G47 P0 X2.0 Y2.0 I45. J.5 R.05 Z-.005 F15.0 E10.0
(ТЕКСТ ГРАВИРОВКИ)

1.6.4. Цикл нарезания резьбы G84

G84 Стандартный цикл нарезания резьбы (рис. 11): F – скорость подачи в дюймах (или мм) в минуту; J – многократный отвод; L – количество отверстий при использовании G91 (режим приращений); R – положение плоскости R (расстояние от поверхности детали); X Y – координаты центра отверстия; Z – координата Z на дне отверстия.

Нарезание резьбы с помощью стандартного цикла требует расчета скорости подачи. Формула расчета скорости минутной

подачи:

$S(\text{мм/мин}) = n_{\text{шп}} \cdot t$ (частота вращения шпинделя * t(шаг резьбы)).

Пример программы

T3 M06 (метчик M16x1,5)
 G90 G54 G00 X10 Y-20 (начальная точка обработки)
 S900 M03 (вращение шпинделя 900 об/мин)
 G43 H03 Z20 M08 (осевая коррекция на инструмент,

СОЖ)

(900 об/мин умножить на шаг резьбы 1,5 мм = 1350 мм/мин)

G84 Z-20 R10 F1350 (параметры цикла)
 X20 Y-20 (нарезание резьбы в точке 1)
 X20 Y20 (нарезание резьбы в точке 2)
 X-20 Y-20 (нарезание резьбы в точке 3)
 X-20 Y20 (нарезание резьбы в точке 4)
 G80 G00 Z10 M09 (отмена цикла, подъем метчика,

выключение СОЖ)

G28 Y0 Z0 (отвод инструмента)
 M30 (конец программы)
 %



Рис. 11. Параметры стандартного цикла нарезания резьбы G84

ЗАДАНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

Варианты заданий приведены в табл. 3. В плоскости интерполяции A постройте контур детали типа «кулачок» с учетом возможности его обработки концевой фрезой диаметром D. Эскиз контура высотой B с проставленными координатами опорных то-

чек и радиус концевой фрезы составляет ваши исходные данные к работе. Возможные варианты контуров приведены в [3]. Режимы резания при составлении программ можно принимать с помощью [4].

Исходные данные необходимо готовить, выдерживая ряд условий: число опорных точек не менее 9; обязательно наличие дуг окружностей не менее 3-х. Координаты опорных точек могут быть найдены из чертежа, как расчетным, так и графическим способом. На плоскости детали необходимо добавить дополнительные элементы С и Е.

Составить управляющую программы обработки детали. С помощью компьютерного тренажера «HAAS» (приложение 1) необходимо осуществить все команды по установке координатной системы детали, привязке инструментов, включить необходимые механизмы станка для смены инструмента, подачи и вращения шпинделя. Ввести управляющую программу в память УСПУ. Получить на экране дисплея изображение траектории инструмента (см. приложение). Оценить правильность отработки программы. Ответить на контрольные вопросы.

Таблица 3

Варианты заданий для самостоятельной работы

№ вар.	A	B	C (гравировка)	D	E (доп. элемент)
1	ХОУ	20	ФИО	10	Линия «Синусоида»
2	ХОУ	22	ФИО	10	Линия «Косинусоида»
3	ХОУ	24	ФИО	10	Линия «Спираль»
4	ХОУ	26	ФИО	10	Линия «Окружность»
5	ХОУ	28	ФИО	10	Линия «Порабола»
6	ХОУ	30	Номер группы	15	Отверстие сквозное Ø30H7
7	ХОУ	32	Номер группы	15	Отверстие сквозное Ø20H8
8	ХОУ	34	Номер группы	15	Отверстие сквозное Ø40E8
9	ХОУ	36	Номер группы	15	Отверстие сквозное Ø18H7
10	ХОУ	38	Номер группы	15	Отверстие сквозное Ø25H9
11	YOZ	10	«ДГТУ»	20	Матрица 5x5 отверстий Ø10H14
12	YOZ	12	«ДГТУ»	20	Матрица 4x5 отверстий Ø10H14
13	YOZ	14	«ДГТУ»	20	Матрица 3x8 отверстий Ø10H14
14	YOZ	16	«ДГТУ»	20	Матрица 3x4 отверстий Ø10H14
15	YOZ	18	«ДГТУ»	20	Матрица 4x3 отверстий Ø10H14
16	YOZ	20	«РОССИЯ»	8	Контроль 1-го размера
17	YOZ	22	«РОССИЯ»	10	Контроль 2-х размеров
18	YOZ	24	«РОССИЯ»	12	Контроль 3-х размеров
19	YOZ	26	«РОССИЯ»	14	Контроль высоты детали
20	YOZ	28	«РОССИЯ»	16	Контроль высоты детали
21	XOZ	30	Номер с 1248	15	Контроль параллельности
22	XOZ	32	Номер с 1	15	Контроль перпендикулярности
23	XOZ	34	Номер с 0025	15	Контроль плоскостности
24	XOZ	36	Номер с 222.258	15	Линия «Синусоида»
25	XOZ	38	Номер с 11111	15	Линия «Косинусоида»

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Приведите примеры использования в реальных условиях гибкого производства таких элементов УП, как подпрограммы, переходы по условию, постоянные циклы.
2. Составьте фрагмент УП, в котором отрабатывается траектория окружности.
3. Составьте подпрограмму, при каждом обращении к которой инструмент обходит окружность все большего радиуса.
4. Найдите ошибки в следующих кадрах УП:
N5 G18 G90 G02 X200 Y300 I100 J150;
N5 G90 G00 G01 X300;
N5 G43H01 Y80 G40;
5. Назовите основные структурные элементы системы ЧПУ.
6. Что такое плоскость интерполяции и зачем она нужна?
7. Перечислите три метода программирования обработки на станке с ЧПУ.
8. Перечислите несколько возможных вариантов применения контактных измерений с помощью измерительного щупа.
9. Каким образом применение переменных в управляющей программе повышает эффективность написания программного кода?
10. Назовите примеры эффективного использования функций масштабирования и поворота в управляющей программе.

ГЛАВА 2. ТЕХНОЛОГИЯ ПРОГРАММИРОВАНИЯ ВЫСОКОГО УРОВНЯ ДЛЯ СЧПУ МЕТАЛЛОРЕЖУЩИМИ СТАНКАМИ

В системе УЧПУ NC-201 представляется возможным в программе описать геометрический профиль в плоскости, используя не только стандартный язык программирования (G1-G2-G3), но и язык программирования высокого уровня GTL. В переводе GTL – геометрический технологический язык, разработанный итальянской компании Olivetty и ставший стандартом программирования обработки на станках с ЧПУ. Этот язык позволяет программировать профиль, состоящий из прямых окружностей, используя только информацию, полученную с чертежа. Система сама вычисляет точки пересечения и точки касания геометрических элементов.

Язык программирования GTL и язык на основе ISO-кодов могут быть использованы одновременно в одной и той же программе, но не для одного и того же профиля. Геометрия GTL функционирует только при абсолютном программировании (G90).

2.1. Элементы векторной геометрии

Определение профиля с использованием GTL основано на использовании четырёх типов геометрических элементов: точки начала отсчета (СК), точки, прямые, окружности. Так как профиль определяется как геометрическими элементами, так и направлением, то для определения геометрических элементов в языке GTL используется векторная геометрия, для которой определение элемента, кроме параметров, необходимых для установления позиции в плоскости, требуется назначение направления движения.

Программирование при помощи GTL, основанное на векторной геометрии, требует для каждой прямой линии назначения направления движения. Условимся, что *направление движения прямой определяется углом, который она образует с положительной осью X*. Угол будет иметь положительный знак, если ось X необходимо повернуть против часовой стрелки до совпадения с направлением линии, и отрицательный — в обратном направлении.

Направление должно быть придано также и окружностям. Условно принимается *за положительное направление движение против часовой стрелки и за отрицательное – по часовой стрелке*. По договоренности придается положительное значение радиусам окружностей с направлением движения против часовой стрелки и отрицательное – в обратном случае. Направление, данное элементу, обычно соответствует направлению движения инструмента по профилю.

2.2. Программирование информации о геометрических элементах

Описание в программе геометрических элементов предусматривает использование следующих строчных символов: ***a*** – угол; ***l*** – линия; ***c*** – окружность; ***d*** – расстояние; ***m*** – модуль; ***o*** – начало отсчета; ***r*** – радиус; ***p*** – точка; ***s*** – номер пересечения; ***b*** – скос.

Необходимость использования для этой информации строчных символов вызвана тем, что эти же заглавные символы используются в языке УЧПУ для другой информации. *Запись геометрических элементов в программе должна быть осуществлена*

до определения профиля и обычно записывается в начале программы.

Формат записи геометрического элемента:

НАЗВАНИЕ ИНДЕКС = <выражение>;

НАЗВАНИЕ – одно из четырех символических названий, предусмотренных для геометрических элементов ***o, p, l, c***;

ИНДЕКС – определяет номер переменной геометрического элемента; этот номер заключен между 0 и 255;

выражение – содержит всю информацию, необходимую для описания геометрического элемента; элементы могут быть определены:

- явным образом, программируя в кадре всю информацию, необходимую для определения геометрического элемента;
- неявным образом: описание с использованием других геометрических элементов (табл. 4).

Таблица 4

Примеры описания элементов

В явном виде:	В неявном виде:
$o1 = X30 Y30 a45$	$l1 = p1, p2$
$p2 = X60 Y30$	$c1 = l1, l2, r15$
$l2 = X30 Y50, a45$	$p1 = o1 X15 Y15$
$l3 = X0 Y0, X100 Y60$	$p3 = l3, c1$
	$c2 = p3, r8$

Формат геометрических определений предусматривает использование разделителей – символа «,» (запятая) для отделения геометрического элемента (прямая – точка – окружность) от последующего геометрического элемента или информации (такой, как радиус «r» или угол «a»).

Таблица 5

Примеры использования разделителя

Разделитель не требуется	Разделитель требуется
$p1 = X30 Y30$	$l2 = I30 J20 r10, X80 Y80$
$c1 = I10 J20 r30$	$l3 = X100 Y100, a45$
$l1 = X20 Y20, X100 Y-10$	$c3 = l1, l2, r18$

2.3. Определение точек начала отсчета

Функция определения точки начала отсчета системы координат (СК) позволяет определить их в прямом формате (явным образом). Обычно информация, находящаяся в программе, от-

носится к системе осей, совпадающих с осями станка. Однако при проектировании деталь может быть выполнена на чертеже с использованием различных декартовых систем: абсолютной системы и других систем (начальных точек) отсчета, которые могут быть приведены к абсолютной системе вращением и смещением осей. Геометрия GTL может быть определена при любой СК.

Формат: $on = X Y a$, где

n - номер точки СК. $X Y$ - координаты новой СК. a - угол вращения (положительный против часовой стрелки).

2.4. Определение точек

Функции определения точек (табл. 6) позволяют определить точки в прямой (явной) форме или в косвенной (неявной) форме. Определение может быть дано как в декартовых координатах, так и в полярных. В полярной системе координат любая точка плоскости может быть определена при помощи длины отрезка m (названной модулем), который соединяет ее с полюсом, и при помощи угла, который образуется отрезком прямой и полярной осью.

Таблица 6

Функции определения точек

Определение	Описание
$pn=(oi) X Y$	Точка n в декартовых координатах (с началом в oi^*) имеет известные координаты $X Y$.
$pn=(oi) m a$	Точка n в полярных координатах (с началом в oi) имеет модуль m под углом a к полярной оси.
$pn=li + lj$	Точка n пересечения двух прямых линий (i и j)**;
$pn=li + cj (, s2)$	Точка n пересечения прямой i и окружности j ;
$pn=ci + lj (, s2)$	Точка n пересечения окружности i и прямой j ;
$pn=ci + cj (, s2)$	Точка n пересечения двух окружностей i и j .

Примечания: * выражения в скобках могут быть опущены

** знак «+» может быть опущен или заменен на «-»

В случае пересечения прямая–окружность или наоборот, существуют два возможных решения: окружность c и прямая l пересекаются в точках $p1$ и $p2$. Проходя прямую l , следуя ее направлению, сначала встречаем точку $p1$ (первое пересечение), а затем – точку $p2$ (второе пересечение). Для выбора второго пересечения $p2$ следует использовать индикатор $s2$. Если он опущен, то выбирается первое пересечение $p1$.

В случае пересечения окружность–окружность существуют два возможных решения: окружности $c1$ и $c2$ пересекаются в точках $p1$ и $p2$. Рассматривается сориентированная прямая, соединяющая центр 1-ой окружности с центром второй окружности.

Она делит плоскость на две полуплоскости. Для выбора точки в правой полуплоскости (если смотреть в направлении ориентированной прямой – $p2$) следует использовать индикатор $s2$. Если он опущен, то автоматически выбирается точка в левой полуплоскости $p1$.

2.5. Определение прямой линии

Функции определения прямых линий (табл. 7) позволяют определить прямую линию в прямой (явной) или косвенной (неявной) форме. Направление прямой линии всегда от первого ко второму среди определяемых элементов. В случае, если прямая касается окружности, возможны два решения, так как прямая, проходящая через точку, может быть касательной к окружности в двух различных точках. Для выбора требуемого решения следует убедиться в том, что в точке касания окружность и прямая имеют одно и то же направление.

Таблица 7

Функции определения прямых линий

Определение	Описание
$ln=(om) I J r, (op) I J r$	Прямая n , касательная к двум окружностям в декартовых СК* m и p с известными координатами центра I, J и радиусами r .
$ln=(om)X Y, (op)X Y$	Прямая n , проходит через две точки, заданные в декартовых СК m и p известными координатами X и Y .
$ln=(om) I J r, (op)X Y$	Прямая n , касательная к окружности, заданной в СК m известными координатами центра I, J и радиусом r , проходит через точку с известными координатами X, Y в СК p .
$ln=(om)X Y, (op) I J r$	Прямая n , касательная к окружности, заданной в СК p известными координатами центра I, J и радиусом r , проходит через точку с известными координатами X, Y в СК m .
$ln=(om)I J r, a$	Прямая n , касательная к окружности, заданной в СК m известными координатами центра I, J и радиусом r , образует угол a с абсциссой.
$ln=(om)X Y, a$	Прямая n , проходящая через точку с известными координатами X, Y в СК m , образует угол a с абсциссой.
$ln=+ci, +cj^{**}$	Прямая n , касательная к двум известным окружностям i и j .
$ln=+ci, pj$	Прямая n , касательная к окружности i , проходит через точку j .
$ln=pi, +cj$	Прямая n , касательная к окружности j , проходит через точку i .
$ln=pi, pj$	Прямая n , проходит через две известные точки i и j .
$ln=+ci, a$	Прямая n , касательная к окружности i , образует угол a с абсциссой.
$ln=pi, a$	Прямая n , проходит через известную точку i и образует угол a с абсциссой.
$ln=+lm, d$	Прямая n , проходит параллельно известной прямой на расстоянии d

Примечания: * выражения в скобках могут быть опущены

** знак «+» может быть опущен или заменен «-»

2.6. Определение окружности

Язык GTL позволяет определить окружности в прямой форме (явной) или в косвенной (неявной) форме. Определяя окружности в косвенной форме, программист должен учитывать совместимость направлений элементов (знак «-» может изменить направление предопределенных элементов).

Таблица 8

Функции определения окружностей

Определение	Описание
$cn=(oi) I J r$	Окружность n в СК i известного радиуса r с координатами центра I, J .*
$cn=(oi) m a r$	Окружность n известного радиуса с центром, определяемым модулем m и углом a наклона к оси абсцисс.
$cn=+li, +lj, r$	Окружность n радиуса r , касательная к двум известным прямым i и j **.
$cn=+li, +cj, r$	Окружность n известного радиуса r , касательная к прямой i и к окружности j .
$cn=+ci, +lj, r$	Окружность n известного радиуса r , касательная к прямой j и к окружности i .
$cn=pi, +lj, r$	Окружность n известного радиуса r , проходящая через известную точку i и касательная к известной прямой j .
$cn=+li, pj, r$	Окружность n известного радиуса r , касательная к заданной прямой i и проходящая через известную точку j .
$cn=+ci, +cj, r$	Окружность n известного радиуса r , касательная к двум известным окружностям i и j .
$cn=pi, +cj, r$	Окружность n известного радиуса r проходит через точку i касательно к окружности j .
$cn=+ci, pj, r$	Окружность n известного радиуса r касается окружности i и проходит через точку j .
$cn=pi, pj, r$	Окружность n известного радиуса r , проходящая через две известные точки i и j .
$cn=pi, +lj$	Окружность n с центром в точке i и касательная к прямой j ;
$cn=pi, pj, pk$	Окружность n , проходящая через три точки i, j, k .
$cn=pm, r$	Окружность n известного радиуса r с центром в точке i .
$cn=+cm, +d$	Окружность n концентрично расположена к известной окружности i на расстояние d .
$cn=pi, +cj (, s2)$	Окружность n с центром в известной точке i касается известной окружности j .

* выражения в скобках могут быть опущены

** знак «+» может быть опущен или заменен «-»

2.7. Определение профиля

Под профилем подразумевается ряд геометрических элементов, записанных в последовательности обхода инструментом и описывающих геометрию обрабатываемой поверхности детали. Профиль, запрограммированный в геометрии GTL, определяется через функции G21 и G20: – G21 устанавливает начало профиля; G20 устанавливает конец профиля.

Профиль может быть открытым и закрытым. Открытый профиль всегда начинается с точки и заканчивается другой точкой. Компенсация радиуса инструмента действует перпендикулярно к первому элементу на точке начала профиля и перпендикулярно к последнему элементу на точке конца профиля. Компенсация радиуса должна быть открыта на первой точке профиля программированием в кадре функций G21 G41/G42 и закрыта на последней точке с функциями G20 G40.

Закрытый профиль начинается и заканчивается одним и тем же геометрическим элементом. Если профиль закрытый, сначала следует запрограммировать последний элемент, а затем после последнего элемента вызвать первый элемент профиля. Первая точка скорректированного профиля – пересечение первого и последнего элементов (первая точка равна последней точке). Компенсация радиуса должна быть открыта в начале профиля в кадре вызова последнего элемента, программируя функции G21 G41/G42, и закрыта в конце профиля в кадре вызова первого элемента с функциями G20 G40.

Когда первый и/или второй элементы являются окружностями, возможны два пересечения. Если не представляется никакой дополнительной информации, система выбирает первое. В случае если необходимо второе пересечение, следует запрограммировать дискриминатор s2. Дискриминатор s2 программируется в кадре вызова последнего элемента в начале профиля и на последнем элементе в конце профиля.

Компенсация радиуса должна быть открыта в начале профиля в кадре вызова последнего элемента и закрыта в конце профиля в кадре вызова первого элемента. Компенсация радиуса отменяется в первом кадре движения осей в плоскости профиля, следующего за функцией G40.

2.8. Пересечение между элементами

В случае пересечения двух прямых возможно только одно решение. В случае пересечения прямая – окружность или окружность – окружность всегда возможны два решения. Система автоматически выбирает первое. Если необходимо получить второе, следует запрограммировать дискриминатор s2 после определения первого элемента.

Правила выбора пересечений те же, что и при косвенном описании точки.

Геометрические элементы профиля могут быть связаны между собой за счет тангенциального сопряжения, пересечения

или присутствия автоматического соединения или фаски.

Соединения между элементами при помощи автоматического радиуса

Если элементы пересекаются, то можно определить соединение между ними (прямые линии или окружности), программируя значение радиуса: принимается положительное в направлении против часовой стрелки, отрицательное – по часовой стрелке.

Соединение r не может быть запрограммировано в кадре, следующем сразу же за кадром с G21, или же в кадре, предшествующем кадру с G20 (т. е. профиль не может начинаться и заканчиваться с соединения). В случае активизации компенсации радиуса инструмент размещается на пересечении двух геометрических элементов, смещенных радиусом инструмента. Если необходимо ввести радиус между двумя элементами, следует запрограммировать нулевой радиус $r0$.

Скосы

Скосы между прямолинейными элементами можно определить, программируя значение скоса без знака, рассматриваемое как расстояние от точки пересечения.

Скос не может быть запрограммирован в кадре, который непосредственно следует за кадром G21 или предшествует G20 (т. е. профиль не может начинаться или заканчиваться скосом). В геометрическом программировании GTL перемещения всегда осуществляются с рабочей подачей, для программирования быстрого перемещения необходимо запрограммировать скорость рабочей подачи F со значением быстрого хода. Если плоскость интерполяции не является той, которая образована осями X и Y , следует вначале определить плоскость, а затем определять элементы профиля.

2.9. Пример программирования обработки профиля

Рассмотрим пример применения технологии GTL для программирования обработки наружного замкнутого контура детали типа «Кронштейн» (рис. 12).

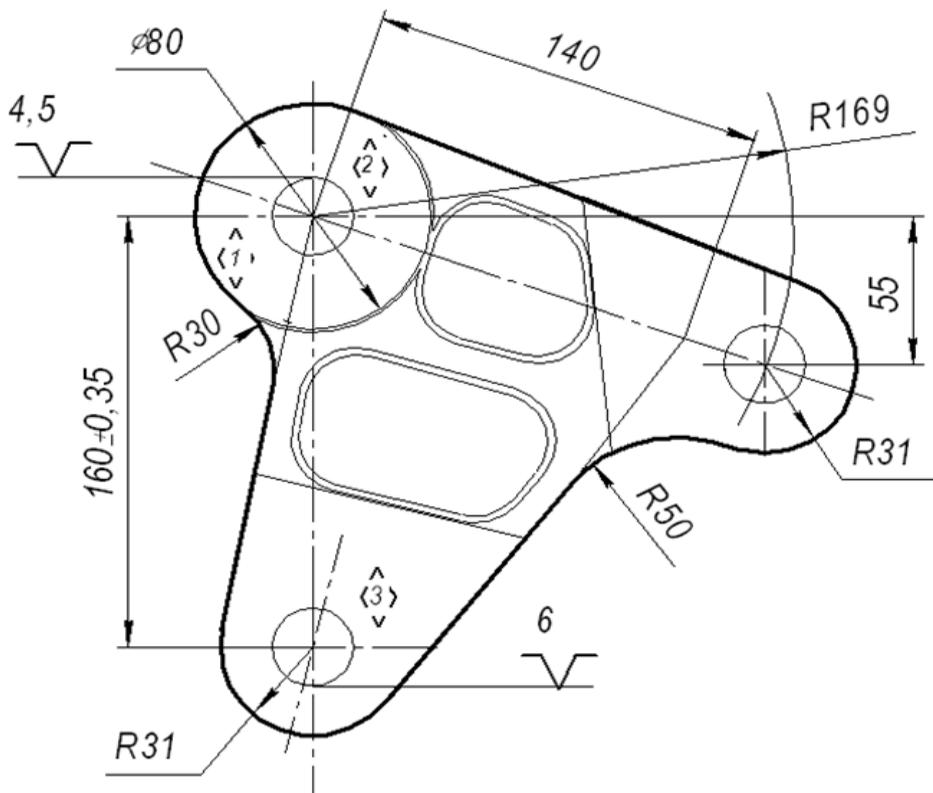


Рис. 12. Эскиз обработки детали типа «Кронштейн»

Для составления схемы расположения геометрических элементов профиля (см. рис. 13, табл. 9) необходимо пронумеровать опорные точки профиля Р, основные и вспомогательные линии L и окружности С.

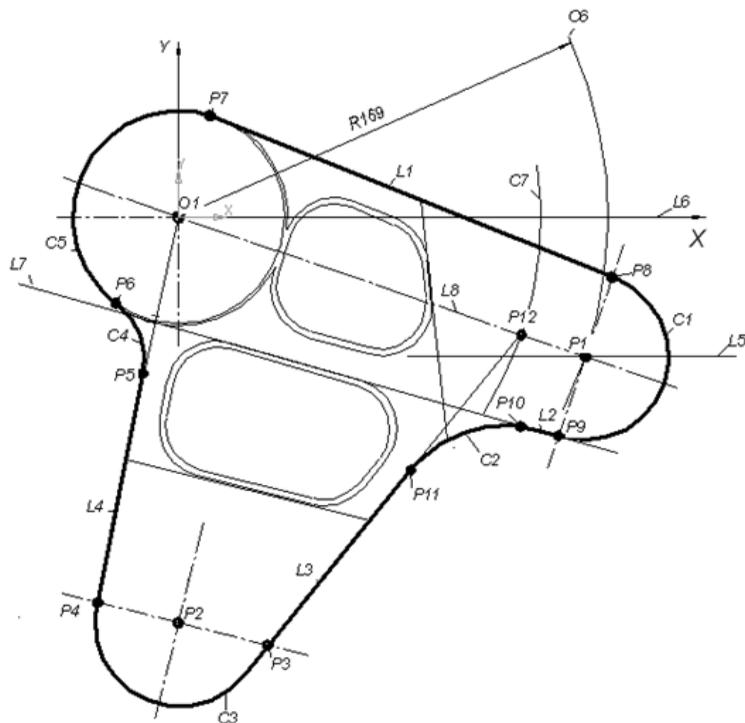


Рис. 13. Схема расположения геометрических элементов

Таблица 9

Геометрическое описание профиля

Определение	Описание
1	2
$o1=X0 Y0, a0$	Определяем СК №1
$c6=X0 Y0, r169$	Окружность С6 известного радиуса с центром в точке 0,0 (направление против часовой стрелки)
$l6=X0 Y0, a0$	Прямая $l6$ проходит через точку 0,0 под углом 0° к абсциссе (направление от начала СК вправо)
$l5=l6, d-55$	Прямая $l5$ проходит параллельно $l6$ на расстоянии 55 мм от начала координат (направление $l5$ от начала СК вправо)
$p1=l5, c6(s2)$	Точка $p1$ лежит на пересечении прямой $l5$ и окружности $c6$ (второе решение)
$c1=p1, r31$	Окружность $c1$ имеет радиус $r31$ и центр в точке $p1$

Продолжение табл. 9

1	2
$p2=X0 Y-160$	Точка $p2$ имеет известные координаты относительно СК1
$c3=p2,r31$	Окружность $c3$ имеет радиус $r31$ и центр в точке $p2$
$c5=X0 Y0, r80$	Окружность $c5$ известного радиуса с центром в точке $0,0$ (направление против часовой стрелки)
$l1=+c1,+c5$	Прямая $l1$, касательная к двум известным окружностям $c1$ и $c5$. (точки касания определены при движении против часовой стрелки по двум окружностям)
$l7=-c1, -c5$	Прямая $l7$, касательная к двум известным окружностям $c1$ и $c5$. (точки касания определены при движении по часовой стрелке по двум окружностям)
$p15=X0Y0$	Точка $p15$ лежит в начале координат СК1
$l8=p15,p1$	Прямая $l8$ проходит через известные точки $p15$ и $p1$ (направление от начала СК вправо)
$c7=X0Y0,r140$	Окружность $c7$ с центром в точке $p15$ имеет радиус $r140$
$p12=c7,l8$	Точка $p12$ лежит на пересечении окружности $c7$ и прямой $l8$ (решение точки пересечения при движении по прямой вправо и по окружности против часовой стрелки)
$l3=-c3, p12$	Прямая $l3$, касательная к окружности $c3$, проходит через точку $p12$ (решение точки касания выбирается при обходе окружности по часовой стрелке от оси абсцисс)
$l4=c3, p15$	Прямая $l4$, касательная к окружности $c3$, проходит через точку $p15$ (решение точки касания выбирается при обходе окружности против часовой стрелки от оси абсцисс)

Пример управляющей программы

$N1$ (DIS,"Пример")

$N2$ $o1=X0 Y0, a0$

$N3$ $c6=X0 Y0, r169$

$N4$ $l6=X0 Y0,a0$

$N5$ $l5=l6, d-55$

$N6$ $p1=l5,c6(s2)$

$N7$ $c1=p1,r31$

$N8$ $p2=X0 Y-160$

$N9$ $c3=p2,r31$

$N10$ $c5=X0 Y0, r80$

$N11$ $l1=+c1,+c5$

$N12$ $l7=-c1, -c5$

$N13$ $p15=X0 Y0$

$N14$ $l8=p15,p1$

$N15$ $c7=X0 Y0, r140$

*N16 p12=c7,l8
N17 l3=-c3, p12
N18 l4=c3, p15
N19 T1.1 S800 F250 M6 M3 M7
N20 G00 X0 Y0
N21 Z-10
N21 G21 G42 c5
N22 l1
N23 c1
N24 l7
N25 r50
N26 l3
N27 c3
N28 l4
N29 r30
N30 c5
N31 G20 G40 l1
N32 G0 Z2
N33 X0 Y0 M30*

ЗАДАНИЕ НА ВЫПОЛНЕНИЕ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

Для заданного варианта профиля (см. приложение 2) выполнить проектирование геометрическое описание его элементов и спроектировать управляющую программу по технологии GTL.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Приведите пример эффективного применения технологии GTL при программировании обработки на станках с ЧПУ.
2. Способы описания геометрических элементов типа «Начало координат».
3. Способы описания геометрических элементов типа «Точка».
4. Способы описания геометрических элементов типа «Прямая».
5. Способы описания геометрических элементов типа «Окружность».
6. Особенности программирования закрытого профиля.
7. Особенности программирования открытого профиля.
8. Программирование скоса.

9. Программирование автоматического радиуса.
10. Если решение профиля не единственно, то какими правилами необходимо руководствоваться. Приведите пример.

ЗАКЛЮЧЕНИЕ

В пособии приводятся сведения, необходимые для уверенного применения в профессиональной деятельности инженера-технолога на любом машиностроительном предприятии. Программирование современного оборудования с компьютерным управлением представляет собой достаточно сложную задачу для специалиста в этой области, требует от него достаточных знаний и умений, формируемых комплексом специальных дисциплин направления «Конструкторско-технологическое обеспечение машиностроительных производств». Надеемся, что данное пособие поможет бакалаврам и магистрантам получить необходимые им теоретические знания и практические навыки в сфере управления современным технологическим оборудованием.

СПИСОК ЛИТЕРАТУРЫ

1. Базовые принципы программирования// HAAS Automation, INC.2800 Sturgis Rd. Oxnard, CA 93030.
2. Фельдштейн Е.Э., Корниенко М.А. Обработка деталей на станках с ЧПУ: учеб. пособие. М.: Новое знание, 2008 – 299 с.
3. Лебедев А.Р. Технология программирования высокого уровня для СЧПУ металлорежущими станками, [электрон. ресурс] – режим доступа: <http://de.donstu.ru/CDOCourses/7ec7b2e5-8711-4978-973b-2da80002f63b/1077/808/index.html>.
4. Гузев В.И., Режимы резания для токарных и сверлильно-фрезерно-расточных станков с числовым программным управлением: справ. /Батуев В.А., Сурков И.В./ Под ред. В.И. Гузеева - М.: Машиностроение, 2005 – 368 с.
5. Диалоговые САПР технологических процессов: учеб. для вузов Ю.М. Соломенцева – М.: Машиностроение, 2005 – 232 с.
6. Белов М.П. Технические средства автоматизации и управления: учеб. пособие. – СПб: СЗТУ 2006 – 184 с.
7. Ключев А.С. Проектирование систем автоматизации технологических процессов – СПб, 2008.
8. Леонтьев А.Г. Микропроцессорные электромеханические системы. учеб. пособие. – СПб: ГТУ, ФТК, 1998 – 109 с.
9. Системное программное обеспечение: уч. пособие /В.И. Аблязов и др. – СПб: ГТУ, 1996 – 66 с.

ПРИЛОЖЕНИЕ 1

Инструкция по применению компьютерного тренажера НААС

Действия при включении устройства

1. Включить питание на пульте.
2. Для сброса сообщений об ошибках нажмите «RESET».
3. Для приведения в рабочее состояние приводов станка необходимо перевести исполнительные органы в исходное состояние. Кнопка «POWER UP RESTASRT».

Установка нуля детали

Чтобы обеспечить точность станочной обработки детали, станок должен знать, где на столе находится деталь. Установив в шпиндель указательный инструмент, выполняйте перемещение с помощью рукоятки до достижения верхнего левого угла детали (см. рис. А1). Эта позиция является нулём детали. Координаты будут введены в параметр G54 на странице Work Offset (рабочая коррекция).

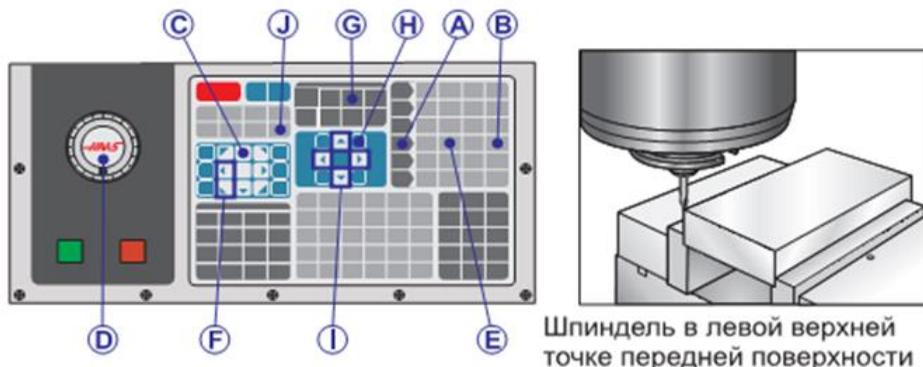


Рис. А1

Значения коррекции можно ввести и вручную. Для этого надо выбрать соответствующую страницу коррекции, переместить курсор в нужный столбец, набрать число и нажать WRITE или F1. Кнопка F1 записывает введенное число в выбранный столбец. Кнопка WRITE (записать) прибавляет введенное значение к числу, находящемуся в выбранном столбце.

Стандартная настройка нуля детали

1. Поместите заготовку в зажимное приспособление и закрепите.
2. Вставьте в шпиндель указательный инструмент.

3. Нажмите кнопку ручного управления (A)
4. Нажмите кнопку 1/100. (B) (Перемещение при вращении рукоятки будет быстрее).
5. Нажмите +Z (C).
6. Рукояткой D переместите ось Z приблизительно на 5 мм над деталью.
7. Нажмите кнопку 001/1. (E) (Перемещение при вращении рукоятки будет медленнее).
8. Рукояткой D переместите ось Z приблизительно на 5 мм над деталью.
9. Выберите ось X или Y (F) и с помощью рукоятки D подведите инструмент к верхнему левому углу детали.
10. Нажмите кнопку Offset (G)
11. Несколько раз нажимайте кнопку Page Up (H), пока не появится страница Work Zero Offset (коррекция нуля).
12. Установите курсор I в G54 столбец X.
13. Нажмите кнопку Part Zero Set (установка нуля детали, J), чтобы ввести значение в столбец оси X. При втором нажатии кнопки Part Zero Set (J) будет загружено значение в столбец оси Y.

***ВНИМАНИЕ!** Не нажимайте кнопку Part Zero Set третий раз, иначе будет введено значение для оси Z. При выполнении программы это приведёт к аварии или сообщению об ошибке.*

Привязка инструментов

Следующий этап выполняется путем прикосновения инструмента к поверхности детали. При этом определяется фактическое расстояние между режущей кромкой и вершиной детали. Другое название процедуры – коррекция на длину инструмента, она обозначается символом H в строке программы станка.

Расстояние для каждого инструмента заносится в таблицу коррекции на инструмент.

1. Установите инструмент требуемого номера в шпиндель.
2. Нажмите кнопку ручного управления A.
3. Нажмите кнопку 0.1/100. (B) (Перемещение при вращении рукоятки будет быстрее).
4. Выберите ось X или Y (C) и с помощью рукоятки D подведите инструмент к центру детали.
5. Нажмите +Z (E)
6. Рукояткой D переместите ось Z приблизительно на 5 мм над деталью.
7. Нажмите кнопку 0.0001/1. (F) (Перемещение при вращении рукоятки будет медленнее).

8. Поместите между инструментом и обрабатываемой деталью лист бумаги. Осторожно опустите инструмент как можно ближе к верхней поверхности детали так, чтобы бумага оставалась не зажатой.

9. Нажмите кнопку Offset (G)

10. Нажимайте кнопку Page Up (H) до появления страницы с заголовком "Coolant - Length - Radius" (СОЖ -длина - радиус) и пролистайте до инструмента требуемого номера.

11. Установите курсор (I) в положение Geometry (геометрия) для требуемого инструмента.

12. Нажмите кнопку Tool Offset Mesur (измерение коррекции на инструмент, J). При этом положение оси Z, показанное в левом нижнем углу экрана, будет помещено в позицию номера инструмента.

***ВНИМАНИЕ!** При выполнении следующего пункта шпиндель будет быстро перемещаться по оси Z.*

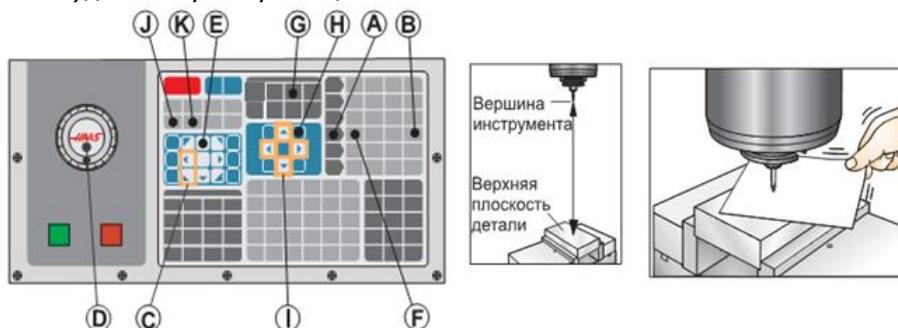


Рис. А2

13. Нажмите кнопку NEXT TOOL (следующий инструмент) (K).

Загрузка управляющей программы

Для создания новой программы нажмите кнопку LIST PROG (список программ), чтобы вызвать список программ. Введите номер программы (Oxxxxx) и нажмите кнопку SELECT PROG (выбор программы) или ENTER (ввод). Если программа с этим именем существует, она будет вызвана. Если такой программы не существует, она будет создана. Чтобы показать новую программу, нажмите кнопку EDIT (редактирование). Новая программа состоит только из названия и символа конца блока (;). Нумерованные программы сохраняются в памяти при выключении станка.

Отработка управляющей программы

Графический режим

Надёжный способ проверки и отладки программы – запустить её в графическом режиме. Станок не будет выполнять никаких перемещений, вместо этого все перемещения будут отображаться на экране.

Графический режим можно запускать из режимов MEMORY (память), MDI (ручной ввод данных) или DNC (прямое числовое управление). Для выполнения программы нажимайте клавишу SETNG/GRAPH до появления страницы GRAPHICS (графика). Для работы прямого числового управления в графическом режиме сначала выберите режим ПЧУ, затем перейдите к графическому отображению и запустите программу на блоке ЧПУ

В графическом режиме существуют три полезных функции, которые можно вызвать нажатием одной из функциональных клавиш (F1, F2, F3 и F4). Клавиша F1 (помощь) отображает краткое описание каждой функции, доступной в графическом режиме. Клавиша F2 (увеличение) увеличивает масштаб изображения на дисплее. Для этого используйте клавиши со стрелками, а затем нажмите кнопку WRITE. Клавиша F3 (положение) предоставляет доступ к различным страницам POSITIONS (положения) в графическом режиме. Клавиша F4 (программа) отображает текущую программу.

В графическом режиме эмулируются не все функции и перемещения станка.

Пробный прогон

Функция DRY RUN (пробный прогон) применяется для быстрой проверки программы без фактической обработки детали. Для запуска пробного прогона в режиме MEM (память) или MDI (ручной ввод данных) нажмите кнопку DRY RUN (пробный прогон). В процессе пробного прогона скорость всех быстрых перемещений и подач выбирается клавишами скоростей ручного управления.

Пробный прогон может быть включен или выключен только после полного завершения программы или нажатия кнопки RESET (сброс). В процессе пробного прогона происходят все необходимые смены инструмента. Скорость шпинделя в ходе пробного прогона можно регулировать клавишами коррекции.

Графический режим не менее удобен и более безопасен, поскольку до окончания проверки программы оси станка не перемещаются.

Выполнение программ

Для выполнения программы необходимо загрузить ее в станок. После загрузки программы и настройки коррекции программу можно запустить клавишей CYCLE START (начало цикла). Реко-

мендуется до начала обработки детали запустить программу в графическом режиме.

Фоновое редактирование

Функция фонового редактирования позволяет во время выполнения одной программы изменять другую.

Для запуска фонового редактирования в режиме MEM (память) нажмите клавишу PRGRM/CONVRS, введите имя редактируемой программы (Oпппп) и нажмите клавишу F4. Если имя программы не указано, на дисплее отображается страница PROGRAM REVIEW (просмотр программы), позволяющая редактировать выполняемую в данный момент программу. Для выхода из режима фонового редактирования выберите любой другой экран или нажмите F4. В процессе выполнения программы можно просмотреть список программ, загруженных в блок ЧПУ, нажав клавишу LIST PROG (список программ).

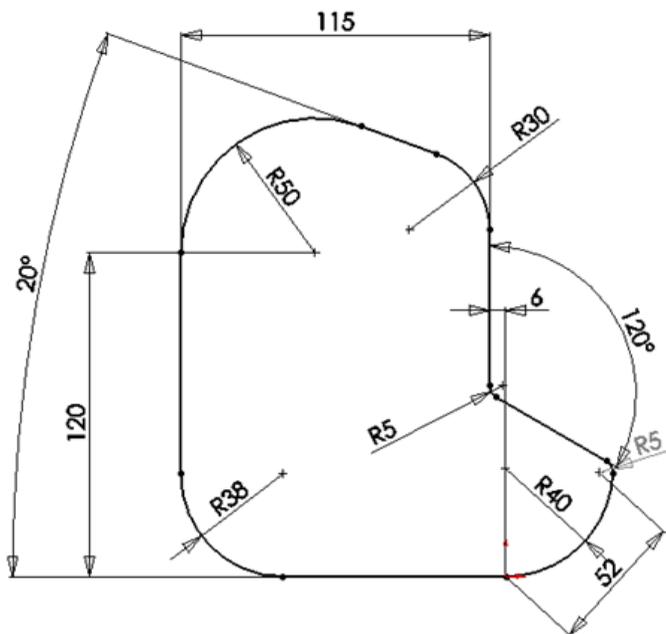
Изменения, внесенные при фоновом редактировании, не оказывают никакого влияния на выполняемую программу и ее подпрограммы. Внесенные изменения вступают в силу при следующем запуске программы.

Кнопка CYCLE START (начало цикла) не может использоваться в процессе фонового редактирования. Если программа содержит программируемый останов (M00), выйдите из режима фонового редактирования (нажмите клавишу F4) и нажмите клавишу CYCLE START (начало цикла) для продолжения выполнения программы.

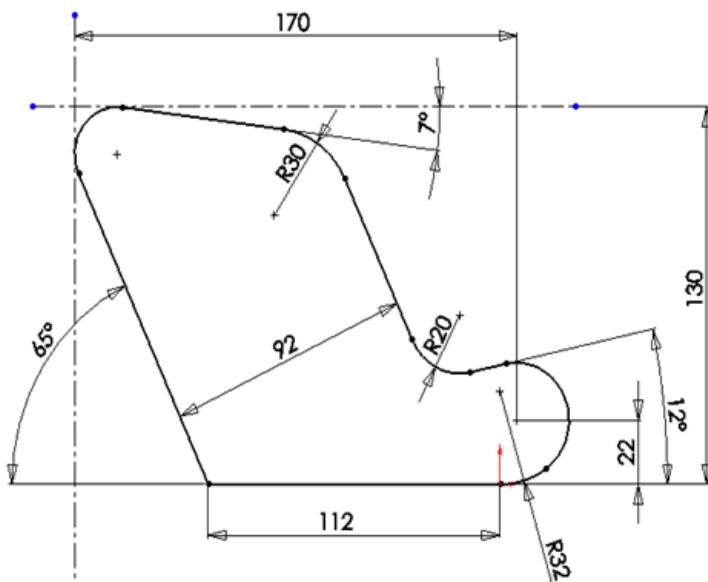
ПРИЛОЖЕНИЕ 2

Варианты заданий для самостоятельной работы

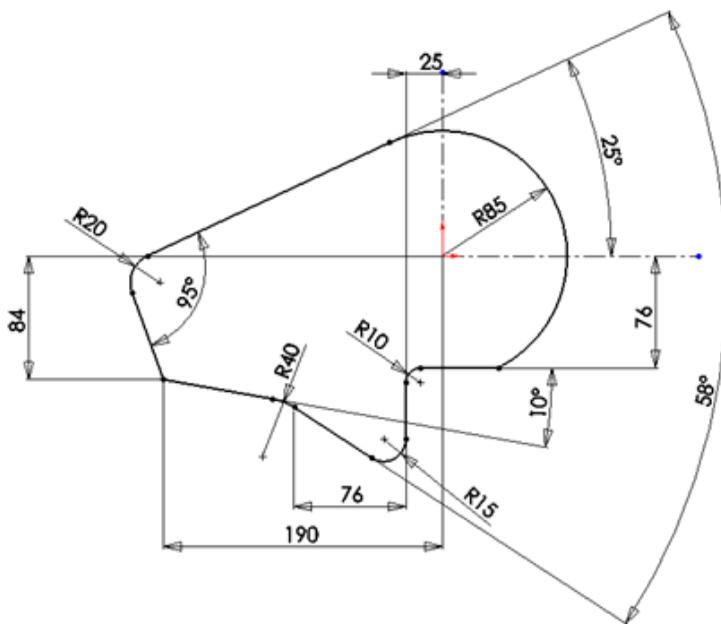
Вариант 1



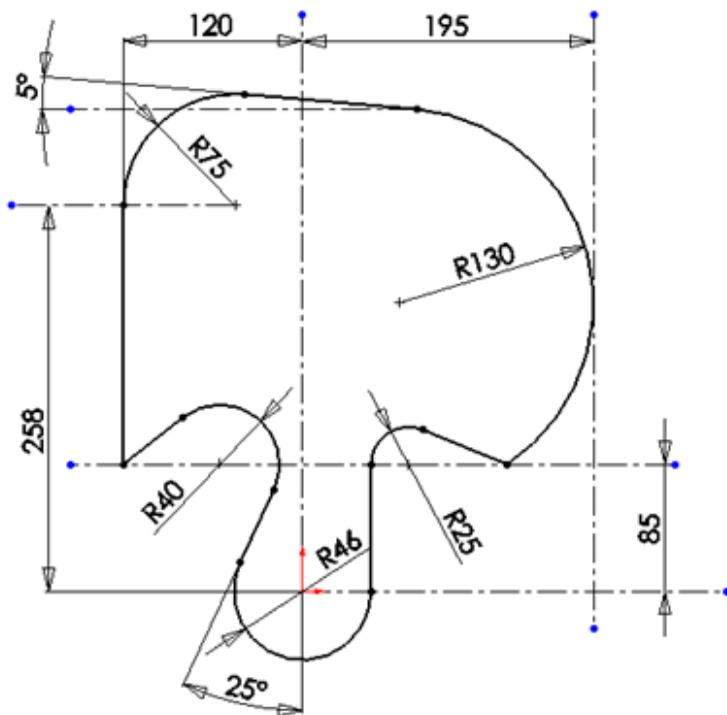
Вариант 2

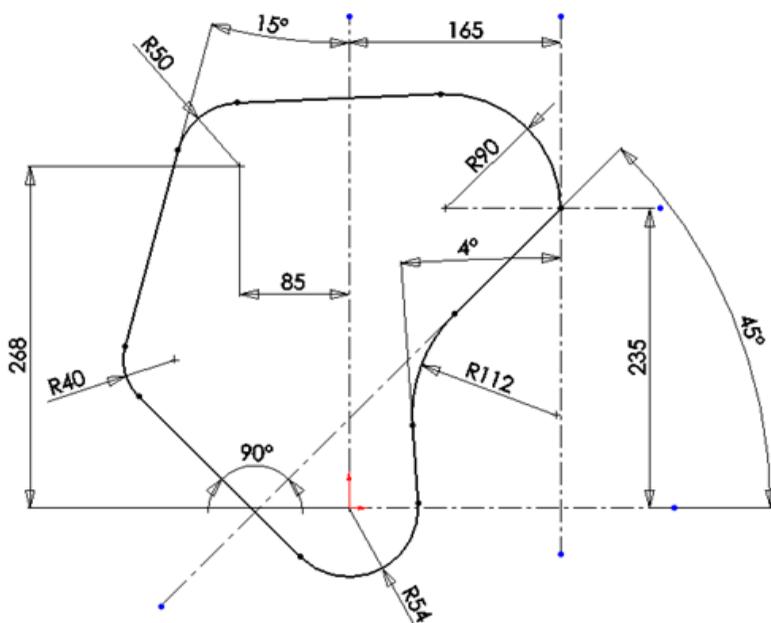


Вариант 3

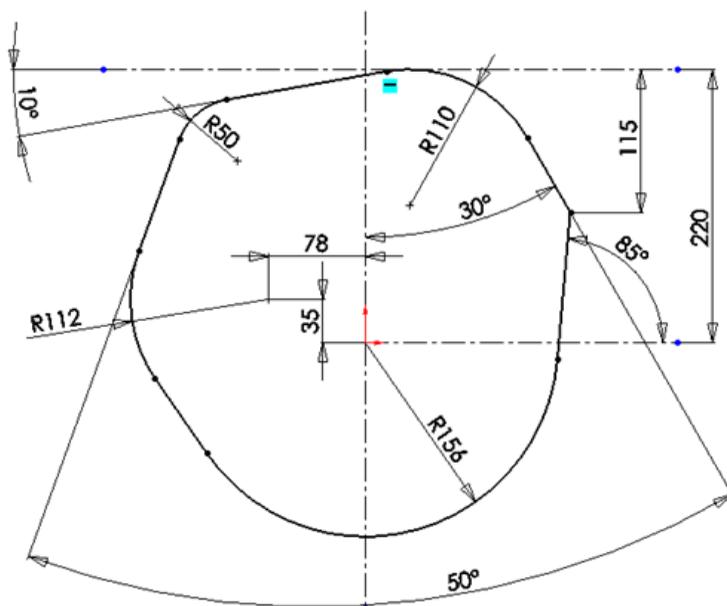


Вариант 6

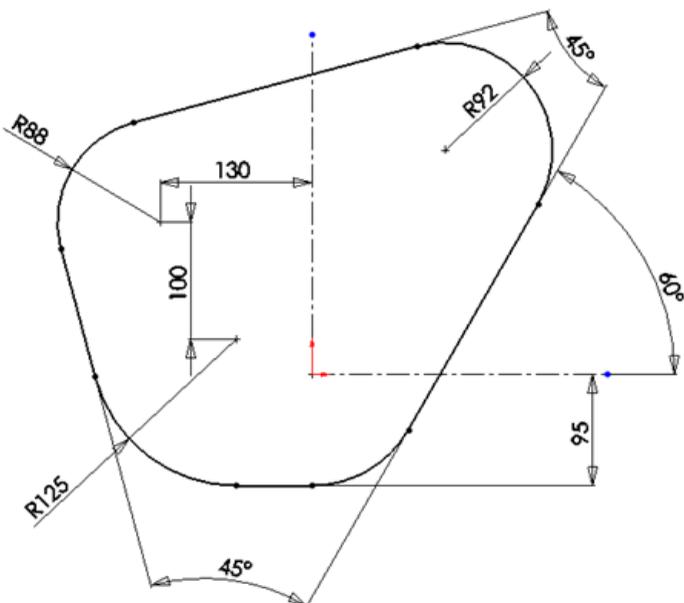
**Вариант 7**



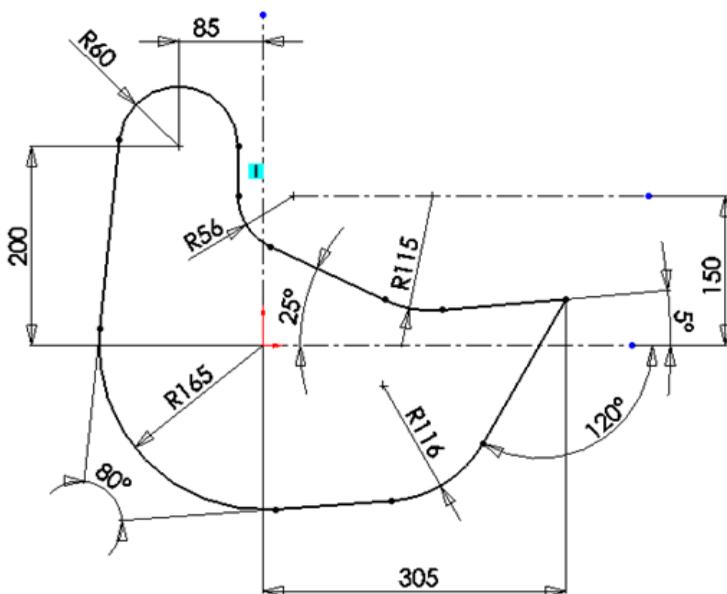
Вариант 8



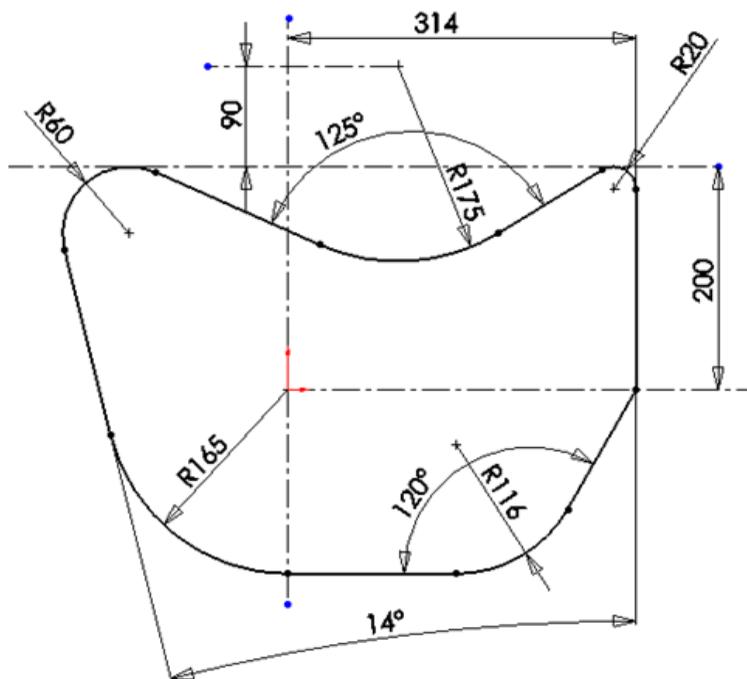
Вариант 9



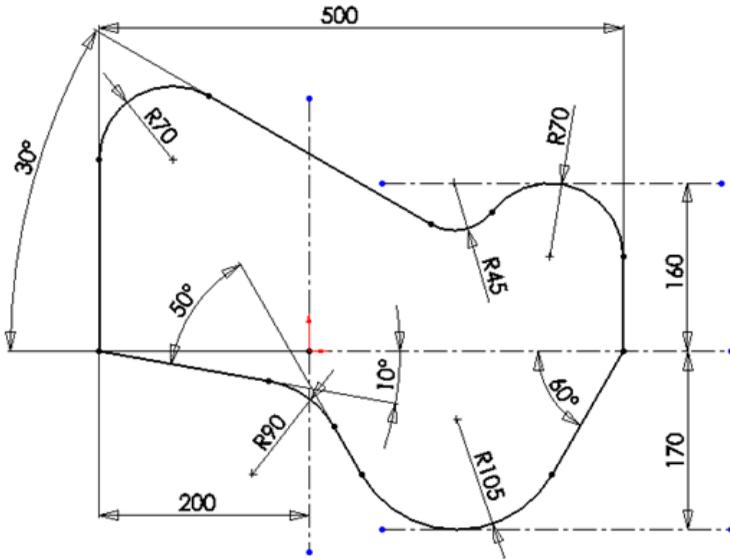
Вариант 10



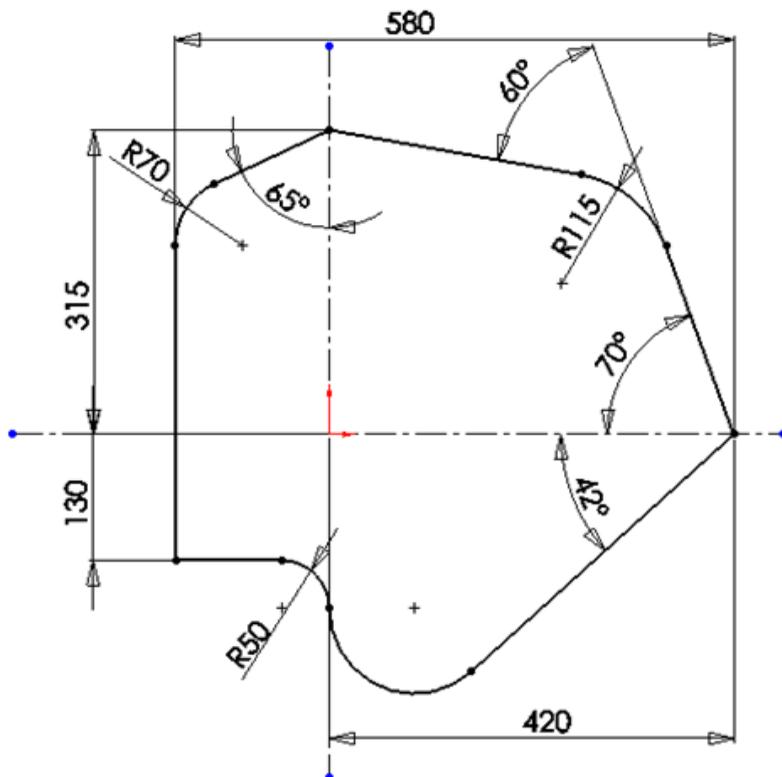
Вариант 11



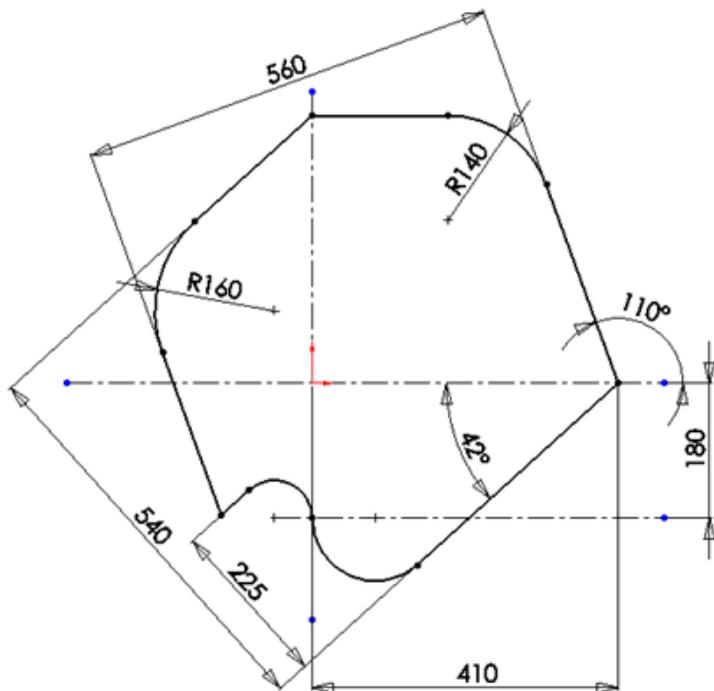
Вариант 12

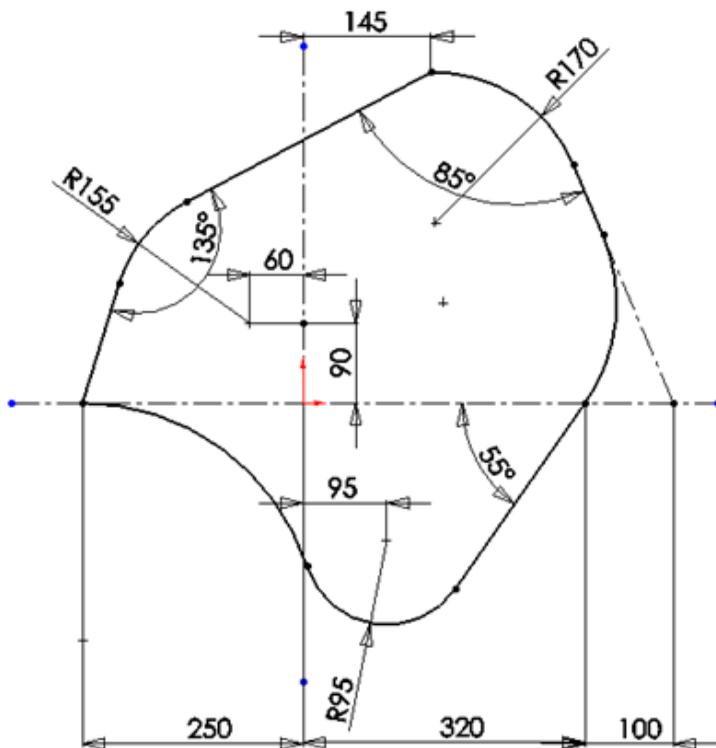


Вариант 13

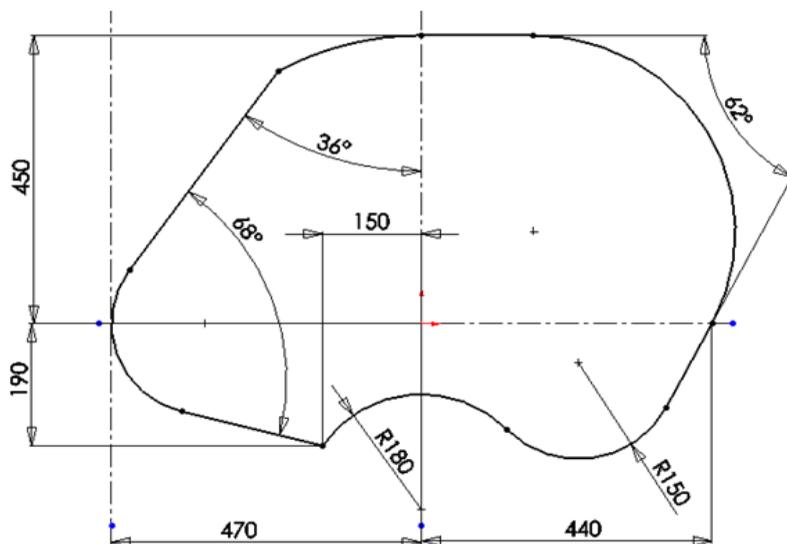


Вариант 14

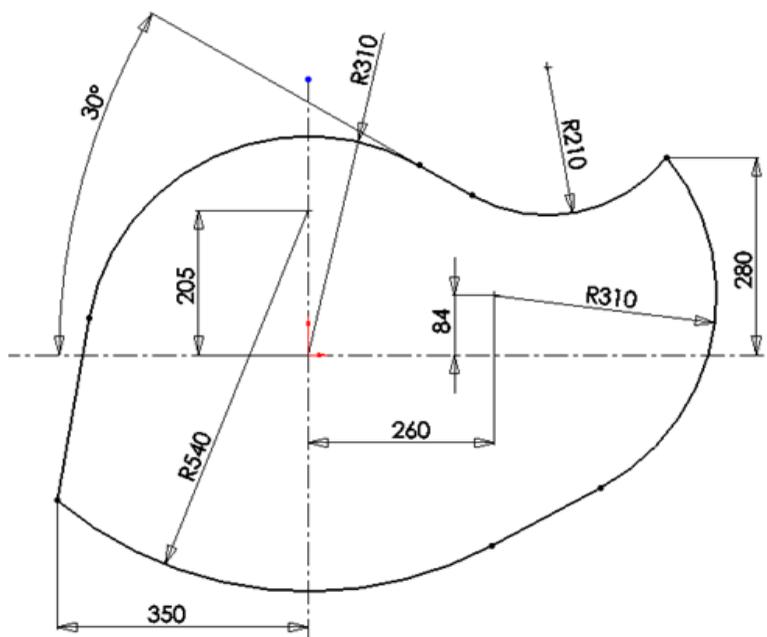
**Вариант 15**



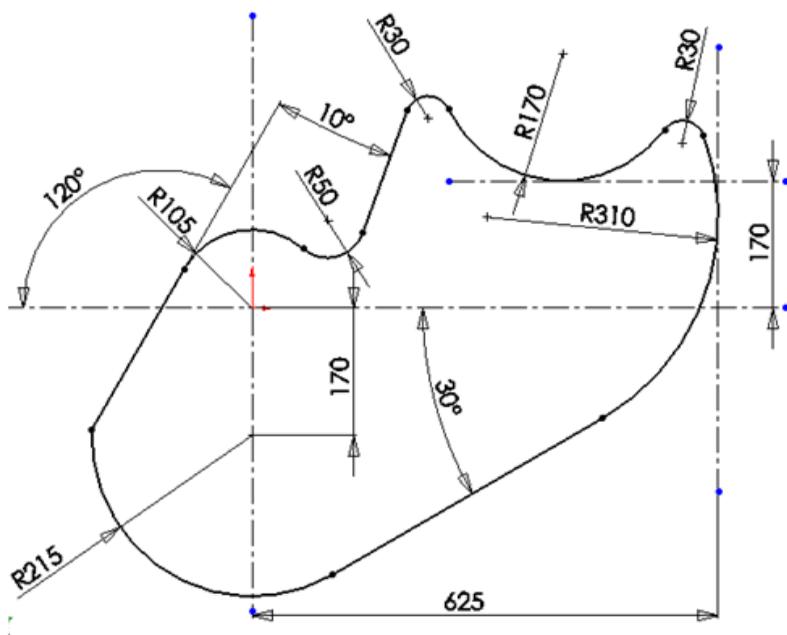
Вариант 16



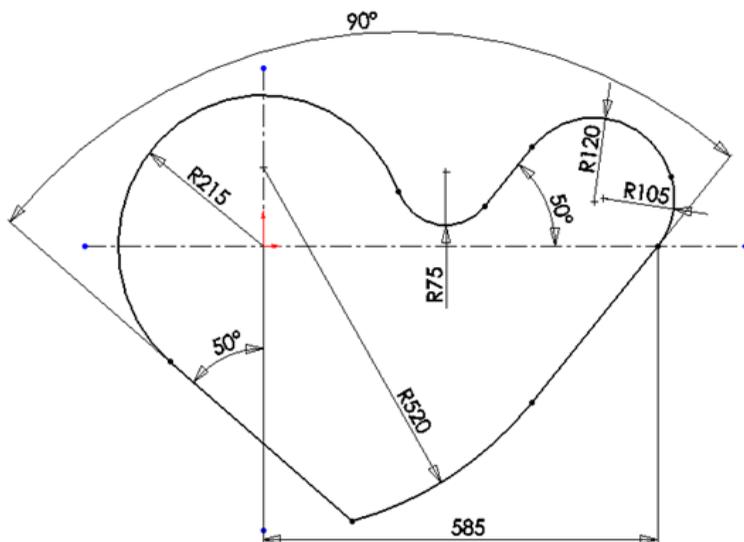
Вариант 17



Вариант 18



Вариант 19



Вариант 20

