

НСТ[®] 3ххТ

Устройство ЧПУ токарным станком

Описание программирования

От версии программного обеспечения № 16.0

Производитель и разработчик: **NCT Ipari Elektronikai kft.**

H1148 Budapest Fogarasi út 7

✉ Почтовый адрес: 1631 Bp. pf.26

☎ Телефон: (+36 1) 467 63 00

☎ Телефакс: (+36 1) 467 63 09

Электронная почта: nct@nct.hu

Вебстраница: www.nct.hu

ВНИМАНИЕ!

 **Замечание:** *Перед использованием нового программного обеспечения просьба внимательно прочесть изменения!*

Новинки и расхождения программного обеспечения варианта № 16 по сравнению варианта № 15.

10.2.3 Компенсация углового положения заготовки. Компенсация углового положения заготовки: на странице [88](#).

Основной поворот можно задавать уже одновременно вдоль всех трёх вращающихся осей, он не ограничен лишь на одну ось.

18.2.1 Цикл чёрного точения (G71): на странице [199](#).

В цикл чёрного точения было введено и чёрное точения 3-го типа (См на странице ?.).

18.2.2 Цикл торцевого чёрного точения (G72): на странице [210](#).

В цикл торцевого чёрного точения было введено торцевое чёрное точение 3-го типа (см на странице ?.).

20.7 Обращение осями, не перпендикулярными одной к другой: на странице [277](#).

Если на станке перемещение одной оси не перпендикулярное к другой, а по сравнению к перпендикуляру образует угол, эту функцию следует использовать.

15.1 Накопитель. Ссылка на коррекции инструмента (T, или D): на странице [124](#).

В управлении прекращен самостоятельный токарный и фрезерный накопитель. В будущем существует один общий накопитель. Если один из каналов настроен для точения, а другой - для фрезерования, их накопитель общим будет.

23.1.16 Значение актуальной коррекции по длине: на странице [318](#).

23.1.18 Значения накопителя инструмента: на странице [320](#).

Содержание

1 Введение.	11
1.1 Программа деталей..	11
1.2 Канал.	12
1.3 Основные понятия.	13
2 Управляемые оси.	17
2.1 Название и номер осей..	17
2.2 Расширенные названия осей.	18
2.3 Назначение осей к каналам.	18
2.4 Система мер осей, и точность отображения позиции.	19
3 Подготовительные функции (коды G).	21
4 Интерполяция.	26
4.1 Позиционирование (G0).	26
4.1.1 Позиционирование линейной интерполяцией.	26
4.1.2 Позиционирование перекрыванием движений быстрого хода.	27
4.2 Линейная интерполяция (G1).	28
4.3 Круговая интерполяция (G2, G3).	30
4.3.1 Винтовая интерполяция по плоскости (G2, G3).	35
4.3.2 Цилиндрическая винтовая интерполяция (G2, G3).	37
4.3.3 Коническая винтовая интерполяция (G2, G3).	40
4.4 Нарезание резьбы с равномерным шагом (G33).	42
4.5 Интерполяция полярных координат (G12.1, G13.1).	44
4.6 Цилиндрическая интерполяция (G7.1).	49
5 Данные координат.	52
5.1 Абсолютное и инкрементное программирование (G90, G91), оператор I, адрес U, V, W.	52
5.2 Преобразование дюймы/метрические единицы (G20, G21).	53
5.3 Программирование по диаметру, или по радиусу.	54
5.3.1 Переключение программирования по радиусу/по диаметру (G10.9).	54
5.4 Задание данных полярными координатами (G15, G16).	56
5.5 Задание и точность данных координат.	58
5.6 Обращение переворотом вращающихся осей.	58
6 Подача.	64
6.1 Подача быстрым ходом.	64
6.2 Рабочая подача.	64
6.2.1 Поминутная подача (G94) и подача за оборот (G95).	65
6.3 Функции управления подачей.	65
6.3.1 Точный останов в конце кадра (G9).	66
6.3.2 Режим точного останова (G61).	66
6.3.3 Нерывное точение (G64).	66
6.3.4 Запрет override и стопа (G63).	66
6.3.5 Автоматическое уменьшение подачи при внутренних углах (G62).	66

6.4 Автоматическое уменьшение подачи при внутренних дуг.	67
7 Ускорение.	69
7.1 Автоматическое замедление при углов в состоянии G64.....	72
7.2 Ограничение ускорений, возникающих вдоль траектории в нормальном направлении.....	75
7.3 Ограничение скачка ускорения.....	77
8 Задержка (G4).....	79
9 Референтная точка.	80
9.1 Автоматической приём референтной точки (G28).	81
9.2 Наводка на референтную точку 2., 3., 4. (G30).	82
10 Системы координат, выбор плоскости.....	84
10.1 Станочная система координат.....	84
10.1.1 Позиционирование в станочной системе координат (G53).....	85
10.2 Системы координат заготовки.....	85
10.2.1 Выбор системы координат заготовки (G54...G59).	86
10.2.2 Выбор расширенных систем координат заготовки (G54.1 P).	88
10.2.3 Компенсация углового положения заготовки.....	88
10.2.4 Установка смещения системы координат заготовки (G10 L2).	90
10.2.5 Установка смещения расширенных систем координат заготовки (G10 L20)	91
10.2.6 Создание новой системы координат заготовки (G92).	91
10.3 Локальная система координат (G52).....	92
10.4 Выбор плоскости (G17, G18, G19).	95
11 Шпиндельные функции.....	97
11.1 Команда чисел оборотов шпинделя (код S).....	97
11.1.1 Ссылка на несколько шпинделей. Расширение адреса S.....	97
11.1.2 Назначение шпинделей к каналам.	98
11.2 Функции M управления шпинделем.....	98
11.3 Обращение диапазонами чисел оборотов.	99
11.4 Главный шпиндель. Выбор главного шпинделя.	100
11.5 Расчёт постоянной скорости резания.	101
11.5.1 Задание расчёта постоянной скорости резания (G96 S, G97).	101
11.5.2 Ограничение чисел оборотов при расчёте постоянной скорости резания (G92 S).	102
11.5.3 Выделение оси для расчёта постоянной скорости резания (G96 P).	103
11.6 Наблюдение за колебанием чисел оборотов шпинделей.	103
11.7 Позиционирование шпинделей.....	103
11.7.1 Ориентировка шпинделей.....	104
11.7.2 Останов шпинделей и замыкание контура регулировки позиции.	104
11.7.3 Программирование позиционирования шпинделей.....	104
11.7.4 Синхронизация двух шпинделей верно позиции.	106
11.7.5 Выключение режима регулирования позиции шпинделей.	108
11.8 Преобразование шпинделя в ось, оси в шпиндель.	108

12 Функция Т.	110
12.1 Программирование смены инструментов.	110
13 Смешанные и вспомогательные функции.	112
13.1 Смешанные функции: (коды М).	112
13.2 Вспомогательные функции (А, В, С, U, V, или W).	114
13.3 Функции разгрузки буфера.	114
14 Организация программы деталей.	116
14.1 Номер кадра (адрес N).	116
14.2 Условный пропуск кадра (/ адрес).	116
14.3 Запись замечаний в программе деталей: (комментария).	117
14.4 Главная программа и подпрограмма.	117
14.4.1 Опознание программ в накопителе. Номер программы (О).	118
14.4.2 Вызов подпрограммы (M98).	118
14.4.3 Возвращение из подпрограммы (M99).	120
14.4.4 Переход внутри главной программы.	122
14.5 Функции М дождания между каналами.	122
15 Коррекция инструмента.	124
15.1 Накопитель коррекции. Ссылка на коррекцию инструмента (Т, или D).	124
15.2 2-й накопитель геометрической коррекции.	127
15.3 Изменение значений коррекций инструмента из программы (G10).	129
15.4 Учёт коррекций по длине инструмента по коду Т.	129
15.5 Учёт коррекций по длине инструмента по коду G (G43.7, G49).	133
15.6 Коррекция радиуса инструмента (G40, G41, G42).	134
15.6.1 Включение расчёта коррекции радиуса. Наводка на контур.	138
15.6.2 Включенное состояние расчёта коррекции радиуса. Движение по контуру.	145
15.6.3 Выключение расчёта коррекции радиуса. Отход от контура.	152
15.6.4 Перемена направления в расчёте коррекции радиуса.	157
15.6.5 Программирование сохранения вектора (G38).	160
15.6.6 Программирование угловой дуги (G39).	161
15.6.7 Проблемы помеха слежения за контуром. Проверка на интерференцию	163
16 Специальные трансформации.	171
16.1 Поворот конфигурации вокруг данной точки (G68, G69).	171
16.2 Масштабирование конфигурации относительно заданной точки (G50, G51)	173
16.3 Отражение конфигурации на одну, или несколько прямых (G50.1, G51.1).	176
16.4 Правила программирования специальных трансформаций.	178
17 Автоматические геометрические расчёты.	180
17.1 Программирование фасок и скруглений.	180
17.2 Задание прямой с углом направления.	181
17.3 Расчёты точек пересечения в плоскости.	183
17.3.1 Точка пересечения двух прямых.	183
17.3.2 Точка пересечения прямой и окружности.	185

17.3.3 Точка пересечения окружности и прямой.	187
17.3.4 Точка пересечения двух окружностей.	189
17.3.5 Сцепление расчётов точек пересечения.	191
18 Токарные циклы.	192
18.1 Простые циклы.	192
18.1.1 Цикл продольного точения (G77).	192
18.1.2 Простой цикл нарезания резьбы (G78).	194
18.1.3 Цикл торцевого точения (G79).	196
18.1.4 Использование простых циклов.	198
18.2 Сложные циклы.	199
18.2.1 Цикл чернового точения (G71).	199
18.2.2 Цикл торцевого чернового точения (G72).	210
18.2.3 Цикл повторения фигуры (G73).	216
18.2.4 Цикл чистового точения (G70).	220
18.2.5 Цикл торцевой проточки (G74).	221
18.2.6 Цикл проточки (G75).	223
18.2.7 Цикл нарезания резьбы (G76).	225
19 Циклы сверления.	231
19.1 Подробное описание циклов сверления.	238
19.1.1 Цикл высокоскоростного глубокого сверления (G83.1).	238
19.1.2 Сверление левой резьбы с уравнильной насадкой (G84.1).	239
19.1.3 Обтачивание с автоматическим отводом инструмента (G86.1).	240
19.1.4 Выключение циклического состояния (G80).	241
19.1.5 Цикл сверления, отвод быстрым ходом (G81).	241
19.1.6 Цикл сверления с задержкой, отвод быстрым ходом (G82).	242
19.1.7 Цикл глубокого сверления (G83).	243
19.1.8 Сверление правой резьбы с уравнильной насадкой (G84).	244
19.1.9 Нарезание резьбы без уравнильной насадки (G84.2, G84.3).	245
19.1.10 Нарезание резьбы без уравнильной насадки, со стружколомом (G84.2, G84.3).	247
19.1.11 Цикл сверления, отвод подачей (G85).	250
19.1.12 Цикл сверления, отвод быстрым ходом при остановленном шпинделе (G86).	251
19.1.13 Ручное воздействие в упорной точке/ Обратная расточка (G87).	252
19.1.14 Цикл сверления, после задержки ручное воздействие в упорной точке (G88).	254
19.1.15 Цикл сверления с задержкой, отвод подачей (G89).	255
19.2 Замечания к использованию циклов сверления.	256
20 Функции управления осями.	257
20.1 Точение многоугольника.	257
20.1.1 Принцип работы точения многоугольника.	257
20.1.2 Программирование точения многоугольника (G51.2, G50.2).	259
20.2 Фрезерование зубчатых колёс способом обкатки (G81.8).	262
20.3 Синхронное перемещение осей.	263
20.4 Перемена осей.	266
20.5 Сверхположенное перемещение осей.	272

20.6 Изменение направления оси.	275
20.7 Обращение осями, не перпендикулярными одной к другой.	277
21 Функции измерения.	284
21.1 Измерение с удалением остаточного хода (G31).	284
21.2 Удаление остаточного хода по пределу момента (G31)..	285
21.3 Автоматическое измерение длины инструмента (G36, G37)..	287
22 Функции безопасности..	289
22.1 Конечное положение.	289
22.2 Ограничение рабочей зоны, устанавливаемой с параметра/программируемой (G22, G23).	290
22.3 Запрещённая изнутри зона.	293
22.4 Наблюдение за запрещённой зоной перед запуском движения..	293
23 Макропрограммирование.	296
23.1 Переменные языка программирования.	297
23.1.1 Ссылка на переменные.	297
23.1.2 Изображение чисел макропеременных.	298
23.1.3 Локальные переменные: #1 – #33.	299
23.1.4 Глобальные переменные: #100 - #499, #500 - #999.	299
23.1.5 Обозначения, используемые при описании системных переменных ...	300
23.1.6 Пустая переменная. Постоянные.	300
23.1.7 Переменные между программой деталей и программой PLC..	301
23.1.8 Сообщения программы деталей..	303
23.1.9 Времена, счётчики заготовок.	305
23.1.10 Переменные, действующие на работу автоматического режима.	307
23.1.11 Запрос состояний поиска кадра и состояний теста.	309
23.1.12 Состояние отражения.	309
23.1.13 Номер главной программы.	310
23.1.14 Модальные информации..	311
23.1.15 Информации о позиции.	316
23.1.16 Значение актуальной коррекции по длине.	318
23.1.17 Прочие информации о позиции.	319
23.1.18 Значения накопителя коррекции инструментов..	320
23.1.19 Смещение нулевой точки заготовки.	320
23.1.20 Прочтение данных инструмента, имеющихся в шпинделе и в дежурных магазинах.	324
23.1.21 Прочтение данных палитр, находящихся на рабочем месте и на месте монтажа.	326
23.2 Команды языка программирования.	328
23.2.1 Команда для присвоения значения: #i = #j.	328
23.2.2 Арифметические действия.	328
23.2.3 Логические действия..	329
23.2.4 Функции.	331
23.2.5 Команды конверсии..	332
23.2.6 Порядок выполнения сложных арифметических действий.	333
23.2.7 Условные выражения.	333
23.2.8 Безусловное разветвление: GOTO n.	334

23.2.9 Условное разветвление.	334
23.2.10 Условная команда.	335
23.2.11 Организация цикла..	335
23.2.12 Косвенные ссылки на оси.	339
23.2.13 Команды выдачи данных..	339
23.3 Вызов макрокоманд, системных макросов, системных подпрограмм.	346
23.3.1 Простой макровывод (G65)..	349
23.3.2 Модальный вызов макроса после каждой команды движения: (G66). ...	351
23.3.3 Модальный макровывод из каждого кадра: (G66.1)..	353
23.3.4 Вызов системного макроса по коду G, заданному параметром..	354
23.3.5 Вызов системного макроса по коду M, заданному параметром.	357
23.3.6 Вызов системной подпрограммы по коду M, заданному параметром. ...	358
23.3.7 Вызов системной подпрограммы по коду A, B, C, S, T, разрешённому параметром..	360
23.3.8 Вызов системной подпрограммы по коду ASCII, заданному параметром	361
23.3.9 Отображение кадров макросов и подпрограмм в автоматическом режиме.	362
23.4 Макрос прерывания.	363
23.5 Команды NC и макроса. Выполнение макрокадров.	367
23.6 Макроцикл фрезерования полости.	369
24 Писание и чтение параметров.	373
24.1 Писание параметров из программы деталей (G10 L52).	374
24.2 Чтение параметров из программы деталей (PRM)..	375
Записи.	377

14 Декабрь 2023 г.

© Copyright NCT 14 Декабрь 2023 г.

По содержанию настоящего описания все издательские права сохраняются за собой. Для допечатки даже сокращённого издания требуется наше разрешение.

Описание составлено с максимальной осмотрительностью и тщательно проверено, однако за возможные ошибки или ошибочные данные *и за истекающие из этого ущербы не берём на себя ответственность.*

1 Введение

1.1 Программа деталей

Программа деталей является совокупностью таких команд, которых управление способно истолковать и на основании которых управляет работой станка.

Программа деталей состоит из кадров. Кадры образуются словами.

Слово: Адрес и Данные

Слово состоит из двух частей: из адреса и из данных. Адрес представляет собой один или несколько символов, а данные - числовое значение, которое может иметь и целое и десятичное значение. Некоторые из адресов могут получить знак, или оператор I. Адреса можно задавать в программе и строчной и прописной буквой, обе варианта принимаются управлением.

Адреса:

Адреса	Значение	Лимит
O	номер программы	0001 - 9999
/	опциональный кадр	1 - 9
N	номер кадра	1 - 999999999
G	подготовительная функция	*
X, Y, Z, U, V, W	координаты длины	I, -, *
A, B, C,	координата угла	I, -, *
R	радиус окружности, вспомогательные данные	I, -, *
I, J, K	координата центра окружности, вспомогательная координата	-, *
E	вспомогательная координата	-, *
F	подача	*
S	число оборотов шпинделя	*
M	смешанная функция	1 - 999999999
T	номер инструмента/номер коррекции	1 - 999999999
D	номер коррекции	1 - 999
L	число повторений	1 - 999999999
P	вспомогательные данные, время задержки	0
Q	вспомогательные данные	0
,C	длина стороны фаски	-, *
,R	радиус скругления	-, *
,A	угол направления прямой	-, *

Адреса, имеющие знак * в столбе лимита, данные могут принимать и десятичные значения.

Для адресов, имеющие знак I и знак -, можно задавать адресу инкрементный оператор или знак.

Не отображаются, и не сохраняются знаки +.

Кадр

Кадры образуются словами.

Кадры отделены в накопителе друг от друга символом LF (Line Feed). В кадрах не обязательно пользоваться номером кадра.

Главная программа и подпрограмма

Программу деталей можно делить на две главные части:

на главные программы, и

на подпрограммы.

Обработка детали описана главной программой. Если в ходе обработки приходится обрабатывать повторяющиеся фигуры в главной программе в различных местах, то эти фрагменты программы не нужно опять описать в главной программе, а можно для этого организовать подпрограмму (субрутин), вызываемую из произвольного места, хоть из другой подпрограммы. Из подпрограммы можно возвращаться в вызывающую программу.

Формат программы в накопителе

Программы деталей, находящиеся в накопителе, представляют собой текстовые файлы с кодировкой ASCII.

1.2 Канал

Обычно на одном станке одновременно работает один инструмент, движением которого управляет программа деталей через устройство ЧПУ.

Если то же самое устройство ЧПУ одновременно управляет траекторией двух или больше инструментов, независимых от движения другого выполнением двух, или больше программ деталей, то речь идёт о работе многоканальной работы.

К каждому каналу можно выделить по одной программе деталей для выполнения.

Если на станке работает одновременно только один инструмент, однако установлен на станок например, загрузочный механизм заготовки, где передвижение заготовки необходимо программировать, в то же время инструмент выполняет точение, даже при этом требуется многоканальная работа.

В те точки программы, где ***приходится дожидаться друг друга движениям траекторий инструментов***, можно запрограммировать ***точки синхронизации***.

К каждому каналу принадлежит отдельная таблица нулевой точки заготовки, таблица коррекции инструментов и макропеременные. Можно обобщить таблицу коррекции инструментов и часть макропеременных, выделяемую параметром для всех каналов.

Основные исполнения устройств ЧПУ типа являются одноканальными, а опционально пригодны для многоканальной работы. Для каждого устройства ЧПУ можно построить не более 8 каналов.

В устройстве ЧПУ в каждом канале можно параметром определить способ работы канала. Это в устройстве ЧПУ типа ***может быть двойко: токарный или фрезерный канал***. В одном устройстве ЧПУ каналы могут быть и смешанными. Например:

1-й канал: токарный

2-й канал: токарный

3-й канал: фрезерный

В дальнейшем это описание содержит описание программирования токарного канала. Если в данном устройстве ЧПУ выстроен и фрезерный канал, его описание программирования содержит книжка под названием описание программирования устройства ЧПУ для управления фрезерным станком и центром обработки типа NCT3xxM.

1.3 Основные понятия

Интерполяция

Устройство ЧПУ способно в ходе обработки перемещать инструмент в вдоль прямой и круговой траектории. Эту деятельность в дальнейшем называем интерполяцией.

Движение инструмента в вдоль прямой:

программа:

G01 X__ Z__

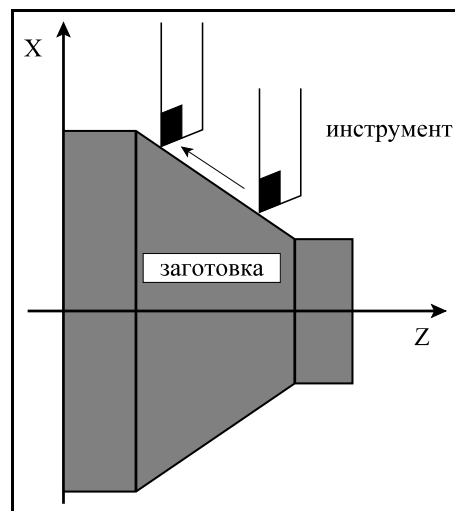


рисунок 1.3-1

Движение инструмента в вдоль дуги круга:

программа:

G02 X__ Z__ R__

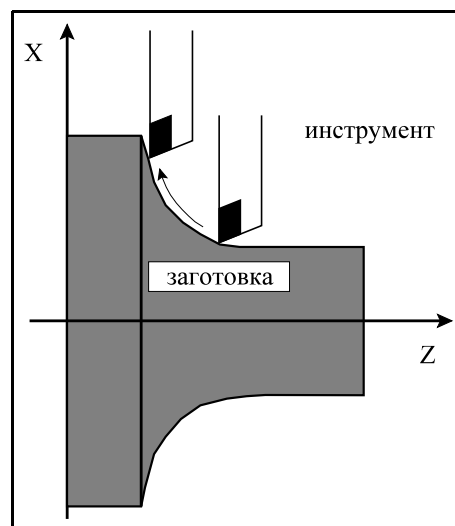


рисунок 1.3-2

Подготовительные функции (коды G)

Тип деятельности, выполняемой в одном данном кадре, описываем с помощью подготовительных функций, или по-иному кодов G. Например: кодом G01 вводится прямая интерполяция.

Подача

Подачей называется скорость инструмента относительно заготовки во время точения. В программе задаётся желаемая подача адресом F и числовым значением. Например: F2 означает 2 мм/оборот.

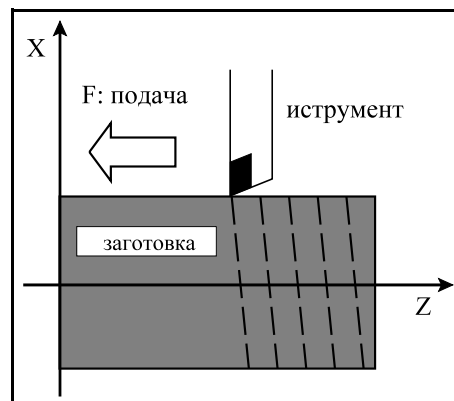


рисунок 1.3-3

Референтная точка

Референтная точка представляет собой постоянную точку на станке, выделённую производителем, характерную для станка. После включения станка суппорты нужно направить в референтную точку, если на станке нет абсолютного измерительного инструмента. Для приёма референтной точки встроен специальный режим в устройство ЧПУ. После этого устройство ЧПУ уже может выполнить и абсолютные команды позиции.

Система координат

Размеры, указанные на рисунке детали, понимаются относительно данной точки детали. Эта точка является нулевой точкой системы координат заготовки. В программу деталей надо записывать эти данные по размеру в адреса координат. Например: X150 Z-100 означает: точка системы координат заготовки с координатой 150; и -100 мм в направлении X и Z.

Для того, чтобы устройство ЧПУ сумел истолковать запрограммированные данные координат, надо задавать расстояние между станочной нулевой точкой и нулевой точкой заготовки. Это происходит замером нулевой точки заготовки.

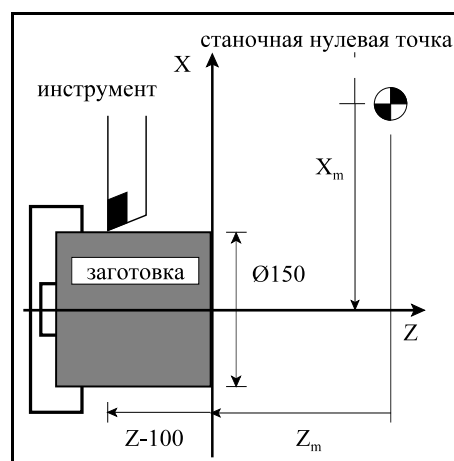


рисунок 1.3-4

Задача абсолютных координат

В случае задачи абсолютных координат инструмент перемещается на расстояние, отсчитанное от точки начала системы координат, то есть в точку с позицией, заданной координатой.

Код задачи абсолютных данных: G90.

Ряд команд

G90 X150 Z2

перемещает инструмент в точку с вышеуказанной позицией, где бы не стоял до выдачи команды.

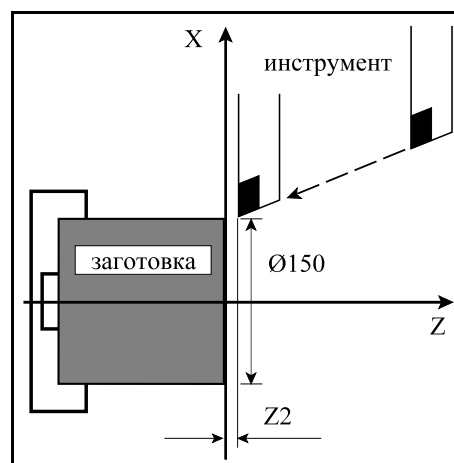


рисунок 1.3-5

Задача координат (инкрементных) с приращением

В случае задачи координат с приращением данные координат понимаются устройством ЧПУ таким образом, чтобы инструмент перемещался на расстояние, отсчитанное от мгновенной позиции.

Код задачи инкрементных данных: G91. Код G91 относится ко всем значениям координат.

Ряд команд

```
G91 X-50 Z-125
```

перемещает инструмент от предыдущей позиции на вышеуказанное расстояние.

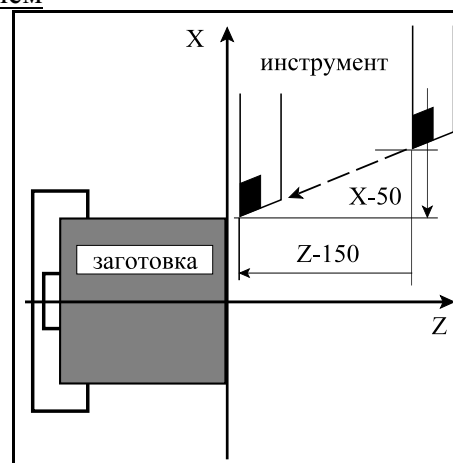


рисунок 1.3-6

Программирование по диаметру

Размер в направлении X можно запрограммировать и по диаметру путём установки параметра.

Модальные функции

В языке программы действие некоторых команд, или величина их значения сохраняются в виде модальных до тех пор, пока не выдаётся команда противоположного значения, или не задаётся другое значение для соответствующей функции. Например: В фрагменте программы

```
N15 G90 G1 X20 Z30 F0.2
N16 X30
N17 Z100
```

принятое в кадре N15 состояние G90 (абсолютная задача данных) и G1 (линейная интерполяция), а также значение F (подача) передаются в виде модальных для кадра N16 и N17. Таким образом не нужно задавать эти коды с кадра в кадр.

Не сохраняющиеся (разовые) функции

Действие некоторых функций, или значение данных имеет силу только в данном кадре. Эти функции называем не сохраняющимися, или разовыми функциями.

Команда чисел оборотов шпинделя

Число оборотов шпинделя задаётся по адресу S. Это обычно называется ещё и функцией S. Команда S1500 назначает, чтобы шпиндель вращался с числом оборотов 1500 об/мин.

Расчёт постоянной скорости резания

Устройство ЧПУ автоматически изменяет число оборотов шпинделя в зависимости от диаметра так, чтобы скорость вершины инструмента была постоянной по сравнению с поверхностью заготовки. Это называется расчётом постоянной скорости резания.

Номер инструмента

В ходе обработки приходится выполнить разные операции со снятием стружки с разными инструментами. Инструменты различаем друг от друга с номерами. Ссылаться на инструменты можно кодом T. Первые двухзначные цифры кода T - это код инструмента (то есть в какой по номеру позиции имеется в револьверной головке), а вторые двухзначные цифры кода T - это номер группы коррекции, относящийся к выбранному инструменту. В программе команда

T0212

означает, что выбрали инструмент с номером 02 и назначали к нему группу коррекции с номером 12.

Смешанные функции

В ходе обработки приходится многократно выполнить операцию включения и выключения. Например: запускать шпиндель, включить охлаждающую воду. Эти операции можно выполнить с помощью смешанной функции или функции M. Например: в ряду команд

M3 M8

значение M3: вращение шпинделя по часовой стрелке, а значение M8: включить охлаждающую воду.

Коррекция по длине

В ходе обработки выполняем различные операции с разными по длине инструментами. Однако, в случае в производстве большой серии ту же операцию приходится выполнять также с разными по длине инструментами, например, из-за полома инструмента. Для того, чтобы описанные в программе детали движения были независимыми от длины инструмента, то есть от его вылета, надо сообщить устройству ЧПУ различные по длине инструментов. Для этого надо измерить длину инструментов. При желании, чтобы вершина инструмента перемещалась в заданную точку, надо вызвать того значения данных по длине, которого в ходе замера задавали.

Это выполняется согласно второму двузначному разряду кода T. С этого момента устройство ЧПУ перемещает вершину инструмента в заданную точку.

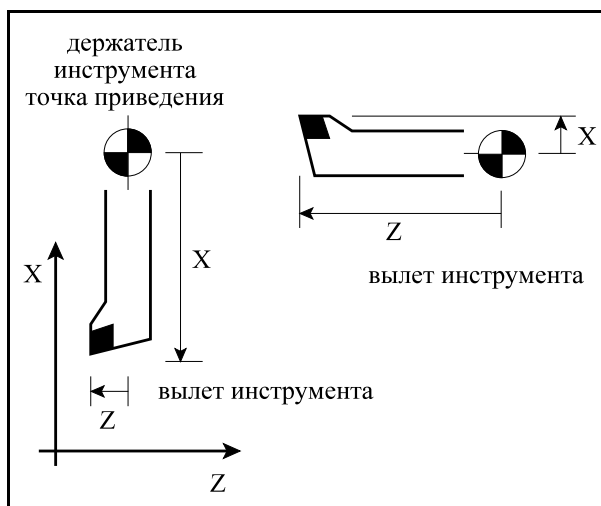


рисунок 1.3-7

Коррекция радиуса

Если при обтачивании контура инструмент совершает движение, не параллельно осями, точный размер получаем в том случае, если проводим по контуру не вершину инструмента, а проводим центр радиуса инструмента перпендикулярно к контуру, на расстояние от него, равное радиусу (r). Для того, чтобы нужно было записать в программу не траекторию центра инструмента, с учётом радиуса инструментов, а фактические данные контура заготовки, нужно вводить коррекцию по радиусу. В группе коррекции, вызванному в программе по адресу T, надо задавать радиус инструмента.

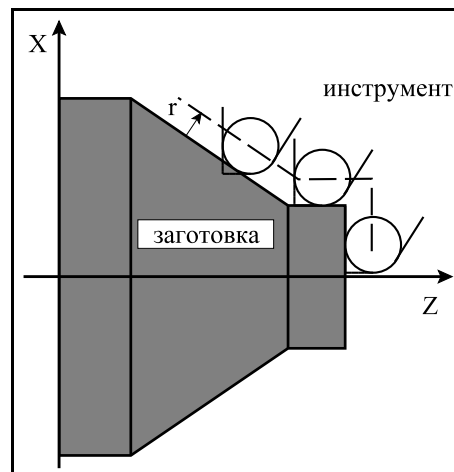


рисунок 1.3-8

2 Управляемые оси

Число осей в основной конструкции	2 оси
Число дополнительных осей	ещё 14 осей не более в том же канале
Максимальное число осей	всего 32 осей, в нескольких каналах

2.1 Название и номер осей

Название управляемых осей можно определить в накопителе параметров. Здесь можно выделить, чтобы какая физическая ось на какой адрес должна перемещаться.

В основной конструкции в устройстве ЧПУ токарным станком название осей: X и Z.

Эти оси устанавливаются параметром N0103 Axis to Plane в качестве **главных осей**.

Название дополнительных осей зависит от типа оси.

Если на токарном станке в качестве дополнительных осей встроены и ось Y, то и ось Y надо устанавливать главной осью параметром N0103 Axis to Plane.

Возможное название дальнейших дополнительных осей, совершающих линейное движение: U, V и W. Если эти оси параллельны какому-то из главных направлений, то **параллельная** оси X **дополнительная ось** имеет название U, а параллельная оси Y - называется V, и параллельная оси Z - называется W.

Параметром N0103 Axis to Plane устанавливается, что какая-то линейная дополнительная ось параллельна ли какой-то основной оси.

Название осей, совершающих вращательное движение: A, B и C. Ось вращения, параллельная направлению оси X имеет название A, параллельная направлению Y называется B, а параллельная направлению Z называется C. Установлением параметра N0106 Axis Properties в ROT=1 можно задавать, что ось является ли осью вращения.

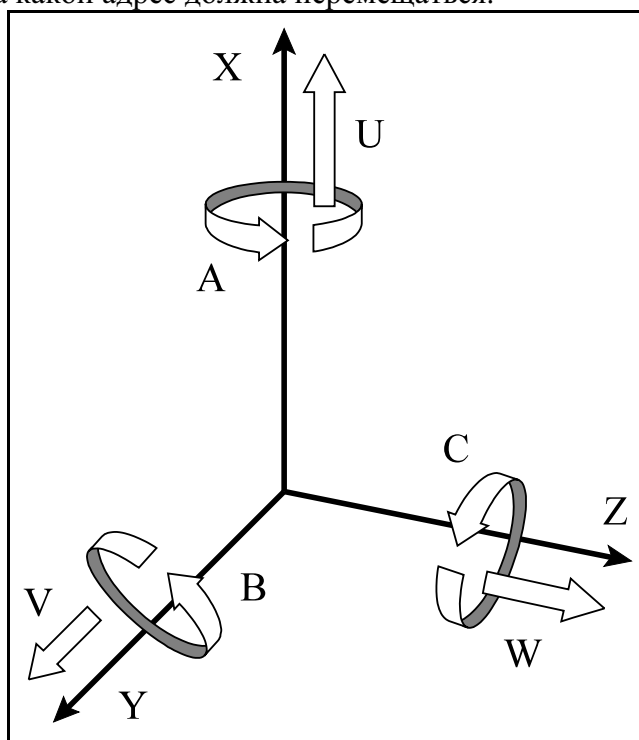


рисунок 2.1-1

Учёт данной **оси** внутри устройства ЧПУ происходит **на основании их номера**. Название осей также надо привязать к номеру оси. Обычная установка: X является осью № 1, Y является осью №2, а Z является осью №3.

В некоторых командах программы, например, в случае ссылки на макропеременные, запрашивающие позицию оси, надо задавать не название оси, а номер оси.

Нумерацию осей спрашивайте у строителя станка.

2.2 Расширенные названия осей

В сложных, многошпиндельных станках не достаточно пользоваться вышеуказанными не более 9 буквами для названия всех осей. Поэтому нужно было вводить расширенные названия осей, при котором не одной буквой, а *не более 3 символами можно задавать название оси*.

Первым символом должен быть X, Y, Z, U, V, W, A, B, или C, чтобы было чего задавать параметром N0100 Axis Name1. Этот параметр надо заполнять даже тогда, если используются названия осей с одним символом.

2-ое, а также 3-ье название символа можно задавать параметром N0101 Axis Name2 и N0102 Axis Name3. Эти могут быть буквы английского алфавита: A, B, C, D, ... Y, Z, а также цифры: 0, 1, 2, ..., 9. Если не использовать название оси 2, или 3, значение параметра 0.

Таким образом можно задавать например, и название оси XDE, но можно использовать и название оси Z1 и Z2.

Если название оси заканчивается на букву, принадлежащее к нему значение можно записать рядом.

XDE127.81

означает: пусть ось XDE перемещается в позицию 127.81..

Если название оси заканчивается номером, после название оси всегда надо написать знак =.

Z1=87.257

означает: пусть ось Z1 перемещается в позицию 87.257.

Естественно, и при использовании расширенными названиями осей надо задавать параметром N0103 Axis to Plane, что какая из осей является главной или основной осью, далее какая из них является параллельной осью.

В дальнейшем в описании используются обычно название осей с одним символом.

2.3 Назначение осей к каналам

Строителем станка назначаются отдельные оси с помощью параметров к различным каналам. Эти установки параметров означают положение после включения.

Оси назначаются к отдельным каналам всегда *на основании их номера*. Значит, *номер оси* является внутри устройства ЧПУ *глобальным*, а *названия осей локальными*, назначенные к отдельным каналам. Естественно, внутри одного канала нельзя иметь ось с таким же названием, однако в разных каналах да. Внутри одного канала устройство ЧПУ управляет не более 16 осями.

Например:

1-й канал:

1-ая ось: X

2-ая ось: Y

3-ья ось: Z

4-ая ось: C

2-й канал:

5-ая ось: X

6-ая ось: Y

7-ая ось: Z

8-ая ось: C

В ходе обработки может понадобиться, чтобы одну или несколько осей использовать в другом канале. В таком случае две оси можно поменять между двумя каналами, одну ось можно передавать другому каналу. Смена происходит на основании номера оси. После смены название оси может сохраниться, но может и измениться.

Смена осей осуществляется строителем станка через программу PLC, например, с помощью функции M. Описание этого содержит всегда инструкция данного станка.

2.4 Система мер осей, и точность отображения позиции

Данные координат задаются *с точностью до 15 цифр* не более. Десятичный знак надо выставить только тогда, если надо позиционировать в точку с не целой координатой. Данные координат могут иметь и знак. Знак “+” не выставится перед цифрой.

Данные координат длины можно задавать в программе в *мм*-ах и в *дюйм*-ах. Это есть **входная система мер**. Входную систему мер можно выбирать из программы кодом G (G21/G20)).

Измеритель хода, оборудованный на станок, может измерить позицию и в мм-ах и в дюйм-ах. Измеритель хода определяет **выходную систему мер**, которую можно задавать устройству ЧПУ на бите IND параметра N0104 Unit of Measure. Внутри одного станка нельзя смещать выходные системы мер между осями.

Если входная и выходная система мер отличается друг от друга, переход выполняется устройством ЧПУ автоматически.

Система мер осей вращения использует всегда *градус*. *Оси вращения можно выделить установкой ROT=1 параметра N0106 Axis Properties. Правильная установка этого параметра важна потому, что устройство ЧПУ для выделённых таким образом осей не выполняет переход дюйм/метрической системы.*

Параметром N0105 Increment System можно устанавливать, что **индикацию позиции с точностью** до скольких десятичных знаков желаем выполнить.

Внутреннее отображение позиции системы не зависит от значения параметра Increment System. Точность:

Для продольных осей в случае метрического измерения: 10^{-6} мм,

Для продольных осей в случае дюймового измерения: 10^{-7} дюйм,

В случае круговых осей: 10^{-6} градус.

Знак	ось	Точность отображения в зависимости от системы мер	
		G21 метрическая	G20 дюймовая
ISA	продольная	0.01 мм	0.001 дюйм
	вращающая	0.01 градус	0.01 градус
ISB	продольная	0.001 мм	0.0001 дюйм
	вращающая	0.001 градус	0.001 градус
ISC	продольная	0.0001 мм	0.00001 дюйм
	вращающая	0.0001 градус	0.0001 градус
ISD	продольная	0.00001 мм	0.000001 дюйм
	вращающая	0.00001 градус	0.00001 градус
ISE	продольная	0.000001 мм	0.0000001 дюйм
	вращающая	0.000001 градус	0.000001 градус

С установкой положения DIA=1 параметра N0106 Axis Properties можно произвольно выделить любую ось для ввода данных и индикацию позиции по *диаметру*.

3 Подготовительные функции (коды G)

Характер команды в данном кадре определяется адресом G и последующим за ним числом. Имеются так называемые *одноразовые коды G*, эффект которых действителен только в данном кадре, и имеются так называемые модальные, или *наследственные функции G*, эффект которых действует до тех пор, пока другим кодом G не выключаем его, или не изменяем этот эффект.

Одноразовые коды G входят в 0-ую группу. Среди модальных кодов G, действующих друг на друга, они получают номер группы, отличающийся от 0.

В одном кадре можно писать несколько кодов G, таким ограничением, что по группам может фигурировать только одна функция, принадлежащая к той же группе.

В код не надо записывать ведущий ноль, но устройство ЧПУ примет его. Например: в программе можно писать и G01 и G1.

Следующая таблица содержит коды G, интерпретируемые устройством ЧПУ, а также их номер группы и функции.

Код G	Группа	Функция	Стр.
G0*	1	позиционирование	26
G1*		линейная интерполяция	28
G2		круговая и винтовая (плоскостная, цилиндрическая, коническая) интерполяция по ходу часовой стрелки	30, 35, 37, 40
G3		круговая и винтовая (плоскостная, цилиндрическая, коническая) интерполяция против хода часовой стрелки	30, 35, 37, 40
G4	0	задержка	79
G9		точный останов в данном кадре	66
G7.1		цилиндрическая интерполяция	49
G10		запрограммированный ввод данных	90, 91, 129,
G11		выключение запрограммированного ввода данных	
G10.9		переключение программирования по радиусу/по диаметру из программы	54
G12.1	21	интерполяция полярной координаты вкл.	44
G13.1*		интерполяция полярной координаты выкл.	44
G15*	17	задание данных полярными координатами выкл.	56
G16		задание данных полярными координатами вкл.	56

Код G	Группа	Функция	Стр.
G17*	02	выбор плоскости $X_p Y_p$	95
G18*		выбор плоскости $Z_p X_p$	95
G19*		выбор плоскости $Y_p Z_p$	95
G20*	6	дюймовое задание данных	53
G21*		метрическое задание данных	53
G22*	04	ограничение рабочей зоны вкл.	290
G23*		ограничение рабочей зоны выкл.	290
G28	0	запрограммированный приём референтной точки	81
G30		наводка во вторую, третью и четвёртую референтную точку	82
G31		измерение с удалением остаточного хода	284
G33	01	нарезание резьбы	42
G36	0	автоматическое измерение длины инструмента в направлении X	287
G37		автоматическое измерение длины инструмента в направлении Z	287
G38		сохранение вектора коррекции радиуса	160
G39		угловая дуга с коррекцией радиуса	161
G40*	07	выключение расчёта коррекции радиуса инструмента	134
G41		расчёт коррекции радиуса инструмента слева	134
G42		расчёт коррекции радиуса инструмента справа	134
G43.7*	08	учёт коррекций по длине инструмента по коду G	133
G49*		коррекция по длине выкл.	130
G50*	11	масштабирование выкл.	173
G51		масштабирование вкл.	173
G50.1*	22	выключение отражения	176
G51.1		включение отражения	176
G51.2	31	обтачивание многоугольника вкл.	259
G50.2*		обтачивание многоугольника выкл.	259
G52	0	смещение координат	92
G53		позиционирование в системе координат станка	85
G54*	14	выбор системы координат первой заготовки	87

Код G	Группа	Функция	Стр.
G55		выбор системы координат второй заготовки	87
G56		выбор системы координат третьей заготовки	87
G57		выбор системы координат четвёртой заготовки	87
G58		выбор системы координат пятой заготовки	87
G59		выбор системы координат шестой заготовки	87
G54.1		выбор расширенной системы координат заготовки	88
G61	15	режим точного останова	68 66
G62		уменьшение подачи при углов	66
G63		запрет override	66
G64*		режим непрерывной обработки резанием	66
G65	0	простой макровывоз	349
G66	12	модальный макровывоз после каждой команды движения	351
G66.1		модальный макровывоз из каждого кадра	353
G67*		удаление модального макровывоза	351
G68	16	плоскостной поворот конфигурации вокруг данной точки	171
G69*		плоскостной поворот выкл.	171
G70	0	чистовой цикл	220
G71		черновой цикл	199
G72		торцевой черновой цикл	210
G73		цикл повторения шаблона	216
G74		цикл торцевой проточки	221
G75		цикл проточки	223
G76		цикл нарезания резьбы	225
G77	1	цикл продольного точения	192
G78		цикл простого нарезания резьбы	194
G79		цикл торцевого точения	196

Код G	Группа	Функция	Стр.
G83.1	09	цикл высокоскоростного глубокого сверления	238
G84.1		сверление левой резьбы с уравнильной насадкой	239
G86.1		обтачивание с автоматическим отводом инструмента	240
G80*		выключение циклического состояния	241
G81		цикл сверления, отвод быстрым ходом	241
G82		цикл сверления с задержкой, отвод быстрым ходом	242
G83		цикл глубокого сверления	243
G84		сверление правой резьбы с уравнильной насадкой	244
G84.2		сверление правой резьбы без уравнильной насадкой сверление правой резьбы без уравнильной насадкой со стружколомом	245
G84.3		сверление левой резьбы без уравнильной насадкой сверление левой резьбы без уравнильной насадкой со стружколомом	245
G85		цикл сверления, отвод подачей	250
G86		цикл сверления, отвод быстрым ходом при остановленном шпинделе	251
G87		вытачивание возвратом, автоматическим/ручным отводом инструмента	252
G88		цикл сверления, ручное воздействие в упорной точке	254
G89		цикл сверления, задержка в упорной точке, отвод подачей	255
G80.8	34	электронный привод выкл.	262
G81.8		электронный привод вкл.	262
G90*	03	абсолютное задание размера	52
G91*		задание размера с приращением	52
G92	00	создание новой системы координат заготовки	91
G94*	05	поминутная подача	65
G95*		подача за оборот	65
G96	13	расчёт постоянной скорости резания вкл.	101
G97*		расчёт постоянной скорости резания выкл.	101
G98*	10	возвращение из цикла сверления в исходную точку	232
G99		возвращение из цикла сверления в точку R (приближения)	232

Исходное положение после включения

Внутри одной группы кодов G, обозначенных с * означают то состояние, которое примет устройство ЧПУ *после включения*.

Там, где внутри одной группы после нескольких кодов G имеется знак *, там на основании параметра N1300 DefaultG1 и N1301 DefaultG2 можно выбирать, какой из них будет действовать *после включения*. Эти следующие:

G00, G01;
G17, G18, G19;
G20, G21;
G22, G23;
G90, G91;
G94, G95.

Исходное положение после нажатия клавиши Reset, а также после конца программы

В следствие нажатия клавиши *reset*, или *конца программы* (M2, M30), если бит CLR параметра N1301 DefaultG2 имеет значение

=0: устройство ЧПУ безусловно принимает состояние, обозначенное в таблице кода G со знаком *, далее восстанавливает значение кода G в исходное положение, имеющееся после включения, установленное параметром N1300 DefaultG1 и N1301 DefaultG2,

=1: на основании значений, заданных параметром CLR G Table1, 2, 3, 4, 5 устройство ЧПУ занимает исходное положение, согласно группам кода G, или оставляет его модальное значение без изменения.

Если бит Cnn (где номер группы кода G будет nn), относящий к группе кода G в параметре CLR G Table1, 2, 3, 4, 5 имеет значение:

=0: устройство ЧПУ приведёт соответствующую группу кода G в исходное положение,

=1: а устройство ЧПУ оставляет соответствующую группу кода G в создавшемся, модальном положении.

4 Интерполяция

4.1 Позиционирование (G0)

Команда позиционирования G0 перемещает инструменты в долъ всех осей, запрограммированных в кадре в заданную точку.

Перемещение выполняется быстрым ходом. Величина быстрого хода определяется на параметре строителем по осям, его из программы устанавливать нельзя.

В случае абсолютного задания данных актуальная заготовка перемещается в системе координат в точку заданной позиции.

В случае инкрементного задания данных инструмент совершает заданное расстояние шагами, отсчитывая от мгновенной его позиции.

Формат кадра:

G0 v

где v - координаты, заданные в кадре. Обозначение v здесь (и в дальнейшем) относится ко всем управляемым осям, использованным в данном канале. Позиционировать можно одновременно в долъ всех осей канала.

Вместо G0 можно задавать и G00.

Пример:

G0 X150 Z2

В кадре можно задавать и другой код G и функция.

Например:

G0 G90 X150 Z2 S2000 M3

Код G0 является модальным, он действителен до тех пор, пока другая команда интерполяции не изменяет его.

Например:

G0 X150 Z2

X20 Z1 (позиционирующий кадр, так как G0 модальный)

После включения в случае положения бита G01=0 параметра N1300 DefaultG1 действителен код G0.

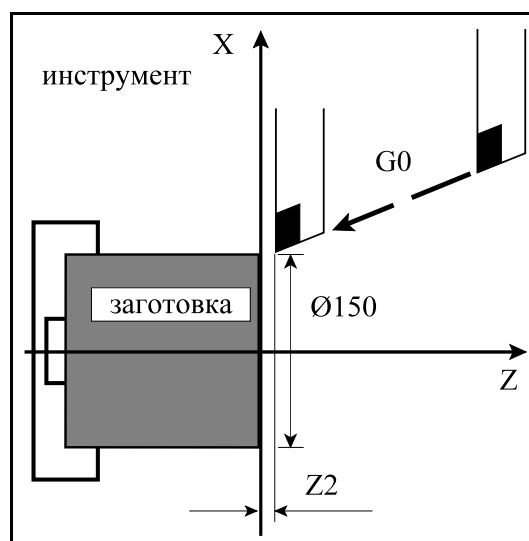


рисунок 4.1-1

4.1.1 Позиционирование линейной интерполяцией

В случае одновременного перемещения нескольких осей, устройство ЧПУ при позиционировании перемещает инструмент в долъ прямой, соединяющей начальную точку с конечной точкой, в случае положения бита ROL=0 параметра N0421 Acc Contr.

Устройство ЧПУ рассчитывает результирующий вектор (v) таким образом, чтобы позиционирование произошло за минимальное время, и скорость не превосходила ни на одной оси значения быстрого хода, установленного для данной оси.

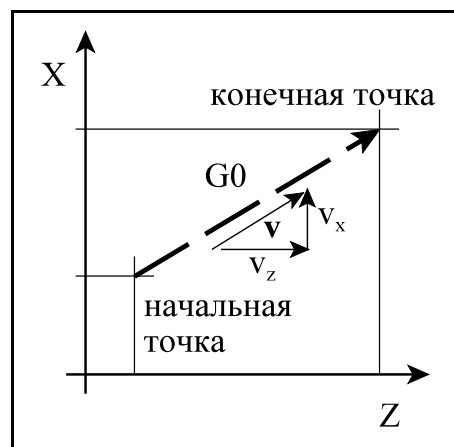


рисунок 4.1.1-1

После завершения движения, в случае положения бита РСН=1 параметра N1337 Execution Config, устройство ЧПУ проверяет сигнал "в позиции".

После сигнала "в позиции" устройство ЧПУ делает задержку на время, установленное параметром N1340 Inpos Timeout, если после этого не приходит сигнал, выводится сообщение 2501 Ошибка позиции.

Параметром N0516 Inpos можно задавать по осям максимальное, но ещё допустимое отклонение, измеренное от позиции.

Установить проверку позиции следует только в обоснованном случае, иначе необоснованно может нарастать время выполнения. Например, выполнив фрагмент программы,

```
G0 X60
Z1
X56
```

на приложенном рисунке видна разность по продолжительности между двумя случаями.

Устройство ЧПУ **всегда выполняет проверку позиции** в конечной точки кадра в состоянии

G61 (режим точного останова), или в том кадре позиционирования, куда записан код

G9 (точный останов), даже при положении параметра РСН=0 (нет проверки позиции).

4.1.2 Позиционирование перекрыванием движений быстрого хода

Позиционирования, выполненные на разных и последующих друг за другом осях, могут быть дальше ускорены перекрыванием движения кадров позиционирования. Это означает, что за время, пока в кадре позиционирования одна из осей замедляет до позиции конечной точки, то другая ось, участвующая в следующем кадре позиционирования, уже начинает ускорять.

Перекрывание включается положением бита ROL=1 параметра N0421 Acc Contr. Перекрывание включается положением бита ROL=1 параметра N0421 Acc Contr.

Параметром N0422 Rapid Reduct. Ratio

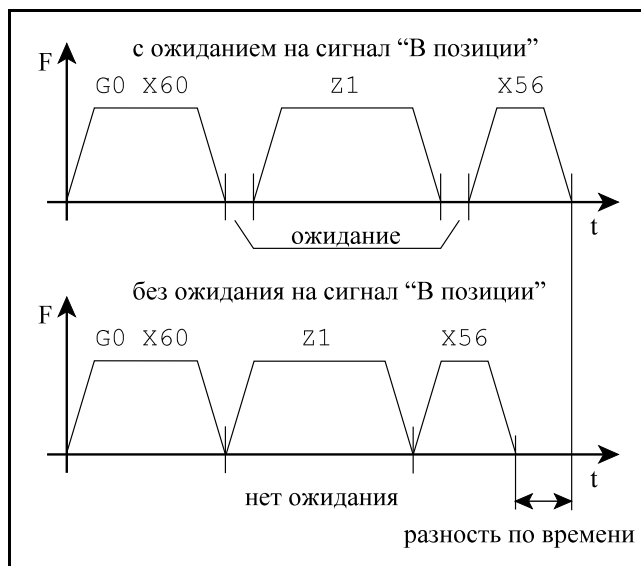


рисунок 4.1.1-2

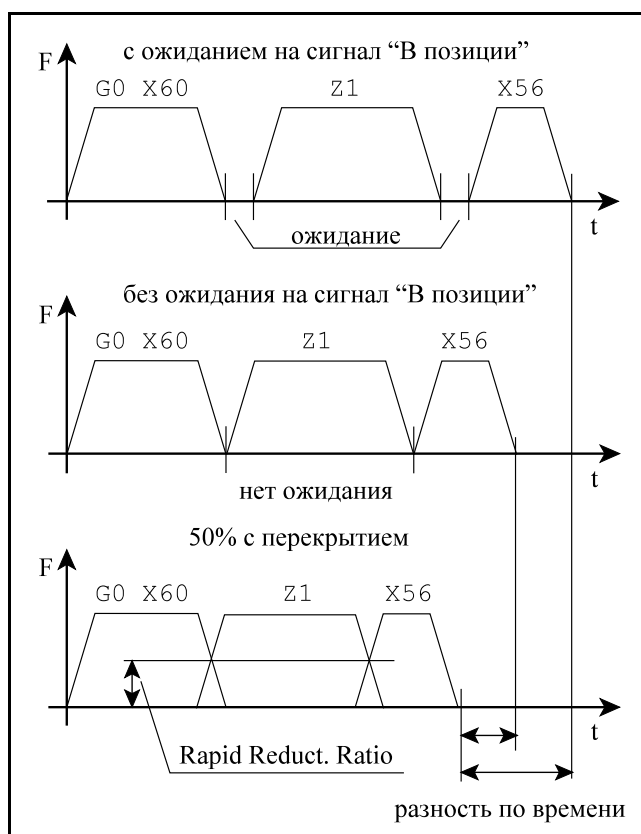


рисунок 4.1.2-1

можно установить в процентах, что в участке замедления предыдущего кадра, после какого в процентах замедления запускается следующий кадр.

Оставшись у примера, представленного предыдущей главой, выполнив фрагмент программы

```
G0 X60
Z1
X56
```

на приложенном рисунке видна разность по времени выполнения.

В случае позиционирования перекрыванием траектория инструмента получится не по угловой, а по скругленной траектории. Из-за этого в программе должны быть осторожными, чтобы от отводили инструмент от заготовки больше привычного.

При запрограммированном в одном кадре позиционировании *нескольких осей* инструмент *перемещается только приблизительно вдоль прямой траектории*, и различные оси доходят до позиции с расхождением по времени.



рисунок 4.1.2-2

Устройство ЧПУ *отложит перекрывание* между кадрами позиционирования и *всегда выполняет проверку позиции* в конечной точке кадра в состоянии

G61 (режим точного останова),

или в том кадре позиционирования, куда записали код

G9 (точный останов),

даже при положении параметра PCH=0 (нет проверки позиции).

4.2 Линейная интерполяция (G1)

Команда линейной интерполяции G1 перемещает инструмент вдоль всех осей, запрограммированных в кадре, в заданную точку вдоль прямой траектории.

Перемещение выполняется с запрограммированной в кадре или модальной подачей F.

В случае абсолютного задания данных актуальная заготовка перемещается в системе координат в точку с заданной позицией.

В случае инкрементного задания данных заданное расстояние совершается шагами, отсчитывая от мгновенной позиции инструмента.

Формат кадра:

G1 v F

где v - заданные в кадре координаты, F- значение подачи. Перемещаться можно одновременно вдоль всех осей канала. Вместо G1 можно задавать и G01.

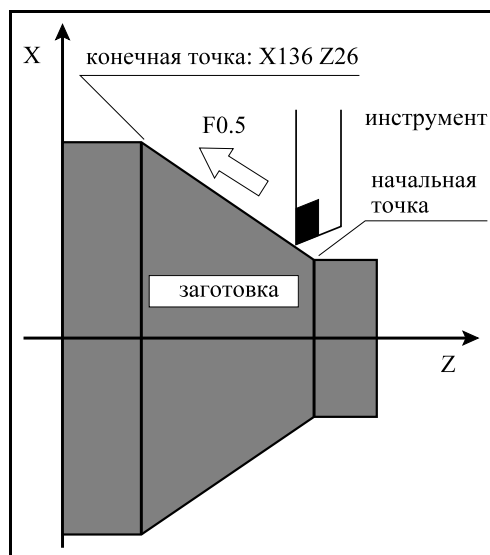


рисунок 4.2-1

Пример:

G1 X136 Z26 F0.5

В кадре можно задавать и другой код G и другая функция.

Пример:

G1 G91 X5 Z10 S2000 M3

G1 модальный код действует до тех пор, пока другая команда интерполяции не изменяет его.

Например:

G1 X136 Z26 F0.5

Z0 (линейная интерполяция с F.5, так как G1 и F модальные)

В случае положения бита G01=1 параметра N1300 DefaultG1, после включения действует код G1.

Подача, запрограммированная по адресу F действует всегда вдоль запрограммированной траектории. Составляющие вдоль оси:

Подача вдоль оси X: $F_x = \frac{\Delta x}{L} F$

Подача вдоль оси Z: $F_z = \frac{\Delta z}{L} F$

.....

Формула продолжается для всех осей, запрограммированных в кадре.

Где: $\Delta x, \Delta z, \dots$: значения перемещения, рассчитанного вдоль соответствующих осей,
L: длина запрограммированного перемещения:

$$L = \sqrt{\Delta x^2 + \Delta z^2 + \dots}$$

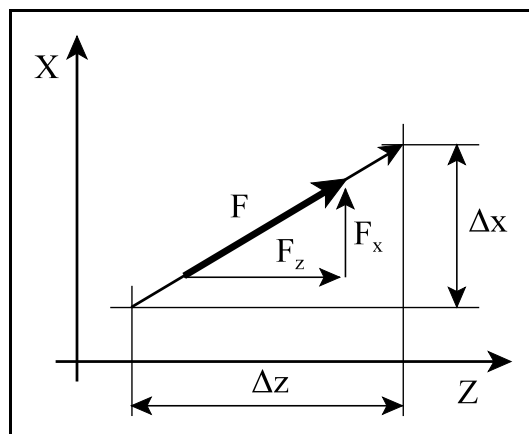


рисунок 4.2-2

Вдоль оси вращения подача понимается в размерности °/мин. В кадре

G1 C270 F120

значение F120: 120 °/мин.

В том случае, если соединить движение линейной оси и оси вращения с линейной интерполяцией, раскладка составляющих подачи происходит на основании вышеуказанных формул.

Например:

в кадре

G91 G01 Z100 C45 F120

составляющие подачи в направлении Z а также

В:



рисунок 4.2-3

Подача вдоль оси Z: $F_z = \frac{100}{\sqrt{100^2 + 45^2}} 120 = 109.4$ мм/мин

Подача вдоль оси C: $F_c = \frac{45}{\sqrt{100^2 + 45^2}} = 49.2$ °/мин

4.3 Круговая интерполяция (G2, G3)

Команда G2, или G3 перемещает инструмент в выбранной плоскости вдоль дуги круга в заданную в кадре точку. Перемещение происходит с запрограммированной в кадре или модальной подачей F.

Формат кадра:

$$G17 \left\{ \begin{matrix} G2 \\ G3 \end{matrix} \right\} X_p - Y_p - \left\{ \begin{matrix} R_- \\ I_- J_- \end{matrix} \right\} F_-$$

$$G18 \left\{ \begin{matrix} G2 \\ G3 \end{matrix} \right\} X_p - Z_p - \left\{ \begin{matrix} R_- \\ I_- K_- \end{matrix} \right\} F_-$$

$$G19 \left\{ \begin{matrix} G2 \\ G3 \end{matrix} \right\} Y_p - Z_p - \left\{ \begin{matrix} R_- \\ J_- K_- \end{matrix} \right\} F_-$$

Круговая интерполяция выполняется в плоскости, выбранной командами G17, G18, G19. В случае **G2** в направлении *по ходу часовой стрелки*, а в случае **G3** *против хода часовой стрелки*:

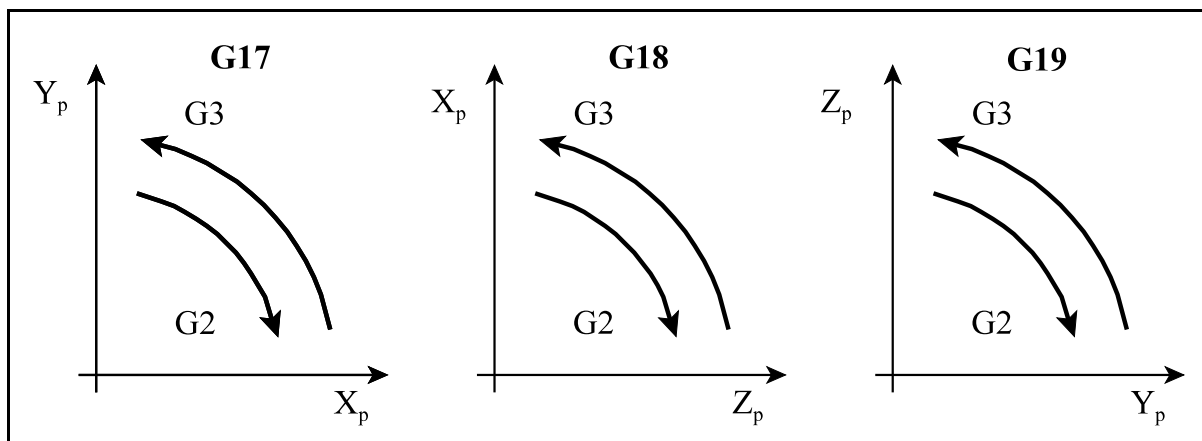


рисунок 4.3-1

В место G2 и G3 можно записать в программу и код G02 и G03.

G2 и G3 модальные коды действуют до тех пор, пока другая команда интерполяции не изменяет их..

Значение X_p , Y_p , Z_p здесь, и в дальнейших:

X_p : ось X, или параллельная ей ось,

Y_p : ось Y, или параллельная ей ось,

Z_p : ось Z, или параллельная ей ось.

Значение X_p , Y_p , Z_p - это **координата конечной точки круга**, заданная в данной системе координат как абсолютные данные, или измеренные от начальной точки инкрементные данные.

Задание дальнейших данных круга может быть двояко:

1-й случай: задание окружности радиусом по адресу R

При этом устройство ЧПУ автоматически вычисляет координаты центра окружности из координаты начальной точки (это та точка, где устройство ЧПУ в момент прочтения периода находится), из координат конечной точки (значение, определённое по адресу X_p , Y_p , Z_p), а также из запрограммированного радиуса окружности R. В случае данного направления обхода (G2, или G3) между начальной и конечной точки можно нарисовать две разные окружности с радиусом R.

Если значение радиуса окружности, то есть **R** задаётся

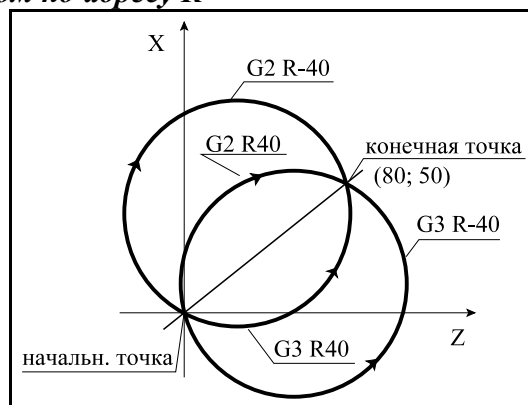


рисунок 4.3-2

положительным числом, устройство ЧПУ интерполирует дугу, **меньшую 180°**
отрицательным числом, то - **большую 180°** .

Например:

1-й участок дуги: G2 X80 Z50 R40

2-й участок дуги: G2 X80 Z50 R-40

3-й участок дуги: G3 X80 Z50 R40

4-й участок дуги: G3 X80 Z50 R-40

2-й случай: задание окружности центром по адресу I, J, K

Устройство ЧПУ истолкует значения, заданные по адресу I, J, K **инкрементно**, так, что вектор, определённый значениями I, J, K показывает от **начальной точки** окружности в **точку центра**.

Значения **I, J, K** должны быть заданы всегда по **радиусу**, даже тогда, если принадлежащие к ним оси установлены для программирования по диаметру.

Например:

В случае G17: G3 X20 Y140 I-50 J-20 (X, Y заданы диаметром)

В случае G18: G3 X140 Z10 I-20 K-50 (X задан диаметром)

В случае G19: G3 Y20 Z70 J-50 K-20 (Y задан диаметром)

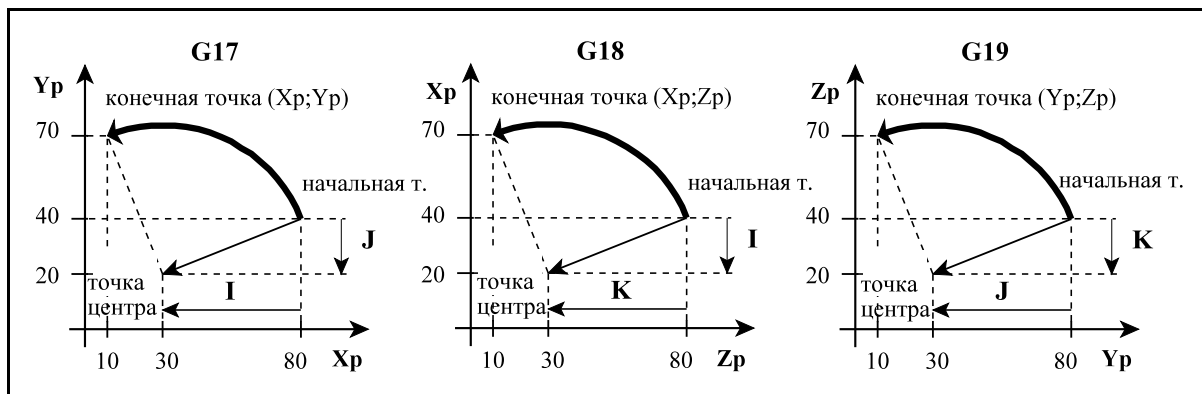


рисунок 4.3-3

По адресу F можно запрограммировать подачу по траектории, которая показывает всегда в направлении касательной окружности и постоянная вдоль всей траектории.

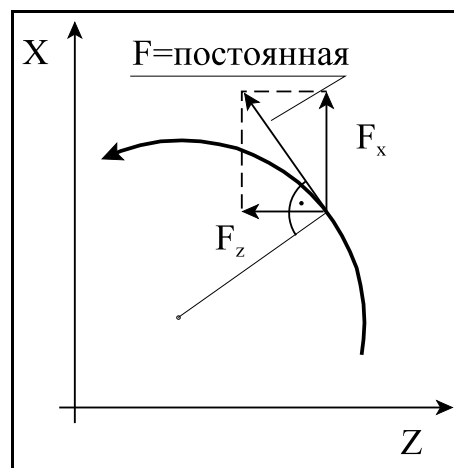


рисунок 4.3-4

Пример:

Программируется траектория на приложенном рисунке.

Программирование окружности абсолютными координатами и заданием R:

```
G90 G18
G0 X0 Z130 M3 S1000
G1 X40 F500
G3 X180 Z60 R70
G2 X100 Z40 R50
G1 Z0
...
```

Программирование окружности абсолютными координатами и заданием центра окружности I, K

```
G90 G18
G0 X0 Z130 M3 S1000
G1 X40 F500
G3 X180 Z60 K-70
G2 X100 Z40 K-50
G1 Z0
...
```

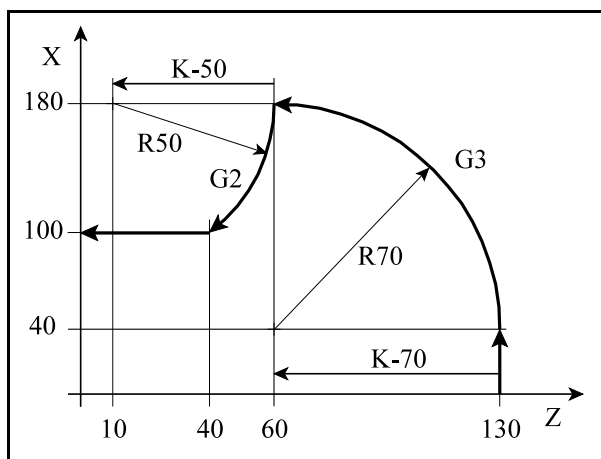


рисунок 4.3-5

Программирование окружности инкрементными координатами и заданием R:

```
G90 G18 G0 X0 Z130 M3 S1000
G91
G1 X40 F500
G3 X140 Z-70 R70
G2 X-80 Z-20 R50
G1 Z-40
...

```

Программирование окружности инкрементными координатами и заданием центра окружности I, K

```
G90 G18 G0 X0 Z130 M3 S1000
G91
G1 X40 F500
G3 X140 Z-70 K-70
G2 X-80 Z-20 K-50
G1 Z-40
...

```

Заданием **I0, J0, K0** можно пренебречь. Например:

```
G0 X0 Z100 F500
G18 G03 X200 Z0 K-100

```

В случае программирования четверти окружности с радиусом 100 мм, с центром начала координат, поскольку центр окружности находится в направлении X на расстояние 0 от центра X0 Z100, не надо написать I0.

Если X_p, Y_p, Z_p , все пренебрежены:

– если задаются координаты точки центра окружности по адресу I, J, K: устройство ЧПУ интерполирует дугу как полную окружность с 360° . Например:

```
G0 X400 Y0 F500
G17 G03 I-100

```

в этом случае устройство ЧПУ интерполирует полную окружность с радиусом 100 мм, с точкой центра X200 Y0

– если задаётся радиус R, например

```
G0 X0 Z200 F500
G18 G03 R100

```

устройство ЧПУ не движется и не выдаёт сообщение об ошибке.

Если период не содержит **ни радиус R, ни I, J, K**, устройство ЧПУ выдаёт сообщение 2015 Ошибка задания окружности.

Если ссылка сделана по адресу **I, J, K вне выбранной плоскости**, устройство ЧПУ 2015 выдаёт сообщение Ошибка задания окружности.

Если **разность радиуса начальной точки и конечной точки окружности**, заданной в кадре G2, G3 больше, чем значение, определённое параметром N1339 Radius Diff, устройство ЧПУ выдаёт сообщение 2012 Ошибка разности радиуса в окружности.

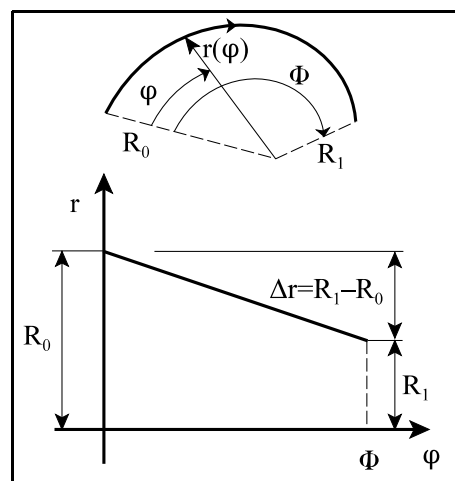


рисунок 4.3-6

Если разность радиуса меньше заданного параметром выше значения устройство ЧПУ перемещает инструмент вдоль такой плоскостной винтовой траектории, у которой радиус линейно меняется в зависимости от центрального угла.

При интерполяции дуги окружности с переменным радиусом, постоянной будет не скорость вдоль траектории, а угловая скорость.

Пусть значение параметра N1339 Radius Diff будет больше, чем 0, например 0.01, иначе устройство ЧПУ выдаёт лишние сообщения.

Ошибку разности радиуса, или интерполяцию дуги окружности с переменным радиусом могут причинить следующие случаи:

Если позиция точки центра окружности, заданная по адресу I, J, K не годная.

Например:

```
G18 G90 G0 X0 Z50
G3 Z-20 K-50
```

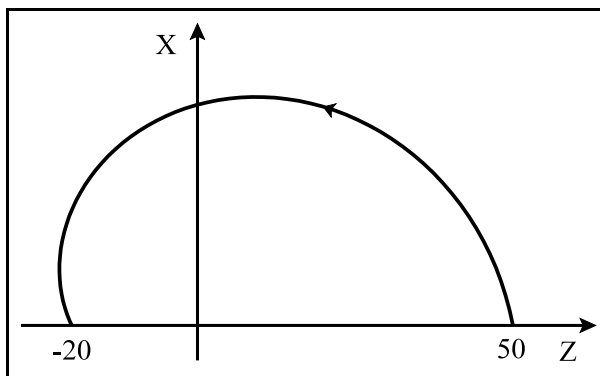


рисунок 4.3-7

Если заданный радиус окружности меньше, чем половина расстояния прямой, соединяющей начальную точку с конечной точкой, устройство ЧПУ считает заданный радиус окружности за радиусом начальной точки окружности, и интерполирует такую окружность с переменным радиусом, точка центра которой лежит на прямой, соединяющей начальную точку с конечной точкой, на расстояние R от точки начала:

```
G18 G0 G90 X0 Z0
G2 X60 Z40 R10
```

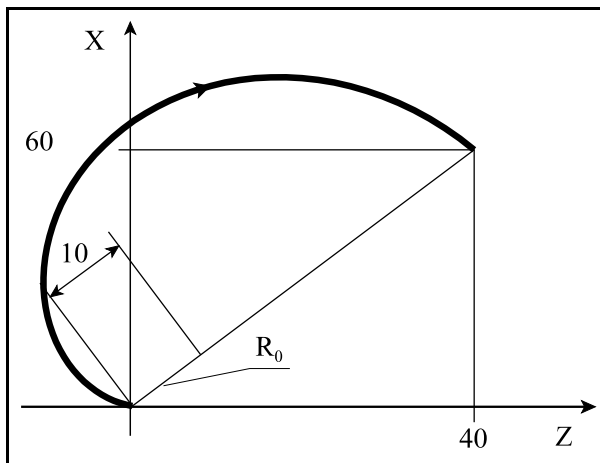


рисунок 4.3-8

Позицию точки центра окружности I, J, K можно задавать и абсолютным значением, рассчитанном от нулевой точки заготовки. Для этого нужна установка бита #2 CCA=1 параметра N1337 Execution Config. Этого случая надо избежать.

4.3.1 Винтовая интерполяция по плоскости (G2, G3)

Командой G2, или G3 можно программировать плоскостную винтовую интерполяцию, что по адресу задавать и число оборотов винта. **Точка центра и конечная точка окружности** задаётся так, чтобы **радиус окружности начальной точки и конечной точки разные**. Перемещение выполняется запрограммированной в кадре или модальной подачей F.

Формат кадра:

$$G17 \left\{ \begin{matrix} G2 \\ G3 \end{matrix} \right\} X_p - Y_p - I - J - L - F -$$

$$G18 \left\{ \begin{matrix} G2 \\ G3 \end{matrix} \right\} X_p - Z_p - I - K - L - F -$$

$$G19 \left\{ \begin{matrix} G2 \\ G3 \end{matrix} \right\} Y_p - Z_p - J - K - L - F -$$

Значение **координат конечной точки винта** X_p , Y_p , Z_p в данной системе координат задаются абсолютными, или измеренные от начальной точки инкрементными данными.

По адресам **I, J, K** задавали **координаты точки центра** винта как **инкрементные** данные, измеренные от начальной точки так, чтобы вектор, определённый значением I, J, K показывает от **начальной точки** винта **к точке центра** винта.

Значения **I, J, K** надо залавать всегда **радиусом**, даже тогда, если принадлежащие к ним оси установлены для прораграммирования по диаметру.

По адресу L задаётся **число оборотов винта**. Каждый заход означает новый оборот, даже тогда, если после этого совершается не полный оборот. Адрес **L представляет собой положительное целое число**.

В ходе винтовой интерполяции устройство ЧПУ линейно изменяет радиус точки начала (R_0) в зависимости от угла поворота (φ) так, что пока совершается поворот, заданный по адресу L и дойдёт до позиции конечной точки, радиус конечной точки соответствовал запрограммированным данным.

Из вышеописанных следует, что в ходе винтовой интерполяции, о чём заполнение адреса L сообщает устройству ЧПУ, радиус начальной точки отличается от радиуса конечной точки. Если заполнен адрес L, устройство ЧПУ никогда не рассматривает значение максимальной разности радиуса, определённое параметром N1339 Radius Diff.

Подача F, заданная в винтовой интерполяции остаётся **постоянной по всей длине винта**.

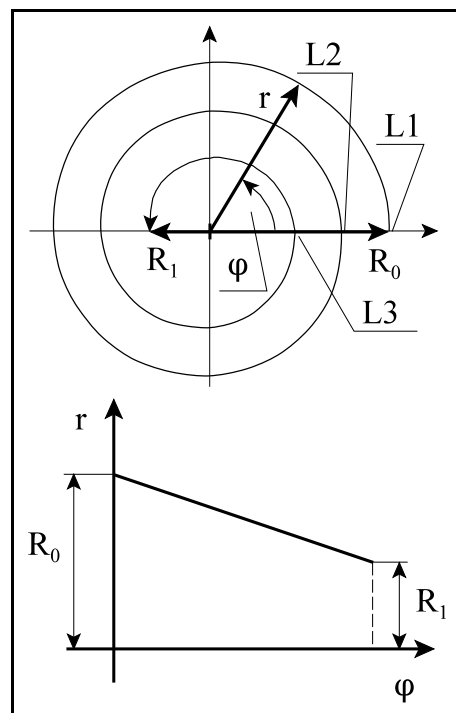


рисунок 4.3.1-1

Пример:

Программируем винт на приложенном рисунке.

Начальная точка винта X180 Y0, точка его центра, измеряя от начальной точки I-90, J0, изменение радиуса за оборот 24, и совершает 2,5 оборота. Итак радиус конечной точки винта:

$$R=90-2*24-24/2=30$$

Число начатых оборотов винта 3.

Программа:

```
G17 G90 G94 M3 S1000
G0 X180 Y0 F1000
G3 X-60 I-90 L3
```

В примере выше координаты *X* и *Y* заданы **в диаметре**.

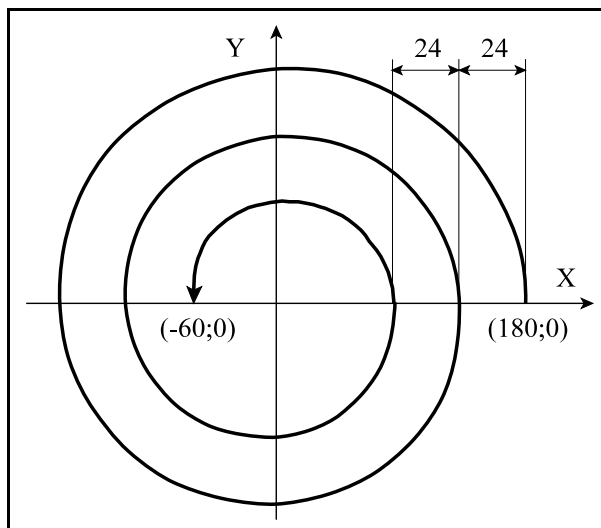


рисунок 4.3.1-2

Если желаем, что по заданному примеру во главе 4.3 Круговая интерполяция (30 страница) устройство ЧПУ не выводило сообщение 2012 Ошибка разницы радиуса в окружности, следует изменить программу так:

```
G18 G90 G0 X0 Z50
G3 Z-20 K-50 L1
```

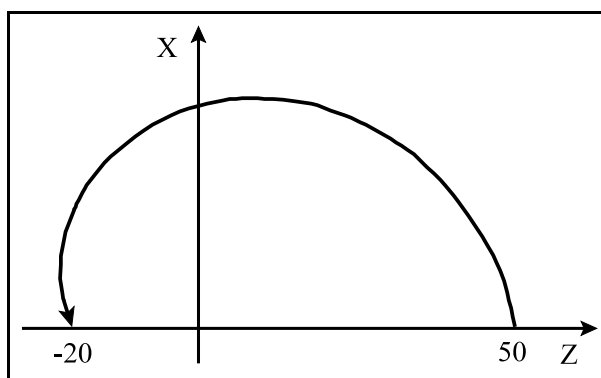


рисунок 4.3.1-3

4.3.2 Цилиндрическая винтовая интерполяция (G2, G3)

Командой G2, или G3 можно запрограммировать цилиндрическую интерполяцию так, что **на оси, перпендикулярной плоскости окружности запрограммировать движение. По адресу L** можно задавать **число оборотов винта**. Перемещение выполняется запрограммированной в кадре, или модальной подачей F.

Формат кадра:

$$G17 \left\{ \begin{matrix} G2 \\ G3 \end{matrix} \right\} X_P - Y_P - \left\{ \begin{matrix} R \\ I - J \end{matrix} \right\} Z_P - q \dots L - F -$$

$$G18 \left\{ \begin{matrix} G2 \\ G3 \end{matrix} \right\} X_P - Z_P - \left\{ \begin{matrix} R_- \\ I - K \end{matrix} \right\} Y_P - q \dots L - F -$$

$$G19 \left\{ \begin{matrix} G2 \\ G3 \end{matrix} \right\} Y_P - Z_P - \left\{ \begin{matrix} R_- \\ J - K \end{matrix} \right\} X_P - q \dots L - F -$$

Задание окружности происходит согласно правилам, заданным у круговой интерполяции. Смещение по оси, перпендикулярной плоскости окружности пропорционально смещению вдоль дуги окружности.

К оси, перпендикулярной окружности, можно задавать перемещение для произвольного числа программируемой в канале оси, указанной в формуле выше с буквой q....

По адресу L можно задавать **число оборотов цилиндрического винта**. Каждый заход означает один новый оборот, даже тогда, если после него не последует полный оборот. Адрес **L** представляет собой **положительное целое число**.

В ходе цилиндрической винтовой интерполяции устройство ЧПУ изменяет подъём винтовой линии линейно в зависимости от угла поворота (ϕ) таким образом, чтобы пока совершается оборот, заданный по адресу L и достигается позиция конечной точки, на осях, находящихся вне плоскости окружности, позиция конечной точки соответствовала запрограммированным данным.

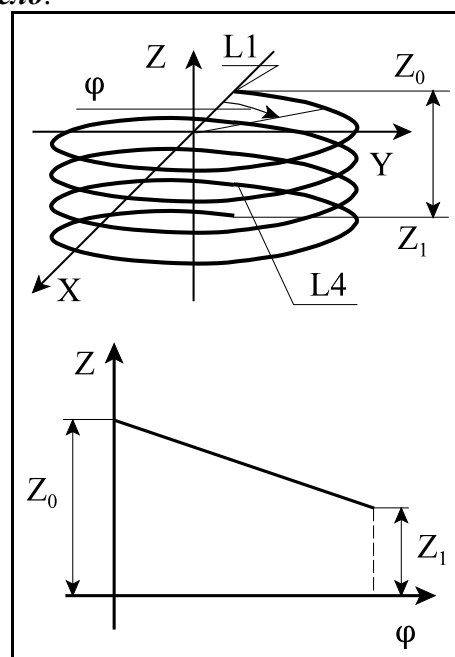


рисунок 4.3.2-1

Примеры:

Цилиндрическую винтовую интерполяцию, показанную на приложенном рисунке, можно задавать следующим образом:

```
G17 G90 G0 X100 Y0 Z0
G03 X0 Y100 Z20 R50 F150
```

В примере выше координаты X и Y заданы *в диаметре*.

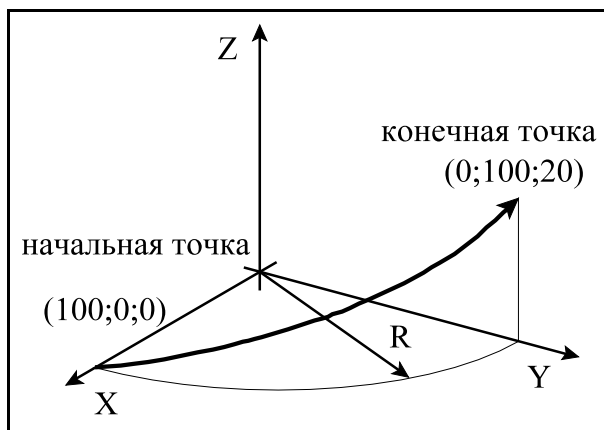


рисунок 4.3.2-2

На боковую поверхность косоугольного цилиндра, показанную на приложенном рисунке, фрезеруется дуга окружности. Ось V параллельна оси Y , которую перемещает устройство ЧПУ вместе с осью Z :

```
G17 G90 G0 X100 Y0 Z0 V0
G03 X0 Y-100 Z50 V20 I-50
```

В примере выше координаты X и Y заданы *в диаметре*.

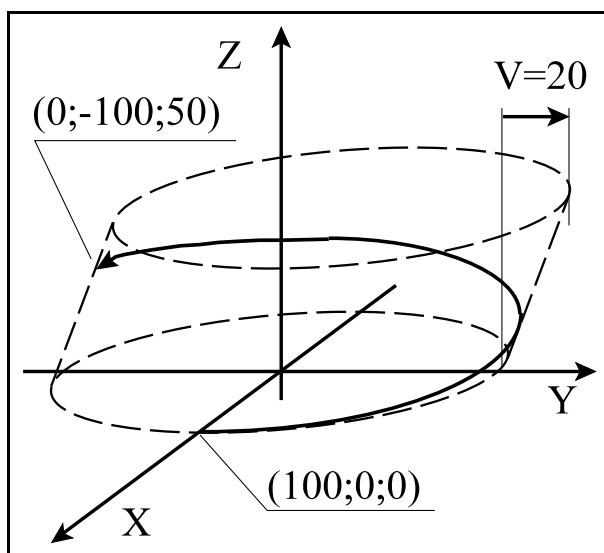


рисунок 4.3.2-3

Запрограммируется цилиндрический винт с 4 оборотами, показанный на приложенном рисунке.

Начальная точка винта $X-100 Y0, Z0$, точка центра измеряя от начальной точки $I-50, J0$, подъем винтовой линии за оборот 5, и совершается 4 целых оборотов.

Программа:

```
G54 G17 G90 ...
G0 X-100 Y0
Z0
G2 I50 Z-20 L4 F100
...
```

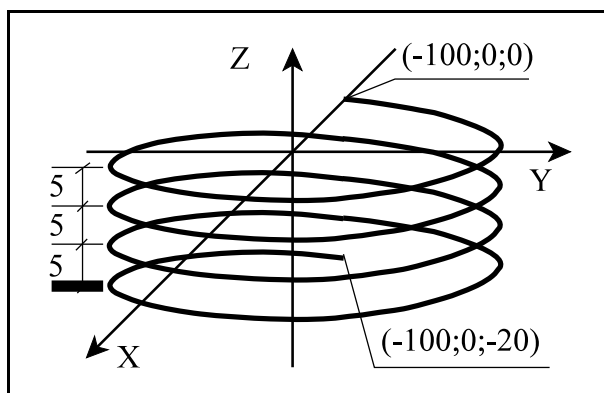


рисунок 4.3.2-4

В примере выше координаты X и Y заданы *в диаметре*.

Заданная по адресу F подача в основном случае достигается **вдоль круговой траектории** при положении бита N1337 Execution Config #3 $HEF=0$. При этом подача по осям получается согласно следующим формулам:

$$F_{iv} = F$$

$$F_q = \frac{L_q}{L_{iv}} F$$

При положении бита $HEF=1$ подача рассчитывается **вдоль круговой траектории**. При этом подача по осям получается согласно следующим формулам:

$$F_{iv} = \frac{L_{iv}}{\sqrt{L_{iv}^2 + L_q^2}} F$$

$$F_q = \frac{L_q}{\sqrt{L_{iv}^2 + L_q^2}} F$$

где L_q : перемещение вдоль оси q,
 L_{iv} : длина дуги окружности,
 F : запрограммированная подача,
 F_q : подача вдоль оси q
 F_{iv} : смещение вдоль дуги окружности

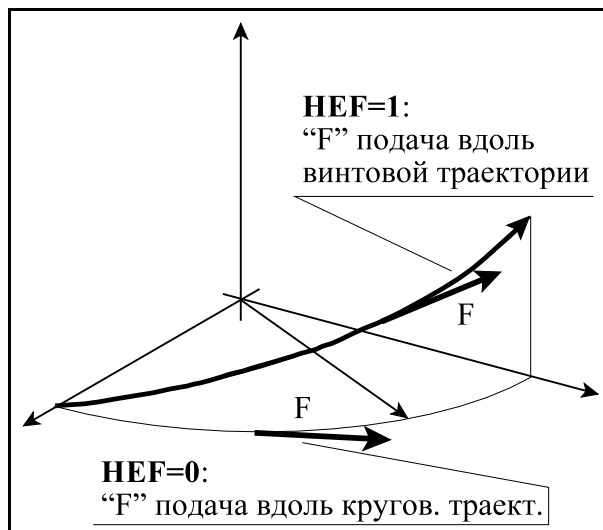


рисунок 4.3.2-5

Заданная **коррекция радиуса инструмента** действительна всегда в плоскости **окружности** вдоль дуги окружности.

В том случае, если **радиус окружности**, заданной в выбранной плоскости **переменный**, интерполяция выполняется вдоль боковой поверхности заданного конуса.

4.3.3 Коническая винтовая интерполяция (G2, G3)

Командой G2, или G3 можно запрограммировать коническую винтовую интерполяцию так, что *запрограммировать движение по оси, перпендикулярной к плоскости окружности. Точка центра и конечная точка окружности* задаётся так, чтобы *радиус начальной точки и конечной точки окружности отличались друг от друга. По адресу L* можно задавать *число оборотов конического винта*. Перемещение выполняется запрограммированной в кадре, или модальной подачей F.

Формат кадра:

$$G17 \left\{ \begin{matrix} G2 \\ G3 \end{matrix} \right\} X_P_Y_P_I_J_Z_P_q...L_F_$$

$$G18 \left\{ \begin{matrix} G2 \\ G3 \end{matrix} \right\} X_P_Z_P_I_K_Y_P_q...L_F_$$

$$G19 \left\{ \begin{matrix} G2 \\ G3 \end{matrix} \right\} Y_P_Z_P_J_K_X_P_q...L_F_$$

Значение X_P , Y_P , Z_P как *координата конечной точки винта* задаётся в данной системе координат абсолютными данными, или измеренными от начальной точки инкрементными данными.

К оси, перпендикулярной окружности, можно задавать перемещение для произвольного числа программируемой в канале оси, указанной в формуле выше с буквой q....

По адресу I , J , K задаются *координаты точки центра* конического винта в выбранной плоскости, *измеренные* от начальной точки *инкрементными* данными так, чтобы определённые значениями I , J , K вектор показывал *от начальной точки в точку центра* винта.

Значения I , J , K надо задавать всегда в *радиусе*, даже тогда, если принадлежащие к ним оси установлены для программирования по диаметру.

По адресу L задаётся *число оборотов винта*. Каждый заход означает один новый оборот, даже тогда, если после ним не последует полный оборот. Адрес L *положительное целое число*.

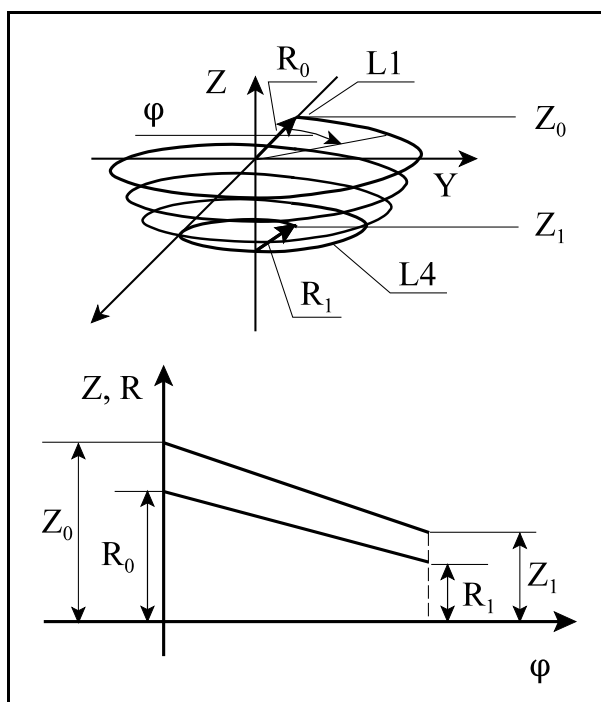


рисунок 4.3.3-1

В ходе конической винтовой интерполяции подъём винтовой линии устройство ЧПУ изменяет в зависимости от угла поворота (ϕ) линейно таким образом, чтобы пока совершается заданный по адресу L оборот и достигается позиция конечной точки, на осях, лежащих вне плоскости окружности, позиция конечной точки соответствовала запрограммированным данным. Вместе с этим устройство ЧПУ линейно изменяет и радиус окружности в зависимости от угла поворота.

Пример:

Запрограммировать конический винт с 4 оборотами, радиус начальной точки которого 50, радиус конечной точки 20 и подъём винтовой линии за оборот 5.

Программа:

```
G17 G90 ...
G0 X-100 Y0
Z0
G2 X-40 I50 Z-20 L4 F100
...
```

В примере выше координаты X и Y заданы в диаметре.

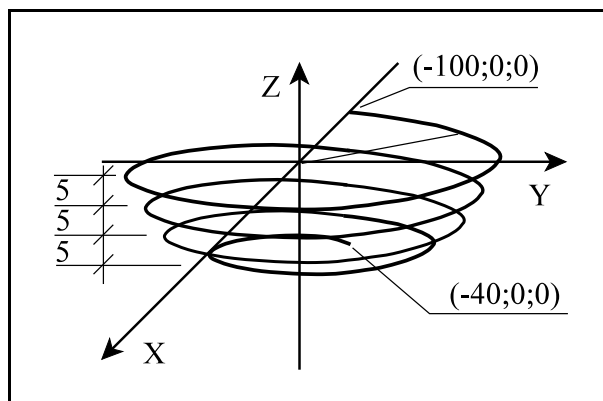


рисунок 4.3.3-2

4.4 Нарезание резьбы с равномерным шагом (G33)

Командой G33 можно запрограммировать нарезание цилиндрической, или конической резьбы с равномерным шагом. Формат кадра:

G33 v F Q

или

G33 v E Q

Для вектора “v” можно записать данные координат не более двух осей. Если для вектора v указаны данные двух координат, устройство ЧПУ нарезает коническую резьбу. Устройство ЧПУ принимает во внимание шаг резьбы для той оси, у которой получается большее перемещение.

Если $\alpha < 45^\circ$, то есть $Z > X$, запрограммированный шаг резьбы принимается во внимание *вдоль оси Z*,

если $\alpha > 45^\circ$, то есть $X > Z$, запрограммированный шаг резьбы принимается во внимание *вдоль оси X*.

Шаг резьбы можно определить двояко:

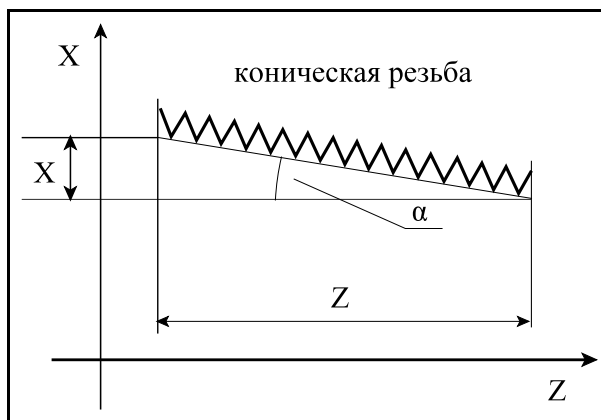


рисунок 4.4-1

Задание шага резьбы по адресу F

Если шаг резьбы задаётся *по адресу F*, тогда данные истолкуются *мм/об*, или *дюйм/об*. Значит, если желаем нарезать резьбу с шагом 2.5 мм, надо запрограммировать F2.5.

Задание шага резьбы по адресу E

Если шаг резьбы задаётся по адресу E, устройство ЧПУ нарезает дюймовую резьбу. Истолкование *адреса E* - это *число витков за дюйм*. Если например, запрограммировать E8, тогда устройство ЧПУ нарезает резьбу с шагом $1/8" = 25.4/8 = 3.175$ мм.

По адресу Q задаётся то значение угла, что *на сколько градусов поворота должен совершить* шпиндель до начала нарезание резьбы, *отсчитывая от нулевого импульса датчика шпинделя*. **Многозаходная резьба** нарезается соответствующим программированием значения *Q*, то есть здесь можно запрограммировать, что при какого угла поворота шпинделя должно начинать устройство ЧПУ разные зарезы. Например, при желании нарезать двузаходную резьбу, первый зарез надо запускать из Q0 (отдельно задавать не надо), а второй зарез - из Q180.

G33 представляет собой модалную функцию. Если запрограммировать один за другой несколько кадров нарезания резьбы, можно нарезать резьбу на произвольную поверхность, ограниченную прямыми



рисунок 4.4-2

участками.

Устройство ЧПУ осуществляет синхронизацию в первом кадре по нулевому импульсу датчика шпинделя, и у последующих кадрах уже не выполняет синхронизацию, следовательно подъём винтовой линии выполняется плавно на всех участках. Исходя из этого и запро-

граммированный поворот угла Q шпинделя принимается во внимание только в первом кадре.

Пример:

Нарезать резьбу с шагом 2 мм:

```

...
G0 G90 X50 Z60
X20
G33 X30 Z12 F2
G0 X50
Z60
...

```

В примере принимается во внимание для программирования диаметр.

☞ **Замечания:**

- Если в кадре нарезания резьбы заполнен адрес и F и E, задаём F0, или E0 устройство ЧПУ выдаёт сообщение об ошибке *2021 Ошибка задания нарезания резьбы.*
- В ходе выполнения команды G33 override значение подачи и шпинделя устройством ЧПУ автоматически принимается за 100% и клавиша Stop стой не имеет действия.
- Из-за ошибки слежения сервосистемы в начале и в конце резьбы необходимо оставлять расстояние инструменту для набега и сбеге вне материала, чтобы подъём нарезки был постоянным по всему участку.
- Перед выдачей команды нарезания резьбы надо выключить состояние расчёта постоянной скорости резания (G96).

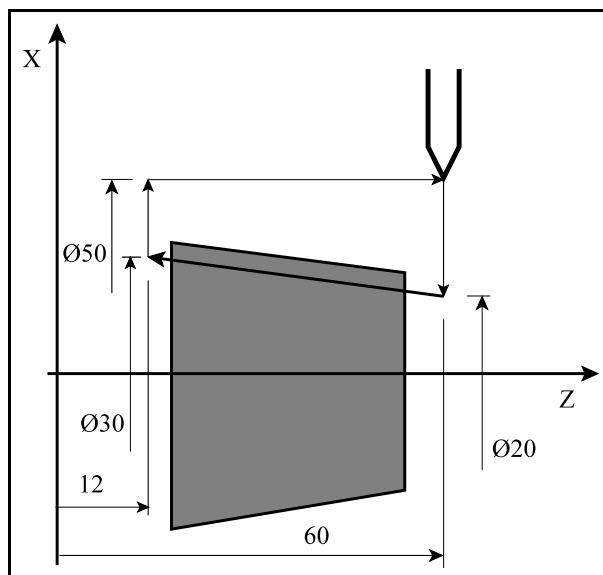


рисунок 4.4-3

4.5 Интерполяция полярных координат (G12.1, G13.1)

Интерполяция полярных координат представляет собой такой режим работы устройства ЧПУ, в котором траекторию контура заготовки, описанную в прямоугольной системе координат (Декартова) проходит перемещением одной **линейной** и одной **вращающейся** оси, то есть в ходе перемещения с минуты на минуту пересчитывает траекторию, заданную прямоугольными координатами для траектории, заданной данными полярных координат.

Команда

G12.1 интерполяция полярных координат вкл.

включает режим полярных координат. В последующем за этим фрагменте программы траекторию фрезы можно описывать в

прямоугольной системе координат с программированием одной **линейной** и одной **вращающейся** оси (**виртуальной линейной** оси). Команда выдаётся всегда в отдельном кадре, и рядом с ним нельзя программировать другую команду.

Команда

G13.1 интерполяции полярных координат выкл.

выключает режим полярных координат. Команда выдаётся всегда в отдельном кадре, и рядом с ним нельзя программировать другую команду. Устройство ЧПУ после включения, или reset примет всегда состояние G13.1.

Выбор линейной и вращающейся оси

Перед включением интерполяции полярных координат надо выбрать одну линейную и одну вращающуюся (виртуальную) оси, которые участвуют в интерполяции полярных координат.

Выбор осей выполняется командой выбора плоскости G17, G18, G19.

G17 X_p_ адрес одной вращающейся (виртуальной) оси_

G18 Z_p_ адрес одной вращающейся (виртуальной) оси_

G19 Y_p_ адрес одной вращающейся (виртуальной) оси_

Первая ось выбранной плоскости будет всегда **линейной осью**. Можно выбирать и параллельную ось.

Вращающаяся ось, заданная в команде выбора плоскости будет вращающейся осью интерполяции полярных координат. После этого команда G12.1 ведёт расчёт данными, заданными адресами линейной и вращающейся осей, заданных таким образом.

Например, команда

G17 X_ C_

назначает ось X линейной, а C вращающейся осью.

А команда

G19 Y_ C_

назначает ось Y линейной, а C вращающейся осью.

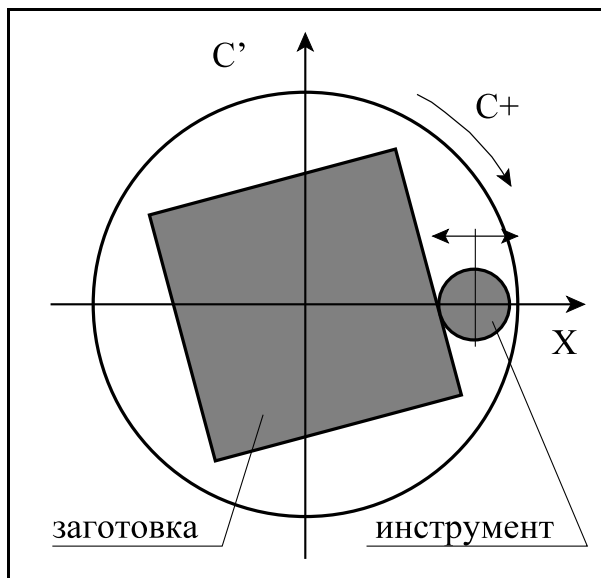


рисунок 4.5-1

Положение нулевой точки заготовки в ходе интерполяции полярных координат
 Перед включением интерполяции полярных координат надо выбирать такую систему координат заготовки, в которой **точка центра вращения вращающейся оси совпадает с началом координат линейной оси** полярной интерполяции.

Например, если С вращающаяся ось а Х а линейная ось, нулевую точку системы координат так надо выбирать по оси Х, чтобы позиция инструмента $X=0$ совпадала осью вращения (С) круговой оси.

Если точка центра вращения вращающейся оси не совпадает первой осью (линейной осью) выбранной плоскости, другими словами точка центра вращения вращающейся оси не совпадает осью вращения инструмента в направлении, перпендикулярной к линейной оси, обрабатываемая заготовка искажается. Мера искажения будет тем больше, чем ближе подойдёт инструмент к началу координат.

Поскольку в этом направлении совсем нет осей, поэтому это отклонение нельзя компенсировать со смещением нулевой точки.

Параметром N0217 Polar Intp. Comp. Amount можно компенсировать это отклонение. Значение компенсации надо написать **всегда по адресу вращающейся оси** в мм-ах или в дюймах.

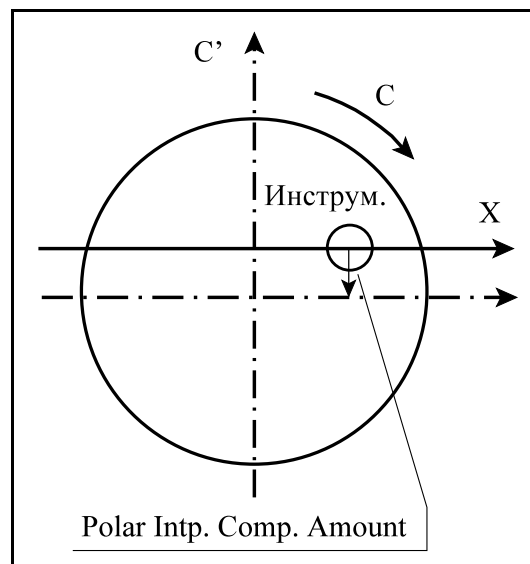


рисунок 4.5-2

Положение осей в момент включения интерполяции полярных координат
 Перед включением интерполяции полярных координат (команда G12.1) необходимо позаботиться о том, чтобы **круговой ось находилась в точке с позицией 0. Позиция линейной оси** может быть и **отрицательной** и **положительной**, но **не может быть 0**.

Программирование данных по длине в ходе интерполяции полярных координат
 Во включенном состоянии интерполяции полярных координат программируем данные по длине на обеих осях, принадлежащих к выбранной плоскости: вращающаяся ось, лежащая в выбранной плоскости будет второй (виртуальной) осью. Если например, командой G17 X_ C_ выбрали ось X, C, адрес C можно программировать так, как в случае выбора плоскости G17 X_ Y_ поступили с осью Y.

В интерполяции полярных координат можно применить и функцию G16 программирования в полярных координатах. При этом, по смыслу, по адресу первой оси выбранной плоскости задаётся полярных радиус, а на вращающейся оси - полярный угол.

В основном случае на программирование виртуальной оси не оказывает влияние, что программирование первой оси выполнена ли в диаметре, для виртуальной оси надо задавать данные координат всегда в радиусе. Если например, интерполяция полярных координат происходит в плоскости X C, значение, написанное по адресу C надо задавать в радиусе, независимо от того, что адрес X задаётся в диаметре, или в радиусе.

Перемещение осей, не участвующих в интерполяции полярных координат
 Инструмент перемещается по этим осям так, как в нормальном случае, независимо от включенного состояния интерполяции полярных координат.

Программирование круговой интерполяции в ходе интерполяции полярных координат

Во включенном состоянии интерполяции задание окружности возможно уже известным образом, радиусом, или программированием координат точки центра окружности. Если выбрать это последнее, адреса I, J, K, согласно выбранной плоскости, надо использовать следующим образом:

G17 X_p_ адрес одной вращающейся (виртуальной) оси _ I _ J _

G18 Z_p_ адрес одной вращающейся (виртуальной) оси _ I _ K _

G19 Y_p_ адрес одной вращающейся (виртуальной) оси _ J _ K _

Использование коррекции радиуса инструмента в случае интерполяции полярных координат

Команду G41, G42 можно использовать обычным образом во включенном состоянии интерполяции полярных координат. Следите за тем, чтобы в группе коррекции вращающегося инструмента код позиции инструмента должен быть Q=0. К его использованию относятся следующие ограничения:

Включение интерполяция полярных координат (команда G12.1) возможно только в состоянии G40,

Если в состоянии G12.1 включить G41, или G42, перед выключением интерполяция полярных координат (команда G13.1) надо программировать G40.

Ограничения программирования в ходе интерполяции полярных координат

А Во включенном состоянии интерполяции нельзя пользоваться следующими командами:

- смена плоскости: G17, G18, G19,
- трансформации координат: G52, G92,
- смена системы координат заготовки: G54, ..., G59,
- позиционирование в станочной системы координат : G53
- приём референтной точки G28, G30 Pp: 2. 3. 4. Набег на референтную точку,
- измерение с удалением остаточного хода G31.

Подача в ходе интерполяции полярных координат

Истолкование подачи во включенном состоянии интерполяции полярных координат происходит привычным при прямоугольной интерполяции образом, в качестве скорости вдоль траектории: она укажет относительную скорость заготовки и инструмента.

В ходе интерполяции полярных координат заданную в системе прямоугольной системы координат траектория исполняется перемещением одной линейной и одной вращающейся осями. Как точка центра инструмента приближается к оси вращения круговой координаты, так вращающаяся ось должна бы всё больше и больше шага совершить за единицу времени для того, чтобы скорость вдоль траектории осталась постоянной. Однако, скорость круговой оси ограничивается максимальной скоростью, допускаемой для вращающейся оси, и определённой параметром. Поэтому вблизи начала координат устройство ЧПУ постепенно уменьшает подачу вдоль траектории, чтобы скорость вращающейся оси не увеличивалась беспрдельно.

Приложенный рисунок показывает тот случай, когда запрограммируем прямые (1, 2, 3, 4), параллельны оси X. К запрограммированной подаче принадлежит перемещение Δx за единицу времени. К перемещению Δx в случае разных прямых (1, 2, 3, 4) принадлежат разные угла поворота ($\varphi_1, \varphi_2, \varphi_3, \varphi_4$). Видно, что чем ближе проходит обработка к началу координат, тем больше угла поворота должна совершить вращающаяся ось за единицу времени, чтобы сумела придерживаться к запрограммированной подаче.

Если в таком случае угловая скорость превосходит значение, установленное для вращающейся оси параметром N0305 Max Feed, устройство ЧПУ постепенно уменьшает подачу вдоль траектории.

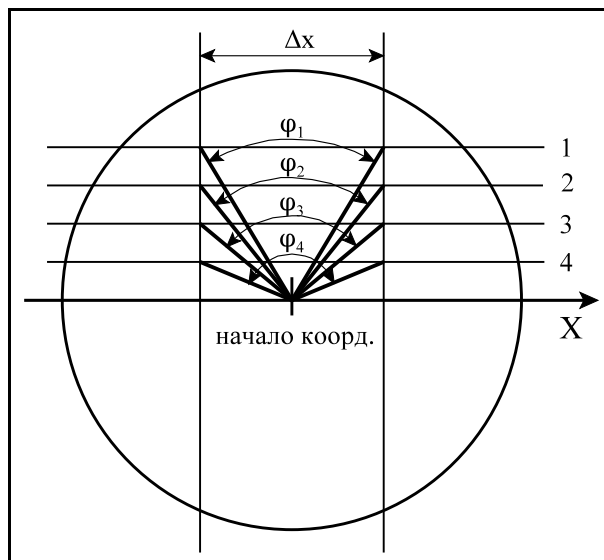


рисунок 4.5-3

Образцовый пример

В следующем описываем показательный пример к использованию интерполяции полярных координат. Участвующие в интерполяции оси: X (линейная ось) и C (вращающаяся ось). Программирование **оси X** выполняется в **диаметре**, а **оси C** в **радиусе**.

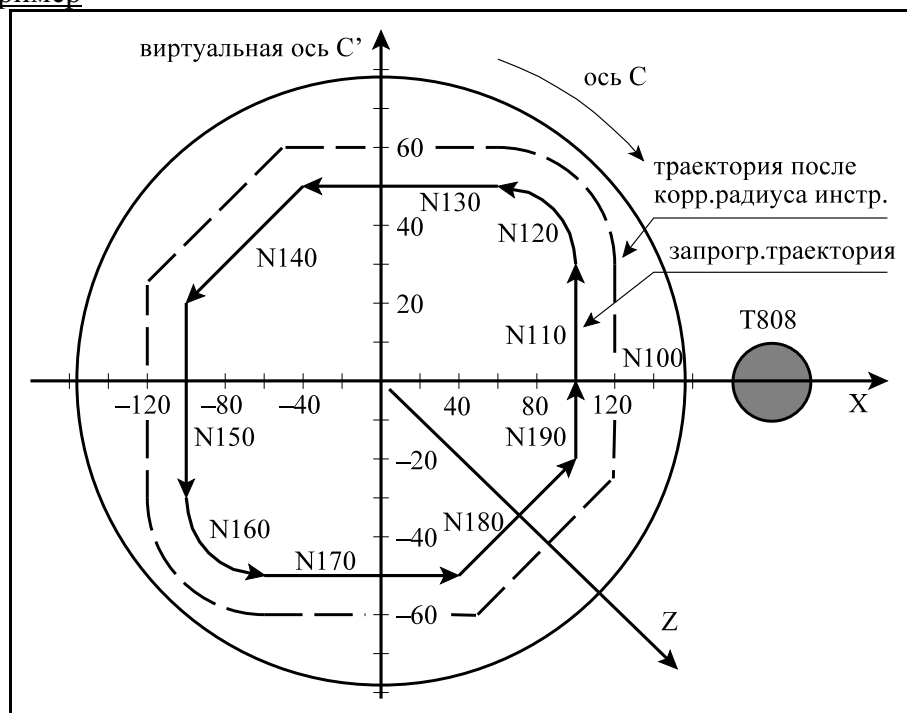


рисунок 4.5-4

...

N050 T808

N060 G59

N070 G17 G0 X200 C0

N080 G94 Z-3 S1000 M3

N090 G12.1

(Точка начала системы координат G59 в направлении X есть ось вращения для C)
(Выбор плоскости X, C; позиционирование $X \neq 0$, на координату $C=0$)

(интерполяция полярных координат вкл.)

4.5 Интерполяция полярных координат (G12.1, G13.1)

```
N100 G42 G1 X100 F1000
N110 C30
N120 G3 X60 C50 I-20 J0
N130 G1 X-40
N140 X-100 C20
N150 C-30
N160 G3 X-60 C-50 R20
N170 G1 X40
N180 X100 C-20
N190 C0
N200 G40 G0 X150
N210 G13.1 (интерполяция полярных координат выкл.)
N220 G0 Z100 (отвод инструмента )
... .
```

4.6 Цилиндрическая интерполяция (G7.1)

Если нужно профрезеровать в боковую поверхность цилиндра траекторию лекала, применяем цилиндрическую интерполяцию. При этом должны совпадать осевая линия цилиндра и с осью вращения вращающейся оси.

Программу описываем как программирование *линейной оси, параллельной осевой линии цилиндра* и *круговой оси, вращающий цилиндр*. В программе перемещение *вращающейся оси* задаётся в единицах угла, в *градусах*, который устройство ЧПУ пересчитывает вдоль боковой поверхности в зависимости радиуса цилиндра в длину дуги, в мм, или дюйм. Перемещение, получаемое после интерполяции преобразует назад в поворот угла для вращающейся оси.

Подачу F , заданную во время цилиндрической интерполяции принимается во внимание всегда *вдоль боковой поверхности цилиндра*.

Команда

G7.1 Qr цилиндрическая интерполяция
 вкл.

включает цилиндрическую интерполяцию, где

Q: адрес вращающейся осей, участвующей в цилиндрической интерполяции
r: радиус цилиндра.

Если например, ось “С” является вращающейся осью, участвующей в цилиндрической интерполяции, и радиус цилиндра 50 мм, тогда цилиндрическую интерполяцию можно включить командой G7.1 C50.

В фрагменте программы, последующей после этого можно описать траекторию, профрезеруемую в боковую поверхность цилиндра заданием линейной и круговой интерполяции. На продольной оси координата задаётся всегда в мм-ах, или в дюймах, а на вращающейся оси - всегда в °-ах.

Команда

G7.1 Q0 цилиндрическая интерполяция выкл.

выключает цилиндрическую интерполяцию, то есть код G тот же самый, как при включении, только надо записать в адрес вращающейся оси 0.

Цилиндрическую интерполяцию (G7.1 C50) включенную согласно вышеуказанному примеру можно выключить командой G7.1 C0.

Устройство ЧПУ после включения, конец программы или reset всегда примет состояние цилиндрическая интерполяция выкл.

Команду G7.1 надо задавать в отдельном кадре.

Выбор линейной и вращающейся оси

Перед включением цилиндрической интерполяции нужно выбирать одну линейную и одну вращающуюся ось, которые представляют собой оси, участвующие в цилиндрической интерполяции.

Выбор осей выполняется командой выбора плоскости G17, G18, G19.

G17 X_p_, или Y_p_ и адрес одной вращающейся оси_

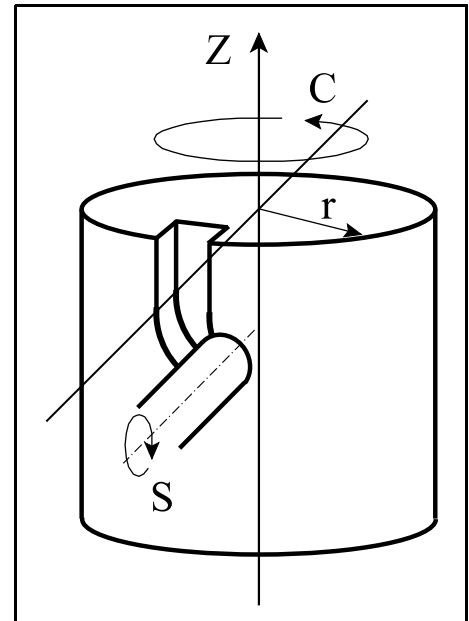


рисунок 4.6-1

G18 X_{r_}, или Z_{r_} и адрес одной вращающейся оси _

G19 Y_{r_}, или Z_{r_} и адрес одной вращающейся оси _

Цилиндрическая интерполяция истолкует направления круговой интерполяции G2, G3 и направление коррекции радиуса инструмента (G41, G42) на основании выбранной плоскости и принадлежащей к ней линейной оси. Круговая ось будет второй осью плоскости. Можно выбирать и параллельную ось.

Например, пусть будет ось Z параллельна осевой линии цилиндра и пусть будет ось C осью вращения цилиндра. При этом перед включением цилиндрической интерполяции или командой

G18 Z_ C_

или командой

G19 Z_ C_

можно выделить линейную и вращающуюся ось.

Круговая интерполяция

В режиме цилиндрической интерполяции возможно задание круговой интерполяции, однако только заданием радиуса R.

В случае цилиндрической интерполяции нельзя выполнить круговую интерполяцию заданием точки центра окружности (I, J, K).

Радиус окружности истолкуется всегда в мм-ах, или в дюймах, никогда не в градусах.

Круговую интерполяцию, например между осями Z и C можно задавать двояко:

G18 Z_ C_

G19 C_ Z_

G2 Z_ C_ R_

G3 C_ Z_ R_

G3 Z_ C_ R_

G2 C_ Z_ R_

При желании описать ту же траекторию, выбором плоскости G18 и G19 направления окружностей можно поменять друг с другом.

Использование коррекции инструмента в случае цилиндрической интерполяции

Команду G41, G42 можно использовать привычным образом во включенном состоянии цилиндрической интерполяции. К её использованию относятся следующие ограничения:

- Включение цилиндрической интерполяции возможно (команда G7.1 Qr) только в состоянии G40,
- Если в состоянии цилиндрической интерполяции включили G41, или G42, перед выключением цилиндрической интерполяции (команда G7.1 Q0) надо программировать G40.

Ограничения программирования в ходе цилиндрической интерполяции

Во включенном состоянии цилиндрической интерполяции нельзя использовать следующие команды:

- выбор плоскости: G17, G18, G19,
- трансформации координат: G52, G92,
- смена системы координат заготовки: G54, ..., G59,
- позиционирование в станойной системе координат: G53,
- круговая интерполяция заданием точки центра окружности (I, J, K),
- циклы сверления.

Показательный пример

Фрезеровать на боковую поверхность цилиндра с радиусом $R=28.65$ мм траекторию с глубиной 3 мм, показанную на приложенном рисунке. Фреза Т параллельна оси Х.

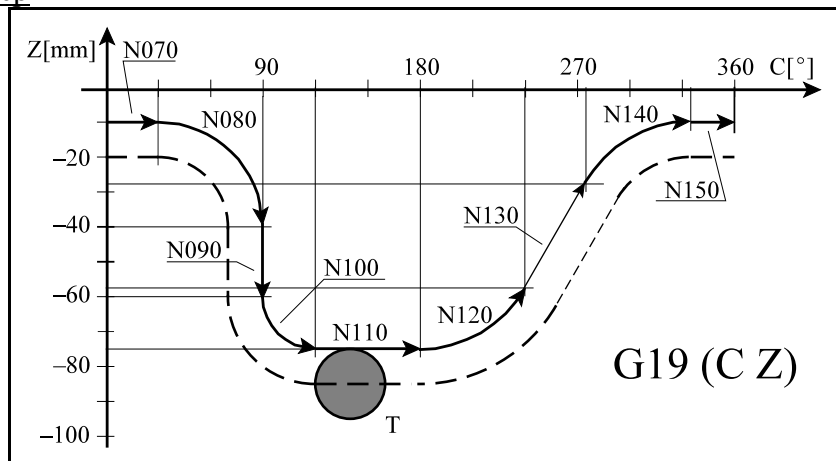


рисунок 4.6-2

Перемещение, выходящее на один градус (1°) боковой поверхности цилиндра:

$$28.65\text{mm} \cdot \frac{1^\circ}{180^\circ} \cdot \pi = 0.5\text{mm}$$

Расположение осей, показанное на рисунке соответствует выбору плоскости G19.

(ЦИЛИНДРИЧЕСКАЯ ИНТЕРПОЛЯЦИЯ)

...

N030 G19 Z-20 C0

N040 G1 X51.3 F100

N050 G7.1 C28.65

(G19: выбор плоскости C-Z)

(глубина врезания 3 мм)

(включение цилиндрической интерполяции, вращающаяся ось: C, радиус цилиндра 28.65 мм)

N060 G1 G42 Z-10 F250

N070 C30

N080 G2 Z-40 C90 R30

N090 G1 Z-60

N100 G3 Z-75 C120 R15

N110 G1 C180

N120 G3 Z-57.5 C240 R35

N130 G1 Z-27.5 C275

N140 G2 Z-10 C335 R35

N150 G1 C360

N160 G40 Z-20

N170 G7.1 C0

(выключение цилиндрической интерполяции)

N180 G0 X100

...

5 Данные координат

5.1 Абсолютное и инкрементное программирование (G90, G91), оператор I, адрес U, V, W

Входные данные координат можно задавать как абсолютные значения и как значения с приращением. При абсолютном задании данных необходимо задавать устройству ЧПУ координаты конечной точки, а при данных с приращением - расстояние, совершаемое в кадре.

G90: Программирование с абсолютным заданием данных

G91: Программирование с заданием данных с приращением

Функции G90, G91 являются модалными. При включении устройство ЧПУ решает на основании бита #7 G91 параметра N1300 DefaultG1, какое состояние принять. В конце программы, или под действием reset действовать будет также код, установленный параметром.

Перемещение в абсолютную позицию возможно только после приёма референтной точки.

Пример:

На основании рисунка можно запрограммировать движение двояко.

```
G90 G01 X100 Z20
G91 G01 X60 Z-40
```

Оператор I действует в состоянии абсолютного задания данных G90. *Он относится только к той координате, после адреса которой стоит.* Значение: модалные данные.

Приведённый выше пример можно решать и следующим образом:

```
G90 G01 XI60 ZI-40
G01 XI60 Z20
G01 X100 ZI-40
```

В случае использования *многосимвольного названия оси*, если название *заканчивается на число*, например, в случае оси X2, оператор I надо ставить после знака = :

```
X2=I100
```

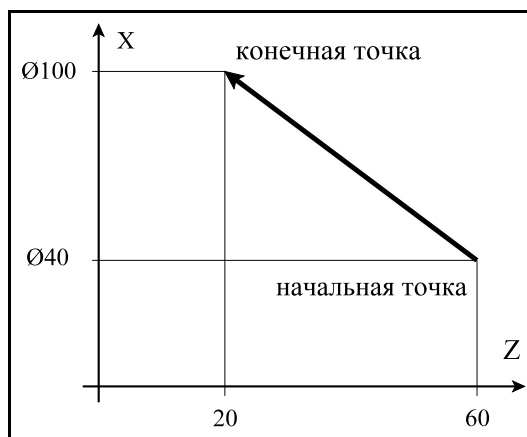


рисунок 5.1-1

Если не назначены осями **адреса U, V, W**, тогда можно их использовать для обозначения инкрементных движений в направлении X, Y, Z:

	адрес абсолютной команды	адрес инкрементной команды
команда движения в направлении X	X	U
команда движения в направлении Y	Y	V
команда движения в направлении Z	Z	W

Принимая во внимание вышеописанных, показательный пример:

```
G90 G01 U60 W-40
G01 U60 Z20
G01 X100 W-40
```

5.2 Преобразование дюймы/метрические единицы (G20, G21)

Входящие данные можно задавать хоть в метрической, хоть в дюймовой системе мер, программированием соответствующего кода G.

G20: Выбор дюймовой системы мер.

G21: Выбор метрической системы мер.

Выбранная система мер остаётся в силу до выдачи команды с противоположным значением, значит, коды G20, G21 модальные.

При включении на основании бита #3 G20 параметра N1300 DefaultG1 решает устройство ЧПУ, какое состояние принять им. В конце программы, или под действием reset действует код, установленный также параметром.

Например:

```
G21 G0 G54 X200 Z50 (позиционирование на X=200, Z=50 мм)
G20 X2 Z1 (позиционирование на X=2, Z=1 дюйм)
```

Изменение системы мер влияет на следующие позиции:

- Данные координат и коррекций, (мм/дюйм)
- Подача (мм/мин, дюйм/мин, мм/об, дюйм/об),
- Постоянная скорость резания (м/мин, feet/мин),
- Значения позиции, коррекции, смещения нулевой точки и подачи выводятся на индикатор всегда в выбранной системе мер,
- При чтении макропеременных (смещение, данные позиции и т. д.) данные прочитываются всегда в выбранной системе мер,
- Величина шага инкрементного jog и маховичка,
- Подача ручного перемещения (jog).

5.3 Программирование по диаметру, или по радиусу

Поскольку сечение заготовок, обрабатываемых на токарных станках, обычно круглое, измерением диаметра заготовки можно проверить, размеры в направлении X целесообразно задавать диаметром. Если на станке имеется и ось Y, обычно целесообразно и размером в направлении Y работать в диаметре. Для того, чтобы устройство ЧПУ истолковало размер в данной оси в диаметре, или в радиусе, можно это задавать в бите #0 DIA параметра N0106 Axis Properties, по осям:

в случае **программирования в радиусе**:
#0 DIA=0

в случае **программирования в диаметре**:
#0 DIA=1

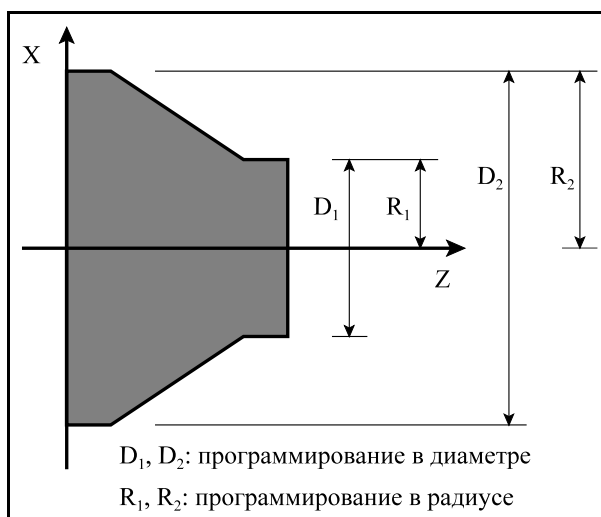


рисунок 5.3-1

Если параметр установлен для программирования диаметра, следует учитывать следующие:

случай	замечание
команда абсолютного движения	задаётся в диаметре
команда инкрементного движения	задаётся в диаметре (на рисунке $D_2 - D_1$)
смещение координат и нулевой точки	задаётся в диаметре
коррекция по длине инструмента	задаётся в диаметре
Параметры, относящиеся к осям, работающим по диаметру в цикле, как например глубина врезания	задаётся всегда в радиусе
при задании круговой интерполяции значения R и I, J, K	задаётся всегда в радиусе
отображение позиции оси	задаётся в диаметре
подача в направлении оси, истолкованной по диаметру	всегда радиус/об, или радиус/мин
величина шага в режиме прокрутки и маховичка	в случае выбора инкремента 1 совершается шаг 1 μm в диаметре

5.3.1 Переключение программирования по радиусу/по диаметру (G10.9)

Способ программирования на отдельных осях в радиусе, или в диаметре определяется положением параметра параметр N0106 Axis Properties при #0 DIA

Если значение бита DIA:

=0: в радиусе,

=1: в диаметре

выполняется программирование оси.

В программе деталей в произвольном месте можно переключить способ программирования. То, что переключение происходит из PLC, или кодом G, определяется битом #5 MGD параметра N0106 Axis Properties

Если значение бита

=0: переключение программирования по радиусу/по диаметру выполняется через индикатор PLC,

=1: переключение программирования по радиусу/по диаметру выполняется кодом G10.9.

Способ переключения из PLC определяется строителем станка.

В дальнейшем описываем переключение кодом G10.9.

Команда

G10.9 v

переключает программирование по радиусу, или по диаметру, где

v: адрес произвольных линейных осей: X, Y, ... и т. д.

Если значение, написанное в адрес оси:

=0: программирование оси по радиусу, если

=1: программирование оси по диаметру.

Например, на фрезерном станке программируем ось X и Y в основном случае по радиусу. Если в данном отрезке программы деталей целесообразно задавать значение X и Y по диаметру, в программу деталей надо записать это:

```
... (X, Y задано в радиусе)
...
G10.9 X1 Y1 (программирование в диаметре)
... (X, Y задано в диаметре)
...
G10.9 X0 Y0 (программирование в радиусе)
... (X, Y задано в радиусе)
...
```

Если на токарном станке программируем ось X в диаметре, но при фрезеровании интерполяцией осью X, С желаем программировать X в радиусе:

```
... (X задано в диаметре, точение)
...
G10.9 X0 (программирование в радиусе)
G12.1 (полярная интерполяция вкл.)
... (X, задано в радиусе, фрезерование)
...
G13.1 (полярная интерполяция выкл.)
G10.9 X1 (программирование в диаметре)
... (X задано в диаметре, точение)
...
```

Код G10.9 надо задавать всегда в отдельном кадре!

В конце программы, или под действием reset установленный через G10.9 способ программирования удаляется и устройство ЧПУ примет основное положение, установленное в бите #0 DIA параметра N0106 Axis Properties.

5.4 Задание данных полярными координатами (G15, G16)

Значения запрограммированных координат можно вводить полярными координатами, то есть в мм-ах, или радиусом, заданным в дюймах и углом, заданным в градусах.

G16: Задание данных полярными координатами вкл.

G15: Задание данных полярными координатами выкл.

G15, G16 являются модалными функциями. После включения, в конце программы, или под действием reset устройство ЧПУ попадает в состояние G15.

Данные, заданные в полярных координатах истолкуются в плоскости, определённой через G17, G18, G19. При задании данных устройство ЧПУ считает адрес первой оси плоскости за радиус, а адрес второй оси - за углом.

G17 X_p (радиус) Y_p (угол)

G18 Z_p (радиус) X_p (угол)

G19 Y_p (радиус) Z_p (угол)

X_p , Y_p , Z_p может быть основная ось X, Y, Z, или параллельная ей ось.

В случае задания угла *в направлении против хода часовой стрелки* направление угла *положительное*, а *в направлении по ходу часовой стрелки* направление угла *отрицательное*.

Данные остальных осей, лежащих вне выбранной плоскости, берутся данными в декартовых (прямоугольных) координатах.

Радиус и угол можно задавать и абсолютными данными и данными с приращением

Когда **радиус** задаётся **абсолютными** данными, начало координат актуальной системы координат будет начальной точкой полярной системы координат:

G90 G16 X100 Y60

И **угол** и **радиус** являются **абсолютными данными**, инструмент набегаёт в точку с 60° с радиусом 100 мм..

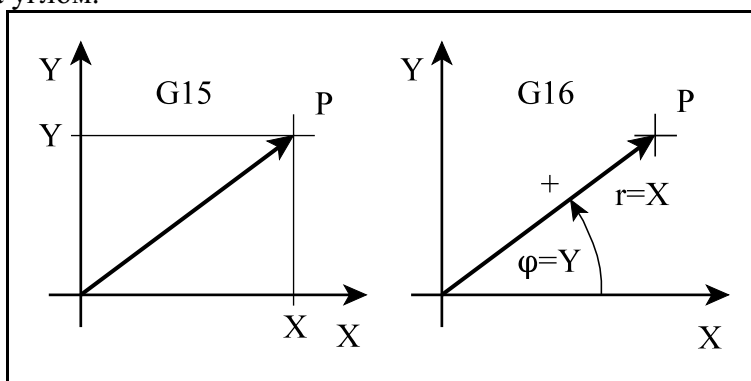


рисунок 5.4-1

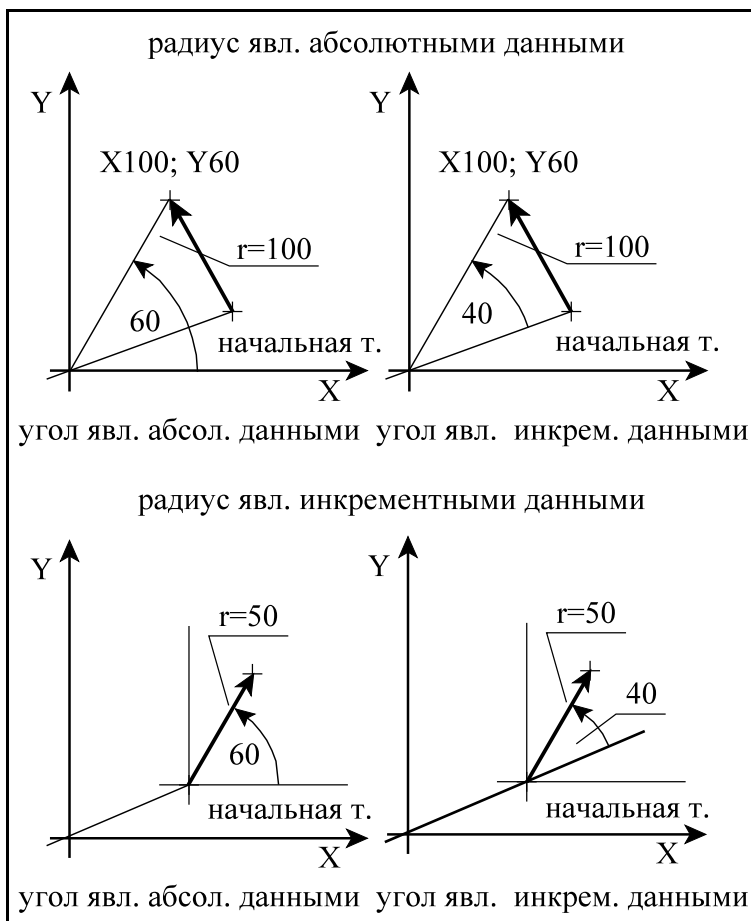


рисунок 5.4-2

G90 G16 X100 YI40

Угол является данными *с приращением*. Инструмент набегает в точку с радиусом 100 мм, по сравнению предыдущему угловому положению на 40° дальше.

Когда определяем **радиус** как значение *с приращением*, перемещение совершается по заданному радиусу, отсчитывая от позиции начала кадра осей в направлении заданного угла:

G90 G16 XI50 Y60

Радиус 50 мм измеряется, отсчитывая от начальной точки и перемещается до абсолютного угла в 60°.

G91 G16 X50 Y40

Радиус 50 мм измеряется, отсчитывая от начальной точки и перемещается до угла, измеренного от начального угла в 40°.

Во включенном состоянии задания данных полярными координатами G16 также можно и окружность программировать. Окружность можно задавать и радиусом и через I, J, K. Однако, в последнем случае устройство ЧПУ считает адрес **I, J, K всегда за прямоугольными данными**.

Если точка центра актуальной системы координат совпадает с точкой центра окружности, заданием данных полярными координатами можно программировать многоповоротную окружность, или винт:

G17 G16 G90 G02 X100 Y-990 Z50 R-100

В кадре выше задан винт с поворотом 2 целых $\frac{3}{4}$, с направлением поворота по ходу часовой стрелки. При программировании многоповоротной окружности следите за тем, что в случае направления G2 надо программировать отрицательный полярный угол, а в случае направления G3 - положительный.

Адреса, которые бывают в следующих командах, устройство ЧПУ не считает за данными полярных координат, даже если включено состояние G16:

- G10 координаты, которые бывают в команде установки,
- G52 смещение координат,
- G92 установка координат,
- G28, G30 координаты промежуточной точки,
- G53 позиционирование в станочной системе координат,
- G68 точка центра поворота системы координат,
- G51 точка центра масштабирования,
- G50.1 точка центра отражения.

Показательный пример: фрезерование шести-угольника

```

N1 G90 G17 G0 X120 Y120 F120
N2 G16 G1 X100 Y60
N3 Y120
N4 Y180
N5 Y240
N6 Y300
N7 Y0
N8 Y60
N9 G15 G0 X120 Y120
    
```

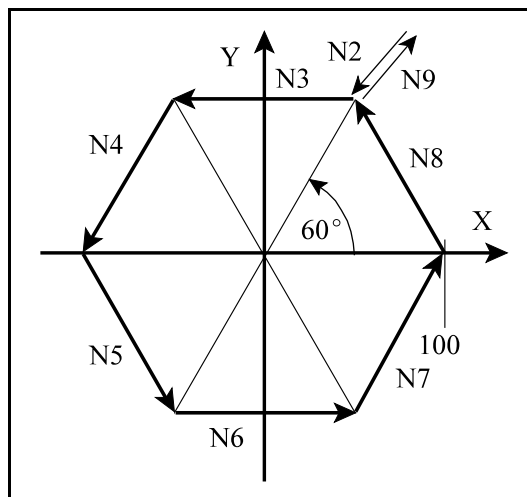


рисунок 5.4-3

5.5 Задание и точность данных координат

Истолкование десятичной запятой в зависимости от применённой системы мер:

- X2.134 означает 2.134 мм, или 2.134 дюйм,
- B24.236 означает 24.236 градусов, если по адресу В задавали данные угла.

Пользоваться десятичной запятой не обязательно:

- X325 означает например 325 мм.

Впереди стоящие нули можно пропускать:

- .032=0.032

После десятичной запятой можно пропускать последующие за ней нули:

- 0.320=.32

Данные координат можно задавать с точностью не более 15 десятичных цифр.

5.6 Обращение переворотом вращающихся осей

Этой функцией можно пользоваться в случае вращающихся осей, то есть тогда, если один из адресов оси, (например С) назначена вращающейся осью .

Под обращением переворотом понимается то, что позиция зарегистрирована на данной оси устройством ЧПУ не между плюс/минус бесконечного, а учитывая периодичность оси, например: между 0° и 360°.

Назначение оси вращающейся осью и его действие

Это назначение надо выполнить установкой положения бита в #1 ROT=1 параметра N0106 Axis Properties. Для вращающейся оси

- устройство ЧПУ **не выполняет дюймы/метрические** конверсию,
- для вращающейся оси **можно разрешить обращение переворотом**.

Разрешение обращения переворотом

Функцию задействует установка бита #0 REN=1 параметра N0107 RollOver Control. Если значение бита:

- =0: устройство ЧПУ обращается вращающейся осью, как линейными осями, и заполнение остальных параметров не имеет действие,
- =1: устройство ЧПУ применяет для вращающейся оси обращение переворотом, суть кото-

рого определяется нижеописанными

Задание хода, совершаемого за один оборот

Параметром N0108 RollAmount задаётся ход, совершаемый за один оборот оси. Значит, если ось совершает за один оборот 360° , на соответствующий параметр надо написать значение: 360. Система может обращаться произвольной периодичностью, не только 360° градусов.

Описанными выше установками параметров позиция вращающейся оси выводится устройством ЧПУ всегда в области между 0° - $+359.999^\circ$, независимо от того, что в какое направление вращалась, и сколько оборотов совершила вращающаяся ось.

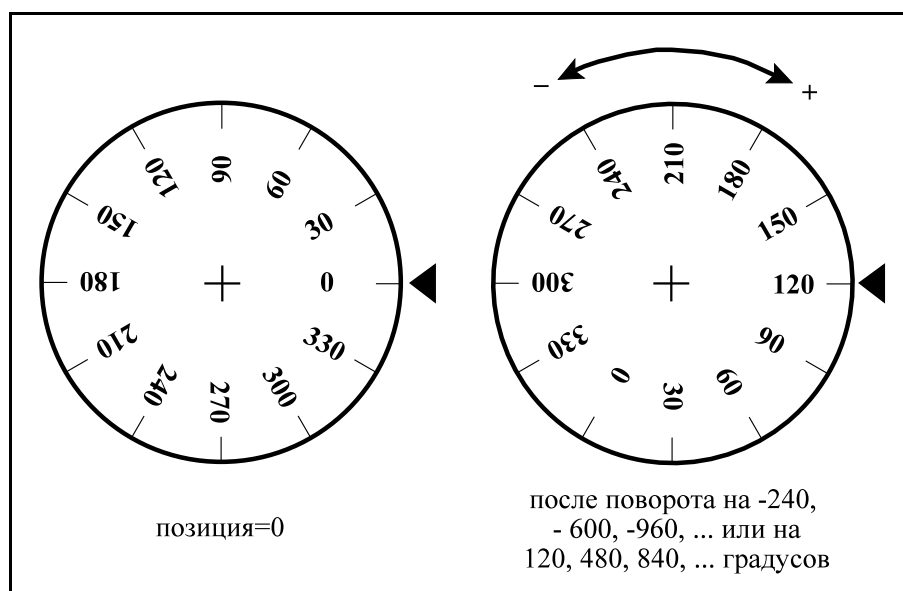


рисунок 5.6-1

Перечень последующих за этим возможностей установок предполагает установку бита в #0 REN=1 параметра N0107 RollOver Control и установку параметра N0108 RollAmount.

Вращение в заданную позицию: основной случай вращения

В случае установки параметра N0107 RollOver Control в

#2 ABS=0 и #1 ASH=0

при абсолютном программировании (G90) вращение выполняется в заданную позицию. Величина перемещения будет всегда меньше значения, заданного параметром RollAmount, значит, в ходе движения всегда срезаются целые повороты.

Пример:

Пусть будет исходная позиция оси C

C=0

Под действием команды

G90 C210

G90 C570

G90 C930

...

перемещается всегда в положительное направление 210° градусов ($\Delta C=210^\circ$), срезая целые

повороты, и позиция конечной точки будет 210° .

Пусть будет исходная позиция оси C

$C=0$

Под действием команды

G90 C-210

G90 C-570

G90 C-930

...

перемещается всегда в отрицательное направление 210 градусов ($\Delta C = -210^\circ$), срезая целые повороты, и позиция конечной точки будет $150^\circ (= -210^\circ)$.

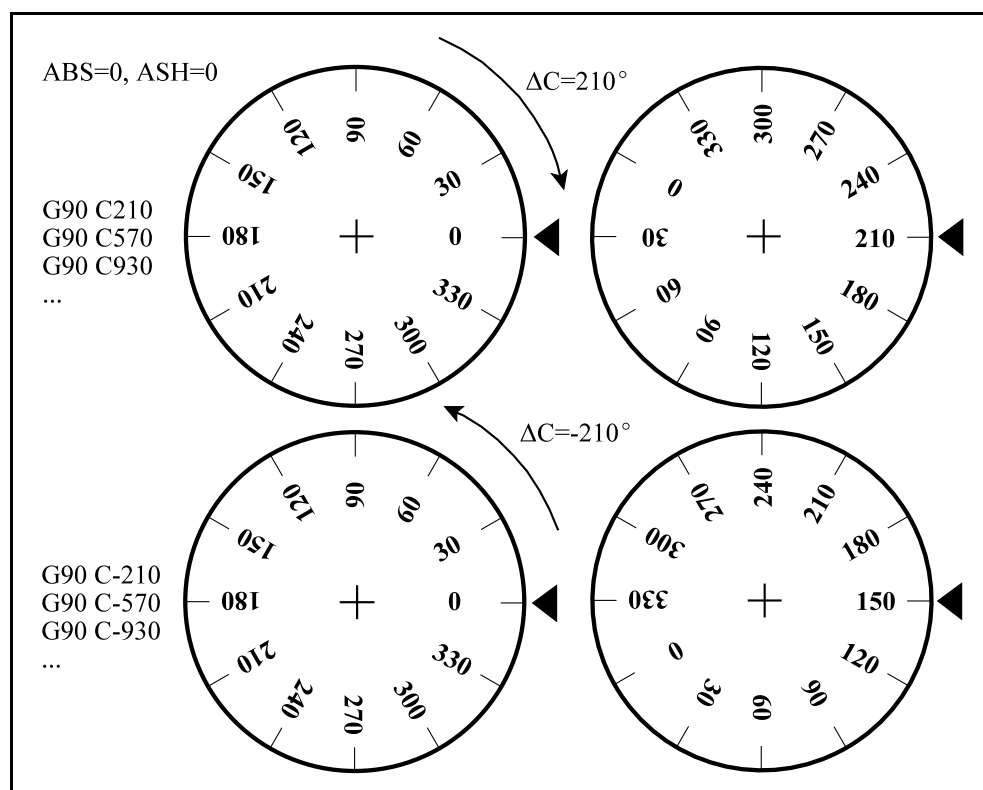


рисунок 5.6-2

Вращение в заданную позицию кратчайшим путём

В случае установки параметра N0107 RollOver Control в

#2 ABS=0 és #1 ASH=1

при абсолютном программировании (G90) вращение выполняется в заданную позицию, вращая в направление, принадлежащее к кратчайшему пути. Величина перемещения будет всегда меньше значения, заданного параметром RollAmount, значит, в ходе движения всегда срезаются целые повороты.

Пример:

Пусть будет исходная позиция оси C

$C=0$

Под действием команды

G90 C210

G90 C570

G90 C930

...

перемещается в отрицательное направление (кратчайшим путём) 150 градусов ($\Delta C = -150^\circ$), срезая целые повороты, и позиция конечной точки будет 210° .

Пусть будет исходная позиция оси C

C=0

Под действием команды

G90 C-210

G90 C-570

G90 C-930

...

перемещается в положительное направление (кратчайшим путём) 150 градусов ($\Delta C = 150^\circ$), срезая целые повороты, и позиция конечной точки будет $150^\circ (= -210^\circ)$.

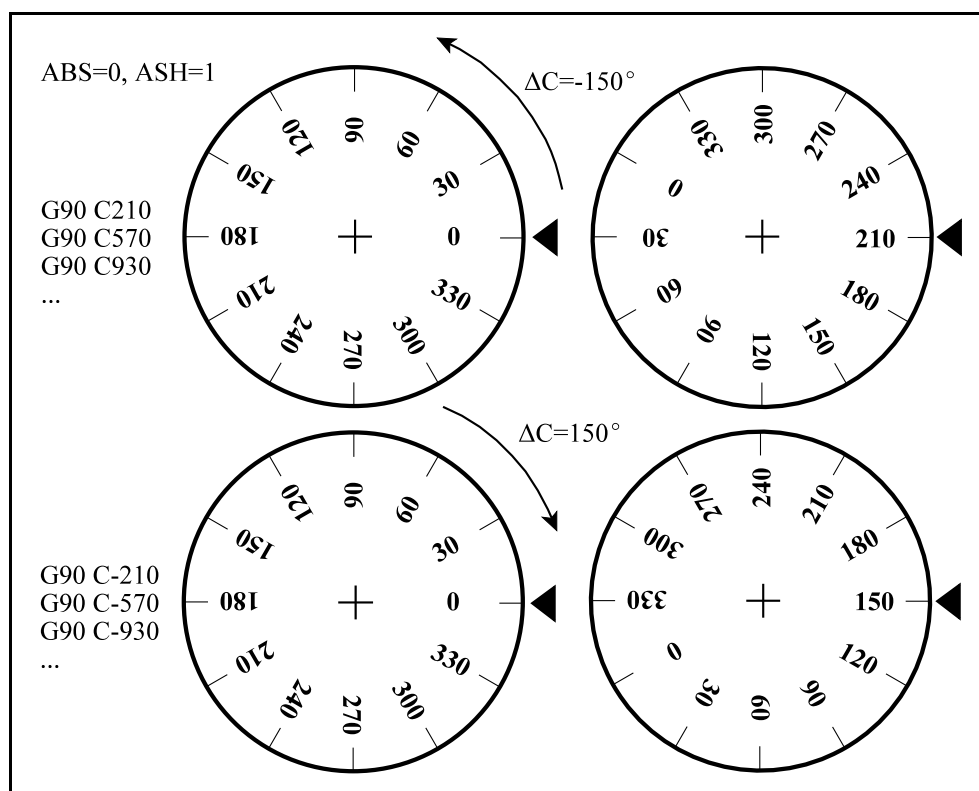


рисунок 5.6-3

Вращение в абсолютное значение заданной позиции в направлении, согласно знаку

В случае установки параметра N0107 RollOver Control в

#2 ABS=1 és #1 ASH=0

при абсолютном программировании (G90) вращение выполняется в абсолютное значение заданной позиции в направлении, согласно знаку. Величина перемещения будет всегда меньше значения, заданного параметром RollAmount, значит, в ходе движения всегда срезаются целые повороты.

Пример:

Пусть будет исходная позиция оси C

C=0

Под действием команды

```
G90 C30
G90 C390
G90 C750
```

...

перемещается в положительное направление 30 градусов ($\Delta C=30^\circ$), срезая целые повороты, и позиция конечной точки будет 30° .

Пусть будет исходная позиция оси C

$C=0$

Под действием команды

```
G90 C-30
G90 C-390
G90 C-750
```

...

перемещается в отрицательное направление 330 градусов ($\Delta C=-330^\circ$), срезая целые повороты, и позиция конечной точки будет 30° .

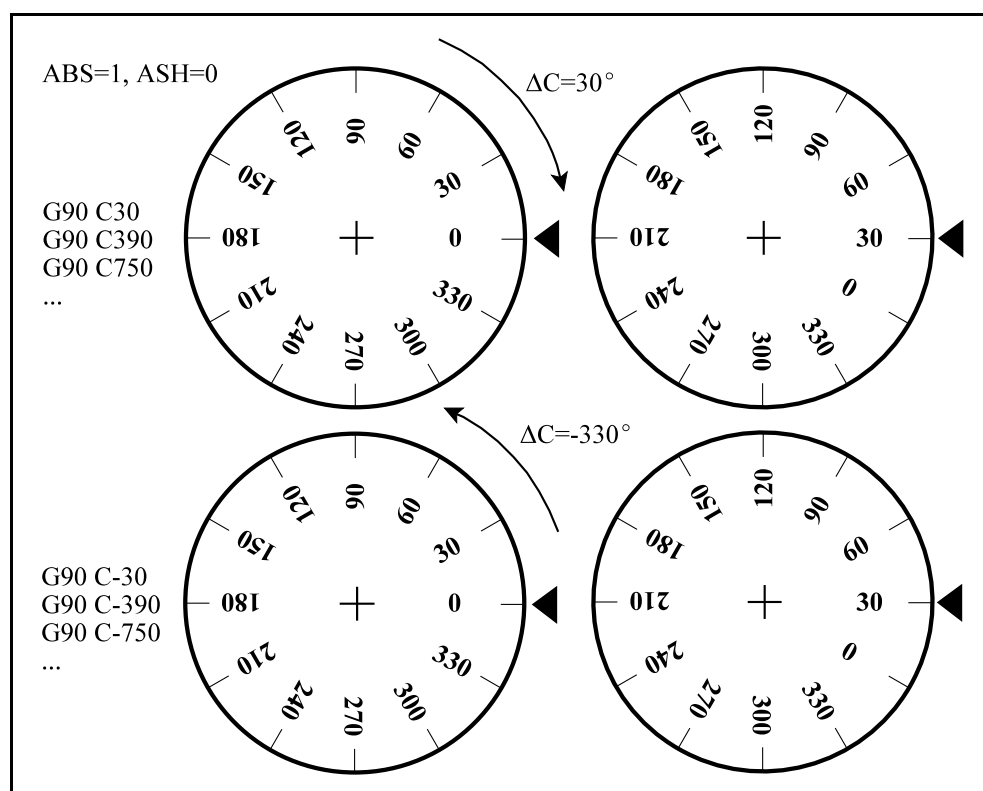


рисунок 5.6-4

☞ **Замечание:** 0 (нуль) является положительным числом! Значит, запрограммируя позицию 0, перемещение совершается в положительное направление.

Пусть будет исходная позиция оси C

$C=30$

Под действием команды

```
G90 C0
```

перемещается в положительное направление 330 градусов ($\Delta C=330^\circ$), и конечная позиция будет 0 градусов.

При желании совершить перемещение в отрицательное направление с 30 градусов до 0 градусов, надо запрограммировать кадр

G90 C-360

При этом перемещение совершается в отрицательное направление 30 градусов ($\Delta C = -30^\circ$), и конечная позиция будет 0 градусов.

Перемещение вращающейся оси в случае инкрементного программирования

В случае программирования инкрементного задания данных перемещения происходит всегда в направлении согласно запрограммированного знака.

В случае положения параметра N0107 RollOver Control

REN=0

для пути, запрограммированного инкрементно, не применяется параметр Roll Amount, значит, перемещение на несколько оборотов можно запрограммировать и инкрементно.

Пусть будет исходная позиция оси C

C=0

При выполнении кадров

G91 C30 (сдвиг на 30 градусов)

G91 C390 (сдвиг на 390 градусов)

G91 C750 (сдвиг на 750 градусов)

...

не срезаются целые обороты. Однако, в конце движения на индикаторе позиции, позиция оси C будет всегда 30 градусов, так как для отражения позиции всегда действительно значение, установленное параметром RollAmount.

В случае положения параметра N0107 RollOver Control

REN=1

и для пути, запрограммированного инкрементно применяется параметр Roll Amount, значит, даже инкрементным заданием нельзя запрограммировать перемещение, большее одного оборота.

Пусть будет исходная позиция оси C

C=0

При выполнении кадров

G91 C30 (сдвиг на 30 градусов)

G91 C390 (сдвиг на 30 градусов)

G91 C750 (сдвиг на 30 градусов)

...

срезаются целые обороты. Однако, на индикаторе позиция оси C будет всегда 30 градусов.

6 Подача

6.1 Подача быстрым ходом

Позиционирование происходит быстрым ходом посредством команды G0. Кроме позиционирования G0 быстрым ходом перемещаются команды G53, G28, G30, а также фазы, позиционирующие циклы.

Значение быстрого хода позиционирования по осям **устанавливается строителем станка параметром в единицах мм/мин, дюйм/мин, или градус/мин**. Величина быстрого хода может отличаться по осям.

Если одновременно несколько осей выполняет перемещение быстрым ходом, значение результирующей подачи рассчитывается устройством ЧПУ так, чтобы проектируемая на оси составляющая скорости ни на одной оси не превышала действующее для той оси значение быстрого хода, заданное параметром, и позиционирование произошло за минимальное время.

Величину подачи быстрого хода изменяет включатель процента быстрого хода (override). Работа override быстрого хода определяется строителем станка в программе PLC. Описание этого смотрите в руководстве, выданным строителем станка.

Значение override быстрого хода не идёт выше 100%.

Подачу быстрого хода всегда останавливает в 0% положения включателя процента быстрого хода.

Не имея действующую референтную точку, будут в силу уменьшенные значения быстрого хода по осям, определённые в поле параметров строителем станка до тех пор, пока не происходит приём референтной точки.

При перемещении суппортов клавишами (jog) перемещения осей, скорость быстрого хода является разным по осям значением, отличающимся от быстрого хода позиционирования, установленного также параметром. Оно по смыслу является меньшим значением скорости позиционирования, чтобы для останова можно было учитывать и время реакции человека.

6.2 Рабочая подача

Подача программируется
по адресу F

Запрограммированная подача действует в линейных кадрах (G01) и в кадрах круговой интерполяции (G02, G03).

Единицу измерения подачи определяют кодами **G94, G95**.

Подача рассчитывается тангенциально вдоль запрограммированной траектории устройством ЧПУ.

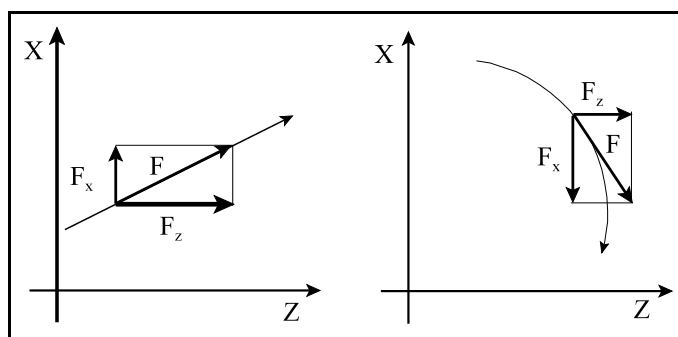


рисунок 6.2-1

F : величина подачи в направлении касательной (запрограммированное значение)

F_x : составляющая подачи в направлении X

F_z : составляющая подачи в направлении Z

$$F = \sqrt{F_x^2 + F_z^2}$$

Запрограммированную подачу можно изменить включателем процента быстрого хода (override), кроме состояния G63, включателя процента и стопа.

Работу override подачи определяет строитель станка в программе PLC. Его описание и предела значений смотрите в руководстве, выданным строителем станка.

Подача является модальным значением.

После включения устройство ЧПУ примет значение, определённое параметром. В состоянии G94 берётся начальное значение подачи с параметра N0300 Default F G94, а в состоянии G95 - с параметра N0301 Default F G95..

Строителем станка в поле параметров ограничено по осям максимальная подача, программируемая на данном станке. Установленное здесь значение истолкуется всегда в поминутной размерности. Это значение является заодно скоростью движения подачи во включенном состоянии включателя СУХОЙ БЕГ.

Значение подачи можно задавать с точностью не более 15 десятичных цифр. Можно пользоваться десятичной запятой. Оно всегда положительное число.

6.2.1 Поминутная подача (G94) и подача за оборот (G95)

Единицу измерения подачи можно задавать в программе кодами G94 и G95:

G94: поминутная подача

G95: подача за оборот

Под поминутной подачей понимается подача, заданная в размерности мм/мин, дюйм/мин, или градус/мин.

Под подачей за оборот понимается подача, выполненная за один оборот шпинделя в размерности мм/об, дюйм/об или градус/об.

Значения являются модальным.

После включения, reset, или конец программы битом #0 G95 параметра N1301 DefaultG2 решается, что устройство ЧПУ примет состояние G94, или G95.

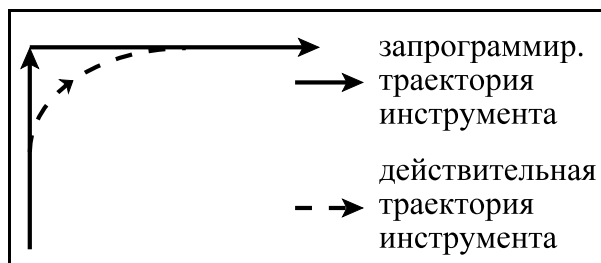
Состояние G94/G95 не влияет на подачу быстрого хода, она понимается всегда в поминутной размерности.

6.3 Функции управления подачей

Функции управления подачей нужны

- при **обработке углов**,
- далее в таком случае, когда технология требует, что включатели **override** или **stop** были **запрещены**.

Если используется непрерывный режим точения, при обработке углов суппорты не могут следить за командами хода, выданными устройством ЧПУ из-за их инертности. При этом инструмент в большей или меньшей мере скругляет углы, в зависимости от подачи.



Если для заготовки требуются острые углы, тогда надо сообщить устройству ЧПУ, чтобы замедлило в конце перемещения, дождало, пока останавливаются оси, и только после этого запустило следующее движение.

рисунок 6.3-1

6.3.1 Точный останов в конце кадра (G9)

Эта функция не является *модальной*, действует только в том кадре, в котором запрограммировали.

В *конце* того *кадра*, где задавали его, устройство ЧПУ замедляет после выполнения интерполяции, *остановится и дожидается сигнал система мер в позиции*.

Если сигнал не приходит до установленного параметром времени, устройство ЧПУ *выдаёт сообщение 2501 Ошибка позиции*.

Эта функция служит для точного обхода острых углов.

6.3.2 Режим точного останова (G61)

Функция является модальной. Удаляется командой G62, G63, G64.

Устройство ЧПУ замедляет *в конце каждого кадра, остановится и дожидается сигнал система мер в позиции*, и только после этого запускает следующий цикл интерполяции.

Если сигнал не приходит до установленного параметром времени, устройство ЧПУ *выдаёт сообщение 2501 Ошибка позиции*.

Эта функция служит для точного обхода острых углов.

6.3.3 Нерывное точение (G64)

Функция является модальной. *После включения, reset, или конец программы устройство ЧПУ примет это состояние*. Следующие коды прекращают это состояние: G61, G62, G63.

В этом режиме после выполнения интерполяции не остановится движение, оси не дожидают сигнала система мер в позиции, а сразу начинается интерполяция следующего кадра.

В этом режиме нельзя обработать острые углы, так как у переходов они скругляются.

6.3.4 Запрет override и стопа (G63)

Функция является модальной. Коды G61, G62, G64 прекращают это состояние.

В этом режиме *не действительны включатель подачи и включатель процента шпинделя (override), а также стоп подачи*. Значение процента берётся за 100%, не зависимо от их положения. После выполнения интерполяции устройство ЧПУ не замедляет, а сразу запускает следующий цикл интерполяции.

Этот режим можно использовать в случае обработки разных резьб.

6.3.5 Автоматическое уменьшение подачи при внутренних углах (G62)

Функция является модальной. Коды G61, G63, G64 прекращают это состояние.

В случае обработки внутренних углов увеличивается усилие, действующее на инструмент на участке перед углами и после углов. Для того, чтобы инструмент не вибрировал, и поверхность осталась соответствующей, во включенном состоянии G62 устройство ЧПУ автоматически уменьшает подачу перед и после внутренних углов.

Уменьшение подачи действует при следующих условиях:

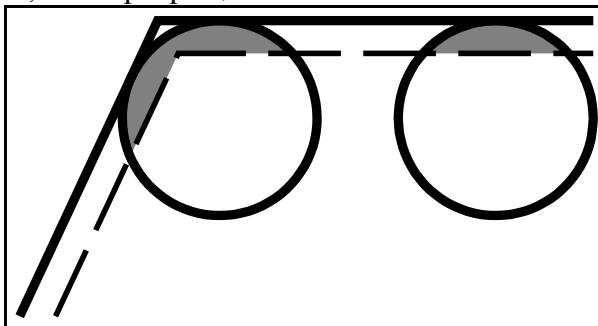


рисунок 6.3.5-1

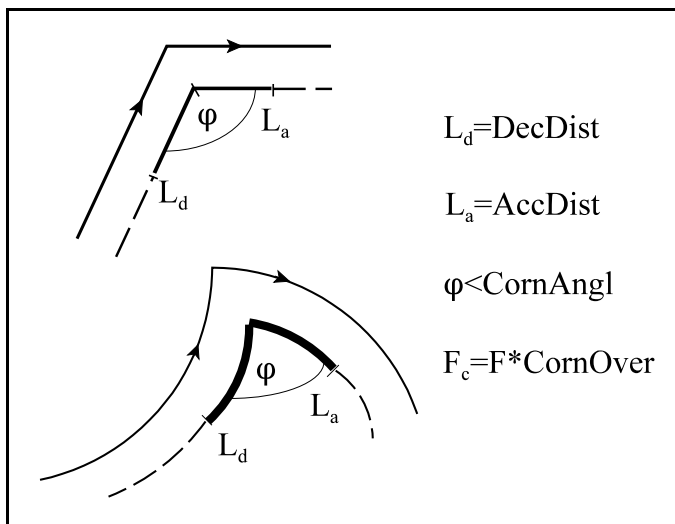
- 1. во включенном состоянии плоскостной коррекции радиуса инструмента (G41, G42),
- 2. между кадрами G1, G2, G3,
- 3. при перемещениях, выполненных в выбранной плоскости,
- 4. если инструмент обходит угол изнутри,
- 5. если угол меньше, чем определённый параметром угол,
- 6. перед углом и после угла на расстояние, определённое параметром.

Функция уменьшение подачи работает при всех четырёх возможных переходах: прямая-прямая, прямая-окружность, окружность-прямая, окружность-окружность.

Значение внутреннего угла φ можно установить параметром N1409 CornAngle в области угла 1–180°.

Перед углом на расстояние L_d начинается замедление, а после угла на расстояние L_a - ускорение. В случае дуги окружности расстояние L_d и L_a учитывается устройством ЧПУ вдоль дуги.

Задание расстояния L_d выполняется параметром N1407 DecDist, а расстояния L_a - параметром N1408 AccDist.



На параметре N1410 CornOver можно задавать числом соотношения между 0 и 1, что до какой меры должно устройство ЧПУ уменьшить подачу у внутренних углов. Подача будет

$$F * \text{CornOver}$$

где F - запрограммированная подача. На подачу, полученную таким образом, влияет и включатель override.

Если в состоянии G62 желаем программировать точный останов, в данный кадр следует записать G09.

6.4 Автоматическое уменьшение подачи при внутренних дуг

Во включенном состоянии плоскостной коррекции радиуса инструмента (G41, G42), запрограммированная в ходе круговой интерполяции подача действует вдоль точки центра радиуса инструмента. При обработке внутренних дуг устройство ЧПУ автоматически уменьшает значение подачи, чтобы запрограммированная подача действовала по радиусу точения. Величина подачи в точке центра радиуса инструмента:

$$F_c = \frac{R_c}{R} F$$

где F_c : подача в точке центра радиуса инструмента (корректированная подача)

R : запрограммированный радиус окружности

R_c : корректированный радиус окружности

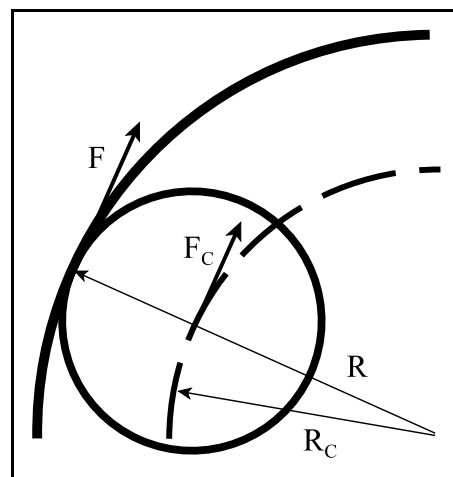


рисунок 6.4-1

F: запрограммированная подача.

Автоматическое уменьшение подачи ограничивается снизу параметром N1406 CircOver, где минимальное уменьшение подачи можно задавать числом соотношения между 0 и 1, то есть для актуальной подачи удовлетворяется условие

$$F_c \geq F * \text{CircOver}$$

Произведение override, вызванное радиусом окружности и значение подачи и углового override будет выдаваться.

7 Ускорение

Ускорение - это мера изменения скорости по времени. Чем короче времени нужно для достижения данной скорости, тем больше ускорение.

Чем больше ускорение желаем достичь, тем больше мощности должны иметь двигателя и привод.

Величина сил, действующие на станок во время движения, в конечном итоге нагрузка станка прямо пропорциональна ускорению, возникающему на станке.

Параметры ускорения устанавливаются строителем станка по осям с учётом вышеописанных двух точек зрения.

Устройство ЧПУ ускоряет всегда значение (векториальное) подачи вдоль траектории. Значение ускорения рассчитывается таким образом, чтобы величина составляющих ускорения по осям ни на одной оси не превосходила значение, установленное на ось.

Ускорение можно установить два вида:

- линейное и
- формы колоколообразной кривой.

В случае **линейного ускорения** во время ускорения, или замедления значение ускорения постоянное, устройство ЧПУ увеличивает подачу по линейной функции при запуске, или уменьшает при останове. Величина ускорения устанавливается параметром по осям.

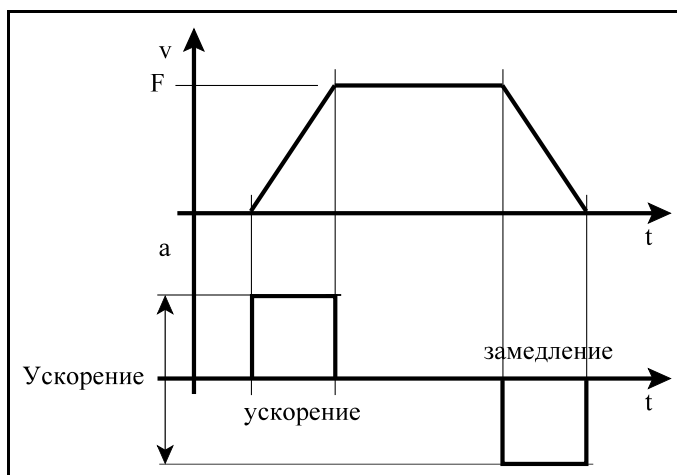


рисунок 7-1

В случае **колоколообразного ускорения** во время ускорения, или замедления меняется и значение ускорения, линейно растёт до достижения установленного значения ускорения или линейно уменьшается перед достижением целевой скорости. В следствие этого форма набега и сбег подачи имеет форму колоколообразной кривой в функции от времени, поэтому называется ускорением формы колоколообразной кривой.

Время T , за которое устройство ЧПУ

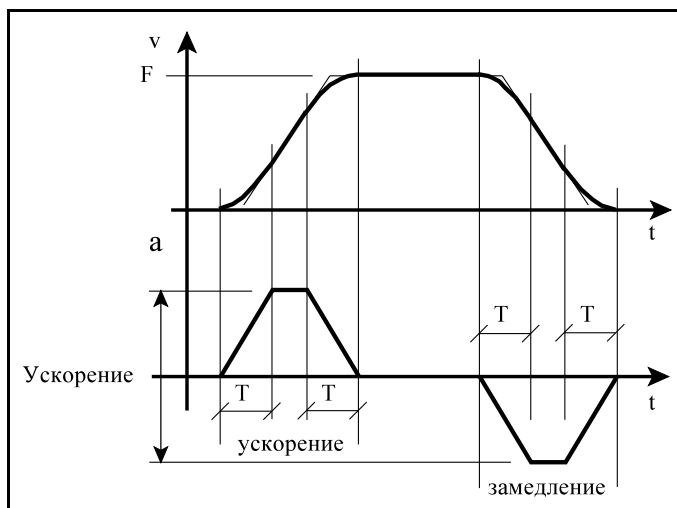


рисунок 7-2

достигает установленное значение ускорения, можно установить по осям, параметром.

Установлением **ускорения без рывка** можно дальше увеличить качество ускорения формы колоколообразной кривой. Включить ускорение без рывка можно установкой бита в #1 JRK=1 параметра N0421 Acc Contr. При этом даже набег и сбег функции ускорения будет иметь форму колоколообразной кривой, иначе говоря и в производной первого порядка (j) ускорения (a) не будет скачка.

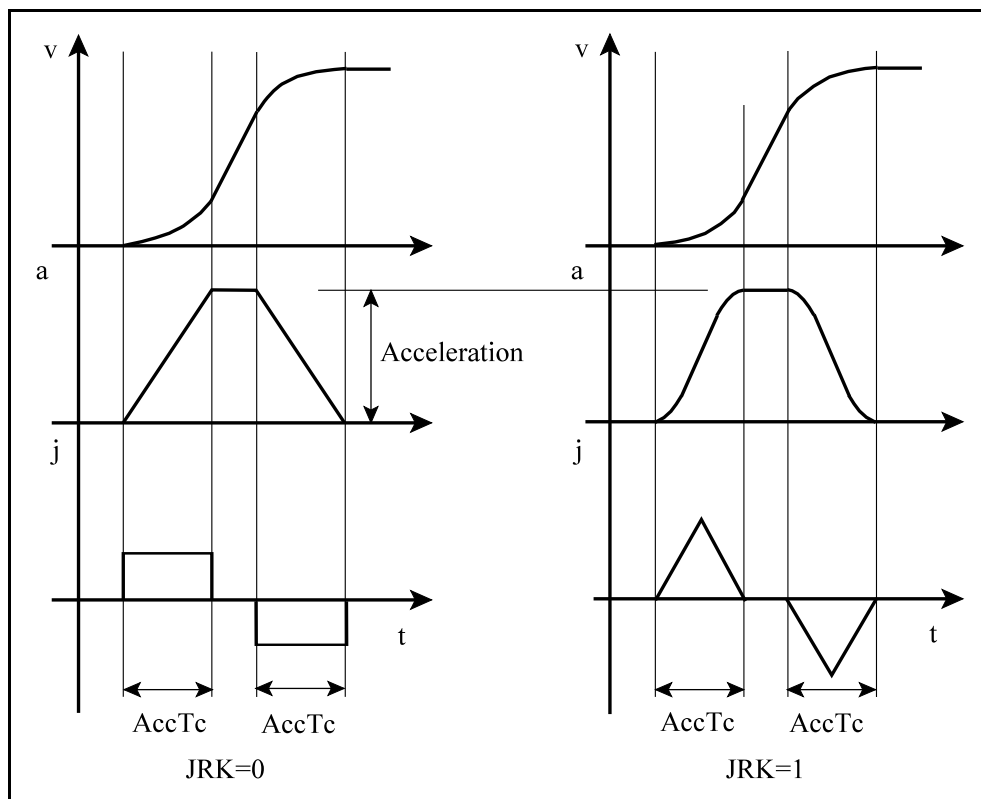


рисунок 7-3

Установкой JRK=1 ускорения без рывка можно установить на станке и большее ускорение, в то же время запуск и останов будет более качественным.

В случае высокоскоростной обработки для достижения требуемой точности необходимо применение скорости с опережающей связью. В этом случае надо установить всегда ускорение с формой колоколообразной кривой!

В нормальных условиях устройство ЧПУ ускоряет или замедляет в следующих случаях:

- при ручных перемещениях,
- в ходе позиционирования быстрого хода (G0) в начале кадра перемещение запускается со скоростью 0, и в конце позиционирования замедляет до скорости 0,
- в случае движения подачи (G1, G2, G3) в состоянии G9, или G61 в конце кадра всегда замедляет до скорости 0,
- замедляет, если подачу останавливаем клавишей STOP, или ускоряет, если подачу запускаем в движение со START-ом,
- остановится замедлением, если после движения следует выполнение функции и в конце

кадра, если включен режим ПО КАДРАМ.

Ускорение до нового значения, большего предыдущего, устройство ЧПУ начинает всегда в ходе выполнения того кадра, в котором задавалась новая подача. Этот процесс может протянуться через несколько кадров.

Замедление до нового значения, меньшего предыдущего, устройство ЧПУ начинает всегда с такого соответствующего предыдущего кадра, что в том кадре, где задавалась новая подача, обработка началась уже с запрограммированной в данном кадре скоростью.

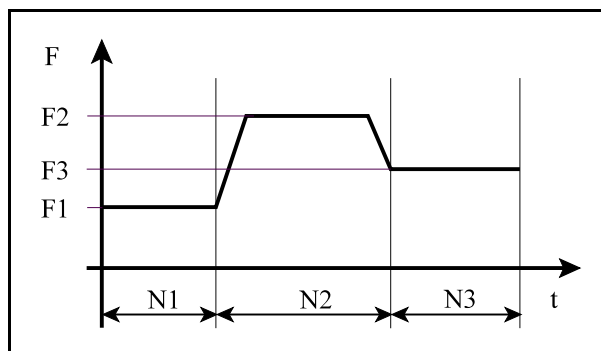


рисунок 7-4

Устройство ЧПУ предварительно следит за тангенциальными изменениями скорости, и регистрирует их. Это необходимо для того, чтобы желаемую целевую скорость достигалась непрерывным ускорением, протянувшимся хоть через выполнение нескольких кадров.

Также может протянуться достижение желаемой целевой скорости через несколько кадров при запуске или останове.

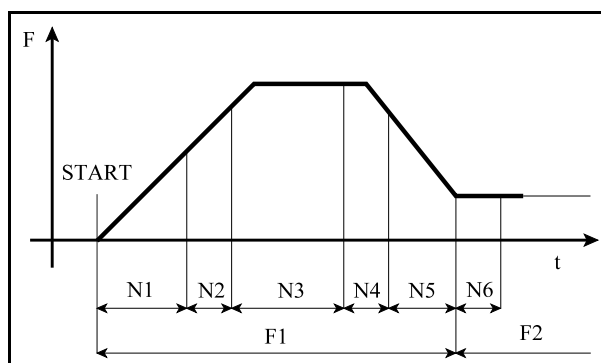


рисунок 7-5

7.1 Автоматическое замедление при углах в состоянии G64

В случае непрерывного точения, в состоянии G64, устройство ЧПУ старается следить за траекторией с запрограммированной подачей.

Если между двумя кадрами находит угол, оно должно замедлить подачу вдоль траектории.

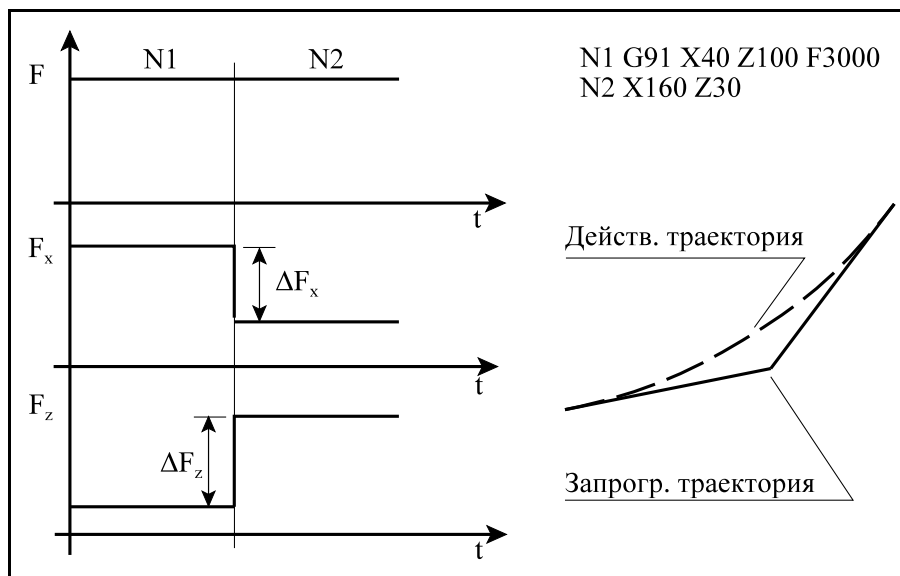


рисунок 7.1-1

Если не замедляет при углах в двух, последующих друг за другом кадрах N1, N2, тогда возникают вдоль некоторых осей разность подачи (ΔF_x , ΔF_z) показанные на рисунке.

Обнаружение изменений подачи (углов), и заодно уменьшение подачи необходимо по двум причинам:

- Изменения подачи (ускорения) по осям, возникающие вследствие резких изменений направления траектории могут быть настолько большими, что без замедления приводы не могут без колебаний следовать за ними, и это может приводить к ущербу точности, далее механически слышком нагружает инструментальный станок.
- Если в ходе точения необходимо получить “острые” углы, но не хочется остановиться и этим увеличить время точения (программировать точный останов G61), тоже надо замедлить. Чем больше замедляем подачу, тем острее получается угол.

Устройство ЧПУ может выполнить обнаружение углов двояко: проследив за изменением угла направления траектории, а также проследив за изменением составляющих подачи по осям. На основании параметров можно выбирать, что на основании какого способа сработало замедление.

Замедление при углов следив за изменением угла направления траектории.

При положении бита #0 FDF=0 параметра N0306 Feed Control, замедление происходит следив за изменением угла направления траектории.

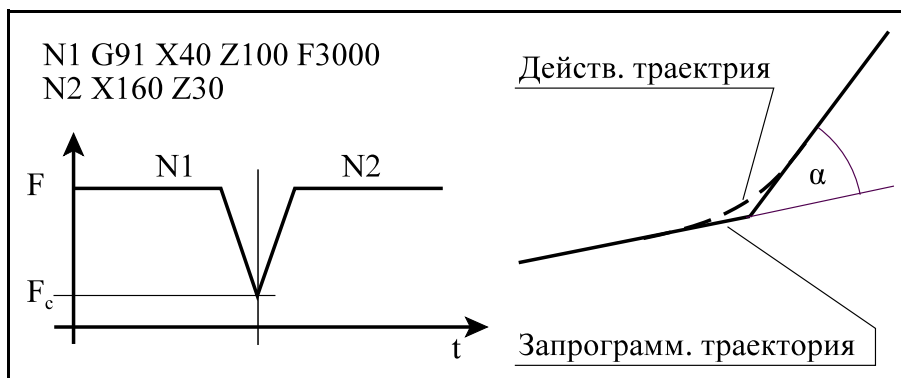


рисунок 7.1-2

Если в точке встречи кадров $N1$, $N2$, показанных на рисунке значение угла α превышает допустимое параметром значение, устройство ЧПУ замедляет подачу до значения F_c .

Значение критического угла можно установить параметром $N0307$ Crit A Diff в градусах: $\alpha = \text{Crit A Diff}$. Значение параметром $N0308$ Feed Corn задаётся, что превышая критического угла, до какой угловой подачи должно замедлить устройство ЧПУ: $F_c = \text{Feed Corn}$.

Чем больше значение критического угла и угловой подачи, тем быстрее идёт обработка, однако тем сильнее нагружается станок и сильнее закругляется угол.

Эта установка не подходит при высокоскоростной обработке!

Замедление при углах, следив за изменением составляющих подачи по осям.

При положении бита #0 FDF=1 параметра N0306 Feed Control, замедление происходит следив за изменением составляющих подачи N0306 Feed Control.

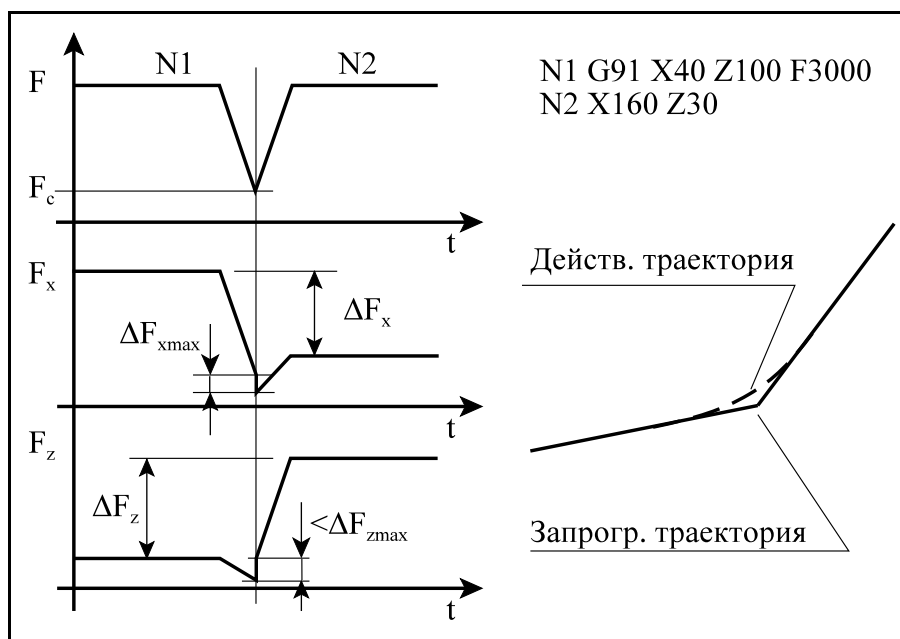


рисунок 7.1-3

Если в точке встречи кадров N1, N2, показанных на рисунке, значение **изменения составляющих подачи** ΔF_x , ΔF_z превышает допустимое параметром максимальное значение, устройство ЧПУ замедляет подачу вдоль траектории F до значения F_c .

Подача замедляется до той меры, чтобы мера изменения подачи не превышала ни на одной оси допустимую для той оси критическую разность подачи (ΔF_{xmax} , ΔF_{zmax}), заданную по осям параметром N0309 Crit F Diff: $\Delta F_{xmax} = \text{Crit F Diff}_x$, $\Delta F_{zmax} = \text{Crit F Diff}_z$.

Чем больше значение критического значения разности подачи, тем быстрее идёт обработка, однако тем сильнее нагружается станок и сильнее закругляется угол.

Устройство ЧПУ на основании установки ускорения сверху ограничивает установленное значение критической разности подачи.

Эту установку надо применить при высокоскоростной обработке!

7.2 Ограничение ускорений, возникающих вдоль траектории в нормальном направлении

Устройство ЧПУ в ходе обработки поддерживает подачу вдоль касательной траектории (в тангенциальном направлении) в постоянном значении. Вследствие этого в тангенциальном направлении не возникают составляющие ускорения. Не так в нормальном направлении (перпендикулярно к траектории, или перпендикулярно к скорости). Составляющие ускорения в нормальном направлении по осям на отдельных осях могут выходить из допустимого на данную ось значения. Для избежания этого скорость вдоль траектории надо ограничивать по мере кривизны траектории.

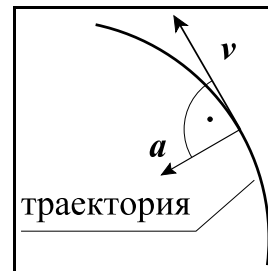


рисунок 7.2-1

Максимальное значение допускаемого ускорения в нормальном направлении можно установить параметром N0402 Normal Acc.

Ограничение ускорения в нормальном направлении при дугах окружности

В ходе обработке дуг окружности величину подачи F ограничивается на основании зависимости

$$F = \sqrt{a \cdot r}$$

где:

а: меньшее из значений ускорения, установленное для осей, участвующих в круговой интерполяции,

г: радиус окружности.

Круговая интерполяция начинается уже с рассчитанной таким образом скоростью.

Устройство ЧПУ не уменьшает скорость на меньшее значение, заданное для подачи параметром N0310 Circ F Min, независимо от указанной выше зависимости.

Внимание:

Эту функцию нельзя перепутать автоматическим уменьшением подачи при обработке внутренних дуг окружности в состоянии G41, G42!

Ограничение ускорения в нормальном направлении при прочих интерполяциях

В случае линейной интерполяции данный участок (траектория) прадва не имеет кривизну, поэтому не имеет и составляющее ускорения в нормальном направлении, однако это верно только для длинных прямых участков. Если **траектория построена из мелких, прямых участков**, как например, это характерно для производства инструментов, тогда получаемая таким образом кривизна траектории может быть значительной, и подачу следует замедлить, как это видно из примера ниже:

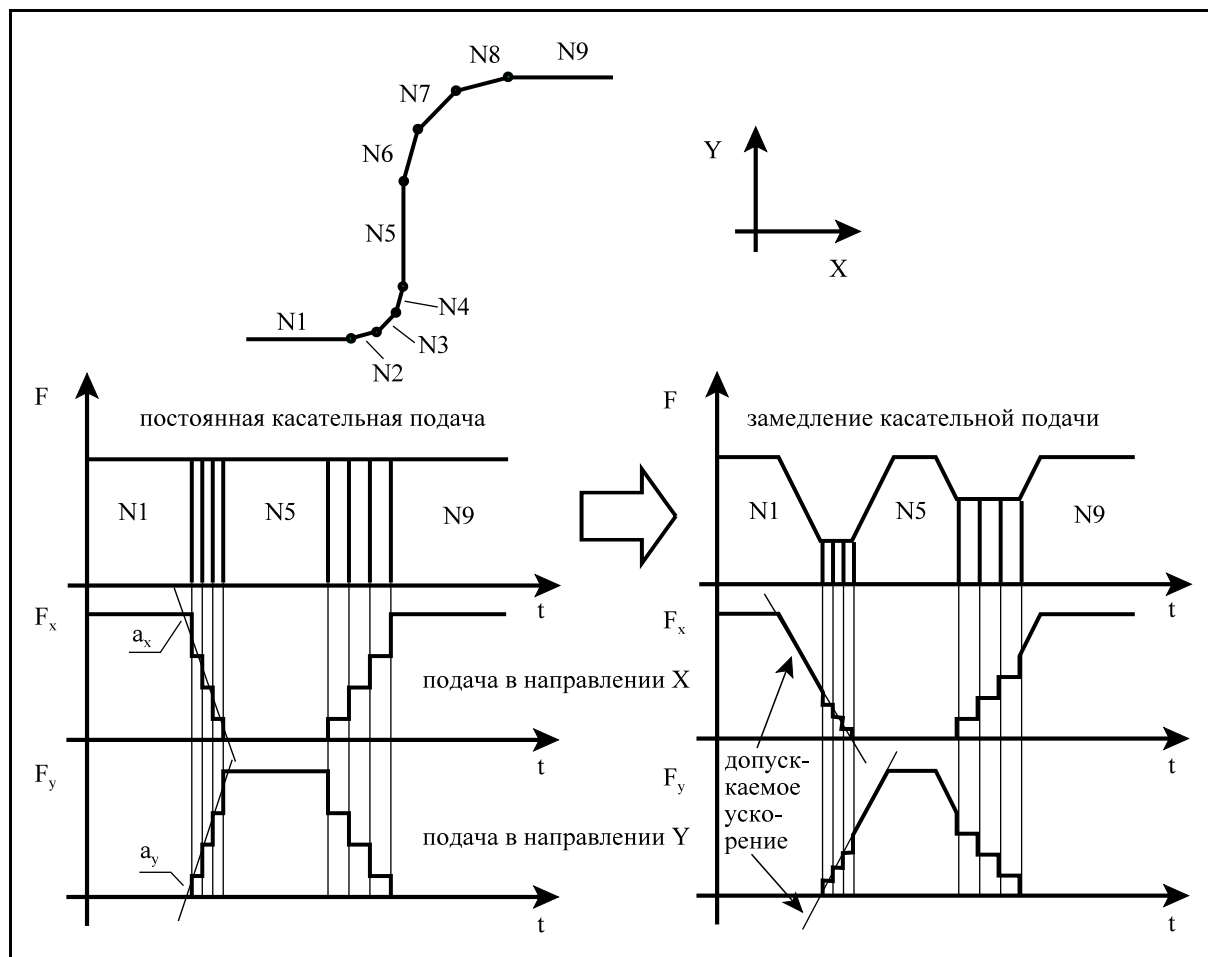


рисунок 7.2-2

В кадрах N2, N3, N4, далее N6, N7, и N8 траектория построена из коротких, прямых участков. Если поддерживать подачу вдоль траектории на постоянном значении (левая диаграмма на рисунке) исходя из геометрии (изменения направления) траектории, крутизна изменения скорости на оси X, далее на оси Y (ускорение в нормальном направлении) может превышать допустимое для данной оси значение.

Поэтому устройство ЧПУ с кадра на кадр проверит, чтобы смог ограничивать ускорения в нормальном направлении. Там, где исходя из геометрии, составляющие ускорения вдоль отдельных осей больше допустимого, там надо замедлить скорость вдоль траектории. Диаграмма с правой стороны показывает, как уменьшается мера изменения скорости (ускорение в нормальном направлении) вдоль отдельных осей под действием замедления подачи вдоль траектории.

В случае чистовой интерполяции G5.1 Q2 устройство ЧПУ также проверяет ускорения в нормальном направлении, возникающие исходя из кривизны траектории, и в случае необходимости уменьшает подачу.

7.3 Ограничение скачка ускорения

На некоторых участках траектории может возникнуть незапный скачок, дёргание в ускорении, вызывающий колебание, механически нагружающее станок и отражающееся на поверхности точения. Таким случаем является например, когда обработка продолжается после прямого участка касающейся окружностью, или когда после дуги окружности последует касательная прямая.

Целью этой функции является, чтобы устройство ЧПУ, путём уменьшения подачи в точке перехода, ограничивало меру скачка в ускорении.

Ограничение скачка ускорения в начале и в конце периода

При переходе из прямого участка в касающуюся дугу окружности подачей F , мера скачка в ускорении рассчитывается из зависимости

$$a = \frac{F^2}{r}$$

где:

a : величина скачка в ускорении,

r : радиус окружности.

Если например, подачей $F6000$ станок входит в дугу окружности с радиусом 10 мм , как это видно на рисунке ниже, числовое значение получаемого на оси Y скачка в ускорении:

$$a = \frac{\left(\frac{6000\text{ мм}}{60\text{ сек}} \right)^2}{10\text{ мм}} = 1000 \frac{\text{мм}}{\text{сек}^2}$$

При желании, чтобы скачок в ускорении не было больше, чем 250 мм/сек^2 , на основании уравнения выше подачу надо уменьшить до $F=50*60=3000\text{ мм/мин}$.

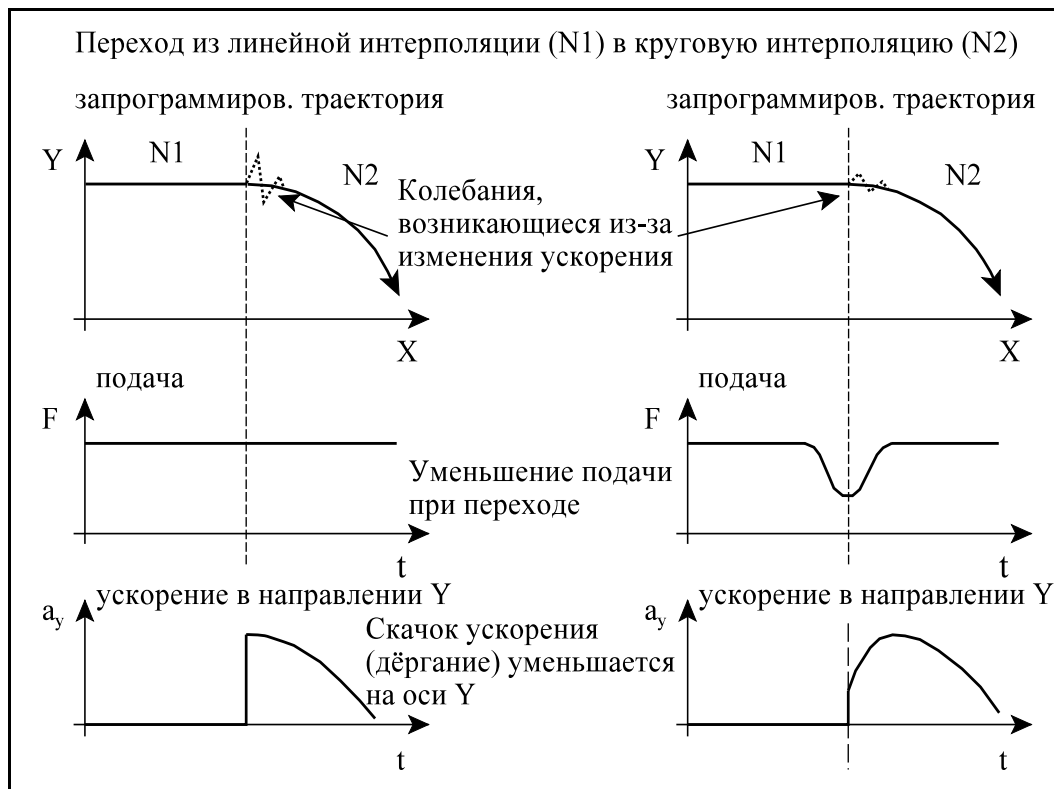


рисунок 7.3-1

В случае дуг окружности скачок в ускорении можно ограничивать параметром N0404 Acc Diff Circ.

Ограничение скачка ускорения в последующих друг за другом линейных кадрах

Если траектория состоит из длинных прямых участков, мера изменения ускорения ничтожная. В этом случае изменение составляющих подачи по осям может ограничивать подачу.

Другая ситуация, если траектория состоит из очень коротких прямых участков. В этом случае может создаться та ситуация, что изменение подачи, выходящее на отдельные оси между прямыми участками мало, из-за чего интерполятор не ограничивает подачу,

однако скачок в ускорении, выходящий на отдельные оси, высокий. В таком случае также надо ограничивать подачу в зависимости от допускаемого скачка в ускорении.

В случае последующих друг за другом прямых участков скачок в ускорении можно ограничивать параметром N0403 Acc Diff.



рисунок 7.3-2

8 Задержка (G4)

Время задержки можно запрограммировать командой

G94 G4 P....

в секундах.

Командой

G95 G4 P....

время задержки можно запрограммировать за обороты шпинделя.

Точность P представляет собой 15 десятичных цифр.

При положении бита #1 SEC=1 параметра N1337 Execution Config задержка происходит всегда в секундах, даже в состоянии G95.

Задержка означает всегда запрограммированное запаздывание выполнения следующего кадра. Функция модально не сохраняется.

9 Референтная точка

Референтная точка - это есть та точка оси, где измерительная система сдаст позицию 0.

На осях, оборудованных инкрементными измерительными инструментами, эту точку надо находить. Этот процесс называется приёмом референтной точки.

При наличии референтной точки можно замеры системы координат заготовки, и встать в абсолютную позицию. Только после приёма референтной точки действуют конечные положения с параметром и запрограммированное ограничение рабочей зоны.

Устройство ЧПУ берёт на учёт позиции не по сравнению референтной точки, а в станочной системе координат.

Нулевая точка **в станочной системе координат** определяется строителем станка, она является отличительной точкой на инструментальном станке.

В станочной системе координат устройства ЧПУ состоит на учёте положения смены, точки вращения вращающихся осей и т.д. Также в станочной системе координат состоят на учёте все компенсации станочной измерительной системы (по шагу резьбы, прямолинейности и т.д.).

В станочной системе координат устройства ЧПУ состоит на учёте **и положение референтной точки**, установленное строителем станка параметром.

Если ось оборудована **инкрементной измерительной системой**, после включения надо принять референтную точку. В ходе приёма референтной точки, суппорты набегает на включатель в направлении, определённом параметром, затем слезая с него находят нулевой импульс измерительной системы, и возмут на учёт наличие референтной точки. Позиция будет значением, заданным параметром в станочной системе координат.

Если ось оборудована **измерительной системой, кодированной по расстоянию**, после включения надо принять референтную точку. В ходе приёма референтной точки, суппорты отправляются в направлении, определённом параметром, находят два нулевых импульса, и возмут на учёт наличие референтной точки. Позиция будет положением нулевого импульса, найденного вторым в станочной системе координат.

Если ось оборудована **абсолютной измерительной системой**, после включения нет необходимости взять на учёт референтную точку.

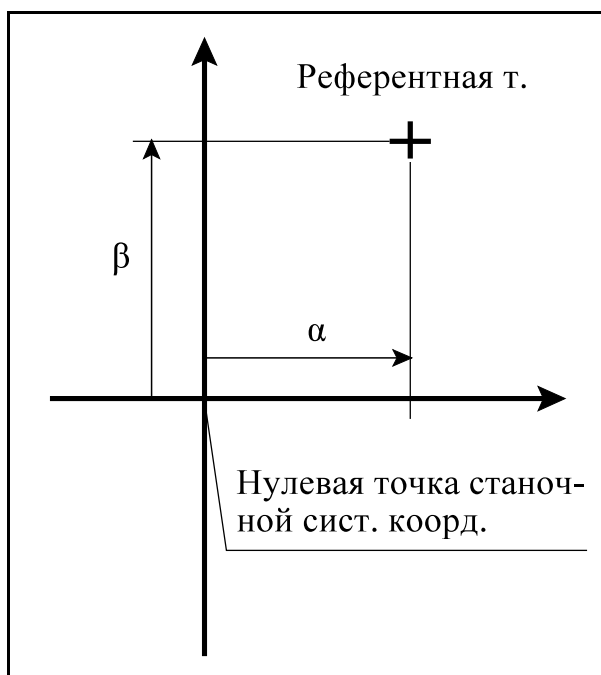


рисунок 9-1

В случае кодированной по расстоянию а также абсолютной измерительной системы референтная точка - это та точка, где измерительная система сдаст позицию 0. Эта точка обычно *не входит в рабочую зону станка*. Поэтому строитель станка эту точку сдвигает внутрь рабочей зоны с заданием параметра, например, вблизи положительного конечного положения, затем замеряет перемещённую референтную точку к началу координат станочной системы координат. Такое перемещение требуется например, при желании использовать команду G28 в программе деталей, например, для выхода инструментами.

9.1 Автоматической приём референтной точки (G28)

Команда

G28 v

совершает наводку на референтную точку на осях, определённых в векторе v. Движение состоит из двух фаз.

Первая фаза

Сначала устройство ЧПУ быстрым ходом занимает промежуточную точку, определённую вектором v, считая за *промежуточные точки* координаты, заданные в *актуальной системе координат заготовки* на осях, определённых вектором v. Заданные значения координат могут быть абсолютными, а также инкрементными значениями. Наводка на промежуточную точку происходит так, что плоскостная коррекция радиуса инструмента удаляется.

Вторая фаза

После этого о промежуточной точки одновременно берётся референтная точка на осях, определённых вектором v.

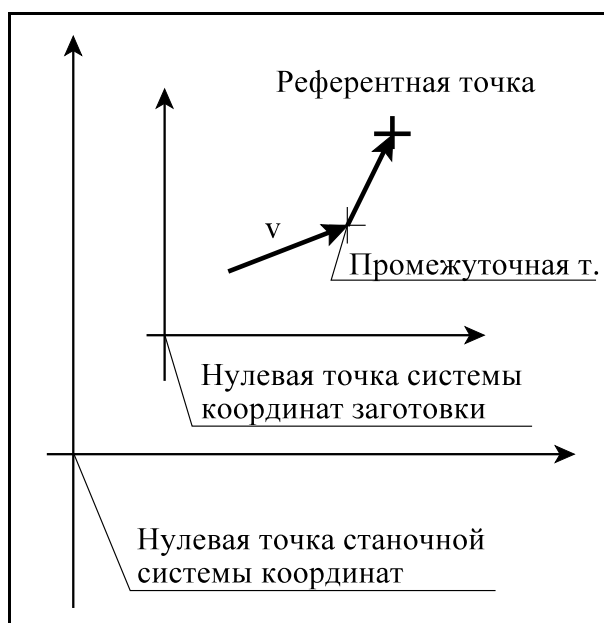


рисунок 9.1-1

Если на данных осях не произошёл приём

референтной точки, согласно порядку, определённому для ручного приёма референтной точки, берётся референтная точка. В таком случае,

- если ось оборудована *инкрементной измерительной системой*, в конце движения станочная позиция будет позицией референтной точки, определённой параметром,
- если ось оборудована *измерительной системой, кодированной по расстоянию*, в конце движения станочная позиция будет позицией второго нулевого импульса.

Если на данных осях уже произошёл приём референтной точки, или ось оборудована абсолютной измерительной системой, ось быстрым ходом занимает позицию референтной точки, заданную в станочной системе координат.

Код G28 модально не передаётся.

Например:

G90 G28 X100 Z50 (промежуточная точка: X=100, Z50)

Если позиция X: X=20, позиция Z: Z=50:

G91 G28 X100 Z50 (промежуточная точка: X=120, Z=100)

☞ **Замечания:**

– Если ещё нет действующей референтной точки, надо присвоить инкрементное значение промежуточным v координатам, имеющимся в команде G28.

9.2 Наводка на референтную точку 2., 3., 4. (G30)

В станочной системе координат параметром можно выделить 3 дальнейших знаменитых точек, которые **назовём референтной точкой 2., 3., 4.**

Эти референтные точки используем на станке в положениях смены, например, для сохранения места смены инструмента, места смены поддона и т.д.

На эти места смены можно перемещаться только после приёма референтной точки.

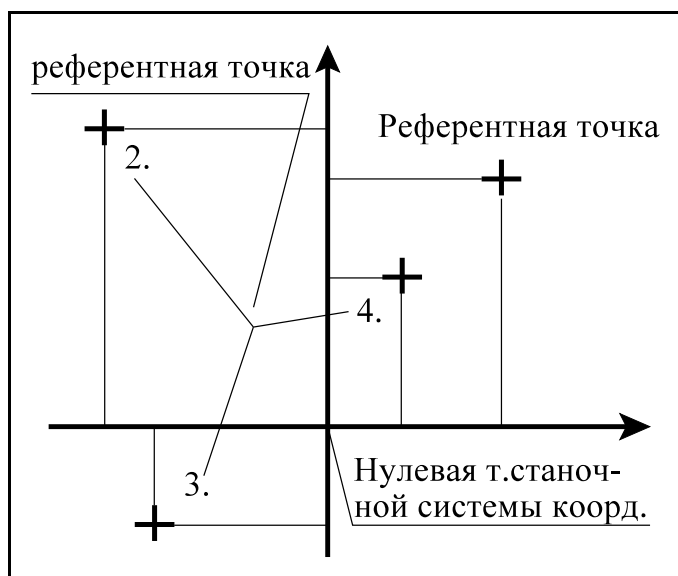


рисунок 9.2-1

Ряд команд

G30 v P

отправляет оси с координатами, определёнными по адресам вектора v на референтную точку, определённую по адресу P.

P2: 2-я референтная точка

P3: 3-я референтная точка

P4: 4-я референтная точка

Движение состоит из двух частей, подобно команде G28.

Сначала устройство ЧПУ быстрым ходом, линейным перемещением **занимает промежуточные координаты**, определённые вектором v , считая за промежуточными точками координаты, определённые вектором v . Заданные значения координат могут быть абсолютными, далее инкрементными значениями. Перемещение происходит всегда в актуальной системе координат. Наводка на конечную точку линейного движения происходит так, что плоскостный вектор коррекции радиуса инструмента удаляется.

Во второй фазе оси, выделённые вектором v , с промежуточной точки быстрым ходом **занимают референтную точку, выбранную по адресу P.**

Наводка на референтную точку выполняется пренебрежением векторов коррекции (длина, смещение, 3-мерный радиус), их не надо удалить перед выдачей команды G30, однако при программировании дальнейших перемещений устройство ЧПУ задействует их. Плоскостный вектор коррекции радиуса инструмента автоматически включается обратно в первом кадре движения.

Код модально не передаётся.

Под действием команды G30 v P1 станок перемещается в референтную точку, его действие

совпадает действием команды G28.

Например:

G90 G30 X100 Z50 P3 (промежуточная точка: X=100, Z=50,
наводка на P3)

Если позиция X : X=20, позиция Y : Z=50:

G91 G30 X100 Z50 P4 (промежуточная точка: X=120, Z=100,
наводка на P4)

10 Системы координат, выбор плоскости

Позиция, в которую желаем перемещать инструмент, задаётся в программе данными координат. Если имеется 2 оси (X, Z), позиция инструмента выражается двумя данными координат X____ Z____.

Сколько осей имеется на станке, столько разные данные координат выражает позицию инструмента. Данные координат истолкуются всегда в данной системе координат.

Устройство ЧПУ различает системы координат трёх видов:

1. станочную систему координат,
2. систему координат заготовки,
3. локальную систему координат.

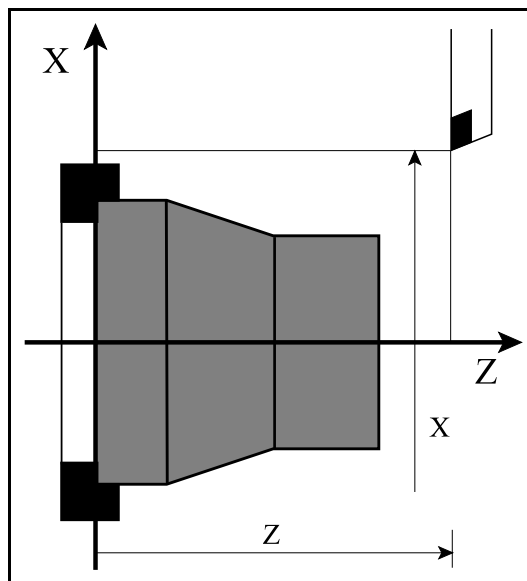


рисунок 10-1

10.1 Станочная система координат

Устройство ЧПУ берёт на учёт позиции не по сравнению референтной точки, а в станочной системе координат.

Нулевая точка в станочной системе координат определяется строителем станка, она является отличительной точкой на инструментальном станке.

В станочной системе координат устройства ЧПУ состоят на учёте положения смены, точки вращения вращающихся осей и т.д. Также в станочной системе координат состоят на учёте все компенсации станочной измерительной системы (по шагу резьбы, прямолинейности и т.д.).

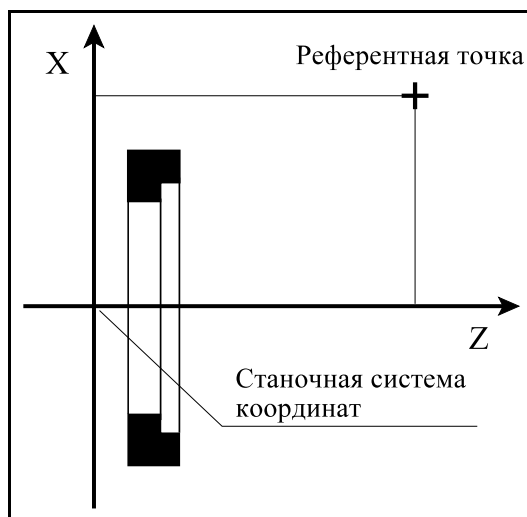


рисунок 10.1-1

В станочной системе координат устройства ЧПУ состоит на учёте **и положение референтной точки**, установленное строителем станка параметром.

Положение станочной системы координат нельзя изменить никакой командой, никаким смещением.

10.1.1 Позиционирование в станочной системе координат (G53)

Под действием команды

G53 v

инструмент перемещается в точку с позицией v станочной системы координат.

- независимо от состояния G90, G91, координатами v обращается **всегда как абсолютными** координатами,
- оператор I выдаёт сообщение 2097 Нелегальное инкрементное перемещение на оси после адреса координат, или в случае использования инкрементной задачи адреса U, V, W,
- перемещение выполняется всегда **быстрым ходом**, подобно команде G0,
- позиционирование выполняется всегда пренебрежением установленных коррекций инструмента (длина, радиус).

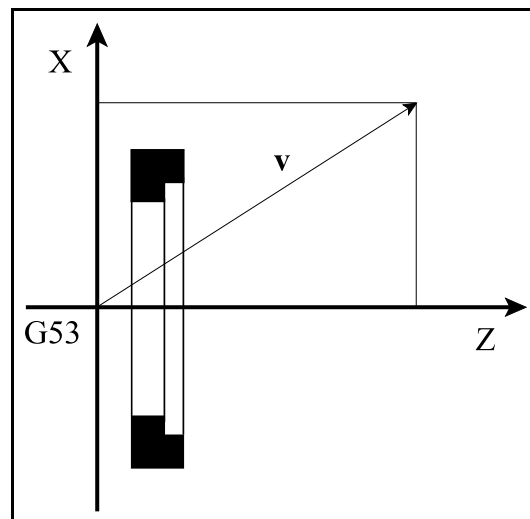


рисунок 10.1.1-2

Команду G53 можно выполнить только после

приёма референтной точки. Команда G53 является **одноразовой**, она действует только в том кадре, где её задавали.

Пример: Под действием команды

G53 X200 Z20

происходит перемещение в заданную точку в станочной системе координат.

☞ **Внимание!** Команда G53 отложит чтение вперёд кадров (буферное действие). Поэтому можно её использовать самостоятельно стоящую команду G53, и без задания координат, для отложения чтения кадров вперёд!

10.2 Системы координат заготовки

Та система координат, в которой пишем программу деталей, называется системой координат заготовки. Начало координат системы координат заготовки состоит на учёте устройством ЧПУ применительно к станочной системе координат.

Начало координат системы координат заготовки устанавливается к соответствующей точке заготовки. Этой точкой является при точении в направлении X ось вращения заготовки, в направлении Z - например, опорная поверхность патрона, или передок заготовки, при фрезеровании например, один из углов заготовки, одно из отверстий, или точка центра плеча и т.д. Установка может быть выполнена:

внутри станка ручным замером или щупом, или
вне станка.

В последнем случае значение, измеренное вне, надо вводить в память устройства ЧПУ. Это можно выполнить ручным вводом данных, или из программы, использованием команд NC.

Связь между системой координат заготовки и каналами

Смещения системы координат заготовки относятся к отдельным осям. Поскольку каждая ось выделена параметром к одному данному каналу, поэтому к каждому каналу принадлежит отдельная таблица смещения заготовки.

Если между двумя каналами поменять одну или несколько осей, оси берут собой смещения нулевой точки в новый канал. В таких случаях после смены осей целесообразно вызвать новую систему координат заготовки вместе с абсолютным позиционированием и после этого продолжать обработку.

10.2.1 Выбор системы координат заготовки (G54...G59)

В основном исполнении состоит на учёте устройством ЧПУ 6 разных систем координат заготовки. **Смещения** системы координат заготовки *по сравнению к началу координат станочной системы координат* надо задавать по осям.

Если нет общего смещения нулевой точки:

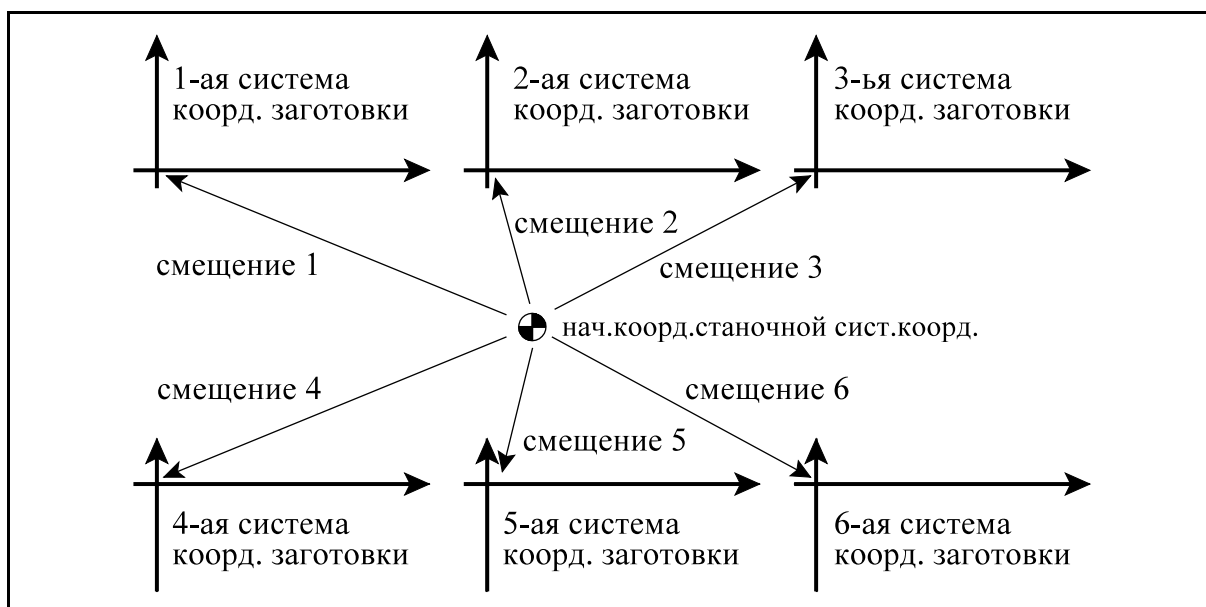


рисунок 10.2.1-1

По сравнению к началу координат станочной системы координат можно сместить все системы координат заготовки. **Общее смещение нулевой точки смещает начало координат всех систем координат заготовки** по сравнению станочной системы координат.

Если имеется общее смещение нулевой точки:

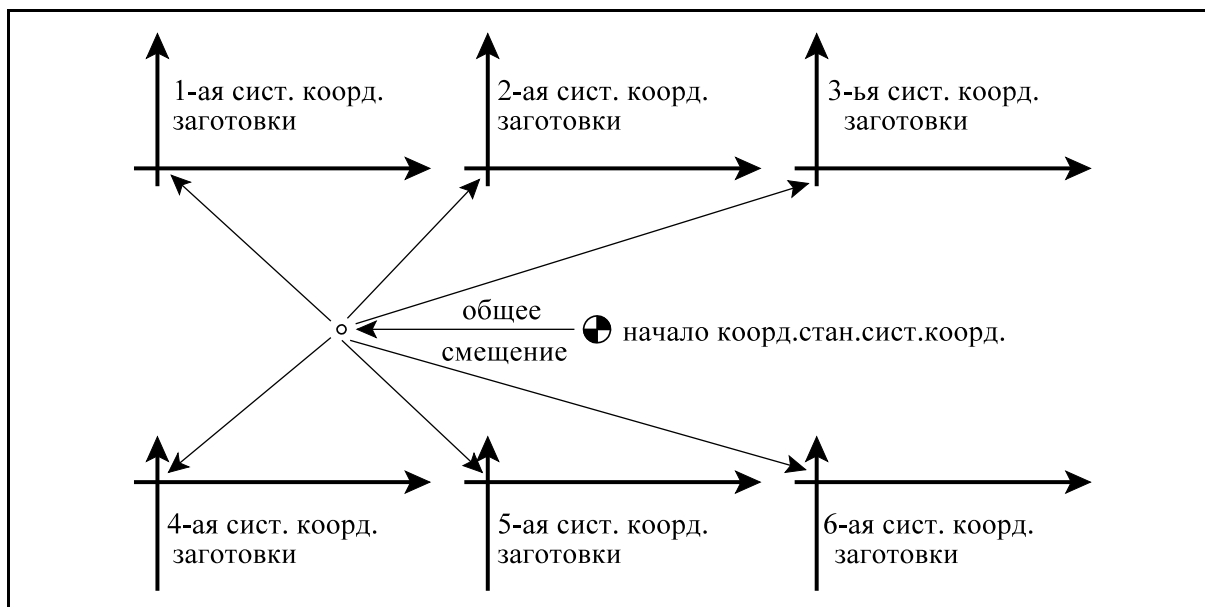


рисунок 10.2.1-2

Командами G54...G59 можно выбирать разные системы координат заготовки.

G54: 1-ая система координат заготовки

G55: 2-ая система координат заготовки

G56: 3-ая система координат заготовки

G57: 4-ая система координат заготовки

G58: 5-ая система координат заготовки

G59: 6-ая система координат заготовки

Модально передающиеся функции.

После включения, приёма референтной точки, reset, или конца программы выбирается G54-ая система координат.

Устройство ЧПУ принимает во внимание абсолютные данные координат кадров интерполяции в актуальной системе координат заготовки.

Например, в случае команды

G56 G90 G00 X80 Z60

3-ая система координат заготовки настроится на точку

X=80, Z=60.



рисунок 10.2.1-3

В случае смены системы координат заготовки позиция инструмента выводится в новой системе координат. Например, согласно рисунку, **смещение** системы координат заготовки **G54** в станочной системе координат

$$X=260 \ Z=80$$

Смещение G55 системы координат заготовки в станочной системе координат

$$X=140, \ Z=180.$$

В системе координат **G54** X', Z' **позиция инструмента**

$$X'=140, \ Z'=90.$$

Под действием команды **G55** **позиция инструмента** истолкуется в системе координат X'', Z'' :

$$X''=260, \ Z''=-50.$$

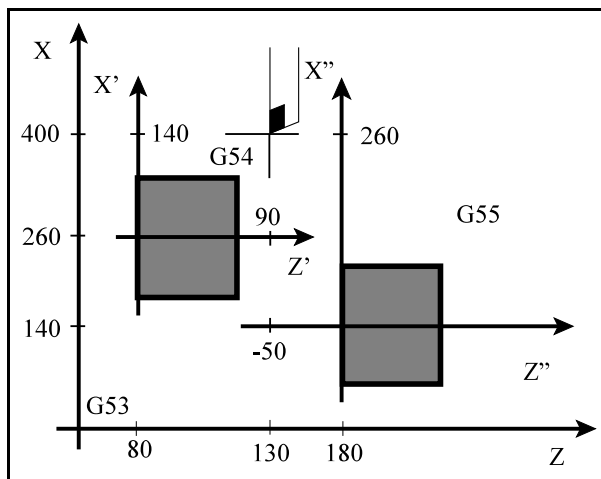


рисунок 10.2.1-4

10.2.2 Выбор расширенных систем координат заготовки (G54.1 P)

Опционально в устройстве ЧПУ можно использовать дальнейшие 99 штук системы координат заготовки. Эти системы координат являются расширением 6 штук основных систем координат, поэтому называются они расширенными системами координат заготовки.

Общее смещение нулевой точки смещает и расширенные системы координат. Расширенные системы координат заготовки **также можно поворачивать**, как основные системы координат.

Командой

G54.1 Pr

можно выбирать расширенные системы координат заготовки, где по адресу P можно задавать номер расширенные системы координат:

$$P = 1, 2, \dots, 99$$

Модально передающаяся функция.

☞ **Внимание!** Адрес P можно использовать для многих целей. Надо следить за тем, чтобы использование адреса P в кадре было однозначное:

G0 G54.1 P16 X100 Z20 M98 P1 (ОШИБКА! Два P в одном кадре)

G0 G54.1 P16 X100 Z20 (Однозначно)

10.2.3 Компенсация углового положения заготовки

Компенсация углового положения заготовки требуется тогда, когда заготовку нельзя установить в направлении, параллельное с главными осями. В дальнейшем эту компенсацию назовём **основным поворотом**.

Истолкование основного поворота

В общей таблице смещения нулевой точки, приведенной ниже, значение столбцов следующее:

X, Y, Z: смещение нулевой точки на 3-х главных осях ,

α, β, γ: углы основного поворота, в плоскости G19 - α, в плоскости G18 - β, в

плоскости G17 - γ .

	X	Y	Z	G17(γ)	G18(β)	G19(α)
G54						
...						
G54.1 P						
...						

Повороты выполняются всегда вокруг главных осей. Порядок поворотов:

1. поворот: вокруг оси X на угол α ,
2. поворот: вокруг оси Y на угол β ,
3. поворот: вокруг оси Z на угол γ .

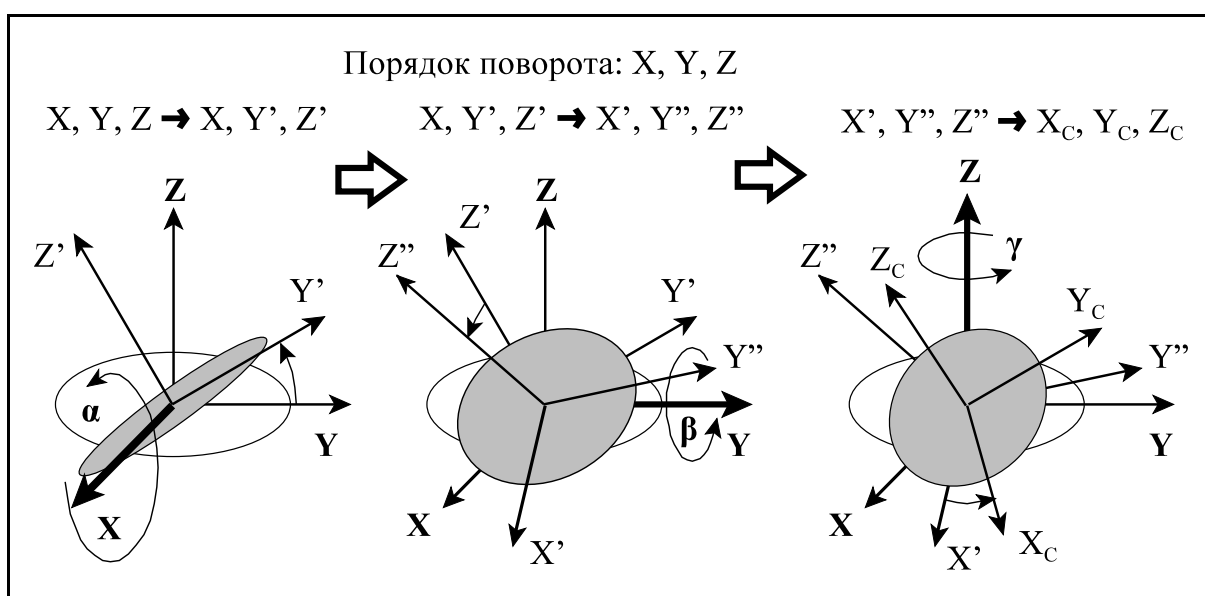


рисунок [10.2.3-1](#)

Направление α , β , γ является всегда направлением трансформаций поворот, вокруг соответствующей оси, согласно правилу правокручения.

Трансформации основного поворота **принимается во внимание всегда в актуальной системе координат заготовки, в случае команд о перемещениях**

К командам о перемещений в станочную позицию (G53, G28, G30) никогда не относится.

В случае ручных перемещений принимается во внимание включатель “Согласно основному повороту” (указатель CP_WMCAXF PLC), то есть применяется поворот вокруг всех трёх осей.

Если поворот на запрограммированную окончательную точку $v[x, y, z]$, происходит на

основании уравнения

$$\mathbf{v}' = \mathbf{M}_Z(\gamma) * \mathbf{M}_Y(\beta) * \mathbf{M}_X(\alpha) * \mathbf{v}$$

где:

$\mathbf{v}'[x', y', z']$ повернутые координаты и

$\mathbf{M}_X(\alpha)$, $\mathbf{M}_Y(\beta)$, $\mathbf{M}_Z(\gamma)$ матрицы поворотов вокруг осей X, Y, Z:

$$\mathbf{M}_X = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha \\ 0 & \sin \alpha & \cos \alpha \end{bmatrix} \quad \mathbf{M}_Y = \begin{bmatrix} \cos \beta & 0 & \sin \beta \\ 0 & 1 & 0 \\ -\sin \beta & 0 & \cos \beta \end{bmatrix} \quad \mathbf{M}_Z = \begin{bmatrix} \cos \gamma & -\sin \gamma & 0 \\ \sin \gamma & \cos \gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Применение основного поворота на токарном станке

Приведенный ниже рисунок показывает пример для применения основного поворота. Контршпиндель S2, имеющийся в револьверной головке T1, достигает заднюю группу инструментов T2 таким образом, что ось поворота S2 образует угол β с осью Z. В этом случае в столбце G19 β можно установить основной поворот. В дальнейшем программу можно написать в системе координат $Z'X'$.

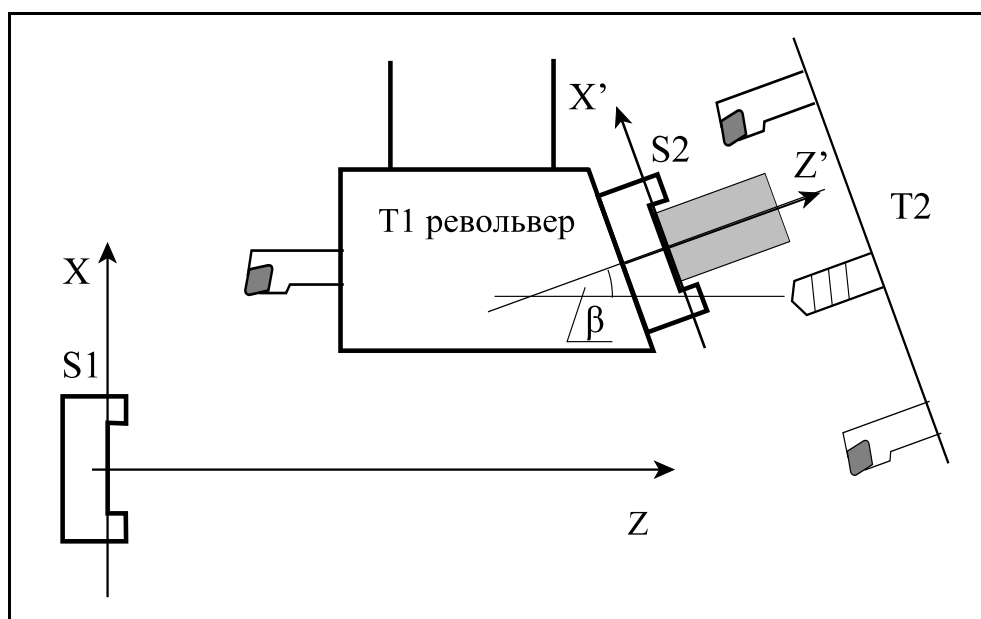


рисунок 10.2.3-2

10.2.4 Установка смещения системы координат заготовки (G10 L2)

Смещение, поворачивание, и общего смещения нулевой точки систем координат заготовки можно установить и командами программы.

Установка выполняется командой

G10 L2 P v I J K

где

P = 0 установка общего смещения нулевой точки
P = 1...6 выбор системы координат заготовки G54, ..., G59
v (X, Y, Z, ...): значения смещения по осям

Смещения осей зачитаются всегда прямоугольными значениями. Данные длины в мм-ах,

или в дюймах, данные угла в градусах.

I: γ угол поворота в плоскости G17
 J: β угол поворота в плоскости G18
 K: α угол поворота в плоскости G19

Угол поворота надо задавать в градусах.

На общее смещение нулевой точки нельзя задавать поворачивание.

Команда G10 является одноразовой.

G90 в состоянии команды абсолютного задания данных попадает в соответствующие регистры смещения значение, написанное на адреса координат и на адрес I, J, K.

G91 в состоянии команды инкрементного задания данных, или в случае использования оператора I данные, написанные на адреса, добавляются к содержанию соответствующего регистра смещения. Оператор I можно использовать только на адреса координат, на адрес I, J, K нельзя.

10.2.5 Установка смещения расширенных систем координат заготовки (G10 L20)

Смещение и поворачивание расширенных систем координат заготовки можно устанавливать и командами программ.

Установка выполняется командой

G10 L20 P v I J K

где

P = 1...n выбор системы координат заготовки G54.1 P1, G54.1 P2, ..., G54.1 Pn

v (X, Y, Z, ...): значения смещения по осям

Смещения осей зачитаются всегда прямоугольными значениями. Данные длины в мм-ах, или в дюймах, данные угла в градусах.

I: γ угол поворота в плоскости G17
 J: β угол поворота в плоскости G18
 K: α угол поворота в плоскости G19

Угол поворота надо задавать в градусах.

Команда G10 является одноразовой.

G90 в состоянии команды абсолютного задания данных попадает в соответствующие регистры смещения значение, написанное на адреса координат и на адрес I, J, K.

G91 в состоянии команды инкрементного задания данных, или в случае использования оператора I данные, написанные на адреса, добавляются к содержанию соответствующего регистра смещения. Оператор I можно использовать только на адреса координат, на адрес I, J, K нельзя.

10.2.6 Создание новой системы координат заготовки (G92)

Под действием команды

G92 v

можно создать новую систему координат заготовки так, чтобы одна выделённая точка, например, вершина инструмента была точкой новой системы координат заготовки с координатой v. После этого любая следующая абсолютная команда истолкуется в этой новой системе координат заготовки, и отображение позиции формируется также в этой системе координат. Координаты, заданные в команде G92 всегда истолкуются как прямоугольные абсолютные значения.

Если например, инструмент находится в точке с координатами

$$X=200, Z=150$$

в актуальной системе координат заготовки X, Z, под действием команды

G92 X120 Z90

создаётся новая система координат X', Y', в которой инструмент попадает в точку с координатами

$$X'=120, Z'=90.$$

Осевые составляющие вектора смещения v' между системами координат X, Z-X', Z':

$$v'_x = 200 - 80 = 120, \text{ а также}$$

$$v'_z = 150 - 90 = 60$$

будут.

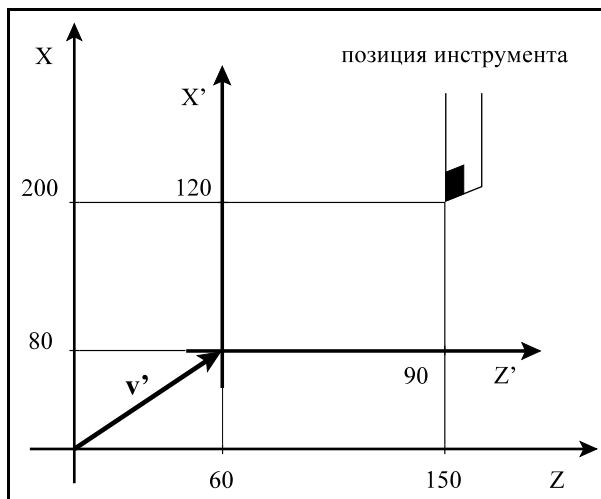


рисунок 10.2.6-3

Команда G92 действует в каждой системе координат заготовки, то есть рассчитанной в одной системе смещение v учитывается и в других.

Установленная командой G92 смещение системы координат заготовки удаляется при включении, в конце программы, и при reset.

Команда **G92** *удаляет вектор коррекции радиуса инструмента*, смещение отсчитывается всегда с позиции вершины инструмента.

Команда **G92** на тех осях, которые фигурируют в команде, *удаляет смещения*, запрограммированные командой **G52** локальной системы координат.

В движении вектор смещение, заданный в команде G92, принимает во внимание поворачивание действующей системы координат заготовки также повернуто.

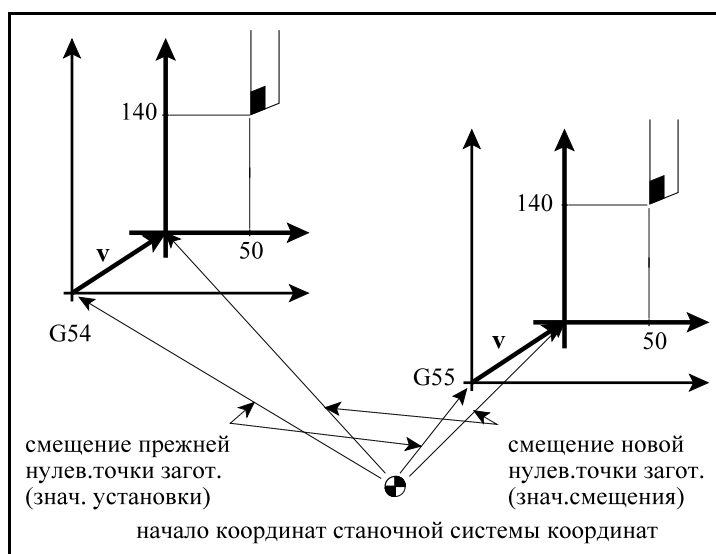


рисунок 10.2.6-4

10.3 Локальная система координат (G52)

В некоторых случаях во время написания программы деталей, легче задавать данные координат в так называемой локальной системе координат, чем в системе координат заготовки. Команда

G52 v

создаёт локальную систему координат.

Если координата v задана **абсолютным значением**, начало координат локальной системы координат попадает в точку системы координат заготовки с координатой v .

Если координата v задана **инкрементным значением**, это смещает начало координат локальной системы координат на v .

Начиная с этого, каждая команда движения, заданная абсолютными координатами, выпол-

няется в новой системе координат. Отображение позиции также происходит в новой системе координат.

Значениями координат v устройство ЧПУ обращается всегда **прямоугольными** данными.

Команда

G90 G52 v0

удаляет смещения в точках с координатой v . При включении, в конце программы, под действием reset удаляются значения смещения, установленные с помощью G52.

Если инструмент находится в точке с координатой

$X=200, Z=150$

в актуальной системе координат заготовки X, Z , под действием команды

G90 G52 X80 Z60

создаётся новая локальная система координат X', Z' , в которой инструмент попадает в точку с координатой

$X'=120, Z'=90$.

Осевые составляющие вектора смещения v' между системами координат X, Z - X', Z' определяются командой G52: $v'_x=80$, и также $v'_z=60$.

При желании создать новую систему координат X'', Z'' , поступить можно двояко:

Абсолютным заданием данных: команда

G90 G52 X120 Z30

в системе координат заготовки X, Z наводит начало координат системы координат X'', Z'' в точку с координатой $X=120, Z=30$. Составляющие вектора v'' создаются заданием значения $v''_x=120, v''_z=30$.

Инкрементным заданием данных: команда

G91 G52 X40 Z-30

смещает начало координат системы координат X', Z' на значение $X'=40, Z'=-30$. Составляющие вектора v создаются заданием $v_x=40, v_z=-30$. Вектор v'' , показывающий положение новой локальной системы координат, в системе координат заготовки X, Z : $v''=v'+v$. Его составляющие: $v''_x=80+40=120, v''_z=60+(-30)=30$.

Позиция инструмента в системе координат X'', Z'' : $X''=80, Z''=120$.

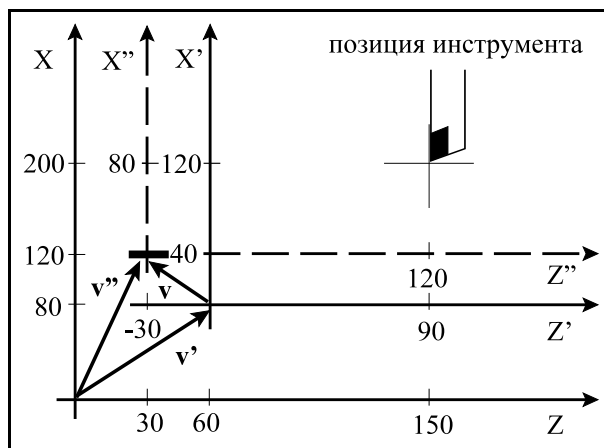


рисунок 10.3-1

Смещение локальной системы координат действует во всех системах координат заготовки.

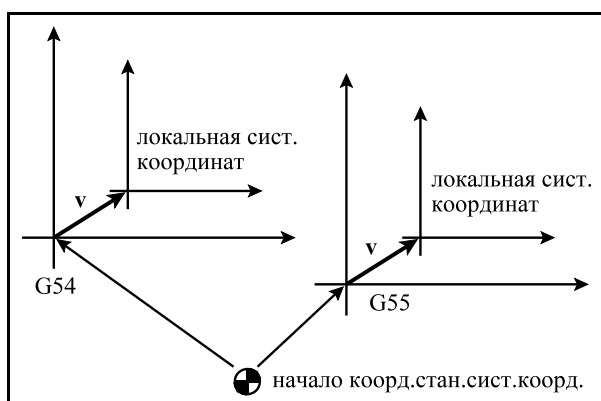


рисунок 10.3-2

Запрограммирование команды **G92** на тех осях, которым задавали значение, *удаляют* смещения, созданные командой **G52**, как будто выдавали команду G52 v0.

Если инструмент находится в точке с координатой $X=240$, $Z=200$ системы координат заготовки X , Z , под действием команды
G52 X80 Z60

в локальной системе координат X' , Z' его позиция будет $X'=160$, $Z'=140$. После этого под действием команды

G92 X80 Z110

в новой системе координат заготовки X'' , Z'' позиция инструмента будет $X''=80$, $Z''=110$. Значит, локальная система координат X' , Z' под действием команды G92 удаляется, как будто выдавали команду G52 X0 Z0.

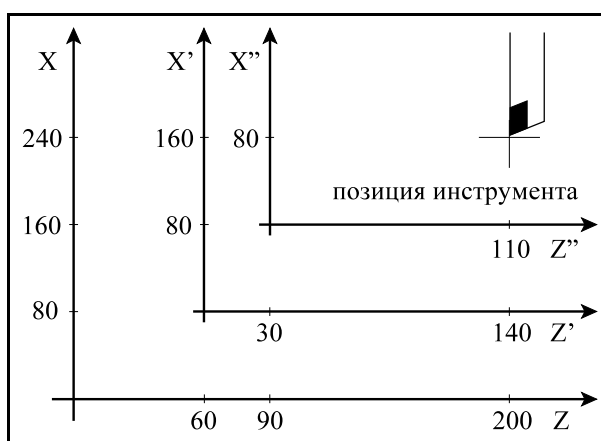


рисунок 10.3-3

10.4 Выбор плоскости (G17, G18, G19)

Плоскость, в которой выполняются

- круговая интерполяция,
- задание данных полярными координатами,
- плоскостное поворачивание системы координат,
- плоскостная коррекция радиуса инструмента,
- позиционирования циклов сверления,
- циклы точения,

можно выбрать следующими кодами G:

G17 плоскость $X_p Y_p$

G18 плоскость $Z_p X_p$

G19 плоскость $Y_p Z_p$,

где: X_p : X , или параллельная ей ось,
 Y_p : Y , или параллельная ей ось,
 Z_p : Z , или параллельная ей ось.

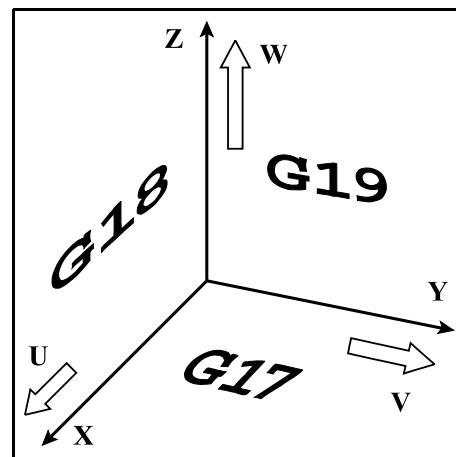


рисунок 10.4-1

Выбранная плоскость называется главной плоскостью.

От адресов осей, запрограммированных в одном кадре зависит, что из параллельных осей какая будет выбрана командой G17, G18, или G19:

Если например, X и U , Y далее V , Z W параллельные оси:

G17 X ___ Y ___ плоскость XY ,

G17 X ___ V ___ плоскость XV ,

G17 U ___ V ___ плоскость UV ,

G18 X ___ W ___ плоскость XW ,

G19 Y ___ Z ___ плоскость YZ ,

G19 V ___ Z ___ плоскость VZ ,

будет выбирать.

Если не задана в одном кадре G17, G18, G19, выбор плоскости остаётся неизменным:

G17 X ___ Y ___ плоскость XY

U ___ Y ___ плоскость XY остаётся.

Если в кадре G17, G18, G19 не задан адрес оси, тогда устройство ЧПУ выбирает главные оси:

G17 плоскость XY ,

G17 X плоскость XY ,

G17 U плоскость UY ,

G17 V плоскость XV ,

G18 плоскость ZX ,

G18 W плоскость WX

будет выбирать.

Команда движения не влияет на выбор плоскости:

под действием

G90 G17 G00 Z100

выбирается плоскость XY , и ось Z перемещается в точку с координатой 100.

Внутри одной программы можно несколько раз сменить плоскость.

После включения, в конце программы, или после reset решается на основании битов #1 G18 и #2 G19 N1300 DefaultG1 параметра #1 G18 és #2 G19, что какая плоскость будет действовать.

Параметром N0103 Axis to Plane можно выделить, что какими адресами осей будет пользоваться устройство ЧПУ в качестве главных и параллельных осей.

Описание часто ссылается на первую и вторую ось выбранной плоскости. Истолкование этого видно на рисунке ниже.

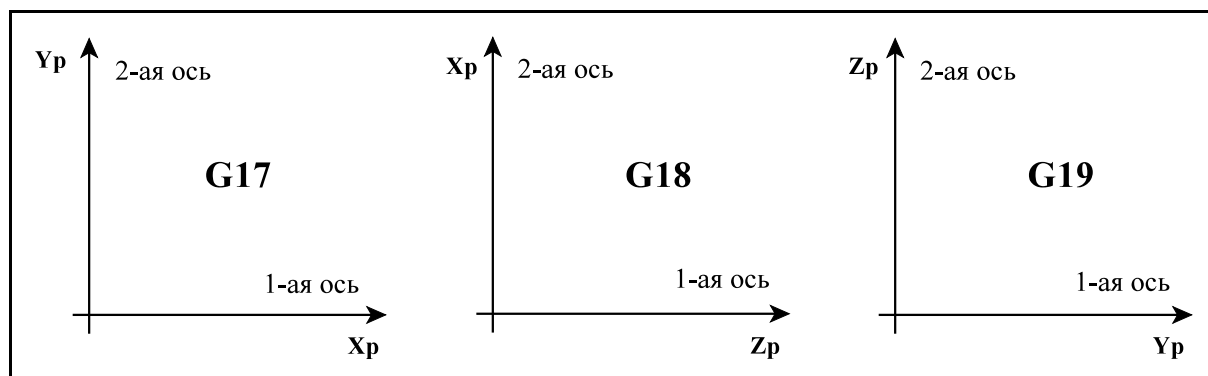


рисунок [10.4-2](#)

11 Шпиндельные функции

11.1 Команда чисел оборотов шпинделя (код S)

На адрес

S nnnnnnnn

написав не более 8-мизначное число, NC передаёт код для PLC.

Адрес S в зависимости от построения данного инструментального станка, PLC может истолковать кодом, или значением в размерности число оборотов/мин.

Если запрограммировать команду движения и чисел оборотов шпинделя (S) в одном и том же кадре, функция S выполняется во время выполнения команды движения, или после неё. Каким образом выполняется, определяется строителем станка.

Значения чисел оборотов, заданные по адресу S, передаются модально. После включения устройство ЧПУ построится кодом S0.

В различных диапазонах передачи шпинделя число оборотов шпинделя имеет минимальный и максимальный предел. Эти пределы определяются строителем станка в поле параметров, и устройство ЧПУ не допускает число оборотов вне этого диапазона.

Устройство ЧПУ может обращаться не более 8 диапазонами чисел оборотов.

11.1.1 Ссылка на несколько шпинделей. Расширение адреса S

Устройство ЧПУ может обращаться не более 16 шпинделями.

Если на станке, или в канале имеется несколько шпинделей, адрес S не достаточно для различения шпинделей. Устройство ЧПУ обеспечивает две возможности для обращения несколькими шпинделями.

Ссылка на шпиндель по адресу S и P

Первой возможностью является задавать по адресу P номер шпинделя, рядом с адресом S. Под действием команды

Snnnnnnnn Pp

устройство ЧПУ передаёт код S и номер шпинделя, написанный по адресу P для PLC.

Язык программирования использует адрес P для различных целей, например, для задержки, для вызванных подпрограмм и т.д. Поэтому ссылки на шпиндель по адресу P надо написать в отдельных кадрах, иначе истолкование адреса P не будет однозначное.

Ссылка на несколько шпинделей расширением адреса S

Второй возможностью является расширение адреса шпинделей. На шпиндели можно ссылаться и заданием не более 3 символов.

Адрес шпинделей должны начинаться всегда буквой S. Параметром N0605 Spindle Name2 и N0606 Spindle Name3 можно задавать два дальнейших символов, они могут быть буквами английского алфавита: A, B, C, D, ... Y, Z, далее цифрами: 0, 1, 2, ..., 9. Если не использовать 2-ое, или 3-ье название шпинделя, значение параметра 0.

Таким образом можно задавать например, название шпинделя SSB, но можно использовать и имена S1 и S2.

Если название шпинделя заканчивается буквой, рядом с ним можно написать принадлежащее к нему значение.

SSB12500

означает: пусть шпиндель SSB вращается числом оборотов 12500/мин.

Если название шпинделя заканчивается числом, после имени надо писать всегда знак =
S1=8700

означает: пусть шпиндель S1 вращается числом оборотов 8700/мин.

В случае ссылки на расширенный адрес шпинделя, устройство ЧПУ передаёт для PLC значение запрограммированного числа оборотов и номер ссылочного шпинделя.

В программе *внутри одного кадра можно ссылаться только на один шпиндель*. Если надо запускать несколько шпинделей, команды надо написать в отдельных кадрах:

S1=500 M3 (S1 500/мин, по ходу часовой стрелки)

S2=1000 M4 (S2 1000/мин, против хода часовой стрелки)

NC ведёт учёт шпинделей всегда на основании их номера. *Нумерация и название также глобальные*, независимые от канала.

11.1.2 Назначение шпинделей к каналам

Отдельные *шпиндели назначает* всегда *программа PLC program* к отдельным *каналам*. Назначение означает, что для данного шпинделя можно выдавать команду чисел оборотов только из программы, работающей в данном канале. Например, если шпиндель S4 назначен к каналу 2, в канале 1 нельзя запрограммировать адрес S4.

Во время прогона программы, программа PLC имеет возможность перенести данный шпиндель в другой канал, например, под действием функции M.

Назначение шпинделей к каналам и перенос его в другой канал определяется всегда строителем инструментального станка.

В каждом канале можно выделить один основной шпиндель параметром N0604 Default Spindle. На назначенный таким образом шпиндель можно всегда ссылаться и по адресу S, даже тогда, если его имя состоит из нескольких символов. Например, пусть будет S2 шпинделем с номером 2. Если во 2-ом канале значение N0604 Default Spindle=2, тогда во 2-ом канале можно ссылаться на шпиндель по адресу и S2 и S.

11.2 Функции M управления шпинделем

Встроенные функции M

Устройство ЧПУ обращается шпинделями со следующими встроенными кодами M:

M3: шпиндель вкл., по ходу часовой стрелки

M4: шпиндель вкл., против хода часовой стрелки

M5: шпиндель выкл. (стой)

M19: ориентировка

Направление понимается всегда со стороны двигателя смотря в сторону шпинделя. Вместо M3, M4, M5 можно писать и M03, M04, M05.

Эти являются встроенными кодами M для управления шпинделем потому, что в ходе выполнения циклов сверления устройство ЧПУ передаёт указанные выше коды для PLC для останова шпинделя, изменения направления, ориентировки шпинделя.

Опциональные коды M

Помимо вышеуказанных, встроенных кодов M, можно выделить для обращения шпинделем и дальнейшие коды M. Эти коды можно установить в одном блоке параметрами N0689 Spindle M Low и N0690 Spindle M High. На параметр Spindle M Low пишем наименьшее

значение блока кода M, а на параметр Spindle M high - наибольшее значение блока. Например, пусть будет Spindle M Low S2=20, а Spindle M High S2=24. Пусть будет функция кодов M следующая:

M20: преобразование шпинделя в ось C

M21: синхронизация шпинделя

M22: синхронизация шпинделя со сдвигом по фазе

M23: подготовка шпинделя к точению многоугольника

M24: замыкание петля шпинделя без ориентировки (M Code for Closing S Loop)

☞ **Внимание!** Указанные выше блока кодов M только один из примеров. Опциональные функции M шпинделей определяются всегда строителем станка, поэтому их описание смотрите в инструкции данного станка!

Встроенные и опциональные функции M, выделённые параметром, устройство ЧПУ учитывает их функциями, исключаящими друг друга. Это означает, что в один кадр можно написать только один такой код M.

Функции M, управляющие шпинделем, относятся всегда к последнему запрограммированному шпинделю:

S1=1500 M4 (S1 вкл. в направление M4)

S2=2000 M3 (S2 вкл. в направление M3)

M5 (S2 стой)

M19 (S2 ориентировка)

S1=0 M5 (S1 стой)

В примере выше видно, если уже ссылались на шпиндель S2, то дальнейшие коды M, управляющие шпинделем относятся к шпинделю S2. Однако, при желании остановить шпиндель S1, надо ссылаться на адрес шпинделя!

11.3 Обращение диапазонами чисел оборотов

Между двигателем шпинделя и шпинделем может быть *передача с изменяемой ступенями*, с помощью которой *можно изменять число оборотов шпинделя*. Устройство ЧПУ одновременно может обращаться не более 8 диапазонами чисел оборотов по шпинделям. Чем ниже диапазон чисел оборотов, где шпиндель находится, тем больше моментом может точить шпиндель.

К каждому диапазону передач можно установить допускаемое минимальное и максимальное число оборотов, ниже или выше которого устройство ЧПУ не допускает число оборотов шпинделя.

Допускаемые числа оборотов могут перекрывать друг друга между отдельными ступенями.

Если диапазоны чисел оборотов не перекрывают друг друга

Например:

минимальное число оборотов 1-го диапазона: 50/мин

максимальное число оборотов 1-го диапазона: 1000/мин

минимальное число оборотов 2-го диапазона: 1001/мин

максимальное число оборотов 2-го диапазона: 4000/мин

На основании запрограммирования кода S900 в перечисленном выше случае, неоднозначно, что шпиндель надо вращать в 1-ом диапазоне.

Если диапазоны не перекрывают друг друга, устройство ЧПУ передаёт для PLC вместе с значением *адреса S* и номером ссылочного шпинделя и код диапазона и PLC *автомати-*

чески включает нужный диапазон.

Если диапазоны чисел оборотов перекрывают друг друга

Например:

минимальное число оборотов 1-го диапазона: 50/мин

максимальное число оборотов 1-го диапазона: 1000/мин

минимальное число оборотов 2-го диапазона: 800/мин

максимальное число оборотов 2-го диапазона: 4000/мин

На основании запрограммирования кода S900 не однозначно, что в каком диапазоне: во 1-ом или во 2-ом диапазоне надо вращать шпиндель.

Если диапазоны перекрывают друг друга, программист должен **выбирать функцией M** тот диапазон, в котором желает вращать шпиндель.

Эти встроенные функции M следующие:

M11: выбор 1-го диапазона

M12: выбор 2-го диапазона

...

M18: выбор 8-го диапазона

Обращение диапазонами на данном станке определяется строителем станка, описание которого смотрите в инструкции данного станка.

11.4 Главный шпиндель. Выбор главного шпинделя

Если на станке, или в канале имеется несколько шпинделей, надо решить, что какой из шпинделей будет “главным шпинделем”. К шпинделю, выделённому **главным шпинделем** относятся следующие функции:

разрешение подачи,

подача за оборот,

расчёт постоянной скорости резания,

нарезание резьбы,

нарезание резьбы с жёстким хвостовиком,

мастер-шпиндель точения многоугольника.

В каждом канале **надо выделить главный шпиндель**. Главным шпинделем можно выделить в данном канале такой шпиндель, который к другому каналу относится, поэтому в этом канале нельзя его запрограммировать, но например, подачу за оборот надо брать с этого шпинделя.

Способ выбора главного шпинделя решает программа PLC на данном станке, и его содержит **описание производителя станка**. Это может выполнен например, функцией M:

M31 (1-й шпиндель является главным шпинделем)

M32 (2-й шпиндель является главным шпинделем)

11.5 Расчёт постоянной скорости резания

Функцию расчёта постоянной скорости резания можно использовать только в случае бесступенчатого привода шпинделя. При этом устройство ЧПУ так изменяет число оборотов шпинделя, чтобы скорость инструмента по отношению поверхности заготовки была всегда постоянной, и равной запрограммированному значению.

Устройство ЧПУ обращается всегда числом оборотов шпинделя, выделённого главным шпинделем.

Значение постоянной скорости резания надо задавать на основании таблицы ниже, в зависимости от входной единицы измерения:

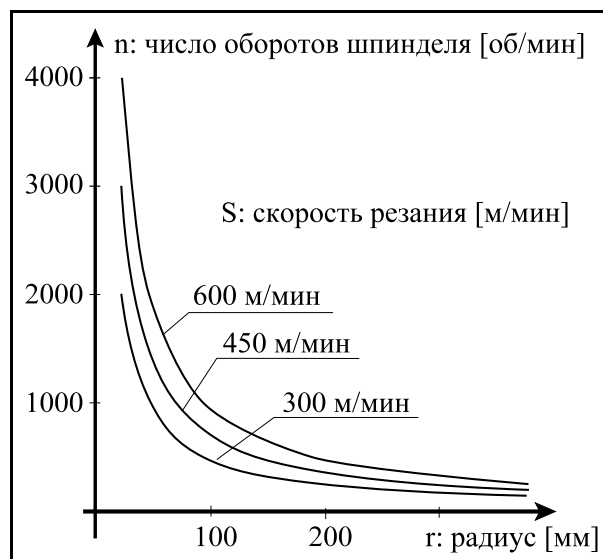


рисунок 11.5-1

входная единица измерения	единица измерения постоянной скорости резания
мм (G21 метрическая)	м/мин (метр/минута)
дюйм (G20 дюймовая)	feet/мин (фут/минута)

11.5.1 Задание расчёта постоянной скорости резания (G96 S, G97)

Команда

G96 S

включает расчёт постоянной скорости резания. По адресу S надо задавать значение постоянной скорости резания в единицах измерения, согласно таблицы выше.

Значение постоянной скорости резания надо задавать всегда по адресу S и расчёт чисел оборотов относится к выделённому главному шпинделю.

Например:

M32 (2-й шпиндель выделён главным шпинделем)

G96 S300 (300 м/мин скорость резания)

(M32 служит здесь только примером, способ выделения главным шпинделем содержится в инструкции данного инструментального станка.)

Команда

G97 S

выключает расчёт постоянной скорости резания.

По адресу S можно задавать желаемое число оборотов главного шпинделя (в размерности оборот/минута). В случае нескольких шпинделей после запрограммирования G97 надо использовать всегда адрес главного шпинделя вместо S . Например:

G97 S2=1200 (число оборотов 2-го шпинделя 1200/мин)

– Для расчёта постоянной скорости резания нулевую точку той оси, на основании позиции

которой надо изменить число оборотов главного шпинделя, надо уставить на ось вращения главного шпинделя.

– Расчёт постоянной скорости резания действует только после того, что главный шпиндель запустили с помощью M3, или M4.

– Значение постоянной скорости резания передаётся модально, даже после того, что командой G97 выключили её расчёт.

G96 S100 M3 (100м/мин, или 100 фут/мин)

G97 S1500 (1500 об/мин)

G96 X260 (100м/мин, или 100 фут/мин)

– Расчёт постоянной скорости резания действует и в режиме G94 (подача/мин).

– Если выключён расчёт постоянной скорости резания командой G97 и не задавали новые обороты главного шпинделя, то остаются действующими последние обороты главного шпинделя, принятые в состоянии G96.

G96 S100 (100м/мин, или 100 фут/мин)

.

.

.

G97 (число оборотов, относящееся к получаемому диаметру X)

– В случае позиционирования быстрым ходом (кадр G0) число оборотов шпинделя не рассчитывается непрерывно, а устройством ЧПУ устанавливается число оборотов, относящееся к позиции, положенной в конечной точке позиционирования.

– Значение постоянной скорости резания определяется после включения параметром N0686 Default Surf Speed.

11.5.2 Ограничение чисел оборотов при расчёте постоянной скорости резания (G92 S)

Командой

G92 S

можно установить максимальное число оборотов главного шпинделя, допускаемое при расчёте постоянной скорости резания. Устройство ЧПУ во включенном состоянии расчёта постоянной скорости резания не разрешает выдавать число оборотов главного шпинделя, выше заданного здесь значения. Единица измерения S в этом случае: об/мин.

Максимальное значение чисел оборотов главного шпинделя надо задаётся всегда по адресу S и ограничение чисел оборотов относится всегда к шпинделю, выделённому главным шпинделем.

– После включения, или, если не ограничивали значение чисел оборотов командой G92, верхний предел чисел оборотов главного шпинделя, в случае расчёта постоянной скорости резания служит максимальное значение, допускаемое для данного диапазона.

– В случае расчёта постоянной скорости резания, числом оборотов главного шпинделя можно задавать и нижний предел параметром N0688 Min Spindle Speed G96, которое может быть выше значения минимального числа оборотов, относящего к диапазону.

– Значение максимального числа оборотов передаётся модально до тех пор, пока не запрограммировано новое.

11.5.3 Выделение оси для расчёта постоянной скорости резания (G96 P)

Ту ось, на основании позиции которой устройство ЧПУ рассчитывает число оборотов шпинделя, в состоянии G96 выделяется параметром N0687 Default G96 Axis.

При желании отличать от выделённой оси, командой

G96 P

можно задавать ту ось, по которой желаем рассчитывать скорость резания.

Истолкование *адреса P: номер оси*.

В команде G96 можно вместе запрограммировать адрес и S и P:

G96 S300 P4 (300м/мин скорость резания с 4-й осью)

Установленное по адресу P значение передаётся модально.

11.6 Наблюдение за колебанием чисел оборотов шпинделей

Устройство ЧПУ наблюдает за колебанием чисел оборотов каждого шпинделя. Разность между запрограммированным числом оборотов, изменённым с помощью override пределами чисел оборотов и между актуальным числом оборотов, измеренным датчиком, определяет колебание.

Если число оборотов шпинделя выходит за допустимым пределом диапазона, установленного параметром строителем станка, NC выводит сообщение для PLC.

После этого PLC выводит сообщение об ошибке и принимает меры для останова шпинделя и течения. Эти содержатся в описании строителя станка.

- Функция наблюдения за колебанием чисел оборотов работает только тогда, если шпиндель оборудован датчиком.
- Наблюдения за колебанием чисел оборотов действует только при вращении шпинделя (в состоянии M3, или M4).

11.7 Позиционирование шпинделей

В случае нормальной обработке NC выдаёт команду числа оборотов, пропорционального запрограммированному числу оборотов приводов шпинделей. При этом привод шпинделя работает в режиме регулируемого числа оборотов.

Некоторые технологические задачи требуют, чтобы шпиндель установили в определённое угловое положение. Это называется позиционированием, или индексацией шпинделя. Перед позиционированием NC включает шпиндель в режим регулируемой позиции. Это практически означает, что в дальнейшем NC уже выдаёт не команду пропорционального кодом S числа оборотов, а с помощью датчика углового положения, оборудованного на шпиндель, измеряет положение шпинделя и в зависимости желаемого углового поворота выдаёт команду как для привода, так и для остальных осей регулируемой позиции. Это является обратной связью позиции.

Для того, чтобы на данном станке позиционировать шпиндель, необходимо оборудовать датчик на шпиндель, или привод шпинделя должен быть таким, чтобы мог работать и в режиме обратной связи позиции.

11.7.1 Ориентировка шпинделей

Ориентировкой шпинделя, или ориентированного останова шпинделя называется та функция, когда шпиндель устанавливается в определённое угловое положение. Это необходимо например, при автоматической смене инструмента, далее для выполнения некоторых циклов сверления.

Положение бита #1 ORI=1 параметра N0607 Spindle Config означает, что данный шпиндель можно ориентировать.

Команда для ориентировки выдаётся функцией

M19

Если на станке, или в канале имеется несколько шпинделей, помимо функции M19 надо выбрать и шпиндель. Например:

S2=0 M19

Ориентировка технически может выполнена двояко.

Если шпиндель не имеет обратную связь регулировки позиции (положение параметра N0607 Spindle Config #2 INX=0), ориентировка выполняется например, с помощью поворота шпинделя на включатель положения, оборудованного на станок.

Если шпиндель можно включить в обратную связь регулировки позиции (положение параметра N0607 Spindle Config #2 INX=1), под действием команды M19 устройство ЧПУ находит нулевой импульс датчика шпинделя. После этого устройство ЧПУ автоматически выполняет замыкание контура регулировки позиции.

Это заодно означает и приём референтной точки шпинделем, то есть после ориентировки можно направить шпиндель в абсолютное угловое положение.

11.7.2 Останов шпинделей и замыкание контура регулировки позиции

Положение параметра N0607 Spindle Config #2 INX=1 означает, что контур регулировки позиции замыкаемый.

В этом случае параметром N0823 M Code for Closing S Loop можно задавать код M, под действием которого шпиндель остановится, замыкает контур регулировки позиции, но не становится в позицию ориентировки (не ищет нулевой импульс датчика).

Например, если значение параметра 24, замыкание контура происходит под действием

M24

Если на станке, или в канале имеется несколько шпинделей, помимо функции M19 надо выбрать и шпиндель. Например:

S2=0 M24

На счёт кода и его работы можно получить сведения от строителя инструментального станка.

Эта функция например, может ускорить выполнение циклов нарезания резьбы с жёстким хвостовиком.

11.7.3 Программирование позиционирования шпинделей

Положение параметра N0607 Spindle Config #2 INX=1 означает, что контур регулировки позиции замыкаемый. Только в этом случае имеется возможность для позиционирования шпинделей.

Параметр устанавливается строителем станка в том случае, если функция была осуществлена на данном шпинделе.

Позиционирование на основании названия оси

Каждому шпинделю можно давать название оси с длиной не более 3-х символов, на которые можно ссылаться после замыкания контура регулирования позиции.

Параметром N0817 Spindle Axis Name1 надо обязательно записать буквы А, В, или С. На втором (параметр N0818 Spindle Axis Name2) и на третьем символе (N0819 Spindle Axis Name3) можно задавать буквы английского алфавита: А, В, С, D, ... Y, Z, далее цифры: 0, 1, 2, ..., 9.

Например:

CS: название 1-го шпинделя

CS2: название 2-го шпинделя

ABC: название 3-го шпинделя

В случае названия, заканчивающиеся цифром, надо использовать знак =. Заданное название конечно не может совпадать с другими названиями.

После названия надо задавать *позицию в градусах*. Позиционирование происходит со скоростью быстрого хода, установленного для шпинделя. Единица измерения быстрого хода 1/мин.

Перед *позиционированием абсолютным* заданием данных надо запрограммировать M19.

Например: ряд команд

S2=0 M19

G90 CS2=30

наводит шпиндель CS2 в позицию 30 градусив, вращая шпиндель в положительное направление.

В случае абсолютного позиционирования из данных, больших 360 градусов, *срезаются целые обороты*, Например: задавая

S2=0 M19

G90 CS2=750

шпиндель поворачивается на 30 градусов. Вращение будет *всегда в направлении кратчайшем пути*: в случае

S2=0 M19

G90 CS2=270

вращение будет в отрицательном направлении на 270 градусов.

Перед *позиционированием инкрементным* заданием данных не надо запрограммировать M19, достаточно только замыкать контур позиции. Если замыкание контура совершается на M24: в случае задания данных

S2=0 M24

G91 CS2=3600

или

S2=0 M24

CS2=I3600

шпиндель совершает в заданном (в примере в положительном) направлении 10 оборотов. В случае инкрементного задания данных устройство ЧПУ *не срезает целые обороты и поступает в направлении, согласно знаку*.

Индексация шпинделей функцией M

Индексировать шпинделей можно использованием кодов M. Индексированием называется вращение шпинделя в дискретные, заранее установленные позиции.

Для этого надо установить следующие параметры:

Битом #7 IDS параметра N0607 Spindle Config - направление индексации: =0 положительное, =1 отрицательное.

Параметром N0820 Start M of Spnd. Pos. - начальное значение кодов M, параметром N0821 No. of M Code for Spnd. Pos. - номер кодов M.

Параметром N0822 Basic Angle of Spnd. Pos. - угол индексации.

Пример:

Шпиндель можно фиксировать по 18 градусов, отсчитывая от позиции ориентировки. Установка следующая:

Spindle Config: #7 IDS=0: индексирует в положительном направлении

Start M of Spnd. Pos.=201 (m=201)

No. of M Code for Spnd. Pos.=20 (n=20)

Basic Angle of Spnd. Pos.=18 ($\varphi=18$)

Таблица ниже задаёт значение различных кодов M:

Код M	Угол поворота α
Mm (M201)	$\alpha=\varphi=18^\circ$
M(m+1) (M202)	$\alpha=2\varphi=36^\circ$
M(m+2) (M203)	$\alpha=3\varphi=54^\circ$
....	
M(m+n) (M220)	$\alpha=n\varphi=360^\circ$

Берём следующий пример на основании приведенных установок выше:

M19 (ориентировка)
M205 (поворот на 90°)
... (сверление)
M210 (поворот на 180°)
... (сверление)

Прежде чем выдавать код M для позиционирования, с помощью M19 выполняем ориентировку шпинделя. При этом шпиндель попадает в позицию $\alpha=0$.

На 90° надо просверлить отверстие, командой M205 шпиндель наводится на $\alpha=5\varphi=5*18=90^\circ$.

Следующее отверстие надо просверлить в позиции 270° . Поскольку кодом M можно перемещать шпиндель только инкрементно, запрограммируем M210, так как при этом перемещение будет $10*18^\circ=180^\circ$.

11.7.4 Синхронизация двух шпинделей верно позиции

Два шпинделя можно так синхронизировать, что оба шпинделя вращались на тех же оборотах и имели во время вращения установленное смещение по фазе по сравнению к другому.

Синхронизация двух шпинделей верно позиции программа PLC осуществляет обычно кодом M. Способ синхронизации содержит описание строителя станка.

Для синхронизации двух шпинделей необходимо оборудовать оба шпинделя датчиком и иметь возможность замыкания контура регулирования позиции.

Во время синхронизации **максимальное число оборотов шпинделей** ограничивается установленной для шпинделей **скоростью быстрого хода**. Это обычно меньше максимального числа оборотов шпинделя.

В ходе синхронизации различаем мастер-шпиндель и вспомогательный шпиндель. Всегда вспомогательный шпиндель синхронизируется под мастер-шпиндель.

Порядок синхронизации следующий:

- если число оборотов мастер-шпинделя выше, чем меньшее из заданных для двух шпинделей (мастер-шпиндель и вспомогательный шпиндель) число оборотов быстрого хода, устройство ЧПУ замедляет до подходящих оборотов,
- замыкается контур регулирования позиции,
- вспомогательный шпиндель раскручивается до числа оборотов мастер-шпинделя, в направлении, совпадающем с мастер-шпинделем или противоположном с ним,
- вспомогательный шпиндель замыкает контур регулирования позиции,
- затем вспомогательный шпиндель натянет свой нулевой импульс на нулевой импульс мастер-шпинделя, или становится на расстояние, заданное импульсом датчика параметром N0685 Spindle Phase Shift от нулевого импульса мастер-шпинделя, со сдвигом по фазе.

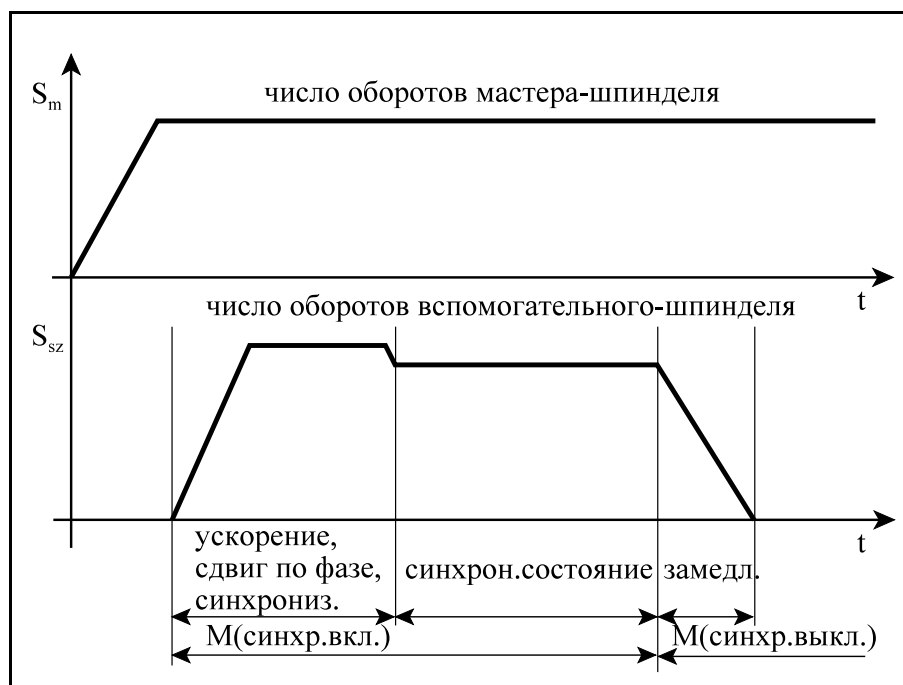


рисунок 11.7.4-1

Пример

На станке с контршпинделем в главном шпинделе (обозначим S1) надо обработать и другую сторону заготовки. Главный шпиндель работает с прутком. Сперва надо синхронизировать контршпиндель, вспомогательный шпиндель (обозначим S2) к главному шпинделю (мастер-шпиндель), закрепить заготовку, затем отрезать её.

Пусть будет M21 синхронизацией, осуществимой совмещенным бегом нулевых импульсов и M22 синхронизацией со сдвигом по фазе.

Если не требуется сдвиг по фазе, фрагмент программы следующий:

```

S1=3000 M3
S2=0 M21 (S2 синхронизация к S1)
...      (Зажим заготовки патроном S2)
...      (Отрезка)
S2=1200 M4 (выключение синхронизации, S2 вращение)

```

Если надо закрепить фасонную заготовку, надо использовать код M22 и установить сдвиг по фазе параметром N0685 Spindle Phase Shift.

☞ **Внимание!** Пример выше является только образцом. В конкретном случае надо поступать согласно описанию строителя станка!

11.7.5 Выключение режима регулирования позиции шпинделей

После ориентировки, позиционирования, или синхронизации шпиндели надо выключить из режима, регулируемого по позиции следующими функциями:

M3, M4, или M5

Например:

```

S1=0 M19 (режим регулирования позиции вкл., S1 ориентиров-
          ка)
CS1=60
...
S1=0 M5 (режим регулирования позиции выкл., шпиндель сто-
        ит)

```

или

S1=2400 M3 (режим регулирования позиции выкл., шпиндель вкл.)

11.8 Преобразование шпинделя в ось, оси в шпиндель

Шпиндельные оси можно использовать только ограниченным образом для обработки потому, что их можно только позиционировать, в интерполяции с другими осями не могут участвовать. Кроме того, шпиндельные оси используют датчик с малым разрешением, работающим на высоких оборотах, а к поворотному столу нужен измеритель хода с большим разрешением, работающим на низких оборотах.

Поэтому в ходе обработки может быть, потребуется преобразовать шпиндель в ось, или ось в шпиндель. Шпиндель токарного станка надо преобразовать в ось C для того, чтобы интерполяцией полярных координат сумели фрезеровать на торцевую поверхность заготовки, или гравировать что-нибудь с цилиндрической интерполяцией на боковую поверхность. После этого ось C можно преобразовать обратно в шпиндель, чтобы продолжать точение.

Поворотный стол В горизонтального центра обработки можно преобразовать в шпиндель, если конструкция станка позволяет этого, чтобы станком можно было выполнить операции точения на заготовке.

Для преобразования шпинделя в ось, или оси в шпиндель требуется соответствующее механическое и электроническое построение инструментального станка. Возможно ли этого на данном станке, содержится описание данного станка. Переход осуществляется всегда программой PLC, соответственно требованиям данного станка.

Порядок преобразования **шпинделя в ось** обычно следующий:

- останов шпинделя, если вращается,
- останов работы привода,

- отключение шпинделя от привода,
- преобразование датчика в большое разрешение,
- перестановка параметров на приводе,
- включение привода обратно,
- подключение оси к приводу и делать видимым индикацию.

Порядок преобразования *оси в шпиндель* обычно следующий:

- ожидание останова оси,
- останов работы привода,
- отключение оси от привода и делать невидным индикацию,
- преобразование датчика в малое разрешение,
- перестановка параметров на приводе,
- включение привода обратно,
- подключение шпинделя к приводу.

Пример:

На токарном станке преобразовать шпиндель S1 в ось C1 для фрезерования, затем преобразовать обратно ось C1 в шпиндель S1 для дальнейшего точения:

S1=0 M20	(S1 преобразование в C1)
G28 G91 C1=0	(приём референтной точки на оси C1)
...	(фрезерование с использованием оси C1)
S1=3000 M3	(переключение обратно C1 в S1, вращение с 3000)
...	(точение с S1)

☞ **Внимание!** Пример выше является только образцом. В конкретном случае надо поступать согласно описанию строителя станка!

12 Функция T

На номер инструмента *кодом T* ссылаемся в программе деталей. Код T можно задавать не более *8 десятичных цифрах*:

Tnnnnnnnn

Передними нолями можно пренебегать.

12.1 Программирование смены инструментов

А программе деталей можно ссылаться в основном двояким образом. Двоякий образ зависит от конструкции инструментального станка. Применяемую технику вызова инструмента в программе деталей задаётся строителем станка.

Случай А: смена на код T

Смена инструментов на станке производится вручную, или устройством для смены инструментов револьверного типа.

При положении бита #0 TCM=0 параметр N1414 Comp. Config on Lathes *код T содержит и код коррекции инструмента* на цифрах нижнего разряда . На оставшихся высших разрядах можно задавать номер инструмента. При положении бита TCM=0 при вызове инструмента производится и смена инструмента.

Параметром N1413 No. of Digits of Offs. No. in T Code решается, что на скольких цифрах надо задавать коррекцию инструмента. Значение параметра может быть 0, 1, 2, 3.

Tnnnnnnnk: коррекция на 1-ом цифре

Tnnnnnnkk: коррекция на 2-х цифрах

Tnnnnnnkkk: коррекция на 3-х цифрах

где:

nnnnnn: номер инструмента

kk: номер ячейки коррекции

Если коррекция задавали на 2-х цифрах

T1236

значение команды:

загрузить инструмент с номером 12 и

вызывать рядом с ним группу коррекции с номером 36.

Если запрограммировать на номер инструмента 0, или запрограммировать только столько цифр на адрес T, сколько цифр необходимо для задания коррекции, смена инструментов не происходит, приглашаем лишь новую коррекцию. Оставаясь при установки выше, например

T12

означает: вызывать группу коррекции 12 и предыдущий инструмент остаётся загруженным.

Если запрограммировать в один и тот же кадр команду движения и номер инструмента (T), выполнение функции T совершается во время выполнения команды движения, или после его выполнения. Способ выполнения определяется строителем станка.

Номер инструмента передаётся программы PLC.

Случай В: смена на функцию М6

Если смена инструмента на станке потребует подготовку инструмента, то есть инструменты находятся в магазине и загрузить их можно рычажным устройством для смены, номер инструмента не должен содержать код коррекции, так как код Т только подготовит инструмент к смене и смена выполняется позже, под действием функции М6.

В этом случае при положении бита #0 ТСМ=1 параметра N1414 Comp. Config on Lathes **код Т не содержит код коррекции инструмента.**

Смена инструмента на станке потребует подготовку инструмента. Эти шаги следующие:

- Надо поискать загружаемый инструмент в магазине инструментов. При этом в программе деталей ссылка на адрес

Tnnnnnnnn

приводит соответствующий инструмент в положение смены. Эта операция выполняется на заднем плане, параллельно с обработкой.

- Суппорты отправить в позицию смены.
- Выполнение смены инструмента функцией

М6

в программе. (можно использовать и М06.) Выполнением смены инструмента подождёт устройство ЧПУ до тех пор, пока не попадает подготавливаемый инструмент Т в положение смены. Под действием этого новый инструмент закладывается в держатель инструмента. Отсюда может продолжаться обработка.

- Предыдущий инструмент возвращается в магазин инструментов. Это действие происходит на заднем фоне, параллельно с обработкой.
- В магазине инструментов начнётся поиск нового инструмента.

В случае выполнения смены на функцию М6 необходимо выполнить следующую установку параметров:

Для поиска кадра установка бита N1338 Block No Search #0 M06=1,
для обращения стойкости резца установка бита N2901 Search Config #1 TLC=0.

Образцовый пример:

T12	М6	(загружается инструмент T12)
T15		(вызов инструмента T15, PLC поищет во время обработки)
...		(обработка инструментом T12)
M6		(загружается инструмент T15)
T8		(вызов инструмента T8, PLC ищет во время обработки)
...		(точение инструментом T15)
T9		(вызов инструмента T9, PLC ищет во время обработки)
M6		(загружается инструмент T8)
...		(точение инструментом T8)

В случае записи в одном кадре команды Т и М6, PLC обычно выполняет функцию Т, затем М6, то есть загружает инструмент. После смены инструментов, М6, целесообразно в следующем кадре запрограммировать вызов инструмента Т, чтобы минимизировать вспомогательное время станка.

Внимание! Приведенный выше пример является только образцовым примером. В конкретном случае надо поступать согласно описанию строителя станка!

13 Смешанные и вспомогательные функции

13.1 Смешанные функции: (коды М)

Записав после адреса М не более 8 значное числовое значение, NC передаёт код для PLC.

Mnnnnnnnn

В кодах М ведущими нолями можно пренебрегать.

Устройство ЧПУ может передавать одновременно 8 различных кодов М для PLC, то есть *в один кадр* можно записать *не более 8 кодов М*.

Порядок выполнения записанных в один кадр функций М определяется строителем станка в *программе PLC*.

Если в одном и том же кадре запрограммировать команду движения и смешанную функцию (код М), смешанная функция выполняется то ли параллельно смешанной функцией, то ли после выполнения команды движения.

Устройство ЧПУ каждый код М передаёт для PLC, даже те, которые выполняются устройством ЧПУ.

Образ выполнения определяется строителем станка.

Функции М управления программой:

М0: запрограммированный стоп

Его выполняет PLC. В конце того кадра, в котором задаётся М0, NC обычно попадает в состояние стопа, остановит шпиндели, выключает охлаждающую воду.

Все модально пердающиеся функции остаются неизменными. Под действием старта перезапускаются шпиндели, включается обратно охлаждающая вода и продолжается программа.

М1: условный стоп

Его выполняет PLC. Его действие тождественное с действием кода М0. Останов будет только во включенном состоянии клавиши УСЛОВНЫЙ СТОП. Если не включена соответствующая клавиша, М1 не действует.

М2, М30: конец программы

Его выполняет PLC. Он означает конец главной программы. Программа PLC заботится о приведении функций станка в основное положение, обычно останавливает вращение шпинделей и выключает охлаждающую воду.

Каждая выполненная М2 или М30 увеличивает на один показание счётчиков заготовок, если только не выделен другой код М параметром N2305 Part Count М для сдвига счётчика.

М96: разрешение макроса прерывания

Код передаётся для PLC, но выполняет его NC. Он разрешает сигнал прерывания, поступивший от PLC, под действием которого вызывается макрос прерывания.

М97: запрет макроса прерывания

Код передаётся для PLC, но выполняет его NC. Запрещает действие сигнала прерывания, поступившего от PLC и прерывает прогон макроса прерывания.

М98: вызов подпрограммы

Код передаётся для PLC, но выполняет его NC. Под его действием вызывается подпрограмма.

М99: конец подпрограммы

Код передаётся для PLC, но выполняет его NC. Под его действием выполнение возвращается на место вызова.

Коды М управления шпинделем**М3, М4, М5, М19:** коды управления шпинделем

Его выполняет PLC. Смотри главу “Функции М управления шпинделем”.

Коды М управления шпинделем, задаваемые параметром

Можно выделить дальнейшие **коды М управления шпинделем** в одном блоке параметрами N0689 Spindle M Low и N0690 Spindle M High. Смотри главу “Функции М управления шпинделем”. **Коды М управления шпинделем** вместе с кодами **М3, М4, М5, М19 являются исключаяющими друг друга функциями**, в одном кадре можно задавать только один из этих кодов.

Его выполняет PLC.

Начальное значение **кодов М**, выполняющих **индексирование шпинделей** можно задавать параметром N0820 Start M of Spnd. Pos., номер кодов М, выполняющих индексирование - параметром N0821 No. of M Code for Spnd. Pos.. Смотри главу “Программирование позиционирования шпинделей”.

Его выполняет PLC.

Коды М, обращающиеся диапазонами чисел оборотов**М11, ..., М18:** коды смены диапазона шпинделей.

Его выполняет PLC. Смотри главу “Обращение диапазонами чисел оборотов”.

Коды М смены инструмента**М6:** код смены инструмента

A PLC hajtja végre.

Группы кодов М, устанавливаемые параметром

На 16 парах параметров можно выделить **16 различных групп кодов М**. Его выполняет PLC.

На параметры N1341 M GR Low 1, ..., N1356 M GR Low 16 надо написать код с наименьшим числом из группы, а на параметр N1357 M GR High 1, ..., N1372 M GR High 16 - код с наибольшим числом.

Группы кодов М надо задавать так, чтобы они были кодами, означающими **исключающих друг друга состояний станка**.

При выполнении программы NC **профильтрует коды М так**, чтобы из кодов М, относящиеся в одну группу, в данном кадре может быть только один, иначе выводится сообщение об ошибке *Противоречивые коды М*

Устройство ЧПУ даже в ходе *поиска кадров* принимает во внимание установленные параметром значения, при отборе кодов М. Отбираются из кодов М, относящиеся к одной группе только код, заданный последним.

Значения этих кодов М программа PLC выводит и на экран в **Окне кодов М**.

Например:

Пусть будут коды М отжима и зажима патрона:

M51: зажим патрона

M52: отжим патрона

M53: отжим патрона при вращении шпинделя

Параметры установлены следующим образом:

N1341 M GR Low 1=51

N1357 M GR High 1=53

Коды М зажима патрона:

M54: сжатие внутрь

M55: сдвиг наружу

Параметры установлены следующим образом:

N1342 M GR Low 2=54

N1358 M GR High 2=55

В программе деталей в один кадр можно записать только M51, или M52, или M53, иначе устройство ЧПУ выводит сообщение об ошибке при прогона программы. То же самое относится и к группе M54, M55.

В ходе поиска кадров из кодов M51, M52, M53 отбирается и выполняется только запрограммированный последним. То же самое относится и к группе M54, M55.

Из соответствующих состояний станка выводится в окне Функции М всегда только код состояний, действующих в данной группе.

Коды М, выполняющие синхронизацию каналов

На параметре N2201 Waiting M Codes Min и N2202 Waiting M Codes Max можно выделить группу кодов М, с которой можно осуществить между каналами синхронизацию, дожидание. Его выполняет PLC.

13.2 Вспомогательные функции (А, В, С, U, V, или W)

На параметре можно выделить рядом с адресом М, S, Т ещё дальнейших 3-х адресов, по которым можно передавать вспомогательные функции для программы PLC. Устройство ЧПУ способное разом передавать все три вспомогательные функции.

На параметре N1333 Aux Fu Addr1, N1334 Aux Fu Addr2, N1335 Aux Fu Addr3 можно выбирать по одному из адресов А, В, С, U, V, W, по которым можно передавать вспомогательную функцию.

Вспомогательным функциям можно присвоить значение не более 8 десятичными цифрами. Если запрограммировать в один и тот же кадр команду движения и вспомогательную функцию, выполнение вспомогательной функции происходит параллельно с командой движения, или после выполнения команды движения.

Порядок выполнения решается строителем станка, и его содержит спецификация инструментального станка.

Например, по адресу В осуществляется индексация поворотного стола.

13.3 Функции разгрузки буфера

В устройстве ЧПУ обработчик кадров **вперёд прочитает кадры, обрабатывает, затем буферизирует их**. Исполнитель (интерполятор и PLC) достаёт из буфера уже обработанные кадры, и выполняет заданные в кадрах движения и функции.

В некоторых случаях может быть необходим *останов прочтения вперёд* кадров. Это может понадобиться *из-за синхронизации деятельности между NC и PLC*.

Например, если для выполнения функции PLC запрашивает одну, или несколько осей от NC, чтобы перемещать их, придётся отложить чтение вперёд. Чтение вперёд можно продолжать только после выполнения функции, после того, как PLC возвратил оси для NC. После этого NC может продолжать обработку уже отсчитывая от позиции оси, изменённой благодаря PLC.

На параметре можно выделить 10 шт. индивидуальных кодов M, 8 шт. групп кодов M, а также все три вспомогательных функций и коды S и T для разгрузки буфера.

Функции разгрузки буфера определяются и устанавливаются строителем станка.

В ходе выполнения программы, например, для учёта коррекции радиуса инструмента (G41, G42) понадобится чтение вперёд кадров. Если при G41, G42 запрограммирована функция разгрузки буфера, устройство ЧПУ отложит расчёт коррекции радиуса инструмента, поэтому контур повреждается.

14 Организация программы деталей

В введении уже представили построение и формат программы деталей. В этой главе изложена организация программ деталей.

14.1 Номер кадра (адрес N)

А Кадры программы снабжены *порядковым номером*. Номерами кадров можно обращаться и как *ярлыками*, на которые можно ссылаться и в других частях программы. Нумерация кадров выполняется командой

Nnnnnnnnn

На адрес N можно написать не более 8 цифр. Пользоваться адресом N не обязательно. Некоторые кадры можно пронумеровать, а других нельзя. Нумерации кадров не обязательно последовать друг за другом в нарастающем порядке.

14.2 Условный пропуск кадра (/ адрес)

Условный пропуск кадра можно программировать командой черты в дроби

/n

Черта в дроби может быть */значение адреса n=1-8*. Числа 1-8 означают порядковые номера включателей. Включатель условного кадра с номером 1 находится на панели оператора устройства ЧПУ.

Оборудование остальных включателей опциональное, определяется строителем станка.

В том случае, если впереди кадра запрограммировать условный пропуск кадра /n, тогда – если n-ый включатель *находится во включенном состоянии, пропускается выполнение* кадра,

– если n-ый включатель *находится во выключенном состоянии, выполняется* кадр.

Если запрограммирован только адрес / впереди, то это относится к включателю № 1:

/ N1200 G0 X200 (пропускается кадр, если включатель № 1 включён)

Можно и так программировать:

/1 N1200 G0 X200 (пропускается кадр, если включатель № 1 включён)

При желании, чтобы устройство ЧПУ принимало во внимание включатель условного кадра хоть и в кадре, предшествующем выполняемому кадру, параметр N1337 Execution Config надо установить в **#4 CBV=0**. При этом команда условного кадра (кадры, начинающиеся со знаком /), *подавляет* прочтения кадра вперёд. В этом случае при **G41, G42** контур *искажается*, однако, включатель условного кадра достаточно *включить во время выполнения предыдущего кадра*, чтобы он действовал.

При желании, чтобы команда / не подавила прочтения кадра вперёд, параметр N1337 Execution Config надо установить в **#4 CBV=1**. При этом команда условного кадра (кадры, начинающиеся со знаком /) *не подавляют* прочтения кадра вперёд. В этом случае при **G41, G42** контур *не искажается*, однако, включатель условного кадра для надёжного действия *надо установить перед выполнением программы*.

Некоторые включатели может использовать и программа PLC для управления прогоном.

Например, в случае станка, оборудованного загрузочным механизмом заготовки, главную программу можно стыковать с помощью M99:

```
... (программа деталей)
M90 (сдвиг счётчика заготовки)
/8 M30 (включателем № 8 управляет счётчиком заготовки)
M99
```

Если счётчик заготовки достиг желаемое количество штук, PLC выключает включатель условного пропуска кадра № 8, программа набежит на M30, и остановится выполнение. Пример выше работает правильно только в случае положения бита #4 CBV=0 параметра N1337 Execution Config, значит, если включателями / подавляется чтения вперёд кадра. В случае положения бита #4 CBV=1 параметра N1337 Execution Config, программа работает правильно тогда, если код M90 установлен на разгрузку буфера.

Перед изменением параметра спросите строителя станка о его действий!

14.3 Запись замечаний в программе деталей: (комментария)

Если часть программы заключать **в круглые скобки ()**, участок между скобками не принимается во внимание обработчиком кадров.

Таким образом в программу деталей можно записать замечания (комментарии).

При желании, чтобы устройство ЧПУ не выполняло фрагмент программы деталей, однако, фрагменты программы не желаем удалить из программы, вставьте желаемый фрагмент в круглые скобки.

Например:

```
N10 G0 X100      (позиционирование на X100)
N20 Z30
(N30 G1 Z60 F0.3
N40 X300)
...
```

В кадр N10 запишем комментарий. Кадр N30 N40 заключаем в скобки, поэтому эти две кадры устройством ЧПУ не принимается во внимание.

14.4 Главная программа и подпрограмма

Различаем двух видов программы: главную программу и подпрограмму. Макропрограммой называем такую подпрограмму, для которой можно передавать аргументы.

В ходе обработки деталей могут встечаться повторяющиеся деятельности, описываемые теми же фрагментами программы. В интересах того, чтобы не надо было повторяющиеся фрагменты много раз написать в программе, из этих фрагментов составим подпрограмму, вызываемую из главной программы.

Построение главной программы и подпрограммы полностью отвечает изложенным в введении.

Разница между ними в том, что пока после выполнения главной программы заканчивается обработка, и устройство ЧПУ ожидает перезагрузки, после заканчивания подпрограммы выполнение возвращается в вызванную программу и оттуда продолжает обработку.

Разница между двумя программами по технике программирования получается из завершения программы. Конец главной программы отмечаем кодом M02, или M30 (пользоваться ими не обязательно), а подпрограмму надо обязательно закрывать кодом M99.

14.4.1 Опознавание программ в накопителе. Номер программы (O)

В накопителе программы расположены в папках под различными названиями, определёнными пользователем. Программы в папках обозначаются именем файла. Устройство ЧПУ считает данный файл программой деталей, то есть данный файл можно запускать как программу деталей тогда, если её расширение (часть после “.”):

имя файла.txt

имя файла.prg

имя файла.nct

или

имя файла.nc

Имя файла программ состоит из символов и цифр.

Подпрограммы сохранены в отдельных файлах, они не могут быть в одном файле с главной программой.

Номер программы

Номер программы

Onnnn.ext

или

Onnnnnnnnn.ext

является специальным именем файла, начинающееся обязательно буквой O и за которым следует 4, или 8 десятичных цифр. На их расширение (.ext) действительны выше изложенные.

Onnnn: *буква O, затем 4 цифр* вместе с передними нолями.

O1 недействительное имя файла,

O0001 действительное имя файла.

или

Onnnnnnnnn: *буква O, затем 8 цифр* вместе с передними нолями.

O01234 недействительное имя файла,

O00001234 действительное имя файла.

14.4.2 Вызов подпрограммы (M98)

Подпрограммы можно вызывать двояким образом: по номеру программы, или по имени файла.

Вызов подпрограммы по номеру программы

Ряд команд

M98 P...

генерирует вызов подпрограммы. На адрес P запишем номер программы вызванной программы. В адресе P можно пропускать передние ноли и после номера нельзя писать расширение. Под действием команды выполнение программы продолжается по подпрограмме, номер которой определяется в адресе P:

вызывающая программа		подпрограмма	замечание
00010			выполнение программы
.....			
.....			
M98 P11	---->	00011	вызов подпрограммы
			выполнение подпрограм-
			мы 00011
			
следующий кадр	<---	M99	возвращение в вызывае-
			мую программу
.....			продолжение программы
.....			00010

В случае подпрограмм, вызванных с номером программы по адресу Р, следующие ограничения действуют к их месту, занятому в системе папок и к имени файла:

- Подпрограммы должны быть в той же папке, как вызывающие их программы.
- К имени файла подпрограммы, вызванной на основании номера программы, действительны ограничения, изложенные в предыдущей главе.
- **Расширение подпрограммы должно совпадать расширением программы, вызывающей её.**

Например, если имя файла главной программы:

Главная программа.prg

Расширение подпрограммы, вызванной из “Главная программа.prg”, должно быть также .prg:

в случае O1234.prg вызов выполняется,

в случае O1234.nct выводится сообщение об ошибке.

Вызов подпрограммы по имени файла

Ряд команд

M98 <подпрограмма.nct>

вызывает подпрограмму под именем “подпрограмма.nct”, находящуюся в одной папке с программой.

Имя файла надо записать между знаками < меньше и > больше.

В этом случае расширение главной программы и подпрограммы не должны совпадать.

Между знаками < и > можно задавать относительный путь к файлу. Путь набирания надо задавать всегда из папки вызывающей программы.

Если вызванная подпрограмма “подпрограмма.prg” **на один уровень ниже** папки вызывающей её программы “программа1.nct”, в подпапке под названием “подпрограммы”:

...

подпрограммы (папка)

программа1.nct (файл)

вызов подпрограммы происходит командой

M98 <\подпрограммы\подпрограмма.prg>

Если вызванная подпрограмма “подпрограмма.prg” **на один уровень выше** папки вызывающей её программы “главные программы”:

...

главные программы (папка)

подпрограмма.prg (файл)

вызов подпрограммы происходит командой

M98 < ..\подпрограмма.prg>

Шаг назад и шаг вперёд можно совершать до нескольких уровней.

Например:

< ..\..\папка1\папка2\папка3\файл.txt>

Однако *длина текста между знаками < и >* может быть *не более 60 символов*.

☞ **Замечание:** При задании пути набирания надо всегда использовать символ \ (backslash), который нельзя перепутать символом / (знак деления).
Для возвращения в системе папок (..) всегда 2 шт. "." точку надо использовать.

Вызов подпрограммы заданием числа повторения

Ряд команд

M98 P... L....

или

M98 <путь к файлу \ имя файла> L....

вызывает одну за другой подпрограмму столько раз, сколько раз заданное по адресу L число показывает.

По адресу L можно задавать не более 8 десятичных цифр.

Если не задавать для L значение, подпрограмма вызывается один раз, то есть устройство ЧПУ предполагает L=1.

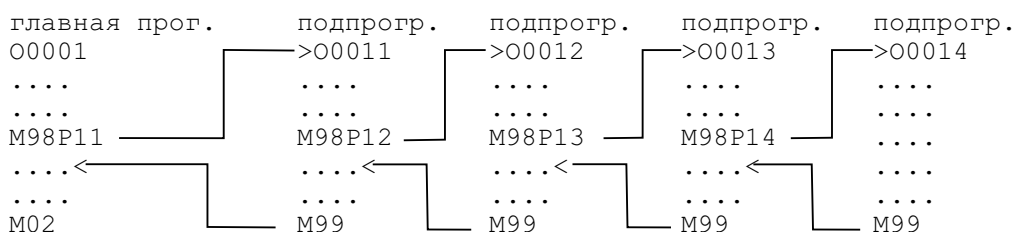
Команда

M98 P11 L6

означает, что вызывать подпрограмму O0011 одну за другой 6 раз.

Многоуровневый вызов подпрограммы

Подпрограмму можно вызывать и из подпрограммы. Вызов подпрограммы (вместе с макровыводами) можно складывать друг в друга до **16 уровней**.



14.4.3 Возвращение из подпрограммы (M99)

Возвращение на кадр, последующий за вызовом

Использование команды

M99

в подпрограмме означает конец подпрограммы, и устройство ЧПУ вернёт её кадру, следующему за вызывающей программой:

вызывающая программа		подпрограмма	замечание
O0010			выполнение программы O0010
.....			
.....			
.....			
N101 M98 P11	--->	O0011	вызов подпрограммы O0011
			выполнение подпрограммы O0011
			возвращение к следующему кадру вызывающей программы
			
N102	<---	M99	продолжение программы O0010

Возвращение к данному кадру

Использование команды

M99 P...

в подпрограмме означает конец подпрограммы, и устройство ЧПУ вернёт её к кадру с номером, заданным по адресу P вызывающей программы. На адрес P можно задавать не более 8 десятичных цифр.

вызывающая программа		подпрограмма	замечание
O0010			выполнение программы O0010
.....			
.....			
.....			
N101 M98 P11	--->	O0011	вызов подпрограммы O0011
			выполнение подпрограммы O0011
			
			
N250	<---	M99 P250	возвращение к кадру N250 вызывающей программы
.....			продолжение программы O0010
.....			

Возвращение с изменением счётчика циклов

Команда

M99 (P...) L...

изменяет счётчик циклов вызывающей программы. Если на L написать 0, подпрограмма вызывается только один раз. Например, если командой

M98 P11 L20

вызывать подпрограмму O0011 и оттуда возвращаться командой

M99 L5

подпрограмма O0011 вызывается всего 6 раз.

Адрес L можно задавать не более, чем на 8 десятичных цифрах.

14.4.4 Переход внутри главной программы

Использование команды

M99

в главной программе приводит к безусловному переходу к первому кадру главной программы, и отсюда продолжается выполнение программы. Использование команды приводит к бесконечному циклу:



Использование команды

M99 P.....

в главной программе приводит к безусловному переходу к кадру с номером, заданным по адресу P главной программы, и отсюда продолжается выполнение программы. Использование команды приводит к бесконечному циклу:



Выход программы из бесконечного цикла возможно то ли с помощью reset, или, если кадр, содержащий команду M99 запрограммировать с условным пропуском кадра,

/ M99

в зависимости от положения включателя с условным пропуском кадра, или пропускает переход, или нет.

14.5 Функции M дожданья между каналами

В случае многоканальной работы может быть понадобится, чтобы в одном канале прогон программы в одной точке дожидает, что программа, работающая в одном или в нескольких каналах дошла до выполнения какой-то операции. Это называется синхронизацией между каналами. Эту синхронизацию можно осуществить т.н. кодами дожданья M.

Коды дожданья M являются *кодами M организации программ, NC обрабатывает их*, для PLC не передаются. *Коды разгрузки буфера M*, то есть такие коды, что пока не происходит синхронизация во всех каналах, прекращается предварительная обработка кадров и продолжается только после синхронизации.

Коды дожданья M можно выделить параметром, их может быть не более 100. Параметром N2201 Waiting M Codes Min можно задавать начальное значение группы и параметром N2202 Waiting M Codes Max конечное значение.

Если например

N2201 Waiting M Codes Min=500 и

N2202 Waiting M Codes Max=599

тогда коды M500, M501, M502, ... , M599 коды M можно использовать для доживания.
Команда

Mm Prrrrrrrrrr

запрограммирует доживание между двумя, или больше каналами. Доживание надо программировать в отдельном ряду.

m: один из кодов доживания M, заданный параметром,

rrrrrrrrrr: номер тех каналов, между которыми надо доживать. Поскольку в системе может быть не более 8 каналов, поэтому адрес P может быть не более, чем из 8 цифр.

Если например, надо дождать между каналами 1 и 2, в программу обоих каналов следует написать для соответствующей точки команду

M501 P12

Программа PLC например, функцией M, или клавишей с помощью указателя CP_NOWT может лишать действия доживания. Это может быть полезно тогда, если желаем прогон программы только в одном из каналов и не желаем комментарием отметить функции доживания M.

Пусть будет минимальное значение кодов M 500, а максимальное значение 599, и пусть будет 3 канала:

Программа канала №1.

... обработка

Программа канала №2.

... обработка

Программа канала №3.

... обработка

N60 M501 P12

дожидает канал 2

... обработка

N100 M501 P12

... обработка

N110 M502 P123

дожидает канал 1 и 2

M502 P123

дожидает канал 1 и 3

N130 M502 P123

... обработка

... обработка

... обработка

Объяснение:

Сперва добежит до кода M501 канал 1 и ждёт до тех пор, пока добежит и канал 2. После синхронизации оба канала продолжают свою программу. В то же время канал 3 непрерывно работает.

До кода M502 добежит сперва канал 3, затем канал 2, и в самом конце канал 1. Канал 3 должен дождаться, чтобы канал 1 и 2 добежал до кода. После этой точки может запускаться обработка во всех трёх каналах.

15 Коррекция инструмента

Для того, чтобы в программе деталей не приходилось принимать во внимание различные значения вылета инструментов, радиусы инструментов и т. д., при задании координат в программе, характеристики инструментов собраны в одной таблице, в так называемой таблице коррекций. Каждый раз при вызове инструмента в программе деталей, надо задавать, что где находятся данные данного инструмента в таблице коррекций. После этого устройство ЧПУ уже проводит инструмент по запрограммированной траектории с учётом ссылочных коррекций.

15.1 Накопитель коррекции. Ссылка на коррекцию инструмента (Т, или D)

Механизм смены инструментов решает, что по какому адресу можно ссылаться на коррекцию инструмента. См. главу “Функция Т”.

Случай А

В случае обменника инструмента револьверного типа при положении бита #0 TCM=0 параметра N1414 Comp. Config on Lathes, **код Т содержит и код коррекции инструмента** на цифрах нижнего разряда. На оставшихся цифрах высшего разряда можно задавать номер инструмента.

Параметром N1413 No. of Digits of Offs. No. in T Code решается, что на скольких цифрах надо задавать коррекцию инструмента. Значение параметра может быть 0, 1, 2, 3.

Tnnnnnnnk: коррекция на 1-ой цифре

Tnnnnnnkk: коррекция на 2-х цифрах

Tnnnnnkkk: коррекция на 3-х цифрах

где:

n...: номер инструмента

k...: номер коррекции

Случай В

Если смена инструмента производится по коду M6, при положении бита #0 TCM=1 параметра N1414 Comp. Config on Lathes, **код Т не содержит код коррекции инструмента**. При этом можно ссылаться на ячейку коррекции, на коррекцию по длине и радиуса вместе по D:

на коррекцию по длине и коррекцию радиуса: по адресу **D**

Цифра после адреса, номер коррекции показывает, что какое значение коррекции вызывается. Предел значения адреса D: 0-999. Передними нолями можно пренебрегать.

Распределение накопителя между каналами

Параметром N1400 No. of Tool Offsets можно задавать **по каналам**, что в данном канале **сколько групп коррекции инструментов** должны быть доступны. К каждой группе принадлежит и коррекция по длине и радиуса.

По всей системе **сумма групп коррекций по токарным каналам** не должна превышать группу **999**. Ссылка на группу коррекций в каждом канале начинается с 1 и продолжается до значения установленного параметра, не смотря на то, что это происходит хоть из программы по адресу Т, D, хоть по таблице коррекций.

Параметром N1412 No. of Common Tool Offsets Т задаётся номер групп коррекций, вызы-

ваемых из *общего, каждого токарного канала*. Ссылаться из токарного канала на фрезерные коррекции нельзя.

Ссылка в каждом канале, начиная с 1 до группы коррекций с номером No. of Common Tool Offsets T показывает на общее значение коррекций, не смотря на то, что это происходит хоть из программы по адресу T, или D, хоть по таблице коррекций.

Пример:

Пусть имеем 3 канала. Пусть будет в 1-ом канале 30 коррекций (No. of Tool Offsets L1=30) во 2-ом 40 и во 3-ем 60. Пусть будет число общих коррекций 10 (No. of Common Tool Offsets T=10).

	1-й канал	2-й канал	3-й канал
N001	от N001 до N010 все три канала читает общие коррекции (T1-T10, D1-D10)		
...			
N010			
...	от N011 до N030 читает свои коррекции (T11-T30, D11-D30)	от N011 до N040 читает свои коррекции (T11-T40, D11-D40)	от N011 до N060 свои коррекции (T11-T60, D11-D60)
N030			
...			
N040			
...			
N060			

Число использованных таким образом групп коррекций: $10+(30-10)+(40-10)+(60-10)=110$ штук.

Распределение накопителя внутри одного канала

С номером коррекции выделяем одну группу таблицы коррекций инструментов для устройства ЧПУ. Элементы этой таблицы следующие:

поряд- ковый номер	X		Y		Z		R		Q
	геом.	износ	геом.	износ	геом.	износ	геом.	износ	
1	123.500	-0.234	87.450	-0.129	267.400	-0.036	1	-0.010	3
2									
3									
...									

Таблица коррекций инструментов содержит вылет инструмента в направлении X, Y и Z,

радиус инструмента (R), а также код позиции инструмента (Q).

Внимание!

Номер коррекции 00 не фигурирует в таблице, имеющиеся на ней значения коррекции всегда ноли, значит, команда T0 или D0 означает удаление коррекций.

Коррекция в направлении X , Y , Z и коррекция радиуса (R) состоит из двух частей: из геометрического значения и из значения износа.

Геометрическое значение: длина/радиус замеренного инструмента. Число со знаком.

Значение износа: мера износа, возникающего в процессе обработки. Число со знаком. В случае ручного ввода данных устройство ЧПУ ограничивает абсолютное значение введенного значения согласно заданному параметром N1415 Max. Amount of Wear Contr. значению.

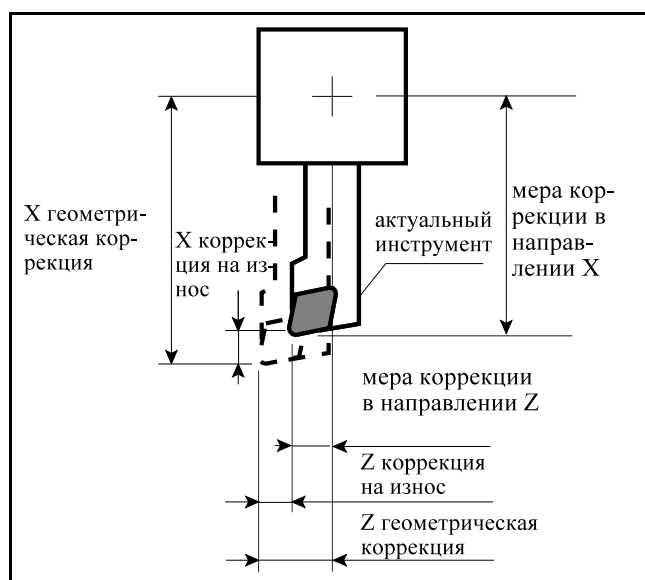


рисунок 15.1-1

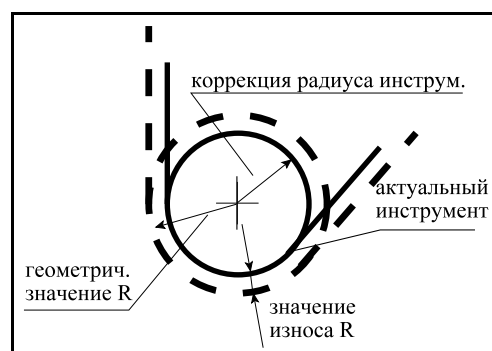


рисунок 15.1-2

Позиция инструмента: Код позиции инструмента (Q) однозначное число, его значение может быть 0,1,...9.

Код позиции инструмента показывает, что смотря с точки центра окружности вершины инструмента в каком направлении находится теоретический вершина инструмента. Следует его задавать всегда с учётом первой и второй оси выбранной плоскости. В случае использования фрезерного инструмента значение Q всегда 0, или 9.

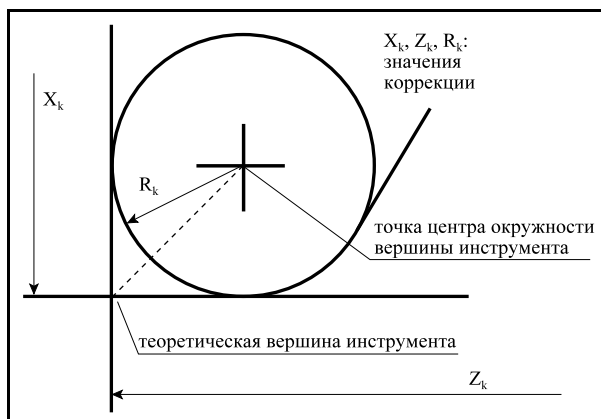


рисунок 15.1-3

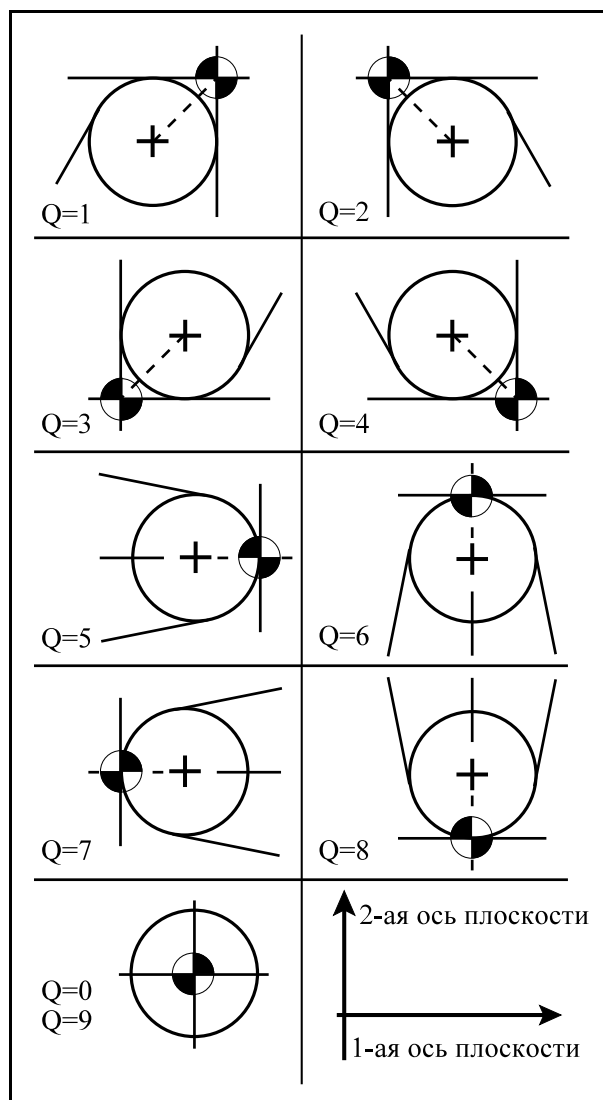


рисунок 15.1-4

Вызванный **код коррекции модальный**, то есть устройство ЧПУ принимает во внимание то же самое значение коррекции до тех пор, пока не получит другую команду T, или D, то есть когда командой T, или D прочитано значение коррекции, тогда изменение таблицы коррекции (например, с программированием G10) уже не имеет действие на прочтённое значение.

Значения коррекций накопителя коррекций сохраняются при выключении.

Значения коррекций инструментов можно установить, а также изменить вводом данных с панели оператора и из программы с использованием команды установки G10.

15.2 2-й накопитель геометрической коррекции

На токарных станках, использующих развёрнутую раскладку инструментов, целесообразно вводить второй накопитель геометрической коррекции, который **относится к коррекции по длине**. Второй накопитель геометрической коррекции такого же размера, как первый, и при ссылке на геометрическую коррекцию по длине, в случае исполнения определённых условий, добавляет значения, сохранённые во втором накопителе геометрической коррекции к значениям, сохранённым в первом накопителе.

Во втором накопителе геометрической коррекции можно задавать положение X , Y , Z держателей инструментов, занятое в станочной системе координат. Это даст возможность, чтобы в первом накопителе геометрической коррекции задавать настоящую длину инструмента x , y , z , то есть, чтобы в первом накопителе геометрической коррекции можно было непосредственно задавать значения вылета, измеренного внешним замером инструмента. Значит, значение учтённой коррекции:

коррекция = 1-ая геометрическая коррекция + 2-ая геометрическая коррекция + коррекция на износ

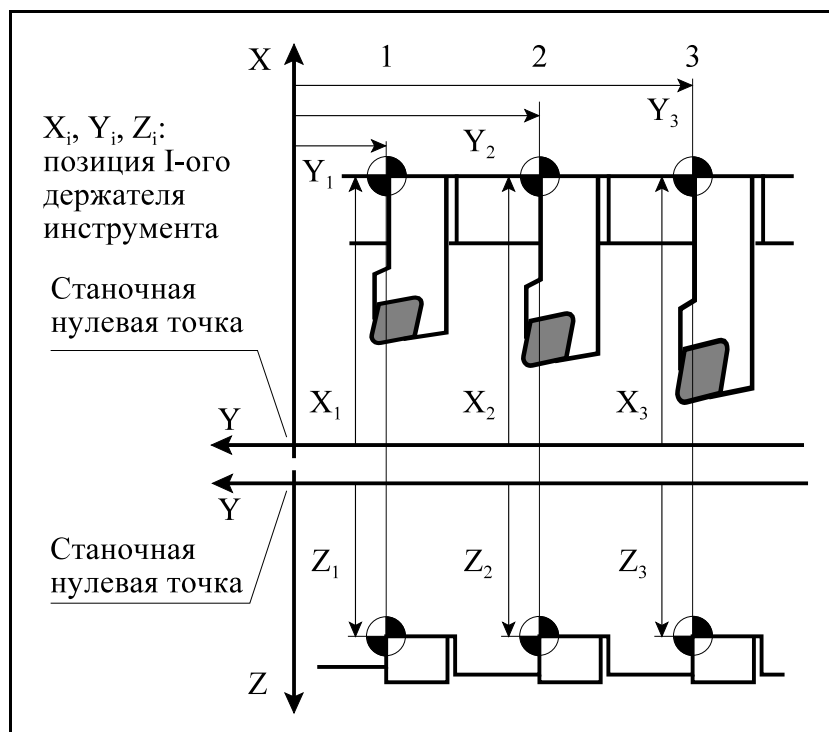


рисунок 15.2-1

Положение бита #4 SGC=1 параметра N1414 Comp. Config on Lathes разрешает доступность второго накопителя геометрической коррекции.

Кроме этого, программа PLC данного станка *разрешает* или *запрещает учёт значения 2-й геометрической коррекции по осям и знака* учитывания.

Замер 2-го накопителя геометрической коррекции и обращение им выполняется строителем станка.

15.3 Изменение значений коррекций инструмента из программы (G10)

Командой

G10 L P X Y Z R Q

можно изменить значения коррекций инструмента из программы. Команда G10 является однократной. Адреса и их значения означают:

L=10: запись геометрического значения

L=11: запись значения на износ

По адресу Р задаётся, что какую группу коррекций желаем изменить:

P: номер группы коррекций

На адресах X, Y, Z, R заносится значение коррекции по длине и коррекции радиуса:

X: значение коррекции по длине на оси X

Y: значение коррекции по длине на оси Y

Z: значение коррекции по длине на оси Z

R: значение коррекции радиуса

В состоянии команды **G90** абсолютного задания данных значение, записанное на адрес X, Y, Z, R попадает в соответствующий регистр коррекций.

В состоянии команды **G91** инкрементного задания данных, или в случае использования оператора I данные, записанные на адрес, добавляется к соответствующему регистру коррекций.

Q: код позиции инструмента (0...9)

Код принимается и заданием L10 и L11.

Командой G10 нельзя изменить данные 2-го накопителя геометрической коррекции.

15.4 Учёт коррекций по длине инструмента по коду T

При положении бита #0 TCM=0 параметра N1414 Comp. Config on Lathes код T содержит и код коррекций инструмента на цифрах нижнего разряда.

Параметром N1413 No. of Digits of Offs. No. in T Code решается, что сколько цифр занимает коррекция в коде T.

Ссылка может быть

Tnnnnnnnk

Tnnnnnnkk

Tnnnnnkkk

где:

n...: номер инструмента

k...: номер коррекции

Номер коррекции вызывает геометрическую коррекцию и коррекцию на износ вместе

Если положение бита #1 GTN=0 параметра N1414 Comp. Config on Lathes, **номер коррекции, вызванной по адресу T**, вызывает **геометрическую коррекцию и коррекцию на износ вместе**.

В случае ссылки Tnnkk:

- значение **nn** устройство ЧПУ передаёт для PLC,
- значение **kk** **геометрической коррекции** и **kk регистра на износ** сложится и это будет значением вызванной коррекции:

$$\text{Коррекция} = \text{Геометрическая}_{kk} + \text{Износ}_{kk}$$

Команда

Tnn00

удаляет коррекцию.

Номер инструмента вызывает геометрическую коррекцию, а номер коррекции - коррекцию на износ

Если положение бита #1 GTN=1 параметра N1414 Comp. Config on Lathes **номер коррекции, вызванной по адресу T**, вызывает **геометрическую коррекцию, а номер коррекции - коррекцию на износ**.

В случае ссылки Tnnkk:

- значение nn устройство ЧПУ передаёт для PLC,
- значение **nn** **геометрической коррекции** и **kk регистра на износ** сложится и это будет значением вызванной коррекции:

$$\text{Коррекция} = \text{геометрическая}_{nn} + \text{Износ}_{kk}$$

Изложенная выше решение поднимает две проблемы. Из какого регистра брать код позиции инструмента: из nn, или из kk, то есть **значение Q**, далее как должен устройство ЧПУ истолковать ссылку **Tnn00**.

Если бит #2 QGW параметра N1414 Comp. Config on Lathes:

- =0: **код позиции инструмента (Q)** берётся из **регистра кода на износ**,
- =1: **код позиции инструмента (Q)** берётся из **регистра геометрического кода**.

Если бит #3 ZCG параметра N1414 Comp. Config on Lathes:

- =0: **0-й номер коррекции (Tnn00)** не удаляет геометрическую коррекцию, только **коррекцию на износ**,
- =1: **0-й номер коррекции (Tnn00)** удаляет **геометрическую коррекцию и коррекцию на износ**.

Под действием кода T смещается система координат на значение коррекции X_k , Y_k , Z_k , принадлежащее к группе коррекций, заданной в коде T. Это означает, что из мгновенной позиции X, Y, Z вычитается значение коррекции направления X_k , Y_k , Z_k , принадлежащее к выбранной группе коррекций. Начиная с этого на индикаторе отражаются координаты не приведённой точки держателя инструмента, а - теоретической вершины инструмента. Под значением коррекции понимаем в дальнейшем всегда сумму геометрической коррекции и коррекции на износ.

Например, под действием ряда команд

```
T0
N10 G0 G90 X700 Z350
N20 T202
N30 X300 Z150
```

в кадре N10 приведённую точку держателя инструмента наводит в точку с координатами X700; Z350 .

В кадре N20 загружается инструмент и вычитает из позиции значения коррекции $X_k=340$, $Z_k=30$, принадлежащее к 2-й группы коррекции. При этом индикатор позиции меняется на значение

$$X=700-340=360; Z=350-30=320.$$

В кадре N20 не происходит никакого движения. В кадре N30 уже наводится теоретическая вершина инструмента в запрограммированную точку X300; Z150, то есть перемещение будет

$$X=300-360=-60; Z=150-320=-170.$$

Если **код T** задавать **вместе с кадром движения**, в позицию конца кадра наводится с учётом нового значения коррекции, определённого кодом T . Однако, если код T содержит и команду смены инструмента, смена инструмента выполняется параллельно с движением, или в конце движения. (Термин смены инструмента определяется строителем станка), Рассмотрим приведенный выше пример:

```
T0
N10 G0 G90 X700 Z350
N20 X300 Z150 T202
```

Сейчас исключим кадр N30 и объединим команды кадра N20 и N30. В кадре N20 в ходе движения наводится уже теоретическая вершина инструмента в точку с координатами

$$X=300, Z=150 ,$$

как в предыдущем примере в случае кадра N30.

Если **по коду T** запрограммировать **только номер коррекции**

```
N10 G0 G90 X700 Z350
N20 X300 Z150 T2
```

смена инструментов не происходит, вызывали всего 2-ую коррекцию в кадре N20.

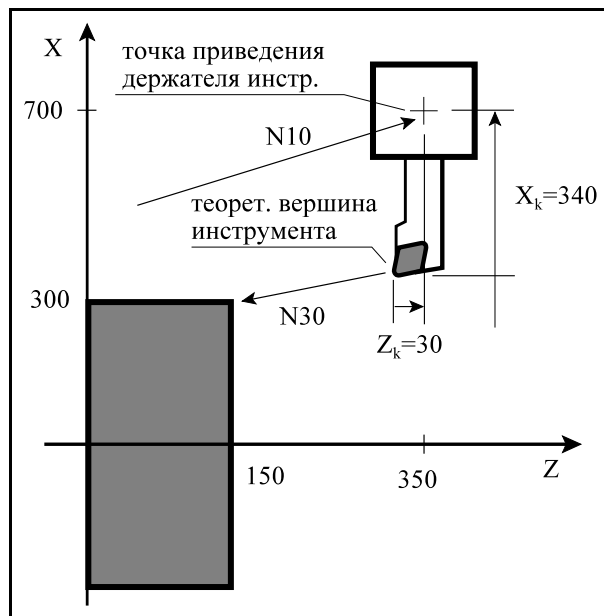


рисунок 15.4-1

Удаление коррекции по длине

0-ая группа коррекции имеет отличительную роль, и означает удаление коррекции.

Под действием команды

Tnn00, vаgу а

T0

удаляется коррекция по длине (nn: произвольный номер инструмента). Процесс противоположный тому, что означал вызов коррекции. Это означает, что к координатам X, Y, Z теоретической вершины инструмента добавляются значения X_k , Y_k , Z_k коррекции, действующие раньше, и в дальнейшем на индикаторе отражаются координаты приведённой точки держателя инструмента. Если удаление коррекции происходит вместе с кадром движения, в запрограммированную конечную точку кадра посылается приведённая точка держателя инструмента.

Например, в образцовом примере

N10 X180 Z120 T202

N20 X200 Z180

N30 X280 Z210 T200

в кадре N10 движение поправляется вызванной коррекцией по сравнению программы, а в кадре N30 удаляется коррекция, то есть установленное в кадре N10 смещение нулевой точки.

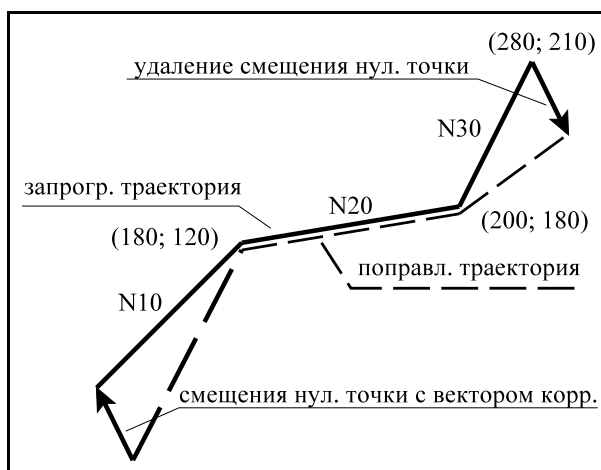


рисунок 15.4-2

15.5 Учёт коррекций по длине инструмента по коду G (G43.7, G49)

При положении бита #0 TCM=1 параметра N1414 Comp. Config on Lathes вызов коррекции по длине инструмента происходит по коду G.

Команда

G43.7

включает коррекцию по длине инструмента. Если вместе с адресом оси программировать, на заданную позицию наводится уже с новой коррекцией.

Адрес D означает: значение коррекции по длине инструмента берётся из группы коррекции, заданной по этому адресу.

В случае *абсолютного задания данных* команда

G43.7 G0 G90 X100 Z10 D5

перемещает вершину инструмента в точку с координатой X100 Z10, с учётом коррекции D5.

В случае *инкрементного задания данных* команда

G43.7 G0 G91 X20 Z10 D1

смещает начальную точку на значение коррекции, затем отсюда, от поправленной начальной точки считается инкрементное смещение.

Логически действие команды

G43.7 G91 X0 Y0 Z0 D2

ось X, Y и Z не выполняет перемещение, только позиция принимает значение, соответствующее новой коррекции.

Действие **G43.7 передаётся модально** совершенно до того, пока из этой группы не получает другую команду. Если изменить значение коррекции с вызовом *нового адреса D*, прежнее значение удаляется, и действовать будет новое значение:

G43.7 X120 Z100 D1 (перемещается на станочную позицию X150 Z=110)

...

X120 Z100 D2 (перемещается на станочную позицию X170 Z=120)

Кадру X120 Z100 D2 модально передалось состояние G43.7, поэтому принимается во внимание новая коррекция, D2.

D0 удаляет коррекцию. Продолжая пример выше:

G43.7 X120 Z100 D2 (перемещается на станочную позицию X170 Z=120)

...

X120 Z100 D0 (перемещается на станочную позицию X120 Z100)

Команда

G49

на всех осях выключает коррекцию по длине инструмента, если в кадр запрограммирован и адрес оси, тогда с движением, или с трансформацией, если в кадр не запрограммировано движение.

После включения, reset и после программы биты #5 G43 и #6 G44 параметра N1300 DefaultG1 решают, что код G43.7 действует. Если обе бита 0, тогда G49 действует.

15.6 Коррекция радиуса инструмента (G40, G41, G42)

В том случае, если использовать только коррекцию по длине инструмента, не возможно точно выточить коническую поверхность, или дугу окружности. В случае использования коррекции по длине инструмента устройство ЧПУ проводит теоретическую вершину инструмента по запрограммированной траектории. Поскольку вершина каждого инструмента имеет меньшее или большее скругление, поэтому поверхность точение будет точная только при движениях, параллельных осям, как это видно на приложенном рисунке. При конических и дугообразных поверхностях диаметр заготовки в каждой точке будет больше запрограммированного. Отклонение показано штриховой линией на рисунке.

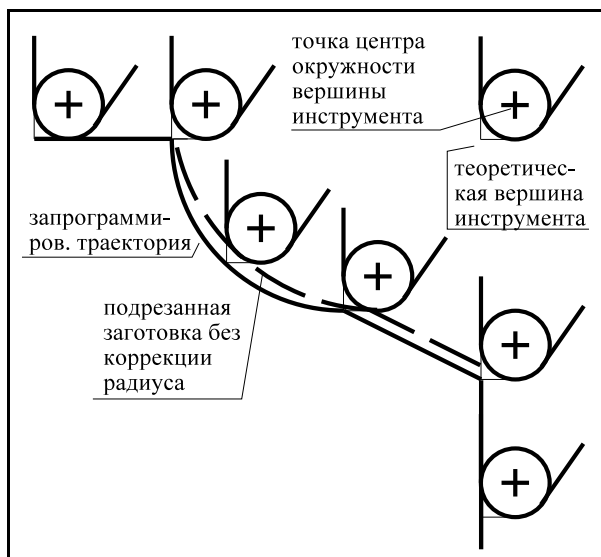


рисунок 15.6-1

Для того, чтобы можно было точно обрабатывать произвольную конфигурацию, и не надо было задавать в программе точки конфигурации по чертежу, независимо от радиуса окружности применённой вершины инструмента, устройство ЧПУ **должно проводить точку центра окружности вершины инструмента параллельно запрограммированному контуру, на расстояние от него, равным радиусу инструмента**. Устройство ЧПУ устанавливает в зависимости от значения **коррекции радиуса инструмента**, зарегистрированной по номеру коррекции, **вызванной по адресу T, или D**, что на какое расстояние проводить траекторию центра инструмента от запрограммированного контура.

Перед наводкой на контур устройство ЧПУ держит на учёта позицию теоретической вершины инструмента.

Для решения того, что по сравнению теоретической вершине инструмента в каком направлении находится точка центра окружности вершины инструмента, нужен **код позиции инструмента**, который задан по адресу Q в таблице коррекций. На основании кода позиции инструмента к позиции теоретической вершины даба-

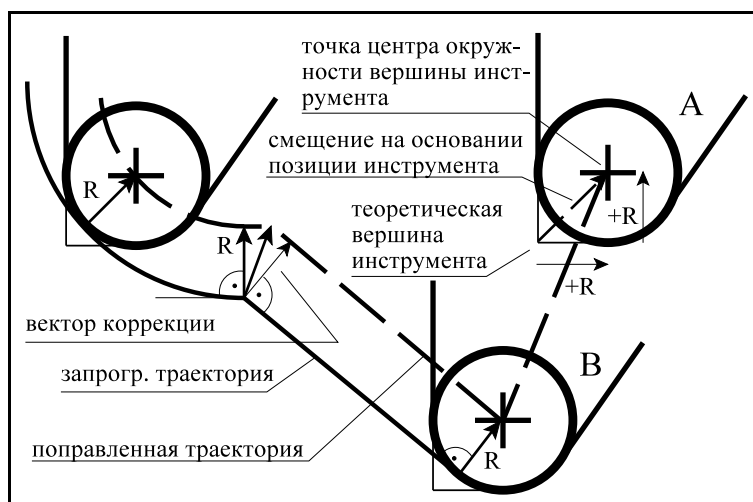


рисунок 15.6-2

вится или вычисляется значение, равное радиусу инструмента (R) в направлении X и Z . (На рисунке положение "А".) После этого наводится **точка центра вершины инструмента** в конечную точку вектора с длиной R , являющегося **вектором коррекции**, перпендику-

лярного в начальную точку запрограммированной траектории. (На рисунке положение “В”.) После этого проводится точка центра окружности вершины инструмента параллельно запрограммированной траектории, на расстояние R от неё.

Вектором коррекции является такой **плоскостной вектор**, который пересчитывается устройством ЧПУ в каждом кадре, и запрограммированные перемещения изменяются векторами начала и конца кадров. Длина и направление полученных векторов коррекции зависит от значения коррекции, вызванного по адресу T и геометрии перехода между двумя кадрами.

Устройство ЧПУ высчитывает векторы коррекции **в плоскости, выбранной командами G17, G18, G19**. Это есть плоскость коррекции радиуса инструмента. Коррекция радиуса не влияет на движения вне этой плоскости. Например: если в состоянии G18 выбрана плоскость X, Z, тогда векторы коррекции вычисляются в плоскости X, Z.

Во время расчёта коррекции радиуса инструмента не разрешается смена плоскости коррекции. Если всё-таки попытаемся, тогда устройством ЧПУ выводится сообщение об ошибке. В том случае, если желаем определить плоскость коррекции, лежащей не вдоль осей главной плоскости, вспомогательные оси надо определить в поле параметров параллельными осями. Например, если U принято параллельной осью, и желаем применить коррекции радиуса инструмента в плоскости Z, U, заданием G18 U__ Z__ можно выделить плоскость.

Расчёт коррекции радиуса инструмента можно включить из программы, и можно выключить:

G40: выключение расчёта коррекции радиуса инструмента

G41: коррекции радиуса инструмента **налево по направлению хода**

G42: коррекции радиуса инструмента **направо по направлению хода**

Команды **G40, G41, G42 передаются модально**. После включения, в конце программы, или под действием reset устройство ЧПУ примет состояние G40.

Команда G41, или G42 включает расчёта коррекции. В состоянии G41 инструмент следит за запрограммированному контуру слева по направлению хода, а в состоянии G42 справа. **Значение коррекции** применённого радиуса инструмента надо задавать **по адресу T, или D**. **Расчёт коррекции выполняется на движения по интерполяции G00, G01, G02, G03**.

Изложенные до сих пор действительны при задании положительной коррекции радиуса инструмента. Однако, **значение коррекции радиуса инструмента может быть и отрицательным**. Это имеет смысл на практике тогда, если например, той же подпрограммой желаем обходить материнскую заготовку, затем сопряжённую с этой отцовскую заготовку. Это можно решать и таким образом, что с помощью G41 выточим материнскую, и с помощью G42 отцов-

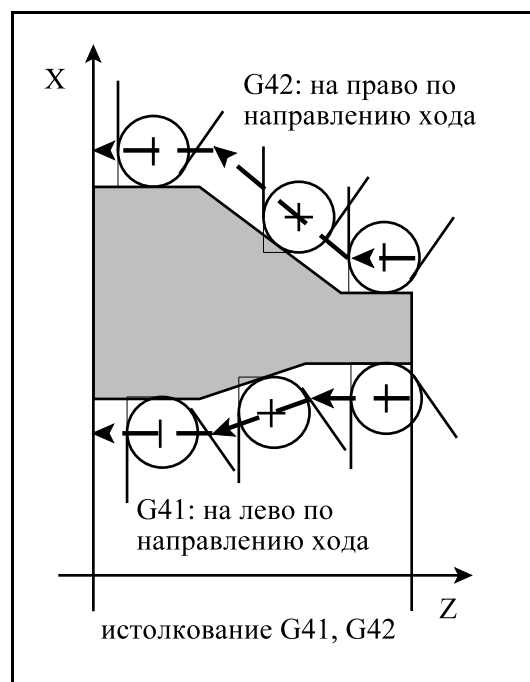


рисунок 15.6-3

кую заготовку. Однако, не нужно эту смену редактировать в программу, если материнскую заготовку выточить с положительной, а отцовскую заготовку с отрицательной коррекцией радиуса. При этом траектория точки центра инструмента сместится противоположной запрограммированной G41, или G42:

	коррекция радиуса: положительная	коррекция радиуса: отрицательная
G41	слева	справа
G42	справа	слева

☞ **Замечание:** В описаниях и рисунках в дальнейшем для простоты работаем всегда положительной коррекцией радиуса.

Команда **G40**, или **Tnn00**, **T0**, **D0** включает расчёт коррекции. Разница между двумя командами в том, что команда T0 ведёт расчёт только с вектором коррекции с длиной 0 и оставляет без изменения состояние G41 или G42. Если после этого совершается ссылка на новый адрес Tnnmm (мм≠0), то D, вектор коррекции рассчитывается с новым радиусом инструмента, в зависимости от состояния G41 или G42.

Однако, если пользуемся командой G40, устройство ЧПУ дальше не высчитывает векторы коррекции.

Включение, или выключение коррекции радиуса имеет определённые правила, которые подробно изложены в следующих главах.

Устройство ЧПУ выполняет команды по коррекции радиуса только в режиме автоматического, или ручного ввода данных. В ручном режиме для индивидуальных кадров не действует. Причина этого следующее. Для того, чтобы устройство ЧПУ сумел высчитывать вектор коррекции в конечной точке кадра, оно должно прочесть следующий кадр, содержащий движение в выбранной плоскости. Вектор коррекции является функцией перехода между двумя кадрами. Из этого видно, что для расчёта вектора коррекции необходимо предварительная обработка нескольких кадров.

Прежде чем приступить к изложению подробностей расчёта коррекции, нужно вводить вспомогательные данные. В точке пересечения двух отрезков, то есть двух кадров, угол между касательными, проведёнными к двум кривым: угол α . Истолкование α зависит от того, что контур обходится слева, или справа.

Устройство ЧПУ выбирает стратегию поворота при точек пересечений в зависимости от угла α . Если $\alpha > 180^\circ$, то есть инструмент работает внутри, рас-

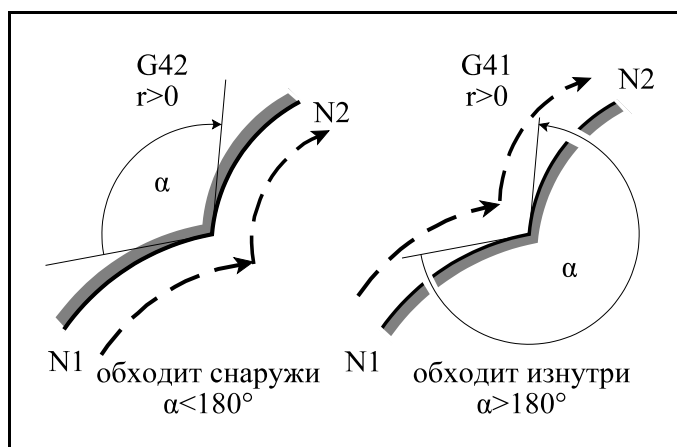


рисунок 15.6-4

считывается точка пересечения между двумя отрезками. Если $\alpha < 180^\circ$, то есть инструмент

обходит снаружи, тогда для обхода вставляются дальнейшие прямые отрезки.

В дальнейшем это описание излагает случаи расчёта коррекции радиуса, в первую очередь с точки зрения задач точения. С точки зрения задач фрезерования случаи расчёта коррекции радиуса изложены в инструкции под названием Описание программирования Устройства ЧПУ Фрезерным станком и центром обработки типа NCT3xxM.

15.6.1 Включение расчёта коррекции радиуса. Наводка на контур.

Под действием команды G41, или G42 из состояния G40 устройство ЧПУ входит в режим расчёта коррекции радиуса. Значение коррекции берётся из ячейки коррекции, заданной по адресу T.

Состояние **G41** или **G42** примет *только в позиционирующем кадре G0, или в кадре G1, выполняющем линейную интерполяцию.*

Если расчёт коррекции желаем включить в периоде (G2, G3), устройство ЧПУ выводить сообщение об ошибке.

Устройство ЧПУ выбирает *стратегию наводки на контур* только тогда, если переключаем *из состояния G40 в состояние G41, или G42.*

Основные случаи включения коррекции видны ниже в функции угла α и возможных переходов: прямая–прямая, прямая–окружность. Рисунки сделаны на случай G42, предполагая *положительную коррекцию радиуса.*

☞ *Замечание:* Значение обозначений рисунков сейчас, и в дальнейшем:

г: значение коррекции радиуса,

L: прямой участок,

C: дуга окружности,

S: место останова в режиме по кадрам,

штриховая линия: траектория точки центра инструмента,

сплошная линия: запрограммированная траектория.

Основные случаи включения коррекции радиуса:

G40	G40
G42 G1 X_ Z_	G42 G1 X_ Z_
X_ Z_	G2 X_ Z_ R_

Наводка на внутренний угол: $180^\circ < \alpha < 360^\circ$

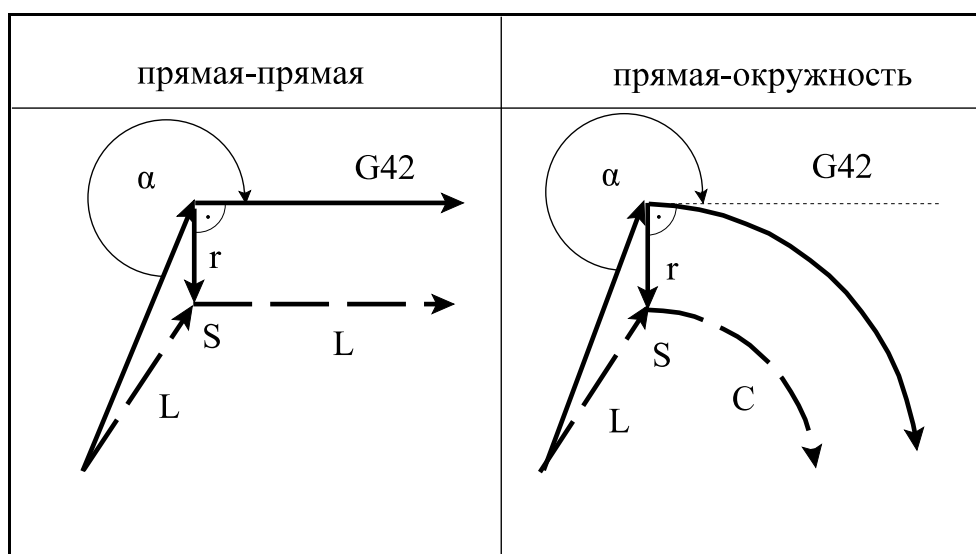


рисунок 15.6.1-1

Наводка на внешний угол под тупым углом: $90^\circ \leq \alpha \leq 180^\circ$

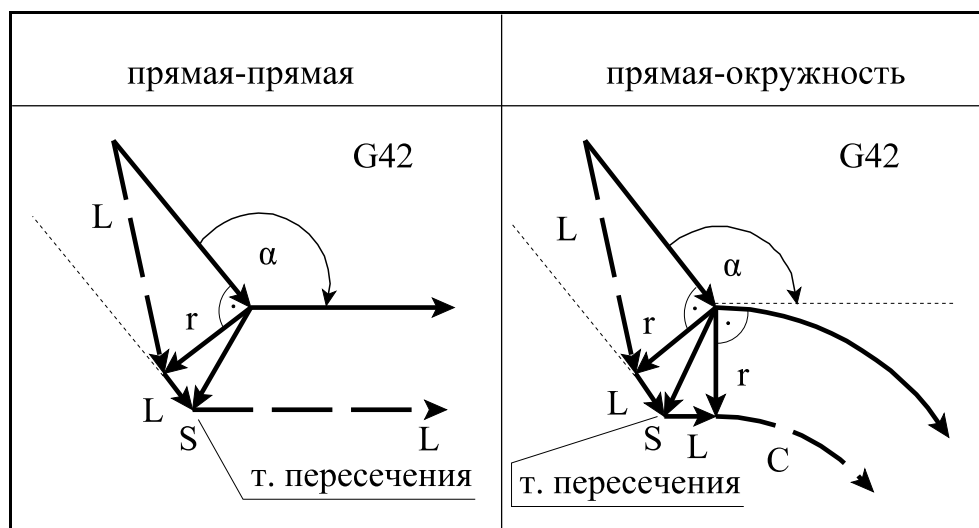


рисунок 15.6.1-2

Наводка на внешний угол под острым углом: $0^\circ \leq \alpha < 90^\circ$

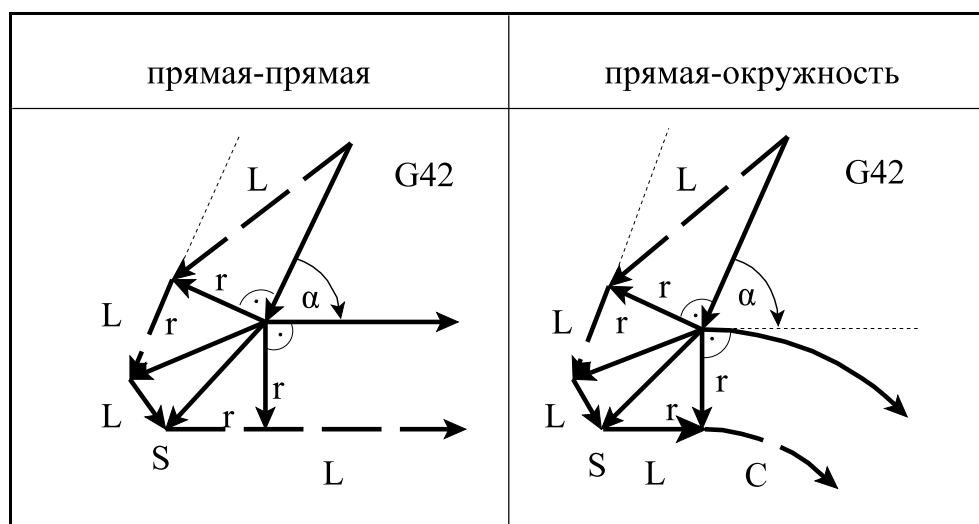


рисунок 15.6.1-3

Пример

Приведенная ниже программа показывает пример для правильного включения коррекции радиуса при позиции инструмента Q3.

Кадр N110 включает коррекцию радиуса. В начало кадра N110 наводится ещё с теоретической точкой вершины инструмента, затем к концу кадра уже окружность касается вершины инструмента к начальной точке кадра N120:

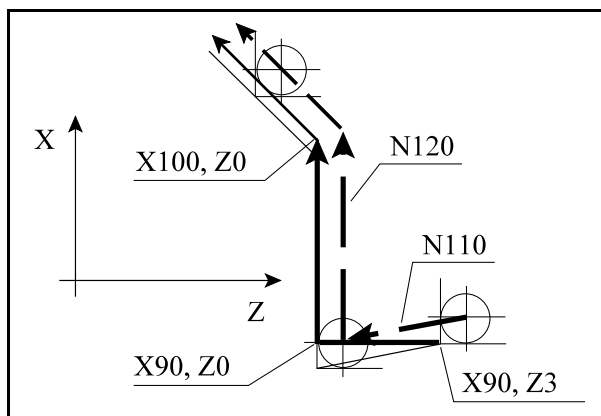


рисунок 15.6.1-4

```
G54 G18
...
N50 G0 X200 Z50
N60 G92 S3000
N70 G96 S600
N80 T101 M3
N90 X160 Z3
N100 X90
N110 G1 G42 Z0
N120 X100
N130 X110 Z-5
N140 Z-25
N150 G2 X130 Z-35 R10
N160 G1 X140
N170 Z-45
N180 G40 X160
N190 G0 X200 Z50
...
```

Включение коррекции радиуса без программирования движения:

Если **включение коррекции** (G41, G42) выполняется в таком кадре, в котором **не участвует ни одного из адресов оси, относящейся к выбранной плоскости**, как в примере ниже, в кадре N110 не заданы адрес ни X, ни Z, тогда в кадре N110 не выполняется движение. Перемещение в конечную точку кадра N120 выполняется так, чтобы между кадром N120 и N130 рассчитывается точка пересечения.

Это не хорошо потому, что таким образом точка X90 Z0 выпадает из обработки! Поправить можно объединением кадров N100 и N120.

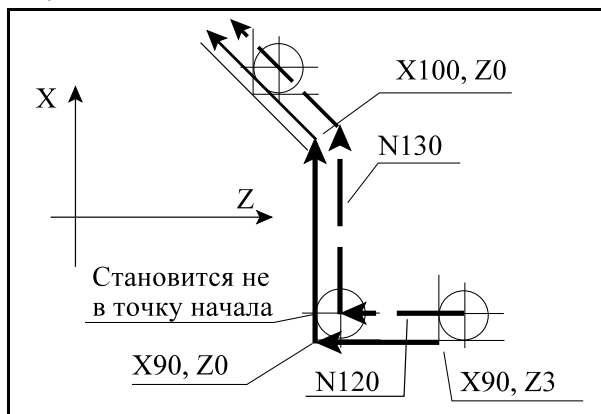


рисунок 15.6.1-5

```
G54 G18
...
N50 G0 X200 Z50
N60 G92 S3000
N70 G96 S600
N80 T101 M3
N90 X160 Z3
N100 X90
```

```

N110 G1 G42
N120 Z0
N130 X100
N140 X110 Z-5
N150 Z-25
N160 G2 X130 Z-35 R10
N170 G1 X140
N180 Z-45
N190 G40
N200 X160
N210 G0 X200 Z50
...

```

Если **включение коррекции** (G41, G42) выполняется в таком кадре, в котором **нет перемещения ни на одной оси, относящейся к выбранной оси**, как в примере ниже, в кадре N110 правда заполнен адрес Z, однако, из-за принадлежащего к нему инкремента 0 нет перемещения, тогда в кадре N110 выполняется перемещение на радиус инструмента.

При этом между кадрами N120 и N130 рассчитывается точка пересечения и так проводится точка центра инструмента. Это не хорошо потому, что таким образом угол X90 Z0 выпадает из обработки! Поправить можно объединением кадров N110 и N120.

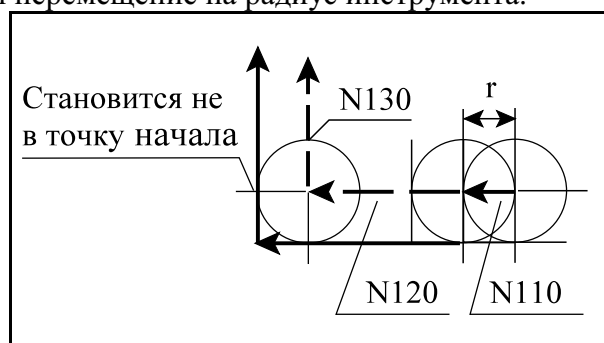


рисунок 15.6.1-6

```

G54 G18
...
N50 G0 X200 Z50
N60 G92 S3000
N70 G96 S600
N80 T101 M3
N90 X160 Z3
N100 X90
N110 G1 G42 G91 Z0
N120 G90 Z0
N130 X100
N140 X110 Z-5
N150 Z-25
N160 G2 X130 Z-35 R10
N170 G1 X140
N180 Z-45
N190 G91 G40 X0
N200 G90 X160
N210 G0 X200 Z50
...

```

В кадре, последующим за включением коррекции радиуса, нет перемещения

Если в кадре, последующим за включением коррекции, в выбранной плоскости, получается 0 перемещения, как в программе ниже в кадре N120, окружность вершины инструмента проводится в конечную точку кадра (N110), выполняющего включение кадра. После этого окружность вершины инструмента проводится в начальную точку кадра (N130), выпол-

няющего следующее движение, перемещается в эту точку, затем инструмент проводится дальше параллельно.

На этой заготовке может причиняться подрезание, эти случаи следует избежать! Поправить можно удалением кадра N120.

```
G54 G18
...
N50 G0 X200 Z50
N60 G92 S3000
N70 G96 S600
N80 T101 M3
N90 X160 Z3
N100 X90
N110 G1 G42 Z0
N120 Z0
N130 X100
N140 X110 Z-5
N150 Z-25
N160 G2 X130 Z-35 R10
N170 G1 X140
N180 Z-45
N190 G91 G40 X0
N200 G90 X160
N210 G0 X200 Z50
...
```

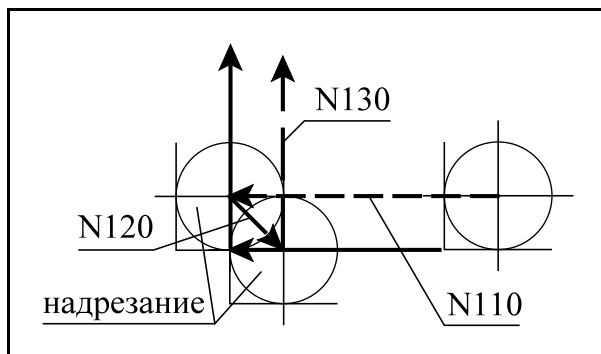


рисунок 15.6.1-7

Включение коррекции радиуса расчётом точки пересечения

Если в кадре (G41, или G42), выполняющий включение коррекции присвоится значение для I, J, K, но только для лежащих в выбранной плоскости (например: в случае G17 для I, J), тогда устройство ЧПУ наводиться в точку пересечения последующего кадра с прямой, определённой с помощью I, J, K, с учётом коррекции радиуса. Значение I, J, K всегда инкрементное, и полученный таким образом вектор показывает в конечную точку того кадра, в котором запрограммирован. Это возможность полезно например, при настройке на внутренние углы.

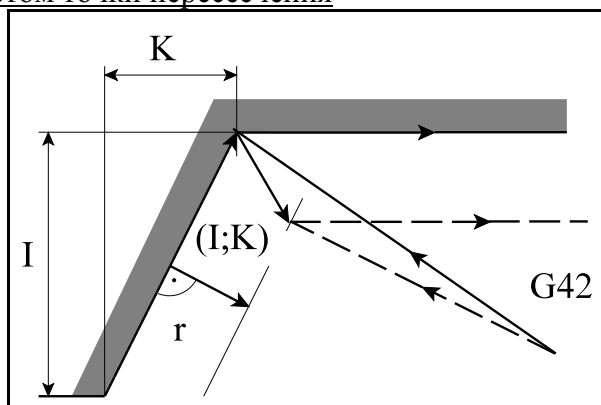


рисунок 15.6.1-8

Пример:

Обрабатывается изнутри шестигранник с длиной грани 100 мм. Инструмент наводится в угол, лежащий в точке с координатой X100 Y0. Кадр N50 приложенной программы с расчётом точки пересечения наводится на контур, заданием I и J. Координаты I и J рассчитываются на основании перемещения, выполненного в кадре N110 программы.

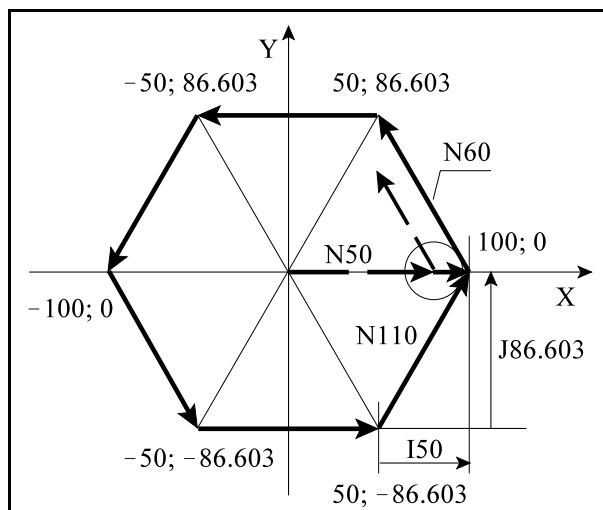


рисунок 15.6.1-9

```

N10 G54 G17 M3 S300
N20 G0 X0 Y0 Z100
N30 G43 Z5 H1
N40 G1 Z-5 F1000
N50 G41 G0 X100 Y0 I50 J86.603 D1
N60 G1 X50 Y86.603
N70 X-50
N80 X-100 Y0
N90 X-50 Y-86.603
N100 X50
N110 X100 Y0
N120 G40 G0 X0 I-50 J86.603
N130 M30

```

Случаи наводки с помощью точки пересечения

В случае задания I, J, K устройство ЧПУ всегда рассчитывает точку пересечения, не зависимо от того, что обрабатывается внутренний, или наружный угол.

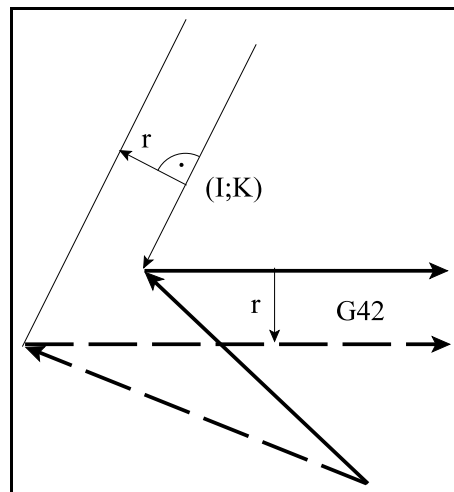


рисунок 15.6.1-10

Если не находит точку пересечения, наводится перпендикулярно в начальную точку последующего кадра.

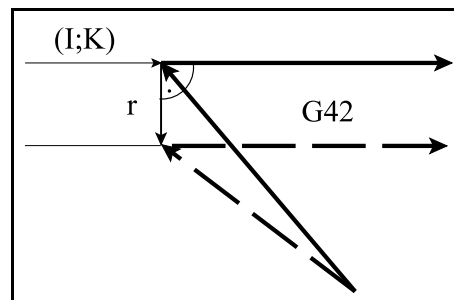


рисунок [15.6.1-11](#)

15.6.2 Включенное состояние расчёта коррекции радиуса. Движение по контуру.

Во включенном состоянии расчёта коррекции радиуса, в выделённой плоскости непрерывно рассчитываются векторы коррекции между кадрами G0, G1, G2, G3. Для расчёта этих векторов необходимо непрерывно прочесть вперёд кадры.

Устройство ЧПУ вперёд прочитает количество кадров, заданное параметром N1404 ВК No. Interf + 2 кадра. Это означает и то, что если между двумя кадрами движения, принадлежащими к выделённой плоскости, имеются и прочие кадры, заданные параметром, например, функция, задержка, движение вне плоскости и т. д., расчёт коррекции всё равно будет непрерывным.

Непрерывность могут прерывать такие коды G и функции, которые вынуждают разгрузку буфера, то есть отложат чтение вперёд кадров.

Основные случаи включенного состояния расчёта коррекции радиуса:

Расчёт точки пересечения в случае внутренних углов: $180^\circ < \alpha < 360^\circ$

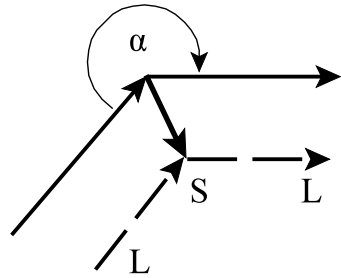
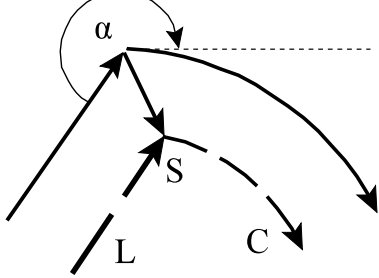
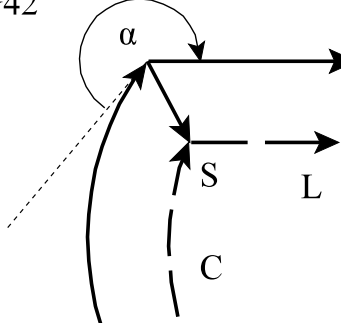
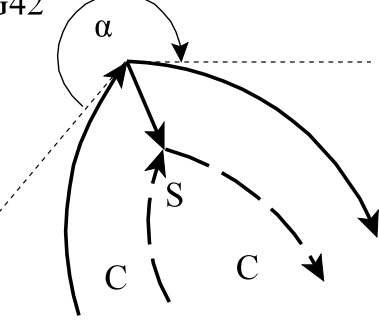
прямая - прямая	прямая - окружность
G42 	G42 
окружность - прямая	окружность - окружность
G42 	G42 

рисунок 15.6.2-1

Обход внешних углов с тупым углом: $90^\circ \leq \alpha \leq 180^\circ$

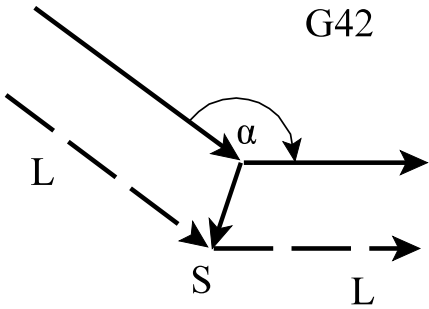
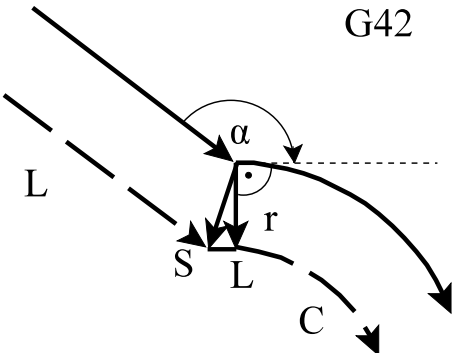
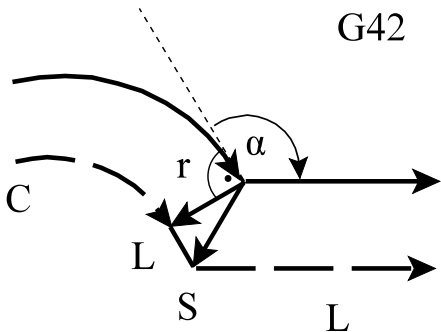
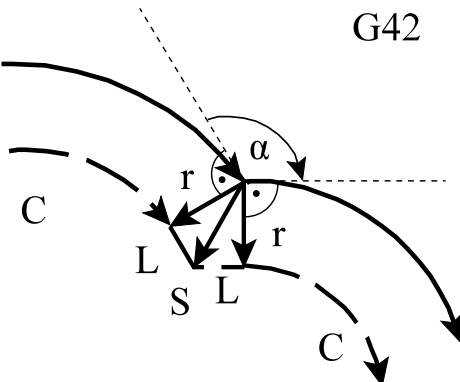
прямая - прямая	прямая - окружность
G42 	G42 
окружность - прямая	окружность - окружность
G42 	G42 

рисунок 15.6.2-2

Обход внешних углов с острым углом: $0^{\circ} \leq \alpha < 90^{\circ}$

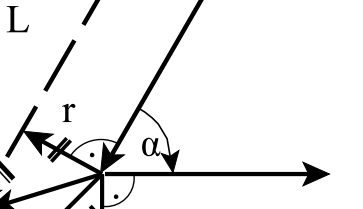
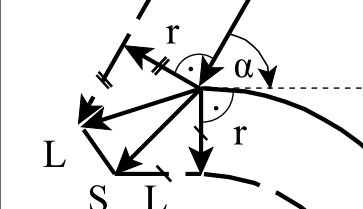
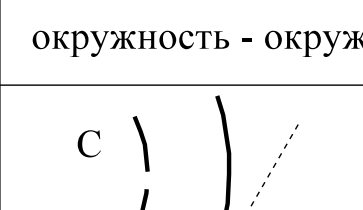
прямая - прямая  G42	прямая - окружность  G42
окружность - прямая 	окружность - окружность 

рисунок 15.6.2-3

Если нет точки пересечения в случае внутренних углов

Бывает, что при некоторых значениях радиуса инструмента не получается точки пересечения. При этом устройство ЧПУ остановится во время выполнения предыдущего кадра и выводится сообщение об ошибке 2047 Нет точки пересечения G41, G42.

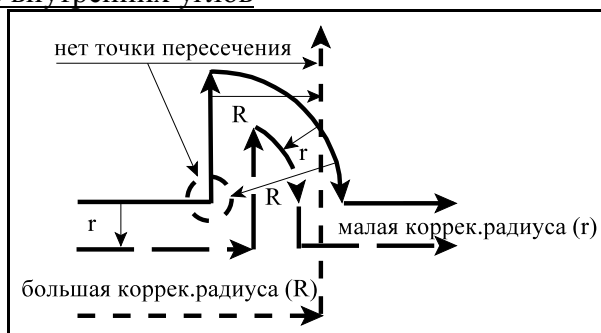


рисунок 15.6.2-4

В выбранной плоскости получится ноль перемещения

В том случае, если во включенном состоянии G41, или G42, в выбранной плоскости в одном из кадров запрограммировано ноль перемещения, или получается ноль перемещения, как в кадре N120 примера, происходит следующее. В конечную точку предыдущего кадра (N110) и в начальную точку последующего кадра (N130) установится перпендикулярный вектор, длина которого совпадает коррекцией радиуса, затем соединяет два вектора линейной интерполяцией. Такими случаями надо быть осторожно потому, что может причинять ненамеренное подрезание, в случае окружности - искажение.

Например:

```
... G91 G18 G42 ...
N110 G1 X100 Z40
N120 Z0
N130 Z90
N140 X-40 Z50
...
```

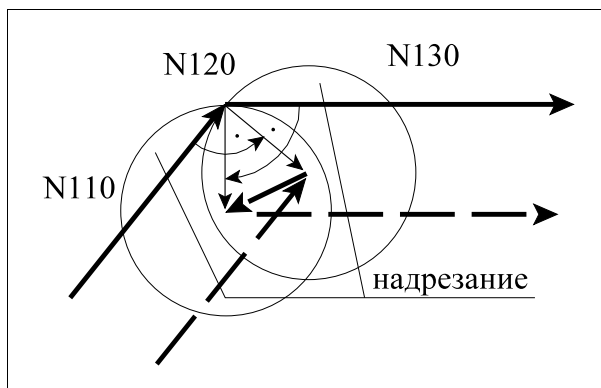


рисунок 15.6.2-5

Коррекция радиуса в случае винтовой линии и окружности с переменным радиусом

Если применить коррекцию радиуса к винтовой линии, или окружности с переменным радиусом, в начальной точке окружности устройство ЧПУ рассчитывает вектор(ы) коррекции под такую воображаемую окружность, радиус которой совпадает с радиусом начальной точки запрограммированной окружности (на рисунке $R1=50$), а начальная точка которой совпадает с запрограммированной точкой центра ($X0, Z0$). В конечной точке окружности вектор(ы) коррекции рассчитываются под такую воображаемую окружность, радиус которой совпадает с радиусом в конечной точки запрограммированной окружности ($R2=20$), а точка центра совпадает с точкой центра запрограммиро-

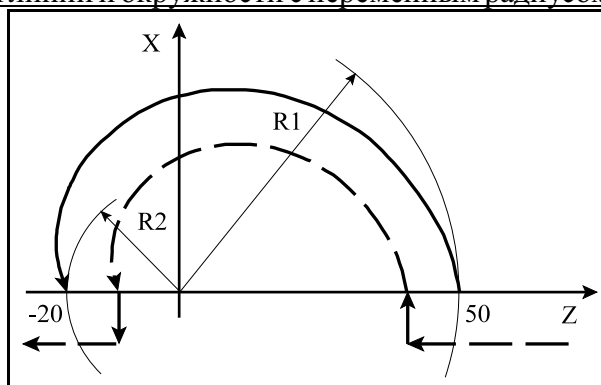


рисунок 15.6.2-6

ванной окружности.

Пример:

```
G0 G18 G41
G1 Z50
G3 Z-20 K-50 L1
G1 Z-30
...
```

Пропускание отрезков обхода

В случае обхода углов с тупым и острым углом может образоваться два или больше векторов коррекции. Если их конечные точки попадают близко друг к другу, или получается движения между двумя точками.

В том случае, если расстояние между двумя векторами на обеих осях меньше, чем установленное параметром N1405 DELTV значение, показанное на рисунке вектор пропускается, и траектория инструмента изменяется согласно рисунку.

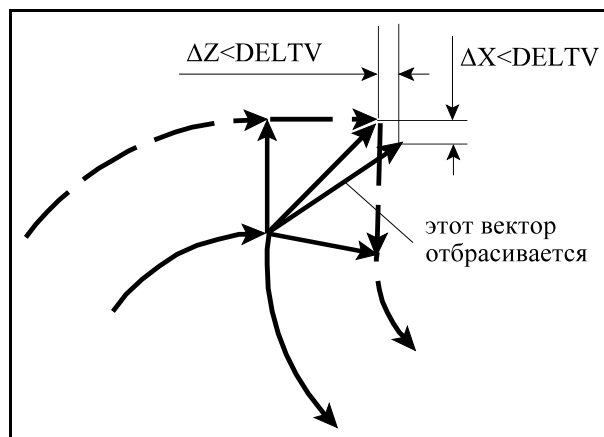


рисунок 15.6.2-7

☞ **Замечание:** При необоснованно большом значении параметра *DELTV* при обходе внешних углов с острыми углами может случиться, что инструмент повреждает угол.

В выбранной плоскости, через несколько кадров нет команды движения

Для того, чтобы устройство ЧПУ правильно обращалось коррекцией радиуса инструмента, например, чтобы сумело рассчитывать точку пересечения между конечной точкой прочитанного кадра и начальной точкой последующего кадра контура, **кадры необходимо вперёд прочитать и обработать их**. Обработанные вперёд кадры попадают в буфер кадров.

На практике может понадобиться, что между двумя плоскостными кадрами движения запрограммировать кадр, не содержащий движения, или кадр, содержащий движение, лежащее не в выбранной плоскости. Эти могут быть например, следующие:

функции: M, S, T

задержка: G4 P

движение вне выбранной плоскости: (G18) G1 Y

вызов подпрограммы: M98 P, и т. д.

Это означает и то, что между двумя кадрами движения, принадлежащими к выделённой плоскости, имеются прочие кадры, например, функции, задержка, движение вне плоскости и т. д., расчёт коррекции всё равно будет непрерывным, до тех пор, пока не заполнится буфер кадров.

Если **заполнился буфер**, устройство ЧПУ выводит сообщение **в начале последнего кадра движения, принадлежащего к выделённой плоскости**

2090 Расчёт коррекции радиуса нельзя продолжать. Буфер полный.

Если в состоянии G41, G42 запрограммировать функцию разгрузки буфера

Непрерывность расчёта коррекции, *прочтение вперёд кадров прерывают* такие коды G и функции (например, некоторые функции M и т. д.), которые вынуждают *разгрузку буфера*

Когда в ходе прочтения устройство ЧПУ прочитает такой код, отложит прочтение дальнейших кадров и дожидается, чтобы разгрузился буфер кадров, то есть все кадры в буфере были выполнены. Это имеет такое следствие, что устройство ЧПУ *отложит расчёт коррекции радиуса*. После этого выполняет функцию, затем после этого начинает прочтение и буферирование кадров.

Коды G, отложившие расчёт коррекции радиуса, следующие:

G22, G23,
G54-G59, G54.1, ...,
G52, G92,
G53,
G28, G30

Функция T также, относящиеся хоть к смене инструментов, хоть к вызову коррекции, отложат расчёт коррекции:

Tnnkk, Tnn00, Tkk

Функция M, отложившие расчёт коррекции радиуса следующие:

M0, M1, M2, M30

Помимо перечисленных выше функции M, можно выделить параметрами и функции разгрузки буфера M и группы кодов M. Кроме функций M можно выделить для разгрузки буфера и S, T, и смешанные функции (A, B, C, U, V, W).

Строителем станка определяет, что какие функции выполняют разгрузку буфера и об этом сообщает в описании станка!

В том случае, если во включённом состоянии G41, или G42 запрограммировать между двумя кадрами движения указанные выше коды G, H, или функции, в конечной точке предыдущего кадра удаляет вектор коррекции, выполняет команду, затем в конечной точке следующего кадра движения восстанавливает его.

Например:

```
...G91 G18 G41...
N110 G1 X-100 Z80
N120 G92 X0 Z0
N130 X100 Z80
...
```

В примере выше из-за кода G92 разгружается буфер. Ситуация подобная и в случае остальных кодов разгрузки буфера.

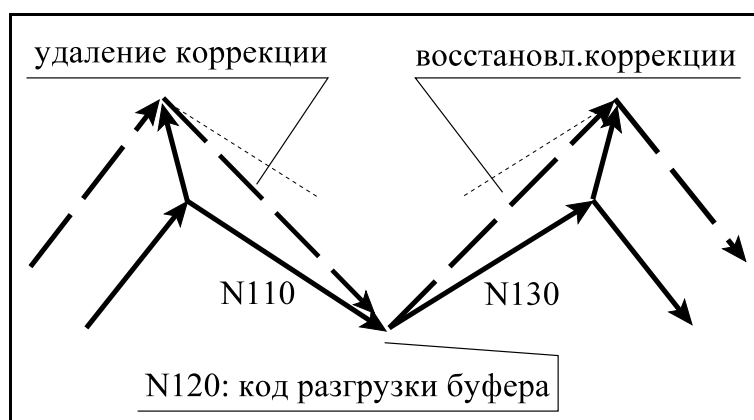


рисунок 15.6.2-8

Если в состоянии G41, G42 запрограммировать условный пропуск кадра

При состоянии бита #4 **CBV=0** параметра N1337 Execution Config, команда условного кадра (кадры, начинающиеся со знаком /) *подавляет* прочтение вперёд кадра. В этом случае

при **G41, G42** контур *искажается*, однако, достаточно **включить** включатель условного кадра *во время выполнения предыдущего кадра*, чтобы действовал.

При состоянии бита **#4 CBV=1** параметра N1337 Execution Config, команда условного кадра (кадры, начинающиеся со знаком /) *не подавляет* прочтение вперёд кадра. В этом случае при **G41, G42** контур *не искажается*, однако, включатель условного кадра для обеспечения надёжного действия *необходимо установить перед выполнением программы*. Действие установки параметра смотри в главе Пропуск условного кадра.

Пример:

```
...G91 G18 G41...
N110 G1 X-100 Z80
/ N120 S2500
N130 X100 Z80
...
```

В примере выше кадр N120 условный.

В случае CBV=0 в конечной точке кадра N110 удаляет коррекцию и в кадре N130 восстанавливает его.

При положении CBV=1 прочтение вперёд не отложится, расчёт коррекции непрерывным будет.

В состоянии G41, G42 желательно избегать программирования условного кадра.

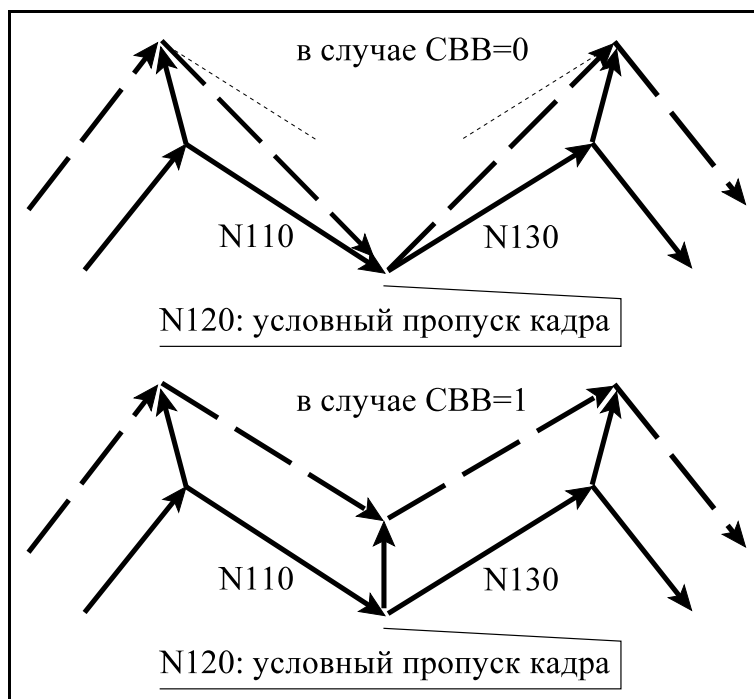


рисунок 15.6.2-9

15.6.3 Выключение расчёта коррекции радиуса. Отход от контура.

Команда G40 выключает расчёт коррекции радиуса инструмента. Команду G40 можно выдавать только линейной интерполяцией. Если в периоде запрограммировать G40, *устройство ЧПУ выводит сообщение об ошибке 2043 G40 в круговой интерполяции.*

Основные случаи выключения коррекции радиуса:

(G42)	(G42)
G01 X_ Z_	G02 X_ Z_ R_
G40 X_ Z_	G40 G1 X_ Z_

Выход из внутреннего угла: $180^\circ < \alpha < 360^\circ$

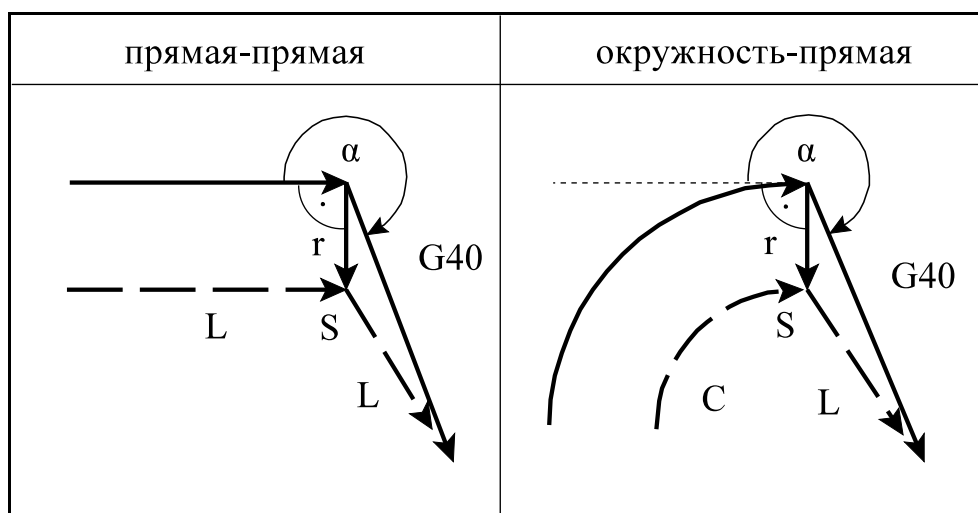


рисунок 15.6.3-1

Выход из внешнего угла под тупым углом: $90^\circ \leq \alpha \leq 180^\circ$

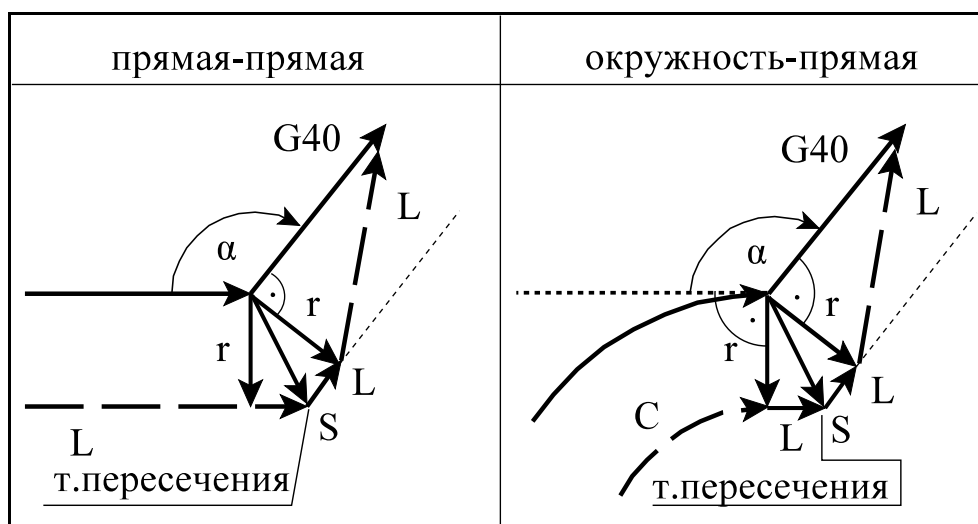


рисунок 15.6.3-2

Выход из внешнего угла под острым углом: $0^\circ \leq \alpha < 90^\circ$

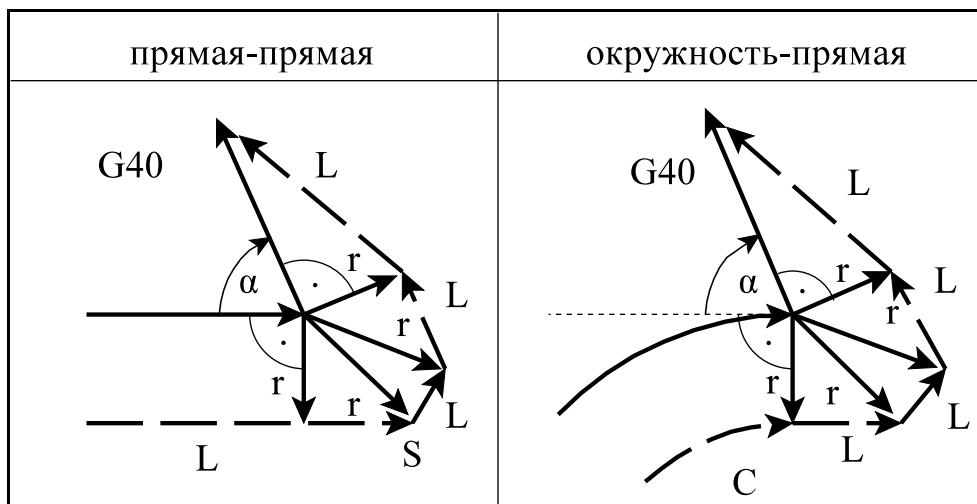


рисунок 15.6.3-3

Пример

Программа ниже показывает пример для правильного выключения коррекции радиуса, при позиции инструмента Q3.

Кадр N180 выключает коррекцию радиуса. В конец кадра N170 наводится по радиусу инструмента, теоретическая вершина инструмента нависает над конечной точкой, затем к концу кадра N180 уже теоретическая вершина инструмента будет в координатах конечной точки:

```

G54 G18
...
N50 G0 X200 Z50
N60 G92 S3000
N70 G96 S600
N80 T101 M3
N90 X160 Z3
N100 X90
N110 G1 G42 Z0
N120 X100
N130 X110 Z-5
N140 Z-25
N150 G2 X130 Z-35 R10
N160 G1 X140
N170 Z-45
N180 G40 X160
N190 G0 X200 Z50
...

```

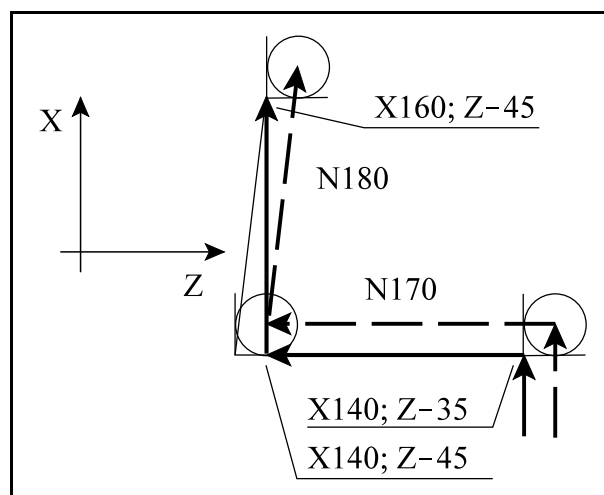


рисунок 15.6.3-4

Выключение коррекции радиуса без программирования движения:

Если **выключение коррекции** (G40) выполняется в таком кадре, в котором **не фигурирует ни одного адреса оси, принадлежащего к выбранной плоскости**, как в примере ниже в кадре N190 не задан адрес ни X, ни Z, тогда в кадре N190 не выполняется движение. В кадре N180 сперва радиус инструмента наводится в конечную точку, затем отводя отступает расстояние, равное радиусу инструмента.

Исправить можно объединением кадра N190 и N200.

```
G54 G18
...
N50 G0 X200 Z50
N60 G92 S3000
N70 G96 S600
N80 T101 M3
N90 X160 Z3
N100 X90
N110 G1 G42
N120 Z0
N130 X100
N140 X110 Z-5
N150 Z-25
N160 G2 X130 Z-35 R10
N170 G1 X140
N180 Z-45
N190 G40
N200 X160
N210 G0 X200 Z50
...
```

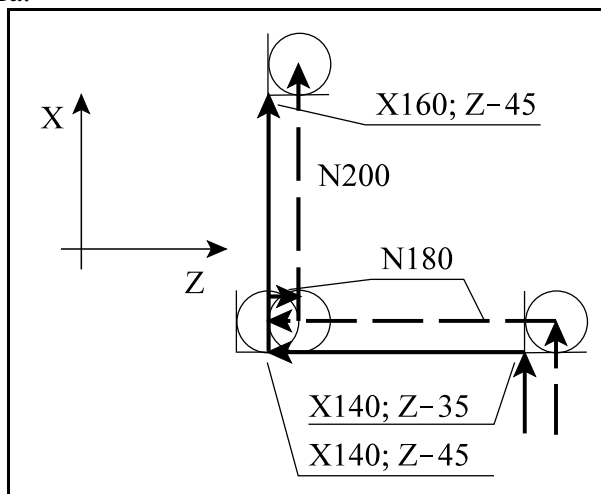


рисунок 15.6.3-5

Если **выключение коррекции** (G40) выполняется в таком кадре, в котором **нет перемещения ни на одной оси, принадлежащей к выбранной плоскости**, как в примере ниже в кадре N190 правда заполнен адрес X, но из-за принадлежащего к нему инкремента 0 нет перемещения, тогда происходит то же самое, как в случае выше, когда не запрограммирован адрес оси.

```
G54 G18
...
N50 G0 X200 Z50
N60 G92 S3000
N70 G96 S600
N80 T101 M3
N90 X160 Z3
N100 X90
N110 G1 G42 G91 Z0
N120 G90 Z0
N130 X100
N140 X110 Z-5
N150 Z-25
N160 G2 X130 Z-35 R10
N170 G1 X140
```

```

N180 Z-45
N190 G91 G40 X0
N200 G90 X160
N210 G0 X200 Z50
...

```

Выключение коррекции радиуса с расчётом точки пересечения

Если в кадре (G40), выполняющем выключение коррекции присвоить значение для I, J, K, но только для лежащих в выбранной плоскости (например: в случае G18 для I, K), тогда устройство ЧПУ наводится в точку пересечения между предыдущим кадром и прямой, определённой с помощью I, J, K. Значение I, J, K всегда инкрементное, и заданное ими вектор показывает вниз от конечной точки предыдущего кадра.

Эта возможность полезна например, при выходе из внутреннего угла.

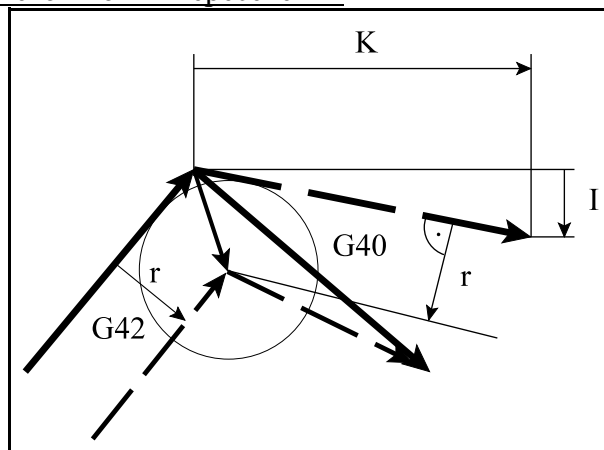


рисунок 15.6.3-6

Пример:

Фрезеровать шестигранник изнутри с длиной грани 100 мм. Выходить с инструментом из угла, лежащего в точке с координатой X100 Y0. Кадр N120 приложенной программы с расчётом точки пересечения отходит с контура, заданием I и J. Координаты I и J рассчитываются на основании перемещения, выполненного в кадре N60 программы.

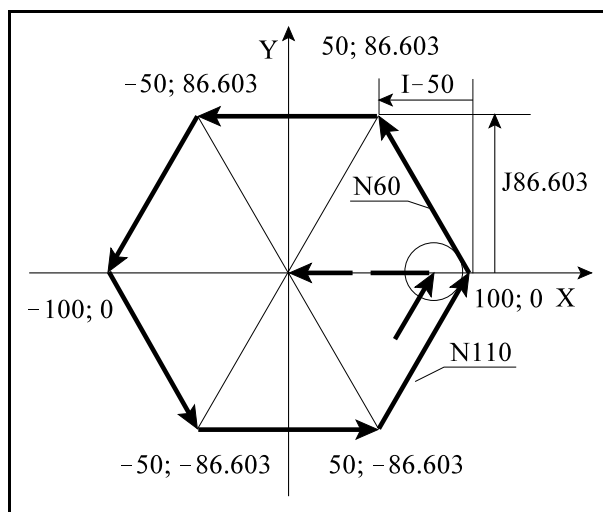


рисунок 15.6.3-7

```

N10 G54 G17 M3 S300
N20 G0 X0 Y0 Z100
N30 G43 Z5 H1
N40 G1 Z-5 F1000
N50 G41 G0 X100 Y0 I50 J86.603 D1
N60 G1 X50 Y86.603
N70 X-50
N80 X-100 Y0
N90 X-50 Y-86.603
N100 X50
N110 X100 Y0
N120 G40 G0 X0 I-50 J86.603
N130 M30

```

Основные случаи отхода с точкой пересечения

В случае задания I, J, K устройство ЧПУ рассчитывает всегда точку пересечения, независимо от того, что обрабатывается внутренний, или внешний угол.

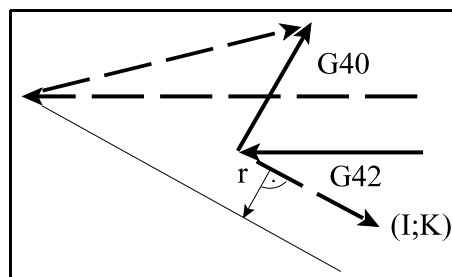


рисунок 15.6.3-8

Если не находит точку пересечения, наводится перпендикулярно в конечную точку предыдущего кадра

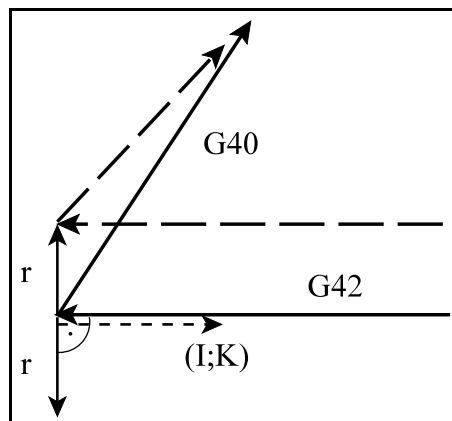


рисунок 15.6.3-9

15.6.4 Перемена направления в расчёте коррекции радиуса.

Направление расчёта коррекции радиуса инструмента, то есть направление слежения за контуром ихложены в таблице:

	коррекция радиуса: положительная	коррекция радиуса: отрицательная
G41	слева	справа
G42	справа	слева

Направление слежения за контуром обратимое и во включенном состоянии расчёта коррекции радиуса инструмента. Это может происходить с программированием G41, или G42, или вызовом коррекции радиуса инструмента по адресу D со знаком, противоположным предшествующему. Когда повернётся направление слежения за контуром, устройство ЧПУ не рассматривает, что находится ли "наружу", или "внутри", а за первый шаг всегда рассчитывает точку пересечения. На рисунках ниже предполагаем положительный радиус инструмента и переключение из G42 в G41:

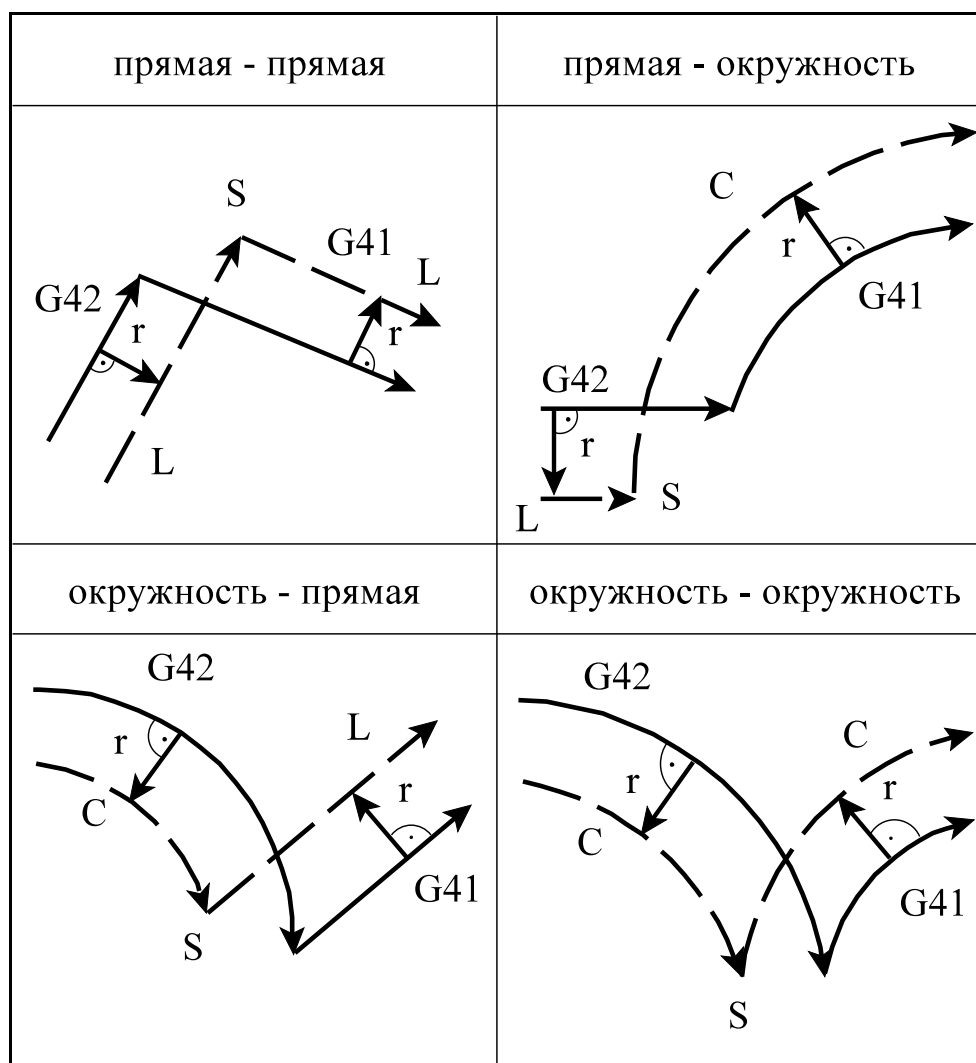


рисунок 15.6.4-1

Пример

Наводка на окружность с радиусом R50 вдоль прямой, слева, с помощью G41 кадром N30 и N40. После этого следим за окружностью справа в кадре N50 с помощью G42. Отход происходит также вдоль прямой, слева, с помощью G41.

```

...
N10 G17 G0 X100 Y0
N20 T101 M3 S500
N30 G41 X80
N40 G1 X50
N50 G42 G3 I-50
N60 G41 G1 X80
N70 G40 X100
...

```

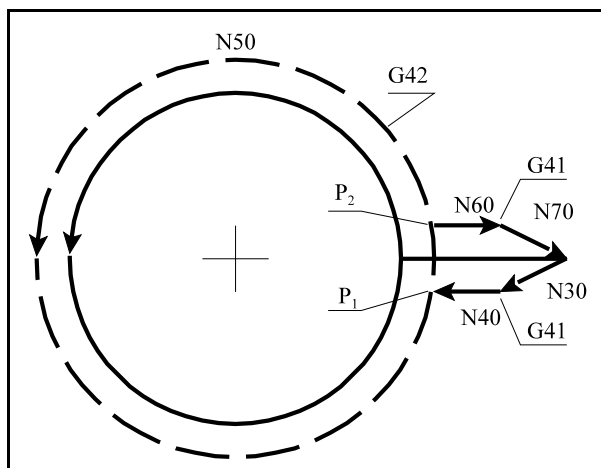


рисунок 15.6.4-2

Если запрограммировать переменную направления на полной окружности

Программируя полную окружность с переменной G41 - G42, может получиться много таких случаев, когда траектория инструмента совершает больше одного полного оборота.

Пример выше показывает этот случай.

Точка центра инструмента совершает одну полную дугу окружности с точки P_1 до точки P_1 , затем дугу с точки P_1 до точки P_2 .

Если нет точки пересечения при переходе прямая- прямая

Если при переходе прямая-прямая не получится точки пересечения, траектория инструмента видна на приложенном рисунке. Перемена направления совершается согласно рисунку с G42 на G41.

При этом конечной точкой будет вектор, в конечной точке кадра, предшествующего перемене направления, установленный перпендикулярно в начальной точке следующей прямой, с размером, равным радиусу инструмента.

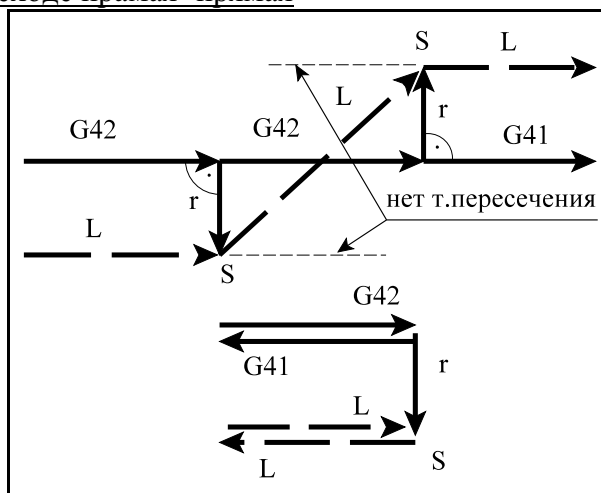


Рисунок выше показывает случай, если не меняется направление траектории, а ниже случай, если перемена направления 180 градусов.

рисунок 15.6.4-3

Если нет точки пересечения при переходе прямая - окружность

Если при переходе прямая - окружность не получится точки пересечения, траектория инструмента видна на приложенном рисунке. Перемена направления совершается согласно рисунку с G41 на G42.

При этом конечной точкой будет вектор, в конечной точке прямой, предшествующей перемене направления, установленный перпендикулярно в начальной точке следующей окружности, с размером, равным радиусу инструмента.

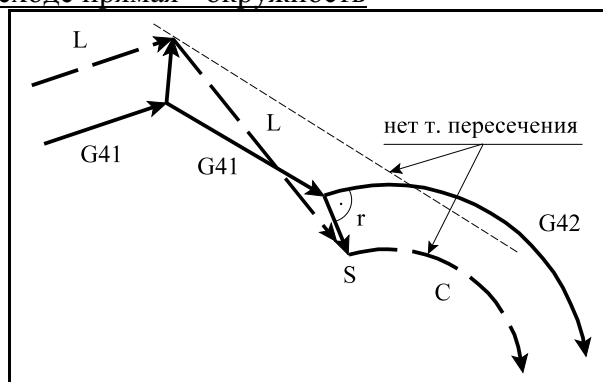


рисунок 15.6.4-4

Если нет точки пересечения при переходе окружность - прямая, или окружность - окружность

Если при переходе окружность - прямая, или окружность - окружность не получится точки пересечения, тогда соединяется конечная точка вектора коррекции, полученная в начальной точке первого периода и конечная точка вектора коррекции, установленного перпендикулярно в начальной точке второго кадра с некорректированной запрограммированной дугой окружности с радиусом R . При этом точка центра соединяющей дуги окружности не будет совпадать с точкой центра запрограммированной дуги окружности. Если перемену направления нельзя выполнить даже с вышеописанным переставлением точки центра окружности, устройство ЧПУ выводит сообщение об ошибке.

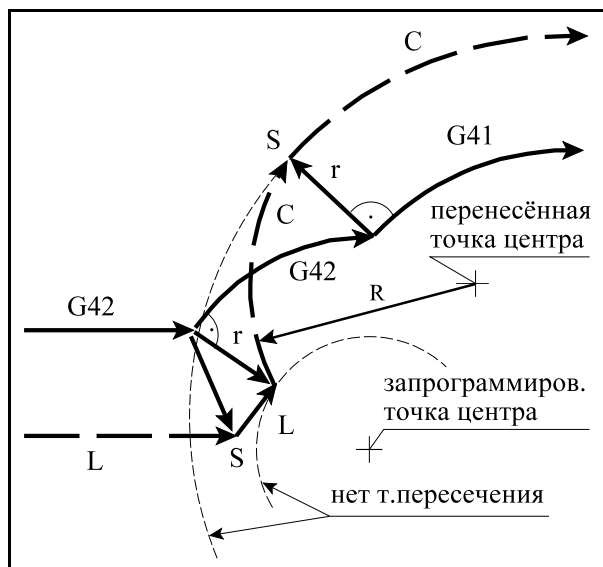


рисунок 15.6.4-5

15.6.5 Программирование сохранения вектора (G38)

Под действием команды

G38 v

устройство ЧПУ во включенном состоянии расчёта плоскостной коррекции радиуса инструмента *сохраняет последний вектор коррекции между предшествующим кадром и кадром G38 и задействует его в конце кадра G38*, независимо от перехода между кадром G38, и последующим кадром.

Код G38 является одноразовым, то есть не передаётся модально. Если необходимо сохранить вектор в нескольких кадрах, последующих друг за другом, G38 надо снова запрограммировать.

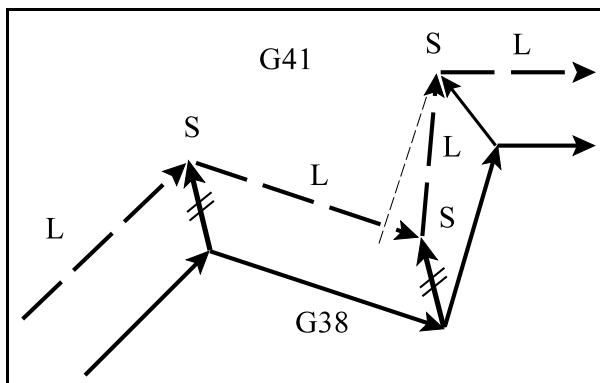


рисунок 15.6.5-1

Пример

Запрограммировать врезание без выключения слежения за контуром:

```
...G18 G42 G91...
N110 G1 Z40
N120 G38 Z50
N130 G38 X140
N140 G38 X-140
N150 Z60
...
```

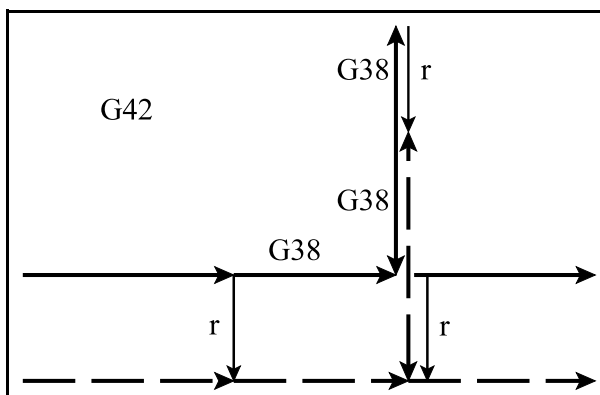


рисунок 15.6.5-2

15.6.6 Программирование угловой дуги (G39)

Программированием кадра

G39 (I J K)

во включенном состоянии расчёта плоскостной коррекции радиуса инструмента можно достигать, чтобы в случае обхода внешних углов устройство ЧПУ не рассчитывало автоматически точку пересечения, или не вставило прямые участки для обхода, а точка центра инструмента перемещалась по дуге окружности, равной радиусу инструмента.

В состоянии G41 *вставляется окружность с радиусом, равным радиусу инструмента* по направлению G02, в состоянии G42 - по направлению G03.

Начальную точку окружности даёт вектор, перпендикулярный в конечную точку траектории предыдущего кадра, с радиусом, равным радиусу инструмента, а конечную точку - вектор, перпендикулярный в начальную точку предшествующего кадра, с длиной, равной радиусу инструмента. **G39 следует программировать в отдельном кадре:**

```
...G18 G91 G41...
N110 G1 Z100
N120 G39
N130 G3 X-160 Z80 K80
...
```

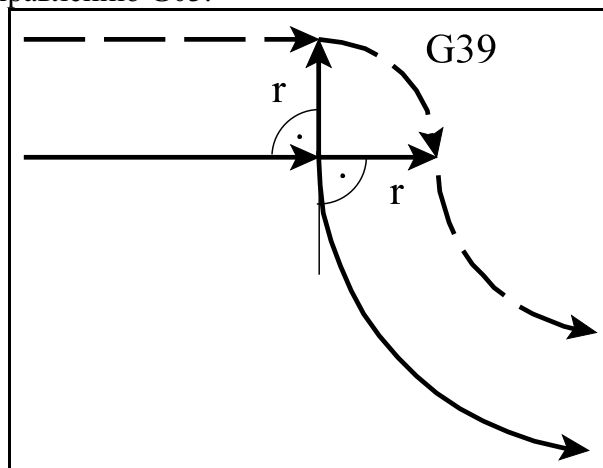


рисунок 15.6.6-1

Если в кадре G39 запрограммировать I, J или K, согласно выбранной плоскости, конечную точку окружности даёт вектор от конечной точки предшествующего кадра, перпендикулярный вектору, определённому через I, J или K, с длиной, равной радиусу инструмента:

```
...G18 G91 G41...
N110 G1 Z100
N120 G39 I-60 K50
N130 G40 X60 Z110
...
```

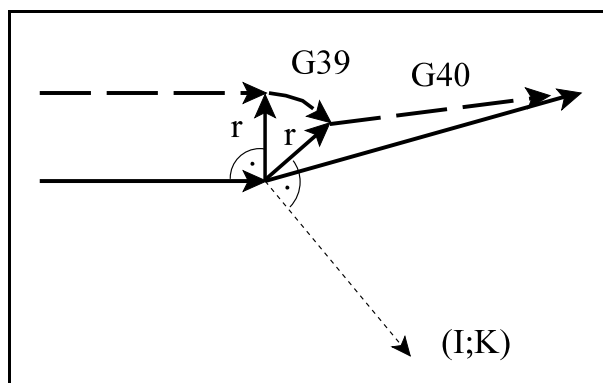


рисунок 15.6.6-2

Для вектора, определённого через I, J или K, действуют предварительно установленные команды для отражения, или для поворота. Команда масштабирования логично не влияет на направление.

В кадре типа G39 нельзя программировать никаких команд движения.

Пример

Фрезеровать паз треугольной формы, где углы надо закруглить.

Для того, чтобы контур был замкнутым, в кадре N110 задаётся G39 с помощью I, J. Данные I, J определяются из перемещения кадра N60.

```

...
N20 G0 G17 G40 G54
N30 X-30 Y150 Z5 M3 S300
N40 G42 X0 Y86.603 D1
N50 Z-2
N60 G1 X-50 Y0 F1000
N70 G39
N80 X50
N90 G39
N100 X0 Y86.603
N110 G39 I-50 J-86.603
N120 Z5
N130 X-40
N140 G40 Y120
...

```

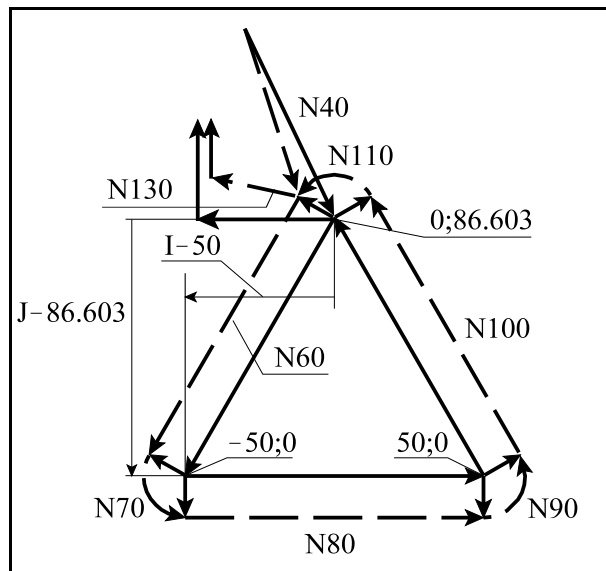


рисунок [15.6.6-3](#)

15.6.7 Проблемы помеха слежения за контуром. Проверка на интерференцию

В ходе выполнения слежения за контуром в множестве случаев бывает, что траектория инструмента будет противоположная запрограммированной траектории. В этом случае, *вопреки намерению программиста, инструмент может врезаться в заготовку*. Это явление называется *помехой слежения за контуром*, или *интерференцией*.

В случае, показанном на рисунке, после расчёта точек пересечения, в ходе выполнения кадра N2 получается траектория инструмента (штриховая линия), противоположная запрограммированной (сплошная линия). Инструмент врезается в заготовку.

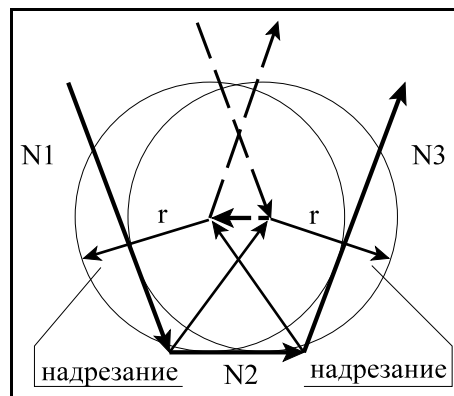


рисунок 15.6.7-1

Для избежания таких случаев, устройство ЧПУ выполняет проверку на интерференцию, при положении бита #0 IEN=1 параметра N1403 Interference. В этом случае *устройство ЧПУ проверяет, что для угла ϕ между запрограммированным перемещением и перемещением, скорректированным коррекцией радиуса, выполняется ли следующее условие: $-90^\circ \leq \phi \leq +90^\circ$* .

Иными словами, устройство ЧПУ проверяет, что имеет ли вектор скорректированного перемещения составляющую, противоположную по направлению запрограммированному вектору перемещения.

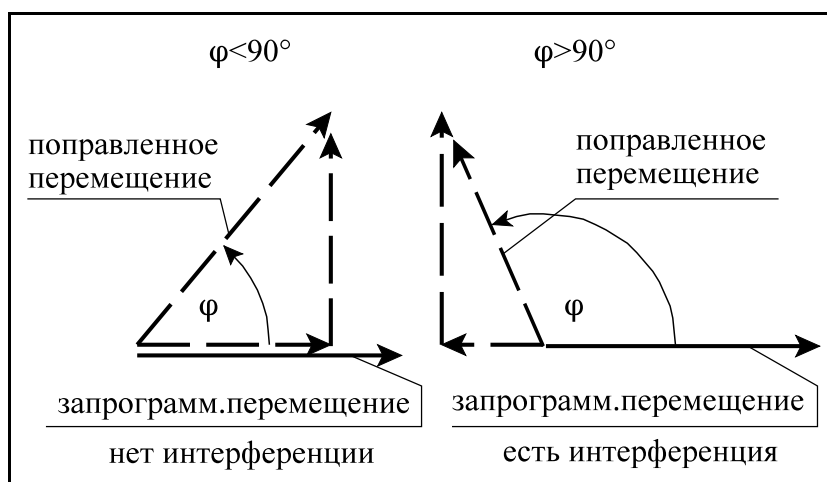


рисунок 15.6.7-2

Если устройство ЧПУ определяет ошибку по интерференции, в зависимости от положения бита #1 AAL параметра N1403 Interference *выводит сообщение об ошибке, или попытается исправить ошибку по интерференции*.

Если $AAL=1$, то устройство ЧПУ *всегда выводит сообщение об ошибке 2049 ошибка по интерференции*, в конце кадра, предшествующего кадра, вызывающего ошибку по интерференции.

Если $AAL=0$, то устройство ЧПУ *попыtetся автоматически исправить ошибку* и выводит сообщение об ошибке только тогда, если автоматическая коррекция не приводит к результату.

Для наблюдения интерференции, далее для автоматического исправления устройство ЧПУ в основном случае прочитает вперёд 3 кадра. Если значение параметра *NI404 ВК No. Interf* больше, чем 0, тогда прочитает вперёд число кадров, равное 3 + значению параметра и выполняет проверку на интерференцию.

Автоматическое исправление ошибки по интерференции

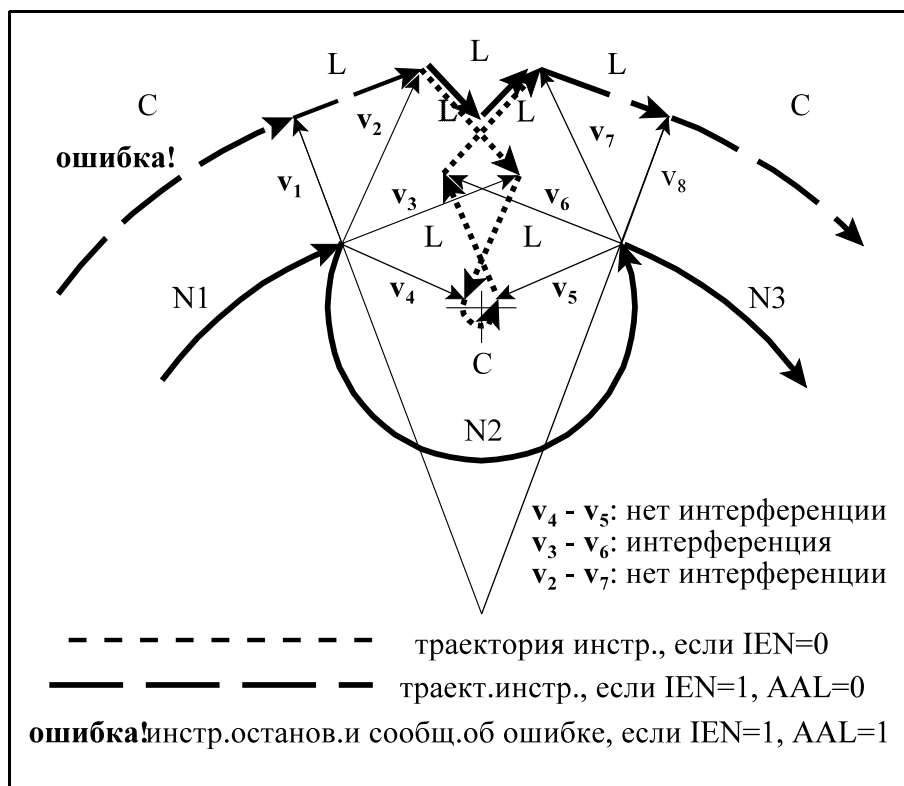


рисунок 15.6.7-3

В случае $AAL=0$, устройство ЧПУ не выводит сообщение об ошибке, а попытается автоматически исправить контур с той целью, чтобы избежать врезания. Ход коррекции следующий:

Коррекция радиуса инструмента (G41) включена в кадре N1, N2, и N3.

Векторы коррекции, рассчитанные между кадрами N1 и N2: $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$ és $\mathbf{v}_4$.

Векторы коррекции, рассчитанные между кадрами N2 и N3: $\mathbf{v}_5$, $\mathbf{v}_6$, $\mathbf{v}_7$ és $\mathbf{v}_8$.

Если между $\mathbf{v}_4$ и $\mathbf{v}_5$ имеется интерференция (получается перемещение, противоположное перемещению N2), рассчитываются точки пересечения между прямыми $\mathbf{v}_3$ и $\mathbf{v}_4$ и между $\mathbf{v}_5$ и $\mathbf{v}_6$, и пропускается дуга окружности N2

Если между $\mathbf{v}_3$ и $\mathbf{v}_6$ имеется интерференция (получается перемещение, противоположное перемещению N2), рассчитываются точки пересечения между прямыми $\mathbf{v}_2$ и $\mathbf{v}_3$ и между $\mathbf{v}_6$ и $\mathbf{v}_7$, и пропускаются перемещения между ними (рисунок выше показывает этот пример).

Если между $\mathbf{v}_2$ и $\mathbf{v}_7$ имеется интерференция (получается перемещение, противоположное перемещению N2), рассчитываются точки пересечения между прямыми $\mathbf{v}_1$ и $\mathbf{v}_2$ и между $\mathbf{v}_7$ и $\mathbf{v}_8$, и пропускаются перемещения между ними,

Если между v_1 и v_8 имеется интерференция, попытается рассчитывать точку пересечения

между кадрами N1 и N3..

Из примера выше видно, что выполнение кадра N1 начинается только тогда, если устройство ЧПУ завершило проверку на интерференцию на кадр N2. Однако, для этого надо зачитать в буфер и кадр N3, и при перехода N2 - N3 рассчитывать векторы коррекции.

Если в кадре N2 в результате расчёта точки пересечения между участками обхода нельзя устранить ошибки по интерференции в случае AAL=0, попробует рассчитывать точку пересечения между кадрами N1 и N3. Если существует точка пересечения, проходит дальше, если нет, выводится сообщение об ошибке 2049 Ошибка по интерференции.

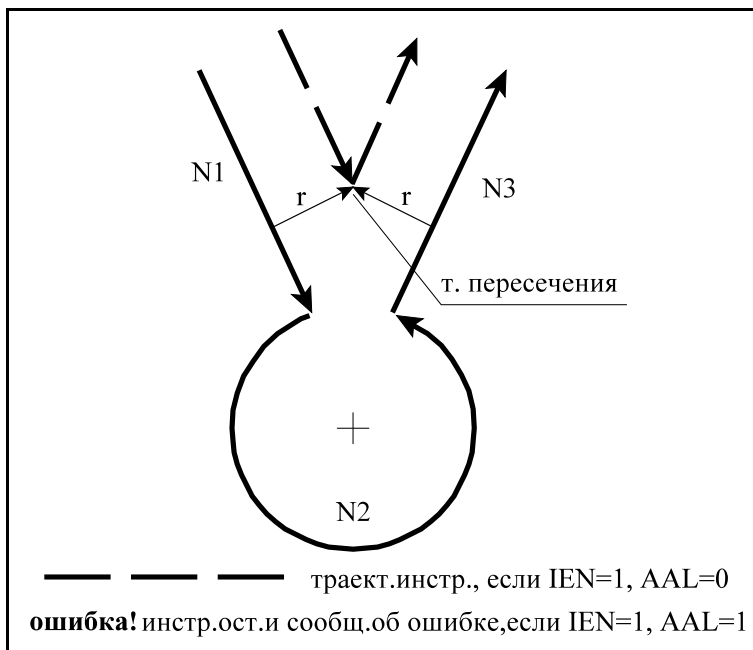


рисунок 15.6.7-4

Наблюдение за интерференцией на несколько кадров вперёд

В примере выше устройство ЧПУ проверит вперёд 3 кадра. Переход между N1 и N2, а также между N2 и N3. Это соответствует положению параметра N1404 BK No. Interf=0.

Значение параметра N1404 BK No. Interf может быть не более 8. При этом описанные выше проверки выполняются для кадров

между N1 и N2 далее N2 и N3

между N1 и N2 далее N3 и N4

...

между N1 и N2 далее N [BK No. Interf+2] и N [BK No. Interf+3] (для перемещения).

В зависимости от положения параметра AAL, после этого в кадре N1 сообщает об ошибке, или попытается исправить её.

Этим можно пользоваться тогда, если инструмент заходит в полость и надо проверить, что уместится ли в ней с данным диаметром.

Если желаем проверить устройством ЧПУ полость, показанную на рисунке, надо установить значение N1404 BK No. Interf=7.

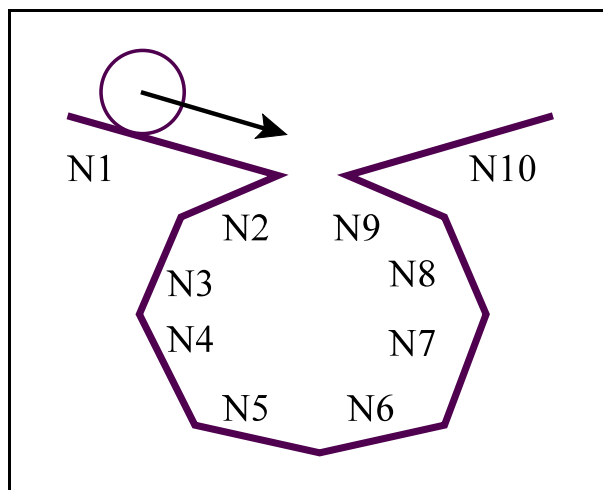


рисунок 15.6.7-5

Типичные ошибки по интерференции

Ниже перечислим несколько типичных случаев, когда устройство ЧПУ ощущает ошибки по интерференции.

Точение ступени, меньшей радиуса инструмента. Устройство ЧПУ в случае $IEN=0$ врезает в заготовку.

В случае $IEN=1$:

если $AAL=0$, рассчитывая точки пересечения между кадрами N1 и N3 избегает врезания,

если $AAL=1$, выводится сообщение об ошибке 2049 Ошибка по интерференции, так как врезает в заготовку.

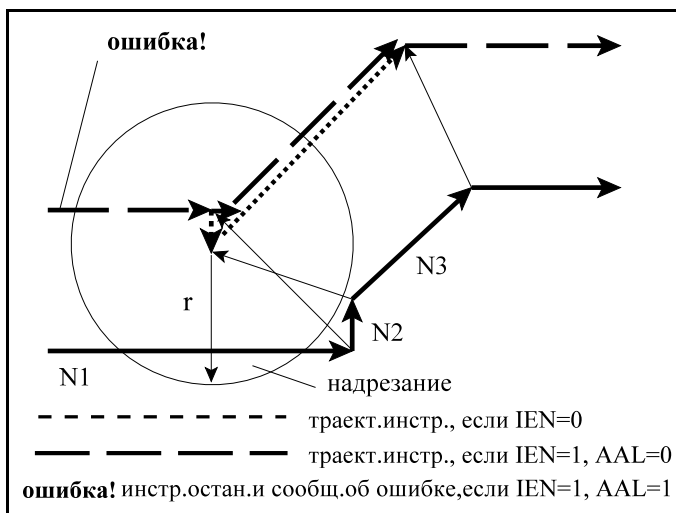


рисунок 15.6.7-6

При обработке внутреннего угла, с радиусом, меньше радиуса инструмента. Устройство ЧПУ в случае $AAL=1$ выводит сообщение об ошибке 2049 Ошибка по интерференции.

В случае $AAL=0$, пропуская окружность, рассчитывая точку пересечения между двумя прямыми, исправит ошибку.

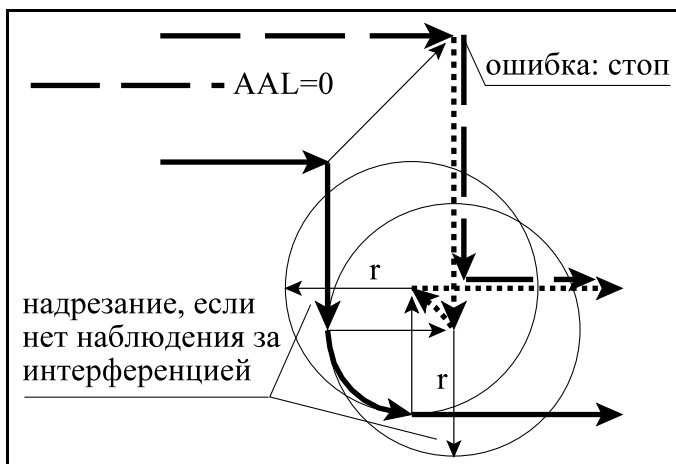


рисунок 15.6.7-7

Точение ступени, меньшей радиуса инструмента вдоль дуги окружности.

Если $AAL=0$, устройство ЧПУ рассчитывает точку пересечения между прямой L_1 и прямыми, соединяющими вектор v_1 и v_3 , чтобы избежать врезания.

Если $AAL=1$, тогда устройство ЧПУ выводит сообщение об ошибке 2049 Ошибка по интерференции, и остановится в предшествующем кадре.

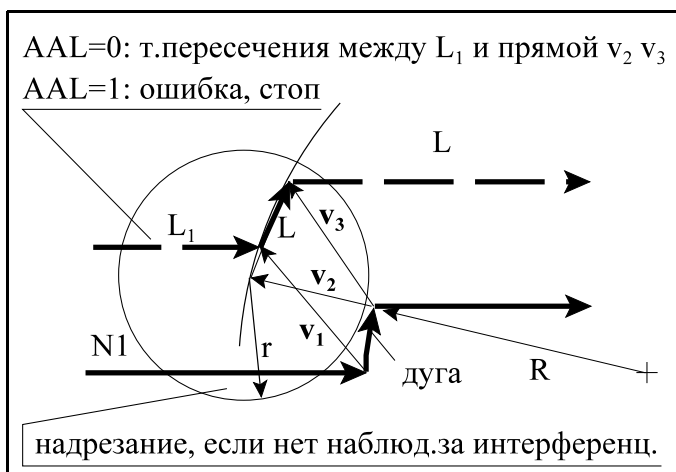


рисунок 15.6.7-8

В случае траектории, показанной на приведённом рисунке, в кадре N2 перемещение по скорректированной траектории противоположное запрограммированному.

Устройство ЧПУ не может поправить, если значение параметра N1404 BK No. Interf равно 0, так как между кадрами N1 и N3 нет точек пересечения, поэтому и в случае AAL=0 и AAL=1 выводит сообщение об ошибке.

Если значение **параметра N1404 BK No. Interf больше, чем 0**, устройство ЧПУ дальше прочитает кадры, поэтому **исправит ошибку по интерференции** с точкой пересечения между N1 и N4, пропуская кадры N2, N3.

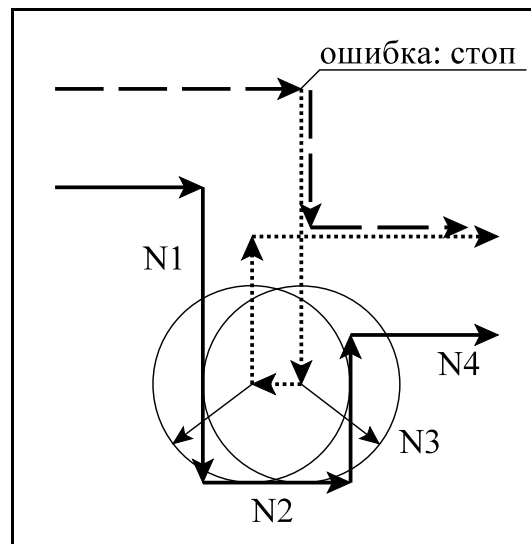


рисунок 15.6.7-9

В случае траектории, показанной на приведённом рисунке, в кадре N2 перемещение по скорректированной траектории было бы противоположное запрограммированному.

В случае AAL=1 в начальной точке кадра N1 выводится сообщение об ошибке.

В случае AAL=0 рассчитывается точка пересечения между кадрами N1 и N3 и проходит до точки пересечения. При дальнейшем перемещении, от точки пересечения получилось бы перемещение, а противоположное кадру N3, по скорректированной траектории, поэтому выводится сообщение об ошибке, если значение параметра N1404 BK No. Interf равно 0.

Если значение **параметра N1404 BK No. Interf больше, чем 0**, устройство ЧПУ прочитает дальше кадры, поэтому **ошибки по интерференции поправит** с точкой пересечения между N1 и N4, пропуская кадры N2, N3.

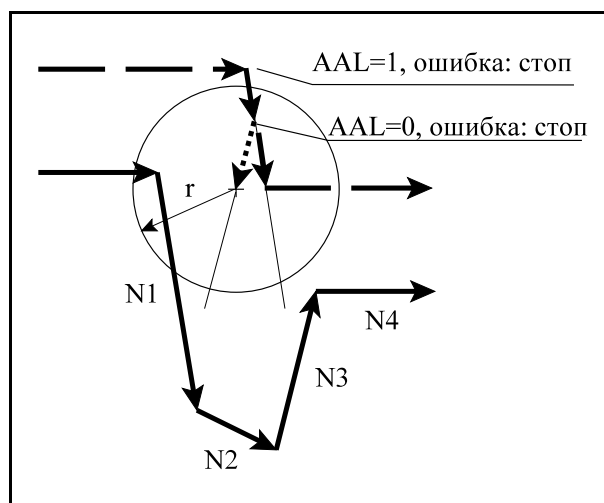


рисунок 15.6.7-10

Сообщает об ошибке по интерференции, но не врезался бы в заготовку

Бывают случаи, когда инструмент действительно не врезался бы в заготовку, но выводится сообщение об ошибке проверки по интерференции.

Если обрабатывать углубление, меньшее коррекции радиуса, в действительности может быть не происходит врезания, как это видно и на рисунке, но устройство ЧПУ выводит сообщение об ошибке 2049 Ошибка по интерференции в случае $AAL=1$, так как в кадре N3 направление перемещения по скорректированной траектории противоположное запрограммированному.

В случае $AAL=0$ согласно рисунку, пропуская кадры N2, N3, N4, соединяя кадры N1 и N5 продолжается обработка.

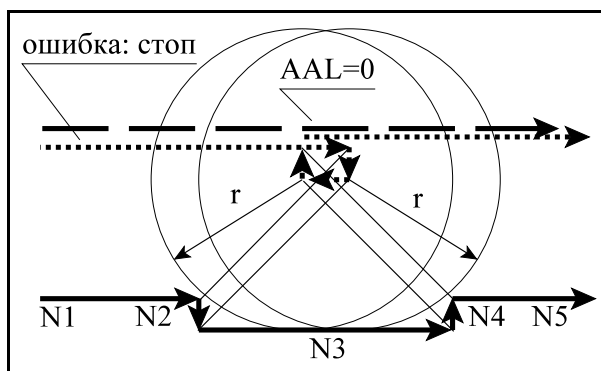


рисунок 15.6.7-11

В примере на приложенном рисунке также выводится сообщение об ошибке по интерференции, поскольку в кадре N2 перемещение по скорректированной траектории противоположное запрограммированному.

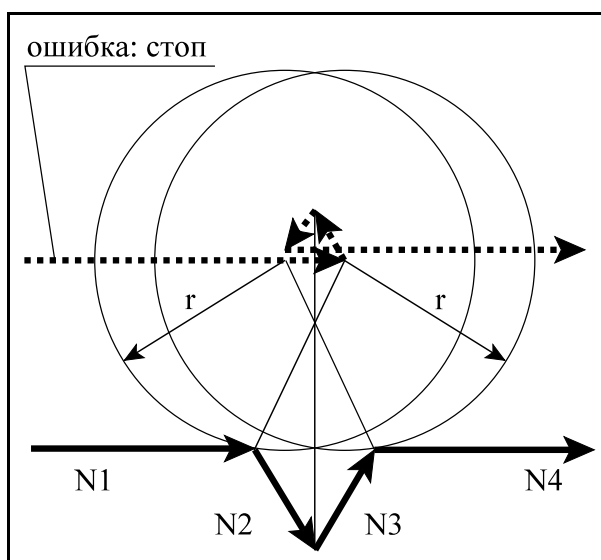


рисунок 15.6.7-12

Вопреки наблюдения за интерференцией, врезается в заготовку

Бывают такие случаи, получаемые из геометрии траектории, когда вопреки наблюдения за интерференцией врезается в заготовку. Ниже перечислим несколько из случаев.

В случае, показанном на приложенном рисунке, в кадре N3 получается перемещение, противоположное по направлению запрограммированному, поэтому в начальной точке кадра N2 выводится сообщение об ошибке по интерференции и останавливается, если значение параметра N1404 BK No. Interf равно 0. Исходя из геометрии (G2) кадра N4 врезается в заготовку.

Если значение параметра N1404 BK No. Interf больше, чем 0, устройство ЧПУ прочитает дальше кадры, поэтому **исправит ошибку по интерференции** с точкой пересечения между N1 и N4, пропуская кадры N2, N3, если AAL=0, впрочем выводит сообщение об ошибке в начале кадра N1.

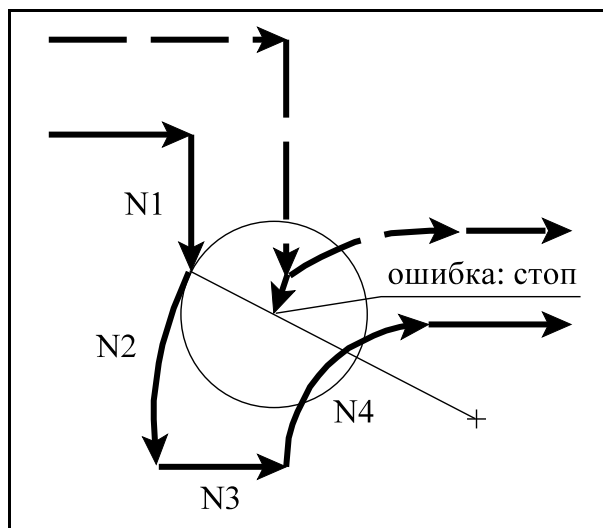


рисунок 15.6.7-13

В случае, показанном на приложенном рисунке, ни в одном кадре из N1, N2, N3 не создаётся перемещение, противоположное запрограммированному, поэтому устройство ЧПУ не выводит сообщение об ошибке по интерференции, если значение параметра N1404 BK No. Interf равно 0, всё таки исходя из геометрии траектории врезает в заготовку.

Если значение параметра N1404 BK No. Interf больше, чем 0, устройство ЧПУ прочитает дальше кадры, поэтому **исправит ошибку по интерференции** с точкой пересечения между N1 и N4, пропуская кадры N2, N3, если AAL=0, впрочем выводит сообщение об ошибке.

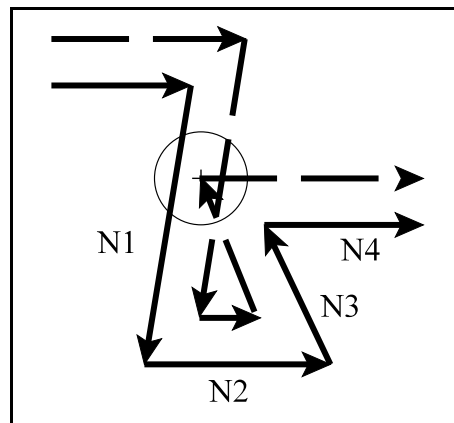


рисунок 15.6.7-14

В случае траектории, показанной на приведённом рисунке, в кадре N2 перемещение по скорректированной траектории было бы противоположное запрограммированному. В случае AAL=1 в начальной точке кадра N1 выводится сообщение об ошибке.

В случае AAL=0 рассчитывается точка пересечения между кадрами N1 и N3 и доходит до точки пересечения. Если бы дальше ходило от точки пересечения, получилось бы движение противоположное кадру N3, поэтому выводится сообщение об ошибке. Однако, в конечной точке кадра уже получится врезание, если значение параметра N1404 BK No. Interf равно 0.

Если значение параметра N1404 BK No.

Interf равно 1, выводится сообщение об ошибке по интерференции.

Если **значение параметра N1404 BK No. Interf равно 2**, рассчитывая точку пересечения между кадрами N1 и N5, исправляет ошибку.

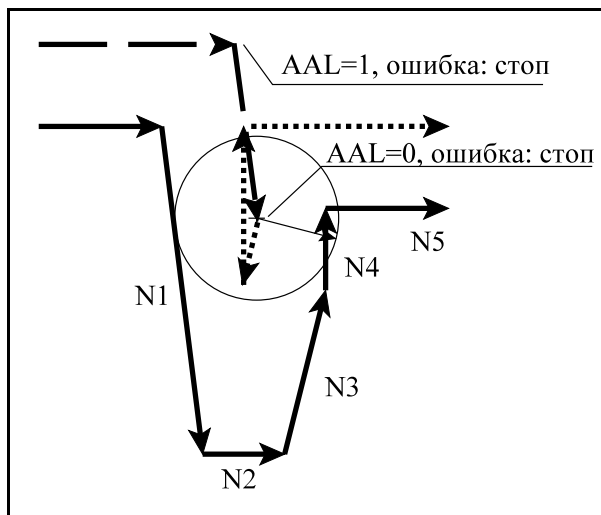


рисунок 15.6.7-15

16 Специальные трансформации

16.1 Поворот конфигурации вокруг данной точки (G68, G69)

Командой

G68 *p q R*

можно повернуть запрограммированную конфигурацию *в плоскости, выделённой с помощью G17, G18, G19.*

По адресу *p и q* задаются *координаты точки центра поворота*. Истолкуются только значения, написанные на координаты *p* и *q* выбранной плоскости.

Во включенном состоянии задания данных полярными координатами истолкуются также записанные *данные координаты p, q в прямоугольной системе координат.*

Координаты точки центра поворота *p, q* можно задавать и в виде *абсолютных данных и данных с приращением*, использованием G90, G91, или оператора I. Данные с приращением понимаются всегда от последней позиции оси (не повернутой) по

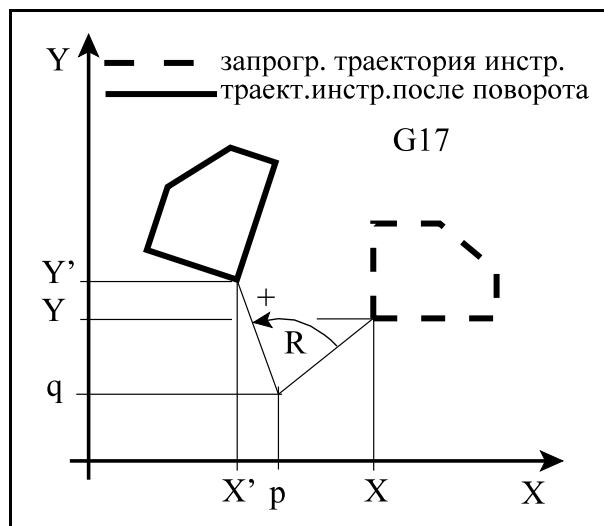


рисунок 16.1-1

последней позиции оси (не повернутой) по программисту.

Если присвоить значение одному из *p* и *q*, или ни одному, точкой центра поворота берётся мгновенная позиция оси.

Уравнение поворота конфигурации в случае G17 в плоскости XY:

$$X' = (X - p) \cos R - (Y - q) \sin R + p$$

$$Y' = (X - p) \sin R + (Y - q) \cos R + q$$

где: *p, q*: точка центра поворота,

R: угол поворота,

X, Y координаты запрограммированной точки,

X', Y': координаты повернутой точки.

По адресу **R**

задаётся *угол поворота*. Записанное на адрес *положительное* значение означает направление *против хода часовой стрелки*, а *отрицательное* значение *по ходу часовой стрелки* означает направление, которое надо истолковать по выбранной плоскости.

Значение, заданное для *R* может быть *абсолютное и инкрементное*. Если угол поворота задаётся инкрементно, тогда к углам поворота, программным предварительно, добавляется значение *R*.

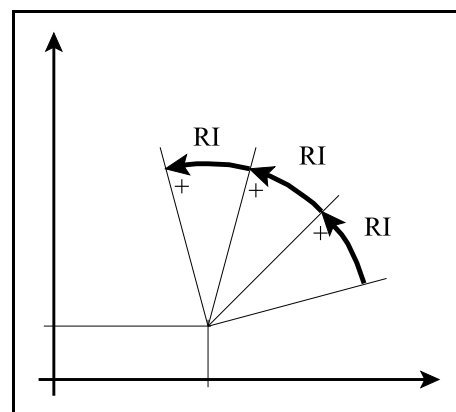


рисунок 16.1-2

Команда

G69

выключает поворот. Удаляет координаты точки центра поворота, и угол поворота также. Эта команда может стоять и рядом с другими.

Образцовый пример:

Полярной интерполяцией надо фрезеровать 3 шт. квадрата, находящиеся на 120 градусов друг от друга на торцевую поверхность заготовки. Квадрат описывается подпрограммой O0001. Заготовка обрабатывается трёхкратным выходом подпрограммы и инкрементным поворотом на 120 градусов в конце подпрограммы. На рисунке цифрами 1, 2, 3 отмечен порядок выполнения.

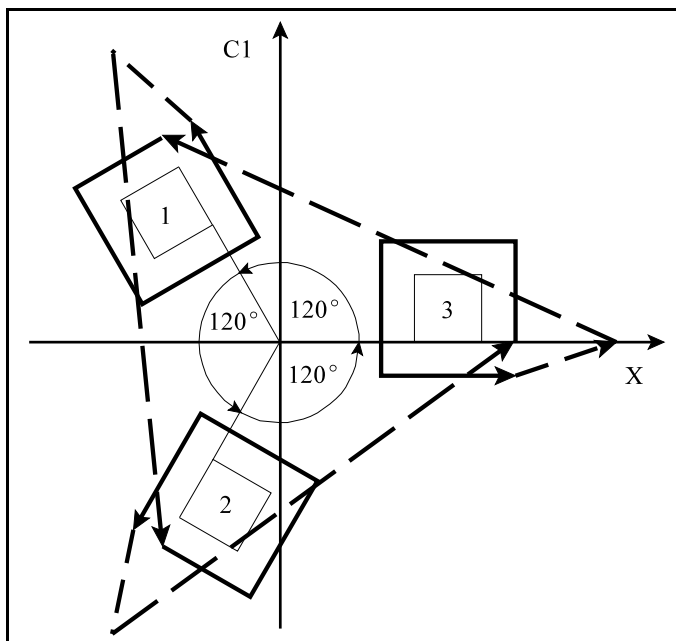


рисунок 16.1-3

Главная программа:

```
G17 G59 G94 G0 X100 Z20 C1=0
M3 S2=500
G12.1
G68 X0 C1=0 R120
M98 P1 L3
G69
G13.1
```

Подпрограмма:

```
O0001
G42 G0 X60 C1=0
G1 Z-2
C1=10
X40
C1=0
X70
G40 G0 X100 C1=0
Z20
G68 X0 C1=0 RI120
M99
```

16.2 Масштабирование конфигурации относительно заданной точки (G50, G51)

Масштабирование включается командой G51. Его задание возможно двояко: общим числовым отношением, действительным для каждой продольной оси, далее различными числовыми отношениями, действительными для главных осей X, Y, Z.

Оба способа задания выключается командой G50.

Масштабирование числовым отношением, действительным для каждой оси (P)

Командой

G51 v P

можно запрограммированную конфигурацию *уменьшить, или увеличить*.

На координатах

v

можно задавать *положение точки центра масштабирования*. Используемые адреса: основные оси X, Y, Z, далее параллельные оси.

Записанные здесь *данные координат v* истолкуются в *прямоугольной системе координат* и во включенном состоянии задания данных полярными координатами также.

Координаты точки центра масштабирования v можно задавать и *абсолютными данными и данными с приращением*, с использованием G90, G91, или оператора I.

Если не задавать значение какому то адресу оси, или ни одному из адресов, точкой центра масштабирования берётся мгновенная позиция оси.

По адресу

P

можно установить *числовое отношение масштабирования*.

Если $P < 1$, получается уменьшение, если $P > 1$, - увеличение.

Уравнение масштабирования конфигурации в пространстве XYZ:

$$X' = (X - X_0)P + X_0$$

$$Y' = (Y - Y_0)P + Y_0$$

$$Z' = (Z - Z_0)P + Z_0$$

Где: X_0, Y_0, Z_0 : координаты точки центра масштабирования v,

P: числовое отношение масштабирования,

X, Y, Z: координаты запрограммированной точки.

X', Y', Z': координаты после масштабирования.

Команда

G50

выключает масштабирование.

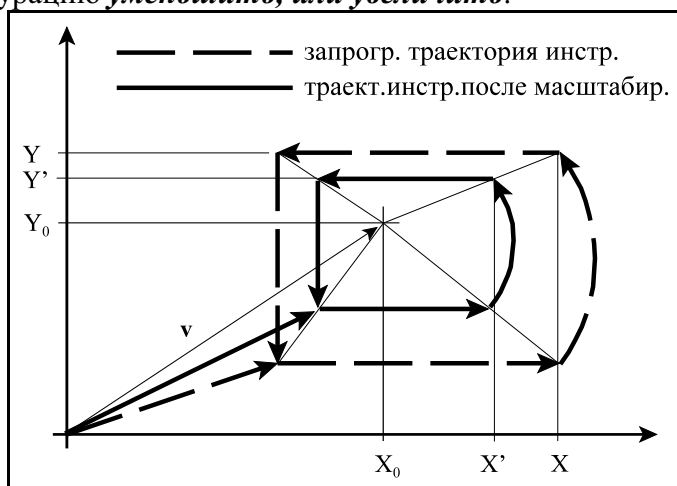


рисунок 16.2-1

Замечание:

Числовое отношение масштабирования (P) не влияет на значение применённой коррекции радиуса.

Образцовый пример:

Если нулевая точка заготовки в направлении Z лежит на опорной поверхности патрона, с помощью G51 можно просто точить меньшую или большую заготовку.

```
G90 G0 X100 Z120
G51 X0 Z0 P0.5
G1 X0 Z100 F150
X80
Z0
G50
G0 X100 Z120
```

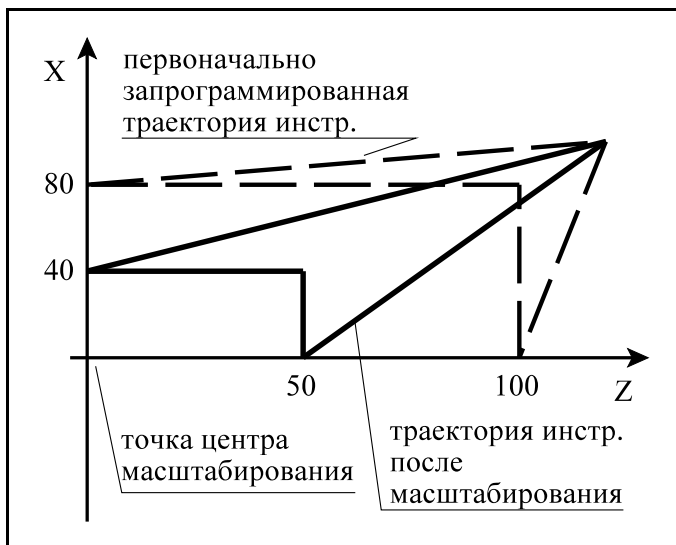


рисунок 16.2-2

Масштабирование на оси X Y Z различными числовыми отношениями (I J K)

Командой

G51 X Y Z I J K

можно запрограммированную конфигурацию *уменьшить, или увеличить*.

Координатами

X Y Z

можно задавать *положение точки центра масштабирования*. Используемые адреса: главные оси X, Y, Z. Записанные здесь *данные координат* в истолкуются в *прямоугольной системе координат* и во включенном состоянии задания данных полярными координатами также.

Координаты точки центра масштабирования *v* можно задавать и *абсолютными данными* и *данными с приращением*, с использованием

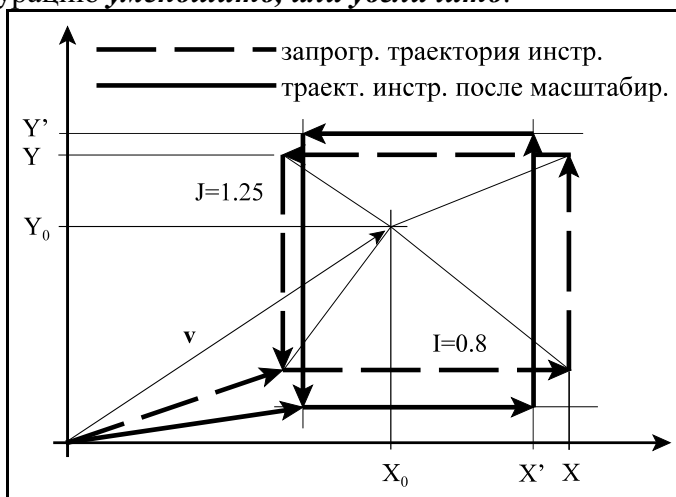


рисунок 16.2-3

G90, G91, или оператора I.

Если не задавать значение какому-то адресу оси, или ни одному из адресов, точкой центра масштабирования берётся мгновенная позиция оси.

По адресу

I J K

можно установить *числовое отношение масштабирования, действующее на оси I-X, J-Y, K-Z*.

Если $I J K < 1$ получается уменьшение, если $I J K > 1$ увеличение.

Уравнение масштабирования конфигурации в пространстве XYZ:

$$X' = (X - X_0)I + X_0$$

$$Y' = (Y - Y_0)J + Y_0$$

$$Z' = (Z - Z_0)K + Z_0$$

Где: X_0, Y_0, Z_0 : координаты точки центра масштабирования v ,
 $I J K$: числовое отношение масштабирования,
 X, Y, Z : координаты запрограммированной точки.
 X', Y', Z' : координаты после масштабирования.

Команда

G50

выключает масштабирование.

 Замечание:

- Числовые отношения масштабирования ($I J K$) не влияют на значение применённой коррекции радиуса.
- Значение $I J K$ может быть и отрицательным числом. При этом устройство ЧПУ выполняет на данной оси не только масштабирование, но и отражение. Если значение $I J K$ равно -1, на данной оси выполняет только отражение.
- Если значение $I J K$ разное, значение круговой интерполяции ($G1, G3$) и значение скругления R изменяется согласно примеру ниже:

G17

G51 X0 Y0 Z0 I0.5 J0.75 K1.25

G1 X100 ,R10 (R=, R* (I+J) / 2)

Y100

X10

G3 X0 Y90 R10 (R=R*I)

G1 Y0

G50

Радиус скругления умножается на среднее значение I и J , а в кадре $G3$ радиус окружности на I .

16.3 Отражение конфигурации на одну, или несколько прямых (G50.1, G51.1)

Командой

G51.1 v

запрограммированную конфигурацию *отражает вдоль выбранных координат* на v.

На координатах

v

можно задавать, что *на какую, параллельную ось прямую, или прямые желаем отразить*.

v может быть: X, Y, Z, или другие адреса осей.

При отражении на прямую, параллельную осью X, надо задавать положение прямой по адресу Y, при отражении на прямую, параллельную осью Y, надо задавать положение прямой по адресу X, и т.д.

Записанные здесь *данные координат v* истолкуются в *прямоугольной системе координат* и во включенном состоянии задания данных полярными координатами также.

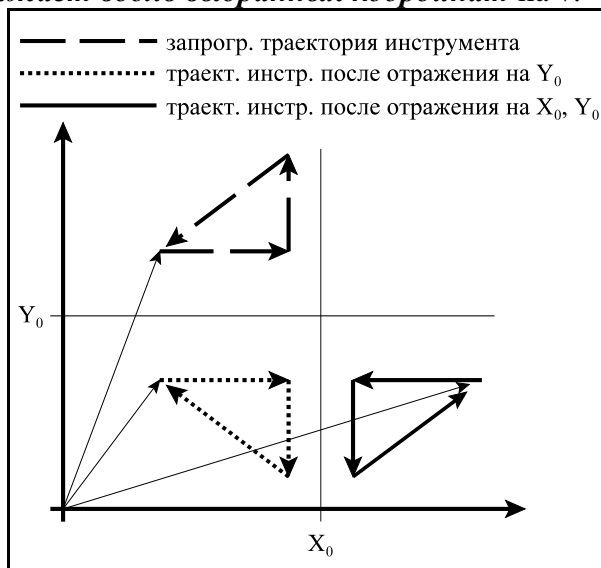


рисунок 16.3-1

Координаты осей отражения v можно задавать и *абсолютными данными и данными с приращением*, с использованием G90, G91, или оператора I.

Если *не задавать значение какому-то адресу оси, то для неё не выполняется отражение*.

Уравнение отражения конфигурации в пространстве XYZ:

$$X' = -(X - X_0) + X_0$$

$$Y' = -(Y - Y_0) + Y_0$$

$$Z' = -(Z - Z_0) + Z_0$$

Где: X_0, Y_0, Z_0 : координаты точки центра отражения v,
 X, Y, Z : координаты запрограммированной точки.
 X', Y', Z' : координаты после отражения.

Отражение на нечётное число осей изменяет направление обхода конфигурации. Вследствие этого изменяются направления кругов (G2-G3) и направление коррекции радиуса инструмента (G41-G42) далее и угол поворота (G68 R), что устройством ЧПУ автоматически принимает во внимание.

Команда

G50.1 v

выключает отражение на осях координат, заданных на v. На координаты v можно написать произвольные данные, их действие просто фиксирует факт выключения.

При включении, или выключении отражения не должно существовать состояние ни поворота (G68), ни масштабирования (G51). В противном случае выводится сообщение об ошибке 2001 Нельзя вкл./выкл. отражение в состоянии G51, или G68!

Отражение в случае двойного держателя инструмента

Если на токарном станке имеется два сменника инструментов, один из них в верхней зоне, это X^+ , а другой в нижней зоне, это X^- , для отражения на ось Z

G51.1 X0

можно использовать в ходе обработки с инструментами, имеющимися в нижнем сменнике инструментов. При желании работать инструментами, имеющимися в сменнике инструментов в нижней зоне X^- , программу можно написать для зоны X^+ и фрагмент программы отразить на ось Z .

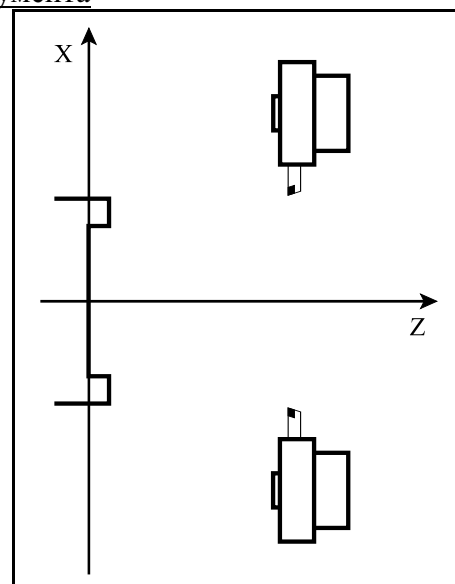


рисунок 16.3-2

Образцовый пример:

Выполнить чистовую обработку заготовки с инструментом T2, имеющимся в нижнем сменнике инструментов. Программу напишем так, как будто инструмент находился бы в верхнем сменнике, и отражаем траекторию на ось Z .

```
G0 X140 Z50
T202 M3 (чистовой резец, Q2)
G51.1 X0
N1 G0 G42 X20 Z10
Z0
X40
Z-20
G2 X60 Z-30 R10
G1 X80
Z-40
X100 Z-50
X140
N2 G40 G0 Z50
G50.1 X0
N3 X140
```

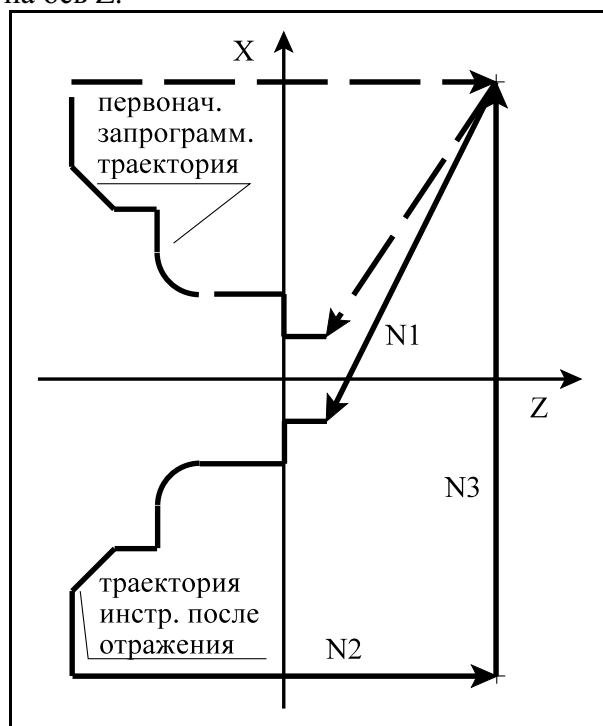


рисунок 16.3-3

16.4 Правила программирования специальных трансформаций

Порядок выполнения команд поворота *G68* и масштабирования *G51* может быть *произвольным*.

Однако, необходимо следить за тем, что если сначала поворачивать и потом масштабировать, тогда **команда поворота действительна и для координат точки центра масштабирования**. Однако, если сначала масштабировать и потом поворачивать, **для координат точки центра поворота действительна будет команда масштабирования**.

Команды включения и выключения двух операций должны быть встроены друг в друга, не должны перекрывать друг друга:

Поворот – масштабирование

```
N1 G90 G17 G0 X0 Y0
N2 G68 X80 Y20 R75
N3 G51 X130 Y50 P0.5
N4 X180 Y20
N5 G1 Y80 F200
N6 X80
N7 Y20
N8 X180
N9 G50
N10 G69 G0 X0 Y0
```

Масштабирование – поворот

```
N1 G90 G17 G0 X0 Y0
N2 G51 X130 Y50 P0.5
N3 G68 X80 Y20 R75
N4 X180 Y20
N5 G1 Y80 F200
N6 X80
N7 Y20
N8 X180
N9 G69
N10 G50 G0 X0 Y0
```

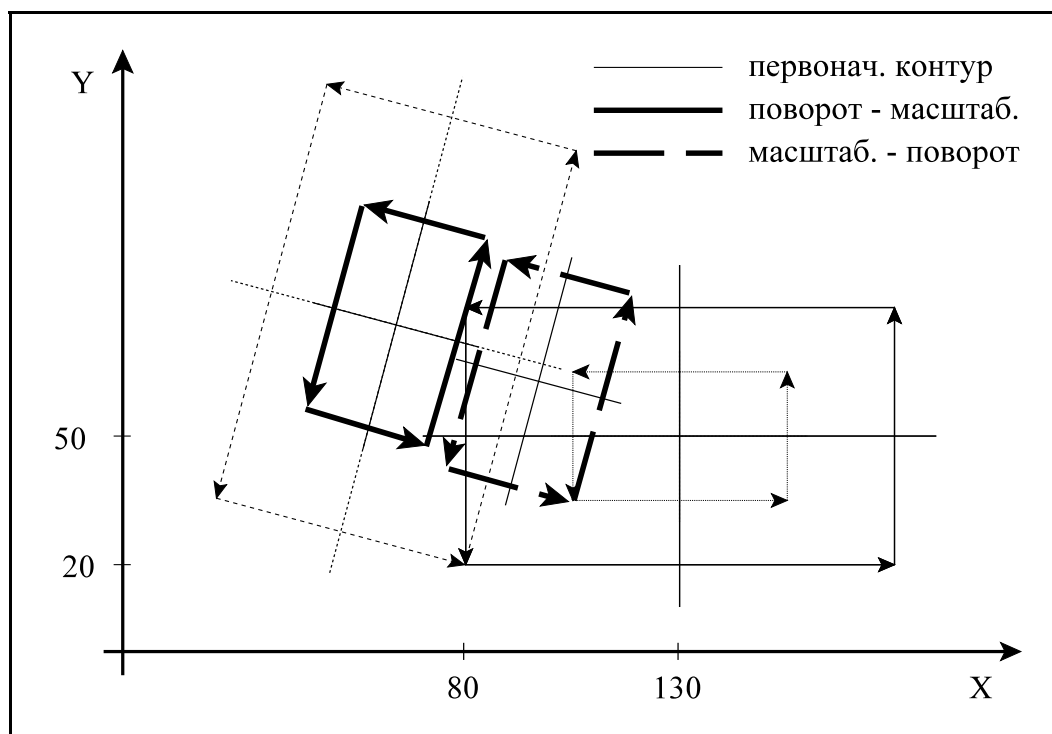


рисунок [16.4-1](#)

По рисунку видно, что не безразлично, в каком порядке применить различные трансформации.

По другому обстоит ситуация с отражением. Включить отражение можно только в состоянии G50 и G69, то есть тогда, если нет состояния команды ни масштабирования, ни поворота.

Однако, во включенном состоянии отражения можно включить как масштабирование, так и поворот.

И для отражения верно, что не должны перекрываться ни командами масштабирования, ни командами поворота, значит, сначала в соответствующем порядке надо выключить поворот и масштабирование, и только потом отражение.

G51.1 ...	(включение отражения)
G51 ...	(включение масштабирования)
G68 ...	(включение поворота)
...	
G69 ...	(выключение поворота)
G50 ...	(выключение масштабирования)
G50.1 ...	(выключение отражения)

17 Автоматические геометрические расчёты

17.1 Программирование фасок и скруглений

Между двумя кадрами, содержащими линейную интерполяцию (G01), или круговую интерполяцию (G02, G03) устройство ЧПУ может вставить автоматически *фаску*, или *скругление в выбранной плоскости*.

С длиной согласно заданному по адресу

,C

(запятая и C) *значению* вставится *равнобедренная фаска* между конечной точке кадра, содержащего адрес ,C и начальной точке последующего кадра.

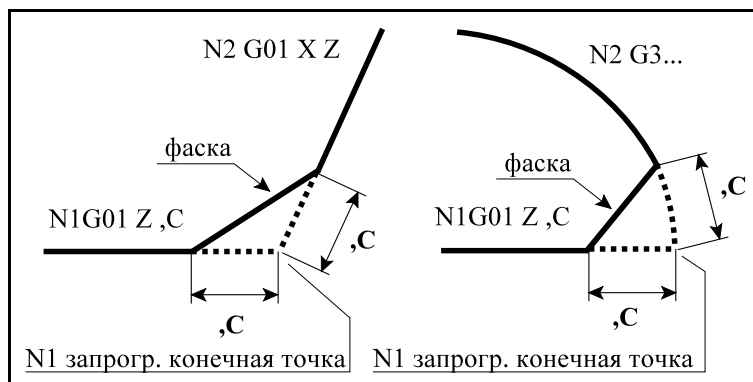


рисунок 17.1-1

Например:

N1 G18 G1 G91 Z30 ,C10

N2 X80 Z10

Заданное по адресу ,C значение покажет, что на какое расстояние от предполагаемой точки пересечения двух последующих друг за другом кадров начинается и также заканчивается фаска. Фаску можно вставить между окружностями, или между окружностями и прямой. При этом значение ,C равно длине хорды, проведённой от точки пересечения.

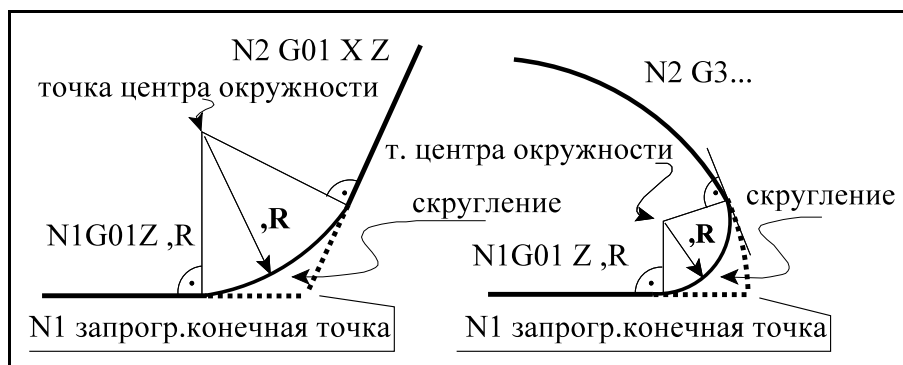


рисунок 17.1-2

С радиусом, согласно заданному по адресу

,R

(запятая и R) *значению* вставится *скругление* между конечной точке кадра, содержащего адрес ,R и начальной точке последующего кадра.

Например:

N1 G18 G91 G01 Z30 ,R8

N2 G03 X60 Z-30 R30

Дуга окружности с радиусом ,R вставится между двумя кадрами таким образом, чтобы

окружность прилегала к обоим элементам траектории.

Команду, содержащую фаску, или скругление можно писать и к концу нескольких, последующих друг за другом кадров, как это показывает и пример ниже:

```
...
G18 G1 X80 ,C10
Z60 ,R22
G3 X160 Z20 R40 ,C10
G1 X220
...
```

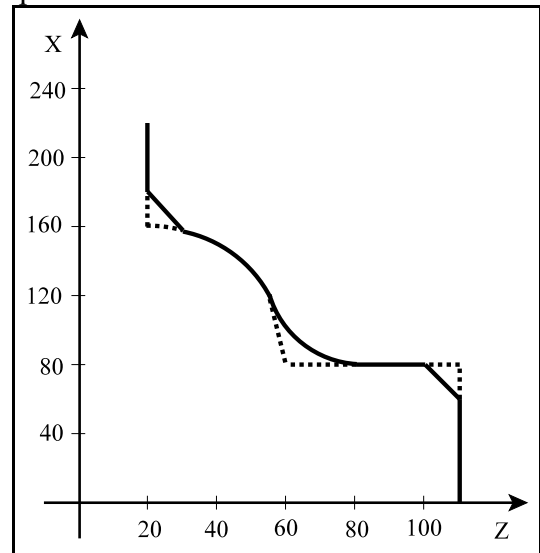


рисунок 17.1-3

☞ *Замечание:*

- Программировать фаску, или скругление можно только между элементами, лежащими в выбранной плоскости (G17, G18, G19), в противном случае устройство ЧПУ выводит сообщение об ошибке 2085 “Нелегальная фаска или скругление”.
- Если длина бедра фаски, или радиус скругления настолько большие, что нельзя припасовывать к запрограммированным кадрам, устройство ЧПУ выводит сообщение об ошибке “Фаска или скругление слишком большие”.
- Если в одном кадре запрограммированы ,C и ,R, устройство ЧПУ выводит сообщение об ошибке 2082 “Или фаска или скругление”.
- В покадровом режиме устройство ЧПУ остановится после выполнения фаски, или скругления, и примет состояние стопа.

17.2 Задание прямой с углом направления

В плоскости, определённой командой G17, G18, G19, *можно задавать прямую одной из координат* выбранной плоскости *и по адресу, A*, задавая её с *углом направления прямой*.

$$G17 \left\{ \frac{G0}{G1} \right\} \left\{ \frac{X_P, A}{Y_P, A} \right\} qF$$

$$G18 \left\{ \frac{G0}{G1} \right\} \left\{ \frac{Z_P, A}{X_P, A} \right\} qF$$

$$G19 \left\{ \frac{G0}{G1} \right\} \left\{ \frac{Y_P, A}{Z_P, A} \right\} qF$$

В формулах выше X_p , Y_p , Z_p вдоль X , Y , Z , или вдоль параллельных им осей означает координату конечной точки прямой, где q означает одну, или несколько произвольных осей, лежащих вне выбранной плоскости.

Задание по адресу $,A$ можно использовать и рядом с кодом G0 и G1.

Угол $,A$ отсчитывается от первой оси выбранной плоскости, и положительное направление противоположно ходу часовой стрелки.

Значение $,A$ может быть и положительное и отрицательное, далее может быть больше 360° , или меньше -360° .

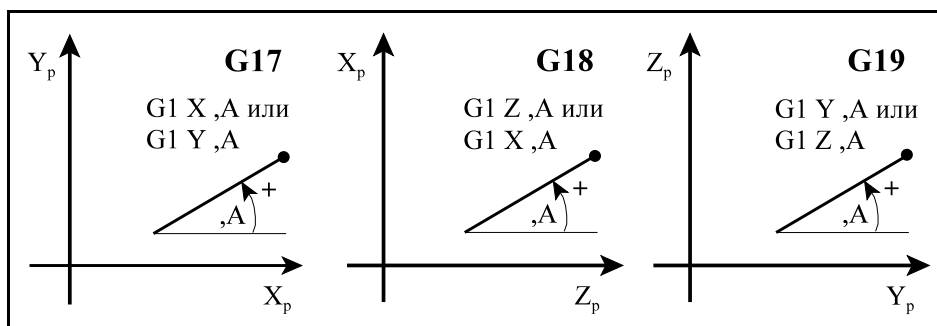


рисунок 17.2-1

Например:

```
(G18 G90) G1 X60 Z120 ...
Z70 ,A150
```

(это задание эквивалентно заданием, где
X117.735 Z70)

```
X180 ,A135
```

(это задание эквивалентно заданием, где
X180 Z38.868)

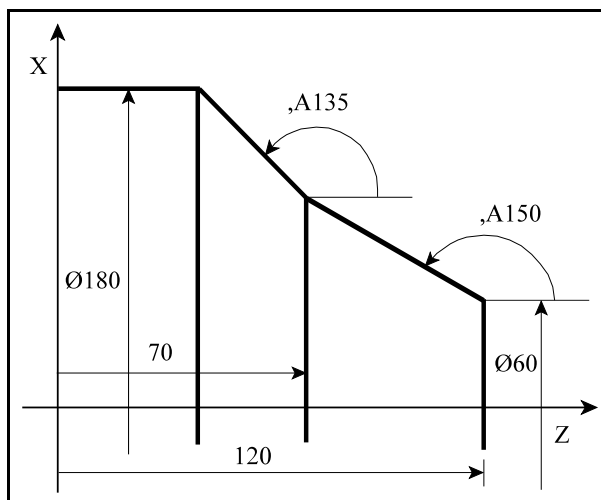


рисунок 17.2-2

☞ **Замечание:**

– В одном кадре можно задавать прямую с углом направления и фаску, или и скругление.

Например:

```
Z100 ,A30 ,C5
X100 ,A120 ,R10
Z-100 ,A210
```

– Задание угла направления по адресу $,A$ можно использовать и в циклах сверления. При этом оно принимается во внимание при выполнении позиционирования в выбранной плоскости вышеописанным образом. Например, кадр

```
G81 G91 X100 ,A30 R-2 Z-25
```

эквивалентный кадром ниже:

```
G81 G91 X100 Y57.735 R-2 Z-25
```

17.3 Расчёты точек пересечения в плоскости

Приведенные здесь расчёты точек пересечения устройство ЧПУ *выполняет только во включенном состоянии расчёта коррекции радиуса инструмента. (G41, vagy G42)*. Если бывает, что не желаем в программе принимать во внимание коррекцию радиуса инструмента, всё таки следует включить его и удалить значение соответствующей коррекции радиуса инструмента.

17.3.1 Точка пересечения двух прямых

Если *вторую прямую* одного из двух последующих кадров, выполняющих линейную интерполяцию, задаётся так, что *задаётся конечная точка прямой на обеих осях, лежащих в выделённой плоскости, и задаётся и угол направления прямой*, устройство ЧПУ рассчитывает точку пересечения прямой, выделённой в первом кадре и прямой, заданной во втором кадре.

Заданная таким образом в первом кадре прямая в дальнейшем *сверхопределённой прямой* называется.

Конечная точка первого кадра, далее начальная точка второго кадра будет расчётной точкой пересечения.

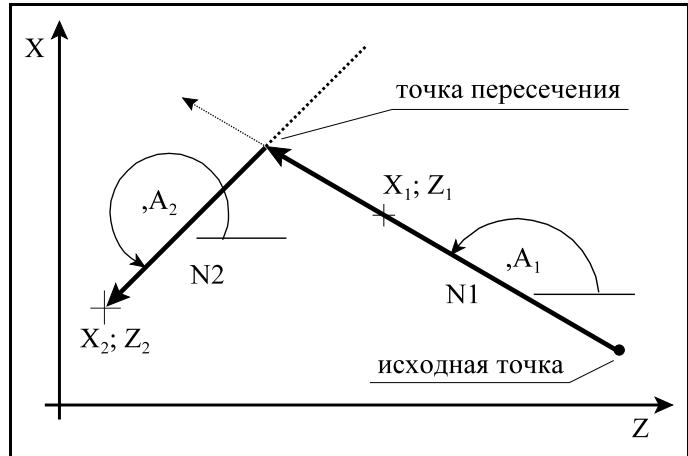


рисунок 17.3.1-1

G17 G41 (G42)

N1 G1 ,A₁ или
X₁ Y₁

N2 G1G90 X₂ Y₂ ,A₂

G18 G41 (G42)

N1 G1 ,A₁ или
X₁ Z₁

N2 G1G90 X₂ Z₂ ,A₂

G19 G41 (G42)

N1 G1 ,A₁ или
Y₁ Z₁

N2 G1G90 Y₂ Z₂ ,A₂

Точка пересечения рассчитывается *всегда в выделённой через G17, G18, G19 плоскости. Первый кадр (N1)* задаётся или *только с углом направления* (,A₁), и в этом случае из исходной точки проводится прямая под соответствующим угловым направлением до точки пересечения, *или задаётся одна произвольная точка прямой* (X₁, Y₁; X₁, Z₁; или Y₁, Z₁) отличающаяся от исходной точки, и при этом точка пересечения рассчитывается с прямой, проходящей через две точки.

Координаты, заданные во втором кадре (N2) устройство ЧПУ истолкует всегда абсолютными данными (G90).

Например:

```
(G18) G90 G41 ...
G0 X20 Z90
N10 G1 ,A150
N20 X40 Z10 ,A225
G0 Z0
...
```

Кадр N10 можно задавать и координатами одной точки прямой:

```
(G18) G90 G41 ...
G0 X20 Z90
N10 G1 X66.188 Z50
N20 X40 Z10 ,A225
G0 X0 Y20
...
```

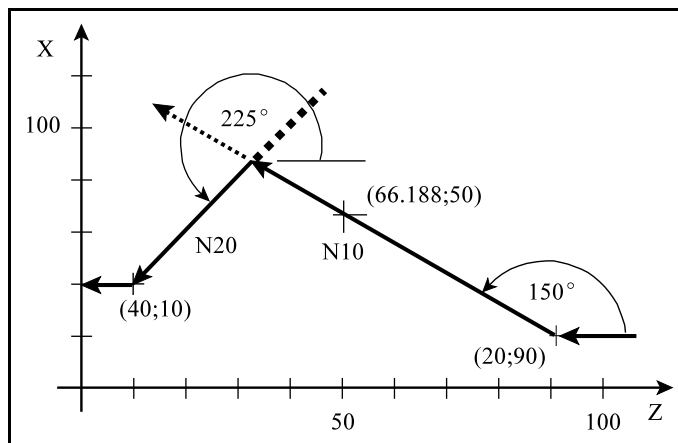


рисунок 17.3.1-2

Обратите внимание на то, что в этом случае координаты X, Z (X66.18 Z50), заданные в кадре N10 устройство не считает за конечной точкой, а только за проходящей точкой, соединяющей начальную точку прямой с заданной точкой.

Расчёт точки пересечения можно комбинировать и заданием фаски, или скругления. Например:

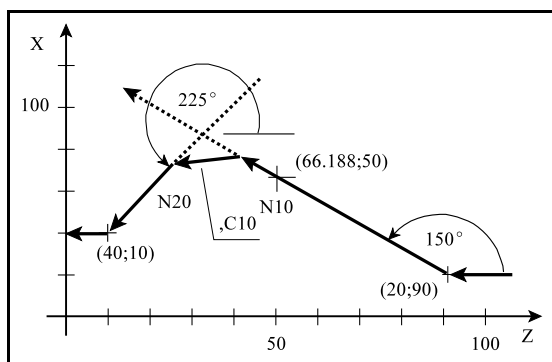


рисунок 17.3.1-3

```
(G18) G90 G41 ...
G0 X20 Z90
N10 G1 X66.188 Z50 ,C10
N20 X40 Z10 ,A225
G0 X0 Y20
...
```

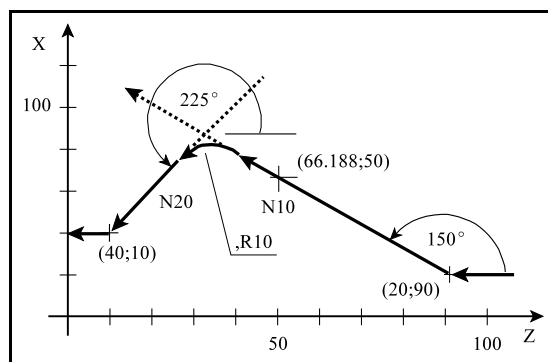


рисунок 17.3.1-4

```
(G18) G90 G41 ...
G0 X20 Z90
N10 G1 X66.188 Z50 ,R10
N20 X40 Z10 ,A225
G0 X0 Y20
...
```

В примерах выше длина фаски измеряется обратно от расчётной точки пересечения, далее подгоняется скругления к расчётной точки пересечения.

17.3.2 Точка пересечения прямой и окружности

Если всед за линейным кадром задаётся период таким образом, что для *окружности задаются координаты конечной точки и точки центра и радиус окружности также*, то есть *окружность будет сверхопределённой*, устройство ЧПУ рассчитывает точку пересечения между прямой и окружностью. *Расчётной точкой пересечения будет конечная точка первого кадра, далее начальная точка второго кадра.*

G17 G41 (G42)

N1 G1 ,A или

 $X_1 Y_1$ N2 G2 (G3) G90 $X_2 Y_2 I J$

R Q

G18 G41 (G42)

N1 G1 ,A или

 $X_1 Z_1$ N2 G2 (G3) G90 $X_2 Z_2 I K$

R Q

G19 G41 (G42)

N1 G1 ,A или

 $Y_1 Z_1$ N2 G2 (G3) G90 $Y_2 Z_2 J K$

R Q

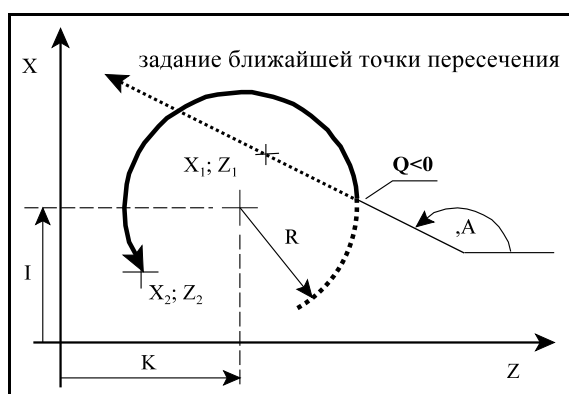


рисунок 17.3.2-1

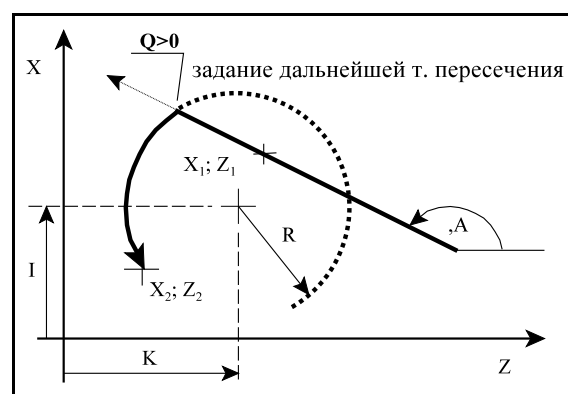


рисунок 17.3.2-2

Точка пересечения рассчитывается *всегда в плоскости, выделённой через G17, G18, G19. Первый кадр (N1)* задаётся или *только с углом направления (,A)*, и в этом случае из исходной точки проводится прямая под соответствующим угловым направлением до точки пересечения, *или задаётся одна произвольная точка прямой* (X_1, Y_1 ; X_1, Z_1 ; или Y_1, Z_1) отличающаяся от исходной точки, и при этом точка пересечения рассчитывается с прямой, проходящей через две точки.

Координаты, заданные во втором кадре (N2) таким образом и координаты **I, J, K**, определяющие **и точку центра окружности**, устройство ЧПУ истолкует **всегда абсолютными данными (G90)**.

По адресу **Q** можно задавать, что какую из получаемых двух точек пересечения должно рассчитывать устройство ЧПУ. **Если значение адреса меньше, чем ноль ($Q < 0$)**, **рассчитывается точка пересечения, лежащая по ближе к направлению прямой**, **если значение адреса больше, чем ноль ($Q > 0$)**, **тогда лежащая по дальше к направлению прямой**. Направление прохода по прямой определяется углом направления.

Рассмотрим следующий пример:

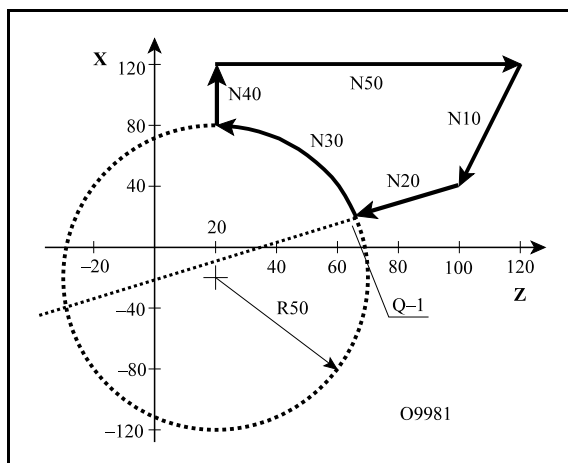


рисунок 17.3.2-3

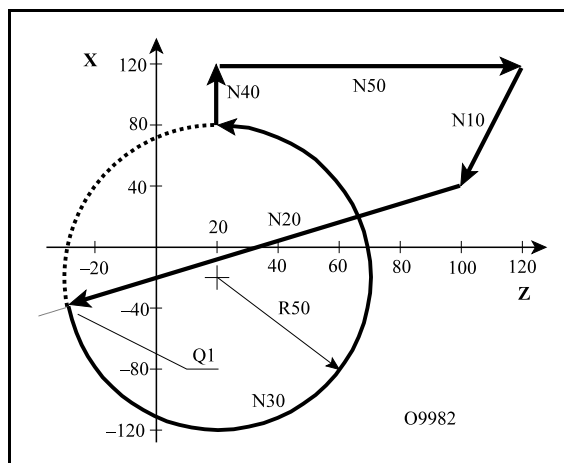


рисунок 17.3.2-4

O9981

```
N10 (G18) G42 G0 X40 Z100
N20 G1 X-40 Z-30
N30 G3 X80 Z20 I-10 K20 R50 Q-1
N40 G40 G0 X120
N50 Z120
```

O9982

```
N10 (G18) G42 G0 X40 Z100
N20 G1 X-40 Z-30
N30 G3 X80 Z20 I-10 K20 R50 Q1
N40 G40 G0 X120
N50 Z120
```

Период N30 G3 является сверхопределённым, поскольку заданы координаты точки центра (I-10 K20 в абсолютном значении), и задан радиус окружности (R50) тоже, устройство ЧПУ рассчитывает точку пересечения прямой, заданной в кадре N20 и окружности, заданной в кадре N30.

В программе O9981 рассчитывается ближайшая точка пересечения, лежащая в направлении прямой, так как в периоде N30 запрограммирован Q-1.

Однако, в программе O9982 рассчитывается дальнейшая точка пересечения, лежащая в направлении прямой, так как в периоде N30 запрограммирован Q1.

Можно комбинировать расчёт точки пересечения прямой - окружности и заданием фаски, или скругления. Например:

```
N10 (G18) G42 G0 X40 Z100 S200 M3
N20 G1 X-40 Z-30 ,R15
N30 G3 X80 Z20 I-10 K20 R50 Q-1
N40 G40 G0 X120
N50 Z120
```

Устройство ЧПУ рассчитывает точку пересечения кадра N20 и N30 и к точке пересечения приложит скругление с радиусом 15мм под действием ,R15, заданного в кадре N20.

17.3.3 Точка пересечения окружности и прямой

Если вслед за линейным кадром задаётся период таким образом, что **прямая сверхопределённая**, то есть задаются координаты конечной точки прямой и угол направления также, устройство ЧПУ рассчитывает точку пересечения между окружностью и прямой. Конечная точка первого кадра, далее начальная точка второго кадра будет расчётной точкой пересечения.

G17 G41 (G42)

N1 G2 (G3) $X_1 Y_1 I J$

или R

N2 G1 G90 $X_2 Y_2 ,A Q$

G18 G41 (G42)

N1 G2 (G3) $X_1 Z_1 I K$

или R

N2 G1 G90 $X_2 Z_2 ,A Q$

G19 G41 (G42)

N1 G2 (G3) $Y_1 Z_1 J K$

или R

N2 G1 G90 $Y_2 Z_2 ,A Q$

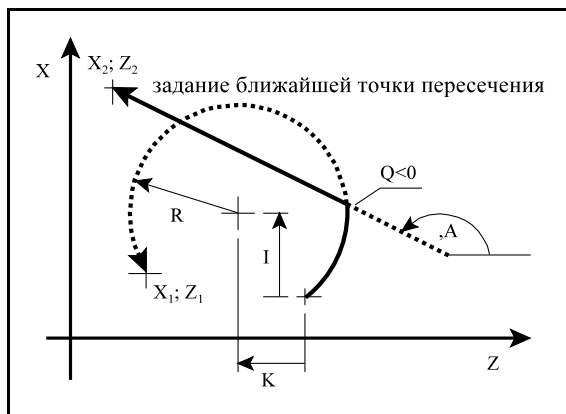


рисунок 17.3.3-1

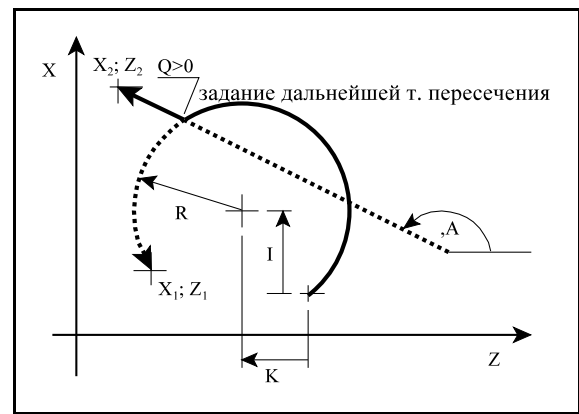


рисунок 17.3.3-2

Точка пересечения рассчитывается всегда в плоскости, выделённой через G17, G18, G19.

Первый кадр (N1), то есть **окружность** задаётся с одной произвольной точкой (X_1, Y_1 ; X_1, Z_1 ; или Y_1, Z_1) и координатой точки центра ($I J$; $I K$; или $J K$), или вместо координаты точки центра можно задавать и радиус окружности (R).

Во втором кадре (N2) **прямая сверхопределяется**, то есть задаются координаты конечной точки прямой ($X_2 Y_2$; $X_2 Z_2$; или $Y_2 Z_2$) и угол направления прямой (A) также. Координаты конечной точки прямой истолкуются устройством ЧПУ всегда абсолютными данными (G90). Надо задавать всегда угол направления вектора прямой, показывающий из получаемой точки пересечения в заданную конечную точку по адресу A , в противном случае наступают движения, противоположные намерению программиста.

По адресу Q можно задавать, что какую точку пересечения рассчитывало устройство ЧПУ из двух получаемых. Если значение адреса меньше, чем ноль ($Q<0$, например: $Q-1$), рассчитывается точка пересечения, лежащая по ближе к направлению прямой, если значение адреса больше, чем ноль ($Q>0$, например: $Q1$) рассчитывается точка пересечения, лежащая по дальше к направлению прямой. Направление прохода по прямой определяется углом направления.

Рассмотрим следующий пример:

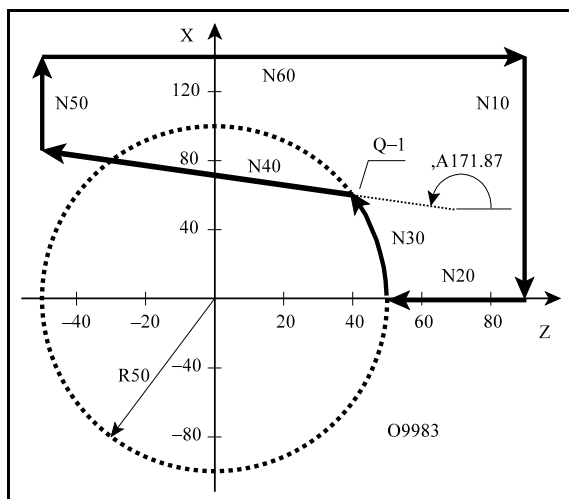


рисунок 17.3.3-3

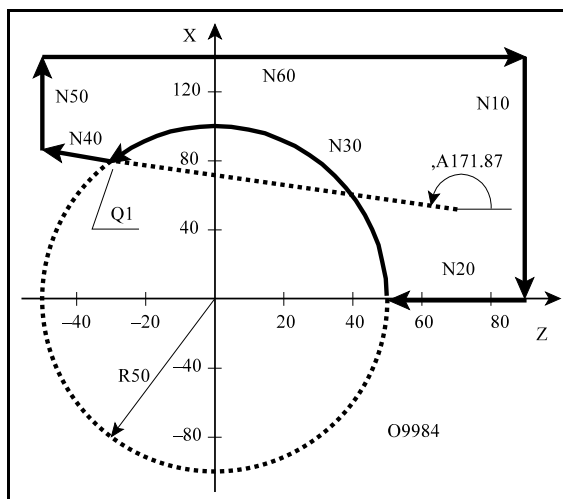


рисунок 17.3.3-4

```
O9983
N10 (G18) G0 X0 Z90
N20 G42 G1 Z50
N30 G3 X0 Z-50 R50
N40 G1 X85.714 Z-50 ,A171.87
Q-1
N50 G40 G0 X140
N60 Z90
```

```
O9984
N10 (G18) G0 X0 Z90 M3 S200
N20 G42 G1 Z50
N30 G3 X0 Z-50 R50
N40 G1 X85.714 Z-50 ,A171.87
Q1
N50 G40 G0 X140
N60 Z90
```

Линейный кадр N40 сверхопределённый, так как заданы и координаты конечной точки прямой (X85.714 Z-50) и угол направления (A171.87) также. Поэтому координаты X0 Z-50 окружности, запрограммированной в предшествующем кадре N30, не считаются значениями конечной точки, только одной точкой, через которой проходит окружность, и конечной точкой будет расчётная точка пересечения.

В программе O9983 задавалась точка пересечения (Q-1), лежащая по ближе по направлению к проходу, а в программе O9984 - лежащая по дальше (Q1).

Задание точки пересечения окружности и прямой можно комбинировать заданием фаски, или скругления. Например:

```
O9983
N10 (G18) G0 X90 X0 M3 S200
N20 G42 G1 Z50
N30 G3 X0 Z-50 R50 ,R15
N40 G1 X85.714 Z-50 ,A171.87 Q-1
N50 G40 G0 X140
N60 Z90
```

В примере в кадре N30 задавали скругление с радиусом 15 мм (R15). Устройство ЧПУ рассчитывает точку пересечения между кадрами N30 и N40, и к получаемому таким образом контуру подгоняется запрограммированное скругление.

17.3.4 Точка пересечения двух окружностей

Если два периода, последующего друг за другом задаётся так, что *для второй окружности задаются координаты конечной точки и точки центра и радиус также*, то есть *вторая окружность будет сверхопределённой*, устройство ЧПУ рассчитывает точку пересечения между двумя окружностями. *Конечная точка первого кадра, далее начальная точка второго кадра будет расчётной точкой пересечения.*

G17 G41 (G42)

N1 G2 (G3) $X_1 Y_1 I_1 J_1$

или $X_1 Y_1 R_1$

N2 G2 (G3) G90 $X_2 Y_2 I_2$

$J_2 R_2 Q$

G18 G41 (G42)

N1 G2 (G3) $X_1 Z_1 I_1 K_1$

или $X_1 Z_1 R_1$

N2 G2 (G3) G90 $X_2 Z_2 I_2$

$K_2 R_2 Q$

G19 G41 (G42)

N1 G2 (G3) $Y_1 Z_1 J_1 K_1$

или $Y_1 Z_1 R_1$

N2 G2 (G3) G90 $Y_2 Z_2 J_2$

$K_2 R_2 Q$

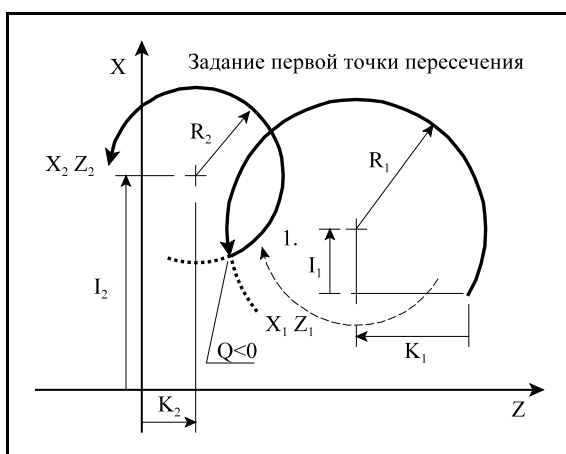


рисунок 17.3.4-1

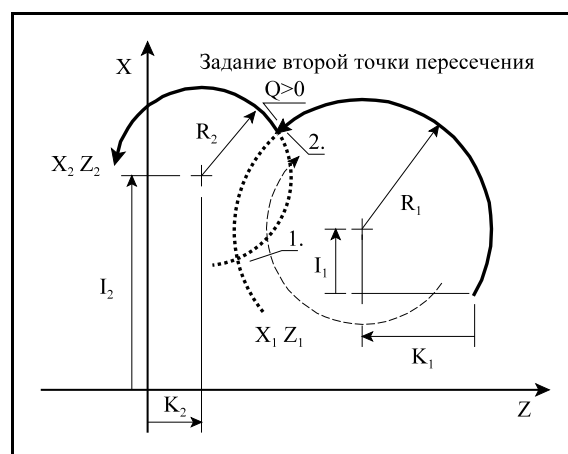


рисунок 17.3.4-2

Точка пересечения рассчитывается всегда в плоскости, выделённой через G17, G18, G19.

Первый кадр (N1) задаётся *или координатами точки центра окружности* ($I_1 J_1; I_1 K_1; J_1 K_1$), *или радиусом окружности* (R_1). В этом кадре истолкование координат точки центра совпадает с основным истолкованием задания окружности, то есть считает за относительным расстоянием, измеренным от начальной точки.

Координаты, заданные таким образом во втором кадре (N2) и *координаты I, J, K*, определяющие *точку центра окружности*, устройство ЧПУ истолкует всегда *абсолютными* данными (G90).

По *адресу Q* можно задавать, что какую точку пересечения рассчитывало устройство ЧПУ из двух получаемых. Если *значение адреса меньше, чем ноль* ($Q<0$, pl: $Q-1$), *рассчитывается первая точка пересечения*, если *значение адреса больше, чем ноль* ($Q>0$, pl: $Q+1$) *рассчитывается вторая точка пересечения.*

Рассмотрим следующий пример:

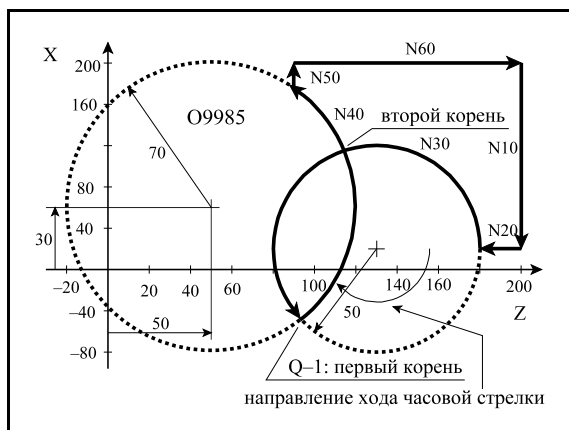


рисунок 17.3.4-3

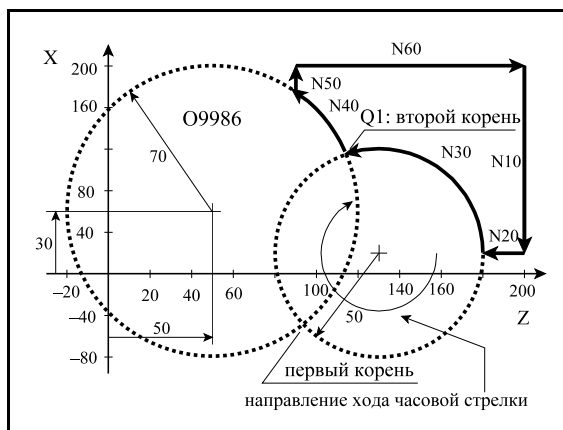


рисунок 17.3.4-4

```
O9985
N10 (G18) G0 X20 Z200
N20 G42 G1 Z180
N30 G3 X-80 Z130 R-50
N40 X174.892 Z90 I30 K50 R70
Q-1
N50 G40 G0 X200
N60 Z200
```

```
O9986
N10 (G18) G0 X20 Z200
N20 G42 G1 Z180
N30 G3 X-80 Z130 R-50
N40 X174.892 Z90 I30 K50 R70
Q1
N50 G40 G0 X200
N60 Z200
```

Период N40 является сверхопределённым, так как заданы и координаты точки центра (I30 K50 абсолютным значением, далее I понимается в радиусе), и радиус тоже (R70). Поэтому координаты окружности X-80 Z130, запрограммированные в предшествующем кадре N30, не считаются значениями конечной точки, а только одной точкой, через которой проходит окружность, и конечная точка будет расчётной точкой пересечения.

В программе O9985 задана ближайшая точка по ходу часовой стрелки (Q-1), а в программе O9986 далёкая (Q1).

Задание точки пересечения двух окружностей можно комбинировать фаской, или скруглением. Например:

```
O9986
N10 (G18) G0 X20 Z200 M3 S200
N20 G42 G1 Z180
N30 G3 X-80 Z130 R-50 ,R20
N40 X174.892 Z90 I30 K50 R70 Q1
N50 G40 G0 X200
N60 Z200
```

В примере в кадре N30 задавали скругление 20мм (,R20). Устройство ЧПУ рассчитывает точку пересечения между кадрами N30 и N40, и к полученному таким образом контуру подгоняет запрограммированное скругление.

17.3.5 Сцепление расчётов точек пересечения

Кадры с расчётом точек пересечения можно сцеплять, то есть можно выделить несколько кадров, следующих друг за другом для расчёта точек пересечения. Устройство ЧПУ будет рассчитывать точки пересечения до тех пор, пока находит в программе сверхопределённые прямые, или окружности.

Рассмотрим следующий пример:

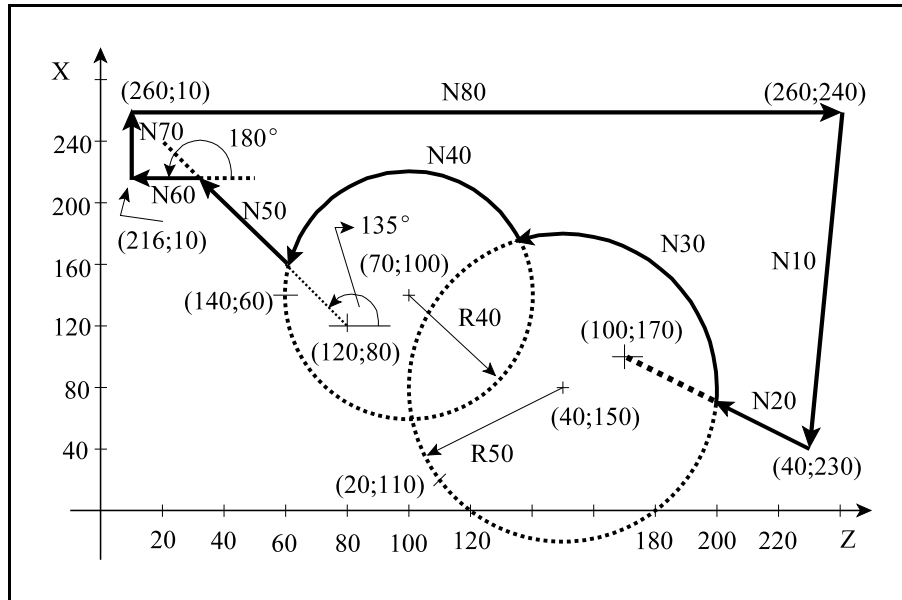


рисунок 17.3.5-1

```

N10 (G18) G0 G42 X40 Z230 F300
N20 G1 X100 Z170
N30 G3 X20 Z110 I40 K150 R50 Q-1
N40 X140 Z60 I70 K100 R40 Q1
N50 G1 X120 Z80 ,A135 Q1
N60 X216 Z10 ,A180
N70 G40 G0 X260
N80 Z240

```

В примере выше кадр N30, N40, N50, N60 является сверхопределённым.

Прямую N20 проводит не до запрограммированной конечной точки (X100 Z170), так как период N30 является сверхопределённым, то есть адреса I K R не заполнены, и по адресу Q задано, что какую точку пересечения должно искать.

И период N30 проводит не до запрограммированной конечной точки (X20 Z110), так как период N40 тоже сверхопределённый.

Последний сверхопределённый кадр в программе - это прямая N60. Поскольку следующий за ним кадр прямой N70 не сверхопределённый, поэтому запрограммированные в кадре N60 координаты X216 Z10 считает не одной проходящей точкой прямой, а за координатами конечной точки кадра N60.

Вообще говоря, можно сказать, что устройство ЧПУ только тогда считает точки координат сверхопределённых кадров прямых и окружностей, лежащих в выделённой плоскости за координатами конечной точки, если после ними уже не следует сверхопределённый кадр.

18 Токарные циклы

18.1 Простые циклы

Простыми циклами считаются цикл продольного точения G77, простой цикл нарезания резьбы G78 и цикл торцевого точения G79.

18.1.1 Цикл продольного точения (G77)

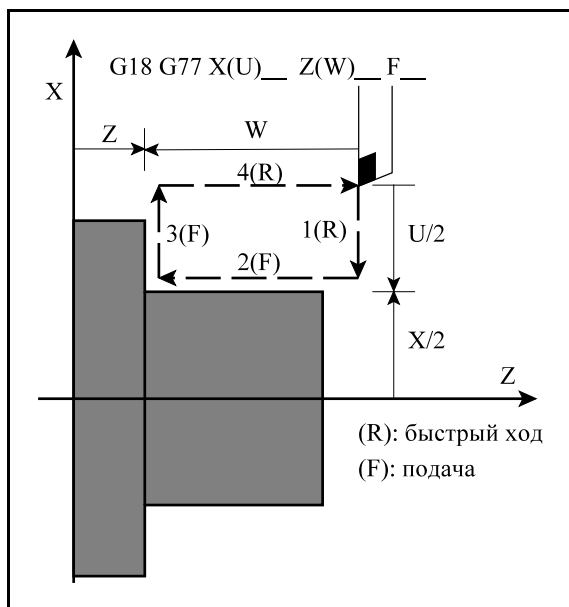


рисунок 18.1.1-1

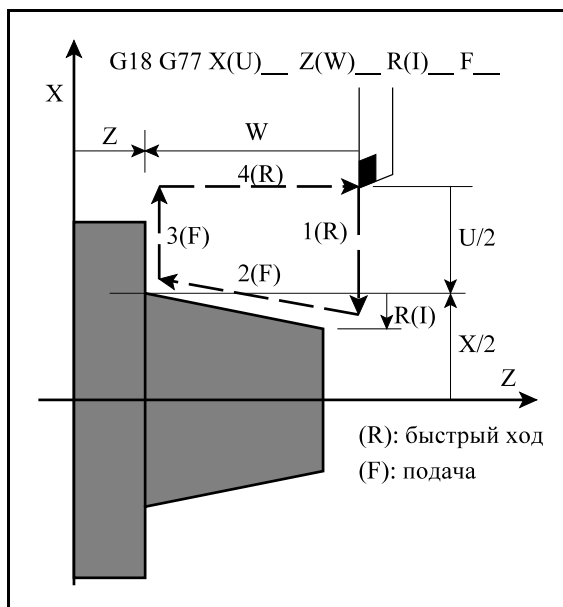


рисунок 18.1.1-2

Цикл цилиндрического продольного точения можно задавать следующим образом:

G18 (Z-X плоскость)
G77 X_p(U) Z_p(W) F

G19 (Y-Z плоскость)
G77 Y_p(V) Z_p(W) F

G17 (X-Y плоскость)
G77 X_p(U) Y_p(V) F

Врезание выполняется вдоль 2-й оси выделённой плоскости и точение вдоль 1-й оси плоскости.

Инкрементное задание данных возможно оператором I, программированием G91, а также заданием адреса U, V, W.

В случае инкрементного задания данных знак данных определяет направление 1-й и 2-й траектории. На рисунке, показывающем продольное точение в плоскости G18, знак как для адреса U, так и для адреса W отрицательный.

Перемещение выполняется запрограммированной подачей в кадре по адресу F, или мо-

дальней подачей по 2-й и 3-й траектории, а быстрым ходом по 1-й и 4-й траектории.

Цикл конического продольного точения можно задавать следующим образом:

G18 (Z-X плоскость)

G77 X_p(U)___ Z_p(W)___ R(I)___ F___

G19 (Y-Z плоскость)

G77 Y_p(V)___ Z_p(W)___ R(K)___ F___

G17 (X-Y плоскость)

G77 X_p(U)___ Y_p(V)___ R(J)___ F___

Конусность можно задавать в зависимости от R, или от выбранной плоскости по адресу I, J, K. В обоих случаях истолкование данных то же самое. Заданные по адресу **R(I, J, K)** данные истолкуются **всегда инкрементными** данными, и понимаются **заданными в радиусе** от позиции, заданной по адресу 2-й оси плоскости. Знак адреса R(I, J, K) определяет направление уклона конуса.

Истолкование остальных адресов совпадает изложенным при циклов цилиндрического продольного точения.

Код G77, и данные, запрограммированные в кадре G77 передаются модально.

В покадровом режиме точение остановится в конце всех четырёх операций (1, 2, 3, 4).

В случае инкрементного перемещения знак адресов U, W, и R(I) влияет на направление движения в состоянии G18 согласно следующим:

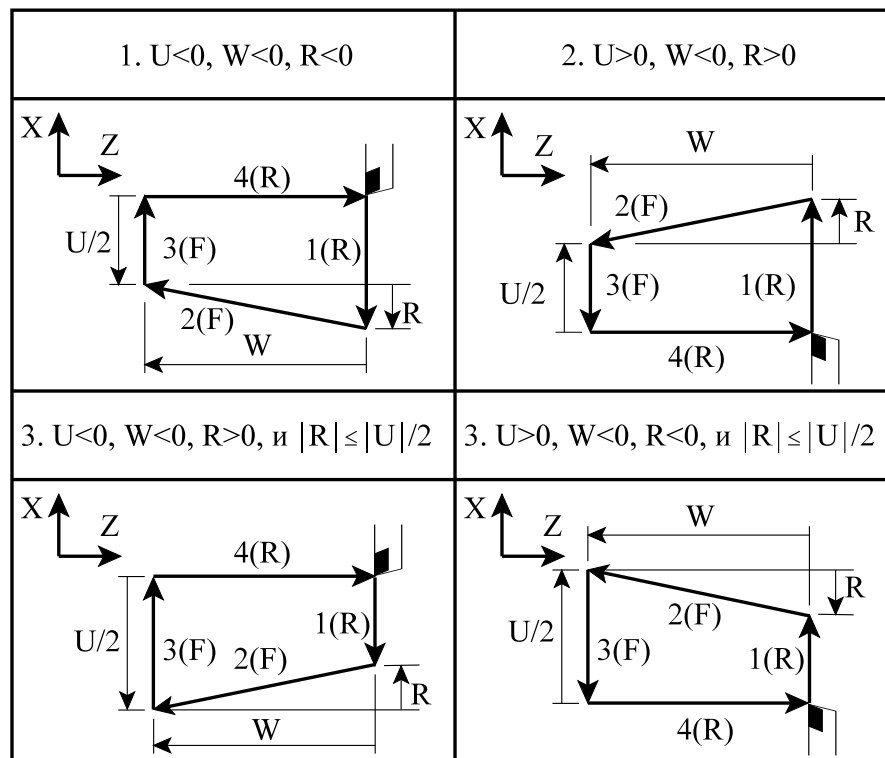


рисунок 18.1.1-3

18.1.2 Простой цикл нарезания резьбы (G78)

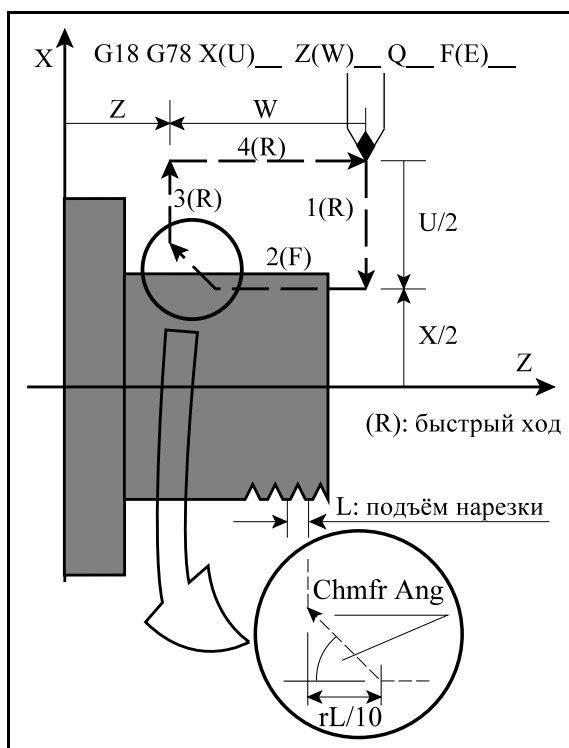


рисунок 18.1.2-1

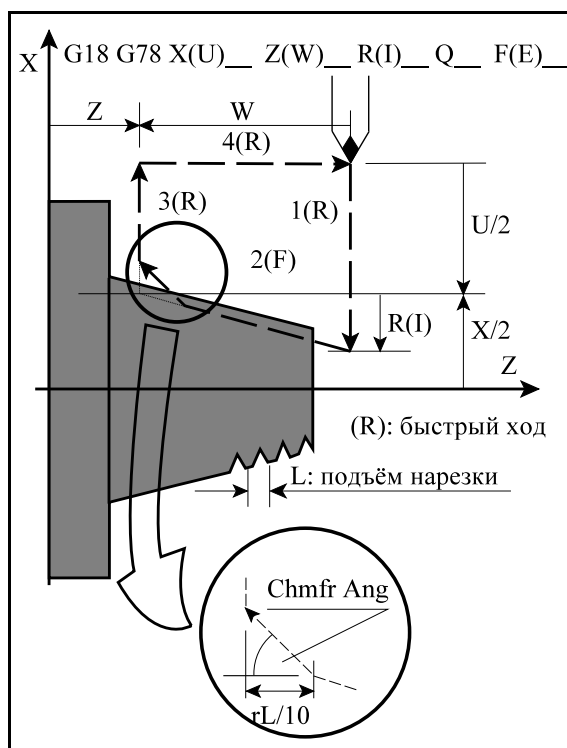


рисунок 18.1.2-2

Простой цикл нарезания резьбы можно задавать следующим образом:

G18 (Z-X sík)
G78 $X_p(U)$ $Z_p(W)$ Q $F(E)$

G19 (Y-Z sík)
G78 $Y_p(V)$ $Z_p(W)$ Q $F(E)$

G17 (X-Y sík)
G78 $X_p(U)$ $Y_p(V)$ Q $F(E)$

Врезание выполняется всегда вдоль 2-й оси выделённой плоскости и нарезает резьбу вдоль 1-й оси плоскости.

Инкрементное задание данных возможно оператором I, программированием G91, а также заданием адреса U, V, W.

В случае инкрементного задания данных знак данных определяет направление 1-й и 2-й траектории. На рисунке показывается нарезание резьбы в плоскости G18, знак как при адреса U, так и при адреса W отрицательный.

В кадре по адресу F программируется подъём нарезки, или по адресу E число витков на один дюйм, а также заданное в °-ах по адресу Q угловое значение начала резьбы, отсчитывая от нулевого импульса датчика, согласно изложенным в кадре G33.

Движение 1, 3, 4 совершается быстрым ходом.

В конце 2-го участка выполняется фаска под углом, установленным параметром N1607

Chmfr Ang.

Длина участка фаски определяется параметром N1606 ThrdChmfr, обозначенным на рисунке с буквой г. Длина участка

$$r \cdot L / 10$$

где: r: значение параметра N1606 ThrdChmfr

L: запрограммированный подъём нарезки

Если например, значение параметра 4 и запрограммированный подъём нарезки F2, тогда длина фаски:

$$2 \cdot (4 / 10) = 0.8 \text{ мм}$$

Цикл конического нарезание резьбы можно задавать следующим образом:

G18 (Z-X плоскость)

G78 X_p(U)___ Z_p(W)___ R(I)___ Q___ F(E)___

G19 (Y-Z плоскость)

G78 Y_p(V)___ Z_p(W)___ R(K)___ Q___ F(E)___

G17 (X-Y плоскость)

G78 X_p(U)___ Y_p(V)___ R(J)___ Q___ F(E)___

Конусность можно задавать в зависимости от R, или от выбранной плоскости по адресу I, J, K. В обоих случаях истолкование данных то же самое. Заданные по адресу **R(I, J, K)** данные истолкуются **всегда инкрементными** данными, и понимаются **заданными в радиусе** от позиции, заданной по адресу 2-й оси плоскости. Знак адреса R(I, J, K) определяет направление уклона конуса.

Истолкование остальных адресов совпадает изложенным при циклов цилиндрического продольного точения.

Угол фаски и в этом случае 45° и длина г измеряется вдоль прямой, параллельной осям.

Код G78, и данные, запрограммированные в кадре G78 передаются модально.

В покадровом режиме точение остановится в конце всех четырёх операций (1, 2, 3, 4).

Действие нажатия клавиши STOP во 2-й операции цикла

С использованием клавиши STOP в любое время можно остановить 1-ую, 3-ую и 4-ую операцию цикла и суппорты останавливаются так, как в случае нормальной интерполяции G0.

Нажатия клавиши STOP действует и во 2-й операции нарезания резьбы, однако, при этом устройство ЧПУ сначала проделает ту же самую фаску, как в конце 2-й операции, затем сначала быстрым ходом отводит вдоль оси X, потом по Z установится в исходную точку. По маршруту спасения клавиша STOP уже не действует.

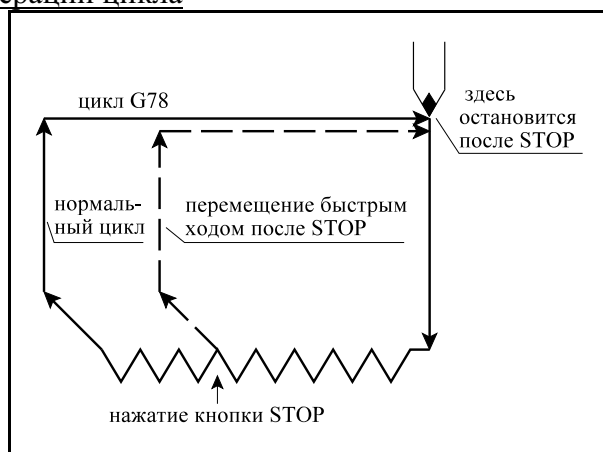


рисунок 18.1.2 -3

18.1.3 Цикл торцевого точения (G79)

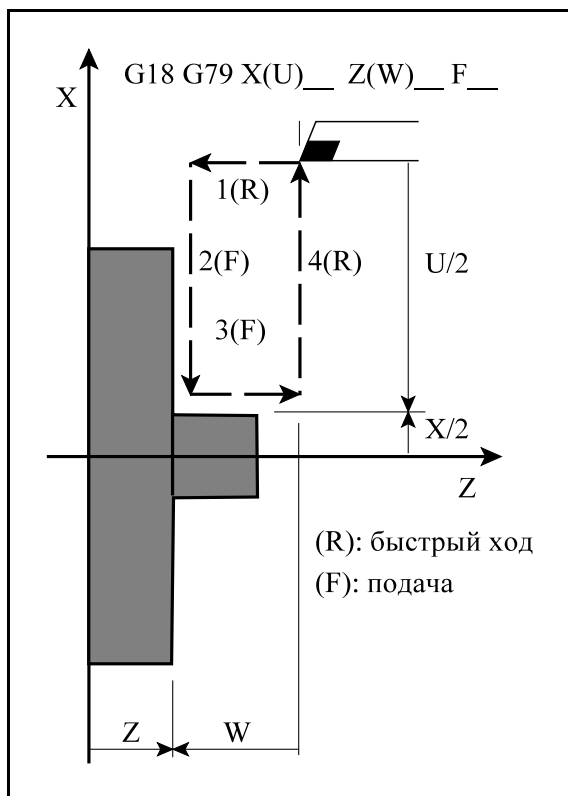


рисунок 18.1.3 -1

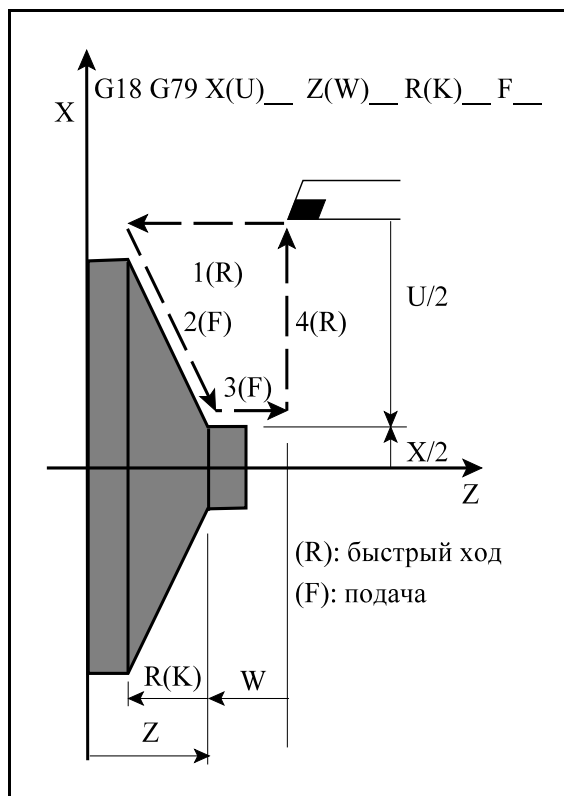


рисунок 18.1.3-2

Цикл торцевого точения можно задавать следующим образом:

G18 (Z-X плоскость)
G79 $X_p(U)$ $Z_p(W)$ F__

G19 (Y-Z плоскость)
G79 $Y_p(V)$ $Z_p(W)$ F__

G17 (X-Y плоскость)
G79 $X_p(U)$ $Y_p(V)$ F__

Врезание выполняется всегда вдоль 1-й оси выделённой плоскости и выполняет торцевое точение вдоль 2-й оси плоскости.

Инкрементное задание данных возможно оператором I, программированием G91, далее и заданием адреса U, V, W.

В случае инкрементного задания данных знак данных определяет направление 1-й и 2-й траектории. На рисунке показывается выполнение торцевого точения в плоскости G18, знак как при адреса U, так и при адреса W отрицательный.

Перемещение выполняется с запрограммированной подачей в кадре по адресу F, или модальной подачей по 2-й и 3-й траектории, а быстрым ходом по 1-й и 4-й траектории.

Цикл конического торцевого точения можно задавать следующим образом:

G18 (Z-X плоскость)

G79 X_p(U)___ Z_p(W)___ R(K)___ F___

G19 (Y-Z плоскость)

G79 Y_p(V)___ Z_p(W)___ R(J)___ F___

G17 (X-Y плоскость)

G79 X_p(U)___ Y_p(V)___ R(I)___ F___

Конусность можно задавать в зависимости от R, или от выбранной плоскости по адресу I, J, K. В обоих случаях истолкование данных то же самое. Заданные по адресу **R(I, J, K)** данные истолкуются **всегда инкрементными** данными, и понимаются **заданными в радиусе** от позиции, заданной по адресу 1-й оси плоскости. Знак адреса R(I, J, K) определяет направление уклона конуса.

Истолкование остальных адресов совпадает изложенным при циклов цилиндрического продольного точения.

Код G79, и данные, запрограммированные в кадре G79 передаются модально.

В покадровом режиме точение остановится в конце всех четырёх операций (1, 2, 3, 4).

В случае инкрементного программирования знак адресов U, W, és R(K) следующим образом влияют на направление движения в состоянии G18:

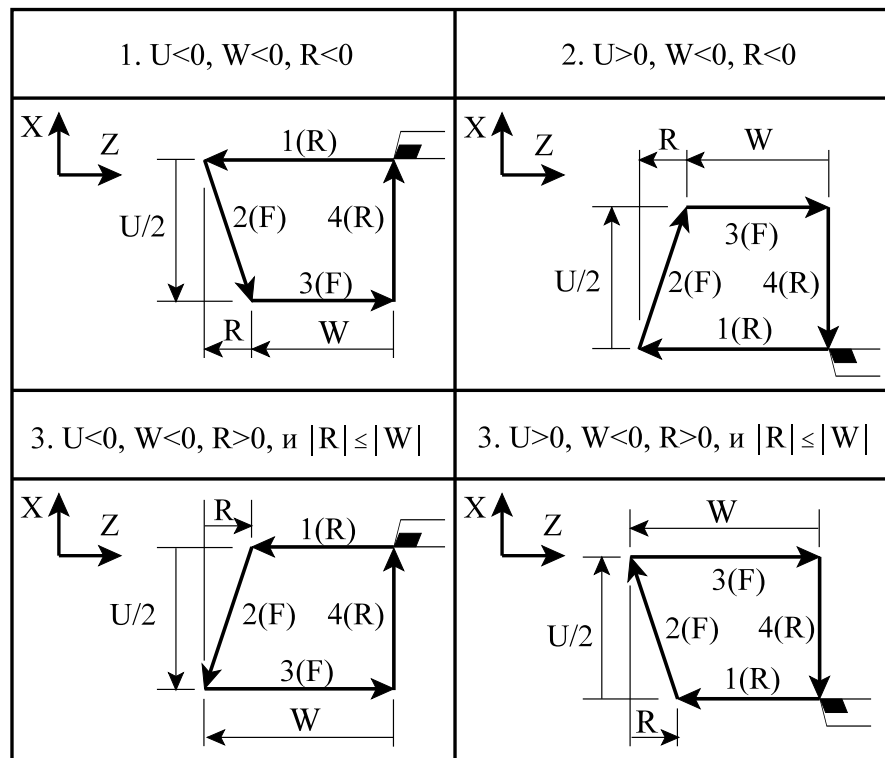


рисунок 18.1.3 -3

18.1.4 Использование простых циклов

Как *код циклов G*, так и *входные параметры передаются модально*. Это означает, что в том случае, если один раз задавали переменные цикла, X(Z, Y), U(W V), Z(Y, X), W(V, U), или R(I, J или K), и их значение неизменное, их не нужно опять записать в программе.

Например:

```
G91 G18...
G77 X-20 Z-50 F0.5
X-30
X-40
X-50
G0... (цикл удалён)
...
```

В примере выше меняется только значение врезания (X), поэтому только этот адрес надо снова заполнять, значение остальных остаётся неизменным.

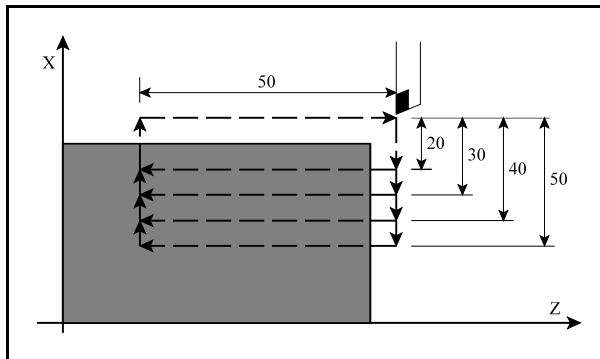


рисунок 18.1.4 -1

Во включенном состоянии цикла выполнение

цикла происходит только тогда, если какое-то переменное, ссылающееся на движения, X(U), Y(V), Z(W) заполнено. Если например, в состоянии цикла выполняется функция в самостоятельном кадре, состояние цикла остаётся включённым, но цикл не повторяется:

```
G18...
G77 U-20 W-50 F0.5 (включает цикл и выполняет его)
T202 (включает цикл, но не выполняет его)
U-30 (выполняет цикл)
G1... (цикл удалён)
...
```

Цикл и модальные переменные удаляют коды G интерполяции, относящиеся к 1-й группе.

В кадры, содержащие простые циклы, можно записать и функции M, S, T. Функции выполняются всегда в 1-й операции цикла, или параллельно с движением, или в конце движения. Если это в некоторых случаях не подходит нам, функцию надо записать в отдельный кадр.

18.2 Сложные циклы

Сложные циклы упрощают описание программы заготовки. Например, надо описать контур окончательного размера заготовки под чистовую обработку. Этот контур заодно определяет основу циклов черновой обработки заготовки (G71, G72, G73). Помимо циклов чернового точения стоит на распоряжение один цикл чистового точения (G70), один цикл нарезания резьбы (G76) и два цикла отрезки (G74, G75) тоже.

18.2.1 Цикл чернового точения (G71)

Имеется два цикла чернового точения, цикл 1-го и 2-го типа.

Цикл чернового точения 1-го типа

Если задан контур окончательного размера заготовки, обозначенный на рисунке точками А–А'–В, цикл G71 врезанием с величиной Δd выполняется черновое обтачивание прутка с припуском на чистовую обработку с величиной $\Delta u/2$ и Δw .

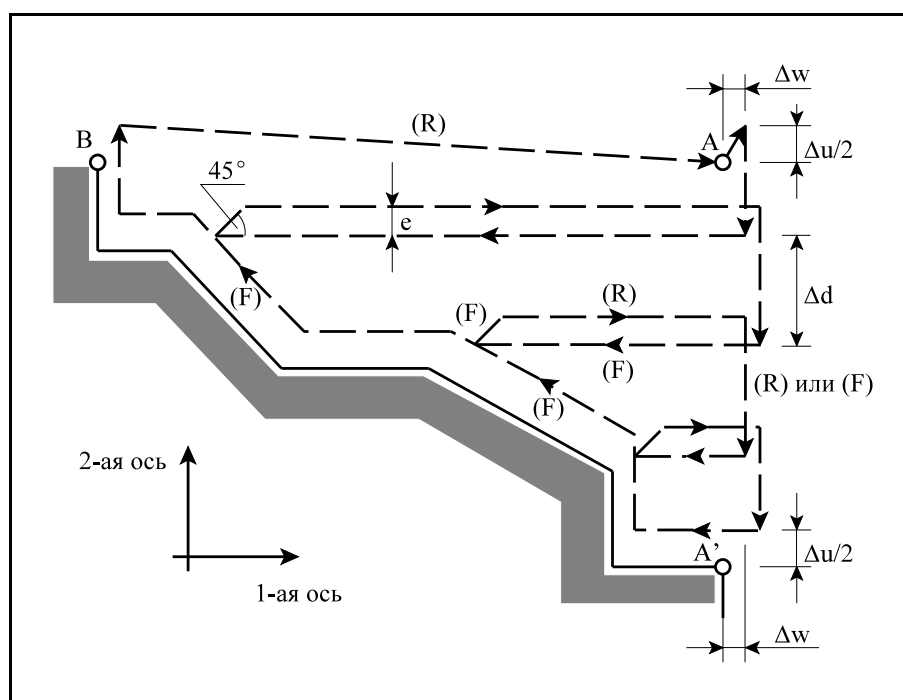


рисунок 18.2.1-1

1-й способ задания:

Задание цикла чернового точения совершается заполнением параметров двух последующих друг за другом кадров.

G18 (Z-X плоскость)**G71 U(Δd) R (e)****G71 P (n_s) Q (n_f) U(Δu) W(Δw) F(f) S(s) T(t)**N(n_s) X(U) ...

...

N(n_f)...**G19 (Y-Z плоскость)****G71 W(Δd) R (e)****G71 P (n_s) Q (n_f) W(Δu) V(Δw) F(f) S(s) T(t)**N(n_s) Z(W) ...

...

N(n_f)...**G17 (X-Y плоскость)****G71 V(Δd) R (e)****G71 P (n_s) Q (n_f) V(Δu) U(Δw) F(f) S(s) T(t)**N(n_s) Y(V) ...

...

N(n_f)...

Где:

Δd : глубина резания. *Положительное* число, понимается всегда в *радиусе*. Если значение глубины резания не задано в программе, тогда устройство ЧПУ берёт его с параметра N1600 Depth of Cut.

e: мера отвода. *Положительное* число, понимается всегда в *радиусе*. Если значение отвода не задано в программе, тогда устройство ЧПУ берёт его с параметра N1601 Escape.

n_s: номер начального кадра части программы (A—A'—B участок), описывающей чистовую обработку.

n_f: номер завершающего кадра части программы (A—A'—B участок), описывающей чистовую обработку.

Δu : величина и направление припуска на чистовую обработку *на 2-й оси выбранной плоскости*. Число *со знаком*, понимается в зависимости от истолкования 2-й оси в *диаметре*, или в *радиусе*.

Δw : величина и направление припуска на чистовую обработку *на 1-й оси выбранной плоскости*. Число *со знаком*, понимается в зависимости от истолкования 1-й оси в *диаметре*, или в *радиусе*.

f, s, t: в процессе цикла не выполняются функции F, S, T, запрограммированные в части

программы, описывающей чистовую обработку от n_s до n_f (A—A'—B участок), а выполняются значения f, s, t , заданные в кадре G71, или полученные раньше модально.

В зависимости от того, что в заданном кадре программировали ли P и Q, решается, что заданное по адресу U(W, V) значение означает Δd , или Δu . Если нет, значение адреса U(W, V) будет Δd , если да, значение адреса U(W, V) будет Δu .

Цикл чернового точения выполняется тем кадром, в котором задавали P и Q.

Перемещение между точками A—A' надо **обязательно** задавать в кадре с номером n_s по адресу P с программированием **G0**, или **G1**. Заданным здесь кодом решается, что в ходе чернового точения **врезание** (движение по направлению A—A') выполняется **быстрым ходом** (в случае программирования **G0**), или **подачей** (в случае программирования **G1**). В этом кадре P(n_s) в случае G18 надо задавать движение всегда в направлении X (в случае G19 в направлении Z, в случае G17 в направлении Y), ссылаться на другую ось нельзя.

Участок A'—B в сути дела является контуром, состоящим из прямых и дуг окружности. **Контур в цикле чернового точения 1-го типа** как в направлении X (Z, Y), так и в направлении Z (Y, X) **должен быть монотонно возрастающим, или монотонно убывающим**, что означает, что нельзя **повернуться назад ни в одном направлении**. Циклом можно пользоваться во всех четырёх квадрантах плоскости. На приложенном рисунке указан и знак припуска на чистовую обработку.

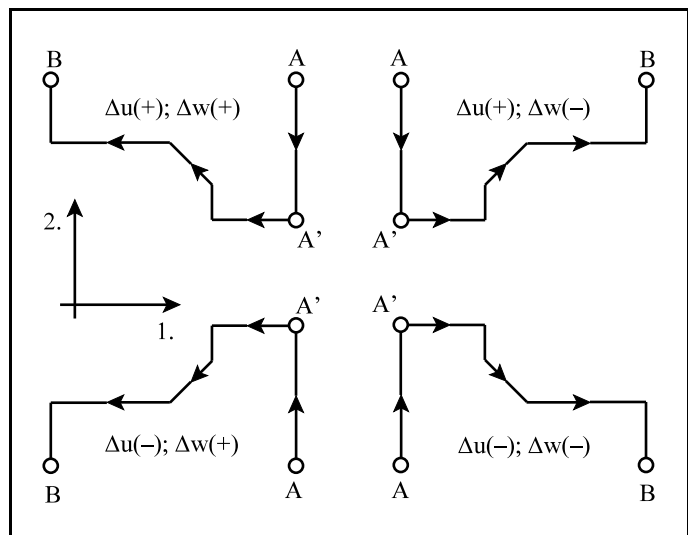


рисунок 18.2.1 -2

Запрограммированные в части программы между кадрами n_s и n_f функциями F, S, T пренебрегается, и выполняются только те, которые запрограммировали в кадре G71 (f, s, t), или раньше. То же самое относится и к постоянной скорости резания, запрограммированной между кадрами n_s и n_f , то есть выполняется **состояние G96, или G97 перед кадром G71, и значение постоянной скорости резания**.

Кадром с номером от n_s до n_f нельзя содержать вызов подпрограммы.

Кадром с номером от n_s до n_f нельзя содержать вызов подпрограммы.

Во время выполнения цикла можно включить расчёт коррекции радиуса инструмента (G41, G42) с таким ограничением, что можно включить между кадрами с номером от n_s от n_f (G41, или G42) и можно выключить расчёт коррекции радиуса (G40):

ПРАВИЛЬНО

$N(n_s) \quad X(U) \quad G41 \quad \dots$
 $(G41) \quad \dots$

$\dots$
 $(G40)$

$N(n_f) \quad G40 \quad \dots$

или

$G41$
 $N(n_s) \quad X(U) \quad \dots$

$\dots$

$\dots$

$N(n_f) \quad \dots$
 $G40$

НЕПРАВИЛЬНО

$G41$

$N(n_s) \quad X(U) \quad \dots$

$\dots$

$\dots$

$G40$

$N(n_f) \quad \dots$

или

$N(n_s) \quad G41 \quad X(U) \quad \dots$

$\dots$

$\dots$

$N(n_f) \quad \dots$
 $G40$

2-й способ задания:

G18 (Z-X плоскость)

G71 $P(n_s) \quad Q(n_f) \quad U(\Delta u) \quad W(\Delta w) \quad D(\Delta d) \quad F(f) \quad S(s) \quad T(t)$

$N(n_s) \quad X(U) \quad \dots$

$\dots$

$N(n_f) \dots$

G19 (Y-Z плоскость)

G71 $P(n_s) \quad Q(n_f) \quad W(\Delta u) \quad V(\Delta w) \quad D(\Delta d) \quad F(f) \quad S(s) \quad T(t)$

$N(n_s) \quad Z(W) \quad \dots$

$\dots$

$N(n_f) \dots$

G17 (X-Y плоскость)

G71 $P(n_s) \quad Q(n_f) \quad V(\Delta u) \quad U(\Delta w) \quad D(\Delta d) \quad F(f) \quad S(s) \quad T(t)$

$N(n_s) \quad Y(V) \quad \dots$

$\dots$

$N(n_f) \dots$

Вводные параметры второго способа задания совпадают с параметрами первого.

Цикл чернового точения 2-го типа

Цикл чернового точения 2-го типа таким же образом надо задавать, как и 1-го типа, его код G71 и вводные параметры те же самые, как для 1-го типа.

Разница имеется в задании начального кадра контура (кадр с номером n_s). Пока в случае вызова 1-го типа в этом кадре нельзя ссылаться на 1-ую ось по адресу $Z(Y, X)$, то есть движение участка $A-A'$ должно быть перпендикулярно к 1-ой оси $Z(Y, X)$, в то же время в случае вызова 2-го типа в этом кадре обязательно надо ссылаться на адрес $Z(Y, X)$. Значит, участок $A-A'$ не должен быть перпендикулярным к 1-ой оси, $Z(Y, X)$.

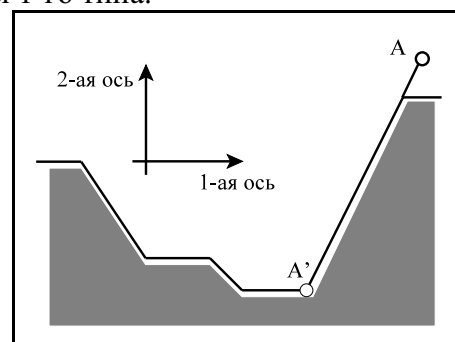


рисунок 18.2.1 -3

Задание 1-го типа

```
G18
G71 U8 R1
G71 P100 Q200 U0.5 W0.2
N100 X(U) ____
...
...
...
N200
```

Задание 2-го типа

```
G18
G71 U8 R1
G71 P100 Q200 U0.5 W0.2
N100 X(U) ____ Z(W) ____
...
...
...
N200
```

В том случае, если приходится пользоваться циклом 2-го типа, однако во вводном кадре контура надо перемещаться только в направлении $X(Z, Y)$, то есть перпендикулярно к оси $Z(Y, X)$, вдоль оси $Z(Y, X)$ следует запрограммировать инкрементное перемещение 0, то есть $Z(Y, X) I0$, или $W(V, U) 0$.

Контур при цикле чернового точения 2-го типа в направлении 2-й оси $X(Z, Y)$ не должен быть монотонно возрастающим, или убывающим, то есть контур может быть отгибающимся, может содержать карманы. Нельзя программировать **припуск на чистовую обработку в направлении 1-й оси** (Δw), должно быть $W(V, U) 0$, в противном случае инструмент может врезаться в один из боков кармана.

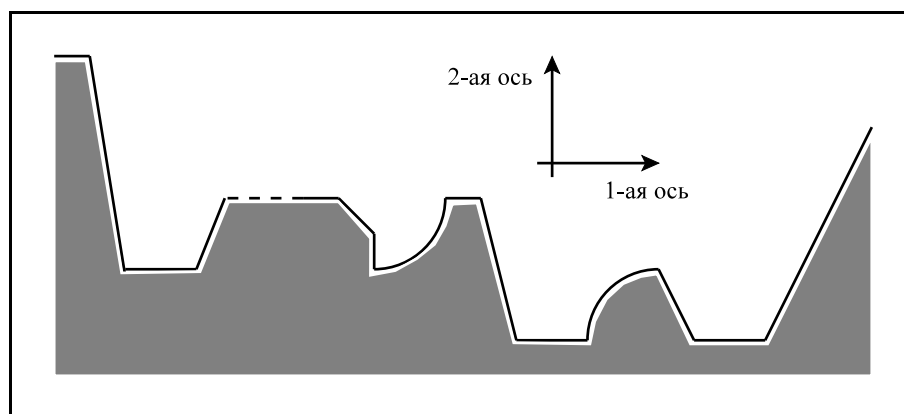


рисунок 18.2.1 -4

18.2.1 Циклы чернового точения (G71)

Однако, в сторону 1-й оси в направлении $Z(Y, X)$ контур и в дальнейшем должен быть **монотонным**, в нём не должен быть отгиба.

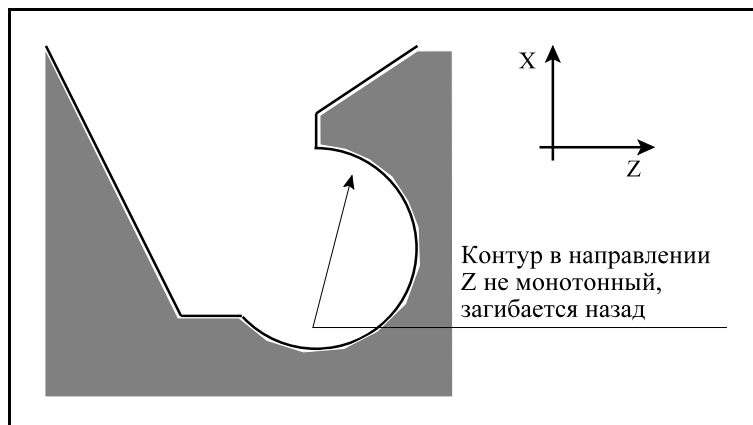


рисунок 18.2.1 -5

При цикле чернового точения 2-го типа отвод выполняется перпендикулярно оси Z , с действующим значением отвода “ e ”.

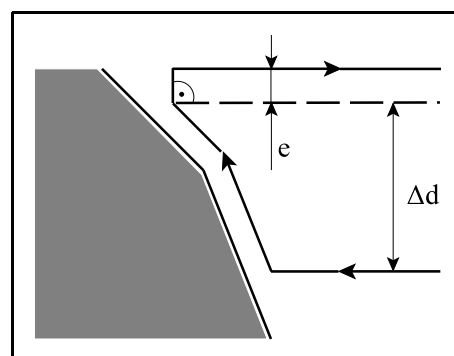


рисунок 18.2.1 -6

Цикл чернового точения 2-го типа на основании положения бита #2 FPT параметра N1611 Turning Cys. Config. может открывать карманы двояким образом. Если FPT:

- =0: цикл начинается с развёрткой **последнего кармана по направлению описания контура**,
- =1: цикл начинается с развёрткой **первого кармана по направлению описания контура**.

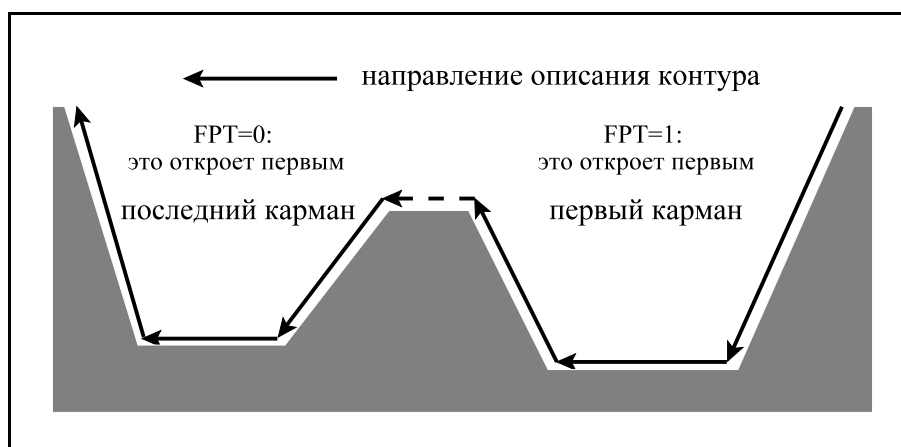


рисунок 18.2.1 -7

Изложенные ниже действительны как для циклов чернового точения 1-го, так и для 2-го типа.

Продолжение программы после выполнения цикла чернового точения

После выполнения цикла чернового точения или с выполнением **кадров, последующих за кадром G71 P Q**, или **после кадра с номером, заданным по адресу Q** продолжается обработка.

В первом случае, если начальный кадр контура следует за кадром G71 P Q, выполнение набегит на описание контура и проделает чистовую обработку.

Во втором случае выполнение программы продолжается кадром, последующим после завершающего кадра, заданным по адресу контура Q, поэтому для чистовой обработки надо программировать цикл чистовой обработки G70. Это полезно тогда, когда тем же инструментом для чернового точения выполняется несколько циклов чернового или торцевого точения, затем загрузив инструмент для чистовой обработки, с использованием цикла чистовой обработки G70 тем же инструментом проведём чистовую обработку различных контуров.

Из перечисленных выше двух возможностей на основании бита #1 SKP параметра N1611 Turning Sys. Config. можно выбирать:

=0: после команды G71, G72, G73 продолжается со следующим кадром,

=1: продолжается кадром, последующим после кадра, заданным в команде G71, G72, G73 по адресу Q.

SKP=0

```
G18
G71 U8 R1
G71 P100 Q200 U0.5 W0.2
(здесь продолжается
программа)
N100
...
N200
(чистовая обработка
завершена)
```

SKP=1

```
G18
G71 U8 R1
G71 P100 Q200 U0.5 W0.2

N100
...
N200
(здесь продолжается
программа)
...
G70 P100 Q200 (чистовая
обработка)
```

Проверка монотонности траектории

Как в цикле чернового точения 1-го типа, так и 2-го типа **вдоль оси чернового точения контур должен быть монотонным**, то есть траектории нельзя содержать отгибающихся элементов. Если траектория не является монотонной, устройство ЧПУ выводит сообщение об ошибке “Контур не монотонный”.

Для монотонности можно установить значение допуска параметром N1612 Tolerance Along Roughing Axis, который принимается во внимание устройством ЧПУ и для циклов G71 и G72. В том случае, если мера отгибания меньше, чем заданное параметром значение, устройство ЧПУ не выводит сообщение об ошибке.

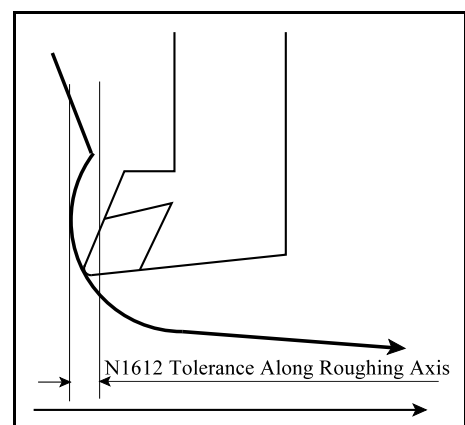


рисунок 18.2.1-8

Цикл чернового точения 1-го типа должен быть **монотонным и вдоль оси, выполняющей точение** (выполняющей врезание), то есть контуру нельзя содержать карман. Если траектория не является монотонной, устройство ЧПУ выводит сообщение об ошибке “Контур не монотонный”.

Для монотонности можно установить значение допуска параметром N1614 Tolerance Along Cutting Axis, который принимается во внимание устройством ЧПУ и для циклов 1 типа, для G71 и G72. В том случае, если мера отгибания меньше, чем заданное параметром значение, устройство ЧПУ не выводит сообщение об ошибке.

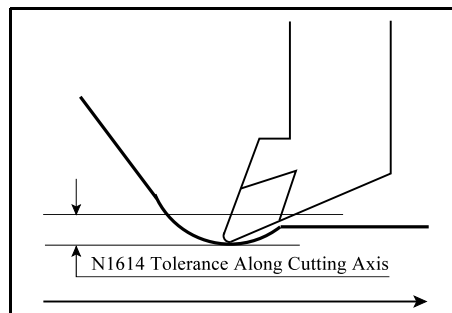


рисунок 18.2.1-9

Как в цикле чернового точения 1-го типа, так и 2-го типа, траектория может быть отгибающейся **при включении коррекции радиуса инструмента** (G41, G42), или **в случае выключения** (G40), даже и тогда, если запрограммированный контур является монотонным.

Причина этого в том, что в состоянии G40 устройство ЧПУ проводит по траектории теоретическую вершину инструмента, а в состоянии G41, G42 проводит окружность вершины параллельно с контуром.

Битом #3 FCK параметра N1611 Turning Sys. Config. можно установить следующее: в циклах чернового точения G71, G72, **проверку монотонности контура**, описанного между кадрами с номером, заданным по адресу P и Q, если значение параметра:

=0: **изменённое с коррекцией радиуса**,

=1: выполняется для первоначальной **траектории, не имеющей коррекцию радиуса**.

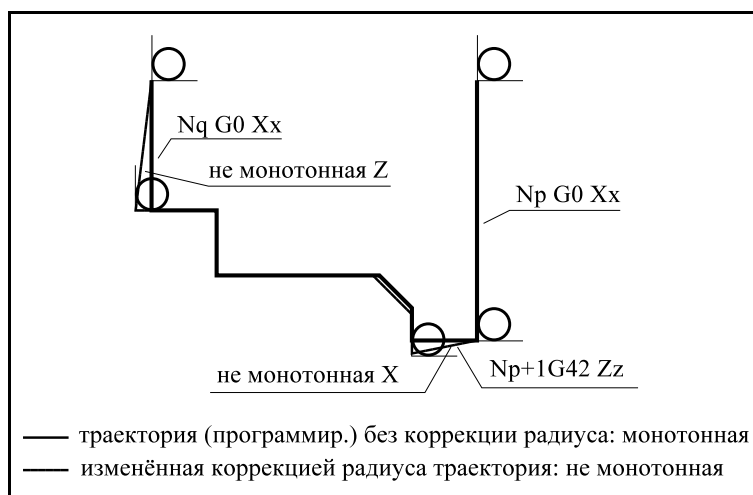


рисунок 18.2.1-10

Образцовые примеры

Перечисленный ниже образцовые примеры предполагают следующие положения бита параметра N1611 Turning Sys. Config.:

#1 SKP=1

#3 FCK=1

Пример для I-го типа цикла G71:

```

G18...
N10 G54 G0 X200 Z50
N20 G92 S3000
N30 G96 S400
N40 T101 M3 (ЧЕРНОВОЙ РЕЗЕЦ, Q3)
N50 G0 X160 Z2
/N60 G71 U4 R1
/N70 G71 P80 Q160 U1 W0.5 F0.5 (D4) (ЧЕРНОВАЯ ОБРАБОТКА)
N80 G0 X100
N90 G1 G42 Z0
N100 X110 ,C2
N110 Z-25
N120 G2 X130 Z-35 R10
N130 G1 X140 ,R2
N140 Z-45 ,C3
N150 X155
N160 G40 X160
N170 G0 X200 Z50
N180 T202 (ЧИСТОВОЙ РЕЗЕЦ, Q3)
N190 G0 X160 Z2
/N200 G70 P80 Q160 (ЧИСТОВАЯ ОБРАБОТКА)
N210 G0 X200 Z50
...

```

Пример для II-го типа цикла G71:

```

G18...
N10 G54 G0 X200 Z50
N20 G92 S3000
N30 G96 S400
N40 T606 M3 (ЧЕРНОВОЙ РЕЗЕЦ, Q8)
N50 G0 X180 Z2
/N60 G71 U6 R4
/N70 G71 P80 Q210 U1 (D6) (ЧЕРНОВАЯ ОБРАБОТКА)
N80 G42 G0 X98 Z2
N90 G1 Z-10
N100 X100
N110 ,A150 (точка пересечения между N110-N120)
N120 X110 Z-40 ,A225
N130 Z-50
N140 ,A150 (точка пересечения между N140-N150)

```

```

N150 G3 X110 Z-110 I55 K-80 R30 Q1 (точка пересечения
                                   между N150-N160)
N160 G1 X100 Z-130 ,A210 Q1 (точка пересечения между N160-
N170)
N170 G2 X100 Z-150 R10 Q-1 (точка пересечения между N170-
N180)
N180 X110 Z-180 I55 K-160 R20 Q-1
N190 G1 X170
N200 Z-190
N210 G40 G0 X180
N220 G0 X200 Z50
N230 T707 (ЧИСТОВОЙ РЕЗЕЦ Q8)
/N240 G70 P80 Q210 (ЧИСТОВАЯ ОБРАБОТКА)
N250 G0 X200 Z50
...

```

Цикл черновой обработки 3-го типа

Цикл черновой обработки 3-го типа можно использовать тогда, если в цикле заданный загибающийся контур, не монотонный в направлении оси X, желаем раскрыть двумя различными инструментами. При этом необходимо запрограммировать *два G71 вызова* для двух различных инструментов, однако *описание контура то же самое для обоих G71 számára*, то есть описание контура не нужно разбить на две части.

При этом первым инструментом *первый цикл раскрывает не загибающиеся части контура*, затем после загрузки второго инструмента *второй цикл - загибающиеся части контура*, то есть карманы.

Программирование цикла черновой обработки 3-го типа

Цикл черновой обработки 3-го типа надо также программировать, как цикл 2-го типа. Единственная разница в том, что цикл 3-го типа обозначается заданием адреса H.

G71 ... H

Истолкование адреса H:

H1: если значение H положительное, циклом раскрываются исключительно части, содержащие не загибающиеся части

H-1: если значение H отрицательное, циклом раскрываются исключительно загибающиеся части

по контуру.

В программе в обоих G71 задаются для кадров, описывающих контур те же самые начальное (адрес P) и завершающее (адрес Q) число кадров:

```
G71 Pp Qq ... H1 (раскрытие не загибающих частей)
```

```
...
```

```
G71 Pp Qq ... H-1 (раскрытие загибающих частей)
```

```
...
```

```
Np X Z (начальный кадр контура)
```

```
...
```

```
Nq (завершающий кадр контура)
```

Образцовый пример к применению цикла чёрного точения 3-го типа

Внимание: Приведенная ниже образцовая программа предполагает положение #1 SKP=0 параметра N1611 Turning Cус. Config.!

```

...
N90 G54 G0 X200 Z50
N100 G92 S3000
N110 G96 S400
N120 T303 M4 (резец чернового точения, Q3)
N130 G0 X180 Z2
/N140 G71 U6 R4
/N150 G71 P240 Q370 U3 H1 (черновое точение монотонных
частей)
N160 G0 X200 Z50
N170 T606 M4 (резец чернового точения, Q8)
/N180 G71 P240 Q370 U3 H-1 (черновое точение карманов)
N190 G0 X200 Z50
N200 T101 M4 (резец чистового точения, Q8)
N210 G70 P240 Q370 (чистовая обработка)
N220 Z50
N230 M30 (конец программы)
N240 G42 G0 X98 Z2 (начало контура)
N250 G1 Z-10
N260 X100
N270 ,A150
N280 X110 Z-40 ,A225
N290 Z-50
N300 ,A150 (X161.962; Z-95.)
N310 G3 X110 Z-110 I55 K-80 R30 Q1
N320 G1 X100 Z-130 ,A210 Q1
N330 G2 X100 Z-150 R10 Q-1
N340 X110 Z-180 I55 K-160 R20 Q-1
N350 G1 X170
N360 Z-190
N370 G40 G0 X180 (конец контура)

```

18.2.2 Цикл торцевого чернового точения (G72)

Цикл торцевого чернового точения (G72) тот же самый, как цикл чернового точения G71, разница в том, что точение совершается параллельно 2-й оси плоскости. **Изложенные в описании цикла G71 действительны и для цикла G72**, поэтому эти случаи здесь отдельно подробно не изложены.

Цикл торцевого чернового точения 1- типа

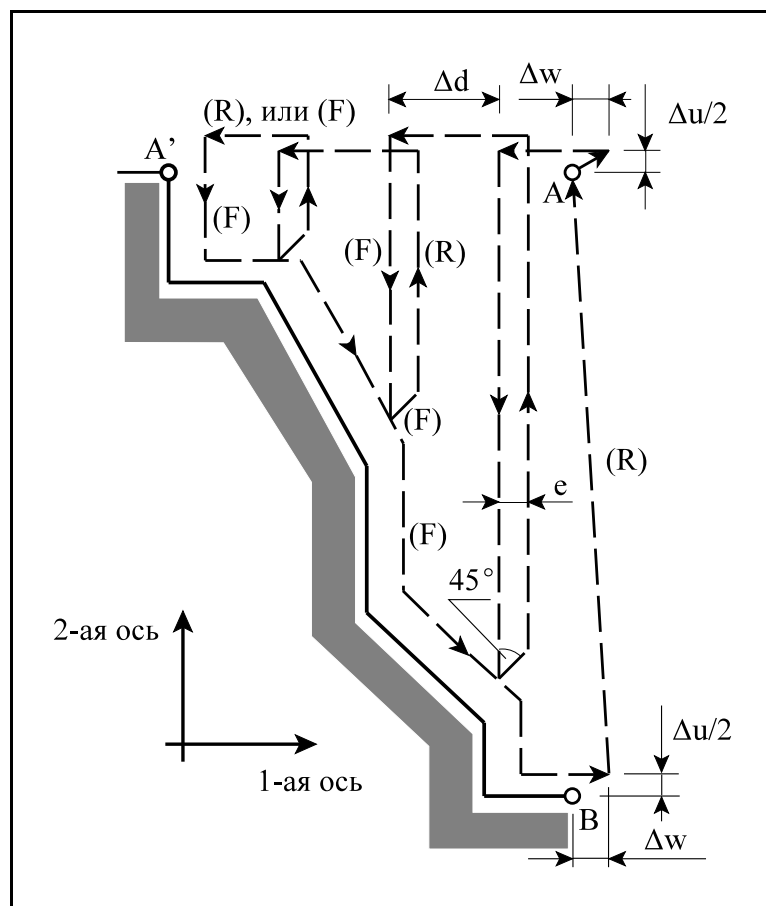


рисунок [18.2.2](#) -1

1-й способ задания:

G18 (Z-X плоскость)

G72 **W**(Δd) **R** (e)

G72 **P** (n_s) **Q** (n_f) **U**(Δu) **W**(Δw) **F**(f) **S**(s) **T**(t)
 $N(n_s)$ $Z(W)$...

...

$N(n_f)$...

G19 (Y-Z плоскость)

G72 **V**(Δd) **R** (e)

G72 **P** (n_s) **Q** (n_f) **W**(Δu) **V**(Δw) **F**(f) **S**(s) **T**(t)
 $N(n_s)$ $Y(V)$...

...

$N(n_f)$...

G78 (X-Y плоскость)

G72 **U**(Δd) **R** (e)

G72 **P** (n_s) **Q** (n_f) **V**(Δu) **U**(Δw) **F**(f) **S**(s) **T**(t)
 $N(n_s)$ $X(U)$...

...

$N(n_f)$...

Значение вводных параметров полностью совпадают изложенными у цикла G71.

2-й способ задания:

G18 (Z-X плоскость)

G72 **P** (n_s) **Q** (n_f) **U**(Δu) **W**(Δw) **D**(Δd) **F**(f) **S**(s) **T**(t)
 $N(n_s)$ $Z(W)$...

...

$N(n_f)$...

G19 (Y-Z плоскость)

G72 **P** (n_s) **Q** (n_f) **W**(Δu) **U**(Δw) **D**(Δd) **F**(f) **S**(s) **T**(t)
 $N(n_s)$ $Y(V)$...

...

$N(n_f)$...

G78 (X-Y плоскость)

G72 **P** (n_s) **Q** (n_f) **V**(Δu) **U**(Δw) **D**(Δd) **F**(f) **S**(s) **T**(t)
 $N(n_s)$ $X(U)$...

...

$N(n_f)$...

Движение между точками $A-A'$ надо задавать в кадре с номером n_s по адресу P. В этом кадре P(n_s) **надо задавать** всегда в случае G18 **движение в направлении Z**, (в случае G19 **в направлении Y**, в случае G17 **в направлении X**), ссылаться на другую ось нельзя.

Циклом можно пользоваться во всех четырёх квадрантах. На рисунке задан знак припуска на чистовую обработку для всех четырёх случаев.

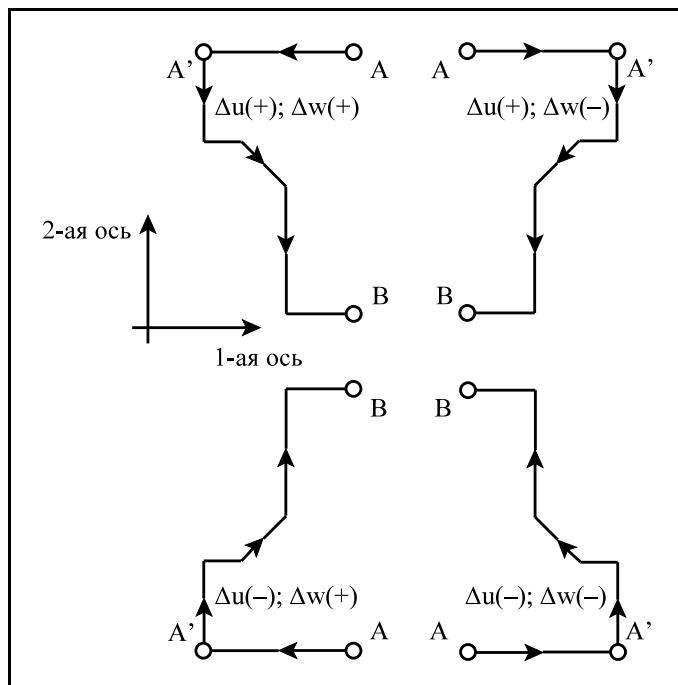


рисунок 18.2.2 -2

Цикл торцевого чернового точения 2- типа

Цикл торцевого чернового точения 2-го типа надо задавать таким же образом, как и 1-го типа, его кода G72 и вводные параметры те же самые, как для 1-го типа.

Разница имеется в задании начального кадра контура (кадр с номером n_s). Пока при вызове 1-го типа в этом кадре нельзя ссылаться на 2-ую ось по адресу X(Z,Y), то есть движение участка $A-A'$ должно быть перпендикулярным к 2-й оси X(Z, Y), в то же время при вызове 2-го типа в этом кадре обязательно **надо сослаться на адрес X(Z, Y)**. Значит, участок $A-A'$ не должен быть перпендикулярным к 2-й оси, X(Z, Y).

Образцовые примеры

Приведенные ниже образцовые примеры предполагают следующие положения бита параметра N1611 Turning Cус. Config.:

#1 SKP=1
#3 FCK=1

Пример для цикла I-го типа G72:

```
G18...
N10 G54 G0 X200 Z50
N20 G92 S3000
N30 G96 S600
N40 T101 M3 (ЧЕРНОВОЙ РЕЗЕЦ, Q3)
N50 G0 X161 Z2
/N60 G72 W5 R1
/N70 G72 P80 Q140 U1 W0.5 (D5) F0.5 (ЧЕРНОВОЕ ТОЧЕНИЕ)
```

```

N80 G41 G0 Z-25
N90 G1 X120 ,C3
N100 Z-17 ,R2
N110 G3 X100 Z-7 R10
N120 G1 Z-3 ,C1
N130 X80
N140 G40 Z2
N150 G0 X200 Z50
N160 T202 (ЧИСТОВОЙ РЕЗЕЦ Q3)
N170 X161 Z2
/N180 G70 P80 Q140 (ЧИСТОВОЕ ТОЧЕНИЕ)
N190 G0 X200 Z50
...

```

Пример для цикла II-го типа G72:

```

G18...
N10 T505 (ЧЕРНОВОЙ РЕЗЕЦ, Q7)
N20 G54 G0 X200 Z50
N30 G92 S3000
N40 G96 S200 M3
N50 G0 X160 Z5
/N60 G72 W4 R1
/N70 G72 P80 Q150 W0.5 (D4) (ЧЕРНОВОЕ ТОЧЕНИЕ)
N80 G41 G0 X160 Z-25
N90 G1 X145 ,R4
N100 X125 Z-10 ,R3
N110 X105 Z-25 ,R4
N120 X95
N130 G2 X15 Z-25 R20 ,R5
N140 G1 X0
N150 G40 Z5
N160 G0 X200 Z50
N170 T202 (ЧИСТОВОЙ РЕЗЕЦ ? Q7)
N180 X160 Z5
/N190 G70 P80 Q150 (ЧИСТОВОЕ ТОЧЕНИЕ)
N200 G0 X200 Z50
...

```

Цикл торцевого чернового точения 3-го типа

Цикл торцевого чернового точения 3-го типа можно использовать тогда, если заданный в цикле, не загибающийся, в направлении оси Z не монотонный контур желаем раскрыть двумя различными инструментами..В это время придётся запрограммировать *два вызова по G72* для различных инструментов, однако *описание контура то же самое для обоих G72*, то есть описание контура не нужно разделить на две части.

При этом первым инструментом *первый цикл раскрывает не загибающуюся часть контура*, затем, после загрузки второго инструмента *второй цикл - загибающиеся части контура*, то есть карманы.

Программирование цикла торцевого чернового точения 3-го типа

Цикл торцевого чернового точения 3-го типа необходимо программировать также, как цикл 2-го типа. Единственная разница в том, что цикл 3-го типа обозначается заданием адреса Н.

G72 ... Н

Истолкование адреса Н:

Н1: если значение Н положительное, циклом раскрываются исключительно не загибающиеся части,

Н-1: если значение Н отрицательное, циклом раскрываются исключительно загибающиеся части

по контуру.

В программе в обеих G72 задаются для кадров, описывающих контур, те же самые начальное (адрес Р) и завершающее (адрес Q) число кадров:

G72 Рr Qq ... Н1 (раскрытие не загибающихся частей)

...

G72 Рr Qq ...Н-1 (раскрытие загибающихся частей)

...

Нr X Z (начальный кадр контура)

...

Нq (завершающий кадр контура)

Образцовый пример к применению цикла торцевой черновой обработке 3-го типа
Внимание: Заданный ниже оразцовый пример предполагает положение #1 SKP=0 параметра N1611 Turning Sys. Config.!

```

...
N80 G54 G0 X200 Z50
N90 G92 S3000
N100 G96 S200
N110 T303 M4 (резец черновой обработки, Q3)
N120 G0 X160 Z5
/N130 G72 W5 R3
/N140 G72 P230 Q300 W2 H1 (черновая обработка монотонных
частей)
N150 G0 X200 Z50
N160 T404 S1=500 M4 (резец чистовой обработки, Q7)
/N170 G72 P230 Q300 W2 H-1 (черновая обработка карманов)
N180 G0 X200 Z50
N190 T505 S1=500 M4 (резец чистовой обработки, Q7)
N200 G70 P230 Q300 (чистовая обработка)
N210 G0 X200 Z50
N220 M30 (конец программы)
N230 G41 G0 X160 Z-25 (начало контура)
N240 G1 X145 ,R4
N250 X125 Z-10 ,R3
N260 X105 Z-25 ,R4
N270 X95
N280 G2 X15 Z-25 R20 ,R5
N290 G1 X0 (Z-30)
N300 G40 (X-2) Z5 (конец контура)

```

18.2.3 Цикл повторения фигуры (G73)

Этим циклом можно пользоваться при черновой обработке закованных, литых, или предварительно обдирочных заготовок, где контуры окончательной формы уже стоят на расстоянии. Цикл повторяет шаг за шагом описанный в программе контур.

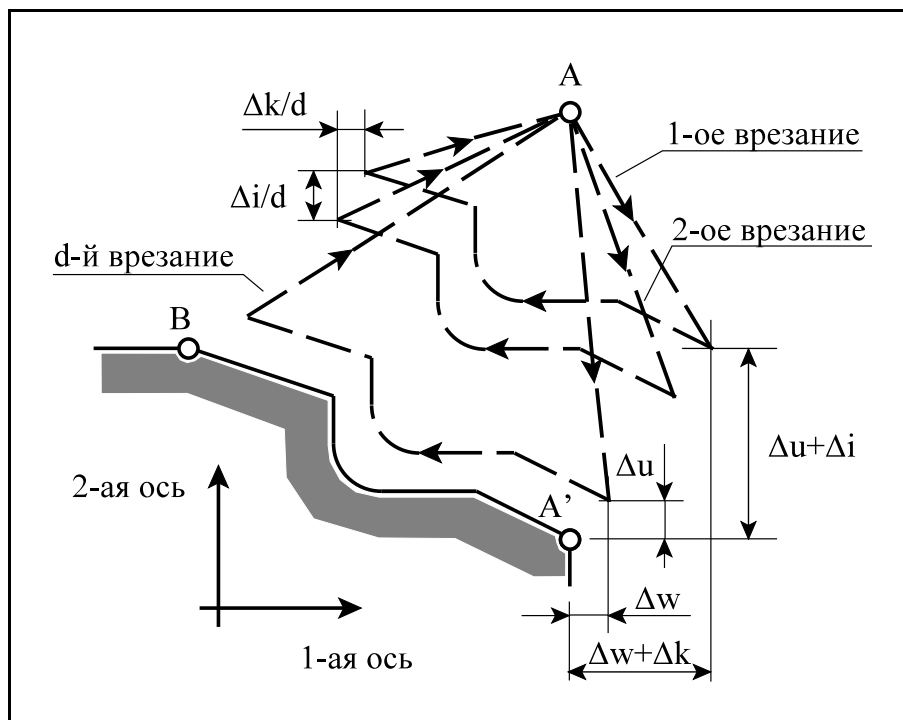


рисунок [18.2.3](#) -1

1-й способ задания:

G18 (Z-X плоскость)

G73 U(Δi) W(Δk) R(d)G73 P(n_s) Q(n_f) U(Δu) W(Δw) F(f) S(s) T(t)N(n_s) ...

...

N(n_f) ...

G19 (Y-Z плоскость)

G73 W(Δi) V(Δk) R(d)G73 P(n_s) Q(n_f) W(Δu) V(Δw) F(f) S(s) T(t)N(n_s) ...

...

N(n_f) ...

G17 (X-Y плоскость)

G73 V(Δi) U(Δk) R(d)G73 P(n_s) Q(n_f) V(Δu) U(Δw) F(f) S(s) T(t)N(n_s) ...

...

N(n_f) ...

Выполняется с рядом команд, где:

- Δi :** величина и направление припуска на черновую обработку вдоль 2-й оси выбранной плоскости. Число *со знаком* понимается всегда и в *радиусе*. Если значение припуска на черновую обработку не задаётся в программе, тогда устройство ЧПУ берёт его с параметра N1603 Relief Ax2.
- Δk :** величина и направление припуска на черновую обработку вдоль 1-й оси выбранной плоскости. Число *со знаком* понимается всегда и в *радиусе*. Если значение припуска на черновую обработку не задаётся в программе, тогда устройство ЧПУ берёт его с параметра N1602 Relief Ax1.
- d:** число врезаний. Если число врезаний не задаётся в программе, тогда устройство ЧПУ берёт его с параметра N1604 Numb of Div.
- n_s :** номер начального кадра фрагмента программы (участок A—A'—B), описывающей чистовую обработку.
- n_f :** номер завершающего кадра фрагмента программы (участок A—A'—B), описывающей чистовую обработку.
- Δu :** величина и направление припуска на чистовую обработку вдоль 2-й оси выбранной плоскости. Число *со знаком*, и понимается в зависимости от истолкования координаты в *диаметре*, или в *радиусе*.
- Δw :** величина и направление припуска на чистовую обработку вдоль 1-й оси выбранной плоскости. Число *со знаком*, и понимается в зависимости от истолкования координаты в *диаметре*, или в *радиусе*.

Значение Δi и Δk , или Δu и Δw , заданные в кадре G73 по адресу U(W, V) и W(V, U) зависят от того, что в данном кадре запрограммировали ли P и Q. То есть, если запрограммировали P и Q, тогда значение U и W будет Δu и Δw , если не запрограммировали, тогда Δi и Δk .

Цикл выполняет их в кадрах, содержащих P и Q. Цикл можно выполнить во всех четырёх квадрантах, в зависимости от знака значений Δi , Δk , Δu , Δw . После каждого врезания и в конце цикла инструмент возвращается в точку "A".

Функции F, S, T, запрограммированные в фрагментах программы между кадрами n_s и n_f , не принимаются во внимание и выполняются те, которые в кадре G73 имеются (f, s, t), или раньше были запрограммированы. То же самое относится к постоянной скорости резания, запрограммированной между кадрами n_s и n_f , то есть осуществляется **состояние G96, или G97 перед кадром G73, и значение постоянной скорости резания.**

Кадрам с номером от n_s до n_f нельзя содержать вызова подпрограммы.

Задание расчёта коррекции радиуса инструмента возможно в кадрах, описывающих цикл, с ограничениями, изложенными в описании функции G71.

Продолжение программы после выполнения цикла повторения фигуры

После выполнения цикла обработка продолжается или выполнением **кадров, последующих за кадром G73 P Q**, или **после кадра с номером, заданным по адресу Q**. В последнем случае циклом чистового точения G70 можно выполнить чистовую обработку заготовки.

На основании бита #1 параметра N1611 Turning Sys. Config. можно выбирать из перечисленных выше двух возможностей:

=0: после команды G71, G72, G73 продолжается со следующим кадром,

=1: продолжается кадром, последующим после кадра, заданным в команде G71, G72, G73 по адресу Q.

2-й способ задания:

G18 (Z-X плоскость)

G73 P (n_s) **Q** (n_f) **U**(Δu) **W**(Δw) **I**(Δi) **K**(Δk) **D**(d) **F**(f) **S**(s) **T**(t)
 $N(n_s)$...
 ...
 $N(n_f)$...

G19 (Y-Z плоскость)

G73 P (n_s) **Q** (n_f) **W**(Δu) **V**(Δw) **K**(Δi) **J**(Δk) **D**(d) **F**(f) **S**(s) **T**(t)
 $N(n_s)$...
 ...
 $N(n_f)$...

G17 (X-Y плоскость)

G73 P (n_s) **Q** (n_f) **V**(Δu) **U**(Δw) **J**(Δi) **I**(Δk) **D**(d) **F**(f) **S**(s) **T**(t)
 $N(n_s)$...
 ...
 $N(n_f)$...

Вводные параметры второго способа задания совпадают с первыми.

Образцовый пример

Приведенный ниже образцовый пример предполагает установку бита #1 SKP=1 параметра N1611 Turning Cys. Config.

```
G18...
N10 G96 S200
N20 G92 S3000
N30 G54 G0 X70 Z20
N40 T111 M3 (ЧЕРНОВОЙ РЕЗЕЦ Q3)
/N50 G73 U10 W4 R4
/N60 G73 P70 Q130 U1 W0.5 (I10 K4 D4)
N70 G42 X10 Z5 F1.4
N80 G1 Z0
N90 X20 ,C1
N100 Z-10
N110 G2 X40 Z-20 R10
N120 G1 Z-30
N130 G40 G0 X42
N140 G0 X70 Z20
N150 T212 (ЧИСТОВОЙ РЕЗЕЦ Q3)
/N160 G70 P70 Q130
N170 G0 X70 Z20
...
```

18.2.4 Цикл чистового точения (G70)

После выполнения чернового точения контура с G71, G72, или G73, описанного кадрами с номером от n_s до n_f , чистовое точение контура выполняется командой G70. Задание чистового точения происходит со следующей командой:

G70 P (n_s) Q (n_f)

n_s : номер начального кадра фрагмента программы, описывающей чистовое точение.

n_f : номер завершающего кадра фрагмента программы, описывающей чистовое точение.

В ходе цикла выполняются функции *F, S, T*, запрограммированные в фрагменте программы, описывающей чистовое точение от n_s до n_f , далее **команды**, относящиеся к **расчёту постоянной скорости резания**, в отличие от цикла G71, G72, G73.

В конце цикла чистового точения инструмент возвращается в исходную точку и зачитается кадр, последующий за G70.

В ходе цикла чистового точения **работает** расчёт **коррекции радиуса инструмента**.

Кадрам с номером от n_s до n_f нельзя содержать вызова подпрограммы.

Для использования циклов чистового точения имеются примеры у циклов G71, G72, G73.

Z(Y, X): абсолютная координата точки “С”, заданной на 1-й оси плоскости

W(V, U): расстояние точки АС инкрементно

Δi: величина врезания на 2-й оси плоскости. *Заданное в радиусе* положительное число.

Δk: величина врезания на 1-й оси плоскости. *Заданное в радиусе* положительное число.

Δd: мера отвода инструмента вниз. Число *заданное в радиусе*. Направление движения всегда противоположное знаку вектора АВ. В конце 1-й проточки пропускается перемещение Δd.

F: подача

На рисунке обозначены с буквой (F) те участки, которые совершаются подачей и буквой (R) те, которые быстрым ходом

Заполнением адреса Z(Y, X), или W(V, U) решается, что в кадре G74 заполнение адреса R определяет e, или Δd. Если адрес заполнен, значение адреса R будет Δd.

Если пропустить заполнение адреса X(Z, Y), или U(W, V) и также заполнение адреса P(Δi), тогда движение происходит только вдоль оси Z, то есть выполняется цикл сверления.

2-й способ задания:

G18 (Z-X плоскость)

G74 X_p(U) Z_p(W) I (Δi) K (Δk) D (Δd) F

G19 (Y-Z плоскость)

G74 Y_p(V) Z_p(W) K (Δi) J (Δk) D (Δd) F

G17 (X-Y плоскость)

G74 X_p(U) Y_p(V) J (Δi) I (Δk) D (Δd) F

Вводные параметры второго способа задания совпадают с первыми.

Образцовый пример

G18...

G0 X100 Z5 S1=1500 M3 F1

G74 R6

G74 X70 Z-100 P6 (I6) Q25 (K25) R2 (D2) F1

G0 X120 Z10

...

1-й способ задания:

G75 R (e)

G75 $X_p(U)$ $Z_p(W)$ $P(\Delta i)$ $Q(\Delta k)$ $R(\Delta d)$ **F**

G75 R (e)

G75 $Y_p(V)$ $Z_p(W)$ $P(\Delta i)$ $Q(\Delta k)$ $R(\Delta d)$ **F**

G75 R (e)

G75 $X_p(U)$ $Y_p(V)$ $P(\Delta i)$ $Q(\Delta k)$ $R(\Delta d)$ **F**

223

2-й способ задания:

G18 (Z-X плоскость)

G75 X_p(U) Z_p(W) I (Δi) K (Δk) D (Δd) F

G19 (Y-Z плоскость)

G75 Y_p(V) Z_p(W) K (Δi) J (Δk) D (Δd) F

G17 (X-Y плоскость)

G75 X_p(U) Y_p(V) J (Δi) I (Δk) D (Δd) F

Вводные параметры второго способа задания совпадают с первыми.

Образцовый приер

G18 . . .

G0 X105 Z-15 S1=1500 M3 F1

G75 R6

G75 X20 Z-30 P15 (I15) Q6 (K6) R2 (D2) F1

G0 X120 Z10

. . .

18.2.7 Цикл нарезания резьбы (G76)

Приложенный рисунок показывает движение цикла нарезания резьбы типа G76.

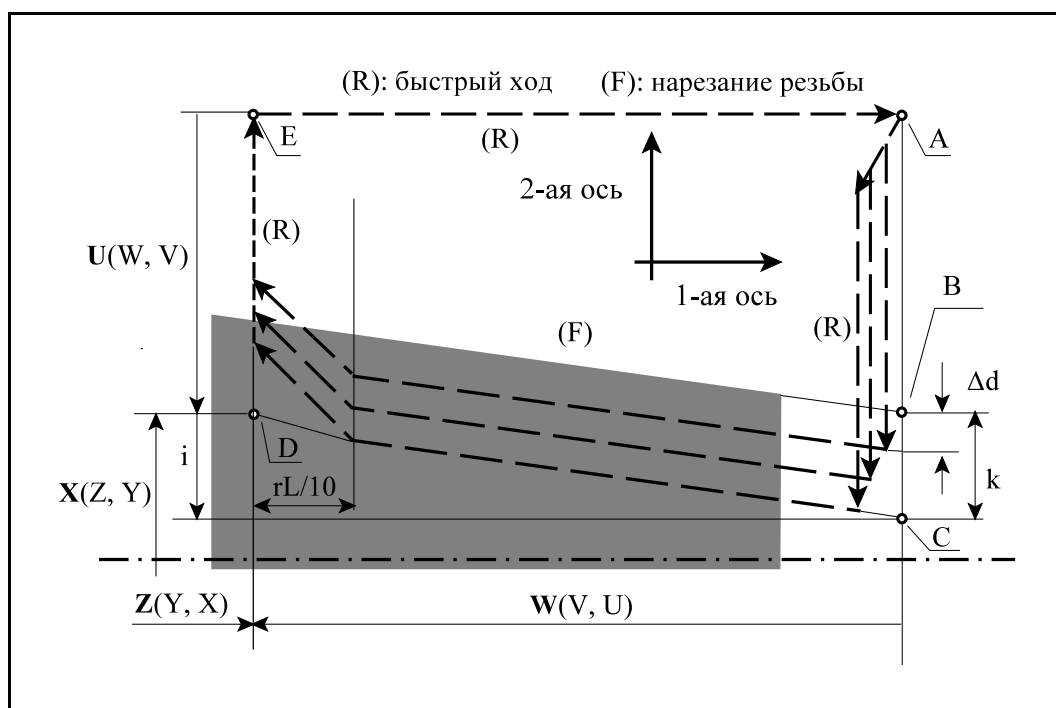


рисунок 18.2.7 -1

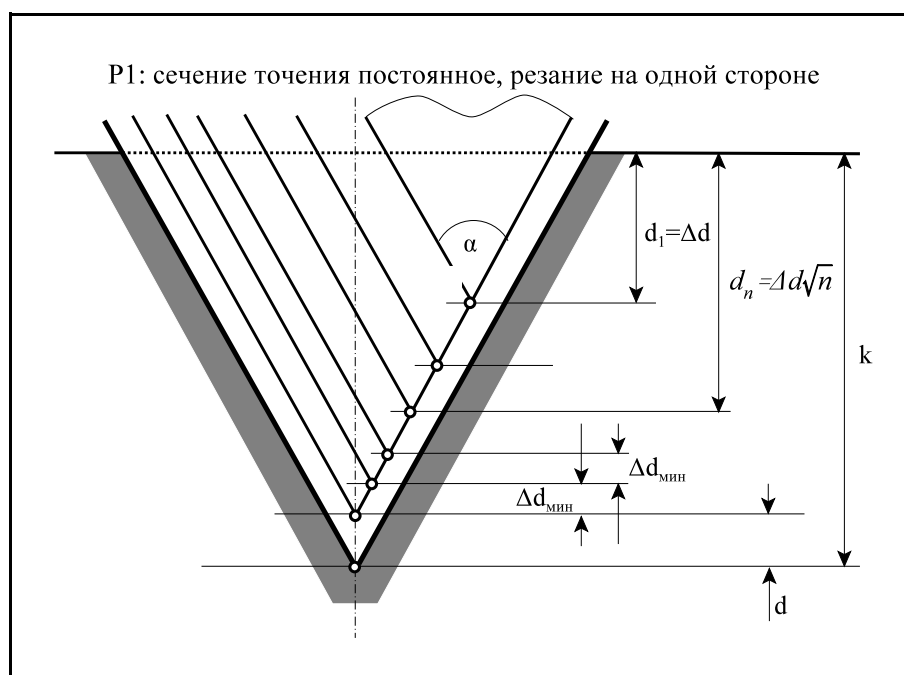


рисунок 18.2.7 -2

1-й способ задания:

G18 (Z-X плоскость)

G76 P (n) (r) (α) Q ($\Delta d_{\min}$) R (d)G76 X_p(U) Z_p(W) P (k) Q (Δd) R (i) F(E)(L)

G19 (Y-Z плоскость)

G76 P (n) (r) (α) Q ($\Delta d_{\min}$) R (d)G76 Y_p(V) Z_p(W) P (k) Q (Δd) R (i) F(E)(L)

G17 (X-Y плоскость)

G76 P (n) (r) (α) Q ($\Delta d_{\min}$) R (d)G76 X_p(U) Y_p(V) P (k) Q (Δd) R (i) F(E)(L)

совершается с рядом команд.

Перечисленные ниже параметры являются вводными данными первого кадра **G76 P (n) (r) (α) Q ($\Delta d_{\min}$) R (d):**

n: *число циклов чистового точения* (n=01...99)

Это значение передаётся модально, и не меняется до тех пор, пока не изменить его. Если в программе не задаётся число циклов чистового точения, тогда устройство ЧПУ берёт его с параметра N1608 Count Fin.

r: *мера фаски* (r=01...99)

При сбегае из резьбы устройство ЧПУ отводит инструмент под углом, заданным параметром N1607 Chmfr Ang. С помощью “r” задаётся длина фаски отвода в зависимости от подъёма нарезки. Длина участка:

$$r \cdot L / 10$$

где: L: запрограммированный подъём нарезки

Это значение передаётся модально, и не меняется до тех пор, пока не изменить его. Если в программе не задаётся мера фаски, тогда устройство ЧПУ берёт её с параметра N1606 ThrdChmfr.

α : *угол лезвия резьбового резца* в градусах (α =01...99)

Это значение передаётся модально, и не меняется до тех пор, пока не изменить его. Если в программе не задаётся угол лезвия резьбового резца, тогда устройство ЧПУ берёт его с параметра N1612 Tool Tip Angle.

Значение n, r и α можно задавать одновременно по адресу P. Поскольку каждое значение выражается двузначным числом, на адрес P надо написать шестизначное число. Например, если число циклов чистового точения n=2, значение фаски 1.5L (r=15), и работаем с резцом 60°, тогда значение P: P021560.

$\Delta d_{\min}$: *значение минимального врезания (положительное число, понимается всегда в радиусе)*

Если в ходе нарезания резьбы в n-ом цикле значение врезания $d_n - d_{n-1} < \Delta d_{\min}$, тогда значение врезания ограничивается всегда с значением $\Delta d_{\min}$. Это значение передаётся модально, и не меняется до тех пор, пока не изменить его. Если в программе не задаётся значение минимального врезания, тогда устройство ЧПУ берёт его с параметра N1609 Min Thrd Cut.

d: *припуск на чистовую обработку (положительное число, понимается всегда в радиусе)*

Это значение передаётся модально, и не меняется до тех пор, пока не изменить его.

Если в программе не задаётся значение припуска на чистовую обработку, тогда устройство ЧПУ берёт его с параметра N1610 Fin Allow.

Устройство ЧПУ выполняет приём перечисленных выше параметров на код G76, если в кадре G76 не заполнен адрес ни X(Z, Y) U(W, V), ни Z(Y, X) W(V, U).

Перечисленные ниже параметры являются вводными данными второго кадра **G76 X(U) Z(W) R(i) P(k) Q(Δd) F(E)(L)**:

X_p(Z_p, Y_p): абсолютная координата точки “D”, заданной на 2-й оси плоскости

U(W, V): расстояние точки AD инкрементно

Z_p(Y_p, X_p): абсолютная координата точки “D”, заданной на 1-й оси плоскости

W(V, U): расстояние точки AD инкрементно

i: *мера конусности a* (понимается всегда в радиусе)

Если i=0, или адрес R не заполнять, нарезается цилиндрическая резьба.

k: *глубина резьбы (положительное число, понимается всегда в радиусе)*

Δd: *глубина первого врезания (положительное число, понимается всегда в радиусе)*

L: *подъём нарезки*

Его программирование совпадает с изложенным при G33. Записанное значение *на адрес F* означает *подъём нарезки*, Записанное значение *на адрес E* означает *число витков за дюйм*.

Нарезание резьбы выполняется только кадром, заполненным согласно вышеописанному, что означает, что адреса X(Z, Y) U(W, V), и Z(Y, X) W(V, U) надо заполнять. Если не заполнен ни одного адреса координат, тогда кадр понимается за кадром установки параметров.

В ходе выполнения цикла *врезание быстрым ходом* выполняется, если за время цикла действует код **G0**, далее модальной *подачей*, если за время цикла действует код **G1**.

Между точками A C и D совершается движение подачи, согласно подъёму нарезки L, заданной по адресу F(E). Остальные участки выполняются быстрым ходом.

Резьба нарезается всегда согласно способу по рисунку, на одной стороне по зависимости

$$d_n = \Delta d \sqrt{n}$$

так, чтобы поперечное сечение точения было постоянное.

В *конце резьбы* нарезается всегда *фаска*, согласно установленным параметрам.

Направление запрограммированного по адресам X(Z, Y) U(W, V), Z(Y, X) W(V, U) движения определяет направление врезания, а также направление нарезания резьбы, а знак адреса R - направление уклона конуса.

Под действием STOP соответственно запрограммированной фаске, отводится инструмент, согласно изложенным у кода G78, затем наводится в начальную точку (A). Под действием клавиши START прерванное врезание начнётся сначала.

2-й способ задания:

G18 (Z-X плоскость)

G76 X_p(U) Z_p(W) I(i) K(k) D(Δ d) A(α) F(E)(L) Q P

G19 (Y-Z плоскость)

G76 Y_p(V) Z_p(W) K(i) J(k) D(Δ d) A(α) F(E)(L) Q P

G17 (X-Y плоскость)

G76 X_p(U) Y_p(V) J(i) I(k) D(Δ d) A(α) F(E)(L) Q P

Устройство ЧПУ считает задание кадра *согласному 2-му способу*, если в кадре *заполнен адрес K(J, I)*.

Вводные параметры n, r, α , $\Delta d_{\min}$ и d также принимаются во внимание устройством ЧПУ и при задании резьбы 2-им способом, как при 1-ом, или с кадром установки параметров **G76 P** (n) (r) (α) **Q** ($\Delta d_{\min}$) **R** (d), или с параметра.

По адресу “A” (α) можно задавать угол лезвия резьбового резца, подобно 1-му способу. Разница в том, что пока у 1-го способа можно задавать только с разбивкой до 1°, целым числом, в то же время у 2-го способа можно и дробным числом. Если адрес “A” не заполнен, его значение α берётся с параметра N1612 Tool Tip Angle.

Истолкование данных i, k, Δd , L совпадает изложенным у 1-го способа. Истолкование остальных адресов:

Q: *угловое значение начала резьбы, отсчитанное от нулевого импульса датчика, заданно в °-ах.* Истолкование адреса совпадает с изложенным у G33.

P: *способ нарезания резьбы.*

Можно выбирать между четырьмя способами врезания, на основании рисунков ниже

P1: поперечное сечение точения постоянное, боковое врезание

P2, P5: поперечное сечение точения постоянное, врезание двустороннее

P3: врезание постоянное, боковое врезание

P4: врезание постоянное, врезание двустороннее

Образцовый пример

G18...

G97 S1000 M3

G0 X36 Z4

G76 P010560 Q0.3 R0.2

G76 X24 Z-50 P3 Q1 R-4 F3 (1-й способ задания)

ИЛИ

G76 X24 Z-50 K3 D1 I-4 A60 F3 P2 (2-й способ задания)

G0 X100 Z50

...

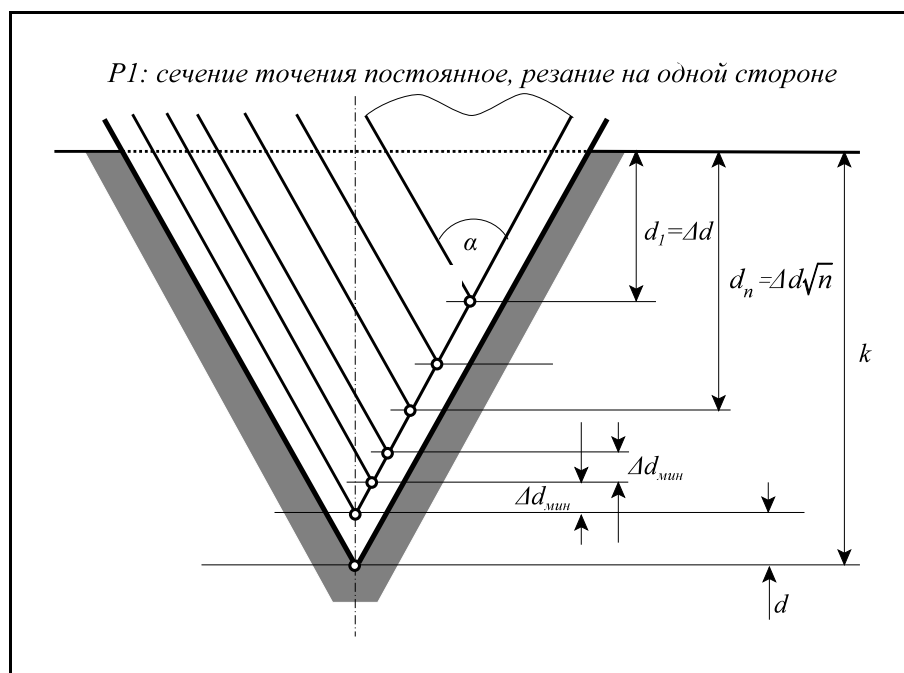


рисунок 18.2.7 -4

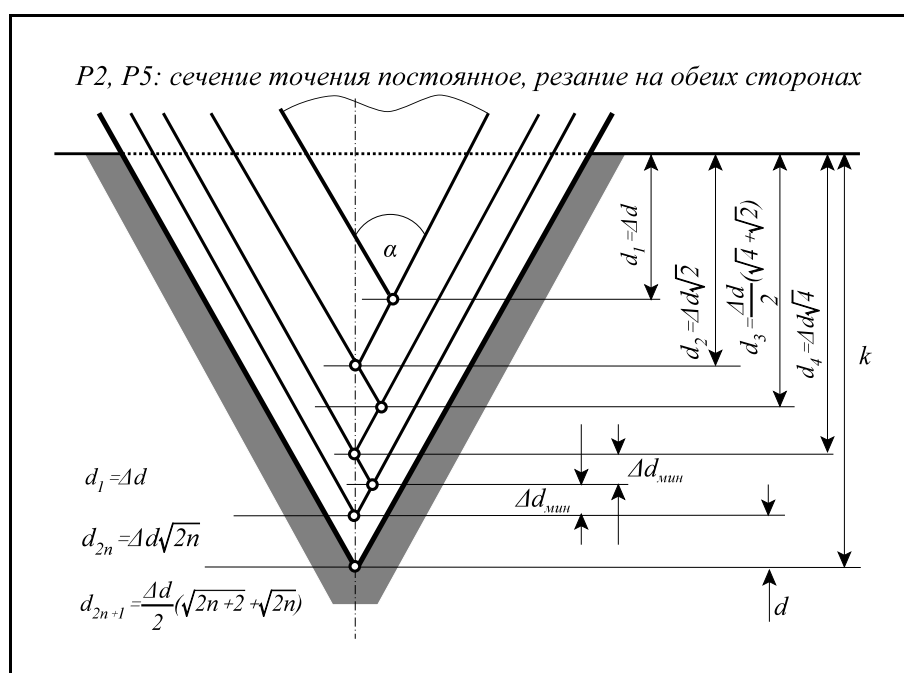


рисунок 18.2.7 -5

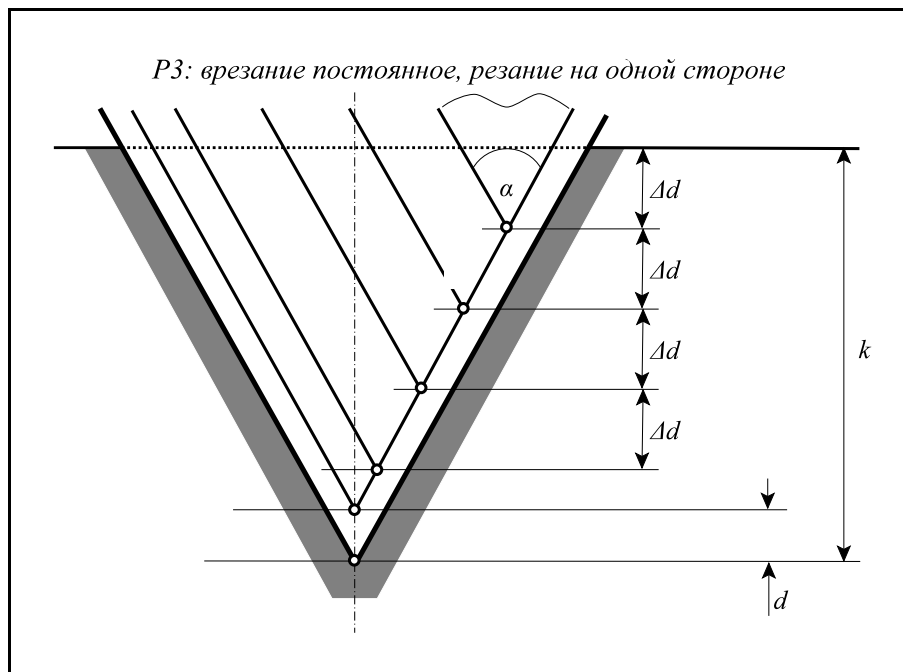


рисунок 18.2.7 -6

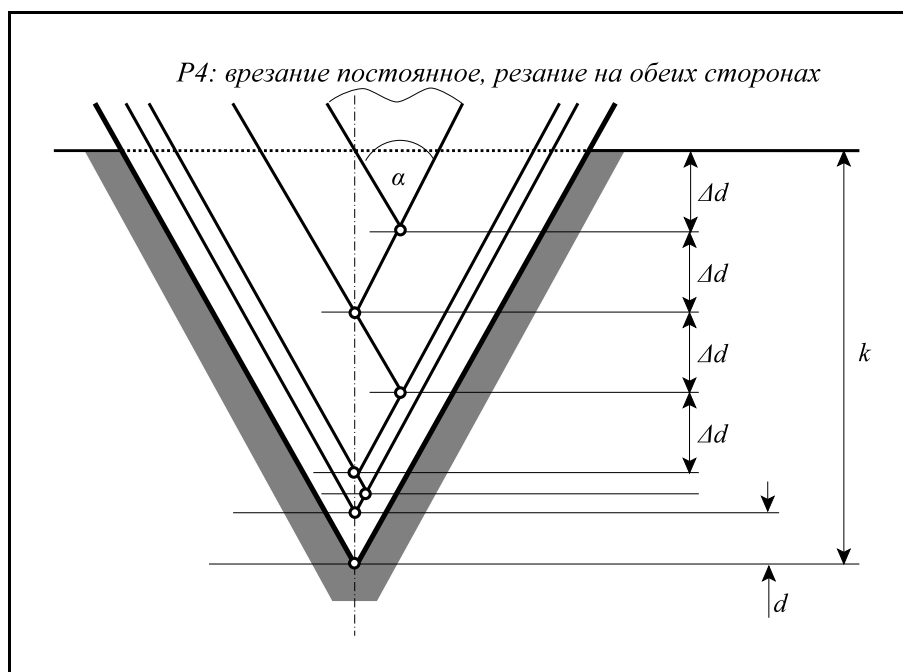


рисунок 18.2.7 -7

19 Циклы сверления

Циклы сверления можно делить на следующие операции:

- | | |
|----------------|---|
| 1-я операция: | позиционирование в выбранной плоскости |
| 2-я операция: | деятельность после позиционирования |
| 3-я операция: | движение быстрым ходом до точки R (точка приближения) |
| 4-я операция: | деятельность в точке R |
| 5-я операция: | сверление до упорной точки |
| 6-я операция: | деятельность в упорной точке |
| 7-я операция: | отвод то точки R |
| 8-я операция: | деятельность в точке R |
| 9-я операция: | отвод быстрым ходом до исходной точки |
| 10-я операция: | деятельность в исходной точке |

Точка R, точка приближения: Инструмент приближается к заготовке до этой точки с движением быстрого хода.

Исходная точка: Позиция оси сверления, которую примет перед запуском цикла.

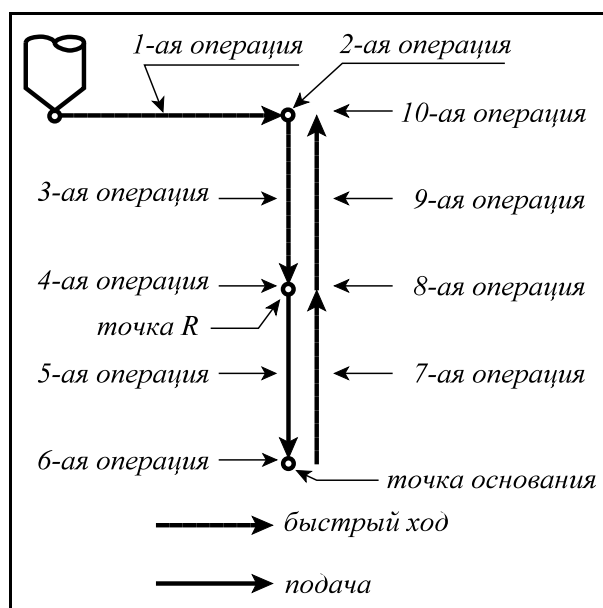


рисунок 19-8

Перечисленные выше операции охватывают общее описание циклов сверления, в конкретных случаях операции могут быть пропущены.

Циклы сверления имеют **плоскость позиционирования**, и **ось сверления**. Плоскость позиционирования и ось сверления выделяется командами выбора плоскости G17, G18, G19.

код G	плоскость позиционирования	сверлильная ось
G17	плоскость $X_p Y_p$	Z_p
G18	плоскость $Z_p X_p$	Y_p
G19	плоскость $Y_p Z_p$	X_p

где: X_p : ось X, или параллельная ей ось

Y_p : ось Y, или параллельная ей ось

Z_p : ось Z, или параллельная ей ось

Оси U, V, W считаются параллельными осям тогда, если они таким определены параметром N0103 Axis to Plane.

При программировании торцевого сверления, где осью сверления является Z, надо выбрать плоскость G17, при боковом сверлении, где осью сверления является X, надо выбрать плоскость G19.

Конфигурация циклов сверления возможно командами G98 и G99:

G98: инструмент в ходе циклов сверления *отводится до исходной точки*. Основное положение это то, что примет устройство ЧПУ после включения, reset или удаления режима цикла.

G99: инструмент в ходе циклов сверления *отводится до точки R*, следовательно при этом пропускается операция 9., 10.

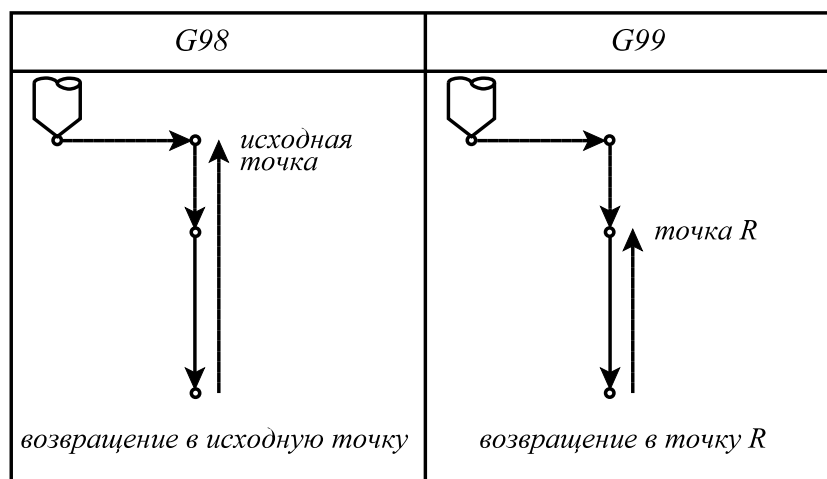


рисунок 19-9

Коды циклов сверления: G83.1, G84.1, G86.1, G81, ..., G89

Этими кодами включается режим циклов, позволяющий передачу переменных циклов модально.

Код G80 выключает режим циклов и удаляет переменных циклов.

Адреса, использованные в циклах сверления и их истолкование:

G17	G	X _p	Y _p	q	I	J	Z _p	R	Q	E	P	F	S	L
G18	G	Z _p	X _p	q	K	I	Y _p	R	Q	E	P	F	S	L
G19	G	Y _p	Z _p	q	J	K	X _p	R	Q	E	P	F	S	L

число повторения
данные сверления

сдвиг после ориентировки шпинделя
позиция отверстия

код сверления

Код сверления:

Истолкование отдельных кодов см. позже.

Коды передаются модально до тех пор, пока не запрограммировать команду G80, или код, относящийся к 1-й группе код G (группа интерполяции: G01, G02, G03, G33), далее пока не удаляется при смене режима.

Пока включено состояние цикла командами G83.1, G84.1, G86.1, G81, ..., G89, до тех пор модальные переменные цикла передаются и между циклами сверления различного типа.

Начальная точка или точка исхода:

Начальная точка - это позиция оси, выделённой для сверления, которая регистрируется:

– при включении режима цикла. Например: в случае

```
N1 G17 G90 G0 Z200
N2 G81 X0 C0 Z50 R150
N3 X100 C30 Z80
```

позиция начальной точки Z=200 и в кадре N2 и N3.

– или при выделении новой оси сверления. Например:

```
N1 G17 G90 G0 Z200 W50
N2 G81 X0 C0 Z50 R150
N3 X100 C30 W20 R25
```

В кадре N2 позиция начальной точки Z=200

В кадре N3 позиция начальной точки W=50

В том случае, если меняется выделения оси сверления, обязательно запрограммировать R, в противном случае устройство ЧПУ выводит сообщение об ошибке 2053 Нет упорной точки или точки R.

Позиция отверстия: X_p, Y_p, Z_p, q

Позицией отверстия берутся из записанных значений координат - не считая оси сверления - все остальные данные координат. Эти могут быть **главные оси выбранной плоскости, параллельные иму оси**, или **прочие произвольные оси, не выделённые для сверления (q: например C)**.

Записанные значения могут быть инкрементными, или абсолютными, значениями, заданными в прямоугольных или полярных координатах, по размерности метрическими или дюймовыми.

Для записанных значений координат действительны команды отражения, вращения, и масштабирования.

Устройство ЧПУ наводится в позицию отверстия с позиционированием быстрого хода, независимо от того, что из 1-й группы какой код был действителен.

Перемещение после ориентировки шпинделя: I, J, K

Если на данном станке имеется возможность для ориентировки шпинделя, в циклах растачивания G76, и G87 отвод инструмента можно выполнить удаляя от поверхности, чтобы вершина инструмента не поцарапала её. При этом можно задавать по адресу I, J и K, что устройство ЧПУ в какое направление должно удалить инструмент от поверхности. Устройство ЧПУ истолкует адреса, согласно выбранной плоскости:

G17: I, J

G18: K, I

G19: J, K

Адреса истолкуются **всегда инкрементными, прямоугольными** данными. Адрес может быть метрическим и дюймовым.

Для данных I, J, K не действительны команды отражения, вращения, и масштабирования. I, J и K являются модальными значениями. Удаляют значения кода G80, или кодов группы интерполяции. Отвод выполняется быстрым ходом.

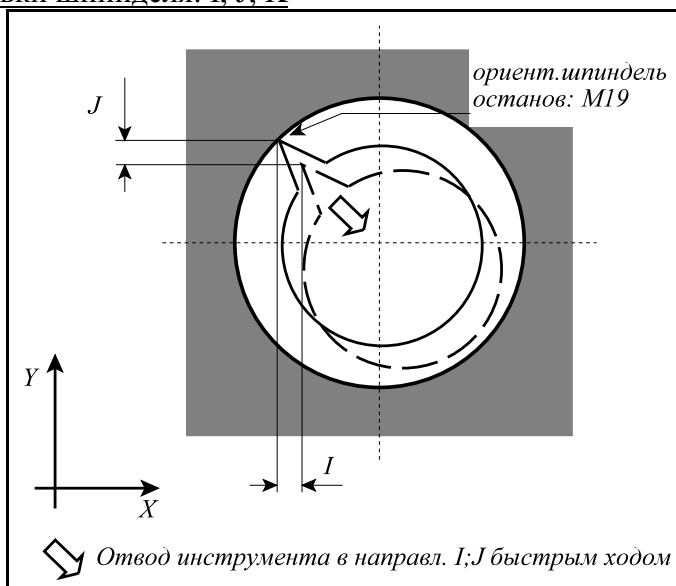


рисунок 19-10

Данные сверления:Упорная точка отверстия: X_{p2} , Y_{p2} , Z_p

Упорную точку отверстия надо задавать по адресу оси сверления. Координаты упорной точки отверстия истолкуются всегда прямоугольными данными. Могут быть дюймовыми, или метрическими, абсолютными, или инкрементными. Если задавать значение упорной точки инкрементно, перемещение считается от точки R.

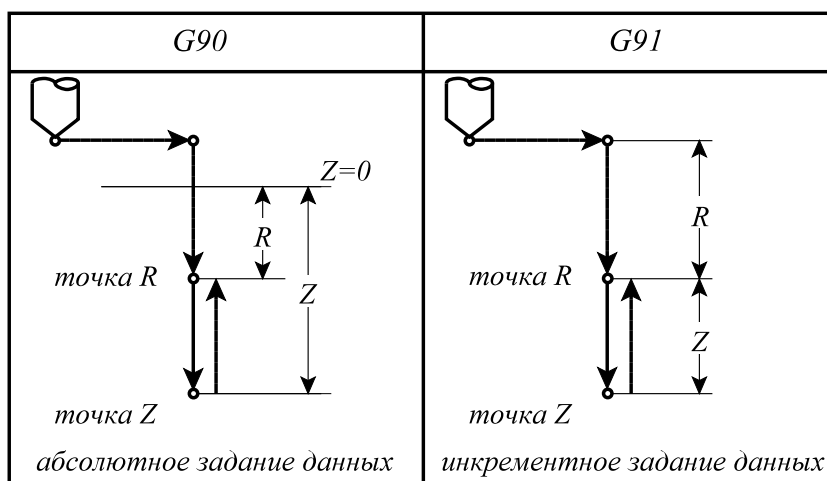


рисунок 19-11

Для данных упорной точки действительны команды отражения и масштабирования. Данные упорной точки передаются модально. Удаляют значения кода G80, или кодов

группы интерполяции. Устройство ЧПУ приближается к упорной точке всегда действующей подачей.

Точка приближения, точка R: R

Точка приближения задаётся по адресу R. Адрес R является всегда прямоугольными данными, которые могут быть инкрементными, абсолютными, метрическими, или дюймовыми. Если данные R инкрементные, их значение отсчитывается от исходной точки. Для данных точки R действительны данные отражения и масштабирования. Данные точки R передаются модально. Удаляют значения кода G80, или кодов группы интерполяции. Устройство ЧПУ приближается к точке R всегда движением быстрого хода.

Значение глубины резания: Q

Значение глубины резания покажет в циклах G83.1 и G83. Они являются всегда инкрементными, прямоугольными, положительными данными. Значение глубины резания передаются модально. Удаляют значения кода G80, или кодов группы интерполяции. Для глубины резания не действительна команда масштабирования.

Вспомогательные данные: E

В цикле G83.1 покажет меру отвода, далее в цикле G83 перед врезанием до этого значения совершается быстрый ход. Они являются всегда инкрементными, прямоугольными, положительными данными. Для вспомогательных данных не действительна команда масштабирования. Значение вспомогательных данных передаются модально. Удаляют значения кода G80, или кодов группы интерполяции. Если они не запрограммированы, тогда устройство ЧПУ берёт необходимые данные с параметра N1500 Return Val G73, а также N1501 Clearance Val G83.

Задержка: P

Это задаёт время задержки на основании отверстия. Для его задания действительны правила, изложенные у G04. Значение задержки передаётся модально. Удаляют значения кода G80, или кодов группы интерполяции.

Подача: F

Им определяется подача. Его значение передаётся модально. Изменить его можно только программированием других данных F, G80 или другой код не удаляет его.

Override отвода: I (%)

В циклах G85, G89, G84.2, G84.3 отвод выполняется в основном запрограммированной подачей по адресу F. По адресу I можно задавать процентное значение override отвода. Если оно не запрограммировано, значение override берётся с параметра.

Число оборотов шпинделя: S

Его значение передаётся модально. Его можно изменить только программированием других данных S, G80 или другой код не удаляет его.

Число повторения: L

Им определяется число повторения циклов. Предел его значения: 1–99999999. Если не заполнен L, берётся значение L=1. В случае L=0 данные цикла сохраняются, но не выполняются. Значение L действует только в том кадре, где его задавали.

Пример для передачи кодов сверления и переменных циклов модально:

```
N1 G17 G0 Z_ M3
N2 G81 X_ C_ Z_ R_ F_
```

В начале режима цикла задание данных сверления (Z, R) обязательно.

```
N3 X_
```

Поскольку в кадре N2 данные сверления были определены и в кадре N3 нужны те же самые, их заполнение лишнее, то есть можно этим пренебречь G81, Z_, R_, F_. Позиция отверстия изменяется только в направлении X, сверло перемещается в этом направлении, затем просверлит то же отверстие, как в кадре N2.

```
N4 G82 C_ Z_ P_
```

Позиция отверстия перемещается в направлении C. Способ сверления получается согласно G82, точка основания Z примет новое значение, точка приближения и подача (R, F) передаются модально из кадра N2.

```
N5 G80 M5
```

Режим цикла и модальные переменные удаляются, кроме F.

```
N6 G85 C_ Z_ R_ P_ M3
```

Поскольку в кадре N5 удаляются данные сверления под действием команды G80, значения Z, R, и P надо снова задавать.

```
N7 G0 X_ C_
```

Режим цикла и модальные переменные удаляются, кроме F.

Примеры для использования повторения цикла:

Если приходится изготовить такие же отверстия на равные расстояния такими же параметрами, число повторения задаётся по адресу L. L действует только в том кадре, в котором его задавали.

```
N1 G90 G19 G0 X300 Z40 C0 M3
N2 G91 G81 X-40 Z100 R-20 F50 L5
```

Под действием команд выше устройство ЧПУ вдоль оси Z просверлит на расстояние 100 мм друг от друга 5 шт. одинаковых отверстий. Позиция первого отверстия Z=140, C=0. Поскольку речь идёт о боковом сверлении (сверлим с осью X) выбрали плоскость G19.

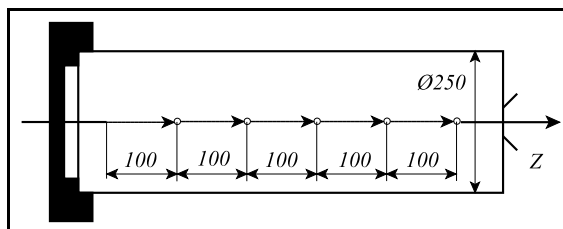


рисунок 19-12

Позиция отверстия была задана под действием G91 инкрементно.

```
N1 G90 G17 G0 X200 C-60 Z50
```

```
N2 G81 CI60 Z-40 R3 F50 L6
```

Под действием команд выше устройство ЧПУ просверлит 6 шт. по 60 градусов отверстий в окружности центров с диаметром 100 мм. Позиция первого отверстия попадает в точку с координатами $X=200$ $C=0$. Поскольку речь идёт о торцевом сверлении (сверлим с осью Z) выбрали плоскость G17.

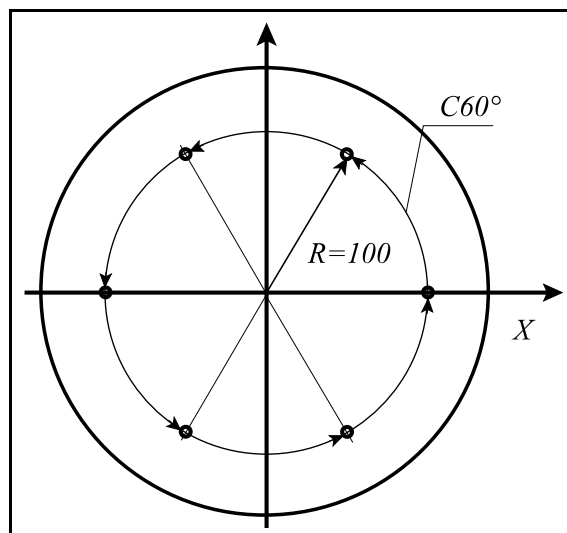


рисунок 19-13

19.1 Подробное описание циклов сверления.

19.1.1 Цикл высокоскоростного глубокого сверления (G83.1)

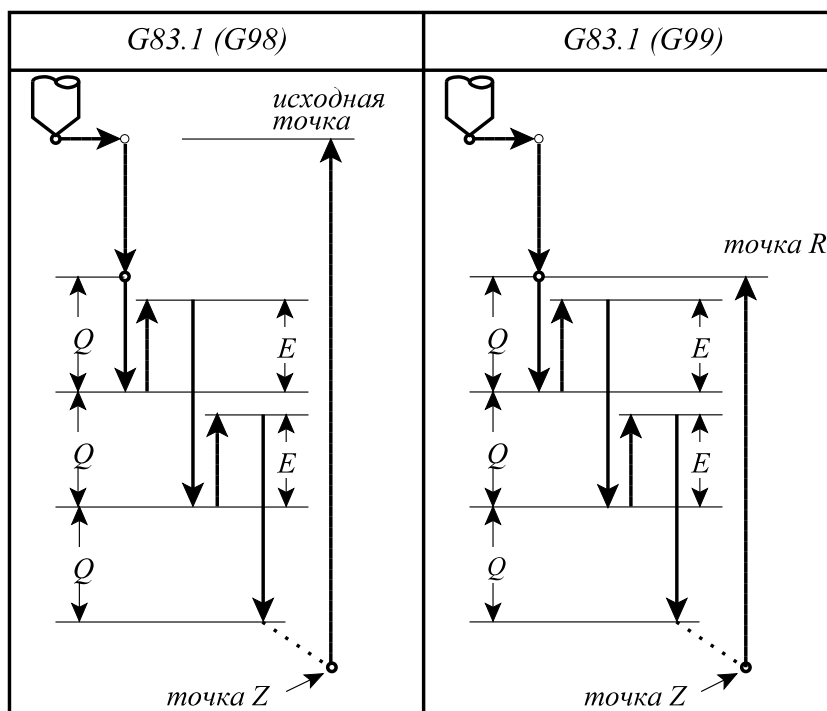


рисунок 19.1.1-1

Переменные, использованные в цикле:

G17 **G83.1** X_p Y_p Z_p R Q E F L

G18 **G83.1** Z_p X_p Y_p R Q E F L

G19 **G83.1** Y_p Z_p X_p R Q E F L

Операции цикла:

- | | |
|----------------|---|
| 1-я операция: | позиционирование в выбранной плоскости быстрым ходом |
| 2-я операция: | — |
| 3-я операция: | движение с быстрым ходом до точки R (точка приближения) |
| 4-я операция: | — |
| 5-я операция: | сверление до упорной точки подачей F |
| 6-я операция: | — |
| 7-я операция: | в случае G99: отвод до точки R быстрым ходом |
| 8-я операция: | — |
| 9-я операция: | в случае G98: отвод до точки исхода быстрым ходом |
| 10-я операция: | — |

Описание 5-й операции сверления:

- заданную по адресу **Q** *глубину резания* подачей просверлит в заготовку,
 - с заданным по адресу **E**, или параметром N1500 Return Val G73 значением быстрым ходом совершает *отвод*,
 - повторно просверлит до глубины Q, отсчитывая от упорной точки предыдущего сверления
 - с заданным по адресу **E** значением быстрым ходом совершает отвод.
- Процесс продолжается до заданной по адресу **Z** упорной точки.

19.1.2 Сверление левой резьбы с уравнильной насадкой (G84.1)

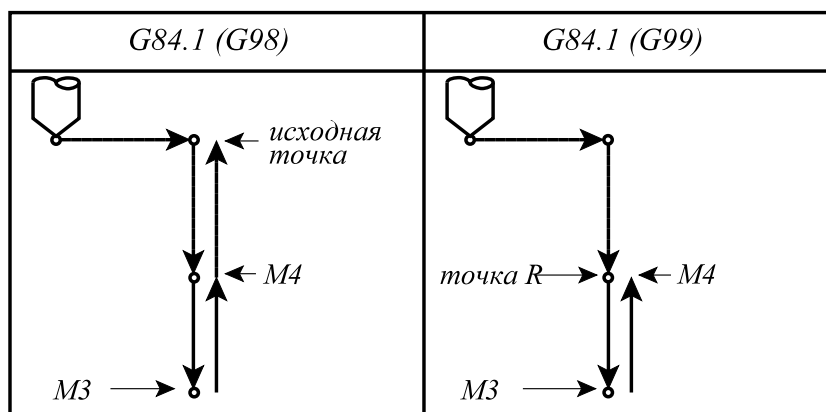


рисунок 19.1.2-1

Цикл применяется только с метчиком, снабженным с уравнильной насадкой.

Переменные, использованные в цикле:

G17 **G84.1** X_p Y_p Z_p R P F L

G18 **G84.1** Z_p X_p Y_p R P F L

G19 **G84.1** Y_p Z_p X_p R P F L

Перед запуском цикла надо включить и далее запрограммировать направление вращения (против хода часовой стрелки) шпинделя M4.

Значение подачи надо задавать в зависимости от подъёма нарезки метчика:

– в состоянии G94 по минутной подачи:

$$F=P \times S$$

где: P: подъём нарезки в размерности мм/об, или дюйм/об

S: число оборотов шпинделя в размерности об/мин

– в состоянии G95 подачи за оборот:

$$F=P$$

где: P: подъём нарезки в размерности мм/об, или дюйм/об

Операции цикла:

- | | |
|----------------|--|
| 1-я операция: | позиционирование в выбранной плоскости быстрым ходом |
| 2-я операция: | – |
| 3-я операция: | движение с быстрым ходом до точки R (точка приближения) |
| 4-я операция: | – |
| 5-я операция: | сверление до упорной точки подачей F, override u stop запрещён |
| 6-я операция: | – задержка согласно значению, заданному по адресу P
– смена направления вращения шпинделя: M3 |
| 7-я операция: | отвод до точки R подачей F, override u stop запрещён |
| 8-я операция: | смена направления вращения шпинделя: M4 |
| 9-я операция: | в случае G98: отвод до исходной точки быстрым ходом |
| 10-я операция: | – |

19.1.3 Обтачивание с автоматическим отводом инструмента (G86.1)

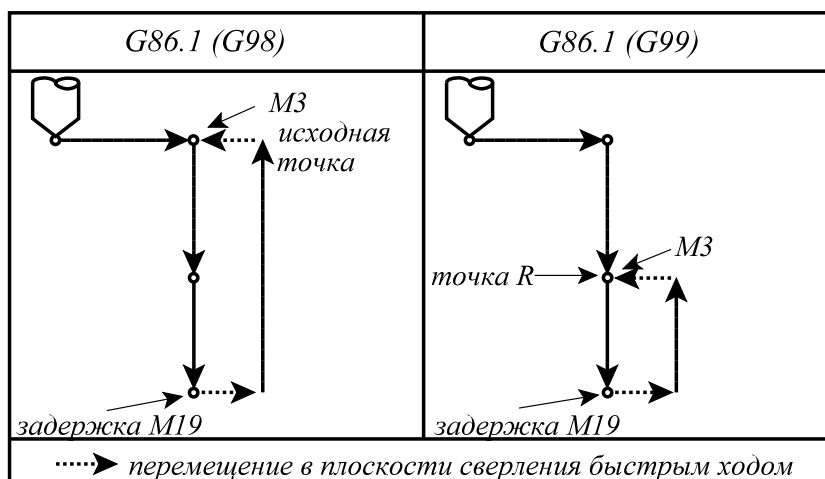


рисунок 19.1.3-1

Циклом G86.1 можно пользоваться только тогда, если построена ориентировка шпинделя в инструментальный станок. Об этом факте сообщает устройству ЧПУ состояние 1 бита ORI параметра N0607 Spindle Config. В противном случае выводится сообщение об ошибке 2137 G76, G87.

Поскольку цикл после обтачивания выполняет ориентировку шпинделя и отводит инструмент от поверхности согласно значению, заданному по I, J и K, при отводе инструмента не царапается поверхность.

Переменные, использованные в цикле:

G17 **G86.1** X_p Y_p I J Z_p R P F L
 G18 **G86.1** Z_p X_p K I Y_p R P F L
 G19 **G86.1** Y_p Z_p J K X_p R P F L

Перед запуском цикла надо выдавать команду M3.

Операции цикла:

- | | |
|----------------|---|
| 1-я операция: | позиционирование в выбранной плоскости быстрым ходом |
| 2-я операция: | — |
| 3-я операция: | движение с быстрым ходом до точки R (точка приближения) |
| 4-я операция: | — |
| 5-я операция: | обтачивание до упорной точки F подачей |
| 6-я операция: | — задержка согласно значению, заданному по адресу P
— ориентировка шпинделя : M19
— отвод инструмента в выбранной плоскости согласно значению I, J, K быстрым ходом |
| 7-я операция: | в случае G99: отвод инструмента до точки R, быстрым ходом |
| 8-я операция: | в случае G99
— отвод инструмента в выбранной плоскости согласно значению I, J, K, противоположно, быстрым ходом,
— перезапуск шпинделя в направлении M3 |
| 9-я операция: | в случае G98: отвод инструмента быстрым ходом до исходной точки |
| 10-я операция: | в случае G98 |

- отвод инструмента в выбранной плоскости согласно значению I, J, K, противоположно, быстрым ходом,
- перезапуск шпинделя в направлении M3

19.1.4 Выключение циклического состояния (G80)

Под действием кода выключается циклическое состояние, переменные цикла удаляются. Z и R примет инкрементное значение 0, остальные переменные 0.

Если в кадр **G80** запрограммировать координаты, и не задавать другую команду, тогда движение выполняется на основании действующего до включения цикла кода интерполяции (1-я группа кодов G, или группа интерполяции).

19.1.5 Цикл сверления, отвод быстрым ходом (G81)

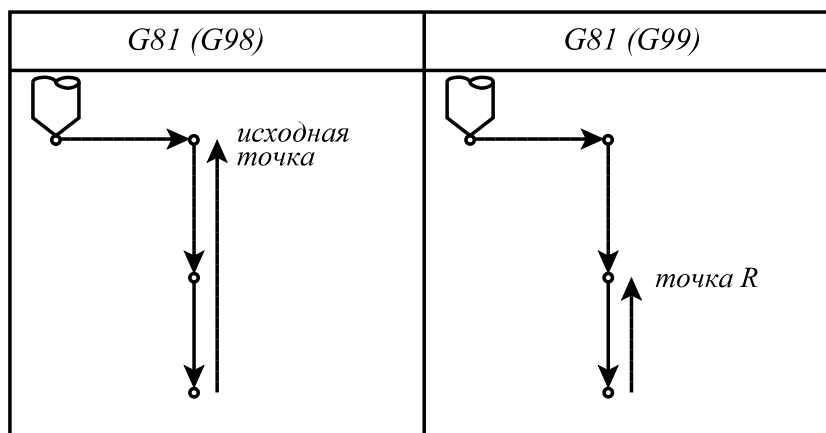


рисунок [19.1.5-1](#)

Переменные, использованные в цикле:

G17 **G81** X_p___ Y_p___ Z_p___ R___ F___ L___

G18 **G81** Z_p___ X_p___ Y_p___ R___ F___ L___

G19 **G81** Y_p___ Z_p___ X_p___ R___ F___ L___

Операции цикла:

- | | |
|----------------|---|
| 1-я операция: | позиционирование в выбранной плоскости быстрым ходом |
| 2-я операция: | — |
| 3-я операция: | движение быстрым ходом до точки R (точка приближения) |
| 4-я операция: | — |
| 5-я операция: | сверление до упорной точки с подачей F |
| 6-я операция: | — |
| 7-я операция: | в случае G99: отвод до точки R, быстрым ходом |
| 8-я операция: | — |
| 9-я операция: | в случае G98: отвод быстрым ходом до исходной точки |
| 10-я операция: | — |

19.1.6 Цикл сверления с задержкой, отвод быстрым ходом (G82)

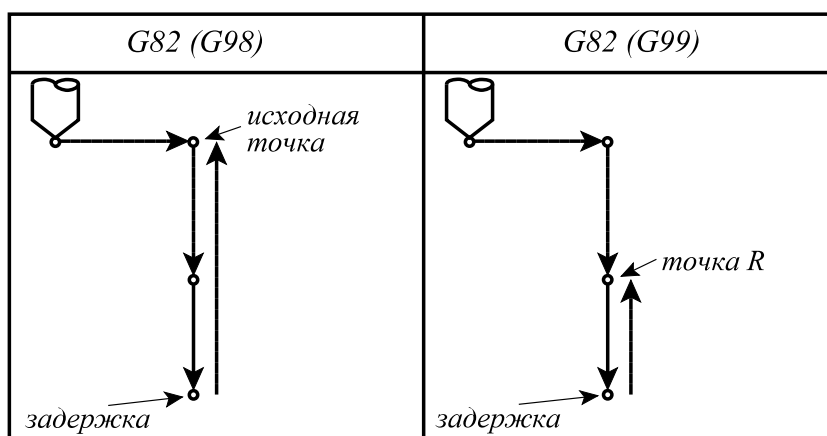


рисунок 19.1.6-1

Переменные, использованные в цикле:

G17 **G82** X_p Y_p Z_p R P F L

G18 **G82** Z_p X_p Y_p R P F L

G19 **G82** Y_p Z_p X_p R P F L

Операции цикла:

- 1-я операция: позиционирование в выбранной плоскости быстрым ходом
- 2-я операция: —
- 3-я операция: движение быстрым ходом до точки R (точка приближения)
- 4-я операция: —
- 5-я операция: сверление до упорной точки подачей F
- 6-я операция: задержка до времени, заданного по адресу P
- 7-я операция: в случае G99: отвод до точки R, быстрым ходом
- 8-я операция: —
- 9-я операция: в случае G98: отвод быстрым ходом до исходной точки
- 10-я операция: —

19.1.7 Цикл глубокого сверления (G83)

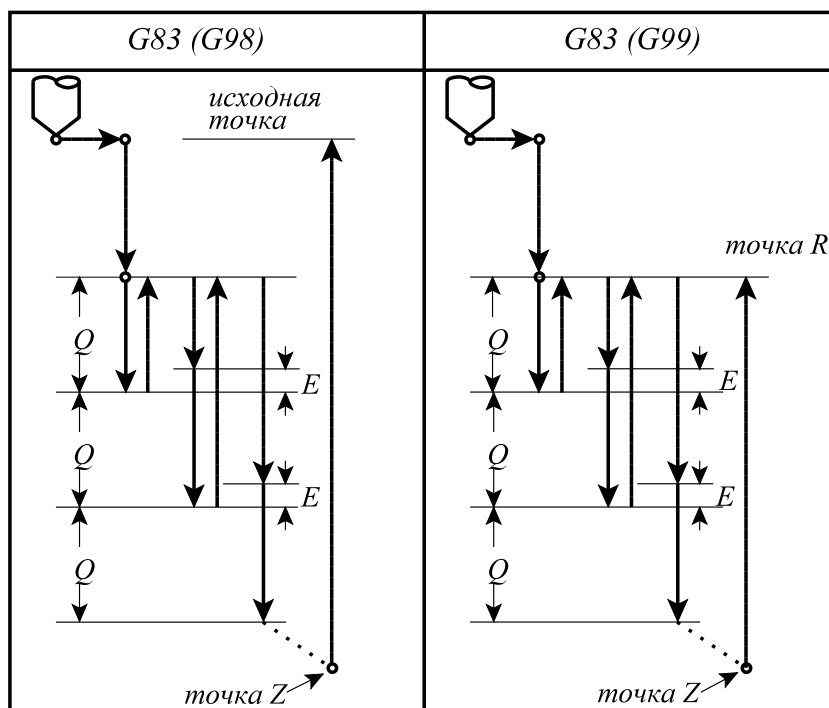


рисунок 19.1.7-1

Переменные, использованные в цикле:

G17 **G83** X_p Y_p Z_p R Q E F L

G18 **G83** Z_p X_p Y_p R Q E F L

G19 **G83** Y_p Z_p X_p R Q E F L

Операции цикла:

- | | |
|----------------|---|
| 1-я операция: | позиционирование в выбранной плоскости быстрым ходом |
| 2-я операция: | — |
| 3-я операция: | движение с быстрым ходом до точки R (точка приближения) |
| 4-я операция: | — |
| 5-я операция: | сверление до упорной точки подачей F |
| 6-я операция: | — |
| 7-я операция: | в случае G99: отвод до точки R, быстрым ходом |
| 8-я операция: | — |
| 9-я операция: | в случае G98: отвод быстрым ходом до исходной точки |
| 10-я операция: | — |

Описание 5-й операции сверления:

- заданную по адресу **Q** *глубину резания* подачей просверлит в заготовку,
- быстрым ходом отвод до точки R,
- быстрым ходом *приближается* до предыдущей глубины *до расстояния “E”*,
- повторно просверлит до глубины Q, отсчитывая от упорной точки предыдущего сверления, подачей F (перемещение E+Q)
- быстрым ходом отвод до точки R

Процесс продолжается до упорной точки, заданной по адресу Z.

Расстояние “E” берётся или из программы из адреса E, или с параметра N1501 Clearance Val G83.

19.1.8 Сверление правой резьбы с уравнильной насадкой (G84)

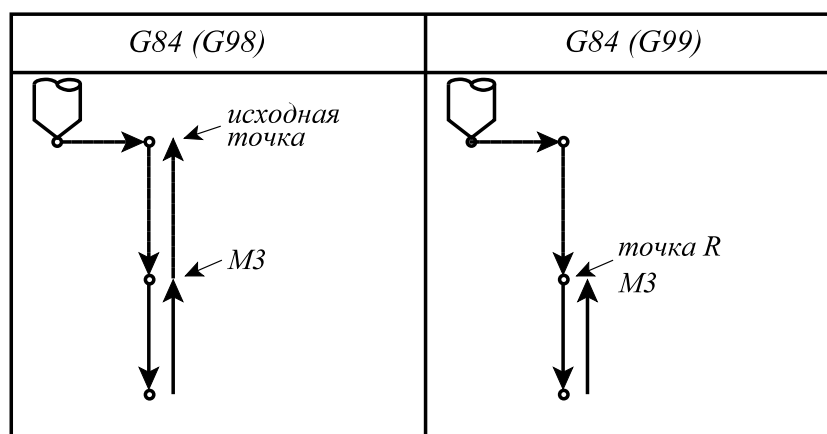


рисунок 19.1.8-1

Цикл применяется только с метчиком, снабжённым с уравнильной насадкой.

Переменные, использованные в цикле:

G17 **G84** X_p Y_p Z_p R P F L

G18 **G84** Z_p X_p Y_p R P F L

G19 **G84** Y_p Z_p X_p R P F L

Перед запуском цикла надо включить направление вращения шпинделя M3 (согласно ходу часовой стрелки).

Значение подачи надо задавать в зависимости от подъёма нарезки метчика:

– в состоянии G94 по минутной подачи:

$$F = P \times S$$

где: P: подъём нарезки в размерности мм/б, или дюйм/об

S: число оборотов шпинделя в размерности об/мин

– в состоянии G95 подачи за оборот:

$$F = P$$

где: P: подъём нарезки в размерности мм/б, или дюйм/об

Операции цикла:

- | | |
|----------------|--|
| 1-я операция: | позиционирование в выбранной плоскости быстрым ходом |
| 2-я операция: | – |
| 3-я операция: | движение с быстрым ходом до точки R (точка приближения) |
| 4-я операция: | – |
| 5-я операция: | сверление до упорной точки подачей F, override u stop запрещён |
| 6-я операция: | – задержка согласно значению, заданному по адресу P
– смена направления вращения шпинделя: M4 |
| 7-я операция: | отвод до точки R подачей F, override u stop запрещён |
| 8-я операция: | смена направления вращения шпинделя: M3 |
| 9-я операция: | в случае G98: отвод быстрым ходом до исходной точки |
| 10-я операция: | – |

19.1.9 Нарезание резьбы без уравнительной насадки (G84.2, G84.3)

Циклы нарезания резьбы без уравнительной насадки можно применить только на таких станках, где шпиндель оборудован датчиком позиции, и привод шпинделя регулируется обратной связью. (Бит INX будет 1 параметра N0607 Spindle Config). В противном случае выводится сообщение об ошибке 2138 Шпиндель нельзя индексировать.

В случае нарезания резьбы метчиком частное подачи сверлильной оси и оборотов шпинделя должно равняться подъёму нарезки метчика. Иначе говоря, при нарезании резьбы метчиком в идеальном случае частное ниже должно быть с минуты на минуту постоянным: $F=P/S$

где: P: подъём нарезки (мм/об, или дюйм/об),
F: подача (мм/мин, или дюйм/мин),
S: число оборотов шпинделя (об/мин).

В цикле сверления левой резьбы G74 и правой резьбы G84 число оборотов шпинделя и подача сверлильной оси управляется совершенно независимо друг от друга. Условие, описанное выше соответственно не может точно удовлетвориться. Это особенно верно на дне отверстия, где подача сверлильной оси и число оборотов шпинделя должны замедлиться и остановиться синхронно друг с другом, затем в противоположном направлении ускориться. В случае выше это условие совсем не соблюдается по технике управления. Проблему выше можно избежать так, что метчик вставляется в шпиндель с подпружиненной уравнительной насадкой, которая выравнивает возникшее колебание по значению частного F/S .

Принцип управления отличается от этого у циклов сверления G84.2, G84.3, позволяющих устранения уравнительной насадки. У этих устройств ЧПУ непрерывно заботится о том, что частное F/S было с минуты на минуту постоянным.

По технике управления в первом случае устройство ЧПУ регулирует только число оборотов шпинделя, а в последнем случае - и его позицию. В циклах G84.2, G84.3 движение сверлильной оси и шпинделя соединяется линейной интерполяцией. Этим способом можно обеспечить постоянство частного F/S и в участках ускорения и замедления.

G84.2: сверления правой резьбы без уравнительной насадки

G84.3: сверления левой резьбы без уравнительной насадки

Переменные, использованные в цикле:

G17 **G84.** X_p Y_p Z_p R P F I S L

G18 **G84.** Z_p X_p Y_p R P F I S L

G19 **G84.** Y_p Z_p X_p R P F I S L

В конце цикла шпиндель остаётся в индексированном состоянии (замкнутый контур регулирования позиции), в случае необходимости программист должен позаботиться о перезагрузке.

Значение подачи надо задавать в зависимости от подъёма нарезки метчика:

– состояние G94 поминутной подачи:

$$F=P \times S$$

где: P: подъём нарезки в размерности мм/об, или дюйм/об

S: число оборотов шпинделя в размерности об/мин
 – состояние G95 подачи за оборот:

$$F=P$$

где: P: подъём нарезки в размерности мм/об, или дюйм/об

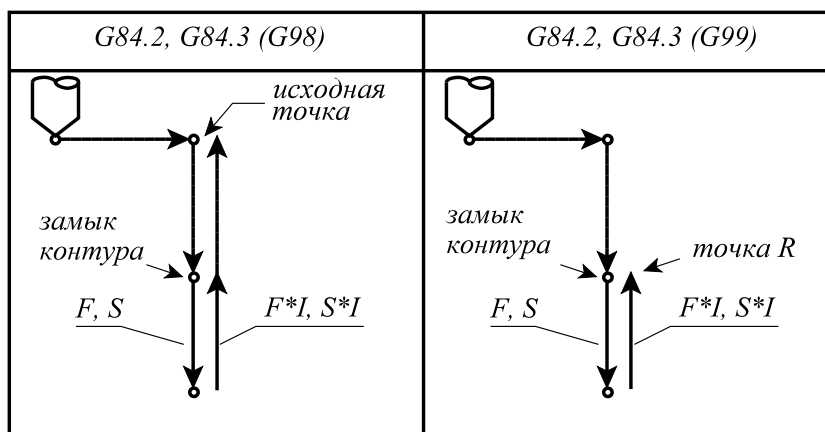


рисунок 19.1.9-1

Операции цикла в случае G84.2:

- 1-я операция: позиционирование в выбранной плоскости быстрым ходом
- 2-я операция: -
- 3-я операция: движение быстрым ходом до точки R (точка приближения)
- 4-я операция: если бит TSC параметра N1503 Drilling Cycles Config.:
 - =0: не ориентирует шпиндель, только замыкает контур регулирования позиции. Код замыкания контура определяется параметром N0823 M Code for Closing S Loop. Этот код передаётся для PLC. (Выполнение быстрее)
 - =1: до начала нарезания резьбы метчиком ориентирует шпиндель и передаст команду M19 для PLC. (Попадает обратно в резьбу)
- 5-я операция: линейная интерпляция между сверлильной осью и шпинделем в направлении вращения шпинделя (+) по ходу часовой стрелки в случае G84.2, далее в направлении вращения шпинделя против часовой стрелки в случае G84.3
- 6-я операция: задержка согласно значению, заданному по адресу P
- 7-я операция: линейная интерпляция между сверлильной осью и шпинделем в направлении вращения шпинделя (-) противоположно ходу часовой стрелки в случае G84.2, далее в направлении вращения шпинделя по часовой стрелки в случае G84.3.

Override отвода I (%)

Override подачи отвода - хоть запрограммирован по адресу I, хоть надо брать с параметра - действует только тогда, если разрешили его назначением 1 бита #4 EOE параметра N1503 Drilling Cycles Config. Если в кадре не задавать значение адресу I, тогда в 7-й операции override подачи отвода будет иметь значение, заданное в %-ах параметра N1506 Extraction Override in Tapping На рассчитанную таким образом скорость действует ещё и включатель процентного значения подачи, то есть подача получит значение

$$F_{отвод} = F_{запрограммированная} * Подача\Override * Extraction\Override\ in\ Tapping / 100$$

Ускорение при отводе

В ходе отвода можно задавать отличающееся значение ускорения (меньшее), чем установлено параметром N1507+n Rn Acc in Tapping для шпинделя, если назначением 1 бита #6 EAE параметра N1503 Drilling Cycles Config. разрешили это. При этом различное по диапазонам значение ускорения берётся с параметра N1523+n Rn Acc in Extract (n=1...8).

- 8-я операция: -
 9-я операция: в случае G98: отвод быстым ходом до исходной точки
 10-я операция: -

19.1.10 Нарезание резьбы без уравнильной насадки, со стружколомом (G84.2, G84.3)

Код цикла:

G84.2 Q__: сверления правой резьбы без уравнильной насадки, со стружколомом

G84.3 Q__: сверления левой резьбы без уравнильной насадки, со стружколомом

В цикле устройство ЧПУ *применяет стружколом тогда*, если запрограммирована *глубина резания по адресу Q*.

Переменные, использованные в цикле:

G17 **G84.** X_p Y_p Z_p R Q E P F S I L

G18 **G84.** Z_p X_p Y_p R Q E P F S I L

G19 **G84.** Y_p Z_p X_p R Q E P F S I L

В конце цикла шпиндель остаётся в индексированном состоянии (замкнутый контур регулирования позиции), в случае необходимости программист должен позаботиться о перезагрузке. Значение кода **G98, G99** то же самое, как в случае всех остальных циклов сверления.

Битом #3 PTC параметра N1503 Drilling Cycles Config. решается, что какой будет *способ ломки стружки*. Если бит PTC:

=0: (согласно G83.1) применяется быстрая ломка стружки.

=1: (согласно G83) применяется нормальная ломка стружки, и отвод совершается до точки R.

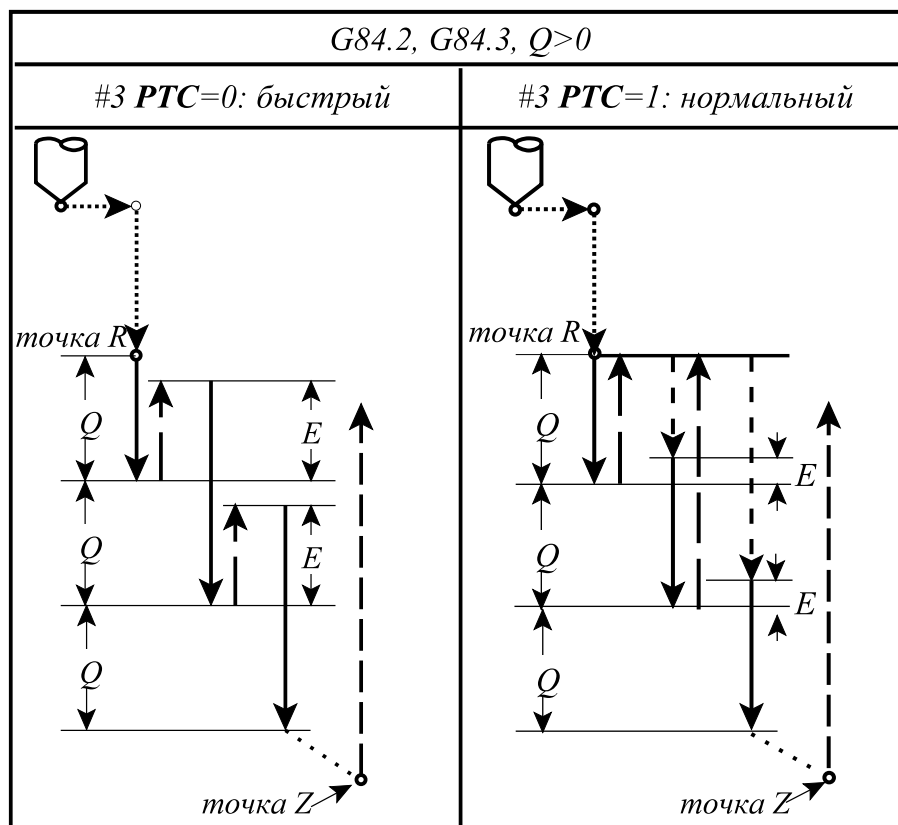


рисунок 19.1.10-1

Истолкование адреса E

В случае **быстрой ломки стружки** ($PTC=0$) **по адресу E** можно задавать **меру отвода инструмента**. Если не заполнен адрес E, устройство ЧПУ берёт расстояние с параметра N1504 Return Val in Tapping.

В случае **нормальной ломки стружки** ($PTC=1$) **по адресу E** можно задавать после отвода **расстояние приближения** после отвода. Если не заполнен адрес E, устройство ЧПУ берёт расстояние с параметра N1505 Clearance Val in Tapping.

Override отвода I (%)

Override подачи отвода - хоть запрограммирован по адресу I, хоть надо брать с параметра - во всех участках действует только тогда, если разрешили его назначением 1 бита #4 EOE параметра N1503 Drilling Cycles Config.

Если в кадре не задавать значение адресу I, тогда override подачи отвода будет заданное в %-ах значения параметра N1506 Extraction Override in Tapping. На рассчитанную таким образом скорость действует ещё и включатель процентного значения подачи, то есть подача получит значение

Фотвод = Fзапрограммированная * Подача Override * Extraction Override in Tapping/100

Ускорение при отводе

В ходе отвода можно задавать и значение ускорения, отличающееся (меньше) от установленного для шпинделя параметром N1507+n Rn Acc in Tapping, если это разрешили назначением 1 бита #6 EAE параметра N1503 Drilling Cycles Config.. При этом различное по диапазонам значение ускорения берётся с параметра N1523+n Rn Acc in Extract ($n=1...8$).

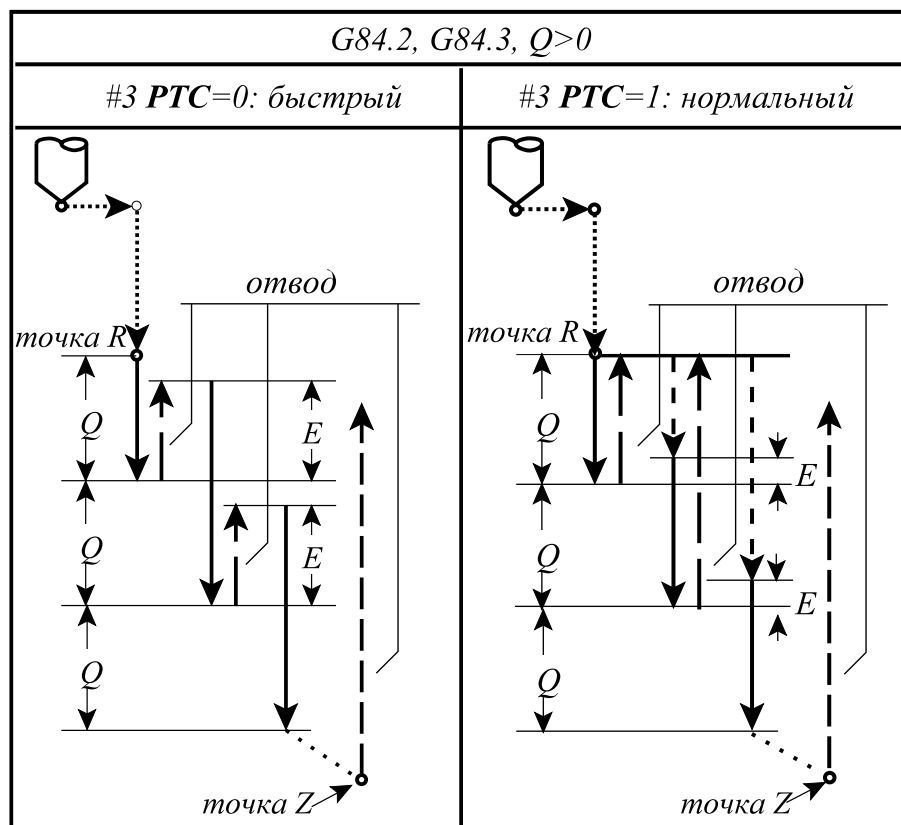


рисунок 19.1.10-2

Override отвода (%) при нормальной ломки стружки

В случае применения **нормальной ломки стружки** (параметр N1503 Drilling Cycles Config. #3 PTC бит 1) в ходе **отвода**, можно применить для запрограммированной F отдельный, отличающийся от 100% **override**, если это разрешить назначением 1 бита #5 ROE параметра N1503 Drilling Cycles Config.

При этом значение override берётся от заданного в %-ах значения параметра N1507 Return Override in Tapping.

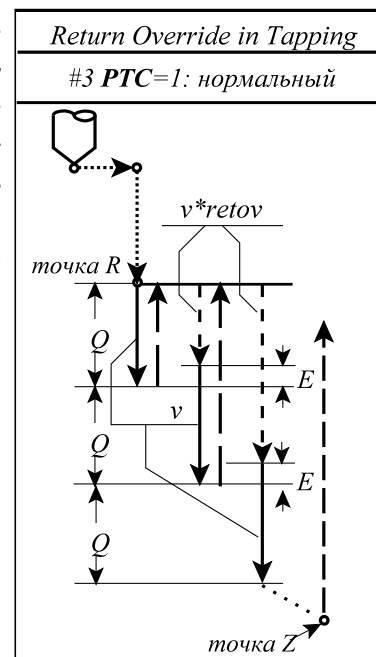


рисунок 19.1.10-3

Для дальнейших вводных параметров цикла действительны изложенные в предыдущей главе.

19.1.11 Цикл сверления, отвод подачей (G85)

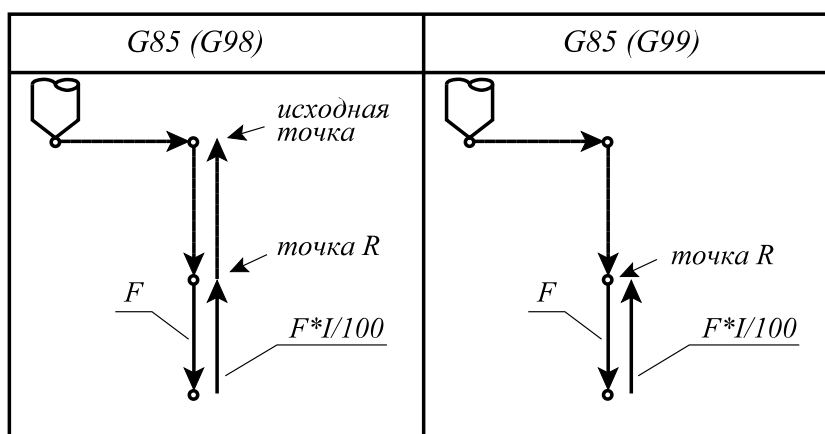


рисунок 19.1.11-1

Переменные, использованные в цикле:

G17 **G85** X_p Y_p Z_p R F I L

G18 **G85** Z_p X_p Y_p R F I L

G19 **G85** Y_p Z_p X_p R F I L

Операции цикла:

- 1-я операция: позиционирование в выбранной плоскости быстрым ходом
- 2-я операция: —
- 3-я операция: движение быстрым ходом до точки R (точка приближения)
- 4-я операция: —
- 5-я операция: сверление до упорной точки подачей F
- 6-я операция: —
- 7-я операция: отвод до точки R, подачей $F \cdot I / 100$

Override отвода I (%)

Если в кадре не задавать значение адресу I, тогда в 7-й операции значение override подачи отвода будет параметр N1502 Extraction Override in G85, G89. На рассчитанную таким образом скорость действует ещё и включатель процентного значения подачи, то есть подача получит значение

Фотвод = Fзапрограммированная * Подача Override * Extraction Override in Tapping/100

- 8-я операция: —
- 9-я операция: в случае G98: отвод быстрым ходом до исходной точки
- 10-я операция: —

19.1.12 Цикл сверления, отвод быстрым ходом при остановленном шпинделе (G86)

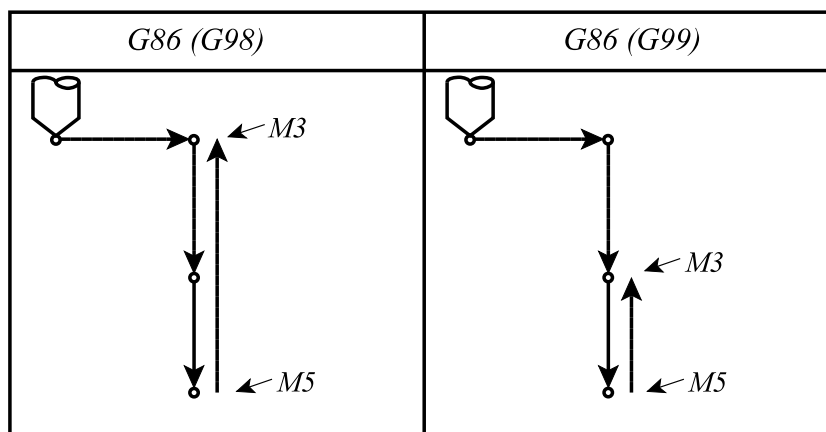


рисунок 19.1.12-1

Переменные, использованные в цикле:

G17 **G86** X_p__ Y_p__ Z_p__ R__ F__ L__

G18 **G86** Z_p__ X_p__ Y_p__ R__ F__ L__

G19 **G86** Y_p__ Z_p__ X_p__ R__ F__ L__

При запуске цикла для шпинделя надо дать направление вращения M3.

Операции цикла:

- | | |
|----------------|---|
| 1-я операция: | позиционирование в выбранной плоскости быстрым ходом |
| 2-я операция: | — |
| 3-я операция: | движение быстрым ходом до точки R (точка приближения) |
| 4-я операция: | — |
| 5-я операция: | сверление до упорной точки подачей F |
| 6-я операция: | останов шпинделя: M5 |
| 7-я операция: | в случае G99: отвод до точки R, g быстрым ходом |
| 8-я операция: | в случае G99: перезапуск шпинделя: M3 |
| 9-я операция: | в случае G98: отвод быстрым ходом до исходной точки |
| 10-я операция: | в случае G98: перезапуск шпинделя: M3 |

19.1.13 Ручное воздействие в упорной точке/ Обратная расточка (G87)

Цикл может быть выполнен устройством ЧПУ двояким образом:

А. Цикл сверления, ручное воздействие в упорной точке

В том случае, если *на станке не выстроено возможность ориентировки шпинделя*, то есть бит #1 ORI равно 0 параметра N0607 Spindle Config, устройство ЧПУ поступает по случаю А.

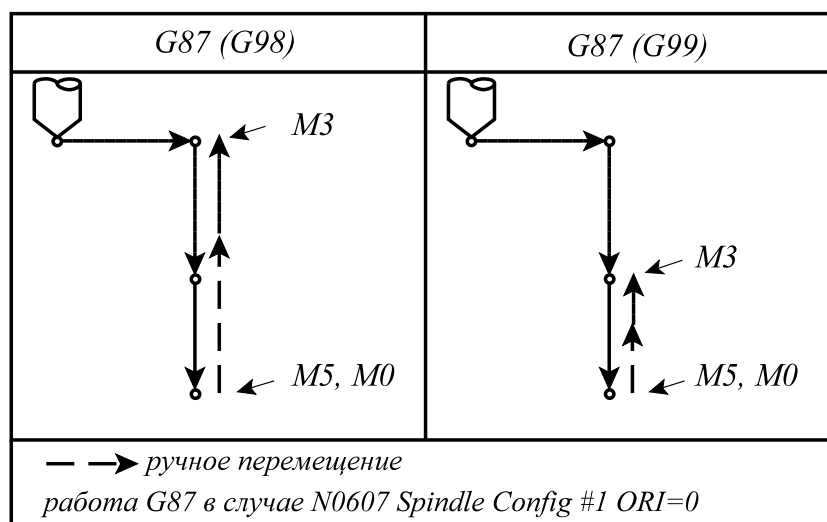


рисунок 19.1.13-1

Переменные, использованные в цикле:

G17 **G87** X_p Y_p Z_p R F L

G18 **G87** Z_p X_p Y_p R F L

G19 **G87** Y_p Z_p X_p R F L

При запуске цикла шпинделю надо дать направление вращения M3.

Операции цикла:

- | | |
|---------------|--|
| 1-я операция: | позиционирование в выбранной плоскости быстрым ходом |
| 2-я операция: | — |
| 3-я операция: | движение быстрым ходом до точки R (точка приближения) |
| 4-я операция: | — |
| 5-я операция: | сверление до упорной точки подачей F |
| 6-я операция: | — останов шпинделя: M5 |
| | — устройство ЧПУ примет состояние STOP (M0), откуда оператор выходит в один из ручных режимов, действующий станок (ПЕРЕМЕЩЕНИЕ, СДВИГ, МАХОВИЧОК), то есть может отодвинуть вершину инструмента от поверхности отверстия, и может отводит инструмент из отверстия. После этого вступив в режим АВТОМАТ, нажав на клавишу старт продолжается обработка. |
| 7-я операция: | в случае G99: после START отвод до точки R, быстрым ходом |
| 8-я операция: | в случае G99: перезапуск шпинделя: M3 |

- 9-я операция: в случае G98: после START отвод быстрым ходом до исходной точки
- 10-я операция: в случае G98: перезапуск шпинделя: M3

В. Обратная расточка, с автоматическим отводом инструмента

В том случае, если **на станке выстроена возможность ориентировки шпинделя**, то есть бит #1 ORI равно 1 параметра N0607 Spindle Config, устройство ЧПУ поступит по случаю В.

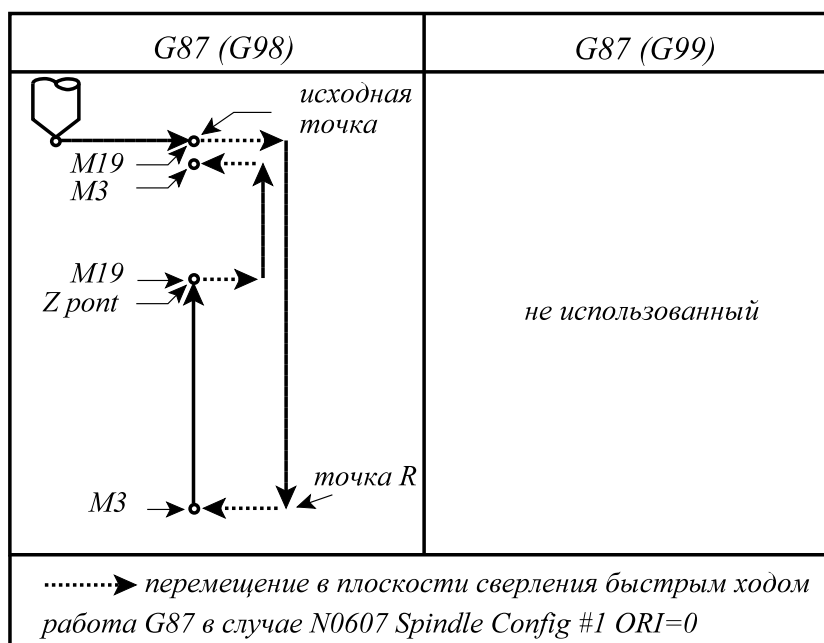


рисунок [19.1.13-2](#)

Переменные, использованные в цикле:

G17 **G87** X_p Y_p I_— J_— Z_p R_— F_— L_—
 G18 **G87** Z_p X_p K_— I_— Y_p R_— F_— L_—
 G19 **G87** Y_p Z_p J_— K_— X_p R_— F_— L_—

При запуске цикла шпинделю надо дать направление вращения M3.

Операции цикла:

- 1-я операция: позиционирование в выбранной плоскости быстрым ходом
- 2-я операция: — ориентировка шпинделя
— отвод инструмента в выбранной плоскости по значению I, J, K быстрым ходом
- 3-я операция: движение быстрым ходом до точки R (точка приближения)
- 4-я операция: — отвод инструмента в выбранной плоскости по значению, противоположному заданному на I, J K, быстрым ходом,
— перезапуск шпинделя в направлении M3
- 5-я операция: расточка до упорной точки подачей F
- 6-я операция: — ориентировка шпинделя: M19
— отвод инструмента в выбранной плоскости по значению I, J, K быстрым ходом
- 7-я операция: —

- 8-я операция: —
 9-я операция: отвод быстрым ходом до исходной точки
 10-я операция: — отвод инструмента в выбранной плоскости по значению, противоположному заданному на I, J а также K, быстрым ходом,
 — перезапуск шпинделя в направлении M3

Из природы цикла следует, что противоположно изложенным раньше, точка приближения, то есть *точка R лежит глубже, чем упорная точка*. Это надо принимать во внимание при программировании сверлильной оси и адресов R.

Поскольку перед расточкой шпиндель выполняет ориентировку, и отводит инструмент от поверхности по значению, заданному на I, J а также K, при заходе можно избежать поломку инструмента.

19.1.14 Цикл сверления, после задержки ручное воздействие в упорной точке (G88)

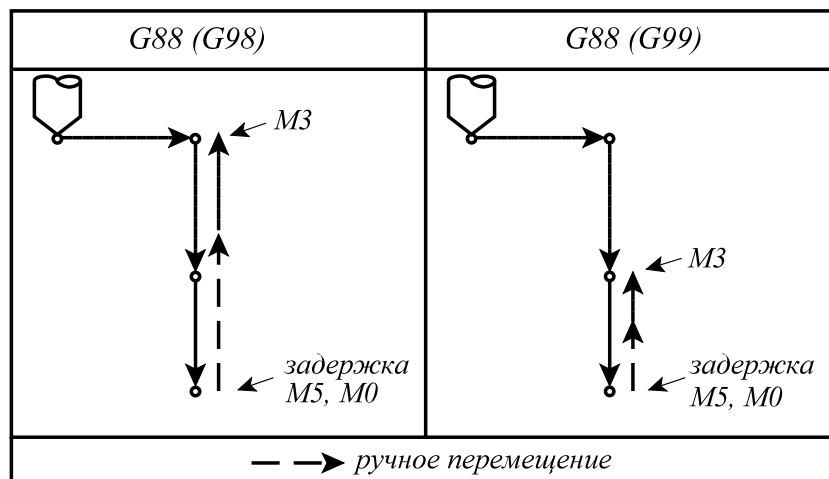


рисунок 19.1.14-1

Переменные, использованные в цикле:

G17 **G88** X_p Y_p Z_p R P F L
 G18 **G88** Z_p X_p Y_p R P F L
 G19 **G88** Y_p Z_p X_p R P F L

При запуске цикла шпинделю надо дать направление вращения M3.

Операции цикла:

- 1-я операция: позиционирование в выбранной плоскости быстрым ходом
 2-я операция: —
 3-я операция: движение быстрым ходом до точки R (точка приближения)
 4-я операция: —
 5-я операция: сверление до упорной точки подачей F
 6-я операция: — задержка до значения P
 — останов шпинделя: M5
 — устройство ЧПУ примет состояние STOP (M0), выходя откуда, оператор может работать станком в каком-то ручном режиме (ПЕРЕМЕЩЕНИЕ, СДВИГ ШАГОМ, МАХОВИЧОК), то есть может отводить вершину

инструмента от поверхности отверстия, и может выводить инструмент из отверстия. После этого выходя из ручного режима, вернувшись в режим АВТОМАТ, после старта продолжается обработка.

- 7-я операция: в случае G99: после START отвод до точки R, быстрым ходом
 8-я операция: в случае G99: перезапуск шпинделя: M3
 9-я операция: в случае G98: отвод быстрым ходом до исходной точки
 10-я операция: в случае G98: перезапуск шпинделя: M3

Цикл то же самый, как в случае A при G87, только имеется задержка перед остановом шпинделя.

19.1.15 Цикл сверления с задержкой, отвод подачей (G89)

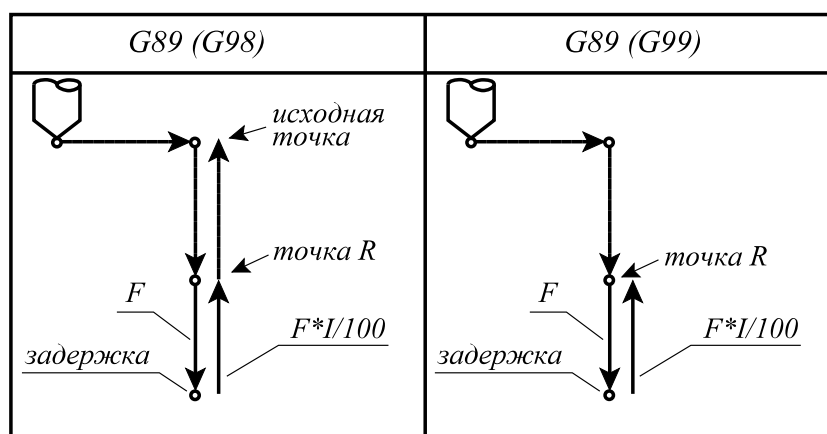


рисунок 19.1.15-1

Переменные, использованные в цикле:

G17 **G89** X_p Y_p Z_p R P F I L

G18 **G89** Z_p X_p Y_p R P F I L

G19 **G89** Y_p Z_p X_p R P F I L

Операции цикла:

- 1-я операция: позиционирование в выбранной плоскости быстрым ходом
 2-я операция: -
 3-я операция: движение быстрым ходом до точки R (точка приближения)
 4-я операция: -
 5-я операция: сверление до упорной точки подачей F
 6-я операция: задержка до значения, заданного по адресу P
 7-я операция: отвод до точки R, подачей F*I/100

Override отвода I (%)

Если в кадре адресу I не присвоили значение, тогда в 7-й операции override подачи отвода будет значение параметра N1502 Extraction Override in G85, G89. На скорость, рассчитанную таким образом, действует ещё и включатель процентного значения подачи, то есть подача будет иметь значение

значение

Готвод = Fзапрограммированная * Override подачи * Extraction Override in G85, G89/100.

- 8-я операция: -

9-я операция: в случае G98: отвод быстрым ходом до исходной точки

10-я операция: -

Цикл совпадает с G85, за исключением задержки.

19.2 Замечания к использованию циклов сверления

- Если в режиме циклов какой-то кадр без кода G содержит любой из следующих адресов, цикл сверления выполняется:

X_p , Y_p , Z_p , или q

где q произвольная, не сверлильная ось.

В противном случае цикл сверления не выполняется.

- Запрограммируя в режиме циклов кадр задержки G04 P, команда выполняется по запрограммированному P, но переменная цикла, относящаяся к задержке **не** удаляется, и **не** изменяется.
- Значение I, J, K, Q, E, R надо задавать в таком кадре, где происходит и сверление, иначе значения **не** сохраняются.

Рассмотрим следующий пример для иллюстрации вышеизложенных:

G81	X_	C_	Z_	R_	F	(цикл сверления выполняется)
	X					(цикл сверления выполняется)
	F					(цикл сверления не выполняется, F изменяется)
	M	S				(цикл сверления не выполняется, код M, S выполняется)
G4	P					(цикл сверления не выполняется, задержка выполняется, переменная цикла по задержке не изменяется)
	I	Q				(цикл сверления не выполняется, запрограммированные значения не регистрируются как переменные цикла)

- Если рядом с циклом сверления программируем и функцию, функция выполняется в конце первой операции, после завершения позиционирования. Если в цикле запрограммирован и L, функция выполняется только в первом проходе.
- Число повторения L модально не передаётся.
- Устройство ЧПУ в покадровом режиме внутри цикла остановится после операции 1, 3, и 10.
- Клавиша STOP не действует в операциях 5, 6 и 7 циклов G74, G84. Если во время этих операций нажать STOP, устройство ЧПУ продолжает свою работу и можно его остановить только в конце операции 7.
- Независимо от положения включателя, подача и override шпинделя всегда 100% в операциях 5, 6 и 7 циклов G84.1, G84.
- Если в кадре цикла запрограммирован Tnnmm, новая коррекция по длине принимается во внимание как при плоскостной позиционировании, так и при сверлении.

20 Функции управления осями

20.1 Точение многоугольника

При точении многоугольника вращаем по сранению друг к другу инструмент и заготовку с определённым соотношением чисел оборотов. Изменяя соотношение чисел оборотов и число режущих кромок вращающегося инструмента, получим многоугольник с разным числом сторон. Произведение соотношения чисел оборотов на число режущих кромок даст число сторон получаемого многоугольника. Если например, соотношение чисел оборотов инструмента и заготовки 2:1, и число режущих кромок вращающегося инструмента 2, можно точить четырёхугольника, если число режущих кромок 3, - то шестиугольника. Таким образом можно просто изготовить точением например, болт с шестигранной головкой, или гайку. Такая обработка намного быстрее, чем фрезерование многоугольника с использованием интерполяции полярных координат, однако полученные боковые стороны не точно плоскостные.

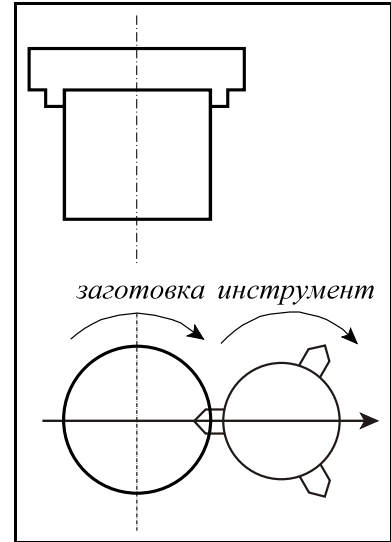


рисунок 20.1-1

20.1.1 Принцип работы точения многоугольника

Пусть будет ось вращения заготовки на координате $X=0, Y=0$.

Пусть будет ось вращения инструмента точка P_0 , координаты которой: $X=A, Y=0$, то есть расстояние оси вращения инструмента от оси вращения заготовки A .

Пусть будет радиус вращающегося инструмента B . Вершина инструмента обозначается P_t . Координаты вершины инструмента в момент $t=0$ будет $X=A-B; Y=0$.

Пусть будет угловая скорость вращения заготовки α , а вращения инструмента будет β .

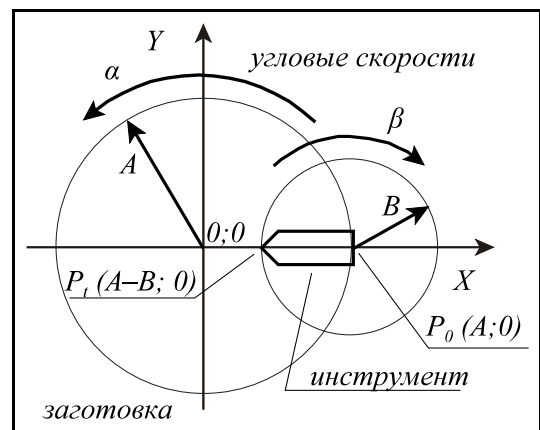


рисунок 20.1.1-1

В момент t координаты вершины инструмента P_t ($x_t; y_t$) будут следующие:

$$x_t = A \cos \alpha t - B \cos(\beta - \alpha)t$$

$$y_t = A \sin \alpha t + B \sin(\beta - \alpha)t$$

Предполагая, что соотношения чисел оборотов инструмента и заготовки 2:1, тогда $\beta=2\alpha$. Подставим в уравнение выше:

$$x_t = A \cos \alpha t - B \cos \alpha t = (A - B) \cos \alpha t$$

$$y_t = A \sin \alpha t + B \sin \alpha t = (A + B) \sin \alpha t$$

Уравнения выше представляют собой уравнения такого эллипса, длина большой оси которой $A+B$, а длина малой оси $A-B$.

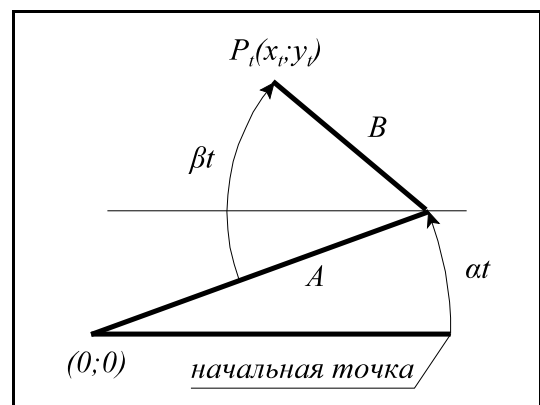


рисунок 20.1.1-2

Если инструменты стоят симметрично по отношению друг к другу на 180° , вытачивается четырехугольник, если на 120° - тогда шестиугольник, предполагая, что соотношения чисел оборотов инструмента и заготовки 2:1.

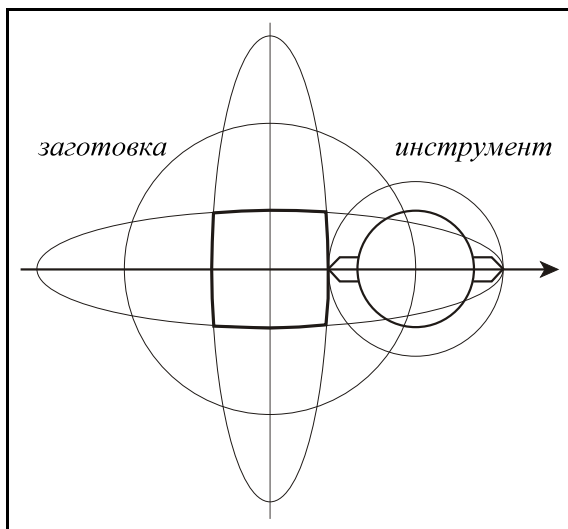


рисунок 20.1.1-3

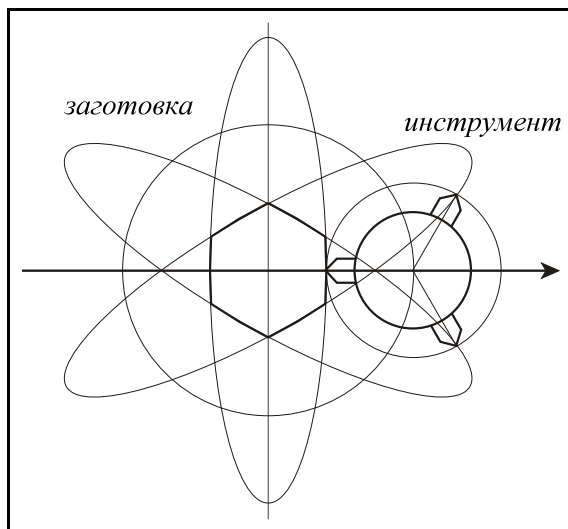


рисунок 20.1.1-4

Кривые, получаемые соотношением чисел оборотов инструмента и заготовки 2:5 отличаются от эллипса, но может быть, даже и таким образом хорошо приближаются к сторонам многоугольника.

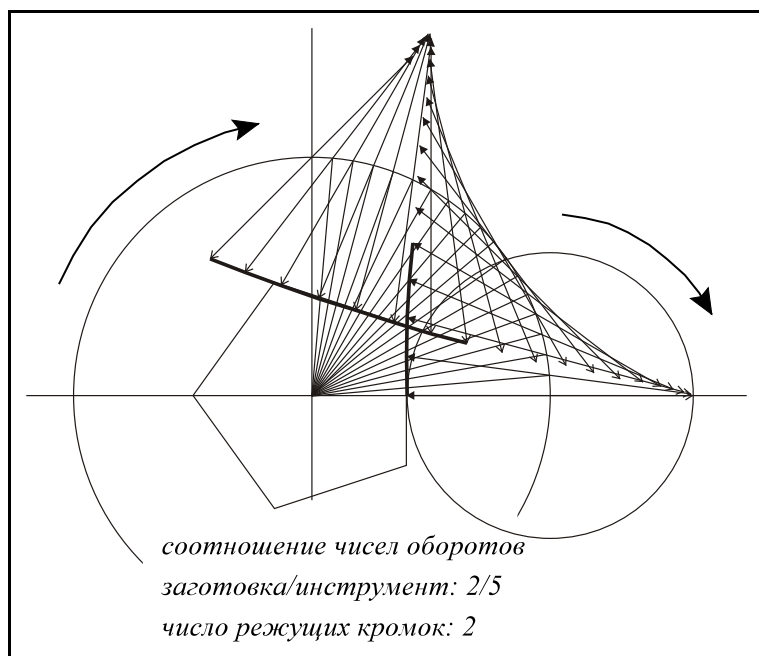


рисунок 20.1.1-5

20.1.2 Программирование точения многоугольника (G51.2, G50.2)

Команда

G51.2 P_ Q_ R__

включает функцию точения многоугольника.

По адресам

P és Q

можно установить соотношением чисел оборотов инструмента и заготовки:

$P/Q = \text{число оборотов заготовки} / \text{число оборотов инструмента}$

Если например, заготовка и инструмент должны вращаться по соотношению 1:2, надо запрограммировать

G51.2 P1 Q2.

По адресу **P** надо задавать всегда *положительное целое* число.

По адресу **Q** можно задавать и *положительное и отрицательное целое* число. В случае задания отрицательного числа направление вращения инструмента будет противоположное направлению вращения заготовки.

Адрес P и Q надо всегда задавать.

Заполнение адреса

R (сдвиг по фазе в градусах)

опциональное. По адресу R можно задавать сдвиг по фазе, то есть *сдвиг по фазе* нулевого импульса шпинделя инструмента по сравнению нулевого импульса шпинделя заготовки *в градусах*. Если не запрограммировать адрес R, нулевой импульс шпинделя заготовки и шпинделя инструмента совпадает при каждом обороте Q/P шпинделя инструмента. Команду G51.2 можно выдавать и во время точения многоугольника, если желаем сдвигать нулевой импульс вспомогательного шпинделя:

G51.2 R__

Если например, желаем выточить четырехугольника резцом с одной режущей кромкой, в первый шаг обрабатываем две противоположные стороны командой

G51.2 P1 Q2.

Расстояние между нулевыми импульсами двух шпинделей 0. Если после этого запрограммировать команду

G51.2 R90

повернув режущую кромку резца на 90° по сравнению заготовки, можно обработать и другие две противоположные стороны.

В ходе точения многоугольника должен быть оборудован датчик как и на шпинделе, вращающем заготовку, так и на шпинделе, вращающем инструмент, и контур регулирования позиции должен быть закрываемым на обоих шпинделях.

В ходе точения многоугольника *максимальное число оборотов шпинделей* ограничивается *скоростью быстрого хода*, установленной на шпиндель. Эта обычно ниже, чем максимальное число оборотов шпинделя.

В ходе точения многоугольника различаем *мастер-шпиндель, вращающий заготовку* и *вспомогательный шпиндель, вращающий инструмент*. *Всегда вспомогательный шпиндель синхронизируется под мастер-шпиндель.*

Мастер-шпинделем точения многоугольника является всегда шпиндель, выделённый в канале главным шпинделем. Способ выбора главного шпинделя определяется строителем станка и содержится в описании данного станка.

Вспомогательный шпиндель точения многоугольника можно выбирать из программы, например, ссылкой на шпиндель и функцией M, что определяется строителем станка и содержится в описании данного станка.

Порядок синхронизации в ходе точения многоугольника следующий:

- если число оборотов мастера-шпинделя больше, чем число оборотов быстрого хода, или получаемое число оборотов вспомогательного шпинделя ($S*Q/P$) больше, чем число оборотов быстрого хода вспомогательного шпинделя, замедляется до соответствующего числа оборотов,
- замыкает контур регулирования позиции,
- вспомогательный шпиндель раскрутится до умноженного на соотношение Q/P числа оборотов мастера-шпинделя ($S*Q/P$), в направлении, идентичном направлению мастера-шпинделя, или противоположного ему,
- вспомогательный шпиндель замыкает контур регулирования позиции,
- затем вспомогательный шпиндель натянет свой нулевой импульс на нулевой импульс мастера-шпинделя, или на размер, равный сдвигу по фазе, заданному по адресу R в градусах от нулевого импульса мастера-шпинделя.

Команда

G50.2

выключает точение многоугольника.

Команду G51.2 и G50.2 надо задавать всегда в отдельном кадре.

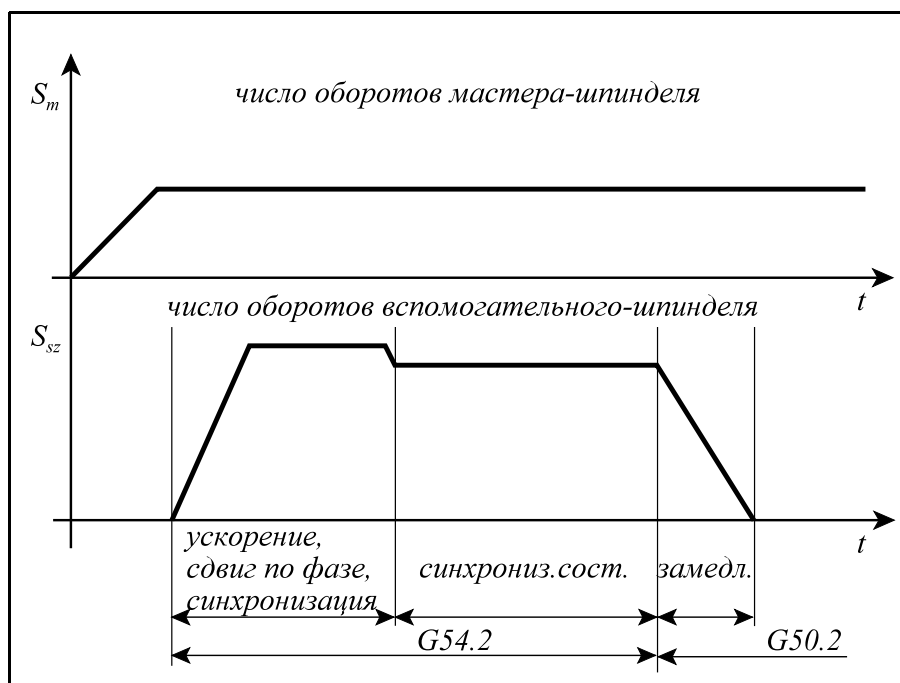


рисунок [20.1.2-1](#)

Действие синхронизации похоже, как действие нулевого импульса шпинделя при нарезании резьбы. Пока не вытащим заготовку из патрона и летучку оставляем в шпинделе инструмента в том же угловом положении, до тех пор движение заготовки и инструмента остаётся в синхроне, можно проходить несколько раз по той же поверхности, например, с целью черновой затем чистовой обработки. Также можно выключить синхронизацию командой G50.2, затем включив её обратно, можно также проходить по поверхности предварительно выточенного многоугольника, предполагая, что запрограммированное соотношение чисел оборотов то же самое, чем прежде.

☞ Внимание!

В ходе программирования многоугольника следите за тем, чтобы получаемое число оборотов для шпинделя инструмента $n_{\text{шпиндель инструмента}} = S \cdot (Q/P)$ никогда не превысило допустимого для него максимальное число оборотов.

Синхронный бег выключается

- аварийным стопом,
- ошибками сервоконтур.

Образцовый пример

Обтачивается шестиугольник. Пусть будет заготовка в шпинделе S1, а инструмент вращается шпинделем S3. Функции M, заданные в примере ниже являются частными, на разных станках могут быть разные коды.

С инструментом с тремя режущими кромками:

```
...
M31 (ВЫДЕЛЕНИЕ ШПИНДЕЛЯ S1 ГЛАВНЫМ ШПИНДЕЛЕМ)
S1=1000 M3
S3=0 M23 (ВКЛЮЧЕНИЕ ШПИНДЕЛЯ S3 К ТОЧЕНИЮ МНОГОУГОЛЬНИКА)
G0 X20 Z1
G51.2 P1 Q-2 (ВКЛЮЧЕНИЕ ТОЧЕНИЯ МНОГОУГОЛЬНИКА)
X15
G1 Z-10 F0.5
G0 X30
Z1
G50.2
S1=0 M5 (ОСТАНОВ S1)
S3=0 M5 (ОСТАНОВ S2)
...
```

С инструментом с одной режущей кромкой:

```
...
G0 X20 Z1
G51.2 P1 Q-2 (ВКЛЮЧЕНИЕ ТОЧЕНИЯ МНОГОУГОЛЬНИКА)
X15
G1 Z-10 F0.5
G0 X30
Z1
G51.2 R120 (СДВИГ ПО ФАЗЕ НА 120 ГРАДУСОВ)
X15
G1 Z-10 F0.5
G0 X30
```

Z1
G51.2 R240 (СДВИГ ПО ФАЗЕ НА 240 ГРАДУСОВ)
X15
G1 Z-10 F0.5
G0 X30
Z1
G50.2
...

20.2 Фрезерование зубчатых колёс способом обкатки (G81.8)

Фрезерование зубчатых колёс способом обкатки содержит описание Программирования устройства ЧПУ фрезерным станком и центром обработки типа NCT3xxM.

20.3 Синхронное перемещение осей

В ходе обработки заготовки может понадобиться синхронное перемещение двух осей. Оси могут находиться и в том же канале, или в разных.

Мастер-осью называется та ось, на которую задаём даты в программе деталей.

Вспомогательной осью называется та ось, которая перемещается в синхроне с мастером-шпинделем. Пока ось является вспомогательным-шпинделем, до тех пор нельзя перемещать ось ни из программы, ни вручную.

Синхронное перемещение приглашает программа PLC например, выполнением кода M.

Параметры и сигналы PLC, применённые при синхронном перемещении

При желании сцеплять две оси для синхронного перемещения, параметром N2101 Synchronous Master, **принадлежащем к вспомогательной оси, надо задавать, что какая по номеру ось является её мастер-осью.**

Мастер-ось может быть в одном и том же канале, но и в разном. Одна мастер-ось может иметь несколько вспомогательных осей. Одна вспомогательная ось может быть и мастер-осью для другой оси.

При синхронном управлении осями можно программировать только мастер-ось, или можно выдавать команду для ручного перемещения (jog, маховичок).

Синхронное перемещение запрашивает PLC, включением индикатора AP_SYNCR в 1, принадлежащего к вспомогательной оси. PLC дожждётся, чтобы NC подтвердил запрос через индикатор AN_SYNCA. Начиная с этого, перемещения мастер-оси получит и вспомогательная ось.

Этот сигнал **обычно могут включить или выключить функции M.**

Например:

```

...
M41 (синхронное перемещение вкл.)
...
M40 (синхронное перемещение выкл.)
...

```

В дальнейшем в этом описании для включения или выключения синхронизации используется пара кодов M40, M41. В случае конкретного станка **код функции и описание работы спрашивайте от строителя станка.**

Синхронное перемещение может быть запущено только строго под действием **функции дождиданья и разгрузки буфера.**

Если синхронное перемещение происходит **между двумя каналами**, в другом канале надо программировать код с дождиданьем:

Программа 1-го канала

```

...
M502 P12
M41 (вкл. синхрон)
M503 P12
...
M504 P12
M40 (выкл. синхрон)
M505 P12
...

```

Программа 2-го канала

```

...
M502 P12
(Ожидает синхрона)
M503 P12
...
M504 P12
(Ожидает выключения)
M505 P12
...

```

Помимо этого, условием синхронного перемещения является, чтобы как на мастер-оси, так и на вспомогательной оси была действующая референтная точка.

При синхронном перемещении **направление движения вспомогательной оси** может совпадать с направлением мастер-оси, но может и отличаться от неё. Направление движения вспомогательной оси можно установить на бите #0 MSY параметра N2102 Synchron Config, принадлежащего к вспомогательной оси. Если значение бита параметра:

- =0: мастер-ось и вспомогательная ось перемещается в одном и том же направлении
- =1: мастер-ось и вспомогательная ось перемещается в противоположном направлении.

Пример для синхронного перемещения осей

Для синхронного перемещения осей берём для примера двуканальный продольный токарный станок.

В шпиндель S1, находящийся в 1-ом канале, питатель прутка поддаёт заготовку и там обрабатывается передняя часть заготовки. В шпинделе S2, находящийся во 2-ом канале, после приёма заготовки обрабатывается её конец.

Заготовку в 1-ом канале вместе с шпинделем S1 перемещает ось Z_1 . А стоячие вертикально инструменты точения T01, ..., T06 перемещают оси X_1 , Y_1 .

Во 2-ом канале шпиндель S2 вместе с заготовкой перемещается в направлении X_2 , Z_2 .

В то время, как в шпинделе S1 обрабатывается передняя часть последующей заготовки, в то же время в шпинделе S2 обрабатывается конец предыдущей заготовки.

Предположим, что оба конца заготовки надо обрабатывать одинаково.

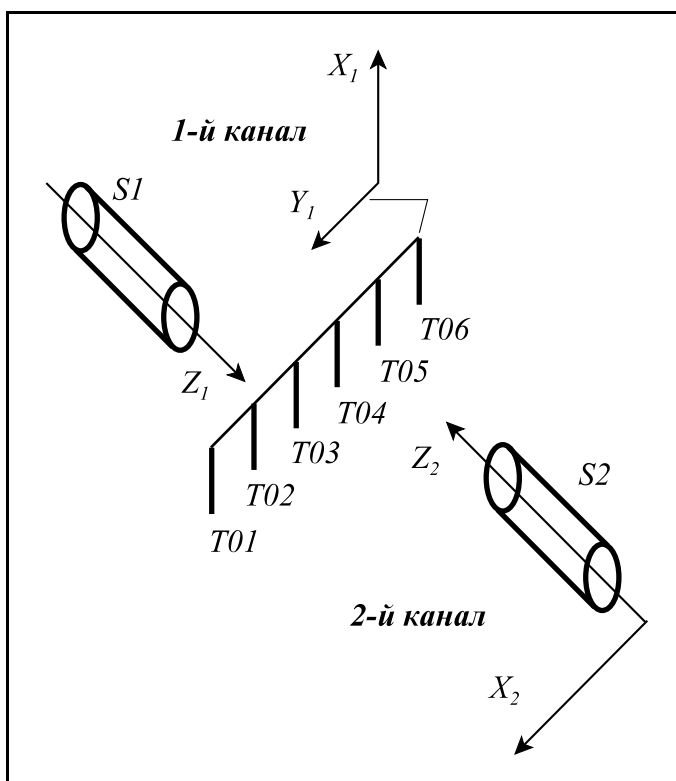


рисунок 20.3-1

Пусть будет в держателе инструмента T01 и T02 инструмент с одинаковой длиной и с одинаковым радиусом скругления. Если встать с осью X_1 , Y_1 в 1-ом канале перед инструментом T01, а с осью X_2 во 2-ом канале перед инструментом T02, **можно и одновременно обрабатывать оба конца заготовки с помощью синхронного перемещения осей Z_1 и Z_2** . Пусть будет ось Z_2 во 2-ом канале синхронной вспомогательной осью оси Z_1 , находящейся в 1-ом канале, что выделяется параметром.

Предположим, что код

M41 сцепляет для синхронного перемещения оси Z_1 и Z_2 , далее для синхронного вращения шпинделей S1 и S2,

а

M40 прекращает синхронную работу.

Коды M41, M40 задаются всегда в 1-ом канале.

Для обработки заготовки, находящейся в шпинделе S1, надо отправить в позицию напротив инструмента **T01** по оси X_1, Y_1, Z_1 в 1-ом канале, а во 2-ом канале оси X_2, Z_2 в позицию напротив инструмента **T02**. После этого следует синхронное точение. Оси Z_1 и Z_2 перемещаются в синхроне. Ось X_2 стоит и конец заготовки в шпинделе S2 обрабатывается перемещением X_1 и Z_2 . Вылет и радиус скругления инструмента T01 и T02 должны быть одинаковыми.

Программа деталей в двух каналах следующая:

Программа, работающая в 1-ом канале	Программа, работающая во 2-ом канале
...	...
T101	T202
G0 X100 Y0 Z10	G0 X0 Z10 (позиция синхрона)
M501 P12 (дожидание)	M501 P12 (дожидание)
M41 (вкл. синхрон Z_1, Z_2)	
M502 P12 (дожидание)	M502 P12 (дожидание)
G42 G0 X0 Z2	(2-й канал не работает)
G1 Z0 F2	
X10 ,C1	
Z-5	
X15	
G40 G0 X100 Z10	
M40 (выкл. синхрон)	
M503 P12 (дожидание)	M503 P12 (дожидание)
... (нормальная обработка)	... (нормальная обработка)

Парковка осей при синхронном перемещении осей

Может быть понадобится, чтобы программа, написанная с синхронным перемещением, работала только на одной стороне, на стороне мастер-оси или вспомогательной оси и не приходилось написать новую программу.

При этом ту сторону, которую не желаем перемещать, можно оставить на “парковке”. Оставить на парковке можно как вспомогательную ось, так и мастер-ось. Устройство ЧПУ и в таком случае рассчитывает соответствующие перемещения как для вспомогательной оси, так и для мастер-оси, но для паркующей стороны не выдаётся команда в движения.

Для оси, участвующей в синхронном перемещении, паркирование запрашивает PLC записыванием индикатора *AP_PARKR* в 1. Устройство ЧПУ подтверждает запрос о паркировании написанием индикатора подтверждения *AN_PARKA* в 1.

Паркование можно вызывать *с клавишей, или функцией M*. В случае конкретного станка *описание работы паркирования спрашивайте от строителя станка.*

20.4 Перемена осей

В ходе обработки заготовки может быть понадобится перемена двух осей. Оси могут находиться в одном и том же канале, или в разных.

Мастер-осью называется ось, с которой переменяем вспомогательную ось.

Вспомогательной осью называется ось, которая вызывает перемену.

Перемену осей вызывает программа PLC, например, выполнением какого-то кода M.

Параметры и индикаторы PLC, применённые при перемене осей

При желании переменять две оси, **параметром** N2104 Composit Axis, **относящимся к вспомогательной оси, надо задавать, что какая по номеру ось является мастер-осью**. Мастер-ось может находиться в одном и том же канале, но и в разном.

При перемене осей можно программировать и мастер-ось и вспомогательную ось, или можно выдавать команду ручного перемещения (jog, маховичок).

Перемену осей вызывает PLC включением индикатора AP_MIXR в 1, относящегося к вспомогательной оси. PLC дожждётся, чтобы NC подтвердил запрос через индикатор AN_MIXA. Начиная с этого две оси переменимы.

Этот индикатор **обычно могут включить или выключить функции M**.

Например:

```
...
M42 (перемену осей вкл.)
...
M40 (перемену осей выкл.)
...
```

В дальнейшем в этом описании для включения или выключения перемены осей используется пара кодов M40, M42. В случае конкретного станка **код функции и описание работы спрашивайте от строителя станка..**

Перемена осей может запускаться строго только под действием **функции с ожиданием и разгрузкой буфера**.

Если перемена осей происходит **между двумя каналами**, в другом канале надо программировать код с ожиданием:

Программа 1-го канала	Программа 2-го канала
...	...
M502 P12	M502 P12
M42 (перемену осей вкл.)	(ожидает перемену)
M503 P12	M503 P12
...	...
M504 P12	M504 P12
M40 (перемену осей выкл.)	(ожидает удаления перемены)
M505 P12	M505 P12
...	...

После перемены осей **направление движения вспомогательной оси** может совпадать с первоначальным направлением, но может быть и противоположным ему. Направление движения вспомогательной оси можно установить на бите #0 MMI параметра N2105 Composit Config, относящегося к вспомогательной оси. Если значение бита параметра:

=0: после перемены осей вспомогательная ось перемещается в первоначальном направлении

=1: после перемены осей вспомогательная ось перемещается в противоположном направлении по сравнению первоначальному.

После перемены осей **направление движения мастер-оси** может совпадать с первоначальным направлением, но может быть и противоположное ему. Направление мастер-оси можно установить на бите параметра N2105 Composit Config, относящегося к мастер-оси. Если значение бита параметра:

=0: после перемены осей мастер-ось перемещается в первоначальном направлении

=1: после перемены осей мастер-ось перемещается в противоположном направлении по сравнению первоначальному.

После перемены осей **программу можно писать на действительной стороне и на виртуальной** (отражённой) **стороне также.**

Программирование на виртуальной стороне можно установить битом #2 МСО параметра N2105 Composit Config, относящегося к вспомогательной оси. Если значение бита параметра:

=0: после перемены осей вспомогательная ось запрограммирована на действительной стороне

=1: после перемены осей вспомогательная ось запрограммирована на виртуальной стороне.

Программирование на виртуальной стороне можно установить битом #2 МСО параметра N2105 Composit Config, относящегося к мастер-оси. Если значение бита параметра:

=0: после перемены осей вспомогательная ось запрограммирована на действительной стороне

=1: после перемены осей вспомогательная ось запрограммирована на виртуальной стороне.

После перемены осей **осям наследственно передаются первоначальные смещения нулевой точки и коррекции по длине.** Если например, переменять между двумя каналами оси X_1 - X_2 и Z_1 - Z_2 , после перемены коррекция по длине, установленная для оси X_1 попадает во 2-й канал и наоборот. Поэтому целесообразно после перемены осей загрузить новое смещение нулевой точки, замеренное после перемены осей, а также новую коррекцию по длине.

Пример для применения перемены осей

Пусть будет *на двухканальном станке вде револьверной головки.*

Первой револьверной головкой обрабатывается **1-й канал**, и может перемещаться вдоль оси X_1 , Z_1 . **Второй головкой обрабатывается 2-й канал** и может перемещаться вдоль оси X_2 , Z_2 .

Из-за приёма заготовки контршпиндель S2, находящийся во 2-ом канале может перемещаться осью W_2 .

В шпинделе S1 обрабатывается одна сторона заготовки, а в шпинделе S2 - вторая сторона. Для обработки в обоих каналах могут понадобиться одинаковые инструменты.

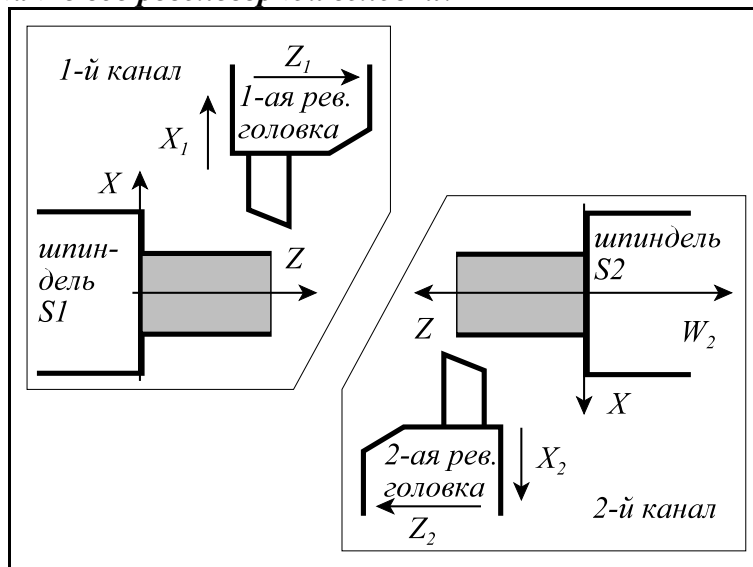


рисунок 20.4-1

В случае сложной заготовки, когда надо работать с многими разными инструментами, может оказываться, что в первой или во второй револьверной головке нет столько мест, чтобы в обоих держать одинаковые инструменты. При этом, если в каком-то канале нужен инструмент, хранившийся в другой револьверной головке, **надо переменять револьверные головки между каналами**, то есть оси X_1 , Z_1 и оси X_2 , Z_2 .

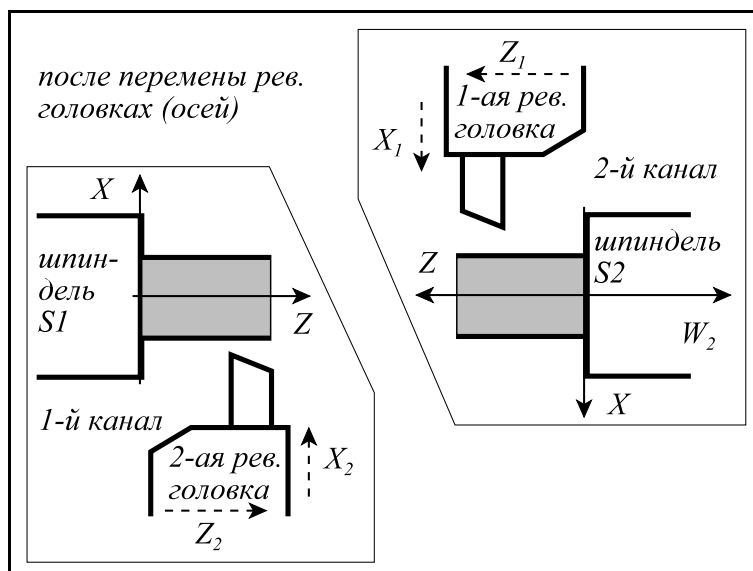


рисунок 20.4-2

После этого в обоих каналах можно выполнить обработку.

Как это видно на приложенном рисунке, *надо переворачивать направление как осей X_1 , Z_1 , так и осей X_2 , Z_2 по сравнению к первоначальному*, записыванием бита ММ1 в 1, для того, чтобы направления приспособились к направлению координат, принятых в канале.

После этого можно решать, что программу желаем написать на *действительной* или на *виртуальной* стороне.

Если после перемены осей желаем написать программу *на действительной стороне* хоть в 1-ом, хоть во 2-ом канале, *движение оси X в положительном направлении приближает инструмент к заготовке*, а движение в отрицательном направлении удаляет от неё. Программировать надо как раз наоборот, как перед переменной осей!

При желании написать программу *на виртуальной стороне*, значит, подобным образом, как перед переменной осей, тогда в бит МСО, относящийся к осям X_1 и X_2 надо записать 1.

При этом устройство ЧПУ написанную таким образом *программу автоматически отражает на ось Z* .

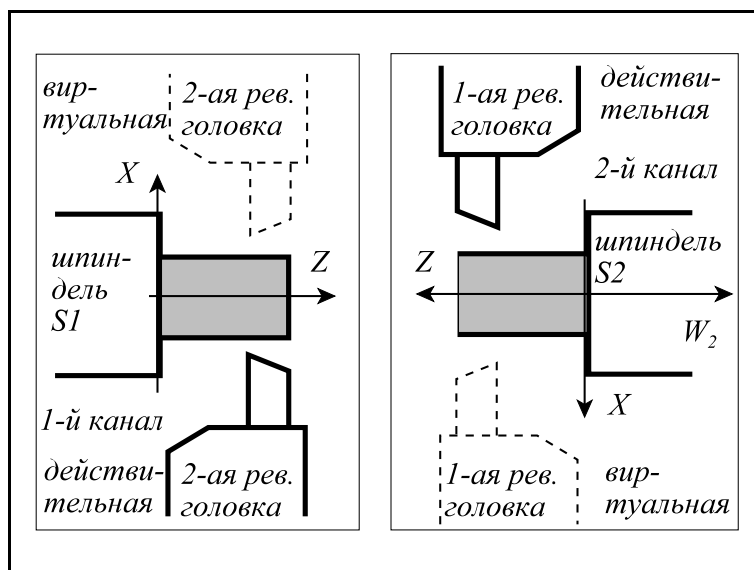


рисунок 20.4-3

На рисунке видна заготовка, находящаяся во 2-ом канале, которую после перемещения осей желаем обработать с инструментами, имеющимися в револьверной головке, перемещаемой осями X_2, Z_2 .

Действительная, абсолютная позиция инструмента в системе координат, прикреплённой к заготовке:

$$x_2 = -120, z_2 = 80.$$

На двух примерах ниже представим, как надо написать **программу на действительной стороне** при положении параметра

$$МСО_{x_2}=0$$

далее как надо написать **программу на виртуальной стороне** при положении параметра

$$МСО_{x_2}=1$$

На виртуальной стороне писать программу

- это такое, как будто не принимали во внимание переменную осей, и работали бы с первоначальной револьверной головкой.

В коррекции радиуса инструмента **код позиции инструмента Q** надо задавать в зависимости от того, что для какой стороны написана программа. В примерах ниже на действительной стороне надо задавать Q2, а на виртуальной стороне - Q3.

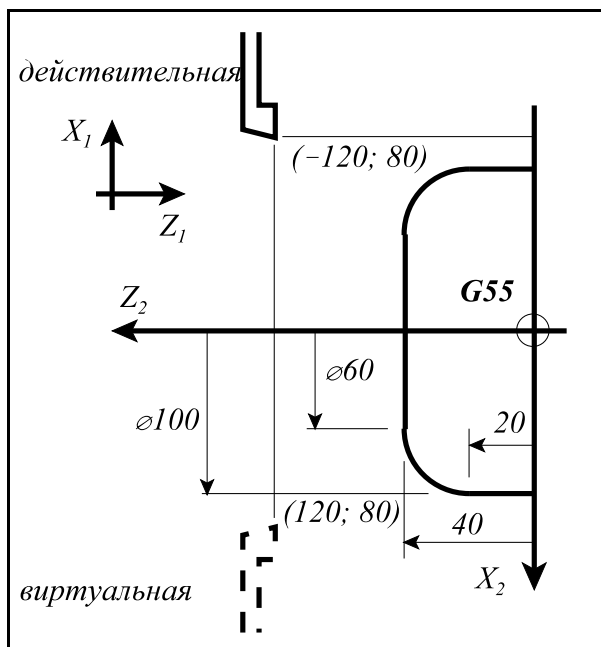


рисунок 20.4-4

Действительная сторона, $МСО_{x_2}=0$:

```
... (обработка осями  $X_2, Z_2$ )
(перемена осей, обработка осями  $X_1, Z_1$ )
M502 P12
G55 T101 (R0.8, Q2)
G0 X-120 Z80
G41 G0 X0 Z40
G1 X-60 F0.5
G2 X-100 Z20 R20
G1 Z0
G0 X-120
G40 Z80
M503 P12
...
```

Виртуальная сторона, $МСО_{x_2}=1$:

```
... (обработка осями  $X_2, Z_2$ )
(перемена осей, обработка осями  $X_1, Z_1$ )
M502 P12
G55 T101 (R0.8, Q3)
G0 X120 Z80
G42 G0 X0 Z40
G1 X60 F0.5
G3 X100 Z20 R20
G1 Z0
G0 X120
G40 Z80
M503 P12
...
```

Естественно, инструмент будет перемещаться всегда по действительной стороне.

Перемена действительной и гипотетической оси

Существует такой случай, когда **приходится изменять одну действительную ось и одну не существующую ось**. Такой случай приводит приложенный рисунок, показывающий кинематику продольного токарного станка.

В шпиндель S1, находящийся в 1-ом канале питатель прутка поддаёт заготовку и там обрабатывается передняя часть заготовки. В шпинделе S2, находящемся во 2-ом канале, после приёма заготовки обрабатывается конец заготовки.

В 1-ом канале заготовку вместе со шпинделем S1 перемещает ось Z_1 . А вертикально стоящие токарные инструменты T01, ..., T06 перемещают оси X_1 , Y_1 .

Во 2-ом канале шпиндель S2 перемещается вместе с заготовкой в направлении X_2 , Z_2 .

Пока в шпинделе S1 обрабатывается передняя часть последующей заготовки, в то же время в шпинделе S2 обрабатывается конец предыдущей заготовки.

Во 2-ом канале нет оси Y_2 .

Предположим, что во 2-ом канале нужно обработать конец заготовки одним из инструментов группы инструментов T01, ..., T06.

При этом **между 1-ым и 2-ым каналом надо изменять оси X , Y и X , Y** . Для того, чтобы механизм перемены завершился, **во 2-ом канале надо выделить одну гипотетическую ось Y_2** .

Положением бита #7 HYP=1 параметра N0106 Axis Properties **можно выделить гипотетическую ось. Гипотетическая ось имеет название** (в нашем случае Y) **и номер**, но среди сервопараметров **не назначен к ним вход и выход физической оси**, то есть к ним не принадлежит вход датчика и выход привода.

После перемены осей, в нашем случае во 2-ом канале, можно ссылаться на ось Y и из программы и ручным перемещением равным образом.

Из рисунка видно, что положительное направление оси Y_2 противоположно положительным направлениям осей Y_1 с той целью, чтобы система координат X_2 , Y_2 , Z_2 была правокрученной. Для этого бит #0 MMI параметра N2105 Composit Config, относящийся к оси Y_2 , надо установить в 1.

Поскольку после перемены осей гипотетическая ось попала в 1-й канал, поэтому здесь нельзя ссылаться на ось Y ни из программы, ни ручным перемещением. Также нельзя ссылаться в 1-ом канале на ось X , так как после перемены ось X перемещала бы шпиндель S2.

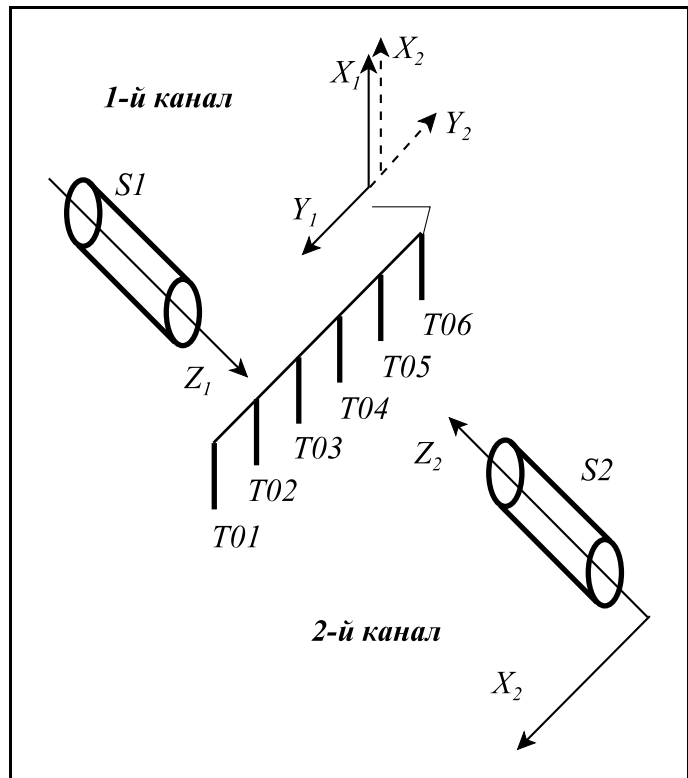


рисунок 20.4-5

20.5 Сверхположенное перемещение осей

В ходе обработки заготовки может быть понадобится, чтобы к перемещению одной оси добавилось перемещение и другой оси. Это называется сверхположенным перемещением осей. Оси могут находиться в одном и том же канале, или в разных каналах тоже.

Мастер-осью называется ось, перемещение которой добавится к перемещению вспомогательной оси.

Вспомогательной осью называется ось, к перемещению которой добавится перемещение мастер-оси. Как мастер-ось, так и вспомогательную ось можно перемещать из программы. Сверхположенное перемещение приглашает программа PLC например, выполнением какого-то кода M.

Параметры и сигналы PLC, применённые при сверхположенном перемещении

При желании сцеплять две оси для сверхположенного перемещения, **параметром N2107 Superimposed Master, относящемся к вспомогательной оси, надо задавать, что какая по номеру ось является её мастер-осью.**

Мастер-ось может находиться в том же канале, но и в другом канале. Одна мастер-ось может иметь и несколько вспомогательных осей. Одна вспомогательная ось может быть и мастер-осью для другой оси.

При сверхположенном управлении осями можно программировать как мастер-ось, так и вспомогательную ось, или выдавать команду ручного перемещения (jog, маховичок).

Сверхположенное перемещение запрашивает PLC включением индикатора AP_SPRPNR в 1, относящегося к вспомогательной оси. PLC дожждётся, чтобы NC подтвердил запрос через индикатор AN_SPRPNA. Начиная с этого перемещения мастер-оси получит и вспомогательная ось и перемещения мастер-оси добавляются к перемещениям вспомогательной оси.

Этот индикатор **могут включить, или выключить обычно функции M.**

Например:

```

...
M43 (сверхположенное перемещение вкл.)
...
M40 (сверхположенное перемещение выкл.)
...

```

В дальнейшем в этом описании используем для включения, или выключения сверхположенного перемещения пару кодов M40, M43. В конкретном случае **код функции и описание работы спрашивайте от строителя станка.**

Сверхположенное перемещение может быть запущено строго только под действием **функции дождиданья и разгрузки буфера.**

Если сверхположенное перемещение происходит **между двумя каналами**, в другом канале надо программировать код с дождиданьем:

Программа 1-го канала

```

...
M502 P12
M43 (сверхположенное перемеще-
ние вкл.)
M503 P12
...
M504 P12
M40 (сверхположенное перемеще-
ние выкл.)
M505 P12
...

```

Программа 2-го канала

```

...
M502 P12
(ожидает включения сверхполо-
женного перемещения)
M503 P12
...
M504 P12
(ожидает выключения сверхполо-
женного перемещения)
M505 P12
...

```

Помимо этого, условием запроса сверхположенного перемещения является, чтобы как на мастер-оси, так и на вспомогательной оси была действующая референтная точка.

При сверхположенном перемещении *к перемещению вспомогательной оси* может добавиться перемещение мастер-оси, но может и вычитаться из него. На бите #0 MSU параметра N2108 Superimposed Config, относящегося к вспомогательной оси, можно установить, чтобы добавилось перемещение мастер-оси к перемещению вспомогательной оси, или вычиталось из него. Если значение бита параметра:

=0: к перемещению вспомогательной оси добавится перемещение мастер-оси

=1: из перемещения вспомогательной оси вычитается перемещение мастер-оси.

Пример для сверхположенного перемещения осей

Пусть будет на двухканальном станке две револьверной головки.

Первой револьверной головкой обращается 1-й канал, и может перемещаться вдоль оси X_1 , а шпиндель S1 вместе с заготовкой перемещает ось Z_1 .

Второй револьверной головкой обращается 2-й канал, и может перемещаться вдоль осей X_2 , Z_2 . Контршпиндель S2, находящийся во 2-ом канале, стоит.

В шпинделе S1 обрабатывается одна сторона заготовки, а в шпинделе S2 обрабатывается другая сторона заготовки.

Предположим, что заготовку, закреплённую в шпинделе S1 обрабатываем снаружи с инструментом, находящимся

в 1-й револьверной головке, а изнутри - с инструментом, находящимся во 2-й револьверной головке.

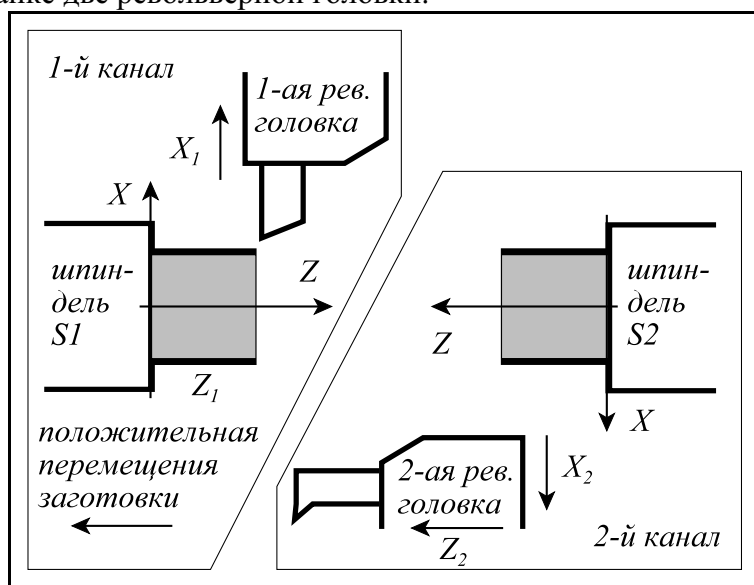


рисунок 20.5-1

Программа, работающая в 1-ом канале, перемещает оси X_1, Z_1 . Одновременно с этим программа, работающая во 2-ом канале перемещает оси X_2, Z_2 . Поскольку *ось Z_1 перемещает заготовку, имеющуюся в шпинделе S1* направо и налево, поэтому *перемещение оси Z_1 надо добавить к перемещению оси Z_2* , чтобы позиция инструмента во 2-й револьверной головке не изменилась по сравнению к заготовке из-за движения Z_1 .

В этой постановке Z_1 является *мастер-осью*, а Z_2 *вспомогательной осью*. Если *мастер-ось Z_1 перемещается в положительном направлении*, надо перемещать и *вспомогательную ось Z_2 в положительном направлении*, чтобы позиция инструмента во 2-й револьверной головке не изменилась по сравнению к заготовке, закреплённой в шпинделе S1. Поэтому на вспомогательной оси надо установить положение бита #0 MSU=0 параметра N2108 Superimposed Config.

Программы, описывающие приложенный чертёж заготовки, следующие:

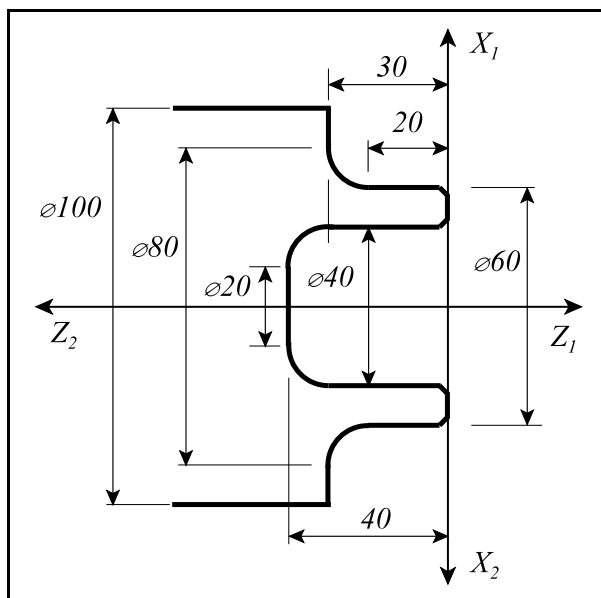


рисунок 20.5-2

Программа, работающая в 1-ом канале

```

...
T101 (позиция инструмента Q3)
G55 G90 G40 G0 X120 Z10
M3 S1=1000
M501 P12
M43 (сверхположенное перемещение)
M502 P12
G42 G0 X50 Z2
G1 X60 ,C3 F0.6
Z-20
G2 X80 Z-30 R10
G1 X100
G40 G0 Z10

M503 P12
M40 (нормальная обработка)
M504 P12
G54 G0 X200 Z20
...

```

Программа, работающая во 2-ом канале

```

...
T2121 (позиция инструмента Q1)
G55 G90 G40 G0 X120 Z-10
M501 P12

M502 P12
G42 G0 X60 Z-5
G1 Z0 F0.8
X40 ,C1
Z30
G2 X20 Z40 R10
G1 X0
G40 G0 Z-10
X120
M503 P12

M504 P12
G54 G0 X150 Z10
...

```

20.6 Изменение направления оси

В устройстве ЧПУ, из программы PLC, можно изменить направление перемещения оси: из команды движения положительного направления получится движение отрицательного направления и наоборот.

Изменение направления оси запрашивает PLC, переключением индикатора AP_MIRR в 1, относящего к данной оси. PLC дожидается, чтобы NC подтвердил запроса через индикатора AN_MIRA. Начиная с этого направление перемещения оси будет противоположное установленному параметром.

Этот индикатор *обычно может включить и выключить функции M.*

В случае конкретного станка *код функции и описание работы запрашивайте от строителя станка.*

Изменение направления оси может запускаться строго только под действием **функции ожидания и разгрузки буфера.**

Изменение направления оси не касается позиционирования (G53), запрограммированного в станочной системе координат и направления движения и позиций (G28, G30 P) посылки в референтную точку.

Пример на изменение направления оси

Револьверная головка T1 с инструментами, имеющимися в ней, обрабатывает переднюю часть закреплённой в шпинделе S1 заготовки. При этом инструмент перемещается по направлению X-Z по отношению к заготовке. После этого заготовка передаётся в шпиндель S2, находящийся в револьверной головке T1 (смотри рисунок). После поворота револьверной головки S2 попадает напротив T2, содержащей раскладку инструментов группы инструментов, которыми обрабатывается задняя часть заготовки. При этом заготовка, закреплённая в S2, перемещается по отношению группы инструментов T2.

Для того, чтобы программы можно было написать традиционным образом (в положительном направлении инструмент удаляется от заготовки), далее, чтобы система координат осталась правосторонней, то есть не изменились направления окружностей, направление коррекции на радиус и т.д., необходимо повернуть и направление движения оси X и Z.

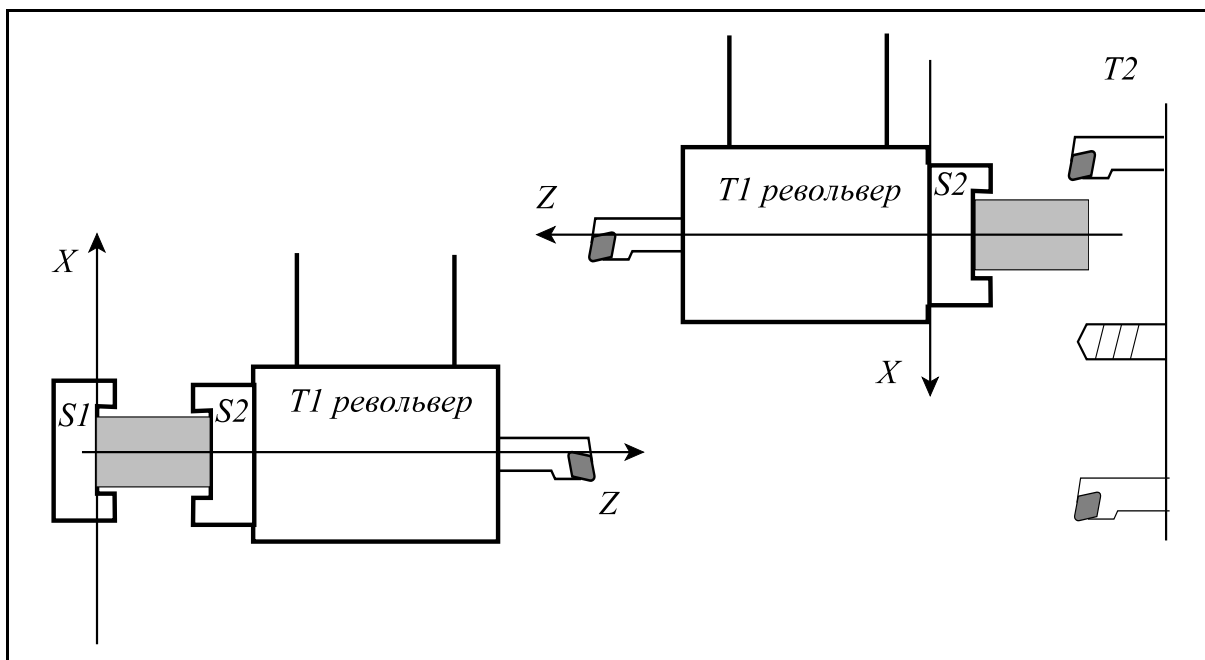


рисунок 20.6-1

Например, пусть будет M44 функцией M, выполняющей изменение направления оси, а кодом M40 удаляется изменение направления.

```

...
G54 T101
...
M44 (изменение направления по оси X-Z)
G55 T2020 (новая нулевая точка и коррекция по длине заготовки)
...
M40 (удаление изменения направления по оси X-Z)
G54 T101 (новая нулевая точка и коррекция по длине заготовки)
...

```

После изменения направления оси необходимо загрузить новую систему координат заготовки, замеренную после изменения направления оси и новую коррекцию по длине, относящуюся к группе T2. Подобная ситуация имеется и после удаления изменения направления.

20.7 Обращение осями, не перпендикулярными одной к другой

Если на станке перемещение одной оси не перпендикулярное к другой, а **образует угол по сравнению перпендикуляру**, необходимо применять обращение осями, образующее угол, или обращение косо́й осью.

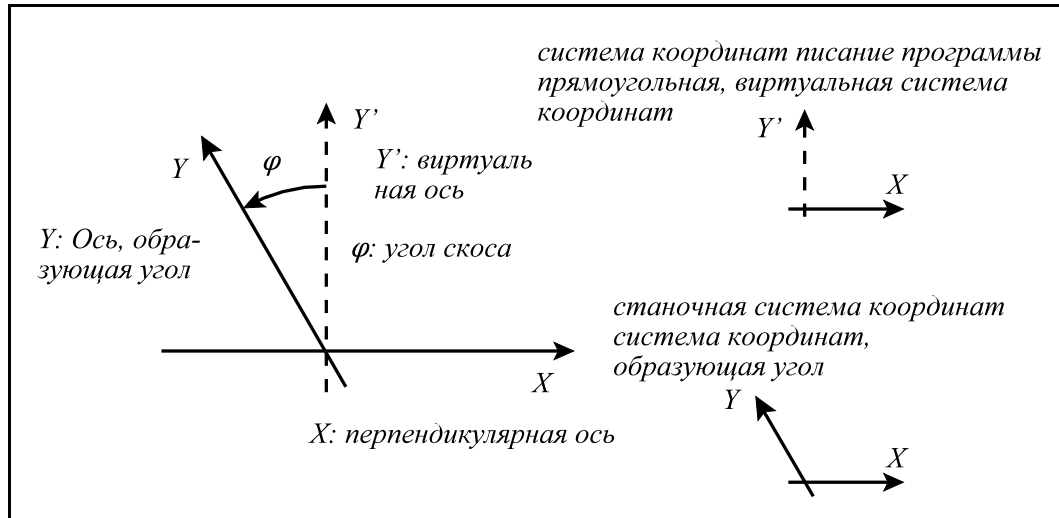


рисунок 20.7-1

В каждом канале можно выделить параметром одну пару осей для обращения осями, образующими угол.

Перпендикулярной осью называем ось в прямоугольной системе координат, **угол образующей осью** или **косой осью** называем ось, не перпендикулярную к этой.

Программа деталей описывается **в прямоугольной, виртуальной системе координат**, на рисунке в системе координат X-Y', где Y' является **виртуальной осью**.

Функция, выполняющая обращение косо́й осью, написанную таким образом программу автоматически пересчитывает в систему координат, образующую угол.

Отображение позиции выполняется **в прямоугольной, виртуальной системе координат**.

Обращение косо́й осью с помощью указателя PLC, например функциями M, можно выключить из программы или включить его.

Перемещение

Обозначение

X_a актуальное перемещение вдоль перпендикулярной оси,

Y_a актуальное перемещение вдоль косо́й оси,

X_p запрограммированное перемещение вдоль перпендикулярной оси,

Y'_p запрограммированное перемещение вдоль виртуальной оси,

φ угол, образованный виртуальной осью и косо́й осью.

Зависимость между запрограммированным и виртуальным перемещением следующая:

$$Y_a = \frac{Y'_p}{\cos \varphi}$$

$$X_a = X_p + cY'_p \operatorname{tg} \varphi$$

где

$c=1$, если перпендикулярная ось (X_p -t) задаётся радиусом,
 $c=2$, если перпендикулярная ось (X_p -t) задаётся диаметром.

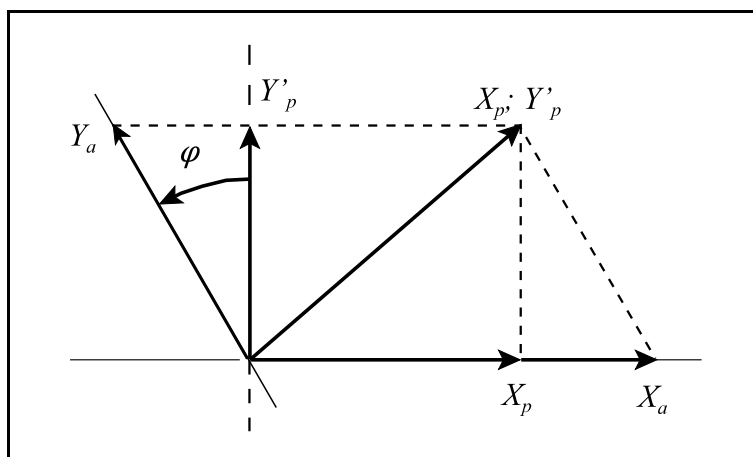


рисунок 20.7-2

Подача

Запрограммированная подача задаётся всегда в прямоугольной системе координат.

Обозначение

F_p запрограммированная подача,

F_{ax} составляющая подачи вдоль перпендикулярной оси,

F_{ay} составляющая подачи вдоль косой оси,

φ угол, образованный между виртуальной осью и косой осью.

Запрограммированная подача разбивается на основании следующей зависимости на компоненты, действующие вдоль реальных осей:

$$F_{ya} = \frac{F_p}{\cos \varphi}$$

$$F_{xa} = F_p + F_p \operatorname{tg} \varphi$$

Выключение обращения косой осью

Обращение косой осью в управлении можно включить а также выключить нажатием кнопки, или функцией М.

Место кнопки или функция М определяется строителем станка. Функцию М можно использовать и во время выполнения программы, при условии, если функция М назначена для разгрузки буфера.

Записав указателя PLC **CP_NOANGCR** в 1, выключается обращение косой осью, как только управлением отключается сигнал подтверждения от **CN_NOANGCA**.

Параметры, устанавливающие обращения косой осью

Включив бит параметра **#0 ANE N0111 Angular Axis Control** в 1, разрешается обращение косой осью.

Параметром **N0112 Slant Angle** можно задавать в градусах отклонение направления косой оси по сравнению к перпендикуляру.

Параметром **N0113 Axis Number of Slant** можно задавать номер косой оси. Задавать можно исключительно только существующие в данном канале оси.

Наконец параметром **N0114 Axis Number of Cartesian** можно задавать номер той перпендикулярной оси, на которую воздействует перемещение косой оси. Задавать можно исключительно только существующие в данном канале оси.

Позиционирование и интерполяция системе координат заготовок

1-й случай: Обращение косо́й осью включено (положение указателя PLC: CN_NOANGCA=0)

Данные позиции и подачи задаются всегда **в виртуальной, прямоугольной системе координат**.

Отображение позиции происходит также **в виртуальной системе координат**.

Позицию перпендикулярной оси интерполятором **корректируется** позицией косой оси.

Пример:

N1 G0 G90 Y100 (позиционирование из P0 в точку P1)
N2 X200 (позиционирование из P1 в точку P2)

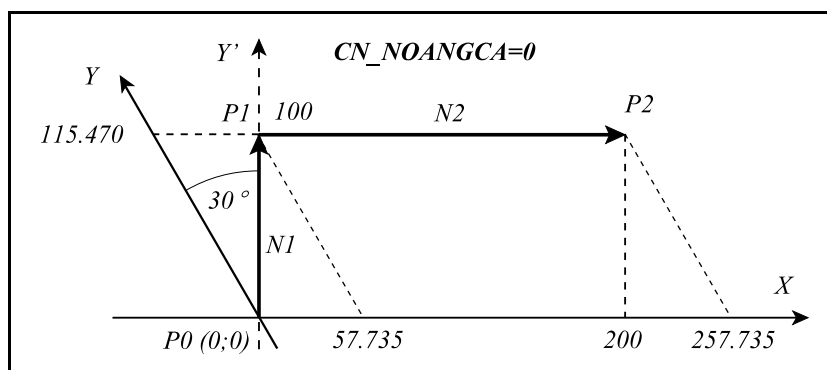


рисунок 20.7-3

Позиции в конце кадра N1 в точке P1:

Запрограммированная позиция

X0.000
Y100.000

Позиция в реальной системе координат

X 57.735
Y115.470

Позиции в конце кадра N2 в точке P2:

Запрограммированная позиция

X200.000
Y100.000

Позиция в реальной системе координат

X257.735
Y115.470

Приведенные выше относятся к **ручным перемещениям**.

Выбирая косую ось для перемещения при jog, или маховичке, **позицию перпендикулярной оси корректируется позицией косой оси**, и перемещение выполняется в виртуальной системе координат.

Например, если нажать кнопку приращения jog Y, и величина шага 1мм, ось Y перемещается по станочному 1.155мм, в то же время ось X - 0.578мм

2-й случай: Обращение косой осью выключено (положение указателя PLC: CN_NOANGCA=1)

Данные позиции и подачи задаются всегда в реальной системе координат.

Отображение позиции происходит также **в реальной системе координат.**

Позицию перпендикулярной оси интерполятором **не корректирует** позицией косой оси.

Пример:

N1 G0 G90 Y100 (позиционирование из P0 в точку P1)
N2 X200 (позиционирование из P1 в точку P2)

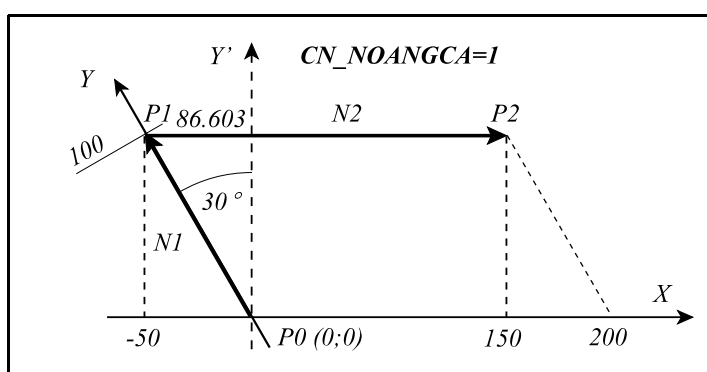


рисунок 20.7-4

Позиции в конце кадра N1 в точке P1:

Запрограммированная позиция

X0.000
Y100.000

Позиция в виртуальной системе координат

X-50.000
Y86.603

Позиции в конце кадра N2 в точке P2:

Запрограммированная позиция

X200.000
Y100.000

Позиция в виртуальной системе координат

X150.000
Y86.603

Изложенные выше относятся и **к ручным перемещениям.**

Например, если нажать кнопку приращения jog Y, и величина шага 1мм, ось Y перемещается 1 мм, в то же время ось X не перемещается.

Позиционирование в станочной системе координат: G53

В станочной системе координат на позиционирование не влияет положение указателя CN_NOANGCA. Перемещение косой оси не компенсирует позицию перпендикулярной оси.

Данные позиции истолкуются в реальной, косой системе координат.

Пример:

N1 G53 Y100 (позиционирование из P0 в точку P1)
N2 G53 X200 (позиционирование из P1 в точку P2)

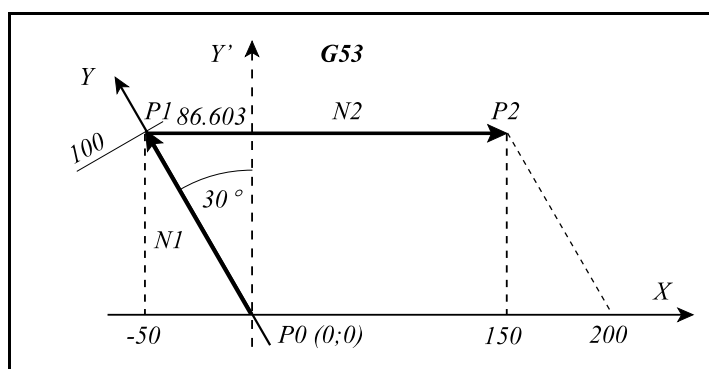


рисунок 20.7-5

Позиции в конце кадра N1 в точке P1:

Виртуальная позиция
X -50.000
Y 86.603

Станочная позиция
X 0.000
Y100.000

Позиции в конце кадра N2 в точке P2:

Виртуальная позиция
X150.000
Y 86.603

Станочная позиция
X200.000
Y100.000

Циклы сверления

Если косая ось выделена осью сверления, сверление возможно только тогда, если не включено обращение косой осью, (положение CN_NOANGCA=0), впрочем выводится сообщение об ошибке “Цикл сверления в положении CN_NOANGCA=1 нельзя”. В случае G84.2, G84.3 выполняется интерполяция между шпинделем и виртуальной осью.

Выполнение кодов G28, G30

Позиция *промежуточных точек* истолкуется всегда **в виртуальной системе координат**. В зависимости от состояния указателя CN_NOANGCA,

перемещение косой оси компенсирует позицию перпендикулярной оси (CN_NOANGCA=0), или не выполняет компенсацию по перпендикулярной оси (CN_NOANGCA=1).

Позиции, заданные параметром N0200 Reference Position1, ..., N0203 Reference Position4 задаются всегда **вдоль косой оси**.

При перемещении из промежуточной точки в позицию референции перемещение косо́й оси не компенсирует позицию перпендикулярной оси.

Ручной приём референтной точки и G28

В случае ручной референтной точки, далее при выполнении G28, если ещё не принята референтная точка она выполняется в зависимости указателя CN_NOANGCA, или не компенсацию перпендикулярной оси.

Задача данных конечного положения

Позиции конечного положения необходимо задавать и принимать во внимание всегда вдоль косо́й оси.

Трансформации

Трансформации (G52, G51, G51.1, G68, G68.2) истолкуются всегда в виртуальной системе координат.

Данные смещения нулевой точки и коррекции по длине

*Данные смещения нулевой точки и коррекции по длине, относящиеся к косо́й оси сохраняются всегда **согласно виртуальной оси**.*

Это необходимо и потому, что если вылет инструментов, или нулевую точку заготовки замеряем внешним измерительным прибором, смещения получим в прямоугольной системе координат.

Замер нулевой точки и коррекции по длине

В ходе замера смещения необходимо считать всегда от виртуальной позиции оси.

Макропеременные

Макропеременные истолкуются следующим образом:

#5001..., #5041..., #5061: информации о позиции **в виртуальной** системе координат

#5021...: информации о позиции в реальной системе координат, **с косо́й осью**

#5081..., #5121: коррекциям по длине **в виртуальной** системе координат

#5101...: отставание в реальной системе координат, **с косо́й осью**

#100651...: смещение нулевой точки маховичка **в виртуальной** системе координат

#5181...: остаточный ход **в виртуальной** системе координат

#10001...: коррекции по длине **в виртуальной** системе координат

#5201..., #2001..., #7001..., #12061...,: смещения нулевых точек и основные повороты **в виртуальной** системе координат

#5521...: динамические смещения нулевых точек **в виртуальной** системе координат

#5501...: актуальные динамические смещения нулевых точек **в виртуальной** системе координат.

G10

В ходе ввода смещения нулевых точек и коррекций заданные данные задаются **в виртуальной** системе координат.

Синхронное перемещение оси, смена оси, совмещённые перемещения

Выше перечисленные перемещения осей возможно между осями, образующими угол между собой таким образом, что косо́я ось другой косо́й осью, а перпендикулярная ось другой перпендикулярной осью:

X1 перпендикулярная ось - пара - X2 перпендикулярная ось

Y1 косая ось -пара - Y2 косая ось.

При иной установке выводится сообщение об ошибке: “Косая ось сикрон, совмещение, перемещение, смена осей ошибка установки”.

21 Функции измерения

21.1 Измерение с удалением остаточного хода (G31)

Под действием команды

G31 v (P) (F)

в точку с координатой *v* с **линейной интерполяцией** запускается **движение**. Движение длится до тех пор, пока не поступит **внешний сигнал удаления** (например, сигнал измерительного щупа), или пока устройство ЧПУ не достигает **позицию конечной точки**, заданную на координатах *v*. После поступления сигнала удаления, или в запрограммированной конечной точке кадра, устройство ЧПУ замедляет и останавливается.

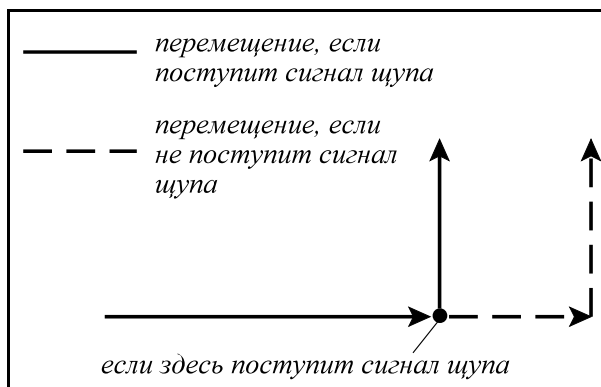


рисунок 21.1-1

По адресу P можно задавать, что из 8 сигналов щупа, вводимых в устройство ЧПУ **какой по номеру сигнал** принимать во внимание в ходе движения:

P1: приём во внимание сигнала 1-го щупа

P2: приём во внимание сигнала 2-го щупа

...

P8: приём во внимание сигнала 8-го щупа.

Заполнение адреса P не обязательно, если не заполнять адрес P, принимается во внимание сигнал 1-го щупа.

Функцию G31 надо использовать всегда в состоянии **G94 (номинутная подача)**. В ходе движения **подачей F**:

- является заданное, или модальное значение F, если бит #0 SKF параметра N3001 G31 Config будет 0,
- является значение подачи, принятое с параметра N0311 G31 Feed, если бит #0 SKF параметра N3001 G31 Config будет 1.

Команда **G31 не передаётся модально**, действует только в том кадре, в котором запрограммирована.

В момент прихода внешнего сигнала **сохраняется позиция осей** на следующих **макропеременных**:

#5061, или #100151, или #_ABSKP[1]: позиция 1-й оси

#5062, или #100152, или #_ABSKP[2]: позиция 2-й оси

...

Сохранённая позиция на макропеременных выше:

- если **поступил внешний сигнал**, это будет позиция, принятая **в момент поступления сигнала**,
- если **не поступил внешний сигнал**, это будет позиция запрограммированной **конечной точки** кадра G31.

Данные позиции

- сохраняются всегда **в актуальной системе координат заготовки**,
- сохраняются **без учёта актуальной коррекции по длине** (G43, G44).

После прихода внешнего сигнала движение остановится с замедлением. При этом позиция конечной точки кадра G31 в зависимости от применённой в кадре подачи в небольшой мере отличается от позиции, сохранённой переменными #5061... при поступлении сигнала. Позиции конечной точки кадра доступны на переменных #5001... Последующий кадр движения начинает действовать, начиная от этих позиций конечной точки.

Выполнение кадра G31 возможно только в состоянии **G15, G40, G50, G50.1 G69, G94**. В противном случае выводится сообщение об ошибке “2055 Щуп в состоянии Gnn”.

Значение, заданное координатами v может быть **и инкрементным и абсолютным**. Если задание координат последующей команды движения инкрементное будет, перемещение считается с той точки кадра G31, где перемещение прекратилось в предшествующем кадре.

Например:

N1 G31 G91 Z100

N2 X100 Z30

В кадре N1 запускается инкрементное перемещение в направлении Z. Если после поступления внешнего сигнала устройство ЧПУ остановится в точке с координатой $Z=86.7$, отсчитывая с этой точки инкрементно совершает 100 шагов в направлении X, 30 шагов в направлении Z в кадре N2.

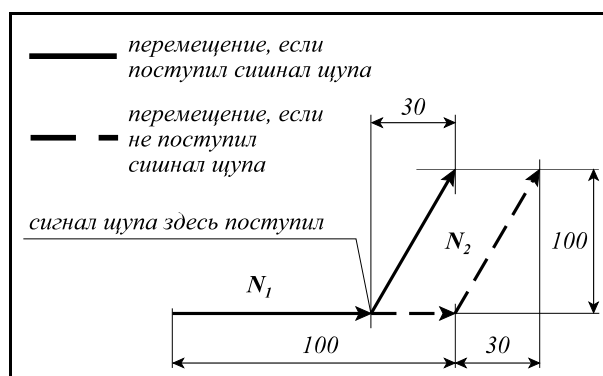


рисунок 21.1-2

Если запрограммировано абсолютное задание данных, перемещение получится следующим образом:

N1 G31 G90 Z200

N2 X200 Z300

Кадр N1 запускает в направлении Z перемещение в точку с координатой $Z=200$. Если после поступления внешнего сигнала устройство ЧПУ остановится в точке с координатой $Z=130$, в кадре N2 перемещение в направлении Z получится $Z=300-130$, то есть $Z=170$.

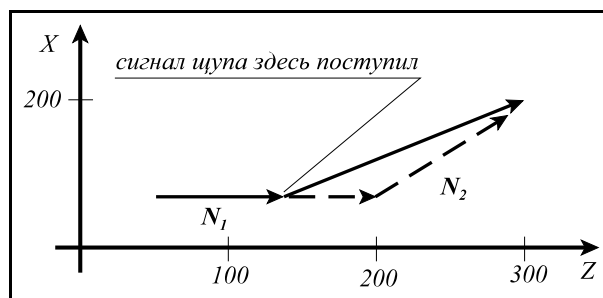


рисунок 21.1-3

21.2 Удаление остаточного хода по пределу момента (G31)

Функцией G31 можно пользоваться и таким образом, что **останов перемещения и удаление остаточного хода** происходит не по внешнему сигналу (например, по сигналу щупа), а по тому, что на одной **выделённой оси момент двигателя достигает заданное значение**, например так, что инструмент прижимаем к какой-то фиксированной поверхности. После достижения предела момента, или в запрограммированной конечной точке кадра устройство ЧПУ остановится, затем переходит к выполнению следующего кадра.

Описанное выше осуществляется функцией

G31 P98 Qq v Ff,

где:

Q: запрограммируемое значение предела момента, в процентах от максимального момента двигателя. Значение Q0 соответствует 0%-у, а Q255 соответствует 100%-у.

Задаваемые значения: Q1-Q254

v: имя и конечная позиция запрограммированной оси, на которой устройство ЧПУ наблюдает за пределом момента. *Его можно задавать только по адресу одной оси.*

F: значение подачи в мм/мин-ах, или в дюйм/мин-ах. (Нужно состояние G94).

В ходе перемещения *подача F:*

- заданное, или модальное значение F, если для параметра N3001 G31 Config бит #0 SKF равно 0,
- значение подачи, взятое от параметра N0311 G31 Feed, если для параметра N3001 G31 Config бит #0 SKF равно 1.

Команда **G31 P98 не передаётся наследственно**, действует только в том кадре, в котором запрограммирована.

Образцовый пример:

```
N1 G31 P98 Q50 Z30 F100
N2 G0 Z200
```

В кадре N1 инструмент достигает поверхность заготовки в точке “A”. Отсчитывая от этой точки ось Z уже не перемещается. Поскольку двигатель ещё не достиг предел момента Q50 (19.6%), устройство ЧПУ не остановит перемещение.

Ошибка позиции на оси Z непрерывно растёт от точки “A” до точки “B”, а при этом момент двигателя непрерывно растёт.

Выдача команды движения продолжается до точки “B”, где двигатель достигает предел момента.

Устройство ЧПУ в точке “B” остановит перемещение, зарегистрирует позицию точки “A”, затем приступает к выполнению следующего кадра, кадра N2.

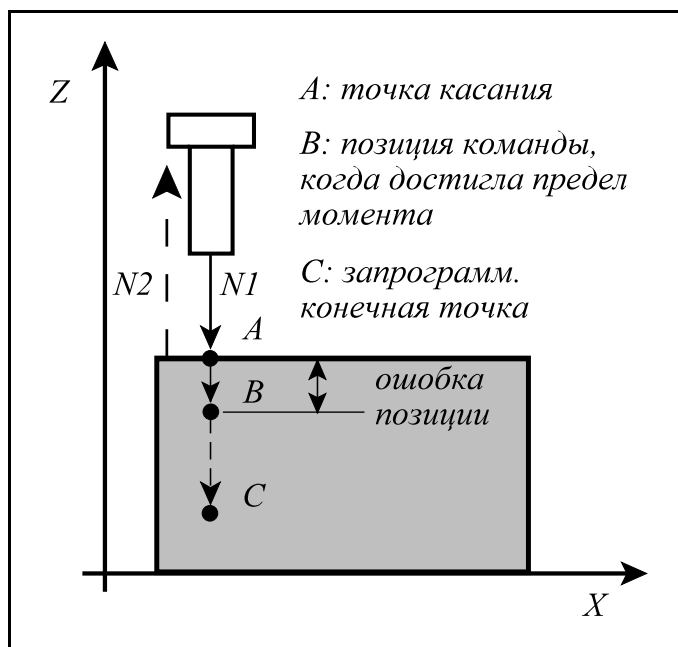


рисунок 21.2-1

Ошибка позиции непрерывно растёт от точки “A” до точки “B”, а при этом момент двигателя растёт, так как инструмент прижимается к заготовке. Поэтому для того, чтобы устройство ЧПУ не набежало на сервоошибку, во время выполнения функции, на параметре N3019 Servo Limit during Torque Limit Skip можно установить предел ошибки, больше установленного на параметре N0520 Serrl2. Устройство ЧПУ выдаёт сообщение “3157

Сервоошибка G31 во время P98 “только после выхода из этого предела. Это сервоошибка не причиняет аварийное состояние, но отложит выполнение программы.

В момент достижения предела момента **позиция оси, зарегистрированная в точке “А” сохраняется** на следующих **макропеременных**:

#5061, или #100151, или #_ABSKP[1]: позиция 1-й оси,

#5062, или #100152, или #_ABSKP[2]: позиция 2-й оси.

...

Позиция, сохранённая на перечисленных выше макропеременных:

- если поступил **сигнал предела момента**, позиция точки “А”,
- если не поступил **сигнал предела момента**, позиция G31 запрограммированной **конечной точки** кадра P98.

Данные о позиции сохраняются

- всегда **в актуальной системе координат заготовки**,
- **без учёта** актуальной **коррекции по длине** (G43, G44).

Сообщения об ошибке

Если не заполнен адрес Q, выводится сообщение об ошибке “2004 Не хватает данные Q”.

Если значение адреса Q меньше, чем 1, или больше, чем 254, выводится сообщение об ошибке “2039 Ошибка задания Q”.

Если в функции имеется ссылка на больше одного адреса оси, выводится сообщение об ошибке “2035 Ошибочное задание данных оси <адрес оси>”, где <адрес оси> представляет собой адрес 2-й запрограммированной оси по порядку нумерации осей.

Если устройство ЧПУ не получает информацию о моменте (тока) от запрограммированной оси, выводится сообщение об ошибке “2156 Ошибка состояния щупа в канале 98”.

21.3 Автоматическое измерение длины инструмента (G36, G37)

Команда

G36 X_

G37 Z_

измеряет и поправляет коррекцию загруженного инструмента по длине в случае G36 по координате X, далее в случае G37 по координате Z. Значение X, Z истолкуются всегда абсолютными данными, в актуальной системе координат заготовки.

Перемещение **до позиции q** - **Rapid Distance** совершается **быстрым ходом**, где Rapid Distance является расстоянием, установленным на оси X параметром N3006 Rapid Distance X, а на оси Z - параметром N3010 Rapid Distance Z, а q - это значение, запрограммированное по адресу X, Z.

После этого перемещение продолжается **подачей**, пока не по-

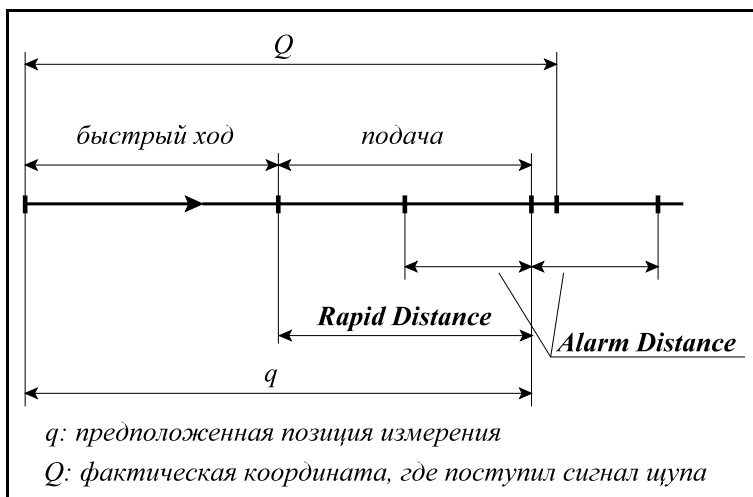


рисунок 21.3-1

ступит сигнал измерительного щупа, или пока устройство ЧПУ **не выводит сообщение об ошибке**. Устройство ЧПУ выводит сообщение об ошибке “2104 Позиция щупа вне пределах”, если сигнал щупа приходит вне окружения с радиусом Alarm Distance предположенной позиции измерения q (запрограммированное по адресу X , или Z). Расстояние Alarm Distance является значением, установленным на оси X параметром N3007 Alarm Distance X , а на оси Z - параметром N3011 Alarm Distance Z .

Значение подачи в кадре G36, G37:

- подача, полученная модально из программы, если бит #0 TLF параметра N3003 G36, G37 Config будет 0, или
- значение, установленное параметром N0312 G37 Feed, если бит #0 TLF параметра N3003 G36, G37 Config будет 1.

Если измерение успешно совершилось, и сигнал щупа поступил в точке с координатой Q , возможны следующие случаи для поправления коррекции в предварительно вызванных регистрах коррекции по длине:

- если бит #1 TMW параметра N3003 G36, G37 Config будет 0, поправляется **геометрическая коррекция**,
- если бит #1 TMW параметра N3003 G36, G37 Config будет 1, поправляется **коррекция на износ**.

Поправка коррекции по длине:

- если бит #2 TCA параметра N3003 G36, G37 Config будет 0, **разность $q-Q$ вычитается** из соответствующей **коррекции**,
- если бит #2 TCA параметра N3003 G36, G37 Config будет 1, **разность $q-Q$ добавляется** к соответствующей **коррекции**.

До начала измерения **надо вызывать соответствующую коррекцию по длине**.

- Команда G36, G37 является однократной.
- Цикл G36, G37 выполняется всегда в актуальной системе координат заготовки.
- Параметры Rapid Distance и Alarm Distance являются всегда положительными значениями. Для двух параметров должны быть удовлетворены следующие условия: Rapid Distance > Alarm Distance.
- Функцию можно вызвать только в состоянии G15, G50, G50.1, G69, G94, в противном случае выводится сообщение об ошибке.

Образцовый пример

```
G55 G15 G50 G50.1 G69 G94
...
G0 X300 Z200
T505
X20 Z100
G37 Z50 F200
Z100
X200
Z50
G36 X40
X200
...
```

22 Функции безопасности

В устройстве ЧПУ можно установить три зоны безопасности:

- **Конечные положения:** определяют пределы хода осей. Ограничивают всегда запрещённые извне зоны, с помощью выключателей, или параметров.
- **Ограничение рабочей зоны:** включается, или выключается из программы функцией G22, G23. Можно установить его и заданием параметров. Может определить запрещённые изнутри и извне зоны.
- **Запрещённая изнутри зона:** ограничивает такую зону, вовнутрь которой нельзя войти. Устанавливается с помощью параметров.

22.1 Конечное положение

Конечные положения ограничивают ход осей. Обращение конечными положениями может совершиться выключателем, или параметром. Конечные положения запрещают всегда наружные зоны.

Обращение конечным положением выключателями

Сигналами выключателя конечного положения обращается программа PLC и передаёт их для NC. Если какая-то ось набежала на выключатель конечного положения, для устройства ЧПУ выводится сообщение

“3018 Набег на положительный выключатель конечного положения на # оси”

или

“3019 Набег на отрицательный выключатель конечного положения на # оси”,
где # название оси.

Недостатком ограничения хода выключателем конечного положения **является, что устройство ЧПУ начинает замедлять после набега на выключатель**. На станках, обладающих большим быстрым ходом из-за длинной **пути замедления надо бы сильно уменьшить ход**, чтобы даже с быстрого хода сумела ось остановиться. Кроме этого, не работает и наблюдение за конечным положением до запуска движения.

Обращение конечным положением параметром

В случае обращения конечным положением параметром устройство ЧПУ в каждый момент знает, насколько далеко ходит от конечного положения, и в зависимости от скорости оси начинает замедление всегда в подходящий момент. Поэтому **конечное положение с параметром максимально может использовать ход станка**.

Если какая-то ось набежала на конечное положение с параметром, устройство ЧПУ выводит сообщение

“3010 Набег на положительное конечное положение на # оси”, или

“3011 Набег на отрицательное конечное положение на # оси”,

где # название оси.

Конечные положения с параметром определяются строителем станка.

Наблюдение за конечным положением с параметром действует только после приёма референтной точки. На тех станках, на которых нет абсолютной измерительной системы,

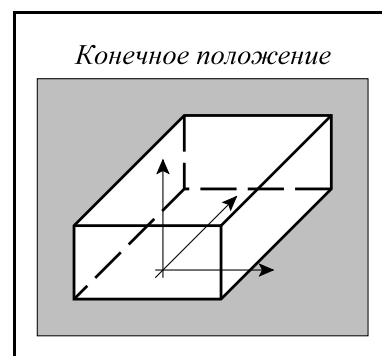


рисунок 22.1-1

то есть надо принять референтную точку, до приёма референтной точки ограничена величина быстрого хода по причинам безопасности. На осях, оборудованных абсолютной измерительной системой, конечное положение с параметром действует сразу после включения.

Параметром можно установить **два диапазона конечного положения**, А и В. Во время работы станка **программа PLC определяет**, что на какой оси и в каком направлении **действовало конечное положение А, или В**.

Например, в нормальном случае на оси Z действует конечное положение “А”. Если при смене инструментов рычаг смены захватит инструмент, программа PLC переключает на диапазон конечного положения В, и при этом диапазон конечного положения оси Z установлен на столько туго, чтобы перемещение оси Z не могло сорвать рычаг смены.

Спуск с конечного положения

Если во время прогона программы, или ручного перемещения какая-то ось набежит на конечное положение, **спускаться оттуда можно только ручным перемещением**.

22.2 Ограничение рабочей зоны, устанавливаемой с параметра/программируемой (G22, G23)

Ограничение рабочей зоны с параметра

Выделить ограничение рабочей зоны можно с установкой следующих параметров. Параметры можно установить только в режиме Редактирования (Edit).

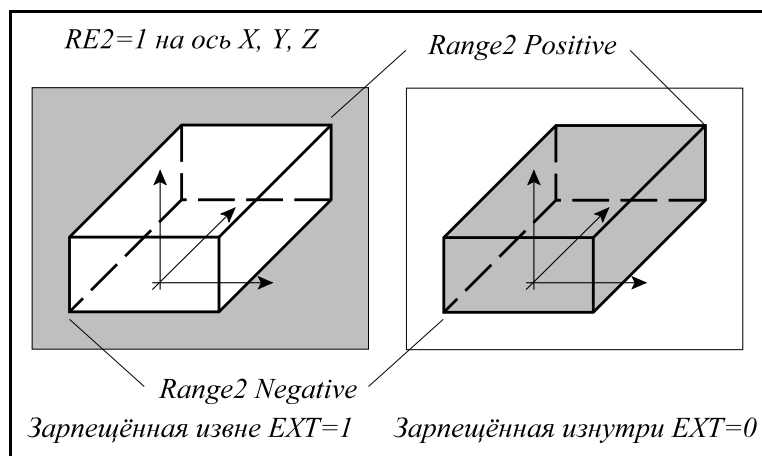


рисунок 22.2-1

Установкой бита #1 RE2 параметра N1000 Range Enable в 1 надо **выделить по осям**, чтобы данная ось участвовала в ограничении рабочей зоны.

Параметром N1006 Range2 Positive и N1007 Range2 Negative надо задавать **в станочной системе координат** точку в положительном и отрицательном направлении желаемой ограничивать рабочей зоны по осям.

Условие

$$N1006 \text{ Range2 Positive} > N1007 \text{ Range2 Negative}$$

должно удовлетвориться для всех выделённых осей!

На параметре можно выделить для ограничения рабочей зоны **все оси**, имеющиеся на станке с помощью разрешающего бита и станочных позиций.

На бите #0 EXT параметра N1001 StrkContr можно задавать, чтобы **выделённая рабочая зона была запрещена изнутри** (в случае EXT=0), **или снаружи** (в случае EXT=1).

Записав бит #1 STE параметра N1001 StrkContr в 1 **надо разрешать ограничение рабочей зоны**. Если бит STE будет 0, ограничение рабочей зоны не работает.

Биты EXT и STE надо установить по каналам. Ограничение рабочей зоны относится всегда к осям, принадлежащим к данному каналу.

После записи, или изменения параметров в одном из ручных режимов надо выдавать индивидуальный кадр

G23

Под действием этого принимается во внимание записанные, или изменённые параметры.

Рисунок показывает ограничение рабочей зоны для осей X, Y, Z. Естественно, можно выделить для этой функции больше и меньше этого осей.

Если станок набежит на запрещённую изнутри зону (EXT=0), устройство ЧПУ выводит сообщение об ошибке

“3042 Запрещённая изнутри зона 2”.

Если станок набежит на запрещённую снаружи зону (EXT=1) в положительном, или отрицательном направлении, устройство ЧПУ выводит сообщение об ошибке

“3040 Запрещённая зона 2 # +” или

“3041 Запрещённая зона 2 # -”

где # название оси.

Выход из запрещённой с параметра рабочей зоны

Если во время прогона программы, или ручного перемещения какая-то ось набежит на запрещённую зону, надо поступить следующим образом:

Если **рабочая зона запрещена снаружи**.

Из зоны можно выходить **ручным перемещением**, подобно спуску с конечного положения.

Если **рабочая зона запрещена изнутри**.

Включением индикатора PLC: После *reset* программы, **удалить ошибку**. Включением индикатора CP_LIM2DIS PLC (например, **нажмите и держите нажато кнопку**) можно отложить наблюдение за 2-й запрещённой зоны, и можно выходить из неё не изменив параметры. *О подробностях спрашивать строителя станка.*

Установкой параметра: **установкой параметра STE=0 выключим** наблюдение за ограничением рабочей зоны, в индивидуальном кадре выдаём команду G23, затем **ручным перемещением покидаем зону**, потом **включим обратно наблюдение** установкой параметра STE=1 и снова выдаём команду G23.

Ограничение рабочей зоны из программы

Команда

G22 X Y Z I J K R

включает наблюдение за ограничением рабочей зоны. Командой можно ограничивать диапазон перемещения осей. Значение адресов команды:

X: предел положительного направления на оси X,

I: предел отрицательного направления на оси X,

Y: предел положительного направления на оси Y,

J: предел отрицательного направления на оси Y,

Z: предел положительного направления на оси Z,

K: предел отрицательного направления на оси Z,

Для заданных данных должны удовлетвориться следующие условия:

$$X \geq I, Y \geq J, Z \geq K$$

Все данные координат (X, Y, Z, I, J, K) надо задавать в станочной системе координат.

По адресу P можно задавать, что наружу, или внутрь выделённой зоны нельзя ходить.

В случае P=0 внутренность выделённой зоны запрещена,

В случае P=1 наружность выделённой зоны запрещена.

Под действием команды G22 устройство ЧПУ не принимает во внимание значения, установленные параметром во 2-ом диапазоне, только те значения, которые задавали в команде G22.

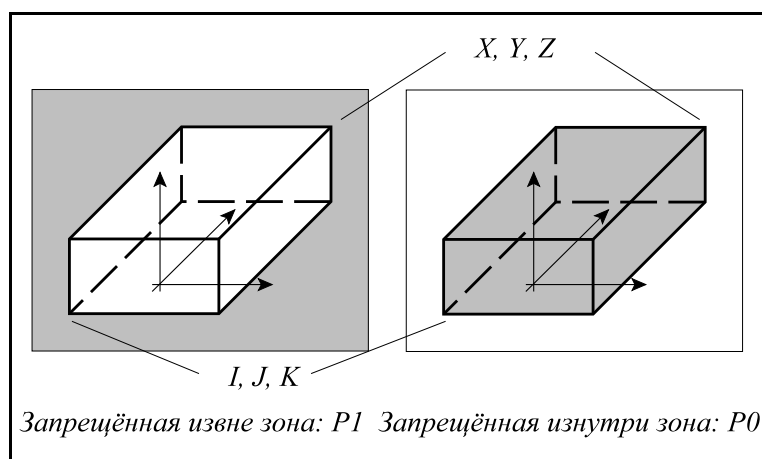


рисунок 22.2-2

Команда

G23

выключает наблюдение за ограничением рабочей зоны.

Команда G23 удаляет пределы диапазонов, установленные в команде G22, и заодно восстанавливает значения наблюдения за 2-ым диапазоном, установленные параметром.

- Ограничение рабочей зоны можно задавать только для **главных шпинделей**.
- Команды G22, G23 надо задавать в самостоятельном кадре.
- Ограничения рабочей зоны будет действовать после включения и приёма станочной референтной точки.
- Если $X=I, Y=J, Z=K$ и $P=0$ вся зона разрешена.
- Если $X=I, Y=J, Z=K$ и $P=1$ вся зона запрещена.

Если станок набежит в зону, запрещённую изнутри (G22 P0), устройство ЧПУ выводит сообщение об ошибке

“3042 Запрещённая изнутри зона 2”.

Если станок набежит в зону, запрещённую снаружи (G22 P1) в положительном, или в отрицательном направлении, устройство ЧПУ выводит сообщение об ошибке

“3040 Запрещённая зона 2 # +” или

“3041 Запрещённая зона 2 # -”

где # название оси.

Выход из запрещённой зоны, запрограммированной функцией G22

Если во время прогона программы, или ручного перемещения какая-то ось набежит на запрещённую зону, надо поступить следующим образом:

Если *рабочая зона запрещена снаружи*.

Из зоны можно выходить *ручным перемещением*, подобно спуска с конечных положений.

Если *рабочая зона запрещена изнутри*.

Включением индикатора PLC: После *reset* программы, *удалить ошибку*. Включением индикатора CP_LIM2DIS PLC (например, *нажмите и держите нажато кнопку*) можно отложить наблюдение за 2-й запрещённой зоны, и можно выходить из неё. *О подробностях спрашивать строителя станка.*

Из программы: В ручном режиме *функцией G23 выключим* наблюдение за ограничением рабочей зоны, *ручным перемещением покидаем зону*, затем *включим обратно наблюдение* повторной выдачей полной команды G22.

Если в устройстве ЧПУ установлена параметром 2-я запрещённая зона, но командой G22 выдаём ограничение рабочей зоны, заданная с помощью G22 зона будет запрещена до тех пор, пока с помощью G23 не удалить его. После этого запрещена будет опять заданная параметром 2-я зона.

22.3 Запрещённая изнутри зона

В устройстве ЧПУ с помощью параметров всегда можно установить запрещённую изнутри зону. Если какая-то ось, а может быть несколько осей набежит на эту зону, или на её границу, устройство ЧПУ выводит сообщение об ошибке “3043 Запрещённая изнутри зона 3”.

Выход из запрещённой изнутри зоны

Включением индикатора PLC: После *reset* программы *удалить ошибку*. Включением индикатора CP_LIM3DIS PLC (например, *нажмите и держите нажато одну кнопку*) можно отложить наблюдение за 3-й запрещённой зоны, и можно выходить из неё не изменив параметры. *О подробностях спрашивать строителя станка.*

Установкой параметра: Если станок набежит на запрещённую изнутри зону, надо включить на всех осях бит #2 RE3 параметра N1000 Range Enable, разрешение наблюдение за запрещённой изнутри зоны, записав бит RE3 в 0. После этого ручным перемещением надо выходить из зоны, затем надо включить обратно биты #2 RE3. Биты RE3 можно устанавливать только в режиме Редактирования (Edit).

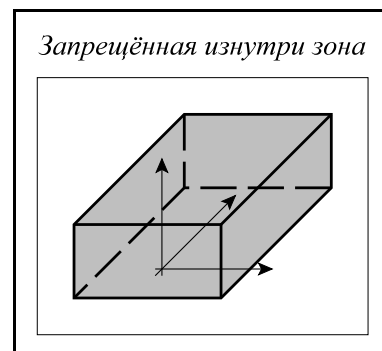


рисунок 22.3-1

22.4 Наблюдение за запрещённой зоной перед запуском движения.

При положении #2 CBM=1 параметра N1001 StrkCont, *прежде чем запускается команда движения*, устройство ЧПУ *рассматривает* в автоматическом режиме, или в режиме ручного ввода данных, или при запуске индивидуального кадра, *что конечная точка данного кадра не попадает ли в какую-то запрещённую зону*.

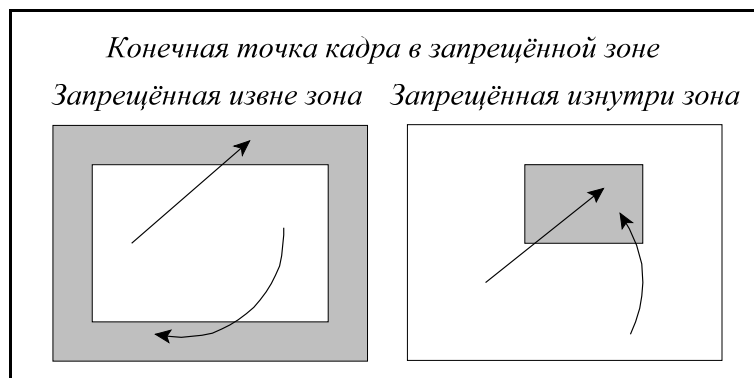


рисунок 22.4-1

Если конечная точка кадра **попадает вне 1-й запрещённой зоны, вне конечного положения**: устройство ЧПУ выводит сообщение

“2056 Конечная точка в положительном конечном положении на # оси”

“2057 Конечная точка в отрицательном конечном положении на # оси”

где # название ссылаемой оси, и **движение не запускается**.

Если конечная точка кадра **попадает в 2-ую, запрещённую изнутри зону**, устройство ЧПУ выводит сообщение

“2060 Конечная точка в 2-й запрещённой изнутри зоне”

и **движение не запускается**.

Если конечная точка кадра **попадает в 2-ую, запрещённую снаружи зону**, устройство ЧПУ выводит сообщение

“2058 Конечная точка в запрещённой зоне 2 # +”

“2059 Конечная точка в запрещённой зоне 2 # -”

где # название ссылаемой оси, и **движение не запускается**.

Если конечная точка кадра **попадает в 3-ью, запрещённую изнутри зону**, устройство ЧПУ выводит сообщение

“2061 Конечная точка в 3-й запрещённой изнутри зоне”

и **движение не запускается**.

При положении #2 CBM=1 параметра N1001 StrkCont, **прежде чем запускается команда движения**, устройство ЧПУ **рассматривает** в автоматическом режиме, или в режиме ручного ввода данных, или при запуске индивидуального кадра, **что траектория данного кадра не пересекает ли запрещённую изнутри зону**.

Если **траектория кадра пересекает 2-ую запрещённую изнутри зону**, но конечная точка не находится внутри запрещённой зоны, устройство ЧПУ выводит сообщение

“2062 Вход во 2-ую запрещённую изнутри зону ”

и **движение не запускается**.

Если **траектория кадра пересекает 3-ью, запрещённую изнутри зону**, но конечная точка не находится внутри запрещённой зоны, устройство ЧПУ выводит сообщение

“2062 Вход в 3-ью запрещённую изнутри зону ”

и **движение не запускается**.

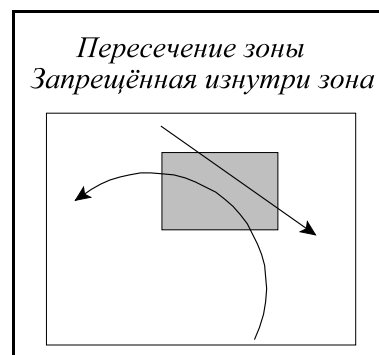


рисунок 22.4-2

В рассмотренных выше случаях ошибка устраняется изменением запрограммированных координат, изменением нулевых точек, или коррекций инструментов.

23 Макропрограммирование

Традиционный язык программирования NC описывает желаемую траекторию заданием кодов G, M, S, T, включает, или выключает разные функции. Отдельным адресам задаётся конкретное численное значение. Например, при желании отправить оси в позицию X50 Y100, тогда надо запрограммировать кадр

```
G0 X50 Y100
```

Используя команды макроязыка, *не нужно задавать конкретное числовое значение данному адресу*, например, X50, а *можно присвоить ей и переменное значение*, например, можно писать в программу и то, что

```
G#105 X#102 Y#110
```

где #105, #102, #110 представляет собой значение трёх разных переменных, которым раньше задавали какое-то значение.

В языке программирования можно использовать различные *аритметические выражения* и *функции* также, как например, сложение, извлечение квадратного корня, синус, и т.д. Можно использовать команды, выполняющие *присвоение значения, проверку условия, разветвление*, выполнение *цикла*.

Язык программирования позволяет *вызов таких подпрограмм, макросов*, которым *можно передавать аргументы* (параметры) из приглашающего кадра.

Заданием параметров можно создать такие т. н. *системные макросы*, а также *системные подпрограммы*, которыми пользователь может расширить, изменить команды традиционного языка программирования с кодами G на основании своих требований.

23.1 Переменные языка программирования

В главной программе, в подпрограммах и макросах отдельным *адресам можно задавать и переменные*, вместо конкретных числовых значений.

Переменным можно присвоить значения в пределах допускаемых значений. Использование переменных программирование можно переделать более гибким.

Переменные могут быть

локальными переменными, которые используются в макровыводах для передачи аргументов,

глобальными переменными, которые доступны на каждом уровне макровывода, далее

системными переменными.

Системные переменные представляют собой такие внутренние данные NC, которые можно прочесть, или можно изменить из программы деталей.

23.1.1 Ссылка на переменные

На переменные можно ссылаться *номером*, а на системные переменные или номером, или *символично*.

Ссылку на переменную вводится всегда со знаком #.

Задание ссылки номером

Идентификатор переменной обозначается *номером, последующим за знаком #*:

#<номер>

Например:

#12

#138

#5106

На переменную *можно ссылаться косвенным* образом, формулой также: #[<формула>]

Например:

#[#120]

означает: в 120-ой переменной имеется тот номер, на какую переменную по номеру ссылается.

#[#120-4]

означает: из числа, имеющегося в 120-ой переменной вычитав 4, получим номер ссылаемой переменной.

Ссылка на системные переменные символом

Ссылку символом также вводится *со знаком #*, затем за ним следует символ (подчёркивание):

#_<символ>

Например:

#_ALM (сообщение об ошибке)

Символическим переменным в некоторых случаях надо задавать и *индекс*. Индекс надо заключить *[и] в квадратные скобки*:

#_<символ>[индекс]

Например:

#_ABSIO[3]

означает: позиция 3-й оси в конце кадра, осевой индекс: [3].

В элементах программного кадра различные адреса могут принимать не только числовые значения, но и значения переменных. После адресов и в случае ссылки на переменную можно использовать знак минус "-", а также оператор I, где это разрешено в случае числового значения. Например:

G#102

если #102=1.0, тогда эта ссылка эквивалентна G1

XI-#24

если #24=135.342, тогда эта ссылка эквивалентна XI- 135.342

– номер кадра: N, условный кадр: / после его адресов ссылка не переменную не разрешено. Номером кадра считается всегда первый адрес N, записанный в кадр.

– Номер переменной нельзя заменить переменной, то есть нельзя писать, что

##120 (ошибочное).

Правильное задание:

#[#120] (правильное).

– Если переменную использовать после адреса, значение переменной не должна превышать область значений, допускаемую для данного адреса. Например,

#112=123456789

после присвоения значения ссылка

M#112

приводит к сообщению об ошибке.

– Если переменную использовать после адреса, значение переменной округляется до значащего цифра, соответствующего адресу. Например:

в случае #112=3.23 M#112 M3 будет,

в случае #112=3.6 M#112 M4 будет.

23.1.2 Изображение чисел макропеременных

Макропеременные за редким исключением являются числами с плавающей запятой.

Макропеременные, не являющимися числом с плавающей запятой, отметим в отдельном описании.

Для изображения чисел с плавающей запятой в устройстве ЧПУ следует по стандарту **IEEE754** изображения чисел **с двойной точностью, с плавающей запятой**. Эти числа изображаем на 64 битах.

Изображение чисел с двойной точностью, с плавающей запятой возможно чисел, начиная от $\pm 5.0 \times 10^{-324}$ до $\pm 1.7 \times 10^{308}$

и 0, с точностью 15-16 десятичных цифр. При вводе надо использовать **десятичную запятую** (,), но при вводе целых чисел не надо её выставить:

#100=256

Не надо вводить ни ведущие ноли, ни последующие ноли. Знак + (положительный) можно пропустить:

#100=134.89654

23.1.3 Локальные переменные: #1 – #33

Локальные переменные используются в данном месте макропрограммы, локально.

Локальные переменные используются обычно *для передачи аргумента*.

Локальные переменные *многоуровневые*, как к главной программе, так и к различным макровыводам принадлежат разные уровни, поэтому они называются локальными. Например, значение #1 в главной программе может быть иное, чем скажем на 2-ом уровне макровыводов.

После возвращения из макровывода локальные переменные данного уровня уничтожаются до #0, удаляются до полной пустоты. **Локальные переменные главной программы уничтожаются в конце программы.**

Сопоставление адресов аргументов с локальными переменными, далее обращение уровнями изложено во главе [23.3](#) Вызов макросов, системных макросов, системных подпрограмм на странице [346](#).

Та локальная переменная, адрес которой не фигурирует в выделении аргументов, она пустая и можно её свободно использовать.

23.1.4 Глобальные переменные: #100 - #499, #500 - #999

В отличие от локальных переменных, *глобальные переменные одинаковые по всему каналу*, независимо от того, что они используются в главной программе, в подпрограмме, или в макросах, далее, что на каком уровне макровывода.

Глобальные переменные *совершенно свободно могут использоваться* в системе, не имеют никакой отличительной роли.

Глобальные переменные делятся на две группы:

Глобальные переменные от #100 до #499 удаляются при выключении.

Значение глобальных переменных от #500 до #599 сохраняется и после выключения.

Для макропеременных от #500 до #999 можно делать защиту от записи с помощью параметров N1702 Write Prt Low и N1703 Write Prt Hig.

На параметр N1702 Write Prt Low запишем первый элемент блока, желаемый защитить от записи, а на параметр N1703 Write Prt Hig - последний элемент блока, желаемый защитить от записи.

Например, если желаемый защитить от записи переменные от #530 до #540, надо установить параметр N1702 Write Prt Low=530 и N1703 Write Prt Hig=540..

Если устройство ЧПУ обращается несколькими каналами, *глобальные переменные по блокам* с помощью параметра можно делать доступным во всех каналах.

Параметром N1700 No. of Common #100 задаётся число глобальных макропеременных от #100 до #499, вызываемых из каждого канала. В каждом канале общими макропеременными будут макропеременные от номера 100 до 100 + No. of Common #100. Этот параметр должен быть меньше, чем 400.

Если значение параметра например, 40, макропеременные от #100 до #139 будут общими для всех каналов.

Параметром N1701 No. Of Common #500 задаётся число глобальных макропеременных от #500 до #999, вызываемых из каждого канала. В каждом канале общими макропеременными-

ми будут макропеременные от номера 500 до 500 + No. of Common #500. Этот параметр должен быть меньше, чем 500.

Если значение параметра например, 30, макропеременные от #500 до #529 будут общими для всех каналов.

23.1.5 Обозначения, используемые при описании системных переменных

Системные переменные представляют собой такие внутренние данные NC, которые можно прочитать, или можно изменить из программы деталей.

На макропеременных с опознавательным номером до 10000, относящихся к осям, можно опознавать оси только от 1 до 20. Для случаев высоких номеров осей были введены опознавательные номера выше 100000. По этим номерам от 1 до 50 можно ссылаться на оси. Например:

#100001 позицию конца кадра 1-й оси

можно опознать и с номером выше. Естественно, можно ссылаться на данные выше и с номерами под 10000 и символами также.

Обозначения, используемые при описании системных переменных, следующие:

[n]: индекс переменной. Может быть например, номер оси, или номер шпинделя

R: свойство переменной: переменная только для чтения

W: свойство переменной: переменная только для писания

R/W: свойство переменной: переменная для чтения и писания.

23.1.6 Пустая переменная. Постоянные

Номер	Символ	Свой-ство	Описание
#0, #3100	#_EMPTY	R	“пустая” постоянная
#3101	#_PI	R	$\pi = 3.14159...$
#3102	#_E	R	Натуральное число: $e = 2.71828...$

Пустая переменная #0, #3100, #_EMPTY (R)

При макровывозе, если какому-то адресу не присвоили значение, в корне макроса значение локальной переменной, относящейся к данному адресу, пустое будет.

Например:

G65 P100 X20 Y30

после вызова в макросе O0100 значение локальной переменной #1 пустое будет, так как в вызове G65 для адреса “A” не присвоили значение. В корне макроса проверкой

#1 EQ #0

можно решать, что при макровывозе заполнили ли адрес “A”

Пустая переменная и число 0 не то же самое, они разные!

Сравнив свойство пустой переменной тем, если значение какой-то переменной 0:

Ссылка на **пустую** переменную в адресе:

если #1=<пустая>	если #1=0
G90 X20 Y#1	G90 X20 Y#1
G90 X20	G90 X20 Y0

Пустая переменная в команде *присвоения значения*:

если #1=<пустая>	если #1=0
#2=#1	#2=#1
#2=<пустая>	#2=0
#2=#1*3	#2=#1*3
#2=0	#2=0
#2=#1+#1	#2=#1+#1
#2=0	#2=0

В случае *проверки условий* разница между значением переменной **пустая** и **0**:

если #1=<пустая>	если #1=0
#1 EQ #0	#1 EQ #0
исполнен	не выполнен
#1 NE 0	#1 NE 0
исполнен	не выполнен
#1 GE #0	#1 GE #0
исполнен	не выполнен
#1 GT 0	#1 GT 0
исполнен	не выполнен

23.1.7 Переменные между программой деталей и программой PLC

Обмен между программой деталей и программой PLC можно осуществлять через следующие переменные.

☞ **Внимание!** Строителем станка определяется, что какие информации передаёт программа PLC для программы деталей через различные системные переменные, а также какие информации принимает от программы деталей.

Номер	Символ	Свой- ство	Описание
#1000...#1031	#_UI[n] n=0 - 31	R	32 шт. <i>битные переменные</i> , переданные от программы PLC для NC. Область значений: 0, 1
#1032	#_UIL[n] n=0	R	32 шт. битные переменные, переданные от программы PLC для NC, переданное как одно 32 битное целое число. Область значений: 0 ... $2^{32}-1$
#1033...#1035	#_UIL[n] n=1, 2, 3	R	Переданные от программы PLC для NC 3 шт. переменные <i>с плавающей запятой</i> .
#1100...#1131	#_UO[n] n=0 - 31	R/W	Переданные от программы деталей для PLC 32 шт. <i>битные переменные</i> . Область значений: 0, 1
#1132	#_UOL[n] n=0	R/W	32 шт. битные переменные, переданные от программы деталей для PLC, переданное как одно, 32 битные целое число. Область значений: 0 ... $2^{32}-1$
#1133...#1135	#_UOL[n] n=1, 2, 3	R/W	Переданные от программы деталей для PLC 3 шт. переменные <i>с плавающей запятой</i> .

23.1.8 Сообщения программы деталей

Из макропрограммы можно выводить сообщение об ошибке, далее можно послать сообщение для оператора:

Номер	Символ	Свой-ство	Описание
#3000	#_ALM	W	Сообщение об ошибке, удалить клавишей Cancel
#3006	#_MSGSTP	W	Останов с сообщением, после Start идёт дальше
#3106	#_MSG	W	Выписывание сообщения в окно списка программ
#3107	#_MSGBOX	R/W	Посылка сообщения в окно Windows

Сообщение об ошибке: #3000, #_ALM (W)

Можно выводить сообщение об ошибке

#3000=nnn(СООБЩЕНИЕ ОБ ОШИБКЕ)

или отмеченное присвоением числового значения

#_ALM=nnn(СООБЩЕНИЕ ОБ ОШИБКЕ)

(nnn: не более трёх цифр), и/или текстовое сообщение об ошибке. Текст надо заключить в круглые скобки (,).

Если программа обнаруживает ошибку в макросе, то есть набегит на такую ветку, где для переменной #3000 присвоили значение, программа выполняется до предшествующего кадра, затем отложится выполнение и на экране появится сообщение об ошибке, заданное в скобках, а также код сообщения в форме

ii4nnn00 (ii: номер канала, в котором ошибка обнаружилась),

то есть к заданному значению #3000 числу nnn добавится 4000. Если не задавали число, код сообщения будет 4000, если не задавали текст, появится только код. Сообщение об ошибке прекращается клавишей CANCEL.

Останов с сообщением: #3006, #_MSGSTP (W)

Под действием присвоения значения

#3006=nnn(СООБЩЕНИЕ)

или

#_MSGSTP=nnn(СООБЩЕНИЕ)

выполнение программы остановится, и на экране появится сообщение, заключённое в круглые скобки, а также код сообщения в форме

ii5nnn00 (ii: номер канала, в котором обнаружилась ошибка),

то есть к заданному значению #3006 числу nnn добавится 5000. Если не задавали число, код сообщения будет 5000, если не задавали текст, появится только код. Выполнение программы продолжается нажав клавишу START, при этом сообщение удаляется с экрана. Команду полезно использовать в таком случае, когда во время выполнения программы потребуется вмешательство оператора.

Выписывание сообщения в окно Каталог программ: #3106, #_MSG (W)

Под действием присвоения значения

```
#3106=nnn(СООБЩЕНИЕ)
```

или

```
#_MSG=nnn(СООБЩЕНИЕ)
```

непрерывно продолжается программа, и появится *текст сообщения в самой верхней строке окна Каталог программ* следующим образом:

```
MSGnnn: (СООБЩЕНИЕ)
```

Текст сообщения остаётся там до тех пор, пока следующей командой #3106, или #_MSG не изменяем её. Под действием RESET, конец программы (M30) сообщение удаляется. Для появления сообщения в самой верхней строке каталога программ, *надо разрешить выписывание* пользованием функциональных клавиш, следующим образом:

F5 Вид - F1 Каталог программ - F9 Установки - F3 #_информация MSG

Можно его использовать например, для указания выполнения фрагментов программ:

```
...  
#3106=1 (черновая обработка фрезой D30)  
...  
...  
#_MSG=2 (чистовая обработка фрезой D20)  
...
```

Посылка сообщений в окно Windows: #3107, #_MSGBOX (R/W)

Под действием присвоения значения

```
#3107=nnn(СООБЩЕНИЕ)
```

или

```
#_MSGBOX=nnn(СООБЩЕНИЕ)
```

остановится выполнение программы (будет состояние STOP) и в строке сообщений управления появится текст сообщения, заключено в круглые скобки, а также код сообщения, в форме

iiбnnnjj (ii: номер канала, в котором выдавали сообщение).

Выписывание сообщения появится не только в верхней полосе штата, но и в середине экрана в *окне сообщений Windows*.

Тип окна сообщений определяется кодом сообщения, Если присвоение значений между **nnn=100 - 199**, тогда окно сообщения появится **клавишей „Ok”**, и оператор может только принимать выписанное сообщение. Если присвоение значений между

nnn=200 - 299, тогда окно сообщения появится **клавишей „Да” „Нет”**.

В этом случае оператор имеет возможность выбирать. При любом ином присвоении значений управление выводит сообщение об ошибке.

После ответа на сообщения, появившиеся в окне Windows, сообщение удаляется и переменная #3107, может принимать следующие значения в различных случаях:

```
#3107=0: в окне щёлкнули на клавишу X (закреть), или нажали клавишу  
        CANCEL,  
#3107=1: щёлкнули на клавишу „OK”,  
#3107=2 : щёлкнули на клавишу „Да”,  
#3107=3 : щёлкнули на клавишу „Нет”,
```

Если при выписывании сообщений в окне сообщений желаем изобразить данный фрагмент сообщения в нескольких строках, имеется возможность расчленить текст *символом „\n”* и таким образом вставить символ перемены строки.

При выписывании значений можно применить *форматирование макропеременных, описанное при команды DPRNT*.

1-й пример:

```
##3107=100 (Значения результата измерения\nX длина:
#140[53]\nY длина: #141[53])
```

2-й пример

```
#140=0.3458
#141=0.9123
(измерение)
(выписывание результата расчёта коррекции)
#_MSGBOX=200 (Значения износа инструмента\nX износ по диамет-
ру: #140[53]\nZ износ по направлению: #141[53]\n Желаете
записать новую коррекцию?)
IF [[#_MSGBOX] EQ 2] GOTO10
GOTO20
N10 (ветвь Да)
(запись износа)
N20
(продолжение программы)
```

23.1.9 Времена, счётчики заготовок

Номер	Символ	Свой-ство	Описание
#3001	#_CLOCK1	R/W	Секундомер в мсек-ах, свободного использования
#3002	#_CLOCK2	R/W	Время точения (мсек)
#3011	#_DATE	R	Дата: год/месяц/день
#3012	#_TIME	R	Часы: час/минута/секунда
#3901	#_PRTSA	R/W	Число изготовленных заготовок
#3902	#_PRTSN	R/W	Число изготавливаемых заготовок

Секундомер в миллисекундах: #3001, #_CLOCK1 (R/W)

Значение переменной можно писать и читать.

Можно измерить время, прошедшее между двумя моментами времени в миллисекундах. Значение переменной при включении устройства ЧПУ запускается с нуля, и считает в сторону нарастания. Всегда считает, когда устройство ЧПУ включено. Можно сводить к нулю с присвоением значения, из программы

```
#3001=0,
```

или

#_CLOCK1=0

командой переменную, которая при этом запускается с 0, а потом можно запросить его значение из программы

#100=#3001

или командой

#100=#_CLOCK1.

Время течения: #3002, #_CLOCK2 (R/W)

Значение переменной можно писать и читать. Измеряется время, проведённое в автоматическом режиме, в состоянии старта, подачей (G1, G2, и т.д.) В миллисекундах, скопленное с начала жизни устройства ЧПУ. Значение переменной можно читать на индикаторе Время/Индикатор счётчиков /Время течения.

Дата: #3011, #_DATE (R)

Из переменной можно читать текущую дату в формате год/месяц/день.

После использования команды

#100=#3011,

или

#100=#_DATE,

если значение переменной #100 например

20140518,

это означает, что дата: 2014 (год), 05 (месяц), 18 (18-ое число).

Часы: #3012, #_TIME (R)

Из переменной можно читать текущее время в формате час/минута/секунда.

После использования команды

#100=#3012,

или

#100=#_TIME

если значение переменной #100 например

153241,

это означает, что время: 15 ч (3ч п/о.), 32 мин, 41 сек.

Число изготовленных/изготавливаемых заготовок: #3901, #_PRTSA / #3902, #_PRTSN (R/W)

Значения переменных можно писать и читать. Число изготовленных заготовок собирает устройство ЧПУ в #3901–й #_PRTSA счётчике. К содержанию счётчика после выполнения каждой функции M02, M30, или после выполнения каждой функции M, выделённой параметром N2305 Part Count M добавит 1.

Когда число изготовленных заготовок достигает число изготавливаемых заготовок (#3902–й #_PRTSN счётчик) исполнитель сообщает об этом PLC.

число изготовленных заготовок #3901, #_PRTSA

число изготавливаемых заготовок #3902, #_PRTSN

Значение числа изготовленных и изготавливаемых заготовок можно читать на индикаторе Время/Индикатор счётчиков Изготовленных и Изготавливаемых.

23.1.10 Переменные, действующие на работу автоматического режима

Номер	Символ	Свой-ство	Описание
#3003	#_CNTL1	R/W	1-я контрольная переменная (покадровая)
#3003 бит 0	#_M_SBK	R/W	Подавление покадрового выполнения. Область значений: 0, 1
#3004	#_CNTL2	R/W	2 -я контрольная переменная
#3004 бит 0	#_M_FHD	R/W	Подавление Stop-а. Область значений: 0, 1
#3004 бит 1	#_M_OV	R/W	Подавление Override и Stop-а. Область значений: 0, 1
#3004 бит 2	#_M_EST	R/W	Подавление точного останова. Область значений: 0, 1

☞ **Внимание:** Значения битов, установленных на контрольных переменных, удаляются на reset и в конце программы!

1-я контрольная переменная: #3003, #_CNTL1 (R/W)

Если значение переменной #3003, или #_CNTL1 будет 1 (или нечётное число), в состоянии выполнения покадров не остановится после выполнения одного кадра до тех пор, пока значение этой переменной не будет 0.

На подавление покадрового выполнения можно сослать битом (заданием 0, или 1) и написанием переменной

#_M_SBK .

Значение переменной после включения, на reset будет 0.

Покадровое выполнение, если значение переменной

0: не подавлено

1: подавлено.

2-я контрольная переменная: #3004, #_CNTL2 (R/W)

Переменной #3004, или #_CNTL2 можно задавать следующие значения:

Значение #3004, или #_CNTL2	Подаление точного останова G61, G9	Подавление Override и Stop	Подавление клавиши Stop
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

Режим точного останова, клавиша override и stop, если число, записанное в соответствующий столбец

0: не подавлено

1: подавлено.

Подавления, перечисленные выше можно задавать записыванием 0, или 1 на битные переменные:

#_M_FHD: подавление клавиши stop:

0: не подавлено

1: подавлено.

Не подавит покадровый режим.

#_M_OV: подавление override и stop (состояние G63):

0: не подавлено

1: подавлено.

Подавит и покадровый режим

#_M_EST: подавление Точного останова G61, G9 (состояние G64)

0: не подавлено

1: подавлено.

23.1.11 Запрос состояний поиска кадра и состояний теста

Номер	Символ	Свой- ство	Описание
	#_CNTBS	R	Запрос состояний поиска кадра и состоя- ний теста
bit 0	#_BSEARCH	R	Состояние поиска кадра
bit 1	#_TEST	R	Режим теста
bit 2	# MCHLOCK	R	Режим станок закрыт

Запрос состояний поиска кадра и состояний теста: # CNTBS (R)

По переменной # CNTBS можно запросить заодно все три состояния.

Возможно и бинарный запрос по переменной # CNTBS:

BSEARCH: устройство ЧПУ выполняет поиск кадра

=0: не выполняет поиск кадра,

=1: поискывает кадр.

TEST: режим теста

=0: нет режима теста,

=1: устройство ЧПУ находится в режиме теста.

```
# MCHLOCK: станок закрыт
```

$=0$: станок не закрыт,

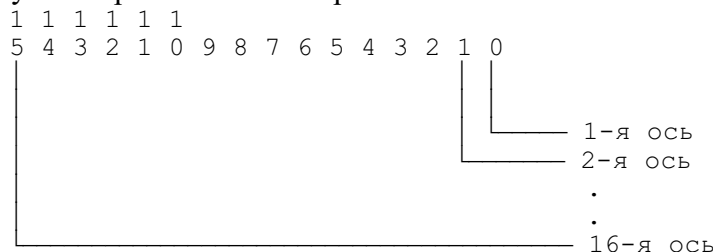
=1: станок закрыт.

23.1.12 Состояние отражения

Номер	Символ	Свой-ство	Описание
#3007	# MIRIMG	R	Состояние отражения

Прочтением переменной #3007 можно установить, что на какой физической оси имеется зарегистрированная действующая команда отражения. Переменная только для чтения.

Истолкуя бинарно значение переменной:



Значение отдельных бит:

0: нет отражения

1: отражение включено

Если например, значение переменной 5, то отражение включено на 1-й и на 3-й оси. Номер оси означает физический номер оси, параметром определяется, что к какому физическому

номеру оси какая по имени ось принадлежит.

23.1.13 Номер главной программы

Номер	Символ	Свой- ство	Описание
#4000	#_MAIN0	R	Номер выполняемой главной программы

Показывает всегда номер актуальной главной программы, выполняемой в данный момент, даже тогда, если как раз осуществляется выполнение какой-то подпрограммы. Если прервать автоматический режим и в MDI выполняем одну программу, то показывает номер программы, выполненной в MDI.

Если имя главной программы начинается не с буквы O и состоит не из максимально 8 цифр, #4000=0 будет.

23.1.14 Модальные информации

Модальные информации, действующие в предшествующем кадре, определяются чтением системных переменных #4001–#4199.

Номер	Символ	Свой-ство	Описание
#4001...#4039	#_BUFG[n] n=1-39	R	Код модальной функции G, заданной последней по группам. n: номер группы кода G
#4101	#_BUFA	R	Значение вспомогательной функции, заданной последней, адрес которой определено параметром N1333 Aux Fu Addr1.
#4102	#_BUFB	R	Значение вспомогательной функции, заданной последней, адрес которой определено параметром N1334 Aux Fu Addr2.
#4103	#_BUFC	R	Значение вспомогательной функции, заданной последней, адрес которой определено параметром N1334 Aux Fu Addr3.
#4107	#_BUFD	R	Значение адреса D, заданного последним
#4108	#_BUFE	R	Значение адреса E, заданного последним
#4109	#_BUFF	R	Значение адреса F, заданного последним
#4111	#_BUFH	R	Значение адреса H, заданного последним
#4113	#_BUFM	R	Значение адреса M, заданного последним
#4114	#_BUFN	R	Значение номера кадра N, заданного последним
#4115	#_BUFO	R	Значение номера программы O, вызванной последней
#4119	#_BUFS	R	Значение адреса S, заданного последним
#4120	#_BUFT	R	Значение адреса T, заданного последним
#4130	#_BUFWZP	R	Номер расширенной системы координат (G54.1 Pn), заданной последней

Модальные информации, действующие в кадре, находящемся в настоящий момент **под выполнением** определяются чтением переменных #4201–4399.

Номер	Символ	Свой- ство	Описание
#4201...#4239	#_ACTG[n] n=1-39	R	Код модальной функции G, действующей в кадре в момент его выполнения, по группам. n: номер группы кода G
#4301	#_ACTA	R	Значение вспомогательной функции, действующей в кадре в момент его выполнения, адрес которой определён параметром N1333 Aux Fu Addr1.
#4302	#_ACTB	R	Значение вспомогательной функции, действующей в кадре в момент его выполнения, адрес которой определён параметром N1334 Aux Fu Addr2.
#4303	#_ACTC	R	Значение вспомогательной функции, действующей в кадре в момент его выполнения, адрес которой определён параметром N1334 Aux Fu Addr3.
#4307	#_ACTD	R	Значение адреса D, действующего в кадре в момент его выполнения
#4308	#_ACTE	R	Значение адреса E, действующего в кадре в момент его выполнения
#4309	#_ACTF	R	Значение адреса F, действующего в кадре в момент его выполнения
#4311	#_ACTH	R	Значение адреса H, действующего в кадре в момент его выполнения
#4313	#_ACTM	R	Значение адреса M, действующего в кадре в момент его выполнения
#4314	#_ACTN	R	Значение номера кадра N, действующего в кадре в момент его выполнения
#4315	#_ACTO	R	Значение номера программы O, действующей в кадре в момент его выполнения
#4319	#_ACTS	R	Значение адреса S, действующего в кадре в момент его выполнения

Номер	Символ	Свой- ство	Описание
#4320	#_ACTT	R	Значение адреса T, действующего в кадре в момент его выполнения
#4330	#_ACTWZP	R	Номер расширенной системы координат (G54.1 Pn), действующей в кадре в момент его выполнения

Модальные информации, действующие в прерванном кадре в момент вызова макроса прерывания, определяются чтением переменных #4401–4599.

Номер	Символ	Свой- ство	Описание
#4401...#4439	#_INTG[n] n=1-39	R	Код модальной функции G, действующей в прерванном кадре, по группам. n: номер группы кода G
#4501	#_INTA	R	Значение вспомогательной функции, действующей в прерванном кадре, адрес которой определено параметром N1333 Aux Fu Addr1
#4502	#_INTB	R	Значение вспомогательной функции, действующей в прерванном кадре, адрес которой определено параметром N1334 Aux Fu Addr2.
#4503	#_INTC	R	Значение вспомогательной функции, действующей в прерванном кадре, адрес которой определено параметром N1334 Aux Fu Addr3.
#4507	#_INTD	R	Значение адреса D, действующего в прерванном кадре
#4508	#_INTE	R	Значение адреса E, действующего в прерванном кадре
#4509	#_INTF	R	Значение адреса F, действующего в прерванном кадре
#4511	#_INTH	R	Значение адреса H, действующего в прерванном кадре
#4513	#_INTM	R	Значение адреса M, действующего в прерванном кадре

Номер	Символ	Свой-ство	Описание
#4514	#_INTN	R	Значение номера кадра N, действующего в прерванном кадре
#4515	#_INTO	R	Значение номера программы O, действующей в прерванном кадре
#4519	#_INTS	R	Значение адреса S, действующего в прерванном кадре
#4520	#_INTT	R	Значение адреса T, действующего в прерванном кадре
#4530	#_INTWZP	R	Номер расширенной системы координат (G54.1 Pn), действующей в прерванном кадре

☞ **Внимание!** В случае модальных кодов G, содержащих точку, например, G51.1, вернёт код без точки так, что номер после точки напишет впереди:

#100=#4022

под действием команды значение #100 будет 151 в состоянии G51.1 и будет 150 в состоянии G50.1.

Систематизацию модальных кодов G по номеру групп показывает следующая таблица. Одноразовые, не модальные коды G относятся в 0-ую группу. 0-ую группу нельзя запрашивать, поэтому коды G, относящиеся туда, не приведенные в таблицу.

Номер группы	Модальные коды G, относящиеся в группу	Функция
1	G0, G1, G2, G3, G33	интерполяция
2	G17, G18, G19	выбор плоскости
3	G90, G91	абсолютное/инкрементное программирование
4	G22, G23	запрещённая зона вкл./выкл.
5	G94, G95	подача поминутная/за оборот
6	G20, G21	задание данных дюйм/метрическое
7	G40, G41, G42	коррекция радиуса вкл./выкл.
8	G43, G44, G43.7 G49	коррекция по длине вкл./выкл.
9	G73, G74, G76, G80, G81, G82, G83, G84, G84.2, G84.3, G85, G86, G87, G88, G89	циклы сверления

Номер группы	Модальные коды G, относящиеся в группу	Функция
10	G98, G99	возвращение из сверлильного цикла в исходную точку/точку R
11	G50, G51	масштабирование вкл./ывкл.
12	G66, G66.1, G67	модальный макровывоз вкл./выкл.
13	G96, G97	постоянная скорость резания вкл./выкл.
14	G54, G54.1, G54.2, G55, G56, G57, G58, G59	выбор системы координат заготовки
15	G61, G62, G63, G64	функции управления подачей
16	G68, G69	плоскостный поворот вкл./выкл.
17	G15, G16	задание данных полярной координатой вкл./выкл.
18		
19		
20		
21	G12.1, G13.1	полярная интерполяция вкл./выкл.
22	G50.1, G51.1	отражение вкл./выкл.
23		
24		
25		
26		
27		
28		
29		
30		
31	G50.2, G51.2	точение многоугольника вкл./выкл.
32		
33		
34	G80.8, G81.8	электронный привод вкл./выкл.

Номер группы	Модальные коды G, относящиеся в группу	Функция
35		
36		
37		
38		
39		

23.1.15 Информации о позиции

Номер	Символ	Свой-ство	Описание
#5001...#5020	#_ABSIO[n] n=1-50	R	Позиция конца кадра оси 1...n
#100001...#100050			
#5021...#5040	#_ABSMT[n] n=1-50	R	Позиция оси 1...n в станочной системе координат
#100051...#100100			
#5041...#5060	#_ABSOT[n] n=1-50	R	Позиция оси 1...n в системе координат заготовки
#100101...#100150			
#5061...#5080	#_ABSKP[n] n=1-50	R	Щупающая позиция оси 1...n в системе координат заготовки
#100151...#100200			

☞ **Внимание!** Чтение информации о позиции происходит на основании номера оси. Прежде чем использовать команду, спрашивайте разъяснение от строителя станка, что какая ось какой номер имеет.

Например, берём двуканальное устройство ЧПУ. В первом канале имеются оси X, Y, Z, во втором оси X, Z. Строитель станка распределил оси следующим образом:

1-й канал:

X – 1-я ось

Y – 2-я ось.

Z – 3-я ось

2-й канал:

X – 4-я ось

Z – 5-я ось

Если в 1-ом канале желаем запрашивать информации о позиции, по оси Z надо использовать индекс 3, если во втором - индекс 5.

Позиция осей в конце кадра: #5001...#5020, #100001...#100050, #_ABSIO[n] (R)

Позиция осей в конце кадра возвращается

в актуальной системе координат заготовки (G54, G55, ...),

в прямоугольных координатах,

не принимая во внимание все коррекции (длина, радиус).

Станочная позиция осей: #5021...#5040, #100051...#100100, #_ABSMT[n] (R)

Станочная позиция осей возвращается

в системе координат станка (G53),

Если данная ось перемещается, возвращённое значение постоянно меняется.

Позиция осей к заготовке: #5041...#5060, #100101...#100150, #_ABSOT[n] (R)

Позиция осей к заготовке возвращается

в актуальной системе координат заготовки (G54, G55, ...),

в прямоугольных координатах,

принимая во внимание все коррекции (длина, радиус).

Если данная ось перемещается, возвращённое значение постоянно меняется.

Щупающие позиции: #5061...#5080, #100151...#100200, #_ABSKP[n] (R)

В кадре G31 позиция, где поступил сигнал щупа, возвращается

в актуальной системе координат заготовки (G54, G55, ...),

в прямоугольных координатах,

принимая во внимание все коррекции (длина, радиус).

Если сигнал щупа не поступил, указанные выше переменные принимают позицию конечной точки, запрограммированную в кадре G31.

После поступления сигнала щупа ось замедляет и остановится. Позиция конечной точки #_ABSIO(n) кадра G31 принимает значение, равное значению после замедления и останова.

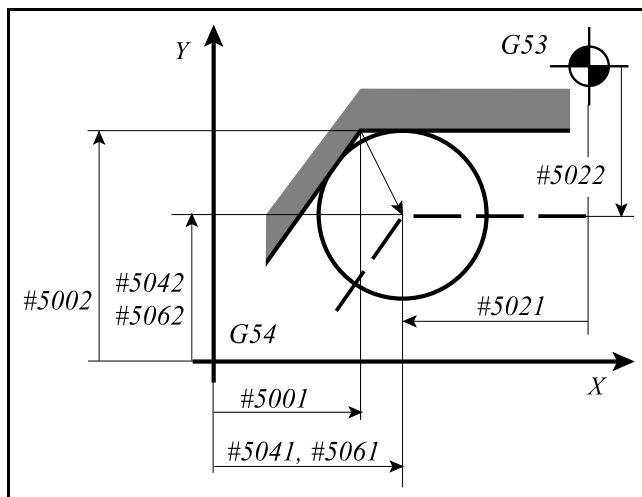


рисунок 23.1.15-1

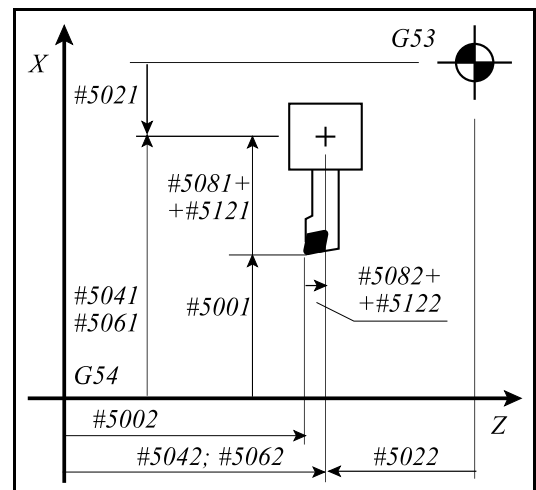


рисунок 23.1.15-2

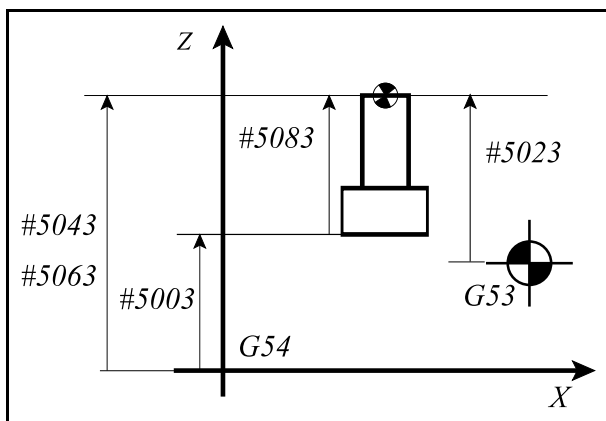


рисунок 23.1.15-3

23.1.16 Значение актуальной коррекции по длине

Номер	Символ	Свой-ство	Описание
#5081	#_TOFSWX	R	Значение коррекции по износу, учтённое на оси X
#100201			
#5082	#_TOFSWZ	R	Значение коррекции по износу, учтённое на оси Z
#100202			
#5083	#_TOFSWY	R	Значение коррекции по износу, учтённое на оси Y
#100203			
#5121	#_TOFSGX	R	Значение геометрической коррекции по длине, учтённое на оси X
#100901			
#5122	#_TOFSGZ	R	Значение геометрической коррекции по длине, учтённое на оси Z
#100902			
#5123	#_TOFSGY	R	Значение геометрической коррекции по длине, учтённое на оси Y
#100903			

Из перечисленных выше переменных можно прочесть значение коррекции по длине (геометрическое + износ), учтённое на X, Z и Y в кадре в момент выполнения

23.1.17 Прочие информации о позиции

Номер	Символ	Свойство	Описание
#5101...#5120	#_SVERR[n] n=1-50	R	Отставание оси 1...n
#100251...#100300			
#100651...#100700	#_MIRTP[n] n=1-50	R	Позиция, смещённая маховичком на оси 1...n
#5181...#5200			
#100801...#100850	#_DIST[n] n=1-50	R	Значение остаточного хода на оси 1...n

Отставание осей: #5101...#5120, #100251...#100300, #_SVERR[n] (R)

Из переменных можно прочесть отставание сервоконтуров осей (ошибка слежения), в вводных единицах измерения.

Позиции, смещённые маховичком: #100651...#100700, #_MIRTP[n] (R)

Если в автоматическом режиме, хоть во время движения, перемещением маховичка поправляем позицию осей (нулевую точку), меру коррекции можно прочесть из переменных выше, по осям, в вводных единицах измерения.

Значение остаточного хода: #5181...#5200, #100801...#100850, #_DIST[n] (R)

Из переменных можно прочесть по осям значение остаточного хода, то есть, что в кадре в момент выполнения сколько хода осталось переди до конечной позиции, в вводных единицах измерения. Это есть то значение, которое устройство ЧПУ выводит на экран позиции для остаточного хода.

23.1.18 Значения накопителя коррекции инструментов

Номер	Символ	Свой-ство	Описание
#10001...#10999	#_OFSXW[n] n=1...999	R/W	Значение износа коррекции по длине в направлении X. n=1...999: номер ячейки по коррекции
#15001...#15999	#_OFSXG[n] n=1...999	R/W	Геометрическое значение коррекции по длине в направлении X. n=1...999: номер ячейки по коррекции
#14001...#14999	#_OFSYW[n] n=1...999	R/W	Значение износа коррекции по длине в направлении Y. n=1...999: номер ячейки по коррекции
#19001...#19999	#_OFSYG[n] n=1...999	R/W	Геометрическое значение коррекции по длине в направлении Y. n=1...999: номер ячейки по коррекции
#11001...#11999	#_OFSZW[n] n=1...999	R/W	Значение износа коррекции по длине в направлении Z. n=1...999: номер ячейки по коррекции
#16001...#16999	#_OFSZG[n] n=1...999	R/W	Геометрическое значение коррекции по длине в направлении Z. n=1...999: номер ячейки по коррекции
#12001...#12999	#_OFSRW[n] n=1...999	R/W	Значение износа коррекции радиуса. n=1...999: номер ячейки по коррекции
#17001...#17999	#_OFSRG[n] n=1...999	R/W	Геометрическое значение коррекции радиуса. n=1...999: номер ячейки по коррекции
#13001...#13999	#_OFST[n] n=1...999		Код позиции инструмента. n=1...999: номер ячейки по коррекции

С помощью перечисленных выше макропеременных можно присвоить значение для всех ячеек по коррекции, далее все ячейки по коррекции можно прочитать из программы.

23.1.19 Смещение нулевой точки заготовкиЗначения смещений нулевой точки

Номер	Символ	Свой-ство	Описание
#5201...#5220	#_WZCMN[n] n=1-50	R/W	Общее смещение нулевой точки заготовки по осям
#100301...#100350			

Номер	Символ	Свой- ство	Описание
#5221...#5240	#_WZG54[n] n=1-50	R/W	Смещение нулевой точки заготовки G54 по осям
#100351...#100400			
#5241...#5260	#_WZG55[n] n=1-50	R/W	Смещение нулевой точки заготовки G55 на оси 1...n
#100401...#100450			
#5261...#5280	#_WZG56[n] n=1-50	R/W	Смещение нулевой точки заготовки G56 на оси 1...n
#100451...#100500			
#5281...#5300	#_WZG57[n] n=1-50	R/W	Смещение нулевой точки заготовки G57 на оси 1...n
#100501...#100550			
#5301...#5320	#_WZG58[n] n=1-50	R/W	Смещение нулевой точки заготовки G58 на оси 1...n
#100551...#100600			
#5321...#5340	#_WZG59[n] n=1-50	R/W	Смещение нулевой точки заготовки G59 на оси 1...n
#100601...#100650			

С помощью перечисленных выше макропеременных можно присвоить значение для всех смещений нулевой точки по осям, далее все смещения нулевой точки можно прочесть из программы по осям.

Значения основных поворотов

Номер	Символ	Свой- ство	Описание
#120001	#_WRG54[1]	R/W	Основной поворот G54 в плоскости XY
#120002	#_WRG54[2]	R/W	Основной поворот G54 в плоскости ZX
#120003	#_WRG54[3]	R/W	Основной поворот G54 в плоскости YZ
#120011	#_WRG55[1]	R/W	Основной поворот G55 в плоскости XY
#120012	#_WRG55[2]	R/W	Основной поворот G55 в плоскости ZX
#120013	#_WRG55[3]	R/W	Основной поворот G55 в плоскости YZ
#120021	#_WRG56[1]	R/W	Основной поворот G56 в плоскости XY
#120022	#_WRG56[2]	R/W	Основной поворот G56 в плоскости ZX
#120023	#_WRG56[3]	R/W	Основной поворот G56 в плоскости YZ
#120031	#_WRG57[1]	R/W	Основной поворот G57 в плоскости XY

Номер	Символ	Свойство	Описание
#120032	#_WRG57[2]	R/W	Основной поворот G57 в плоскости ZX
#120033	#_WRG57[3]	R/W	Основной поворот G57 в плоскости YZ
#120041	#_WRG58[1]	R/W	Основной поворот G58 в плоскости XY
#120042	#_WRG58[2]	R/W	Основной поворот G58 в плоскости ZX
#120043	#_WRG58[3]	R/W	Основной поворот G58 в плоскости YZ
#120051	#_WRG59[1]	R/W	Основной поворот G59 в плоскости XY
#120052	#_WRG59[2]	R/W	Основной поворот G59 в плоскости ZX
#120053	#_WRG59[3]	R/W	Основной поворот G59 в плоскости YZ

С помощью перечисленных выше макропеременных можно задавать поворот в произвольной главной плоскости (но только в одной) по системам координат заготовки, где точка центра поворота является точкой начала системы координат.

Значения расширенных смещений нулевой точки

Номер	Символ	Свойство	Описание
#7001...#7020	#_WZP1[n] n=1-50	R/W	Смещение нулевой точки заготовки G54.1 P1 на оси 1...n
#101001...#101050			
#7021...#7040	#_WZP2[n] n=1-50	R/W	Смещение нулевой точки заготовки G54.1 P2 на оси 1...n
#101051...#101100			
...	...	...	...
#7941...#7960	#_WZP48[n] n=1-50	R/W	Смещение нулевой точки заготовки G54.1 P48 на оси 1...n
#103351...#103400			
#103401...#103450	#_WZP49[n] n=1-50	R/W	Смещение нулевой точки заготовки G54.1 P49 на оси 1...n
#103451...#103500			
...	...	...	...

С помощью перечисленных выше макропеременных можно присвоить значение для всех нулевых смещений по осям, далее все смещения нулевой точки можно прочитать из программы.

Значения расширенных основных поворотов

Номер	Символ	Свой-ство	Описание
#120061	#_WRP1[1]	R/W	Основной поворот G54.1 P1 в плоскости XY
#120062	#_WRP1[2]	R/W	Основной поворот G54.1 P1 в плоскости ZX
#120063	#_WRP1[3]	R/W	Основной поворот G54.1 P1 в плоскости YZ
#120071	#_WRP2[1]	R/W	Основной поворот G54.1 P2 в плоскости XY
#120072	#_WRP2[2]	R/W	Основной поворот G54.1 P2 в плоскости ZX
#120073	#_WRP2[3]	R/W	Основной поворот G54.1 P2 в плоскости YZ
...	...	...	...

С помощью перечисленных выше макропеременных можно задавать поворот в произвольной главной плоскости (но только в одной) по каждой расширенной системе координат заготовки, где точка центра поворота является точкой начала системы координат.

Значения динамических смещений нулевой точки

Значения таблицы динамических смещений нулевой точки с помощью переменных можно писать и читать:

Номер	Символ	Свой-ство	Описание
#5521...#5540	#_FOFS1[n] n=1-50	R/W	Динамическое смещение нулевой точки G54.2 P1 по осям
#117051...#117100			
#5541...#5560	#_FOFS2[n] n=1-50	R/W	Динамическое смещение нулевой точки G54.2 P2 по осям
#117101...#117150			
...	...		...
#5661...#5680	#_FOFS8[n] n=1-50	R/W	Динамическое смещение нулевой точки G54.2 P8 по осям
#117401...#117450			

Номер актуальных динамических смещений нулевой точки

Во время прогона программы номер действующих динамических смещений нулевой точки можно прочесть с помощью следующей команды:

Номер	Символ	Свой-ство	Описание
#5500	#_FOFSP	R	Номер актуальных динамических смещений нулевой точки (G54.2 Pp P=1-8)
#117000			

Во время прогона программы значение актуальных динамических смещений нулевой точки можно прочесть с помощью следующих переменных. Эти значения не тождественны значениям, записанным в таблицу динамических смещений нулевой точки, а они представляют собой (повернутые) значения, поправленные на основании мгновенной позиции столов.

Номер	Символ	Свой-ство	Описание
#5501...#5520	#_FOFSVAL[n] n=1-50	R	Значение актуальных динамических смещений нулевой точки по осям
#117001...#117050			

23.1.20 Прочтение данных инструмента, имеющихся в шпинделе и в дежурных магазинах

Номер	Символ	Свой-ство	Описание
#8400		R/W	Адрес шпинделя, или дежурного магазина

На макропеременной

#8400 (10, 11, 20, 21, ... можно писать, читать)

надо задавать, что в каком шпинделе, или дежурном магазине имеются инструменты, для которых желаем прочесть данные по обращению инструментами через макропеременные #8401, #8402, Определению поддаются только номера магазинов, заданных в скобках. Если на станке имеется несколько шпинделей, или дежурных магазинов, за информацию обращайтесь к строителю станка.

Если на станке имеется только один шпиндельный магазин, а дежурного магазина нет, для макропеременной #8400 не надо присвоить значение, макропеременные #8401, ... относятся всегда к инструменту, находящемуся в шпинделе.

Прочитаемые данные следующие:

Номер	Символ	Свой- ство	Описание
#8401		R	Номер данных (порядковый номер таблицы оператора инструментов)
#8402		R	Номер типа инструмента (код T)
#8403		R	Значение счётчика стойкости резца
#8404		R	Максимальная стойкость инструмента
#8405		R	Значение предупреждающей стойкости резца
#8406		R	Штат стойкости резца
#8407		R	Битные данные пользователя
#8408		R	Информация об инструменте
#8409		R	H: номер ячейки коррекции радиуса (в фрезерном канале)
#8410		R	D: номер ячейки коррекции радиуса (в фрезерном канале)
#8411		R	S: число оборотов шпинделя
#8412		R	F: подача
#8413		R	G: номер ячейки геометрической коррекции (в токарном канале)
#8414		R	W: номер ячейки коррекции по износу (в токарном канале)
#8431		R	Пользовательские данные 1
#8432		R	Пользовательские данные 2
#8433		R	Пользовательские данные 3
#8434		R	Пользовательские данные 4
#8435		R	Пользовательские данные 5
#8436		R	Пользовательские данные 6
#8437		R	Пользовательские данные 7
#8438		R	Пользовательские данные 8
#8439		R	Пользовательские данные 9
#8440		R	Пользовательские данные 10

Номер	Символ	Свой-ство	Описание
#8441		R	Пользовательские данные 11
#8442		R	Пользовательские данные 12
#8443		R	Пользовательские данные 13
#8444		R	Пользовательские данные 14
#8445		R	Пользовательские данные 15
#8446		R	Пользовательские данные 16
#8447		R	Пользовательские данные 17
#8448		R	Пользовательские данные 18
#8449		R	Пользовательские данные 19
#8450		R	Пользовательские данные 9

Номер пользовательских данных можно задавать параметром N2903 No. of Custom Columns. Первые пользовательские данные всегда битные данные.

Перечисленные выше данные только для чтения.

Если выбранный макропеременной #8400 шпиндель, или дежурный магазин пустой (нет в нём инструмента), значения макропеременных:

#8401=0 (номер данных)

#8402= #8403= ...= #8450= #0 (пустой)

С помощью макропеременных можно вызывать коррекции, назначенных к инструменту (H, D), или технологические параметры (F, S). Если например, на код смены инструмента (M06) выделить вызов подпрограммы, в подпрограмму можно записать:

...

M6

#8400=10 (1. шпиндельный магазин)

H#8409 D#8410 S#8411 F#8412

...

23.1.21 Прочтение данных палитр, находящихся на рабочем месте и на месте монтажа

Номер	Символ	Свой-ство	Описание
#8500		R/W	Адрес рабочего места, или места монтажа

По макропеременной

#8500 (10, 11 можно писать, читать)

надо задавать, что мы желаем прочитать данные палитры, находящейся на рабочем месте (#8500=10), или на месте монтажа (#8500=11) через макропеременные #8501, #8502,

Поддаются определению только номера магазинов, заданных в скобках.

Если на станке имеется лишь одно рабочее место и нет места монтажа, не надо присвоить

значение макропеременной #8500, макропеременные #8501, ... относятся всегда к палитрам, находящимся на рабочем месте.

После включения она установится первоначальным значением #8500=10 .

Можно прочесть следующие данные:

Номер	Символ	Свой- ство	Описание
#8501		R	Идентификатор палитры
#8502		R	Штат
#8503		R	Приоритет
#8504		R	Число выполненных программ
#8505		R	Число выделённых программ
#8531		R	Пользовательский 1
#8532		R	Пользовательский 2
#8533		R	Пользовательский 3
#8534		R	Пользовательский 4

23.2 Команды языка программирования

При описании различных команд используем выражение

$\#i = \langle \text{формула} \rangle$

$\langle \text{формула} \rangle$ может содержать арифметические действия, функции, переменные, постоянные.

Вообще в $\langle \text{формула} \rangle$ —е ссылаемся на переменные $\#j$ и $\#k$.

$\langle \text{формула} \rangle$ может стоять не только на правой стороне команды присвоения значения, а в кадре NC и различные адреса могут принимать вместо переменной конкретные числовые значения или формулу.

На порядок выполнения действий можно влиять квадратными скобками

[,]..

23.2.1 Команда для присвоения значения: $\#i = \#j$

Код команды: =

Под действием команды переменная $\#i$ примет значение переменной $\#j$, то есть в переменную $\#i$ попадает значение переменной $\#j$.

Разрешено и присвоение данных

$\#i = \#i \langle \text{действие} \rangle \#j$

Выполняем действие значением переменной $\#i$ с переменной $\#j$, результат которого попадает в переменную $\#i$.

Например:

$\#100 = \#100 + 1$

это означает, что новое значение $\#100$ будет равно старому значению плюс 1.

Можно присвоить значение и битным переменным:

$\#1100 = 1$

$\#_UO[3] = 0$

$\#_M_SBK = 1$

Если значение, присвоенное битной переменной

$0 < \text{значение} \leq 1$

значение берётся за 1. Если значение больше, чем 1

$1 < \text{значение}$

выводится сообщение об ошибке “2092 Предел значения превышен”.

☞ **Внимание:** По левой стороне команды присвоения значения может стоять переменная только для писать, или для чтения/для писания (W, или W/R). При желании придавать значение переменной только для чтения (R), выводится сообщение об ошибке “2161 $\#nnnn$ макропеременная только для чтения”.

23.2.2 Арифметические действия

Минус с одним операндом: $\#i = - \#j$

Код действия: -

В следствие действия переменная $\#i$ будет равной переменной $\#j$ по абсолютному значению, но противоположной по знаку.

$\#100 = - \#12$

Сложение: $\#i = \#j + \#k$

Код действия: +

В следствие действия переменная $\#i$ примет сумму значений переменных $\#j$ и $\#k$.

$\#1 = \#2 + 3.25$

G0 X[#100+#101]

Вычитание: $\#i = \#j - \#k$

Код действия: -

В следствие действия переменная $\#i$ примет разность значений переменных $\#j$ и $\#k$.

$\#100 = \#100 - \#102$

G1 Z[25.34-2.48]

Умножение: $\#i = \#j * \#k$

Код действия: *

В следствие действия переменная $\#i$ примет произведение значений переменных $\#j$ и $\#k$.

$\#3 = \#1 * \#2$

$\#100 = \#101 * 5.65$

Деление: $\#i = \#j / \#k$

Код действия: /

В следствие действия переменная $\#i$ примет частное значений переменных $\#j$ и $\#k$.

Значение $\#k$ не может быть 0. В противном случае устройство ЧПУ выводит сообщение об ошибке “2093 Попытка деления на ноль”.

$\#3 = \#1 / \#2$

$\#100 = 542.23 / \#3$

Образование модуля: $\#i = \#j \text{ MOD } \#k$

Код действия: MOD

В следствие действия переменная $\#i$ примет остаток деления переменных $\#j$ и $\#k$.

Значение $\#k$ не может быть 0. В противном случае устройство ЧПУ выводит сообщение об ошибке “2093 Попытка деления на ноль”. В случае

$\#120 = 27 \text{ MOD } 4$

значение переменной $\#120$ будет равно 3.

Примеры для использования арифметических действий:

Использованием квадратных скобок [,] можно влиять на порядок выполнения действий.

$\#100 = \#101 + \#102$

$\#100 = \#100 - 3$

$\#100 = [\#101 + \#102 * 5.27] / 4.1$

G0 X[[#100 + 2]/4] Y100

23.2.3 Логические действия

Логические действия обращаются переменными всегда как 32 битным целым числом со знаком. Если выполняется логическое действие на переменной по исходу с плавающей запятой, например, $\#100$, перед выполнением действия проверяется, что значение переменной $\#i$ удовлетворяет ли следующим условиям:

$-(2^{32} - 1) \leq \#i < 2^{32}$,

то есть десятично

$-4294967297 \leq \#i < 4294967296$.

Если нет, выводится сообщение об ошибке “2129 Ошибочное действие на #”.

Если значение переменной попадает внутрь пределов, действие выполняется 32 битным

целым числом. Если результат действия попадает в переменную по исходу с плавающей запятой, например, #100, результат превращает обратно в число с плавающей запятой.

Отрицание: #i = NOT #j

Код действия: **NOT**

В следствие действия переменная #j сперва преобразуется в 32 битное число с фиксированной запятой. После этого берётся от числа с фиксированной запятой отверженное по битам значение на все 32 бита.

Или: #i = #j OR #k

Код действия: **OR**

В следствие действия переменная #j и #k сперва преобразуется в 32 битное число с фиксированной запятой.

В следствие действия в переменную #i попадает логическая сумма значений переменных #j и #k по битам, на все 32 бита. Там, где на одинаковых разрядах обоих чисел на обоих местах стоит 0, на тот разряд в результате попадает 0, а в прочем 1.

Исключающее или: #i = #j XOR #k

Код действия: **XOR**

В следствие действия переменная #j и #k сперва преобразуется в 32 битное число с фиксированной запятой.

В следствие действия в переменной #i так суммируются значения переменных #j и #k по битам, что там, где на одинаковых разрядах находятся одинаковые числовые значения, в результате на то место попадает 0, там, где находятся разные числовые значения, туда попадает 1, на всех 32 битах.

И: #i = #j AND #k

Код действия: **AND**

В следствие действия переменная #j и #k сперва преобразуется в 32 битное число с фиксированной запятой.

В следствие действия в переменную #i попадает логическое произведение значений переменных #j и #k по битам, на все 32 бита. Там, где на одинаковых разрядах обоих чисел находится на обоих местах 1, туда на разряд в результате попадает 1, а в прочем 0.

Пример для использования логических действий:

32 битную макропеременную #1132, идущую в сторону PLC, манипулируем следующем образом: верхний 8 битов переменной (31 - 24) надо ставить так, чтобы нижние биты остались неизменными. Для оставления нижних 24 битов неизменными и удаления верхних 8 битов берём следующую битмаску:

битмаска гексадесятично

00FFFFFF,

преобразив в десятичную:

16777215

Сохраняем битмаску на переменной #100:

#100=16777215

Сохраним на переменной #101 значение, оставляющее нижнего 24 бита #1132, и удаленного на верхних 8 битах значение:

#101=#100AND#1132

На переменную #102 напишем устанавливаемый битный образец:

битный образец гексадесятично:

```

9В000000,
преобразив в десятичную:
-1694498816

```

Сохраним битный образец на переменной #102:

```
#102= -1694498816
```

Затем напишем на переменную #1132 так, чтобы нижние 24 бита остались неизменными:

```
#1132=#101XOR#102
```

Используя скобки надлежащим образом, вместо приведенных выше четырёх строк можно и следующее писать:

```
#1132=[16777215AND#1132]XOR[-1694498816]
```

23.2.4 Функции

Извлечение квадратного корня: #i = SQRT #j

Код функции: **SQRT**

В следствие действия переменная #i примет квадратный корень переменной #j.

```
#101=SQRT[#100+4]
```

Значение переменной #j может быть только положительным числом, или 0:

$$\#j \geq 0$$

Если аргумент отрицательный, выводится сообщение об ошибке “2122 Извлечение корня из отрицательного значения”.

Синус: #i = SIN #j

Код функции: **SIN**

В следствие действия переменная #i примет синус переменной #j. *Значение #j понимается всегда в градусах.*

```
G1 X[SIN30-#101]
```

Косинус: #i = COS #j

Код функции: **COS**

В следствие действия переменная #i примет косинус переменной #j. *Значение #j понимается всегда в градусах.*

```
#102=COS[#1+#2]
```

Тангенс: #i = TAN #j

Код функции: **TAN**

В следствие действия переменная #i примет тангенс переменной #j. *Значение #j понимается всегда в градусах.*

Если значение аргумента #j

$$\#j = (2n+1) \cdot 90^\circ, \text{ где } n=0, \pm 1, \pm 2, \dots$$

выводится сообщение об ошибке “2116 Абсолютное значение аргумента не меньше, чем 90”.

```
#1=TAN[#2*15.6]
```

Арксинус: #i = ASIN #j

Код функции: **ASIN**

В следствие действия переменная #i примет арксинус переменной #j. *Результат задаётся в градусах, значение #i попадает между +90° и -90°.*

```
#101=ASIN[-0.5] (#101=-30 будет)
```

Аргумент #j должен удовлетворить следующим условиям:

$$-1 \leq \#j \leq 1$$

В противном случае выводится сообщение об ошибке “2117 Абсолютное значение

аргумента больше, чем 1".

Арккосинус: #i = ACOS #j

Код функции: **ACOS**

Arkuszkoszinusz: #i = ACOS #j

Код функции: **ACOS**

В следствие действия переменная #i примет арккосинус переменной #j. *Результат задаётся в градусах, значение #i попадает между 0° и 180°.*

#101=ACOS[-0.5] (#101=120 будет)

Аргумент #j должен удовлетворить следующим условиям:

$-1 \leq \#j \leq 1$

В противном случае выводится сообщение об ошибке "2117 Абсолютное значение аргумента больше, чем 1".

Арктангенс: #i = ATAN #j

Код функции: **ATAN**

В следствие действия переменная #i примет арктангенс переменной #j. *Результат задаётся в градусах, значение #i попадает между +90° и -90°.*

#101=ATAN[-0.5] (#101=-26.565 будет)

Показатель: #i = EXP #j

Код функции: **EXP**

В следствие действия переменная #i примет #j-ую степень натурального числа (e).

#100=EXP1 (#100=2.71828... то есть "e" будет)

Натуральные логарифмы: #i = LN #j

Код функции: **LN**

В следствие действия переменная #i примет натуральный логарифм числа #j.

#100=LN2.718281828 (#100=0.999... ln(e)=1 будет)

Если значение аргумента #j

$\#j \leq 0$

выводится сообщение об ошибке "2118 Значение аргумента не положительное".

23.2.5 Команды конверсии

Образование абсолютного значения: #i = ABS #j

Код функции: **ABS**

В следствие действия переменная #i примет абсолютное значение переменной #j.

#100=ABS[-3.1] (#100=3.1 будет)

#100=ABS[5.25] (#100=5.25 будет)

#100=ABS[0] (#100=0 будет)

Округление по абсолютному значению вниз: #i = FIX #j

Код функции: **FIX**

Действие отбрасывает дробную часть переменной #j и это значение попадает в переменную #i.

Например:

#130=FIX4.8 (#130=4 будет)

#131=FIX[-6.7] (#131=-6 будет)

Округление по абсолютному значению вверх : #i = FUP #j

Код функции: **FUP**

Действие отбрасывает дробную часть переменной #j и в абсолютном значении увеличивает на 1.

Например:

```
#130=FUP12.1 (#130=13 будет)
#131=FUP[-7.3] (#131=-8 будет)
```

23.2.6 Порядок выполнения сложных арифметических действий

Перечисленные выше арифметические действия и функции можно комбинировать. **Порядок выполнения действий**, или правило precedence:

функция – мультипликационные действия – аддитивные арифметические действия.

Например, порядок выполнения действий в следующем примере:

```
#110 = #111 + #112 * COS #113
                        1
                    2
                3
```

Изменение порядка выполнения действий

Использованием квадратных скобок [и] можно изменить порядок выполнения действий. Пример для применения скобок трёхкратной глубины:

```
#120 = COS [ [ [ #121 - #122 ] * #123 + #125 ] * #126 ]
                1
            2
        3
    4
1 5
```

Цифры показывают порядок выполнения действий. Видно, что внутри скобок одинакового уровня для порядка выполнения действий действует упомянутое выше правило precedence.

Открывающие [и закрывающие] скобки надо задавать в паре. Если число открывающих [скобок меньше, чем закрывающих, выводится сообщение об ошибке “2064 Синтаксическая ошибка”. Если число закрывающих] скобок меньше, чем открывающих, выводится сообщение об ошибке “2121 Правой скобки не найти”.

23.2.7 Условные выражения

Язык программирования знает следующие условные выражения:

равно (=):	#i EQ #j
не равно (≠):	#i NE #j
больше, чем (>):	#i GT #j
меньше, чем (<):	#i LT #j
больше, или равно (≥):	#i GE #j
меньше, или равно (≤):	#i LE #j

По обеим сторонам условного выражения переменную можно заменить и формулой. Перечисленные выше выражения могут стоять после команд IF, или WHILE.

23.2.8 Безусловное разветвление: GOTO_n

Под действием команды **GOTO_n** выполнение программы продолжается безусловно от команды с номером *n* той же программы. Номер *n* можно заменить и переменной, или формулой также. *Номер кадра*, на который переходим с командой GOTO, *должен фигурировать в начале кадра*.

Если не находить выделённый номер кадра, выводится сообщение об ошибке “2125 Не найти кадра Nnnnn”.

Можно переходить кадром вперёд:

```
...  
GOTO160 (переходит на кадр N160)  
...  
N160 G0 X100  
...
```

или можно и назад. В этом случае может создаваться и бесконечный цикл:

```
...  
N160 G0 X100  
...  
#1=100  
#2=60  
...  
GOTO[#1+#2] (переходит на кадр N160)  
...
```

23.2.9 Условное разветвление: IF[<условное выражение>] GOTO_n

Если завершается [<условное выражение>], закрытое обязательно в квадратные скобки, выполнение *программы продолжается от кадра с номером n той же программы*.

Если не завершается [<условное выражение>], *выполнение программы продолжается следующим кадром*.

Если после IF не последует проверки условий, в зависимости от типа ошибки, выводится сообщение об ошибке “2064 Синтаксическая ошибка”, или “2121 Ошибочный терминатор =”.

Например

```
...  
#100=52.28  
#101=16.87  
...  
IF[#100GT#101] GOTO210 (переходит на кадр N210)  
...  
N210 G0 X0 Y100 Z20  
...
```

23.2.10 Условная команда: IF[<условное выражение>] (THEN)<команда>

Если завершается [<условное выражение>], выполняется команда за THEN.

Если не завершается [<условное выражение>], выполнение программы продолжается следующим кадром.

В команде можно пренебречь THEN

IF[<условное выражение>] команда
выполнение ряда команд то же самое.

```

...
#100=0
#101=0
#1=20
...
IF[#1EQ20] THEN#100=3.15 (#100=3.15 будет)
IF[#100GT3.15] #101=5 (#101=0 остаётся)
...

```

23.2.11 Организация цикла: WHILE[<условное выражение>] DOm ... ENDm

Пока завершается [<условное выражение>], повторно выполняются кадры после DO m до кадра END m. То есть устройство ЧПУ проверит, что удовлетворилось ли условие, если да, выполняет часть программы между DOm и ENDm, затем под действием команды ENDm программа возвращается к повторной проверке условия после WHILE.

Если не завершается [<условное выражение>], выполнение программы продолжается кадром после ENDm.

Если пренебречь WHILE [<условное выражение>], то есть цикл описывается командами DOm ... ENDm, часть программы между DOm и ENDm выполняется до бесконечного времени.

возможные значения m: 1, 2, 3.

Случай m>3 приводит к выводу сообщения об ошибке “2002 Данные DO вне пределов значений”.

Если после WHILE последует не проверка условий, а команда присвоения значения, выводится сообщение об ошибке “2121 Ошибочный терминатор: =”.

Если за командой WHILE не последует DO в том же кадре, выводится сообщение об ошибке 2110 WHILE, без DO”.

Если перед командой ENDm нет DOm, выводится сообщение об ошибке “2125 Не найти кадра: DOm”.

Расскроем полость, в направлении Z с глубиной 5.4 мм, в направлении X с шириной 20 мм, в центре которой в направлении X лежит нулевая точка, а в направлении Z - на вершине полости. Пусть будет глубина резания 0.2 мм. Задачу можно решить с организацией цикла:

```
#1=0          (целевая позиция в направлении Z)
#2=10         (целевая позиция в направлении X)
G0 X#2        (перемещение по X в начальную точку)
Z[#1+1]       (перемещение по Z в начальную точку)
WHILE[#1GT-5.39] DO1
(цикл, пока по Z не достигли глубину 5.4 мм)
#1=#1-0.2     (врезание 0.2 мм Z)
#2=-#2        (перевернуть направление движения по X)
G1 Z#1 F100    (врезание по Z)
X#2 F500       (фрезерование вдоль X)
END1
G0 Z20
...
```

Правила организации циклов:

– Команду DOn надо задавать перед командой ENDm:

```
:
END1
:
:          (ОШИБОЧНАЯ)
:
DO1
```

В случае выше кадром END1 выводится сообщение об ошибке “2125 Не найти кадра DOn”.

– Команды DOn и ENDm должны стоять в паре:

```
:
DO1
:
DO1          ОШИБОЧНАЯ
:
END1
:
```

или

```
:
DO1
:
END1          ОШИБОЧНАЯ
:
END1
:
```

– То же самое опознавательное число можно много раз использовать:

```

:
DO1
:
END1
:
: ПРАВИЛЬНАЯ
:
DO1
:
END1
:

```

– Пары DOm ... ENDm можно вкладывать друг в друга до трёхкратной глубины:

```

:
DO1
:
DO2
:
DO3
:
: ПРАВИЛЬНАЯ
:
END3
:
END2
:
END1
:

```

– Парам DO m ... END m нельзя перекрывать друг друга:

```

:
DO1
:
DO2
:
: ОШИБОЧНАЯ
:
END1
:
END2

```

– Изнутри цикла можно делать разветвление снаружи:

```
:  
DO1  
:  
GOTO150  
:  
: ПРАВИЛЬНАЯ  
:  
END1  
:  
N150  
:
```

– В цикл можно входить извне:

```
:  
GOTO150  
:  
DO1  
:  
:  
:  
N150  
:  
END1  
:
```

или

```
:  
DO1  
:  
N150  
:  
:  
:  
END1  
:  
GOTO150  
:
```

– Изнутри цикла возможно вызывать подпрограмму, или макровывоз. Подпрограммы, или циклы внутри пользовательских макросов можно опять вкладывать друг в друга до трёхкратной глубины :

```

:
DO1
:
M98...          ПРАВИЛЬНАЯ
:
G65...          ПРАВИЛЬНАЯ
:
G66...          ПРАВИЛЬНАЯ
:
G67...          ПРАВИЛЬНАЯ
:
END1
:

```

23.2.12 Косвенные ссылки на оси

С помощью команды

#i=AXNUM[<адрес оси>]

на основании адреса оси можно запрашивать номер оси в данном канале. При желании запрашивать номер оси к не существующему адресу оси, выводится сообщение об ошибке “2017 Запрещённый адрес <название оси>”.

Командой

AX[<номер оси>]=

можно косвенным образом выдавать команду, не на основании адреса оси, а по номеру оси. Если в канале не найти оси с заданным номером, выводится сообщение об ошибке “2018 Ошибочный физический номер оси: n”.

Например, вместо кадра

G0 X10 Y20 Z30

можно писать следующий ряд команд

```

#101=AXNUM[X]      (запрашивание номера оси X)
#102=AXNUM[Y]      (запрашивание номера оси Y)
#103=AXNUM[Z]      (запрашивание номера оси Z)
G0 AX[#101]=10 AX[#102]=20 AX[#101]=30

```

☞ **Внимание!** Если используем адрес оси с несколькими символами, адрес оси не должно быть AX, или AXN.

23.2.13 Команды выдачи данных

Устройство ЧПУ знает следующие команды выдачи данных:

POPEN	открыть периферию
FOPEN	открыть файл
BPRNT	выдача бинарных данных
DPRNT	выдача десятичных данных
PCLOS	закрыть периферию
FCLOS	закрыть файл

Эти команды выдачи данных используются для выдачи значений символов и переменных. Выдача происходит в памяти устройства ЧПУ. Используются они например, для сохране-

ния результатов измерений, занесение в журнал и т.д.

Открыть периферию: **ROPENn**

Прежде чем выдавать команду выдачи данных, надо открыть соответствующую периферию, через которую будет происходить выдача данных. Выбор соответствующей периферии происходит числом *n*:

n = 31 память устройства ЧПУ

При открывании периферии высылается на периферию и символ %, значит, каждая выдача данных начинается символом %.

После команды *ROPEN* командой *DPRNT* надо выдавать одно имя файла. Если файл уже существует, без вопроса изменяет старого, если ещё не существует, откроет новый. Файл попадает в ту папку, где программа работает, и примет расширение программы.

Например:

```
...
#100=4567
ROPEN31
DPRNT[O#100[4]]      (Имя файла будет O4567)
...
```

или

```
...
ROPEN31
DPRNT[ABC]            (Имя файла будет ABC)
...
```

Закреть периферию: **PCLOSn**

Периферию, открытую командой *ROPEN*, надо закрыть командой *PCLOS*. После команды *PCLOS* надо назвать номер закрываемой периферии. В нашем случае:

```
...
PCLOS31
...
```

При закрытии высылается на периферию ещё один символ %, то есть каждая выдача данных закрывается символом %.

Открыть файл: **FOPENn Prrrrr, FOPENn <имя файла>**

Прежде чем выдавать команду выдачи данных, надо открыть файл, куда желаем записать данные.

Открытие файла на основании номера программы

Открывает файл с номером, заданным

FOPENn P(номер программы)

по адресу *P* (номер программы).

Правила программирования номера программы, заданного по адресу *P* изложены во главе “[14.4.1 Идентификация программ в накопителе. Номер программы \(O\)](#)” на странице [118](#). **Правила расширения и размещения в накопителе файлов, открытых с номером программы** совпадают с правилами подпрограмм. Смотри главу “[14.4.2 Вызов подпрограммы \(M98\)](#)” на странице [118](#).

Например:

```
...
FOPEN1 P12          (Имя файла будет 00012)
...
```

Открыть файл на основании имя файла

Открывает файл, заданную

FOPENn <имя файла>

между знаками < >.

Правила задания имя файла изложены во главе “[14.4.1 Идентификация программ в накопителе. Номер программы \(O\)](#)” на странице [118](#). **Правила расширения и размещения в накопителе файлов, открытых с номером программы** совпадают с правилами подпрограмм. Смотри главу “[14.4.2 Вызов подпрограммы \(M98\)](#)” на странице [118](#).

Например:

```
...
FOPEN1 <данные.txt> (данные имени файла.txt будет)
...
```

Типы открытия файла

Тип открытия файла задаётся числом, написанным вместо “n”:

n = 1: Создать новый файл

Если файл уже существует, тогда система выводит сообщение об ошибке “2144 Файл (Ooooo) уже существует”, или “2146 Заданный файл уже существует”, и прогон программы не продолжается.

n = 2: Если файл не существует, тогда создаёт, если существует, тогда откроет его, и удаляет его содержание.

Содержание, выданные командой BPRNT, или DPRNT будет писать сначала файла.

n = 3: Открыть файл для прибавления

Если файл не существует, тогда выводится сообщение об ошибке “2147 Файл (Ooooo) не найти”, или “2132 Файл не найти”, и прогон программы не продолжается. Если файл уже существует, тогда откроет его, и **начиная с конца файла прибавит** содержания, выданные командой BPRNT, или DPRNT.

n = 4: Создать фай, или открыть для прибавления

Если файл ещё не существует, тогда **создаёт, если существует, тогда откроет, и начиная с конца файла прибавит** содержания, выданные командой BPRNT, или DPRNT.

n = 5: Открыть файл с удалением

Если файл ещё не существует, тогда выводится сообщение об ошибке “2147 Файл (Ooooo) не найти”, или “2132 Файл не найти”, и прогон программы не продолжается. Если файл уже существует, тогда **откроет его, и удаляет его содержание.** Содержание, выданные командой BPRNT, или DPRNT будет писать сначала файла.

Закрывать файл: FCLOS

Открытый файл можно закрывать командой FCLOS. Одновременно можно создать, или открыть для писания один файл.

Бинарная выдача данных: BPRNT[...]

Формат команды BPRNT:

```
BPRNT[ a #b [c] ... ]
```

число цифр под десятичной запятой
переменная
символ

Команда высылает символы по коду ASCII, а переменные бинарно.

– Высылаемые символы:

алфавитные символы: A, B, ..., Z
 нумерические символы: 1, 2, ..., 0
 специальные символы: *, /, +, -

Вместо символа * устройство ЧПУ высылает код пробела (space).

- Значение переменных выдаётся устройством ЧПУ на 4 байтах, то есть на 32 битах, начиная с байта с самым большим разрядом. После номера переменных в скобках [] надо задавать число цифр после десятичной запятой. При этом устройство ЧПУ преобразует значение переменной с плавающей запятой в такое значение с фиксированной запятой, в котором число значащих десятичных цифр будет значением, заданным в скобках []. Возможные значения c: 1, 2, ..., 8. Например, если
 $\#120 = 258.647673$ и [3] — выдаётся $258648=0003F258h$
- Пустая переменная задаётся бинарным кодом 00000000h.
- В конце выдачи данных устройство ЧПУ автоматически выдаёт один символ каретка назад (CR) и один символ рычага интервала (LF).

Пример:

```
#110=318.49362
#120=0.723415
#112=23.9
BPRNT[C*/X#110[3]Y#120[3]M#112[0]]
```

Значения переменных #110, #120 и #112 после округления и гексадесятичного преобразования следующие будут:

#110=318.49362	—————	318494=0004DC1Eh
#120=0.723415	—————	723=000002D3h
#112=23.9	—————	24=00000018h

Выдаваемые символы:

7	6	5	4	3	2	1	0	
0	1	0	0	0	0	1	1	C
0	0	1	0	0	0	0	0	Пробел (Space)
0	0	1	0	1	1	1	1	/
0	1	0	1	1	0	0	0	X
0	0	0	0	0	0	0	0	00
0	0	0	0	0	1	0	0	04
1	1	0	1	1	1	0	0	DC
0	0	0	1	1	1	1	0	1E
0	1	0	1	1	0	0	1	Y
0	0	0	0	0	0	0	0	00
0	0	0	0	0	0	0	0	00
0	0	0	0	0	0	1	0	02
1	1	0	1	0	0	1	1	D3
0	1	0	0	1	1	0	1	M
0	0	0	0	0	0	0	0	00
0	0	0	0	0	0	0	0	00
0	0	0	0	0	0	0	0	00
0	0	0	1	1	0	0	0	18
0	0	0	0	1	1	0	1	Каретка назад (Carriage Return)
0	0	0	0	1	0	1	0	Рычаг интервала (Line Feed)

Десятичная выдача данных: DPRNT[...]

Формат команды DPRNT:

```
DPRNT[ a #b [c d] ... ]
```

— число цифр после десятичной запятой
 — число цифр до десятичной запятой
 — переменная
 — символ

Все символы и цифры выдаются по коду ASCII.

- Правила выдачи символов смотри команду **BPRNT**.
- Для выдачи значения переменных надо задавать, что на сколько десятичных целых и дробных цифр была выдана переменная. Задание цифр надо закрыть в скобки []. Для задания цифр должно удовлетвориться условие $0 < c + d < 16$. Выдача чисел начинается с их высшего разряда.
- При выдаче цифр выдаётся отрицательный знак (-).
- Если определена десятичная запятая ($d > 0$), выдаётся каждая ноль, и последующие ноли, вместе с десятичной запятой (.).
- Если $d = 0$, или не задано d , не выдаётся ни десятичная запятая, ни ноль.
- Если бит #0 PNT параметра N1757 Print Contr
 =0 вместо знака + и ведущие ноли выдаётся пробел (space),
 =1 знак + ведущие ноли не выдаются.
- Пустая переменная выдаётся кодом 0.

- В конце выдачи данных устройство ЧПУ автоматически выдаёт один символ каретка назад (CR) и один символ рычага интервала (LF).

Пример:

```
#130=35.897421
#500=-150.8
#10=14.8
DPRNT[X#130[53]Y#500[53]T#10[20]
```

Значения переменных #130, #500 и #10 после округления следующие будут:

#130=35.897421	_____	35.897
#500=-150.8	_____	150.8
#10=14.8	_____	15

Выдача данных при положении #0 PNT=0 параметра N1757 Print Contr:

7	6	5	4	3	2	1	0	
0	1	0	1	1	0	0	0	X
0	0	1	0	0	0	0	0	Пробел (space)
0	0	1	0	0	0	0	0	Пробел (space)
0	0	1	0	0	0	0	0	Пробел (space)
0	0	1	0	0	0	0	0	Пробел (space)
0	0	1	1	0	0	1	1	3
0	0	1	1	0	1	0	1	5
0	0	1	0	1	1	1	0	Десятичная запятая (,)
0	0	1	1	1	0	0	0	8
0	0	1	1	1	0	0	1	9
0	0	1	1	0	1	1	1	7
0	1	0	1	1	0	0	1	Y
0	0	1	0	1	1	0	1	Отрицательных знак (-)
0	0	1	0	0	0	0	0	Пробел (space)
0	0	1	0	0	0	0	0	Пробел (space)
0	0	1	1	0	0	0	1	1
0	0	1	1	0	1	0	1	5
0	0	1	1	0	0	0	0	0
0	0	1	0	1	1	1	0	Десятичная запятая (,)
0	0	1	1	1	0	0	0	8
0	0	1	1	0	0	0	0	0
0	0	1	1	0	0	0	0	0
0	1	0	1	0	1	0	0	T
0	0	1	0	0	0	0	0	Пробел (space)
0	0	1	1	0	0	0	1	1
0	0	1	1	0	1	0	1	5
0	0	0	0	1	1	0	1	Каретка назад (Carriage Return)
0	0	0	0	1	0	1	0	Рычаг интервала (Line Feed)

Выдача данных при положении #0 PNT=1 параметра N1757 Print Contr:

7	6	5	4	3	2	1	0	
0	1	0	1	1	0	0	0	X
0	0	1	1	0	0	1	1	3
0	0	1	1	0	1	0	1	5
0	0	1	0	1	1	1	0	Десятичная запятая (,)
0	0	1	1	1	0	0	0	8
0	0	1	1	1	0	0	1	9
0	0	1	1	0	1	1	1	7
0	1	0	1	1	0	0	1	Y
0	0	1	0	1	1	0	1	Отрицательных знак (-)
0	0	1	1	0	0	0	1	1
0	0	1	1	0	1	0	1	5
0	0	1	1	0	0	0	0	0
0	0	1	0	1	1	1	0	Десятичная запятая (,)
0	0	1	1	1	0	0	0	8
0	0	1	1	0	0	0	0	0
0	0	1	1	0	0	0	0	0
0	1	0	1	0	1	0	0	T
0	0	1	1	0	0	0	1	1
0	0	1	1	0	1	0	1	5
0	0	0	0	1	1	0	1	Каретка назад (Carriage Return)
0	0	0	0	1	0	1	0	Рычаг интервала (Line Feed)

Замечания:

- Порядок команд выдачи данных обусловленный: сперва надо открыть соответствующий файл командой POPEN, или FOPEN, потом следует выдача данных командой BPRNT, или DPRNT, и наконец надо закрыть открытый файл командой PCLOS, или FCLOS.
- Открытие и закрытие файла можно задавать в любом месте программы. Например, можно открыть в начале программы, и закрыть в конце программы, и в это время в любом фрагменте программы, находящемся между двумя командами, можно выслать данные.
- Команда M30, или M2, выполненная во время выдачи данных, прерывает передачу данных. Для избежания этого перед выполнением команды M30 надо ожидать во время передачи данных.

23.3 Вызов макрокоманд, системных макросов, системных подпрограмм

*Вызов макрокоманд похож к вызову подпрограмм, той разницей, что **макросы**, в отличие от подпрограмм, **могут иметь вводные переменные**, то есть **аргументы**.*

Новые макросы можно вызывать также, как подпрограммы, на основании их номера программы, или имени файла также.

Системными макросами и системными подпрограммами** называем те макросы, далее подпрограммы, вызов которых привязываем к одному данному **адресу, выделённому параметром** (например: G, M и т.д.). **К имени файла системных макросов и подпрограмм далее к их позиции в накопителе относятся специальные правила.

***Макровывоз также может быть многоуровневым.** Макрос можно вызывать из подпрограммы, и из макроса подпрограмму. Общий уровень вызова подпрограмм и макросов может быть **не более 16**.*

***Возвращение из макросов** выполняется кодом
M99.*

***Макровывозы могут быть одноразовыми, или модалными.** Модалные макровывозы можно удалить кодом
G67.*

Задание аргументов

Для макропрограмм можно передавать аргументы. ***Аргументами являются такие конкретные числовые значения, заданные определённым адресам, которые при макровывозе сохраняются в соответствующих локальных переменных (#1, #2, ..., #33).*** Макропрограмма (специальная подпрограмма) может использовать эти локальные переменные, то есть макровывоз представляет собой такой специальный вызов подпрограмм, где главная программа может передать переменные (параметры) для подпрограммы.

Возможно двоякое выделение аргументов:

1-й вид выделения аргумента:

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

1-й вид выделения аргументов передаёт параметры макросам через букв английского алфавита. Из перечисленных выше символов отдельные вызовы могут занять буквы для другой цели (**G, P, L**), при этом их нельзя использовать для передачи параметров. Адреса можно заполнять в произвольном порядке, не требуется их записать согласно алфавиту в вызывающий кадр.

2-й вид выделения аргумента:

A B C I1 J1 K1 I2 J2 K2 ... I10 J10 K10

2-й вид выделения аргументов передаёт аргументы макросам помимо адресов A, B, C через адреса I, J, K. По адресам I, J, K можно выделить не более 10 разных групп аргументов. Адреса можно заполнять в произвольном порядке. Если в одном кадре выделён несколько аргументов на один и тот же адрес, переменные примут соответствующее значение согласно порядку выделения.

Связь между локальными переменными и аргументами:

	<i>1-й вид</i>	<i>2-й вид</i>
#1	A	A
#2	B	B
#3	C	C
#4	I	I1
#5	J	J1
#6	K	K1
#7	D	I2
#8	E	J2
#9	F	K2
#10	G	I3
#11	H	J3

	<i>1-й вид</i>	<i>2-й вид</i>
#12	L	K3
#13	M	I4
#14	N	J4
#15	O	K4
#16	P	I5
#17	Q	J5
#18	R	K5
#19	S	I6
#20	T	J6
#21	U	K6
#22	V	I7

	<i>1-й вид</i>	<i>2-й вид</i>
#23	W	J7
#24	X	K7
#25	Y	I8
#26	Z	J8
#27	—	K8
#28	—	I9
#29	—	J9
#30	—	K9
#31	—	I10
#32	—	J10
#33	—	K10

Индексы после адресов I, J, K показывают порядок выделения аргументов.
По адресам можно передавать и десятичную запятую, и знак.

Обращение адресом N:

Записанный в кадр *первый адрес N берётся за номером кадра, а второй адрес N запишется уже как аргумент* на локальной переменной #14:

```
/N130 X12.3 Y32.6 N250
N130: номер кадра
#24=12.3
#25=32.6
#14=250 (аргумент)
```

Если адрес N уже один раз записали как аргумент, следующая ссылка на этот адрес N приводит к выведению сообщения об ошибке “2017 Запрещённый адрес N”.

Смешанное выделение аргумента

1-й и 2-й вид выделения аргументов может существовать и вместе внутри одного кадра, устройство ЧПУ примет его. Выводится сообщение об ошибке тогда, если два раза желаем сослаться на переменную с данным номером. Например:

```
A2.12 B3.213 J36.9 J-12 E129.73 P2200
```

Локальные переменные примут значения адресов в следующем порядке:

```
A: #1=2.12
B: #2=3.213
J: #5=36.9 (первая J)
J: #8=-12 (вторая J)
E: #8=ОШИБКА, переменная #8 уже выделена была
```

этом примере для #8 уже присвоил значение второй адрес J (его значение – 12). Поскольку и значение адреса E примет переменная #8, устройство ЧПУ выводит сообщение об ошибке “2017 Запрещённый адрес E”.

Однако, если поменять порядок выдачи второго адреса J и адреса E,

A2.12 B3.213 J36.9 **E129.73 J-12** P2200

не выводится сообщение об ошибке и значение адреса J запишется на следующую переменную J, на #11:

A: #1=2.12

B: #2=3.213

J: #5=36.9 (первая J)

E: #8=129.73

J: #11=-12 (попадает на третью переменную J, так как вторая переменная J, #8 уже занята)

☞ **Внимание:** Для передачи аргументов можно использовать только адреса с одним символом, то есть, C, R, A далее адреса осей и шпинделей с несколькими символами нельзя использовать!

Уровни аргументов

Макровывоз также, на подобие вызову подпрограмм, *может быть многоуровневым*, но совместный уровень вызова подпрограмм и макросов может быть не более 16.

Из-за этого локальные переменные #1 ... #33 тоже многоуровневые. Уровень локальных переменных, относящихся к главной программе 0 (нулевой уровень), потом вызовы макросов заполняют уровни 1-й, 2-й, и т.д., совсем до 16 не более.

Вызов подпрограмм, или системных подпрограмм не изменяет уровень локальных переменных.

После возвращения из макровывоза уничтожаются локальные переменные данного уровня, удаляются до #0. **Локальные переменные главной программы** уничтожаются в конце программы.

Вызов макросов на основании номера программы

Вызывается макропрограмма с номером,

P(номер программы)

заданным по адресу P (номер программы)

Правила программирования номера программ, заданного по адресу P, изложены во главе “[14.4.1 Идентификация программ в накопителе. Номер программы \(O\)](#)” на странице [118](#). **Правила расширения и размещения в накопителе макросов, открытых с номером программы** совпадают с правилами подпрограмм. Смотри главу “[14.4.2 Вызов подпрограммы \(M98\)](#)” на странице [118](#).

Вызов макросов на основании имени файла

☞ **Внимание!** На основании имени файла *можно вызывать только* не модальные *макросы типа G65*, а модальные макросы 1-го типа G66 и G66.1 нельзя!

Вызывается макропрограмма с именем файла, заданным

<имя файла>

в скобках < >.

Правила задания имени файла изложены во главе “[14.4.1 Идентификация программ в накопителе. Номер программы \(O\)](#)” на странице [118](#). **Правила расширения и размещения в накопителе макросов, заданных с именем файла** совпадают с правилами подпрограмм. Смотри главу “[14.4.2 Вызов подпрограммы \(M98\)](#)” на странице [118](#).

Вызов и место в накопителе системных макросов и подпрограмм

Системные макросы и подпрограммы *можно вызывать по адресам, заданным параметром*.

К размещению в накопителе системных макросов и подпрограмм относятся более строгие правила, чем при вызове макросов и подпрограмм.

Системные макросы и подпрограммы должны быть в накопителе *в отдельных по каналам папках*, в папке *SystemMacros* в *каталоге Programs*:

..\Programs\SystemMacros\Channel1\

..\Programs\SystemMacros\Channel2\

..\Programs\SystemMacros\Channel8\

Имена файлов должны начинаться с буквы **O**, за которой должны следовать **4 десятичных цифр**. **Расширение** файла должно быть **.nct**. Например:

O9010.nct

23.3.1 Простой макровывод (G65)

Под действием команды

G65 P(номер программы) L(число повторения) <выделение аргумента>

G65 <имя файла> L(число повторения) <выделение аргумента>

вызывается макропрограмма с номером, заданным по адресу P (номер программы), или с именем файла, заданным в скобках < > , повторно одна за другой, согласно числу повторения, заданным по адресу L.

Вызов *не модальный*.

Правила выделения аргумента в случае вызова G65

Можно использовать оба выделения аргумента.

В вызове G65 нельзя использовать адреса G, P, L для передачи аргумента, но заданные значения записываются в соответствующую локальную переменную.

Многократный вызов G65

Если из макроса вызывать снова макрос, на уровень макроса увеличивается и уровень локальных переменных.

главная программа	макрос	макрос	макрос	макрос
0-й уровень	1-й уровень	2-й уровень	3-й уровень	4-й уровень
	O _____	O _____	O _____	O _____

G65 P

G65 P

G65 P

G65 P

M99

M99

M99

M99

локальные переменные

0-й уровень	1-й уровень	2-й уровень	3-й уровень	4-й уровень
#1	#1	#1	#1	#1
	:	:	:	:
#33	#33	#33	#33	#33

При вызове первого макроса локальные переменные от #1 до #33 главной программы сохраняются, и при вызове на 1-ом уровне локальные переменные примут заданные значения аргументов. В случае повторного вызова макроса из первого уровня локальные переменные от #1 до #33 первого уровня сохраняются, и локальные переменные при вызове на втором уровне примут значения аргумента, заданных при их вызове. В случае многократном вызове локальные переменные предыдущего уровня сохраняются и локальные переменные на следующем уровне примут значения аргумента, заданных при их вызове. В случае M99, при возвращении из вызванного макроса в вызывающую программу, локальные переменные, сохранённые на предыдущем уровне восстанавливаются в том же состоянии, как они сохранились при вызове.

Обрабатывается образец отверстия с помощью вызова G65. По адресу "А" задаётся для макроса код сверления, заполнение остальных адресов происходит способом, привычным при циклов сверления.

Пусть будет **главная программа** следующая:

G54 G17 X0 Y0 Z20

...

G65 P300 A81 Z-2 R2 F300 S500 M3 (разметка кернером с G81)

G65 P300 A83 Z-30 R2 Q6 E1 F100 S1000 (сверление с G83)

G65 P300 A84.2 Z-30 R2 S1000 F1000 (сверление резьбы с G84.2)

G0 Z20

...

В макросе O0300 через локальные переменные принимаем аргумента вызова, данные, относящиеся к шпинделю

число оборотов S#19

направление оборотов M#13

вводные параметры сверления

код цикла сверления A#1

глубину сверления Z#26

координату точки R R#18
 значение подачи F#9
 величину глубины резания Q#17
 расстояние точки приближения E#8

После этого следует перечисление координат отверстия X, Y.

Дерево **макроста O0300**, выполняющего сверление:

S#19 M#13 (запуск шпинделя)
 G#1 X0 Y0 Z#26 R#8 F#9 Q#17 E#8 (установка цикла сверления)
 X100
 Y100
 X0
 X50 Y50
 G80
 M99

23.3.2 Модальный вызов макроста после каждой команды движения: (G66)

Под действием команды

G66 P(номер программы) L(число повторения) <выделение аргумента>

заданная с номером по адресу P (номер программы) **макропрограмма вызывается после каждой команды движения** повторно одна за другой, согласно числу повторения, заданном по адресу L.

Модальный вызов.

Выделённый макрос вызывается до тех пор, пока не запрограммировать команду удаления

G67

модальности макровывода.

Правила выделения аргумента в случае вызова G66

Можно использовать оба выделения аргумента.

Вызове G66 нельзя использовать адреса G, P, L для передачи аргумента, но заданные значения записываются в соответствующую локальную переменную.

Пример

В данном сегменте программы деталей после каждого движения надо высверлить одно отверстие:

Главная программа

```
...
G66 P1250 Z-100 R-1 X2 F130 (Z: точка основания, R: точка
      R отверстия, X: время задержки, F: подача)
N100 G91 G0 X100
N110 Y30
...
N180 X150
G67
```

Начиная с кадра N110 включительно до кадра N180, в конце позиционирования вызывается макропрограмма O1250 и выполняется описанная там операция сверления вводными параметрами, заданными в кадре G66.

Макропрограмма:

```
O1250
G0 Z#18      (позиционирование быстрым ходом в направлении Z в позицию -1, заданную по адресу R)
G1 Z#26 F#9   (сверление подачей 130 мм/мин, заданной по адресу F до точки основания -100, заданной по адресу Z)
G4 P#24       (задержка на дне отверстия до значения 2 сек, заданной по адресу X)
G0 Z-[#18+#26] (отвод инструмента в исходную точку)
M99          (возвращение в вызывающую программу)
```

Макропрограмма O1250 из кадра N100 главной программы понаследует состояние G91, поэтому позиционирования происходят инкрементно.

Многократный вызов G66

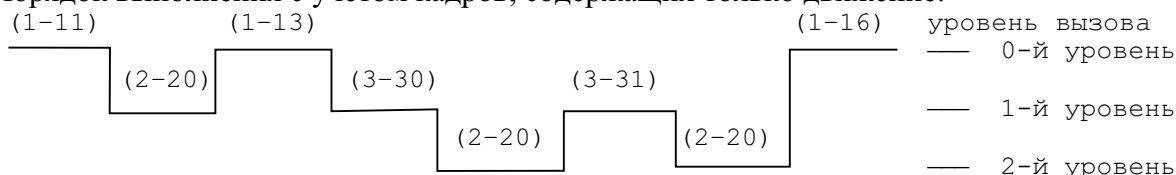
В случае многократного вызова макросов типа G66 после выполнения каждой команды движения *сначала вызывается макрос, вызванный последним, и из этого вызываются в обратном порядке ходу макросы, вызванные раньше*. Посмотрим следующий пример:

```
O0001
...
N10 G66 P2
N11 G1 G91 Z10      (1-11)
N12 G66 P3
N13 Z20             (1-13)
N14 G67             (G66 P3 удаление вызова)
N15 G67             (G66 P2 удаление вызова)
N16 Z-5             (1-16)
...

O0002
N20 X4              (2-20)
N21 M99

O0003
N30 Z2              (3-30)
N31 Z3              (3-31)
N32 M99
```

Порядок выполнения с учётом кадров, содержащих только движение:



Из заключенных в скобках чисел первое означает номер программы, выполняемой в данный момент, а второе - номер кадров, выполняемого в данный момент.

Каждый вызов G66 надо удалить отдельной командой G67. Первая G67 удаляет вызванную последним G66, затем поступая в обратную сторону следующая удаляет остальные вызовы. Заданная в кадре N14 команда G67 удаляет вызванного в кадре N12

макрос (O0003), заданная в кадре N15 удаляет макрос (O0002), заданного в кадре N10.

23.3.3 Модальный макровывоз из каждого кадра: (G66.1)

Под действием команды

G66.1 P(номер программы) L(число повторения) <выделение аргумента>

все последующие кадры истолкуются выделением аргумента, и заданная с номером по адресу P (номер программы) **макропрограмма вызывается для каждого кадра** повторно одна за другой, согласно числу повторения, заданным по адресу L.

Действие команды то же самое, как будто каждый кадр являлся бы макровывозом G65:

```
G66.1 P L
X Y Z = G65 P L X Y Z
M S = G65 P L M S
X = G65 P L X
G67
```

Модальный вызов.

Выделённый макрос вызывается до тех пор, пока не запрограммировать команду удаления модальности макровывоза

G67.

Правила выделения макровывоза:

1. **В кадре, выполняющем включение** (где запрограммировали G66.1 P L):
в вызове G66.1 нельзя использовать адреса G, P, L для передачи аргумента, но заданные значения записываются в соответствующую локальную переменную.
2. **В кадрах, последующих за командой G66.1:**
можно использовать и адрес G, L, P.
Адрес G (G: #10) с таким ограничением, что устройство ЧПУ принимает *в одном кадре только одну ссылку на один адрес G*, если запрограммировали больше адресов G, выводится сообщение об ошибке “2017 Запрещённый адрес G”.

Правила выполнения кадра в случае G66.1:

Выделённый макрос будет вызван уже из того кадра, где задавали код G66.1, с учётом правила выделения аргументов в пункте 1.

Начиная с кадра, последующего за кодом G66.1, до кадра, содержащего код G67, каждый кадр NC приводит к макровывозу на основании правила выделения аргументов в пункте 2. Не вызывается макрос, если находит пустой кадр, например: N1240, где ссылка была только на один адрес N, а также из кадра, содержащего макрокоманду.

Многократный вызов G66.1

В случае многократного вызова макросов типа G66.1 **сперва вызывается макрос, вызванный последним, при прочтении каждого кадра** обращаясь адресами этого кадра за аргументом, затем прочитав кадры этого макроса и обращаясь ими аргументом, вызывается макрос, заданный на один раньше.

Каждый вызов G66.1 надо удалить отдельной командой G67. Первая G67 удаляет вызванный последним G66.1, затем следующая G67 поступая обратно удаляет остальные вызовы.

Следующий фрагмент программы написан в плоскости XY, когда на станке была головка с вертикальной позицией.

```
...
G1 X40 Y30 F1000
G3 X0 Y50 I-40 J-30
G1 Y0
...
```

Со временем головку переоборудовали в головку с горизонтальной позицией. В интересах того, чтобы программу, написанную в плоскости XY не надо было снова написать, надо поменять оси Y и Z, а направление оси X поменять на противоположное, чтобы система координат осталась правокрученной. Вызовом макроса O0400 повернём направление движения оси X и поменяем оси Y и Z:

Главная программа:

```
...
G66.1 P400 (вызов макроса смены)
G18 X0 Y0 Z-5 (выделение плоскости, позиционирование)

G1 X40 Y30 F1000 (вызов O0400)
G3 X0 Y50 I-40 J-30 (вызов O0400)
G1 Y0 (вызов O0400)
G67 (удаление модального вызова)
...
```

Макрос:

```
IF[#10EQ66.1] GOTO10 (защита от рекурсивного вызова)
G#10 X-#24 Y#26 Z#25 I-#4 K#5 R#18 F#9 (замены)
N10 M99
```

Уже кадром G66.1 P400 главной программы вызывается макрос O0400. При этом на переменной #10 проходит код 66.1. На первый вызов сразу возвращаемся. Второй кадр макроса принимает аргументы и обрабатывает их: меняет оси Y и Z и направление X.

23.3.4 Вызов системного макроса по коду G, заданному параметром

Можно выделить индивидуальные коды G, далее блоки кодов G, задавая параметром, по каналам, к макровывозу.

Выделение 10 индивидуальных кодов G к вызову системных макросов

В поле параметров *можно выделить по каналам не более 10 разных кодов G*, по которым желаем инициировать макровывозы. При этом вместо ряда команд

Nn G65 Pp <выделение аргумента>

надо писать ряд команд

Nn Gg <выделение аргумента>.

В поле параметров надо установить, что какой номер программы должен вызывать приглашающий код G. Код G65, G66, G66.1, G67 нельзя задавать для такой цели.

Эти параметры следующие:

N1704 G(9010): код G, вызывающий программу с именем O9010.nct

N1705 G(9011): код G, вызывающий программу с именем O9011.nct

:

N1713 G(9019): код G, вызывающий программу с именем O9019.nct

Если на параметр написать 0, не вызывается макрос с данным номером программы.

Если *на G0* желаем инициировать *макровывоз*, надо писать на параметр *1000*.

Выделение группы кодов G к вызову системного макроса

С помощью следующих параметров *можно выделить блок кодов G*, по каналам, к макровыводу.

На параметре N1714 Start G Macro задаётся начальное число блока кодов G, десятичным целым числом,

На параметре N1715 Start Prg No задаётся номер программы макроса, относящегося к коду G, заданному параметром Start G Macro

На параметре N1716 No. of G Codes задаётся число элементов блока кодов G.

Если No. of G Codes=0, на эти коды G не вызывается макрос.

Пример

Пусть будет, например, начальный код группы G2000. Надо установить Start G Macro=2000.

Если код G2000 вызывает программу с номером O3400, тогда Start Prg No=3400.

На параметре No. of G Codes задаются номера кодов G, относящихся к группе. Если в группе имеется 50 кодов, тогда No. of G Codes=50.

После этого сопоставление кодов G с номерами программ:

1-й код G	G2000	→	1-й номер программы	O3400
2-й код G	G2001	→	2-й номер программы	O3401
3-й код G	G2002	→	3-й номер программы	O3402
.....				
5-й код G	G2049	→	50-й номер программы	O3449

Выделение группы кодов G, содержащих десятичную запятую к вызову системного макроса

С помощью следующих параметров можно выделить к макровыводу по каналам такой блок кодов G, код которых содержит десятичную запятую и одну десятичную цифру.

На параметре N1717 Start Dec G Macro задаётся начальное число блока кодов G, с десятичной запятой и одной десятичной цифрой,

На параметре N1718 Start Prg No. Dec G задаётся номер программы макроса, относящегося к коду G, заданному параметром Start Dec G Macro,

На параметре N1719 No. of Dec G Codes задаётся число элементов блока кодов G, содержащих десятичную запятую.

Если No. of Dec G Codes=0 на эти коды G не вызывается макрос.

Пример

Если начальный код группы например, G310.5, параметр заполним следующим образом: Start Dec G Macro=310.5.

Если код G310.5 вызывает программу с номером O4000, тогда Start Prg No. Dec G=4000.

На параметре No. of Dec G Codes задаётся число кодов G, относящихся к группе. Если в группе имеется 10 кодов, тогда No. of Dec G Codes=10.

После этого сопоставление кодов G с номерами программ:

1-й код G	G310.5	→	1-й номер программы	O4000
2-й код G	G310.6	→	2-й номер программы	O4001
3-й код G	G310.7	→	3-й номер программы	O4002
.....				
10-й код G	G311.4	→	10-й номер программы	O4009

Преобразование вызовов кода G в модальный

Если при индивидуальных вызовах для индивидуальных кодов G, при блочных вызовах **на параметр** N1714 Start G Macro, или N1717 Start Dec G Macro **задаётся отрицательное значение**, тогда выделённый **код G, или группа G** производит **наследственный (модальный) вызов**. Например, если G(9011)=120, тогда в программе команда G120 приводит к модальному вызову. Тип вызова определяется состоянием параметра

N1755 Macro Contr #0 MEQ=0: вызов G типа G66

N1755 Macro Contr #0 MEQ=1: вызов G типа G66.1

Если значение параметра 0, макрос вызывается в конце каждого кадра движения. Если значение параметра 1, макрос вызывается на каждый кадр.

Ссылка в дереве макроса G на тот же самый код G

Если **стандартный код G выделить к пользовательскому вызову** (например, G01) и **в дереве макроса** ссылаться опять на этот код, эта ссылка уже не приводит к повторному макровывозу, а устройство ЧПУ **истолкует его обычным кодом G** и выполняет его.

Если **не стандартный код G выделить к пользовательскому вызову** (например G101) и **в дереве макроса** ссылаться опять на **вызывающий код G**, (нашем случае на G101) **устройство ЧПУ выводит сообщение об ошибке** “2123 Не осуществлённая функция: G<номер>”.

В дереве макроса G ссылка M, S, ..., в дереве вызова M, S, ... ссылка G

- из системного макровывоза G системный вызов подпрограмм/макроса M, S, T, A, B, C, ASCII,
- из системного вызова подпрограмм/макроса M, S, T, A, B, C, ASCII системный вызов макроса G,

в зависимости от позиции параметра разрешается:

N1755 Macro Contr #1 ENC=0: не иницирует вызов (выполняется как обычный код M,S, ...G),

N1755 Macro Contr #1 ENC=1: иницирует вызов.

Область аргументов системных макросов G:

- если код типа G65 или G66, область аргументов, назначенная к G65, а также P и L,
- если код типа G66.1, тогда к области аргументов действуют изложенные там.

Удаление модального вызова производится **командой G67**.

23.3.5 Вызов системного макроса по коду М, заданному параметром

Можно выделить индивидуальные коды М, далее один блок кодов М, задавая параметром, по каналам, к макровывозу.

Выделение 10 индивидуальных кодов М к вызову системных макросов

В поле параметров можно выделить по каналам не более 10 разных кодов М, на которые желаем инициировать макровывозы. При этом вместо ряда команд

Nn G65 Pr <выделение аргументов>

надо писать ряд команд

Nn **Mm** <выделение аргументов>

В этом случае **код Mm не передаётся для PLC**, а вызывается макрос с соответствующим номером программы.

В поле параметров надо установить, что какой номер программы должен вызывать вызывающий код М. Эти параметры следующие:

N1733 M(9020): код М, который вызывает программу с номером O9020

N1734 M(9021): код М, который вызывает программу с номером O9021

:

N1742 M(9029): код М, который вызывает программу с номером O9029

Если на параметр написать 0, не вызывается макрос с данным номером программы.

Выделение группы кодов М к вызову системных макросов

С помощью следующих параметров можно выделить к макровывозу по каналам блок кодов М.

На параметре N1743 Start M Macro задаётся начальное число блока кодов М, десятичным целым числом,

На параметре N1744 Start Prg No. M Macro задаётся номер программы макроса, относящегося к коду М, заданного параметром Start M Macro,

На параметре N1745 No. of M Macro Codes задаётся число элементов блока кодов М

Если No. of M Macro Codes=0, не вызывается макрос на эти коды М.

Пример

Пусть будет например, начальный код группы M500. Надо установить Start M Macro=500. Если код M500 вызывает программу с номером O3500, тогда Start Prg No. M Macro=3500.

На параметре No. of M Macro Codes задаётся число кодов М, относящихся к группе. Если в группе имеется 20 кодов, тогда No. of M Macro Codes=20.

После этого сопоставление кодов М с номерами программ:

1-й код М	M500	→	1-й номер программы	O3500
2-й код М	M501	→	2-й номер программы	O3501
3-й код М	M502	→	3-й номер программы	O3502
.....				
20-й код М	M520	→	20-й номер программы	O3519

Положение в кадре кода М, запускающего макровывоза

Код М, выделённый в поле параметров, иницирующий макровывоз может опередить в кадре только / и алрес N (номер кадра), вводные аргументы могут следовать только после этого.

Макрос, вызванный на код М, не сохраняется модально

Кодом М задаётся всегда *модально не сохраняющийся вызов*, типа G65.

Ссылка в дереве макроса М на тот же код М

Если **в дереве макроса** повторно *ссылаться на тот же код М*, макрос не вызывается повторно, а **код М передаётся для PLC**.

В дереве макроса М ссылка G, S, ..., в дереве вызова G, S, ... ссылка М

- из макровызова, запускаемого на код М системный вызов подпрограмм/макроса G, S, T, A, B, C, M, ASCII,
- из вызова системной подпрограмм/макрос G, S, T, A, B, C, M, ASCII вызов макроса, запускаемого на код М,

зависимости от позиции параметра разрешается:

N1755 Macro Contr #1 ENC=0: не инициирует вызов выполняется как обычный код M, S, ...G),

N1755 Macro Contr #1 ENC=1: инициирует вызов.

Область аргументов кодов М, запускающих макровызов:

В кадре, содержащего *макровызов, запущенный на код М можно использовать обе области аргументов*, и №1 и №2.

После задания вызывающего кода М, второй код М истолкуется уже как аргумент и код передаётся переменной #13.

23.3.6 Вызов системной подпрограммы по коду М, заданному параметром

Можно выделить индивидуальные коды М, далее один блок кодов М, задавая параметром, по каналам, к вызову подпрограммы.

Разница между кодами М, вызывающими подпрограмму, или макрос имеется в том, что код М, вызывающий подпрограмму не передаёт никакого аргумента для подпрограммы, а вызывающий макрос код М передаёт.

Выделение 10 индивидуальных кодов М к вызову системной подпрограммы

В поле параметров можно выделить по каналам не более 10 разных кодов М, на которые желаем вызывать подпрограмму. В этом случае **код М не передаётся для PLC**, а подпрограмма вызывается с соответствующим номером программы, то есть вместо команды

Nn Gg Xx Yy M98 Pp

можно выдавать следующую команду:

Nn Gg Xx Yy **Mm**

В поле параметров надо установить, что какой номер программы должен вызвать вызывающий код М. Эти параметры следующие:

N1720 M(9000): код М, который вызывает программу с номером O9000

N1721 M(9001): код М, который вызывает программу с номером O9001

:

N1729 M(9009): код М, который вызывает программу с номером O9009

Если на параметр написать 0, не вызывается подпрограмма с данным номером программы.

Выделение группы кодов М к вызову системной подпрограммы

С помощью следующих параметров можно выделить один блок кодов М к вызову подпрограммы по каналам.

На параметре N1730 Start M SubP задаётся начальное число блока кодов М, десятичным целым числом,

На параметре N1731 Start Prg No. M SubP задаётся номер программы подпрограммы, относящегося к коду М, заданным параметром Start M SubP,

На параметре N1732 No. of M Codes задаётся число элементов блока кодов М.

Если No. of M Codes=0, не вызывается подпрограмма на эти коды М.

Пример

Пусть будет например, начальный код группы M600. Надо установить Start M SubP=600. Если код M600 вызывает подпрограмму с номером O3600, тогда Start Prg No. M SubP=3600.

На параметре No. of M Codes задаётся число кодов М, относящихся к группе. Если в группе имеется 30 кодов, тогда No. of M Codes=30.

После этого сопоставление кодов М с номерами программ:

1-й код М	M600	→	1-й номер программы	O3600
2-й код М	M601	→	2-й номер программы	O3601
3-й код М	M602	→	3-й номер программы	O3602
.....				
30-й код М	M630	→	30-й номер программы	O3629

Положение в кадре кода М, запускающего вызов подпрограммы

Код М можно написать в кадре на произвольное место.

Ссылка в дереве подпрограммы М на тот же код М

Если **в подпрограмме** повторно **ссылаться на тот же код М**, подпрограмма не вызывается повторно, а **код М передаётся для PLC**.

В дереве подпрограммы М ссылка G, S, ..., в дереве вызова G, S, ... ссылка М

- из вызова подпрограммы, запускающейся на код М, вызов системной подпрограммы/макроса G, S, T, A, B, C, M, ASCII,
- из вызова системной подпрограммы/макроса G, S, T, A, B, C, ASCII, вызов подпрограммы, запускающейся на код М,

в зависимости от позиции параметра разрешается:

N1755 Macro Contr #1 ENC=0: не инициирует вызов (выполняется обычным кодом M,S, ...G),

N1755 Macro Contr #1 ENC=1: инициирует вызов.

23.3.7 Вызов системной подпрограммы по коду A, B, C, S, T, разрешённому параметром

По коду A, B, C, S, T можно вызывать подпрограмму по каналам, разрешив параметром.

Выделение кода A, B, C, S, T к вызову подпрограммы

Параметр N1746 ABCST

при положении бита #0 AM=1 по коду A вызывается подпрограмма O9030.nct,
при положении бита #1 BM=1 по коду B вызывается подпрограмма O9031.nct,
при положении бита #2 CM=1 по коду C вызывается подпрограмма O9032.nct,
при положении бита #3 SM=1 по коду S вызывается подпрограмма O9033.nct,
при положении бита #4 TM=1 по коду T вызывается подпрограмма O9034.nct..

В случае адресов, выделённых к вызову подпрограммы, записанное в программу значение A, B, C, S, T не передаётся ни интерполятору (если адрес A, B, или C выделён для оси), ни для PLC, а код A, B, C, S, T приводит к вызову перечисленных выше подпрограмм.

При этом, если например, код T выделён для вызова подпрограммы, кадр

Gg Xx Yy Tt

эквивалентный к следующим двум кадрам:

#199=t

Gg Xx Yy M98 P9034

Передача аргумента кода A, B, C, S, T для подпрограммы

Значения, переданные адресам A, B, C, S, T записываются как аргументы в следующие **глобальные переменные**:

код A → #195

код B → #196

код C → #197

код S → #198

код T → #199

После этого подпрограмма может использовать эти переменные.

Положение в кадре кода A, B, C, S, T, запускающего вызов подпрограммы

Коды можно написать в кадре на произвольное место.

Ссылка в дереве подпрограммы A, B, C, S, T на вызывающий код

Если **в подпрограмме, вызываемой по адресу A, B, C, S, T** повторно **ссылаться на вызывающий адрес**, подпрограмма не вызывается повторно, а **код A, B, C, S, T передаётся интерполятору, или для PLC**.

В дереве подпрограммы A, B, C, S, T ссылка G, S, ..., в дереве вызова G, S, ... ссылка A, B, C, S, T

- из вызова подпрограммы, запускающейся по коду A, B, C, S, T, системный вызов подпрограммы/макроса G, M, S, T, A, B, C, ASCII (если вызов производится не из вызывающего адреса),
- из вызова системной подпрограммы/макроса G, M, S, T, A, B, C, ASCII вызов подпрограммы, запускающейся по коду A, B, C, S, T (если вызов производится не из вызывающего адреса),

в зависимости от позиции параметра разрешается:

N1755 Macro Contr #1 ENC=0: не инициирует вызов (выполняется обычным кодом M, S, ...G),

N1755 Macro Contr #1 ENC=1: инициирует вызов.

23.3.8 Вызов системной подпрограммы по коду ASCII, заданному параметром

На 4 разные коды ASCII, заданные параметром, можно вызывать подпрограмму по каналам. *Из кодов ASCII можно выбирать буквы английского алфавита.*

Выделение кодов ASCII к вызову подпрограммы

На параметре

N1747 ASCII Code SubP1

N1748 ASCII Code SubP2

N1749 ASCII Code SubP3

N1750 ASCII Code SubP4

можно установить 4 разных кодов (буквы английского алфавита). После этого устройство ЧПУ на эти коды вызывает подпрограммы с номером (Onnnn), установленные следующими параметрами:

N1751 Prg No. ASCII Call1

N1752 Prg No. ASCII Call2

N1753 Prg No. ASCII Call3

N1754 Prg No. ASCII Call4

Передача аргумента кодов ASCII для подпрограммы

Значения, заданные выделённым адресам ASCII, как аргументы записываются в следующие глобальные переменные:

1-й код ASCII → #191

2-й код ASCII → #192

3-й код ASCII → #193

4-й код ASCII → #194

После этого подпрограмма может использовать эти переменные.

Положение в кадре кода ASCII, запускающего вызов подпрограммы

Коды можно написать в кадре на произвольное место.

Ссылка на вызывающий код в дереве подпрограммы, вызывающейся по коду ASCII

Если в подпрограмме, вызывающейся по коду ASCII, повторно ссылаться на вызывающий адрес, подпрограмма не вызывается повторно, а код ASCII передаётся интерпретатору, или для PLC.

В дереве подпрограммы ASCII ссылка G, S, ..., в дереве ссылки G, S, ... ссылка ASCII

- из вызова подпрограммы, запускающейся по коду ASCII, вызов системной подпрограммы/макроса G, M, S, T, A, B, C, ASCII (если вызов производится не из вызывающего адреса),
- из вызова системной подпрограммы/макроса G, M, S, T, A, B, C, ASCII вызов подпрограммы, запускающейся по коду ASCII (если вызов производится не из вызывающего адреса),

в зависимости от позиции параметра разрешается:

N1755 Macro Contr #1 ENC=0: не инициирует вызов (выполняется обычным кодом M, S, ...G),

N1755 Macro Contr #1 ENC=1: инициирует вызов.

23.3.9 Отображение кадров макросов и подпрограмм в автоматическом режиме.

В основном случае кадры макросов и подпрограмм устройство ЧПУ внесёт в список в момент их выполнения. Однако, имеется возможность и запретить внесения кадров в список. При этом устройство ЧПУ считает за один кадр целый макрос, или подпрограмму и в режиме по кадрам не остановится у внутренних кадров.

Внесение в список можно разрешить, или запретить битами #0 MD8 и #1 MD9 параметра N1756 List Contr. Бит MD8 регулирует внесение в список пронумерованных от 8000 до 8999 подпрограмм и макросов, а бит MD9 - от 9000 до 9999.

Если значение параметра MD8, или MD9 будет 0, при выполнении пронумерованных от 8000 до 8999 подпрограмм и макросов, далее от 9000 до 9999 кадры подпрограмм и макросов не будут внесены в список, в режиме по кадрам выполнение не остановится у внутренних кадров.

При положении 1 параметра MD8, или MD9 их кадры тоже занесены в список, далее в режиме по кадрам устройство ЧПУ остановится и у внутренних кадров.

23.4 Макрос прерывания

Во время прогона программы деталей можно отложить выполнение актуальной программы, вызывать другую программу, затем после её выполнения продолжать отложенную программу.

Событиями, вызывающими прерывание, могут быть, например, отсутствие напряжения в сети, поломка инструмента, или сбор данных в определённых интервалах времени.

Программа, вызываемая при отложении называется **макросом прерывания**.

Вызов макроса прерывания запускается сигналом, выставленным из программы PLC а именно CP_MINT.

В программе деталей **вызов макроса прерывания надо разрешить**, иначе сигнал PLC не действует.

Помимо этого, надо выделить ту программу, которую желаем запускать под действием сигнала.

Команда

M96 P(номер программы)

или

M96 <имя файла>

разрешает работу сигнала прерывания CP_MINT, поступающего от программы PLC. Под действием сигнала прерывания **вызывается программа** с номером, **заданным** по адресу P (номер программы).

Правила программирования номера программ, заданная по адресу P изложены во главе “[14.4.1 Идентификация программ в накопителе. Номер программы \(O\)](#)” на странице [118](#). **Правила расширения макросов прерывания, заданных с номером программ и их размещения в накопителе** совпадают с правилами подпрограмм. Смотри главу “[14.4.2 Вызов подпрограммы \(M98\)](#)” на странице [118](#).

Место макросов прерывания в накопителе

Положение бита #5 SYM параметра N1758 Intrrt Contr решает, что где NC должен искать макрос прерывания в накопителе:

- в случае #5 SYM=0: всегда в той папке, в которой имеется главная программа, даже тогда, если макрос прерывания разрешили не в главной программе командой M96 P,
- в случае #5 SYM=1: всегда в каталоге, относящейся к соответствующему каналу папки SystemMacros.

Возвращение из макроса прерывания происходит **кодом M99**.

Команда

M97

запрещает вызов макроса прерывания, то есть макрос прерывания не вызывается больше, даже если поступает сигнал. Reset, далее конец программы тоже запрещает макрос прерывания.

Макрос прерывания работает исключительно в

автоматическом режиме, или
режиме ручного ввода данных,

когда он находится в состоянии

start.

Разрешение работы функции

Положение бита #0 USD=1 *параметра* N1758 Intrrt Contr *разрешает работу макроса прерывания* в устройстве ЧПУ.

Если значение бита #0 USD=0, макрос прерывания не работает, PLC свободно может использовать коды M96, M97 в качестве функции.

Разрешение прерывания другим кодом

При положении бита #4 MCD=0 параметра N1758 Intrrt Contr *разрешение/запрет* макроса прерывания происходит парой кодов *M96/M97*.

При положении бита #4 MCD=1 параметра N1758 Intrrt Contr разрешение/запрет макроса прерывания происходит отличающейся от пары кодов M96/M97.

При этом можно задавать *параметром* N1759 M Code MI On, что *какой код M должен разрешить* прерывание, а параметром N1760 M Code MI Off задаётся то, что *hogy какой код M должен выключить* прерывание.

Это может быть полезно тогда, когда PLC уже использовал код M96, или M97 для какой-то функции.

Прерывание типа подпрограмм, или типа макроса

При положении бита #1 STP=1 параметра N1758 Intrrt Contr прерывание будет *типа подпрограмм*, то есть вызов не увеличивает уровень локальных переменных, значение локальных переменных в вызванной программе то же самое, как в прерванной программе. В отличие от этого, при положении бита STP=0 прерывание будет *типа макроса*, уровень локальных переменных в вызванной программе увеличивается, то есть локальные переменные разными будут в вызывающей программе и в программе прерывания.

Прерывание с управлением набегавшей кромкой и уровней

С управлением набегавшей кромкой называется управление, когда макрос прерывания вызывается по сигналу *набегавшей кромки* CP_MINT PLC. Если сигнал остаётся включенным и в дальнейшем после возвращения из макроса прерывания, макрос не вызывается снова, только тогда, если сигнал выключается, затем снова включается обратно.

В отличие от этого, в случае прерывания *с управлением уровнем*, если после возвращения из макроса прерывания сигнал CP_MINT ещё включён, макрос прерывание *вызывается снова*, до тех пор, пока сигнал CP_MINT не выключается.

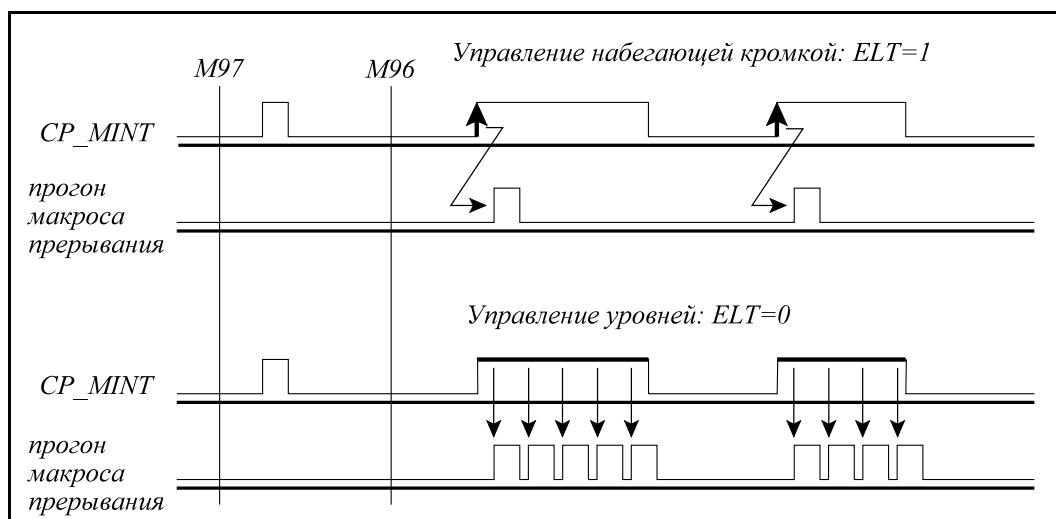


рисунок 23.4-1

Битом #3 ELT параметра N1758 Intrrt Contr решается, что какой должен быть макровывоз. Если #3 ELT=1, макровывоз будет с управлением набегающей кромкой, если #3 ELT=0, тогда с управлением уровнем.

Прерывание во время выполнения кадра, или в конце кадра

Пользователь может решать, что пусть начинается прогон макроса прерывания **в момент поступления сигнала**, или пусть дождётся, чтобы в актуальном кадре закончилось движение и прогон начался **только в конце кадра**.

При положении бита #2 TPI=0 параметра N1758 Intrrt Contr макрос прерывание вызывается сразу, а при положении бита #2 TPI=1 только в конце кадра.

Модальные информации в макросе прерывания

Вызов макроса прерывания отличается от простого вызова подпрограммы.

В случае вызова подпрограммы (M98 Pr) подпрограмма наследует модально из вызывающей программы

#4001...#4130 действующие в предыдущем кадре модальные информации и

#4201...#4330 действующие в кадре в момент выполнения модальные информации.

Эти две информации естественно отличаются друг от друга, ведь из-за буферизации выполнение отстает от предварительной обработки кадров.

Отличается от этого положение в случае макроса прерывания.

Прерывание происходит во время выполнения программы, поэтому **макрос прерывания (M96 Pr) может прочитать модальные информации, действующие в кадре в момент выполнения, то есть может прочитать их в прерванном кадре на переменных**

#4401...#4530.

Модальные информации #4001...#4130, а также #4201...#4330 инициализируются после вступления в макрос прерывания.

В ходе выполнения макроса прерывания переменные #4001...#4130 и #4201...#4330 работают привычным образом, а в то же время переменные #4401...#4530 сохраняют модальные информации, действующие в момент прерывания.

Возвращение из макроса прерывания с M99

Устройство ЧПУ *сохраняет модальные информации, действующие в момент прерывания*, затем *после возвращения с M99 восстанавливает сохранённые модальные информации*.

Если прерывать программу во время интерполяции, после возвращения устройство ЧПУ заканчивает интерполяцию, если прерывать в конце кадра, берёт следующий кадр и выполняет его.

Поэтому в том случае, если в макросе прерывания нужно запрограммировать и движение, целесообразно после вступления в макрос прерывания сохранить позицию конца кадра (#5001...), затем перед возвращением встать обратно в эту позицию.

Возвращение из макроса прерывания с M99 Pr

Если из макроса прерывания возвращаться с командой M99 Pr, обработка продолжается от кадра с номером, заданным по адресу P прерванной программы.

После возвращения с M99 Pr не восстанавливаются сохранённые модальные информации, а продолжается программа с модальными информацией, сформированными в макросе прерывания.

23.5 Команды NC и макроса. Выполнение макрокадров

Кадры NC и макрокадры

В языке программирования различаются *кадры NC* и *макрокадры*.

Кадрами NC считаются кадры, описанные традиционными кодами G, M и т.д., даже тогда, если значения отдельных адресов могут принимать не только числовые значения, но и переменные, или формулы.

Макрокадрами считаются следующие кадры:

- кадры, содержащие команду присвоения значения: #i=#j
- кадры, содержащие условную команду, или команду организации цикла: IF, WHILE
- кадры, содержащие команду контроля: GOTO, DO, END
- кадры, содержащие макровывод: G65, G66, G66.1, G67, или те коды G, или M, которые запускают макровывод.
- кадры, содержащие вызов подпрограммы (запускающие подпрограмму на M98 P, или на A, B, C, S, T, M)
- кадры, содержащие код возвращения из подпрограммы, или из макроса (M99)

Синхронизация команд (G53)

При положении бита #6 MBM=1 параметра N1301 DefaultG2 (способ мультибуфера) подготовитель кадров прочитает вперёд как кадры NC, так и макрокадры, обрабатывает их, и вставит их в уравнивательный накопитель (буфер). В ходе выполнения программы исполнители, интерполятор и PLC, берут кадры из уравнивательного накопителя. Это необходимо для того, чтобы интерполятор мог непрерывно перемещать оси и не понадобилось ожидать обработку следующего кадра.

В результате обработки кадров вперёд и буферизации исполнитель отстаёт от подготовителя кадров на много сотен кадров. Например, подготовитель кадров обрабатывает уже 1500-й кадр, но исполнитель выполняет лишь 1000-й кадр..

Это является причиной того, что различаем переменные #4001...#4130 (#_BUFx) и #4201...#4330 (#_ACTx). Пока первые дают информацию о кадрах, вставленных в уравнивательный накопитель последними, в то же время последние дают информацию о кадрах, выполняемых в тот момент.

Бывают случаи, когда приходится отложить предварительную обработку кадров и буферизацию кадров, например, если желаем дождаться данную позицию какой-то оси во время выполнения программы.

Команда

G53,

написанная в отдельный кадр, отложит прочтение вперёд кадров. Дождётся, пока буфер кадров опорожняется, и только после этого приступает к прочтению и обработке следующего кадра.

Выполнение макрокадров

Устройство ЧПУ может выполнять макрокадры параллельно с выполнением кадров NC, или вслед за ними. Параметром, регулирующим выполнение кадров NC и макрокадров является N1755 Macro Contr #2 SBM. Если параметр:

- =0: кадры NC и макрокадры выполняются в порядке, описанном в программе,
- =1: во время выполнения кадров NC выполняются макрокоманды.

Пример:

SBM=0

```
%O1000
...
N10 #100=50
N20 #101=100
N30 G1 X#100 Y#101
N40 #100=60 (присвоение значения
после N30)
N50 #101=120 (присвоение значения
после N30)
N60 G1 X#100 Y#101
```

Присвоение значения, описанное в кадрах N40 и N50 выполняется после выполнения кадра N30.

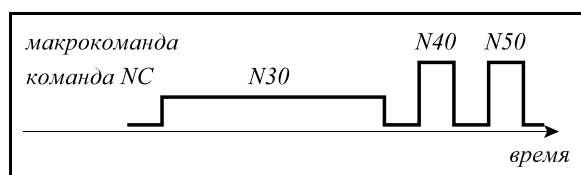


рисунок 23.5-1

SBM=1

```
%O1000
...
N10 #100=50
N20 #101=100
N30 G1 X#100 Y#101
N40 #100=60 (присвоение значения во
время N30)
N50 #101=120 (присвоение значения во
время N30)
N60 G1 X#100 Y#101
```

Присвоение значения, описанное в кадрах N40 и N50 выполняется во время движения в кадре N30.

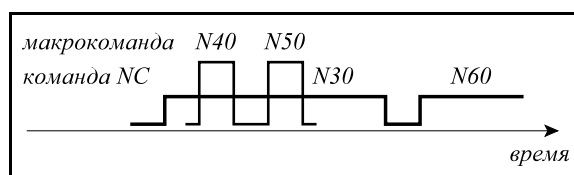


рисунок 23.5-2

☞ Последствия:

- выполнение программы медленнее,
- если прервать выполнение кадра N30, затем снова запускать обработку, продолжение обработки проигнорируется, поскольку кадр N40, N50 ещё не изменил переменные кадра N30.

☞ Последствия:

- выполнение программы быстрее,
- если прервать выполнение кадра N30, затем снова запускать обработку, обработку нельзя продолжать, только тогда, если запустить поиск кадра на кадр N30, поскольку кадр N40, N50 уже изменил переменные кадра N30.

23.6 Макроцикл фрезерования полости

Команда

G65 P9999 X Y Z I J K R F D E Q M S T

запускает цикл фрезерования полости.

Перед вызовом цикла надо разместиться над геометрическим центром полости в выбранной плоскости, на безопасное расстояние от поверхности заготовки. В конце цикла инструмент отводится в эту же точку.

Источование адресов кадра:

X: размер полости в направлении X

Y: размер полости в направлении Y

Z: размер полости в направлении Z

Команда G17, G18, G19 решает, что из трёх координат какая будет длиной, шириной или глубиной полости. Например: в случае G17 глубина полости Z, из X и Y длиннее координата будет длиной полости, а короче шириной. Эти значения задаются как положительные числа в абсолютном значении.

R: радиус углов полости

По адресу R надо задавать возможные округления углов полости. Если не заполнять адрес R, округление углов полости будет равняться радиусом инструмента.

I: безопасное расстояние в направлении глубины в случае G19

J: безопасное расстояние в направлении глубины в случае G18

K: безопасное расстояние в направлении глубины в случае G17.

В зависимости от выбранной плоскости по адресу I (G19), J (G18), или K (G17) надо задавать в кадре безопасный припуск в направлении инструмента. Устройство ЧПУ предполагает при запуске цикла, что вершина инструмента находится на такое расстояние от поверхности заготовки. А во время профрезерования полости, когда завершилась обработка данного уровня, на это расстояние отводится инструмент, чтобы для обработки следующего уровня стать в начальную точку.

D: адрес ящика, содержащего коррекцию радиуса инструмента

Номер регистра коррекции радиуса инструмента, использованного в программе, обязательно задавать по адресу D. Вообще фрезерование полости надо выполнить в состоянии G40.

E: ширина резания в процентах диаметра фрезы

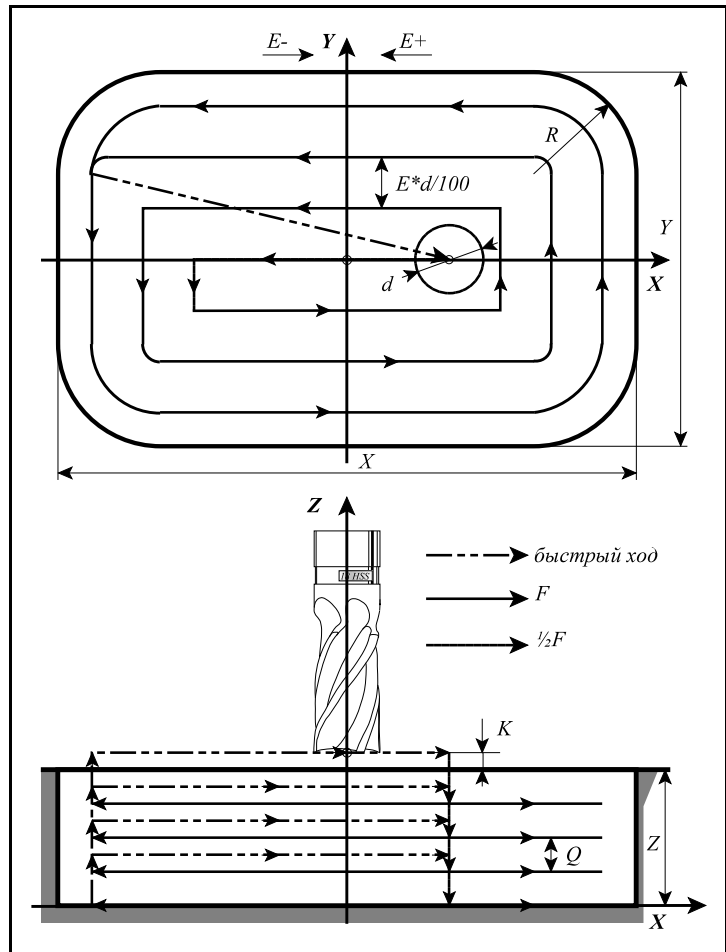


рисунок 23.6-1

со знаком $+$: обработка против хода часовой стрелки

со знаком $-$: обработка по ходу часовой стрелки

По адресу E можно сообщить устройству ЧПУ две информации. Значение E задаёт, что какая должна быть ширина резания в процентах диаметра фрезы. Если оно не задано, устройство ЧПУ автоматически предполагает $+83\%$. Устройство ЧПУ изменяет данные, заданные по адресу E в зависимости от ширины полости так, чтобы при профрезеровании одного уровня значение резания было равномерное. Однако изменение может быть только в сторону уменьшения. Знак адреса E подскажет направление фрезерования. Если E+, то есть положительное, обработка совершается против хода часовой стрелки, если E-, то есть отрицательное, обработка совершается по ходу часовой стрелки.

Q: глубина резания

По адресу Q задаётся глубина резания в применённой системе мер, то есть в мм-ах, или в дюймах. Устройство ЧПУ в зависимости от глубины полости может пересматривать запрограммированное значение в интересах равномерного разделения глубины резания. Однако изменение может быть только в сторону уменьшения.

F: подача

По адресу F задаётся величина подачи, применённой в ходе цикла. Если адресу F не задавать значение, принимается во внимание модальное значение F. Принимается 50%-ов значение F в следующих случаях:

- Когда начнётся снятие одного уровня и инструмент просверлит до глубины Q в направлении инструмента,
- При фрезеровании полости в продольном направлении до тех пор, пока инструмент загружен на обоих сторонах.

M S T: функция

В кадре, вызывающего фрезерование полости можно задавать одну функцию M, или S, T, которую устройство ЧПУ выполняет перед началом фрезерования.

Экстремистские случаи фрезерования полости:

Если не задана ширина полости, берётся двойной радиус углов полости и это будет шириной полости.

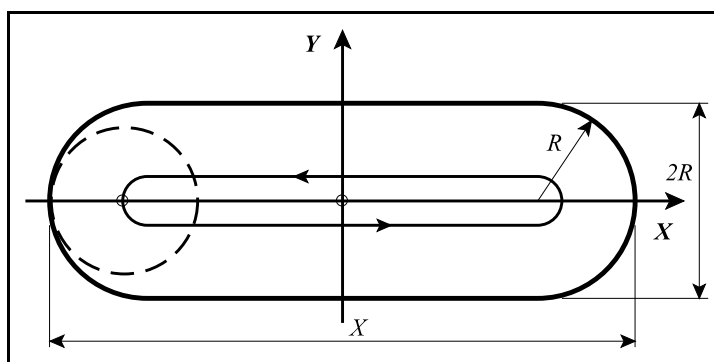


рисунок [23.6-2](#)

Если не задана ни ширина полости, ни радиус округления углов, за шириной полости (паз) берётся диаметр применённого инструмента.

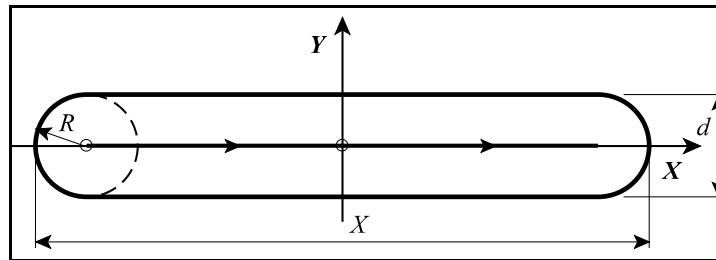


рисунок 23.6-3

Если не задана ни длина, ни ширина полости, только запрограммирован адрес R, тогда снимается круглая полость с радиусом R.

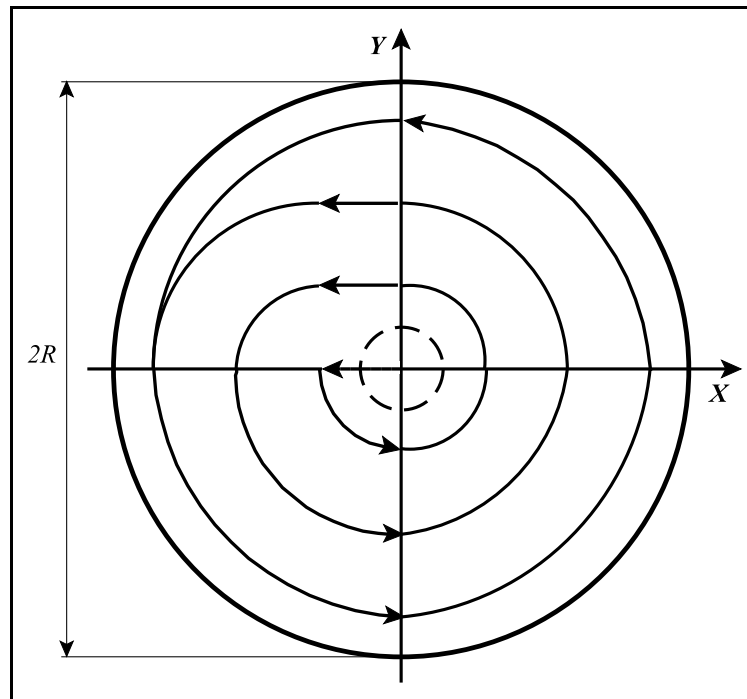


рисунок 23.6-4

Если не задана ни длина, ни ширина, ни радиус полости, тогда цикл превращается в сверление.

Сообщения об ошибке, выводимые в ходе выполнения фрезерования полости:

MACRO ERROR 1: ошибка заполнения кадра. Возможные причины:

- Не задана глубина полости,
- Не задан радиус инструмента,
- не задана глубина резания.

MACRO ERROR 2: ошибка задания размера. Возможные причины:

- Если размер, заданный для длины, или для ширины полости меньше, чем двойной радиус полости,
- Если размер полости по длине, или по ширине меньше, чем вызванный по адресу D диаметр инструмента,

- Если значение, заданное для ширины резания 0, или вызванный диаметр инструмента 0,
- Если значение глубины резания 0, то есть на адрес Q запрограммировано 0.

24 Писание и чтение параметров

С помощью параметров можно установить работу устройства ЧПУ согласно требованиям. Параметры сохраняются в незабывающей памяти устройства ЧПУ. Параметры устройства ЧПУ можно переписать, и прочесть из программы деталей.

Каждый *параметр* имеет одно, не более **четырёхзначное опознавательное число**, которое идентифицирует параметр в накопителе. К этим **опознавательным числам может принадлежать одно** (в случае глобальных параметров), **или несколько** различных (по каналам, по осям, индексированных по шпинделям) **значений**. Параметры можно классифицировать **по их изображению цифер, по их кругу действия и способу их взятия на учёт**.

Классификация параметров по их изображению цифер

Параметры битного типа: их область значений может быть 0, или 1.

Параметры целого типа (DWORD): их область значений может быть без знака: 0,...4294967295, или со знаком: -2147483648... +2147483647.

Параметры с плавающей запятой (double): их область значений может быть в случае отрицательных чисел: $-1.7 \times 10^{308} \dots -5.0 \times 10^{-324}$ в случае положительных чисел: $5.0 \times 10^{-324} \dots 1.7 \times 10^{308}$.

Классификация параметров по их кругу действия

Глобальные параметры: глобальными называем те параметры, которые действительны во всех каналах устройства ЧПУ. Такими параметрами являются например, N2201 Waiting M Codes Min и N2202 Waiting Codes Max, которые назначают группу кодов M, регулирующих доживания между каналами и действительны для всех каналов. В случае глобальных параметров к каждому опознавательному числу принадлежит **одно** значение параметра.

Параметры, задаваемые по каналам: это такие параметры, на которые можно написать разные по каналам значения. Таким параметром является например, группа N1500 Return Val G73, на которой можно устанавливать расстояние отвода в цикле сверления G73, и на которую можно написать разные по каналам значения. В случае параметров, задаваемых по каналам, к каждому опознавательному числу может принадлежать **не более 8** значений параметра.

Параметры, задаваемые по осям: это такие параметры, на которые можно написать разные по осям значения. Таким параметром является например, бит #0 DIA группы N0106 Axis properties, на котором по осям можно устанавливать, что данную ось желаем программировать в радиусе, или в диаметре. В случае параметров, задаваемых по осям, к каждому опознавательному числу может принадлежать **не более 32** значений параметра.

Параметры, задаваемые по шпинделям: это такие параметры, на которые можно написать разные по шпинделям значения. Таким параметром является например, группа N0608 Spindle Encoder Counts, на котором по шпинделям можно задавать число импульсов датчика, оборудованного на отдельные шпиндели. В случае параметров, задаваемых по шпинделям, к каждому опознавательному числу может принадлежать **не более 16** значений параметра.

Классификация параметров по их способу взятия на учёт

Параметры, взятые на учёт во время прогона: Эти параметры после их изменения устройство ЧПУ сразу принимает во внимание.

Параметры, взятые на учёт после перезагрузки: Эти параметры после их изменения устройство ЧПУ принимает во внимание только после перезагрузки (после выключения, затем включения).

Параметры, писаемые в случае аварийного состояния: Изменение этих параметров устройство ЧПУ разрешает только в аварийном состоянии.

Подробное описание отдельных параметров изложены в описании “Устройство ЧПУ NCT3xx инструментальными станками Параметры”.

24.1 Писание параметров из программы деталей (G10 L52)

Команда

G10 L52 (писание параметра вкл.),

написанная в отдельной строке, включает функцию писания параметра. После этого в отдельной строке надо задавать номера и значения отдельных параметров следующим образом:

N_ R_ (писание глобальных параметров)

N_ P_ R_ (писание параметра канала/оси/шпинделя)

...

...

Конец писания параметра закрывается командой

G11 (конец писания параметра)

написанная в отдельной строке.

Истолкование адресов N, P, Q, R:

N: опознавательное число параметра (0-9999). Ведущими нулями можно пренебречь.

☞ **Внимание!** Командой G10 L52 можно изменить только параметр с таким опознавательным числом, для взятия на учёт которого не требуется перезагрузка, или для изменения которого не требуется аварийное состояние!

P: номер канала (1-8), оси (1-32), или шпинделя (1-16). В скобках задаётся, что в отдельных случаях какое значение P можно задавать.

☞ **Внимание!** Если писать параметр, задаваемый по каналам, и не заполнять адрес P, командой изменяется всегда параметр того канала, в котором работает программа..

Q: в случае битного параметра число писаемого бита задано от 0 до 7,

R: значение параметра. Инкрементный оператор I, принятый при значении R, который инкрементирует актуальное значение параметра со значением, заданным по адресу RI. В случае данных с плавающей запятой можно использовать десятичную запятую (.).

Параметры, изменённые командой G10 L52 сохраняются, значит, изменение будет действительное и при следующем включении.

При ошибочном заполнении адресов N, P, Q, R выводится сообщение об ошибке “2002 <N, P, Q, или R> данные вне предела значений”.

На адрес N: если нет параметра с таким опознавательным числом,

На адрес P: если нет канала/оси/шпинделя с таким номером,

На адрес Q: если значение Q меньше, чем 0, или больше, чем 7,

На адрес R: если на параметр с опознавательным числом N написано на R значение вне допускаемого предела значений.

Если какой-нибудь из данных N, P, Q, R отсутствует, выводится сообщение об ошибке “2004 <N, P, Q, R> отсутствуют данные”.

При желании изменить такой параметр, для взятия на учёт которого требуется перезагрузка, или для изменения которого требуется аварийное состояние, тогда выводится сообщение об ошибке “2193 Параметр Nnnnn нельзя изменить командой G10”.

Пример для писания параметра:

```

...
G10 L52                (включение писания параметра)
N107 P4 Q0 R1          (N0107 RollOver Control A4- REN)
N1339 (P1) R0.5        (N1339 Radius Diff)
N1746 (P1) Q1 R1       (N1746 ABCST BM=1)
G11                    (выключение писания параметра)
...

```

24.2 Чтение параметров из программы деталей (PRM)

Командой

#a=PRM[#b,#c]

и командой для присвоения значения

#a=PRM[#b,#c] / [#d]

значение *любого параметра* устройства ЧПУ *можно прочитать без ограничения*.

Значение аргументов:

#a: писаемая макропеременная.

#b: опознавательное число параметра. Можно задавать косвенно на макропеременной, или непосредственно с числовым значением.

#c: При чтении битного параметра это число бит параметра. Можно задавать косвенно на макропеременной, или непосредственно с числовым значением.

#d: номер канала (1-8), оси (1-32), или шпинделя (1-16). В скобках задано, что в отдельных случаях на #d какое значение можно задавать. Можно задавать косвенно на макропеременной, или непосредственно с числовым значением.

☞ **Внимание!**

Если читать параметр, задаваемый по каналам, и пропустить часть / [#d] из команды, команда читает параметр всегда того канала, в котором программа работает.

Если аргументы команды PRM задавать на макропеременных и значение макропеременной #0 (пустое), это имеет такое действие, как будто аргумент не задавали.

Аргументы могут быть и результатами макровыражений.

Если команда PRM стоит не на правой стороне команды для присвоения значения, выводится сообщение об ошибке “2017 Запрещённый адрес PRM”.

Если отсутствует аргумент #b команды PRM, то есть отсутствует номер параметра, выводится сообщение об ошибке “2064 Синтаксическая ошибка”.

Если отсутствует число бита #с команды PRM, или отсутствует аргумент /[#d], выводится сообщение об ошибке “2194 Функция PRM <2., 3.> отсутствует параметр”.

Пример для чтения параметров:

```
...
#101=PRM[107,0]/[4]      (N0107 RollOver Control A4- REN)
#102=PRM[1339]/[1]      (N1339 Radius Diff)
#103=PRM[1746,1]/[1])   (N1746 ABCST BM=1)
#104=PRM[2201]          (N2201 Waiting M Codes Min)
...
```

Записы