

**Краткое руководство пользователя программного обеспечения
«Симулятор токарного станка с ЧПУ»**

<https://sunspire.site/ru/>

1. Общее описание программного продукта

Симулятор токарного станка с ЧПУ представляет собой мультимедийное приложение, предназначенное для базового ознакомления начинающих специалистов машиностроительного профиля с принципами программирования операций токарной обработки деталей с использованием стандартного (ISO) G-кода.

Основная задача приложения – синтаксический анализ (парсинг) кода управляющих программ с целью построения графической модели траекторий режущего инструмента в трёхмерном пространстве.

Основные функции приложения:

- редактирование кода управляющих программ токарного станка;
- операции с файлами управляющих программ;
- настройка геометрических параметров режущего инструмента;
- непрерывное/пошаговое выполнение блоков управляющих программ;
- трёхмерная визуализация перемещений инструмента в рабочем пространстве станка;
- упрощённая визуализация обрабатываемой поверхности детали;
- расчёт режимов обработки;
- краткое справочное руководство по использованию G-кода.

Основные ограничения приложения: упрощённая модель поверхности резания, поддержка только двух управляемых осей, упрощённая модель элементов оснастки станка.

Тип целевого вычислительного устройства и поддерживаемая платформа: IBM-совместимый персональный компьютер под управлением Microsoft Windows, персональный компьютер Apple Macintosh под

управлением MacOS, мобильные устройства на базе операционных систем Android и iOS.

Графическая составляющая программного обеспечения использует компонентную базу OpenGL 2.0. Графический интерфейс пользователя программы реализован на русском, английском и китайском языках.

Многоплатформенная поддержка позволяет использовать программный продукт на различных вычислительных устройствах, включая интерактивные доски, смартфоны, планшетные и настольные компьютеры, что, в свою очередь, повышает гибкость и мобильность образовательного процесса, соответствуя современному уровню информатизации образования.

Минимальные системные требования:

- центральный процессор: Intel/AMD, не менее 2 ГГц;
- ОЗУ: не менее 1 Гб;
- видеопамять: не менее 512 Мб;
- разрешение экрана: не менее 1024x768x32;
- OpenGL 2.0 (а также Vulkan на мобильных платформах);
- DirectX версии 9.0.c (для ОС Microsoft Windows);
- стандартная клавиатура и компьютерная мышь с колесом прокрутки (сенсорный экран для мобильных платформ);
- средства воспроизведения звука (аудиоколонки или наушники).

Программное обеспечение имеет специальную версию, поддерживающую аппаратное взаимодействие с физическим ЧПУ-контроллером GSK 980 TDi. Данная версия программы поставляется в комплекте программно-аппаратного учебного комплекса. Взаимодействие программной и аппаратной составляющих учебного комплекса выполняется в соответствии с сетевым протоколом прикладного уровня ModBus. Прямое

подключение ЧПУ-контроллера к персональному компьютеру осуществляется посредством интерфейса RS-485.

2. Описание графического интерфейса пользователя

Главный экран приложения представляет собой интерактивную трёхмерную сцену с изображением геометрической модели токарного станка с ЧПУ, помещённой в условное окружающее пространство. Наблюдение за моделью осуществляется с помощью виртуальной камеры, ракурс которой можно изменять в процессе работы.

Наряду с трёхмерными графическими элементами на главном экране отображаются двумерные элементы управления – кнопки, переключатели и основная текстовая информация о моделируемых процессах (рис. 1).

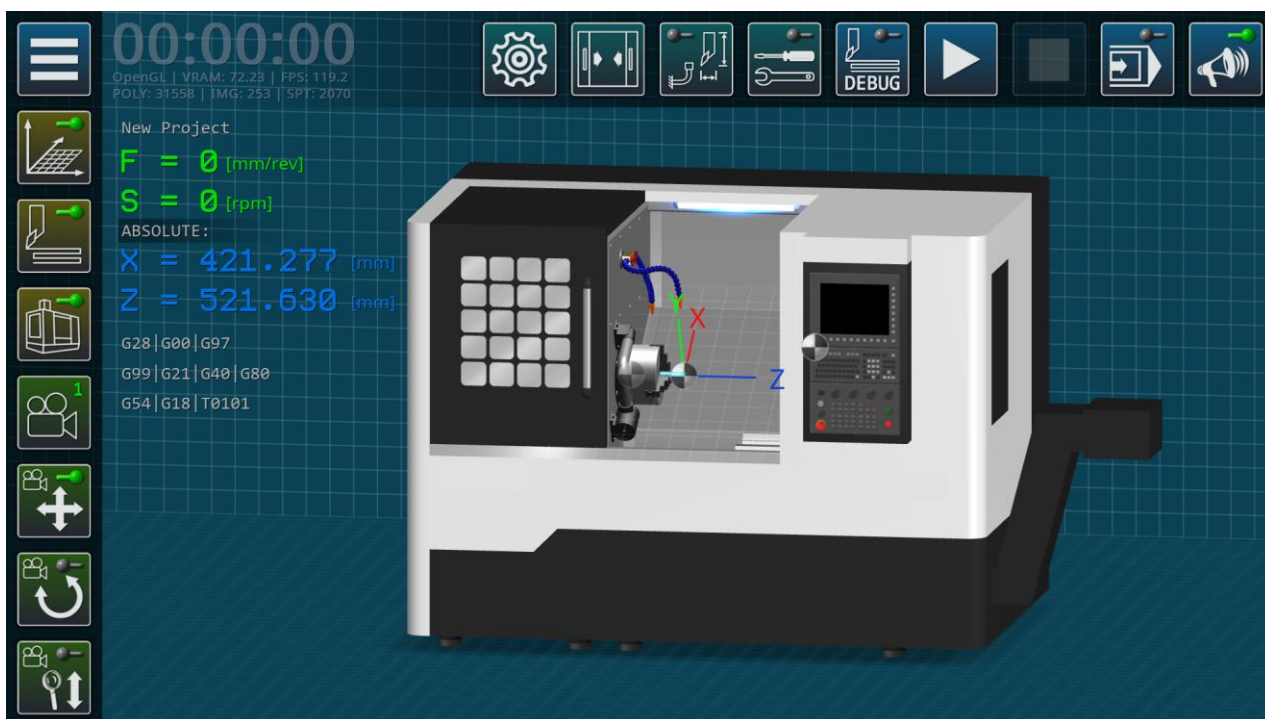


Рисунок 1 – Главный экран приложения

В левой части главного экрана расположена вертикальная панель основных функциональных кнопок (табл. 1).

Таблица 1 – Основные функциональные кнопки главного экрана



Изображение кнопки	Наименование функции	Описание функции
	Главное меню	Отображение главного меню приложения
	Система координат	Отображение/скрытие системы координат и координатной сетки рабочего пространства станка
	Траектории инструмента	Отображение/скрытие графической модели траекторий перемещения режущего инструмента
	3D модель станка	Отображение/скрытие комплексной графической модели элементов фрезерного станка
	Текущая камера	Переключение режима виртуальной камеры
	Режим смещения	Режим планарного смещения виртуальной камеры (для устройств с сенсорным экраном)
	Режим вращения	Режим вращения виртуальной камеры (для устройств с сенсорным экраном)
	Режим масштабирования	Режим приближения/отдаления виртуальной камеры (для устройств с сенсорным экраном)

В верхней части главного экрана расположена горизонтальная панель кнопок управления процессом симуляции и настроек (табл. 2).

Таблица 2 – Кнопки управления процессом симуляции и настроек

Изображение кнопки	Наименование функции	Описание функции
	Настройки приложения	Отображение панели настроек приложения для выбора локализации и параметров симуляции
	Автоматическая дверь	Изменение состояния автоматической двери токарного станка (открытие/закрытие)
	Режим привязки инструмента	Режим определения вылета инструмента с помощью измерительного датчика
	Дополнительная панель	Отображение дополнительной панели кнопок управления заготовкой
	Режим отладки	Быстрая отладка управляющей программы и построение полной модели траекторий инструмента
	Запустить	Запуск управляющей программы станка в режиме непрерывного выполнения кадров
	Приостановить	Приостановка управляющей программы станка
	Остановить	Прекращение выполнения управляющей программы станка

Таблица 2 – Продолжение




Изображение кнопки	Наименование функции	Описание функции
	Пошаговое выполнение	Включение/выключение режима пошагового выполнения управляющей программы станка
	Воспроизведение звуков	Включение/выключение звукового сопровождения процесса симуляции

Для управления заготовкой предусмотрены дополнительные функциональные кнопки (табл. 3).

Таблица 3 – Кнопки дополнительной панели для управления заготовкой

Изображение кнопки	Наименование функции	Описание функции
	Отрезанный фрагмент	Отображение/скрытие фрагмента заготовки, образованного в результате операции отрезки
	Восстановить заготовку	Возврат контура заготовки в исходное состояние
	Смена установка	Зеркальное отражение контура заготовки вдоль главной оси
	Сместить заготовку вправо	Смещение контура заготовки вправо

Таблица 3 – Продолжение

Изображение кнопки	Наименование функции	Описание функции
	Сместить заготовку влево	Смещение контура заготовки влево
	Сместить систему координат	Смещение активной системы координат в нулевые точки станка
	Сбросить систему координат	Возврат исходных заданных корректоров заготовки

При нажатии на кнопку «Главное меню» на экране отображается радиальное меню выбора вспомогательных диалоговых экранов приложения (рис. 2).



Рисунок 2 – Главное меню приложения

Кнопка 1 – «Создать проект» – предназначена для сброса текущих параметров симуляции. Данная функция используется, если после работы с открытым файлом необходимо начать новый проект управляющей программы. При нажатии на кнопку «Создать проект» на экране отображается модальный диалоговый экран подтверждения сброса текущих параметров симуляции.

Кнопка 2 – «Открыть проект» – предназначена для выбора и открытия файла управляющей программы станка. При нажатии на кнопку «Открыть проект» отображается экран открытия файла (рис. 3), в основной области которого выводится список каталогов и список файлов управляющих программ, записанных в рабочий каталог проектов.

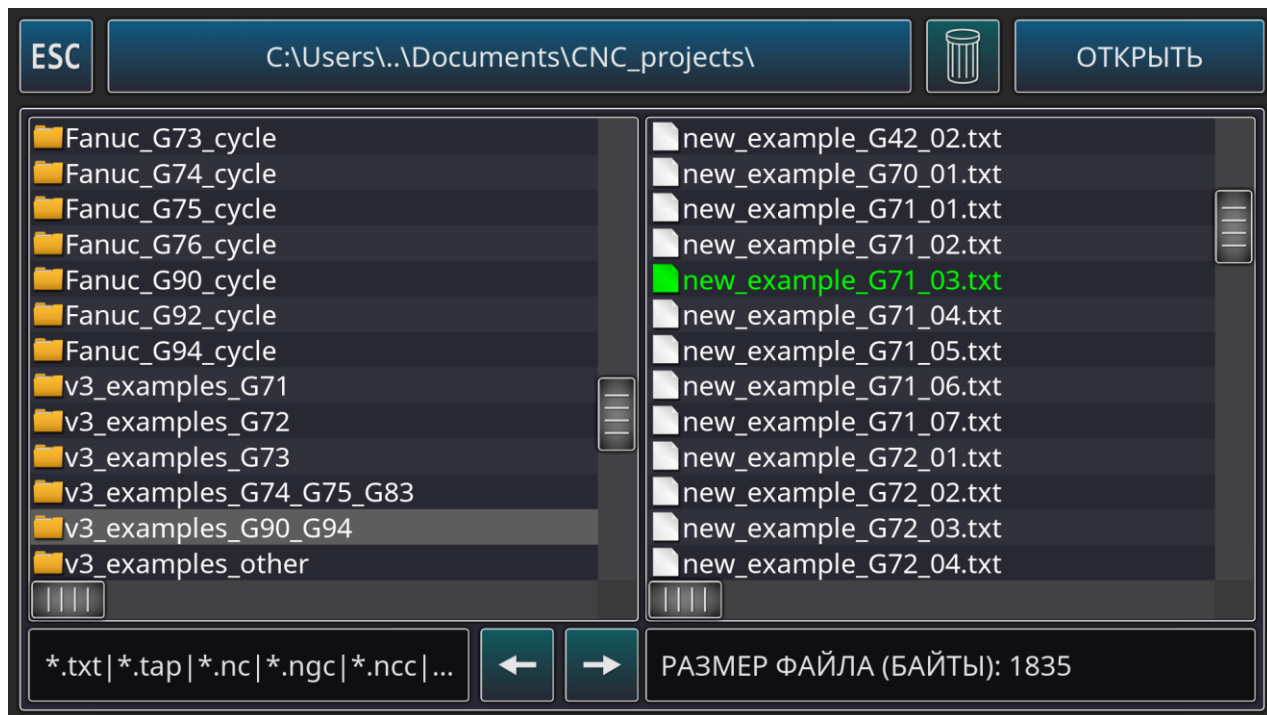


Рисунок 3 – Экран выбора открываемого файла

В нижней части экрана открытия файла расположен фильтр типов файлов, позволяющий отображать в списке файлы заданного формата. Симулятор поддерживает 6 форматов файлов управляющих программ: *.txt, *.tap, *.nc, *.ngc, *.ncc, *.csdata. В правой нижней части экрана отображается размер выбранного файла в байтах. Выбранный файл в списке выделяется

зеленым цветом. Если файл выбран, в верхней правой части экрана активизируется кнопка «Открыть».

Также в верхней правой части экрана присутствует кнопка «Удалить файл», которая становится активной при выделении файла. В мобильной версии приложения в верхней части экрана открытия файла отображается дополнительная кнопка «Отправить», позволяющая отправить выбранный файл с помощью внешнего приложения (например, почтового клиента).

Кнопка 3 – «Сохранить проект» – предназначена для выбора локального пути и имени сохраняемого файла управляющей программы станка. При нажатии на кнопку «Сохранить проект» отображается экран сохранения файла (рис. 4), принцип взаимодействия с которым аналогичен экрану открытия файла. Имя сохраняемого файла вводится в специальное текстовое поле, расположенное в правой нижней части экрана, без указания расширения. Расширение файла выбирается с помощью переключателя в нижней левой части экрана.

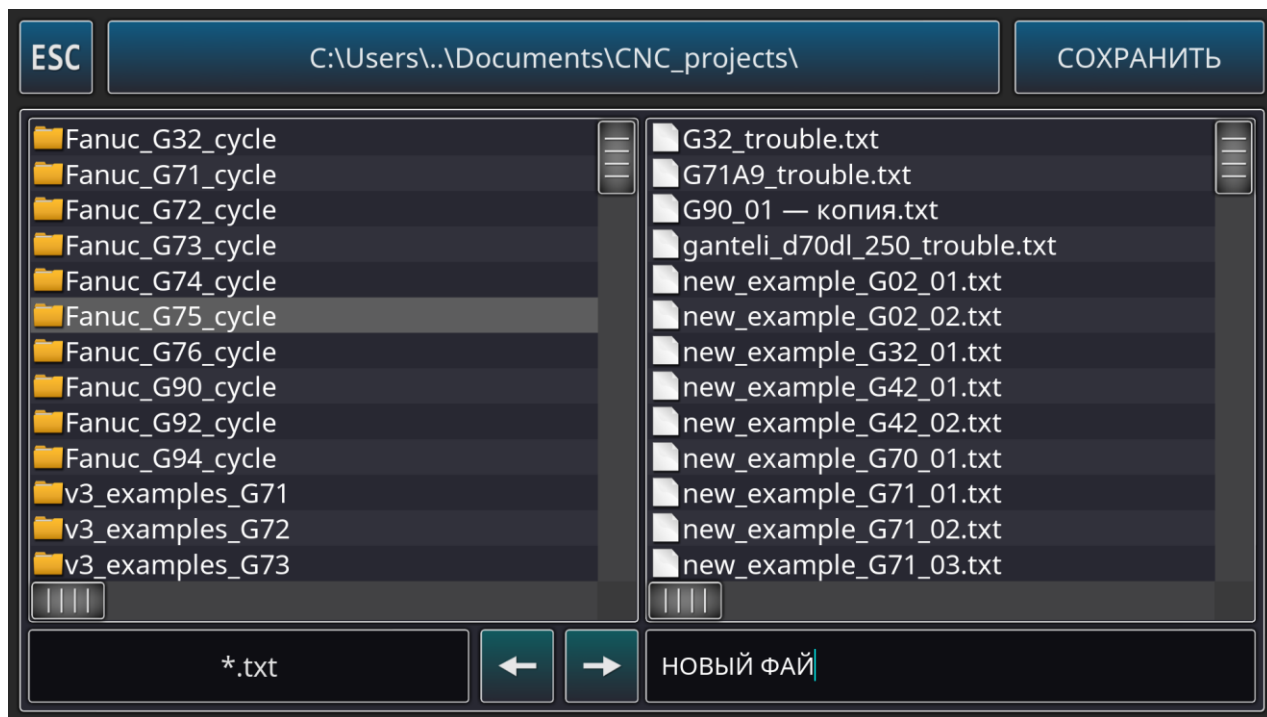


Рисунок 4 – Экран сохранения файла

Файлы управляющих программ и вложенные каталоги стандартных примеров управляющих программ расположены в каталоге «C:\Users\...\Documents\CNC_projects» в системе Microsoft Windows. В системе Android путь расположения файлов управляющих программ по умолчанию: «/storage/emulated/0/CNC_projects» или «/storage/emulated/0/Android/data/com.virtlab.cncsim/files/CNC_projects» (для поздних версий ОС Android). В системах iOS и MacOS файлы управляющих программ хранятся в рабочем каталоге приложения.

Кнопка 4 – «Лицензионная информация» – предназначена для отображения экрана вывода основной лицензионной информации о лицензиаре и лицензиате программного продукта.

Кнопка 5 – «Завершение работы» – предназначена для отображения диалогового экрана завершения работы симулятора.

Кнопка 6 – «Параметры заготовки» – предназначена для отображения экрана настройки параметров обрабатываемой заготовки (рис. 5).



Рисунок 5 – Экран настройки параметров заготовки

В левой части экрана параметров заготовки расположен блок числовых и строковых полей с кнопками увеличения/уменьшения значений. С помощью данных параметров можно задать геометрические размеры обрабатываемой заготовки и выбрать цвет моделируемой поверхности. В правой части экрана отображается чертёж заготовки с основными размерами и визуализацией способа закрепления заготовки. Чертеж заготовки обновляется адаптивно к изменению значений параметров.

В верхней части экрана расположены заголовочные кнопки, предназначенные для переключения режима настройки заготовки. По умолчанию при открытии диалогового экрана параметров заготовки осуществляется настройка основных геометрических параметров обрабатываемой заготовки. Кнопка «Системы координат» предоставляет возможность настройки относительного положения систем координат заготовки (рис. 6).

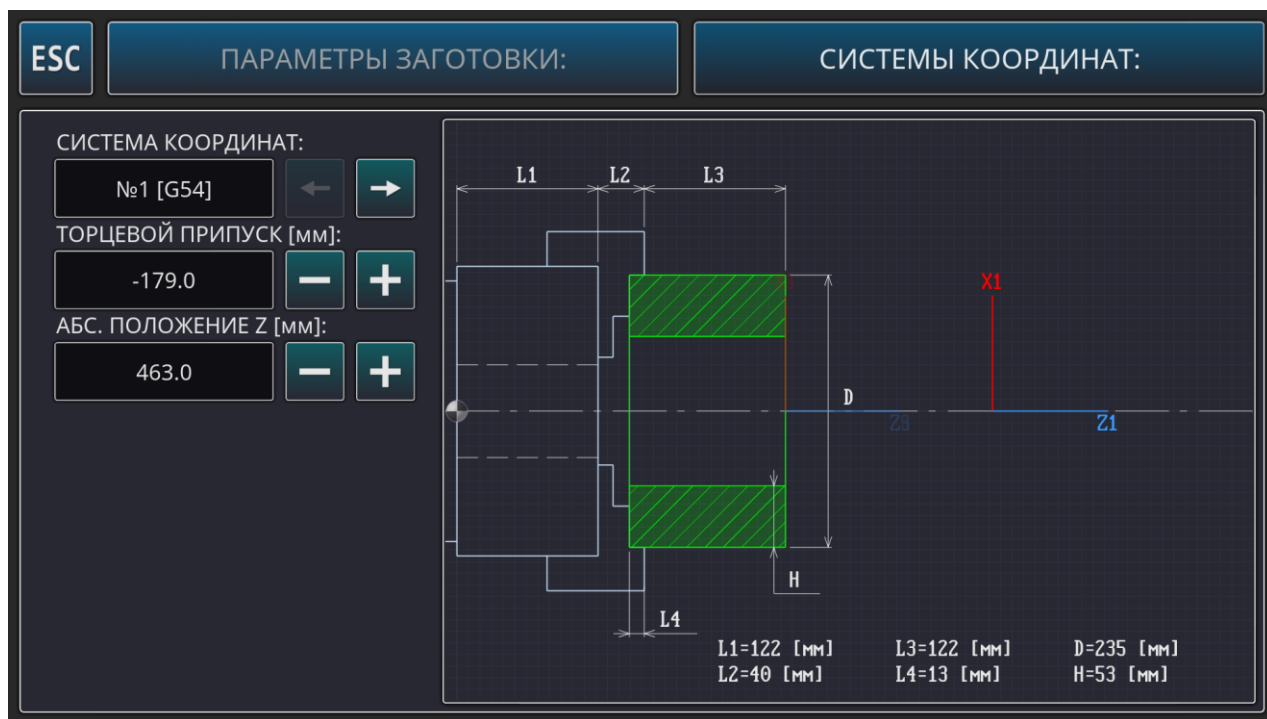


Рисунок 6 – Экран настройки систем координат

Для предварительной настройки доступны 6 рабочих систем координат заготовки. Дальнейшее переключение между рабочими системами координат в процессе симуляции осуществляется программно путём использования специальных кодов управления G54-G59. Координаты рабочих нулей можно задавать в абсолютных значениях (относительно точки машинного нуля), либо путём ввода величины торцевого припуска относительно правого торца заготовки.

Информация о параметрах заготовки сохраняется в начале файла управляющей программы в виде комментариев.

Кнопка 7 – «**Параметры инструмента**» – предназначена для отображения экрана настройки параметров режущего инструмента (рис. 7).



Рисунок 7 – Экран настройки режущего инструмента

В левой части экрана параметров инструмента расположен блок числовых и строковых полей с кнопками увеличения/уменьшения значений, а также переключения позиций. С помощью данных параметров можно задать

тип используемого в симуляции режущего инструмента, конструкцию, а также геометрические размеры рабочей части режущего инструмента.

В правой части экрана отображается упрощенный чертеж выбранного инструмента и его технологической оснастки с выводом основных размеров. Обновление чертежа осуществляется в реальном времени при изменении параметров инструмента.

В верхней части экрана расположены заголовочные кнопки аналогично экрану параметров заготовки. Кнопка «Таблица корректоров» отображает на экране сводную таблицу геометрических размеров всех используемых инструментов с соответствующими номерами корректоров, используемыми программной функцией переключения инструмента T0_0_.

Кнопка 8 – «Редактор управляющих программ» – предназначена для отображения текстового редактора кода управляющих программ на главном экране приложения (рис. 8).

Основная часть текстового редактора представлена списком строк управляющей программы, доступных для редактирования с физической клавиатуры. Редактирование текста управляющей программы станка осуществляется только в режиме остановленной симуляции. При выполнении процесса симуляции режим редактирования текста управляющей программы недоступен. При этом прокрутка текста управляющей программы осуществляется автоматически, а текущий выполняемый блок программы выделяется зеленым цветом.

В верхней части текстового редактора расположены вкладки, предназначенные для переключения управляющих программ текущего проекта (всего 5). Используя дополнительные вкладки, можно включать в управляющую программу внешние подпрограммы, нумерация которых соответствует номерам вкладок.

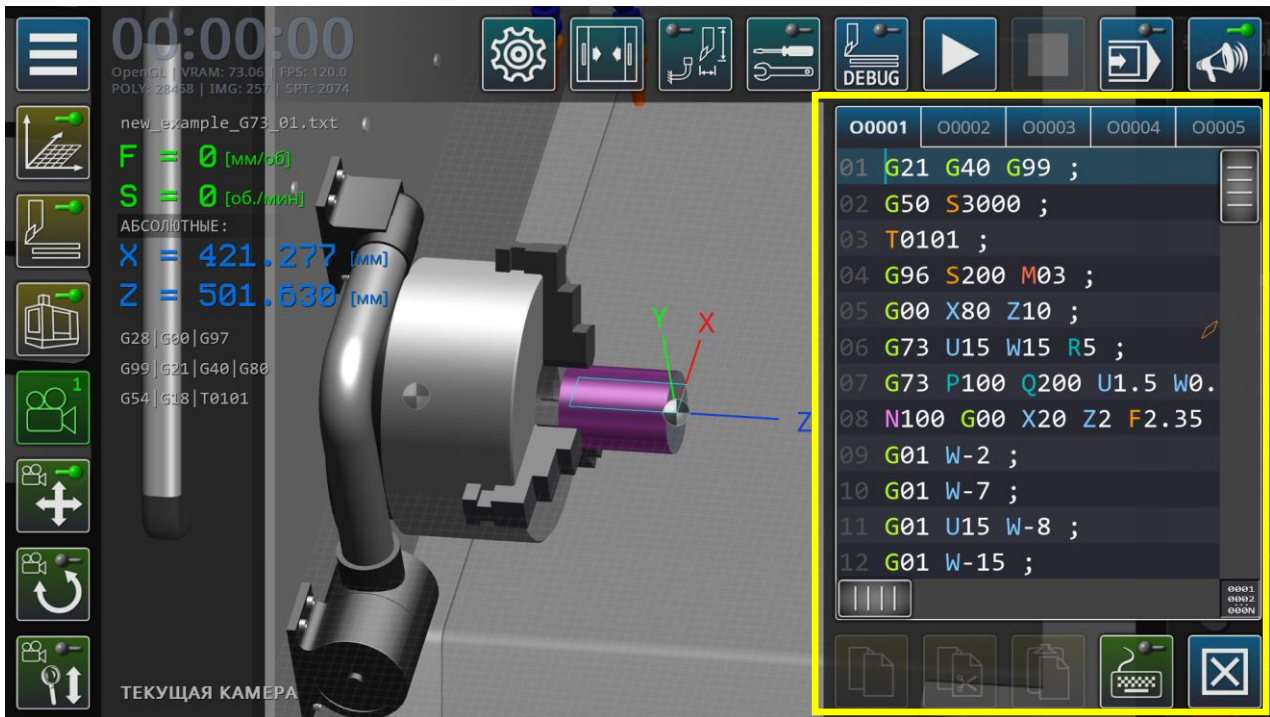


Рисунок 8 – Редактор управляющих программ в стандартном (уменьшенном) режиме отображения

В нижней части текстового редактора расположены основные функциональные кнопки: «Копировать», «Вырезать», «Вставить», «Виртуальная клавиатура» и «Закреть редактор» (рис. 9).

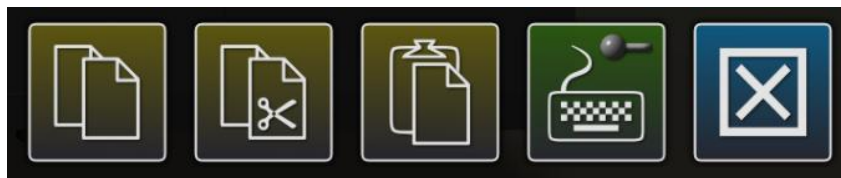


Рисунок 9 – Основные функциональные кнопки текстового редактора

Кнопка-переключатель «Виртуальная клавиатура» позволяет переключить редактор в полноэкранный режим (рис. 10). В полноэкранном режиме текстовый редактор отображается в левой части экрана, а в правой части экрана отображается виртуальная буквенно-цифровая клавиатура, что позволяет редактировать текст на устройствах, оснащенных сенсорным

экраном при отсутствии физической клавиатуры (например, на планшетных компьютерах или смартфонах).

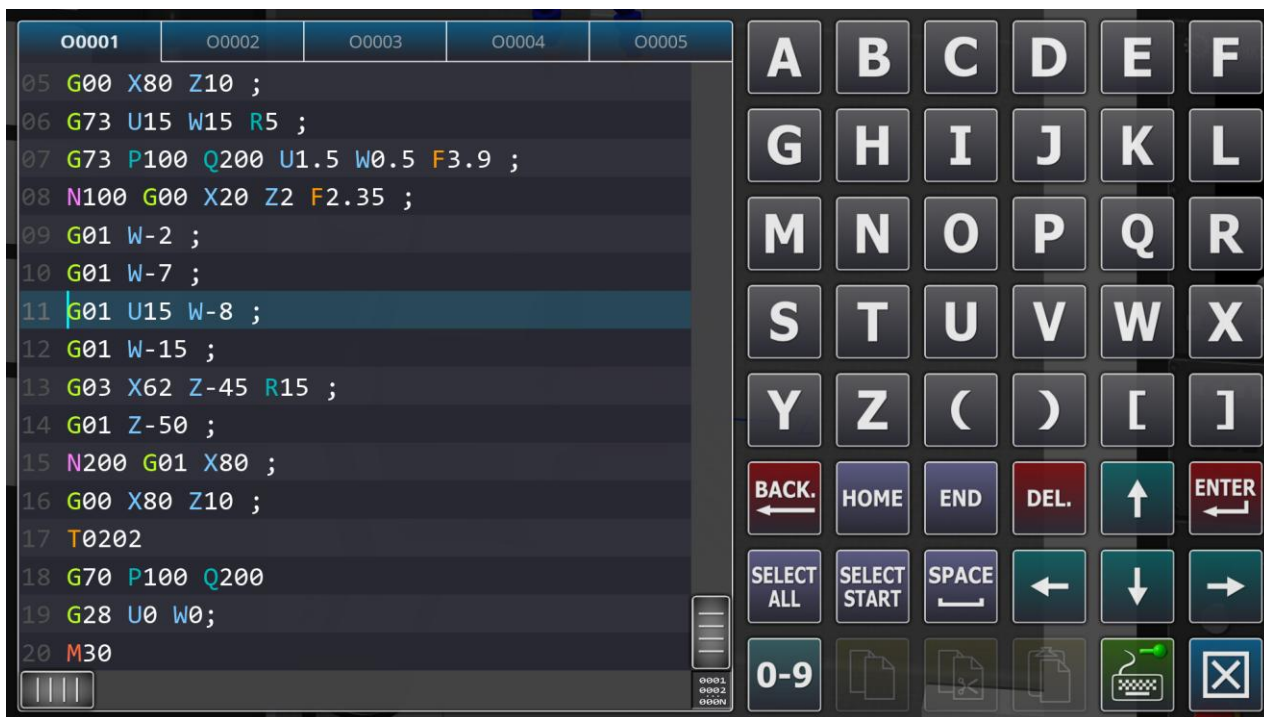


Рисунок 10 – Редактор управляющих программ в полноэкранном режиме отображения

В режиме редактирования текст кода управляющей программы имеет цветовую маркировку по группам буквенных кодов и адресов. Буквенные коды и адреса одной группы (например, технологические параметры T/S/F или координаты X/Z/U/W/I/K) обозначаются одним цветом. В режиме выполнения управляющей программы цветовая маркировка кодов и адресов не применяется.

Кнопка 9 – «Калькулятор режимов обработки» – предназначена для отображения экрана калькулятора режимов токарной обработки (рис. 11).



Рисунок 11 – Экран калькулятора режимов обработки

Экран калькулятора условно можно разделить на две части. В левой части производится расчёт оборотов шпинделя и минутной подачи по заданной скорости резания. В правой части производится расчёт скорости резания и оборотной подачи по заданным оборотам шпинделя. Числовые поля параметров, являющихся исходными данными для расчёта, имеют кнопки увеличения и уменьшения значения. Числовые поля, не имеющие кнопок приращений, предназначены для вывода рассчитанных значений.

Кнопка 10 – «Справочное руководство» – предназначена для отображения экрана встроенной справочной системы приложения (рис. 12). Справочное руководство содержит основные сведения о приложении и используемых инструментах числового программного управления. Справочная информация структурирована по 7 разделам: «О приложении», «Описание модели», «Перечень команд», «Описание G-кодов», «Описание M-кодов», «Постоянные циклы», «Операторы MacroB».



Рисунок 12 – Экран встроенной справочной системы

3. Принцип управления виртуальной камерой с помощью мыши

Основные манипуляции с камерой в режиме №1 могут осуществляться с помощью компьютерной мыши. При этом нажатие и удерживание левой кнопки мыши с сопутствующим перемещением мыши приводит к перемещению точки фокусировки камеры во фронтальной плоскости пространства. Нажатие и удерживание правой кнопки мыши с сопутствующим перемещением мыши приводит к вращению камеры относительно точки фокусировки. Углы вращения (азимут и элевация) камеры ограничены габаритами пространства модели. Изменение дистанции камеры осуществляется вращением колеса прокрутки в прямом и обратном направлениях (рис. 13).

На устройствах с сенсорным экраном управление камерой осуществляется с помощью жестов (Touch Screen).

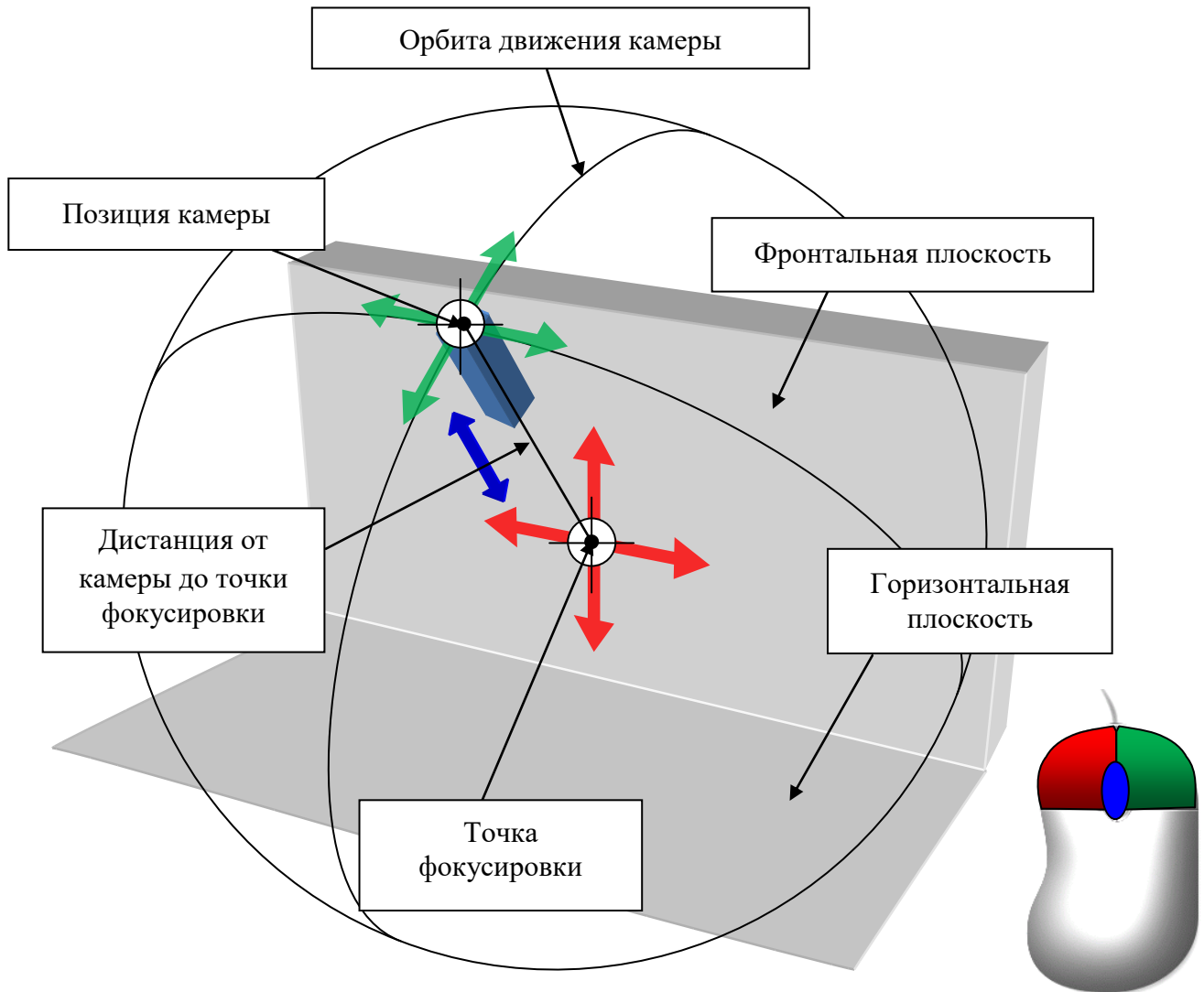


Рисунок 13 – Схема управления виртуальной камерой с помощью компьютерной мыши

4. Режим построения модели траекторий режущего инструмента или режим отладки управляющей программы

В режиме отладки синтаксический анализ (парсинг) управляющей программы осуществляется ускоренно. При этом в настройках приложения можно выбрать соответствующий режим симуляции – «полный» (с

моделированием формы обрабатываемой детали) или «только траектории» (с построением траекторий без фактических перемещений инструмента). В режиме отладки трёхмерная модель станка не отображается (рис. 14).

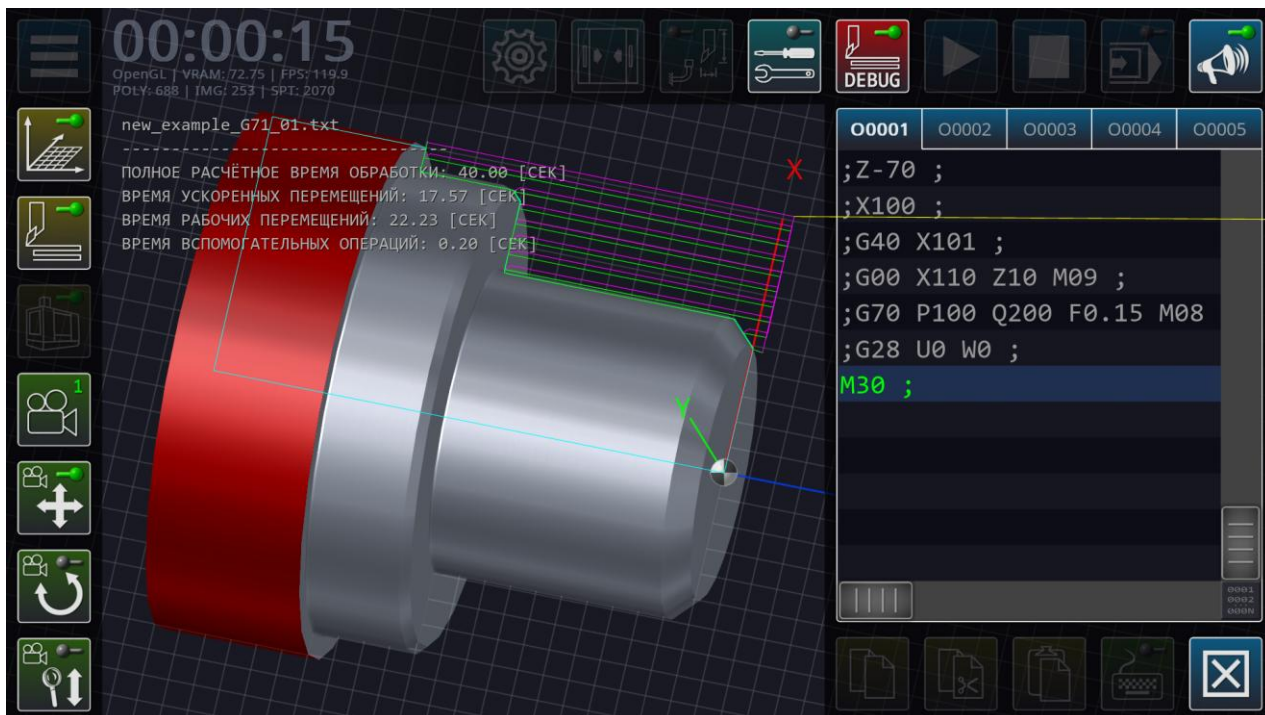


Рисунок 14 – Вид главного экрана программы-симулятора в режиме отладки управляющей программы

В левой части главного экрана отображаются расчётные показатели времени, затраченного на выполнение основных и вспомогательных технологических операций в минутах и секундах.

Если модель обрабатываемой детали построена, геометрические размеры фрагментов контура продольного сечения детали могут быть измерены с помощью функции измерения. Измерение производится по двум координатным осям. Для измерения выбранного участка левой кнопкой мыши необходимо выбрать две контрольные точки, лежащие на контуре обработанной детали (рис. 15). Измерения могут проводиться по двум произвольным точкам, относящимся к разным участкам контура детали.

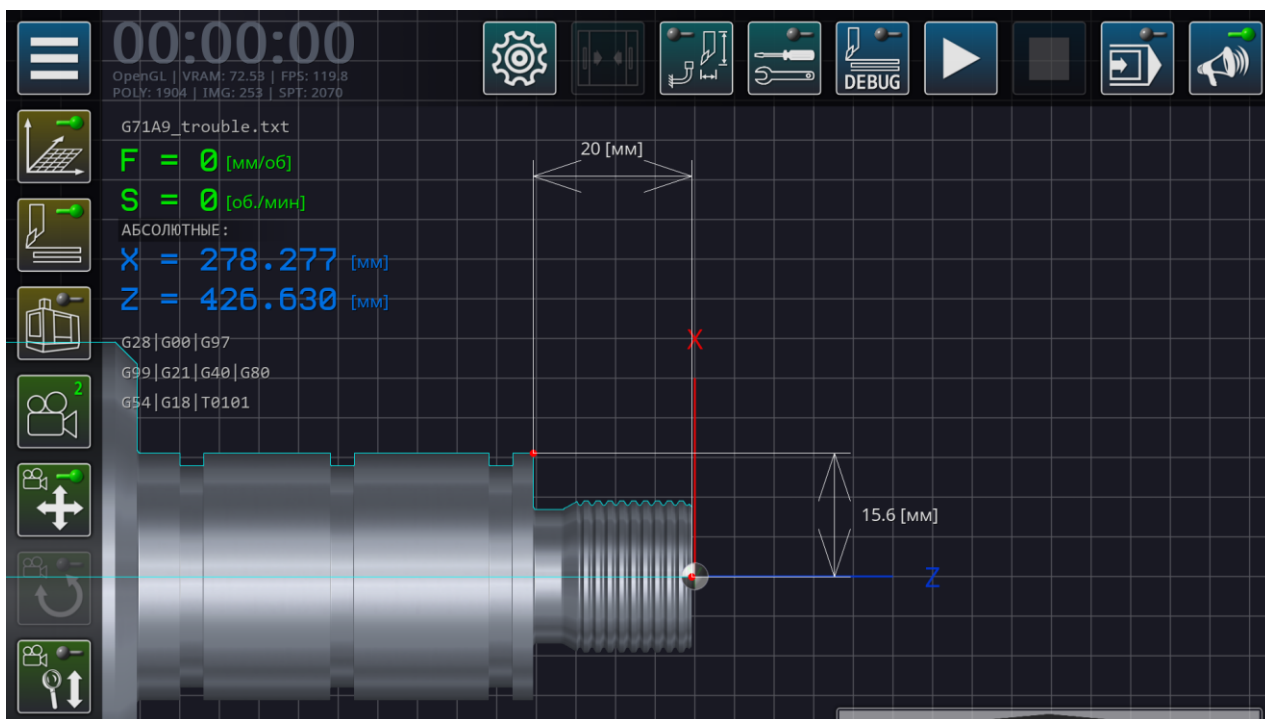


Рисунок 15 – Измерение участков контура обработанной детали

5. Режим привязки инструмента с помощью измерительного датчика

В режиме привязки инструмента модель заготовки скрывается, а датчик привязки инструмента опускается в рабочее положение. На главном экране отображается основная геометрическая информация о датчике – его координаты в референтной системе координат станка, а также размер контактной площадки. В правой части экрана отображается панель управления процессом привязки инструмента. В левой нижней части экрана отображается таблица корректоров инструмента (рис. 16). Таблица корректоров инструмента дублирует таблицу корректоров из раздела настройки инструмента. Используя кнопки панели управления в правой части экрана можно: сбросить значения всех корректоров инструмента, выбрать текущий измеряемый инструмент, установить пространственное положение инструмента с помощью маховиков, имитирующих пульт ручного генератора импульсов (MPG), а также ввести измеренное значение в таблицу.

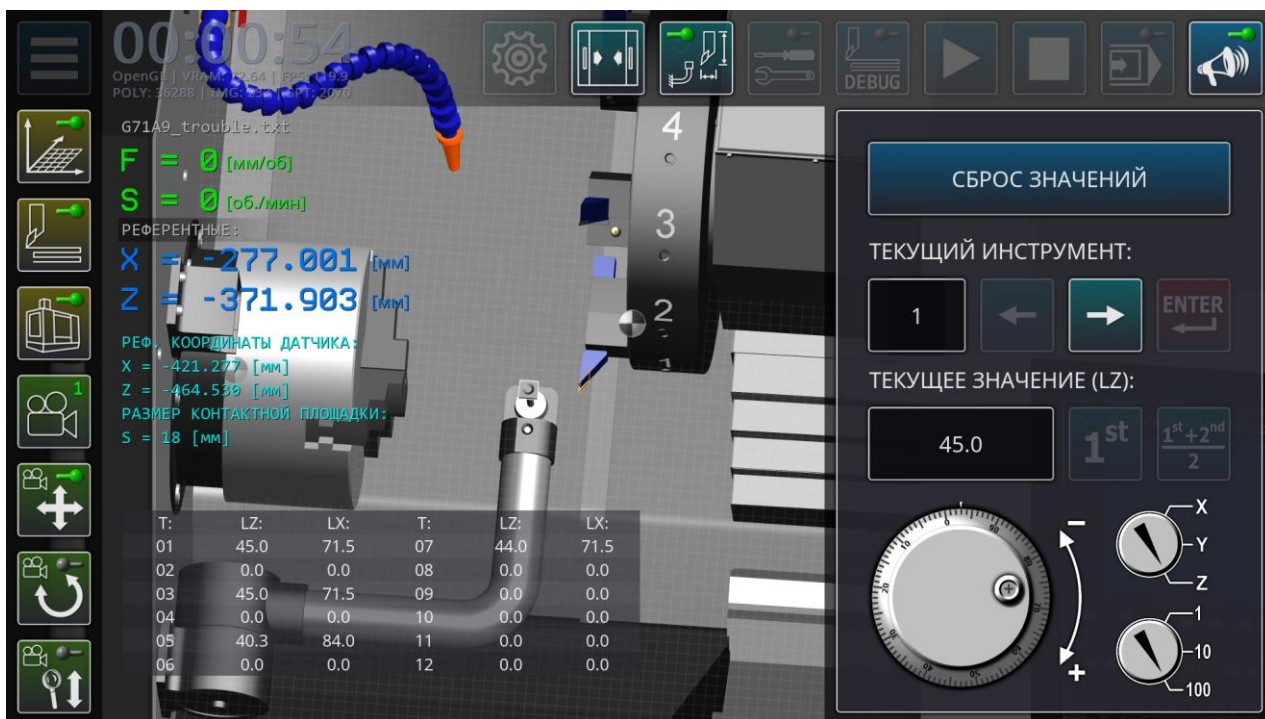


Рисунок 16 – Вид главного экрана программы-симулятора в режиме привязки заготовки

По аналогии с реальным пультом MPG на панели управления в правой части экрана можно выбрать текущую ось станка, скорость ручной подачи, а также осуществить подачу инструмента вращением соответствующего маховика.

Если измеряемый инструмент не касается поверхности датчика, кнопки ввода значения находятся в неактивном состоянии (рис. 17.а). В момент касания инструментом датчика, кнопки ввода измеренного значения становятся активными (рис. 17.б.). Кнопка «1st» предназначена для ввода в таблицу текущего измеренного значения, а кнопка « $(1^{st}+2^{nd})/2$ » - для ввода среднего между текущим и предыдущим значениями, что позволяет привязывать резцы с осевой симметрией, когда касание датчика производится не режущей пластиной, а левой и правой сторонами державки (рис. 18).



Рисунок 17 – Вид панели управления процессом привязки инструмента в режиме позиционирования (а) и в режиме записи измеренного значения (б)



Рисунок 18 – Привязка симметричного резца

6. Краткая справочная информация по программированию операций токарной обработки в симуляторе

Данный раздел дублирует встроенную справочную систему программы-симулятора. Далее описаны основные принципы программирования токарной обработки с применением стандартного (ISO) G-кода. Правила обработки и выполнения команд в симуляционной модели максимально приближены к реальным системам управления на базе GSK, Fanuc, Fagor и др. в спецификации ISO-кода, но могут отличаться от реальных систем.

Симуляционная модель поддерживает 3 системы кодов (А, В, С), включающие различные коды управления и буквенные адреса (табл. 4).

Таблица 4 – Поддерживаемые G-коды управления

Система кодов			Группа	Описание функции
А	В	С		
G04	G04	G04	00	Выдержка времени
G28	G28	G28	00	Возврат в референтную позицию
G50	G92	G92	00	Смещение системы координат
G52	G52	G52	00	Ввод локальной системы координат
G53	G53	G53	00	Установка системы координат станка
G65	G65	G65	00	Вызов макропрограммы
G70	G70	G72	00	Цикл чистовой обработки
G71	G71	G73	00	Цикл продольного чернового точения
G72	G72	G74	00	Цикл поперечного чернового точения
G73	G73	G75	00	Цикл чернового контурного точения
G74	G74	G76	00	Цикл точения торцевых канавок
G75	G75	G77	00	Цикл точения боковых канавок
G76	G76	G78	00	Цикл нарезания резьбы
G00	G00	G00	01	Ускоренное позиционирование
G01	G01	G01	01	Линейная интерполяция
G02	G02	G02	01	Круговая интерполяция CW
G03	G03	G03	01	Круговая интерполяция CCW
G32	G33	G33	01	Нарезание резьбы с постоянным шагом
G34	G34	G34	01	Нарезание резьбы с переменным шагом
G90	G77	G20	01	Цикл продольного точения

Таблица 4 – Продолжение

Система кодов			Группа	Описание функции
А	В	С		
G92	G78	G21	01	Цикл нарезания резьбы
G94	G79	G24	01	Цикл поперечного точения
G96	G96	G96	02	Постоянная скорость резания
G97	G97	G97	02	Отмена постоянной скорости резания
---	G90	G90	03	Абсолютное позиционирование
---	G91	G91	03	Относительное позиционирование
G98	G94	G94	05	Подача в минуту
G99	G95	G95	05	Подача на один оборот
G20	G20	G70	06	Ввод в дюймах
G21	G21	G71	06	Ввод в мм
G40	G40	G40	07	Отмена коррекции на радиус
G41	G41	G41	07	Коррекция на радиус слева
G42	G42	G42	07	Коррекция на радиус справа
G80	G80	G80	10	Отмена постоянного цикла сверления
G83	G83	G83	10	Цикл сверления торцевой поверхности
---	G98	G98	11	Возврат в исходную плоскость
---	G99	G99	11	Возврат в плоскость R
G54	G54	G54	14	Установка системы координат №1
G55	G55	G55	14	Установка системы координат №2
G56	G56	G56	14	Установка системы координат №3
G57	G57	G57	14	Установка системы координат №4
G58	G58	G58	14	Установка системы координат №5
G59	G59	G59	14	Установка системы координат №6

M00 – Приостановка выполнения программы;

M01 – Приостановка выполнения программы;

M02 – Конец программы;

M03 – Включение вращения шпинделя по часовой стрелке;

M04 – Включение вращения шпинделя против часовой стрелки;

M05 – Остановка вращения шпинделя;

M08 – Включение подачи СОЖ;

M09 – Выключение подачи СОЖ;

M30 – Конец программы;

M97 – Вызов внутренней подпрограммы;

M98 – Вызов внешней подпрограммы;

M99 – Конец подпрограммы;

X – Абсолютная координата по оси x (в диаметрах);

Z – Абсолютная координата по оси z;

U – Относительная координата по оси x (в диаметрах) – для системы кодов A;

W – Относительная координата по оси z – для системы кодов A;

F – Линейная скорость перемещения инструмента;

S – Частота вращения шпинделя;

T – Команда смены инструмента (пример, T0202);

R – Радиус дуги, параметр цикла сверления или токарного цикла;

L – Число вызовов подпрограммы или число повторов постоянного цикла;

Q – Параметр цикла сверления или токарного цикла;

P – Идентификатор подпрограммы, время или параметр токарного цикла;

I – Смещение центра дуги по оси x;

K – Смещение центра дуги по оси z, параметр нарезания резьбы.

Далее будут рассмотрены коды системы A.

6.1. Команды линейной интерполяции G00 и G01

Стандартный формат блоков команд G00 и G01:

G00 X_ (U_) Z_ (W_);

G01 X_ (U_) Z_ (W_) (F_);

&

Команда G00 предназначена для ускоренного перемещения инструмента по прямой линии. Код G00 является модальным, т.е. действует до тех пор, пока не будет вызвана другая команда из той же группы, например, G01, G02 или G03. Данная команда активна в исходном состоянии системы ЧПУ. Ускоренное позиционирование применяется для подвода инструмента к заготовке перед началом обработки, а также для отвода инструмента от заготовки после выполнения обработки.

Команда G01 предназначена для линейного перемещения инструмента со скоростью рабочей подачи, заданной с помощью адреса F в одном блоке с G01 или в предыдущих блоках управляющей программы. Код G01 является модальным. Перемещение с линейной интерполяцией используется для резания по прямой линии.

6.2. Команды круговой интерполяции G02 и G03

Стандартный формат блока команд G02 и G03:

G02 (G03) X_ (U_) Z_ (W_) I_ K_ (R_) (F_);

Круговая интерполяция применяется для вытачивания криволинейных поверхностей, форма которых описывается дугой окружности определённого радиуса. Применяются два метода программирования дуги.

Первый метод заключается в задании координат центра дуги с помощью адресов I, K и конечной точки X (U), Z (W), при этом радиус дуги вычисляется автоматически. Второй метод предполагает указание радиуса дуги R и координат конечной точки X (U), Z (W), при этом координаты центра дуги вычисляются автоматически.

Круговая интерполяция по часовой стрелке задается с помощью команды G02, а круговая интерполяция против часовой стрелки – с помощью команды G03, соответственно. При указании центра дуги адресами I, K задаются смещения относительно начальной точки дуги по осям x и z,

соответственно (рис. 19.а). Если дуга задаётся радиусом R, система управления учитывает знак радиуса для вычисления координат центра дуги (рис. 19.б).

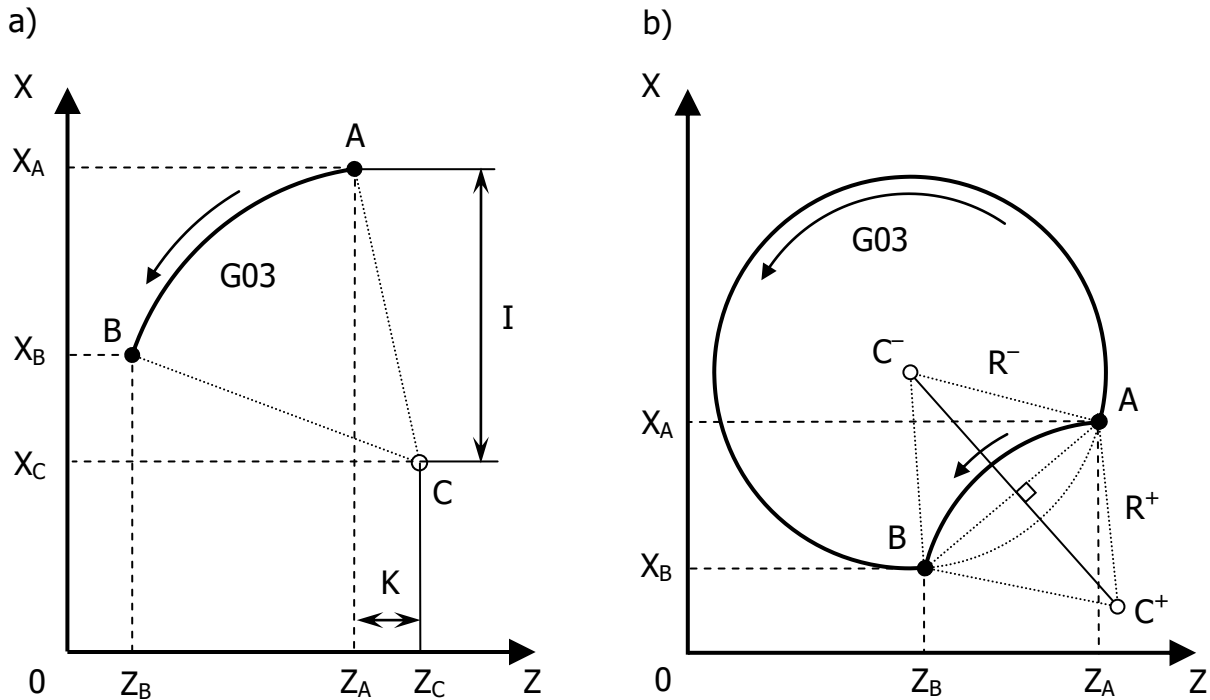


Рисунок 19 – Принцип программирования дуг

При задании дуги система управления проверяет расстояния от центра дуги до начальной и конечной точек. Если расстояния имеют расхождение, превышающее допустимое значение, система управления выдаст сообщение об ошибке.

6.3. Программирование фасок и закруглений

Между блоками линейных перемещений могут быть добавлены фаски и закругления. Для задания фаски используются дополнительные адреса I или K:

G01 X_ (U_) Z_ (W_) I_ (K_);

После адреса I (K) следует числовое значение, задающее фаску. Фаска может быть построена в местах сопряжений «отрезок–отрезок».

Для задания закругления используется дополнительный адрес R:

G01 X_ (U_) Z_ (W_) R_;

После адреса R следует числовое значение, задающее радиус закругления. Закругление может быть построено в местах сопряжений «отрезок–отрезок».

Обязательные условия для программирования фасок и закруглений:

- фаска/закругление программируется только в блоке, задающем перемещение G01 на рабочей подаче;
- следом идущий блок также должен задавать перемещение G01 на рабочей подаче;
- не допускаются пустые блоки или блоки-комментарии, идущие следом за блоком, задающим фаску/закругление;
- следом идущий блок не должен включать в себя G-коды, не относящиеся к линейной интерполяции;
- если конечная точка фаски/закругления выходит за пределы траектории, выдаётся сообщение об ошибке.

6.4. Команда задержки выполнения программы G04

Стандартный формат блока команды G04:

G04 P_ [X_] [U_];

Команда G04 предназначена для выполнения задержки на время, заданное с помощью адреса P. Также система управления допускает использование адресов X и U для указания времени задержки. Если время задано с десятичной точкой, система управления определяет заданное значение как секунды, например, P2.5 – задержка выполнения управляющей программы на 2.5 секунды. Если время задано без десятичной точки, система управления определяет заданное значение как миллисекунды, например, P4000 – задержка выполнения управляющей программы на 4000 миллисекунд или 4 секунды. Команда G04 может программироваться в одном блоке с другими

командами, при этом задержка будет выполнена по завершению выполнения блока.

6.5. Команды выбора системы мер G20, G21

Стандартный формат блока команд G20 и G21:

G20 (G21) _;

Команда G20 устанавливает режим программирования в дюймах. Команда G21 устанавливает режим программирования в миллиметрах. Коды G20 и G21 являются модалными. Режим программирования по умолчанию (G20 или G21) определяется настройками приложения.

6.6. Команда возврата в референтную точку G28

Стандартный формат блока команды G28:

G28 X_ (U_) Z_ (W_);

Команда G28 предназначена для ускоренного перемещения исполнительных органов в референтную точку станка с возможностью задания промежуточной точки.

Количество координат, заданных в одном блоке с кодом G28, определяет количество осей, по которым осуществляется возврат в нулевую точку. Если команда G28 задана без указания координат X (U), Z (W), система управления осуществляет возврат инструмента в референтное положение одновременно по двум осям без перехода в промежуточную точку.

Координаты X (U), Z (W), заданные в одном блоке с кодом G28, определяют положение промежуточной точки (в активной рабочей системе координат), через которую система управления осуществляет возврат исполнительных органов станка в нулевые положения.

Ниже представлен пример использования команды G28.

G28 U100 W0;

В данном блоке программы осуществляется перемещение инструмента на 100 единиц по оси x (в диаметрах) относительно исходной точки и последующее перемещение в референтную точку.

Рекомендуется использовать команду G28 в режиме относительного позиционирования (коды U и W) во избежание столкновения инструмента с деталью.

В случае если команда G28 вызывается в режиме автоматической коррекции на радиус инструмента (G41/G42), вектор коррекции временно обнуляется. После выполнения команды G28 автоматическая коррекция на радиус инструмента возобновляется.

6.7. Команды однопроходного нарезания резьбы G32 и G34

Стандартный формат блоков команд G32 и G34:

G32 X_ (U_) Z_ (W_) F_;

G34 X_ (U_) Z_ (W_) K_ F_;

Команда G32 предназначена для нарезания цилиндрической/конической резьбы с постоянным шагом за один проход. Код G32 является модальным, т.е. действует до тех пор, пока не будет вызвана другая команда из той же группы, например, G00, G01, G02 или G03. Перед вызовом команды G32 должен быть установлен режим постоянной скорости вращения шпинделя (G97), а также режим подачи на один оборот шпинделя (G99). Шаг резьбы задаётся адресом F в одном блоке с G32 или в предыдущих блоках управляющей программы.

Для нарезания резьбы с переменным шагом используется команда G34. Изменение шага программируется адресом K. Числовое значение после адреса K является приращением шага резьбы на один оборот шпинделя. Для увеличивающегося шага: $K > 0$, для уменьшающегося: $K < 0$, соответственно.

6.8. Команды управления режимом коррекции на радиус инструмента G40, G41, G42

Стандартный формат блоков команд G40, G41 и G42:

G41 (G42) _;

...

G40 _;

Система управления имеет встроенный алгоритм расчета эквидистанты при обработке контуров в режиме автоматической коррекции на радиус инструмента. Алгоритм расчета эквидистанты основан на общих принципах вычислений системы управления Fanuc. Режим автоматической коррекции на радиус инструмента включается с помощью модальных кодов G41 и G42. Если по ходу движения необходимо сместить инструмент влево от программируемой траектории на величину его радиуса при вершине, применяется команда G41 (рис. 20.а). Для смещения инструмента вправо от траектории используется команда G42 (рис. 20.б). Система управления автоматически вычисляет путь, по которому должен следовать инструмент, базирующийся на запрограммированном контуре детали и значении радиуса инструмента, находящемся в таблице корректоров инструмента.

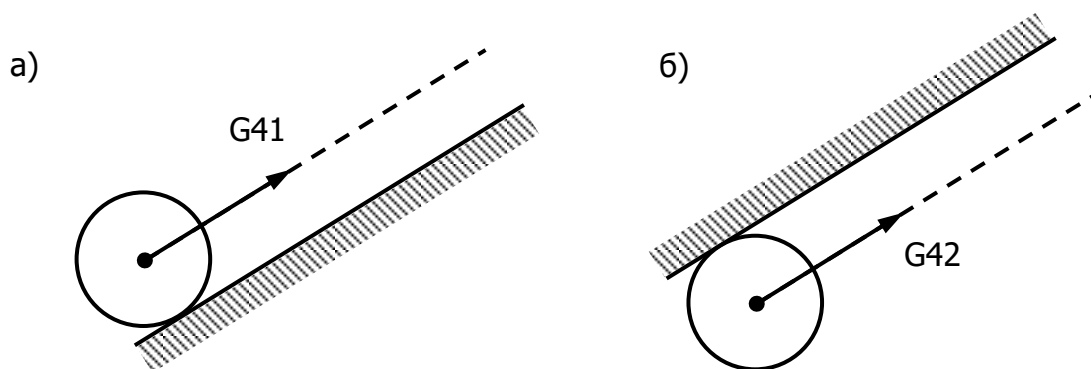


Рисунок 20 – Принцип перемещения инструмента в режиме автоматической коррекции на радиус

Команды G41 и G42 отменяются модальным кодом G40. Также вектор коррекции на радиус инструмента временно обнуляется при выполнении команд G28, G53, T0101. Режим коррекции на радиус не поддерживается при выполнении токарных циклов G71– G76, G83, G90, G92, G94.

В случае завершения программы (M00, M01, M02 или M30) с включенной коррекцией на радиус инструмента, система управления выдаст сообщение об ошибке.

Включение режима коррекции на радиус инструмента (G41 или G42) возможно, только если активны команды G00 или G01 (прямолинейное движение). Если режим коррекции включается на участке круговой интерполяции (G02 или G03), система управления выдаст сообщение об ошибке. Таким образом, если команда G41 (G42) программируется в одном блоке с прямолинейным перемещением, вход на эквидистанту осуществляется плавно по прямой линии, начало которой расположено на запрограммированной траектории, а конец – на эквидистанте следующего перемещения (рис. 21 а и б).

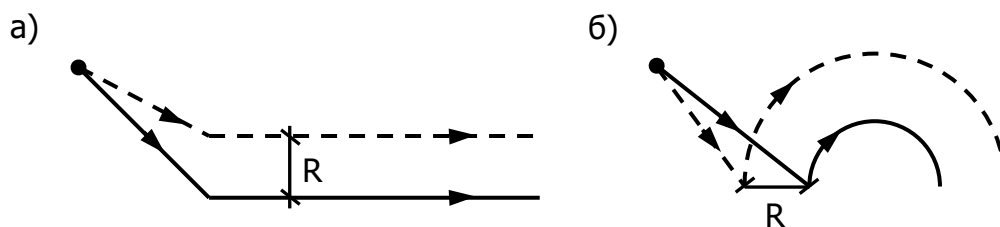


Рисунок 21 – Включение режима автоматической коррекции на радиус инструмента в одном блоке с линейным перемещением

В процессе расчета эквидистанты система управления проверяет следующие блоки управляющей программы, содержащие перемещения, с целью определения точек пересечения текущего участка со следующим участком рассчитываемого пути в плоскости x-z. Обход острых углов с

внешней стороны осуществляется путем построения дополнительного кругового перемещения (рис. 22 а и б).

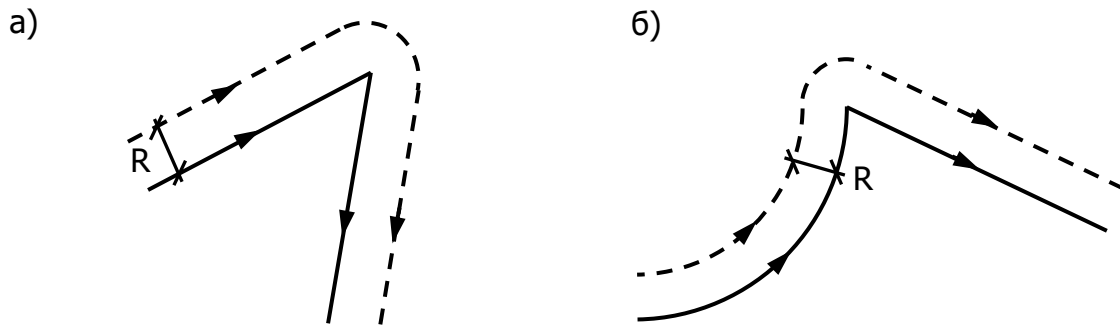


Рисунок 22 – Принцип обхода острого угла в режиме автоматической коррекции на радиус инструмента

При обходе дуги с внутренней стороны в случае, если радиус запрограммированной дуги меньше радиуса инструмента, система управления выдаст сообщение об ошибке. Кроме того, если запрограммированный контур в местах сопряжения дуговых участков с другими участками траектории имеет самопересечения, эквидистанта рассчитывается по принципу уменьшения количества самопересечений окончательного пути.

Команда отмены режима автоматической коррекции на радиус инструмента G40 аналогичным образом должна программироваться в блоке с линейным перемещением (G00 или G01). Если режим коррекции отключается на участке круговой интерполяции (G02 или G03), система управления выдаст сообщение об ошибке. Таким образом, если команда G40 программируется в одном блоке с прямолинейным перемещением, выход с эквидистанты осуществляется плавно по прямой линии, начало которой расположено на эквидистанте предыдущего перемещения, а конец – на траектории текущего перемещения (рис. 23 а и б).

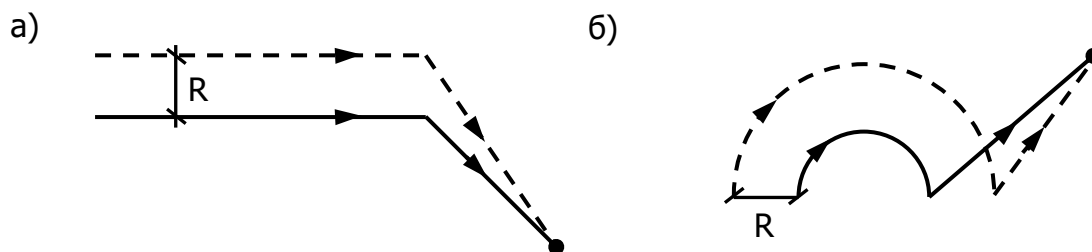


Рисунок 23 – Выключение режима автоматической коррекции на радиус инструмента в одном блоке с линейным перемещением

В режиме автоматической коррекции на радиус инструмента направление вектора коррекции может быть изменено вызовом противоположенной команды. Например, если после включения режима коррекции слева G41 (в последующих блоках программы) задать команду G42, конечная точка эквидистанты данного участка пути будет рассчитана с противоположенной стороны траектории (рис. 24). Изменение направления вектора коррекции возможно только на прямолинейных участках траектории.

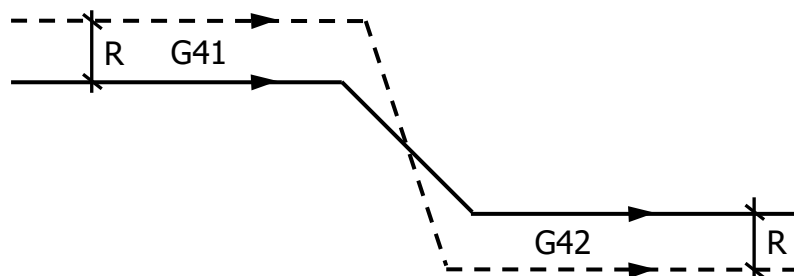


Рисунок 24 – Смена направления вектора коррекции на линейном участке траектории

Ниже представлен пример использования команд G42 и G40.

G50 S3000;

N10 G21 G99 G40;

G00 G28 U0 W0;

T0101;

X60 Z2;
 G96 S300 M4;
 G01 G42 Z0 F0.3;
 Z-15;
 G02 X140 Z-55 R40;
 G01 X160;
 G40 X164;
 G00 X200 Z150;
 G28 U0 W0;
 M30;

На рисунке 25 ниже проиллюстрирован обрабатываемый в данном примере контур.

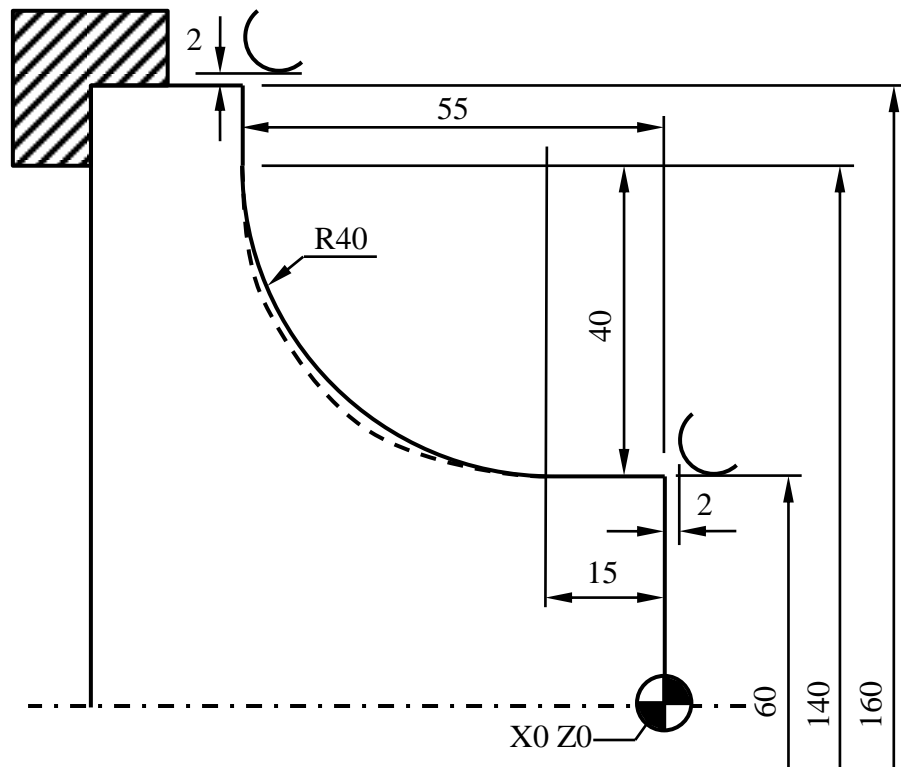


Рисунок 25 – Пример использования радиусной коррекции на дуговом участке контура

Пунктиром изображена траектория движения мнимой вершины резца.

При выполнении перемещений в режиме автоматической коррекции на радиус инструмента на дисплее отображаются фактические координаты инструмента, перемещающегося по эквидистанте. Линия эквидистанты также отображается совместно с линией основной траектории.

6.9. Команда смещения системы координат/ограничения скорости вращения шпинделя G50

Стандартный формат блока команды G50:

G50 X_ Z_; (смещение активной системы координат)

G50 S_; (ограничение скорости вращения шпинделя)

Команда G50 предназначена для программного смещения активной рабочей системы координат в произвольную точку. При этом выполняется смещение начала системы координат таким образом, чтобы текущее положение инструмента соответствовало значениям, заданным в одном блоке с командой G50.

Сама команда G50 не вызывает осевых перемещений. Смещение по команде G50 можно отменить путем программирования еще одного смещения G50, чтобы изменить текущую коррекцию детали обратно на первоначальное значение.

Ниже представлен пример использования команды G50 для преобразования системы координат G54.

(G54)

N1 G00 X150 Z200;

N2 G50 X0 Z0;

N3 G01 X100;

N4 G50 X250 Z200;

В блоке N1 инструмент ускоренно перемещается в точку [x=150; z=200] активной рабочей системы координат (G54). В блоке N2 происходит смещение

рабочей системы координат таким образом, что текущее положение инструмента теперь определяется координатами $[x=0; z=0]$. В блоке N3 инструмент на линейной подаче переходит в точку $[x=100; z=0]$. В блоке N4 рабочая система координат возвращается в первоначальное положение.

После остановки процесса симуляции запрограммированные смещения G50 обнуляются.

Команда G50 отменяет все действующие смещения G52. Недопустимо использование команды G50 в одном блоке с командой G53.

Если после команды G50 не заданы координаты, а задан адрес S, команда выполняет ограничение максимальной скорости вращения шпинделя в режиме постоянной скорости резания (G96).

6.10. Команда установки локальной системы координат G52

Стандартный формат блока команды G52:

G52 X_ Z_;

Команда G52 создаёт локальную систему координат внутри активной рабочей системы координат путём смещения нуля активной рабочей системы координат на заданные с помощью адресов X и Z значения. Заданное смещение отменяется повторным вызовом команды G52 с нулевыми значениями X и Z. Таким образом, локальная система координат совмещается с рабочей системой координат.

При установке локальной системы координат последовательно запрограммированные команды перемещения в режиме абсолютного позиционирования являются значениями координат в локальной системе координат. Для каждой рабочей системы координат (G54-G59) создаётся собственная локальная система координат.

Ниже представлен пример использования команды G52.

(G54)

N1 G00 X50 Z50;

N2 G52 X100;

N3 G01 X0;

N4 G02 X100 R50;

N5 G52 X0;

В блоке N1 осуществляется ускоренное перемещение инструмента в точку $[x=50; z=50]$ активной рабочей системы координат (G54). В блоке N2 устанавливается локальная система координат: начало рабочей системы координат смещается на 100 единиц в положительном направлении оси x, при этом положение инструмента в рабочей системе координат определяется точкой $[x=-50; z=50]$. В блоке N3 инструмент линейно перемещается в точку $[x=0; z=50]$ локальной системы координат. В блоке N4 осуществляется круговое движение по часовой стрелке в точку $[x=100; z=50]$ локальной системы координат. В блоке N5 локальная система координат совмещается с рабочей системой координат (G54) по оси x.

После вызова команды G52 при выполнении перемещений на дисплее отображаются координаты инструмента в локальной системе координат.

Команда G52 и сопровождающие её параметры всегда программируются в отдельном блоке.

6.11. Команда перехода в систему координат станка G53

Стандартный формат блока команды G53:

G53 _;

При выполнении команды G53 система управления временно отменяет смещения рабочих координат и использует систему координат станка. Команда G53 является немодальной, т.е. действует только в том блоке, в

котором задана. Совместно с кодом G53 можно программировать линейные либо круговые перемещения.

Команда G53 работает только в режиме абсолютного позиционирования. Если команда G53 задана в режиме относительного позиционирования, система управления выдаст сообщение об ошибке.

В случае если команда G53 вызывается в режиме автоматической коррекции на радиус инструмента (G41/G42), вектор коррекции временно обнуляется. После выполнения команды G53 автоматическая коррекция на радиус инструмента возобновляется.

Недопустимо использование команды G53 в одном блоке с командами G28 или G50.

6.12. Команды переключения рабочей системы координат G54-G59

Стандартный формат блока команд G54-G59:

G54 (G55, G56 ...) _;

Данная симуляция предоставляет возможность использования шести предварительно определённых рабочих систем координат. Выбор и установка (в качестве активной) рабочей системы координат осуществляется с помощью модальных кодов: G54, G55, G56, G57, G58, G59.

Смещения нулей каждой рабочей системы координат относительно машинного нуля координатного стола настраиваются предварительно в соответствующем разделе экрана параметров заготовки. В процессе выполнения управляющей программы эти смещения могут изменяться с помощью команд G50 и G52.

При выполнении перемещений на дисплее отображаются координаты инструмента в активной рабочей системе координат.

Команды G54-G59 можно задавать в отдельном блоке программы либо в блоках совместно с другими командами.

6.13. Команда вызова макропрограммы G65

Стандартный формат блока команды G65:

G65 P_ (L_) (A_ B_ C_ D_ E_ F_ H_ I_ J_ K_ M_ Q_ R_ S_ T_ U_ V_ W_ X_ Y_ Z_);

Команда G65 служит для вызова макропрограммы, код которой расположен в основной управляющей программе или в отдельной программе (с собственной нумерацией блоков). Если макропрограмма расположена в коде основной программы, она должна иметь заголовок вида OXXXX, заданный отдельным блоком. Также код макропрограммы может быть задан в дополнительной вкладке текстового редактора. В одном блоке с командой G65 обязательно задаётся параметр P. Целочисленный параметр P определяет четырёхзначный номер макропрограммы.

Необязательный целочисленный параметр L определяет количество последовательных вызовов макропрограммы, например, L3 означает, что макропрограмма будет вызвана 3 раза подряд. Допускается до 999 последовательных вызовов макропрограмм. Если параметр L не задан в блоке с командой G65, макропрограмма вызывается один раз.

Любая вызываемая макропрограмма должна заканчиваться кодом завершения M99. После завершения макропрограммы управление передаётся блоку, следующему за блоком с командой вызова макропрограммы (G65).

Макропрограммы используют пространство локальных переменных #1-#33, значения которым можно присваивать путём ввода буквенных аргументов после команды G65. Любые изменения значений локальных переменных в коде макропрограммы действуют только в самой макропрограмме и не возвращаются в основную программу.

Основной метод передачи аргументов в макропрограмму предполагает использование различных буквенных адресов A-Z (кроме G, L, N, O, P). При

этом каждый буквенный адрес соответствует определённому номеру локальной переменной:

A – #1	I – #4	T – #20
B – #2	J – #5	U – #21
C – #3	K – #6	V – #22
D – #7	M – #13	W – #23
E – #8	Q – #17	X – #24
F – #9	R – #18	Y – #25
H – #11	S – #19	Z – #26

Пример: G65 P1001 L12 A35.5 B–14.2 Q8;

В данном примере макропрограмма O1001 вызывается 12 раз подряд, и при вызове макропрограммы в неё передаются значения локальных переменных: #1=35.5; #2=–14.2; #17=8.

Альтернативный метод передачи аргументов в макропрограмму предполагает использование буквенных адресов A, B, C, I, J, K. При этом каждый из адресов I, J, K может задаваться до 10 раз в одном блоке с кодом G65. Соответствие буквенных адресов номерам локальных переменных следующее:

A – #1	I4 – #13	I8 – #25
B – #2	J4 – #14	J8 – #26
C – #3	K4 – #15	K8 – #27
I1 – #4	I5 – #16	I9 – #28
J1 – #5	J5 – #17	J9 – #29
K1 – #6	K5 – #18	K9 – #30
I2 – #7	I6 – #19	I10 – #31
J2 – #8	J6 – #20	J10 – #32
K2 – #9	K6 – #21	K10 – #33
I3 – #10	I7 – #22	
J3 – #11	J7 – #23	
K3 – #12	K7 – #24	

Пример: G65 P1002 L3 I68.4 J-13 K4 I-18.5 J-9 K50.2 I19.2 J-1;

В данном примере макропрограмма O1002 вызывается 3 раза подряд, и при вызове макропрограммы в неё передаются значения локальных переменных: #4=68.4; #5=-13; #6=4; #7=-18.5; #8=-9; #9=50.2; #10=19.2; #11=-1; #12=88.8.

Макропрограммы обрабатываются по аналогии с внешними подпрограммами. Допускается использование до 20 вложенных макропрограмм со своей структурой кода – условными и безусловными переходами, циклами и т.д.

6.14. Команды переключения режима вращения шпинделя G96, G97

Стандартный формат блока команд G96/G97:

G96 (G97) S_;

Модальная команда G96 включает режим постоянной скорости резания. При этом после адреса S задаётся величина скорости резания в размерности [м/мин] или [фут/мин] в зависимости от выбранной системы измерений (G21 или G20). Фактические обороты шпинделя в режиме G96 рассчитываются автоматически в зависимости от текущего положения инструмента по оси X. Перед использованием команды G96 необходимо ввести ограничение на максимальную скорость вращения шпинделя командой G50.

Команда G97 включает режим постоянной скорости вращения шпинделя, независящий от текущего положения инструмента по оси X. Скорость вращения шпинделя [об/мин] программируется напрямую адресом S.

6.15. Команды переключения режима подачи G98, G99

Стандартный формат блока команд G98/G99:

G98 (G99) F_;

Модальная команда G98 включает режим минутной подачи. При этом значение рабочей подачи, заданное адресом F, определяется в размерности [мм/мин] или [дюйм/мин] в зависимости от выбранной системы измерений

(G21 или G20). В режиме G98 скорость рабочих перемещений определяется только заданным значением F независимо от скорости вращения шпинделя.

Модальная команда G99 включает режим обратной подачи. При этом значение рабочей подачи, заданное адресом F, определяется в размерности [мм/об] или [дюйм/об] в зависимости от выбранной системы измерений (G21 или G20). В режиме G99 скорость рабочих перемещений определяется заданным значением F, умноженным на число оборотов шпинделя S. Если шпиндель остановлен, поступательное движение суппорта не осуществляется.

6.16. Команды приостановки выполнения программы M00, M01

Стандартный формат блока команд M00/M01:

_ M01 (M00);

В данной симуляции коды M00 и M01 выполняются одинаково, и служат для приостановки процесса выполнения управляющей программы до её ручного возобновления. Код M00/M01 может программироваться как в отдельном блоке, так и в одном блоке с другими командами, при этом приостановка управляющей программы будет выполнена по завершению выполнения блока.

В данной симуляции код M00/M01 может использоваться для контроля корректности траекторий и режимов обработки в целях отладки управляющей программы.

6.17. Команды завершения выполнения программы M02, M30

Стандартный формат блока команд M02/M30:

_ M02 (M30);

В данной симуляции коды M02 и M30 выполняются одинаково, и служат для завершения процесса выполнения управляющей программы. Код M02/M30 может программироваться как в отдельном блоке, так и в одном блоке с

другими командами, при этом управляющая программа будет завершена после выполнения блока.

При обнаружении кода M02/M30 парсинг управляющей программы завершается. При этом автоматически отключается вращение шпинделя и подача СОЖ. Внутренние подпрограммы, вызываемые кодом M97 необходимо программировать после кода M02/M30.

6.18. Команды управления вращением шпинделя M03, M04, M05

Стандартный формат блока команд M03, M04 и M05:

M03 (M04) S_;

Команды M03 и M04 предназначены для включения вращения шпинделя по часовой стрелке и против часовой стрелки, соответственно. При включении вращения шпинделя должна быть задана скорость вращения [об/мин] адресом S в одном блоке с командой M03/M04 или в предыдущих блоках.

Вращение шпинделя отключается командой M05.

6.19. Команда переключения инструмента T

Стандартный формат блока команды T:

T_;

Команда переключения инструмента состоит из двух чисел (например, «T0101»), первое из которых указывает на положение инструмента в револьверной голове, а второе – на номер инструмента в таблице корректоров инструмента. При этом система управления применит значения осевых вылетов, соответствующие заданному номеру корректора.

6.20. Команды управления подачей СОЖ M08, M09

Стандартный формат блока команд M08 и M09:

_ M08;

...

_ M09;

В данной симуляции код M08 служит для включения подачи смазочно-охлаждающей жидкости (СОЖ) в зону резания. Подача СОЖ отключается кодом M09. Коды M08 и M09 могут программироваться как в отдельном блоке, так и в одном блоке с другими командами, при этом операция подачи/отключения СОЖ будет выполнена по завершению выполнения блока.

6.21. Команды управления подпрограммами M97, M98, M99

Стандартный формат блока команд M97, M98 и M99:

M97 P_ (L_);

...

M98 P_ (L_);

...

M99;

Команда M97 служит для вызова внутренней подпрограммы, код которой расположен в основной управляющей программе после команды завершения M02/M30. В одном блоке с командой M97 обязательно задаётся параметр P. Целочисленный параметр P определяет номер блока начала подпрограммы, например, P100 означает передачу управления блоку N100 начала подпрограммы, при этом блок начала подпрограммы обязательно должен содержать адрес N.

Необязательный целочисленный параметр L определяет количество последовательных вызовов подпрограммы, например, L3 означает, что внутренняя подпрограмма будет вызвана 3 раза подряд. Допускается до 999 последовательных вызовов подпрограмм. Если параметр L не задан в блоке с командой M97, подпрограмма вызывается один раз.

Команда M98 служит для вызова внешней подпрограммы. По аналогии с макропрограммами (G65) внешняя подпрограмма может быть расположена в основной управляющей программе или в отдельной программе (с собственной

нумерацией блоков). Если подпрограмма расположена в коде основной программы, она должна иметь заголовок вида OXXXX, заданный отдельным блоком. Также код подпрограммы может быть задан в дополнительной вкладке текстового редактора. В одном блоке с командой M98 обязательно задаётся параметр P. Комплексный параметр P определяет номер подпрограммы, а также может содержать число вызовов подпрограммы. Так, например, параметр P31002 означает, что внешняя подпрограмма O1002 будет вызвана 3 раза подряд. В данной симуляции, если внешняя подпрограмма расположена в дополнительной вкладке текстового редактора, её номер должен соответствовать номеру вкладки текстового редактора. Если параметр P не содержит число вызовов подпрограммы, допускается использование параметра L аналогично формату команды M97.

Любая вызываемая подпрограмма должна заканчиваться кодом завершения M99. После завершения подпрограммы управление передаётся блоку, следующему за блоком с командой вызова подпрограммы (M97 или M98).

В данной симуляции допускается многоуровневый вызов подпрограмм – многократный вызов подпрограммы из подпрограммы. При этом следует соблюдать порядок вызова и завершения вложенных подпрограмм.

6.22. Цикл черновой продольной контурной обработки G71

Стандартный формат блока команды G71:

G71 U_ R_;

G71 P_ Q_ U_ W_ F_ S_;

где: [первая строка] U – глубина обработки для черновых проходов (режим программирования в радиусах) [мм]; R – расстояние отхода после окончания каждого прохода [мм]; [вторая строка] P – порядковый номер первого кадра описания контура; Q – порядковый номер последнего кадра

описания контура; U – величина (режим программирования в диаметрах) и направление снятия чистового припуска по оси X [мм]; W – величина и направление снятия чистового припуска по оси Z [мм]; F – скорость подачи при черновой обработке; S – частота вращения шпинделя/скорость резания при чистовой обработке.

Ниже представлен пример использования цикла G71 (рис. 26).

G00 X32 Z3;

G71 U2 R0.2;

G71 P100 Q150 U0.3 W0.3 F0.15;

N100 G01 X10 Z0;

N110 G01 Z-8;

N120 G01 X20 Z-18;

N130 G01 X30;

N140 G01 Z-25;

N150 G01 X32;

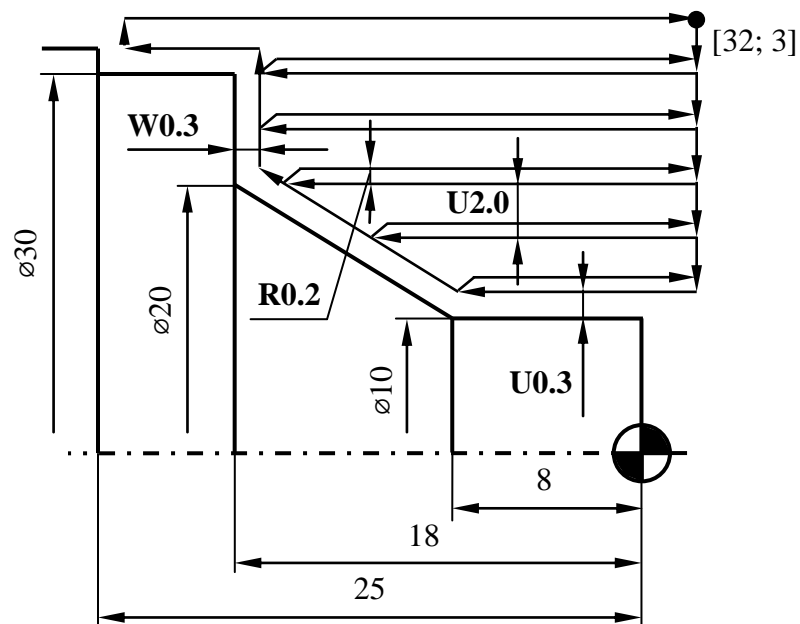


Рисунок 26 – Схема траекторий при выполнении цикла G71

6.23. Цикл черновой поперечной контурной обработки G72

Стандартный формат блока команды G72:

G72 W_ R_;

G72 P_ Q_ U_ W_ F_ S_;

где: [первая строка] W – глубина обработки для черновых проходов [мм]; R – расстояние отхода после окончания каждого прохода [мм]; [вторая строка] P – порядковый номер первого кадра описания контура; Q – порядковый номер последнего кадра описания контура; U – величина (режим программирования в диаметрах) и направление снятия чистового припуска по оси X [мм]; W – величина и направление снятия чистового припуска по оси Z [мм]; F – скорость подачи при черновой обработке; S – частота вращения шпинделя/скорость резания при чистовой обработке.

Ниже представлен пример использования цикла G72 (рис. 27).

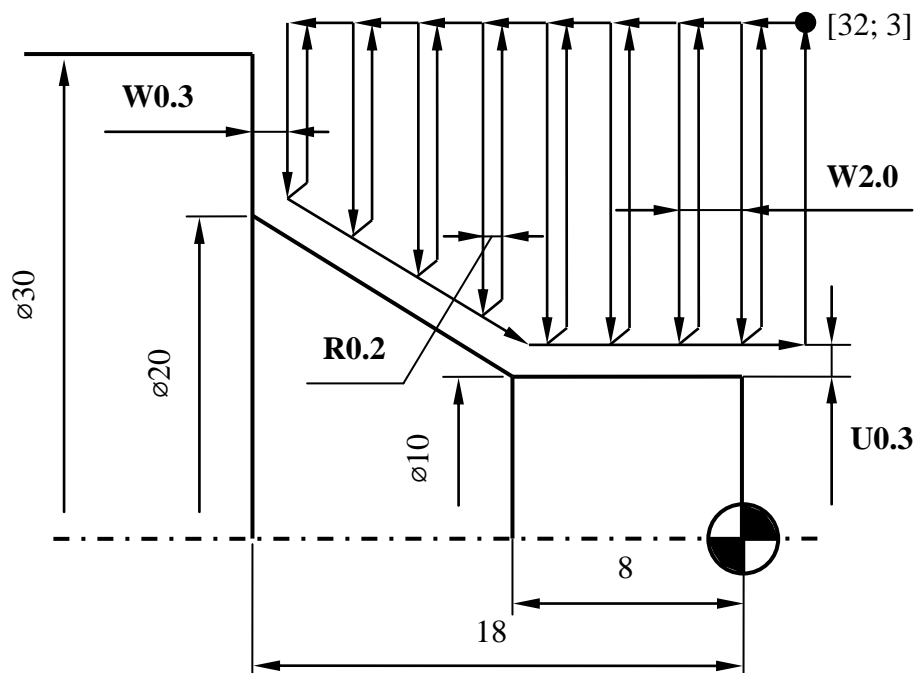


Рисунок 27 – Схема траекторий при выполнении цикла G72

```
G00 X32 Z3;  
G72 W2 R0.2;  
G72 P100 Q130 U0.3 W0.3 F0.15;  
N100 G01 X10 Z0;  
N110 G01 Z-8;  
N120 G01 X20 Z-18;  
N130 G01 X30;
```

6.24. Цикл контурной обработки G73

Стандартный формат блока команды G73:

```
G73 U_ W_ R_;  
G73 P_ Q_ U_ W_ F_ S_;
```

где: [первая строка] U – величина (режим программирования в радиусах) и направление снятия суммарного припуска по оси X [мм]; W – величина и направление снятия суммарного припуска по оси Z [мм]; R – количество последовательных проходов при снятии черного припуска, включая полустроковой проход; [вторая строка] P – порядковый номер первого кадра описания контура; Q – порядковый номер последнего кадра описания контура; U – величина (режим программирования в диаметрах) и направление снятия чистового припуска по оси X [мм]; W – величина и направление снятия чистового припуска по оси Z [мм]; F – скорость подачи при черновой обработке; S – частота вращения шпинделя/скорость резания при чистовой обработке.

Ниже представлен пример использования цикла G73 (рис. 28).

```
G00 X32 Z3;  
G73 U2 W2 R3;  
G73 P100 Q150 U0.3 W0.3 F0.15;  
N100 G01 X10 Z0;
```



```

G00 X32 Z3;
G70 P100 Q140 F0.1 S2000;
N100 G01 X10 Z0;
N110 G01 Z-8;
N120 G01 X20 Z-18;
N130 G01 X30;
N140 G01 Z-25;

```

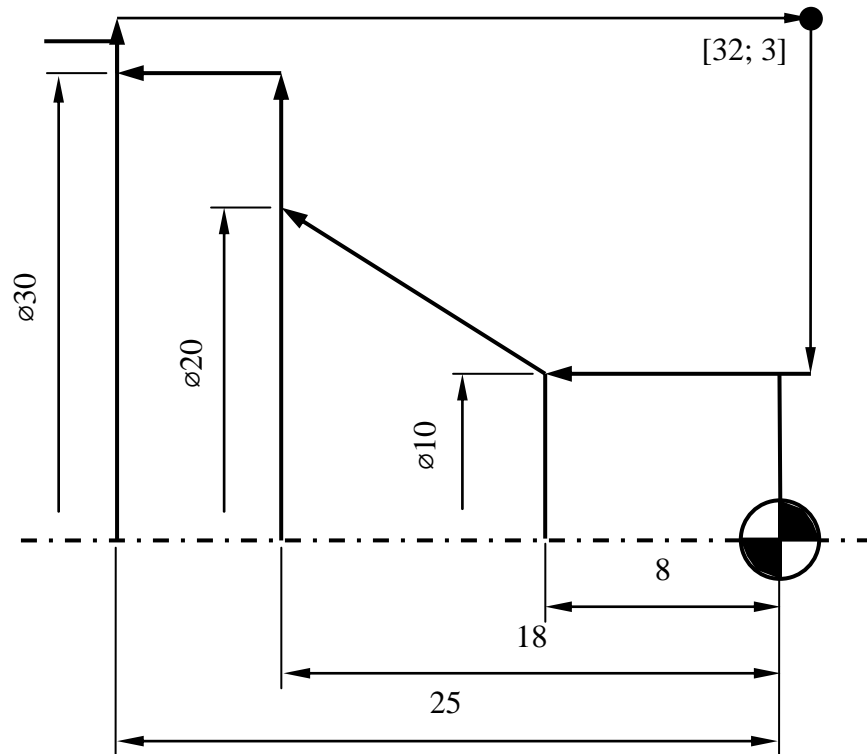


Рисунок 29 – Схема траекторий при выполнении цикла G73

6.26. Цикл автоматической обработки боковых канавок G75

Стандартный формат блока команды G75:

G75 R_;

G75 X_(U_) Z_(W_) P_ Q_ F_;

где: [первая строка] R – расстояние, на которое отводится инструмент после завершения шага проточки [мм]; [вторая строка] X(U) – координата

где: [первая строка] R – расстояние, на которое отводится инструмент после завершения шага проточки [мм]; [вторая строка] X(U) – координата конечной точки по оси X [мм]; Z(W) – координата конечной точки по оси Z [мм]; P – шаг проточки по оси X [мкм]; Q – шаг проточки по оси Z [мкм]; F – скорость подачи.

Цикл G74 аналогичен циклу G75 за исключением того, что обработка канавки осуществляется вдоль оси Z. Данный цикл может использоваться для прерывистого сверления торцевых отверстий.

6.28. Цикл автоматического нарезания резьбы G76

Стандартный формат блока команды G76:

G76 Pxxuyzz Q_ R_;

G76 X_(U_) Z_(W_) R_ P_ Q_ F_;

где: [первая строка] xx – двузначное число чистовых проходов; yy – двузначное число, определяющее размер фаски; zz – двузначное число, определяющее угол режущей кромки инструмента; Q – минимальная глубина нарезания резьбы (режим программирования в радиусах) [мкм]; R – суммарная глубина нарезания при чистовых проходах [мм]; [вторая строка] X(U) – координата конечной точки нарезания резьбы по оси X [мм]; Z(W) – координата конечной точки нарезания резьбы по оси Z [мм]; R – величина перемещения по оси X при нарезании конической резьбы (не программируется при нарезании цилиндрической резьбы) [мм]; P – высота резьбы [мкм]; Q – глубина нарезания резьбы для первого прохода [мкм]; F – шаг резьбы по оси Z.

Ниже представлен пример использования цикла G76 (рис. 31).

M03 S150;

G00 X90 Z10;

G76 P020060 Q1000 R0.5;

G76 X70 Z-30 P5000 Q2000 F5.5;

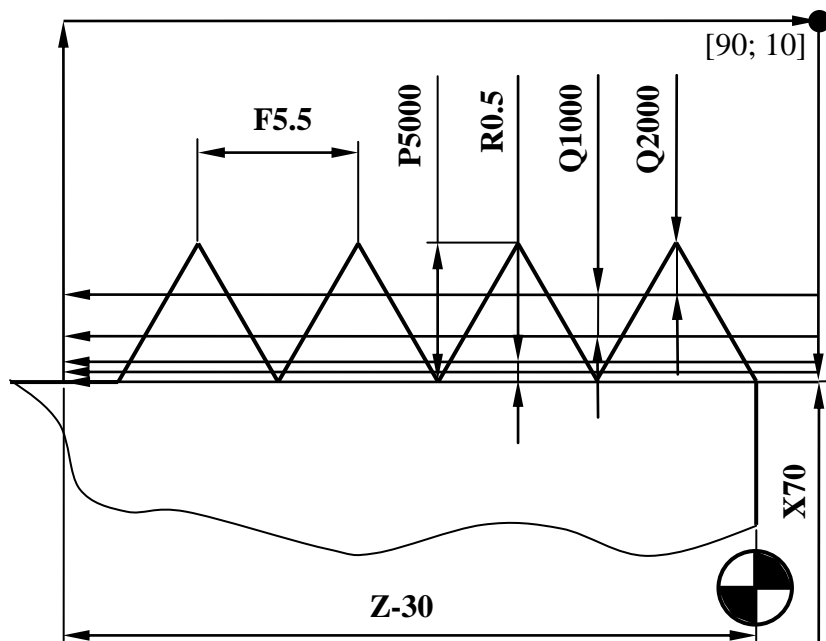


Рисунок 31 – Схема траекторий при выполнении цикла G76

6.29. Цикл продольной черновой обработки G90

Стандартный формат блока команды G90:

G90 X_ (U)_ Z_ (W)_ R_ F_;

где: X(U) – координата конечной точки первого прохода по оси X [мм];

Z(W) – координата конечной точки первого прохода по оси Z [мм]; R –

изменение радиуса основания конуса [мм]; F – скорость подачи.

Ниже представлен пример использования цикла G90 (рис. 32).

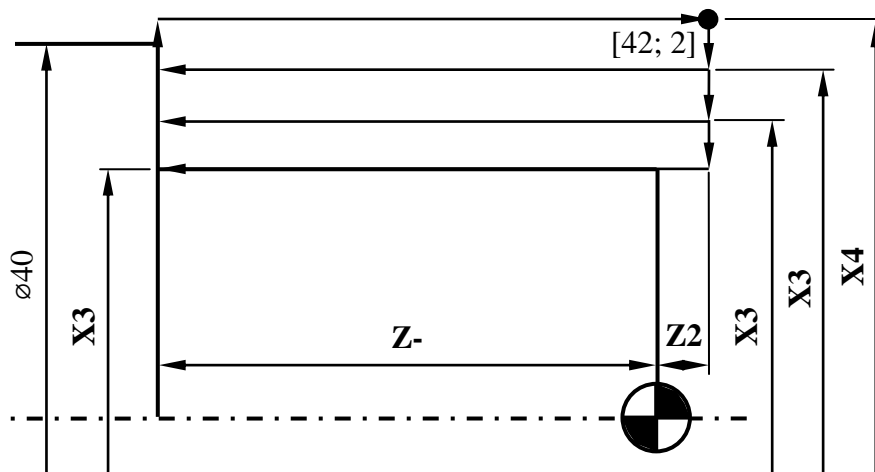


Рисунок 32 – Схема траекторий при выполнении цикла G90

G00 X42 Z2;
 G90 X37 Z-25 F0.1;
 X34;
 X31;

6.30. Цикл торцевой черновой обработки G94

Стандартный формат блока команды G94:

G94 X_(U_) Z_(W_) R_ F_;

где: X(U) – координата конечной точки первого прохода по оси X [мм];
 Z(W) – координата конечной точки первого прохода по оси Z [мм]; R –
 изменение радиуса основания конуса [мм]; F – скорость подачи.

Ниже представлен пример использования цикла G94 (рис. 33).

G00 X32 Z1;
 G94 X10 Z-2 F0.1;
 Z-4;
 Z-6;

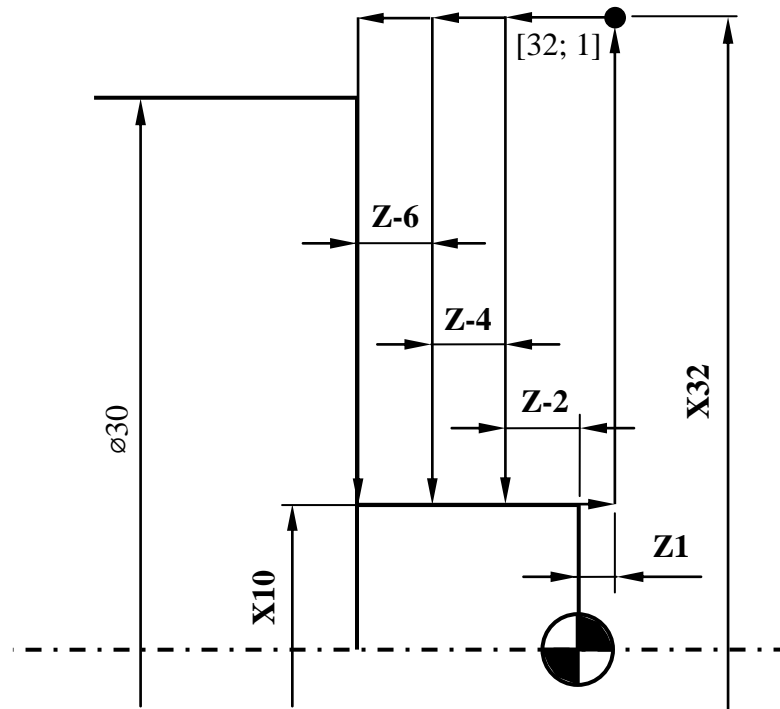


Рисунок 33 – Схема траекторий при выполнении цикла G94

6.31. Цикл нарезания резьбы G92

Стандартный формат блока команды G92:

G92 X_(U_) Z_(W_) R_ F_;

где: X(U) – координата конечной точки первого прохода нарезания резьбы по оси X [мм]; Z(W) – координата конечной точки первого прохода нарезания резьбы по оси Z [мм]; R – величина перемещения по оси X при нарезании конической резьбы (не программируется при нарезании цилиндрической резьбы) [мм]; F – шаг резьбы по оси Z.

Ниже представлен пример использования цикла G92 (рис. 34).

G00 X42 Z2;

G92 X30 Z-25 F1.0;

X29;

X28;

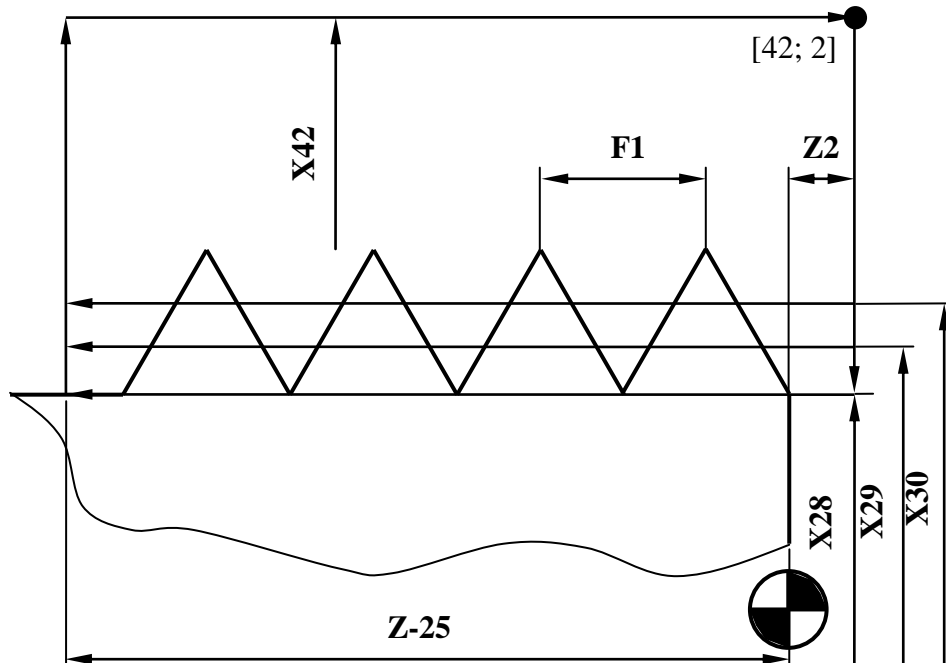


Рисунок 34 – Схема траекторий при выполнении цикла G92

6.32. Стандартный цикл многопроходного (прерывистого) сверления

G83

Стандартный формат блока команды G83:

G83 X_(U_) Z_(W_) R_ P_ Q_ F_;

где: X(U) – координата конечной точки по оси X [мм]; Z(W) – координата конечной точки по оси Z [мм]; R – абсолютная координата плоскости отвода инструмента по оси Z [мм]; P – время выдержки на дне отверстия в миллисекундах; Q – шаг сверления по оси Z [мкм]; F – скорость подачи.

Цикл G83 выполняет сверление отверстия с периодическим полным выводом инструмента из отверстия. Дополнительный параметр Q определяет глубину одного рабочего прохода. После каждого рабочего прохода инструмент отводится либо в исходную позицию, либо в плоскость отвода, после чего сверление продолжается до достижения конечной глубины отверстия.

6.33. Команда отмены стандартного цикла G80

Стандартный формат блока команды G80:

G80 _;

Модальная команда G80 отменяет активный постоянный цикл G83. Код G80 может программироваться как в отдельном блоке, так и в одном блоке с другими командами. По умолчанию при запуске управляющей программы активна команда G80.

6.34. Параметрическое программирование

Симуляционная модель позволяет интерпретировать основные возможности языка параметрического программирования Macro B: работу с переменными, безусловный и условный переходы, а также циклы с предусловием.

6.34.1. Работа с переменными

Представленная система управления работает с общими и локальными переменными вида #_. Имя переменной (номер) задаётся целым числом. Перед номером переменной указывается символ решётки «#». Переменной можно присвоить значение с помощью оператора присваивания « \leftarrow ».

Системой управления интерпретируются арифметические и логические выражения, а также математические функции с использованием вложенных скобок «[...]».

Поддерживаемые арифметические операторы:

- 1) равенство: $\#1=\#2$;
- 2) сложение: $\#1=\#2+\#3$;
- 3) вычитание: $\#1=\#2-\#3$;
- 4) умножение: $\#1=\#2*\#3$;
- 5) деление: $\#1=\#2/\#3$.

В операции деления недопустимо использование нулевого значения делителя, в противном случае система управления выдаст сообщение об ошибке.

Поддерживаемые математические функции:

- 1) арксинус: $\#1=ASIN[\#2]$;
- 2) синус: $\#1=SIN[\#2]$;
- 3) арккосинус: $\#1=ACOS[\#2]$;
- 4) косинус: $\#1=COS[\#2]$;
- 5) арктангенс: $\#1=ATAN[\#2]$;
- 6) тангенс: $\#1=TAN[\#2]$;
- 7) квадратный корень: $\#1=SQRT[\#2]$;
- 8) абсолютное значение: $\#1=ABS[\#2]$;
- 9) округление: $\#1=ROUND[\#2]$;

- 10) экспонента: #1=EXP[#2];
- 11) натуральный логарифм: #1=LN[#2];
- 12) округление в меньшую сторону: #1=FIX[#2];
- 13) округление в большую сторону: #1=FUP[#2].

Аргументами тригонометрических функций SIN, COS и TAN являются значения в угловых градусах.

Система управления контролирует корректность аргументов функций ASIN, ACOS, TAN, SQRT, а также LN. Числовое значение аргумента функций ASIN и ACOS должно соответствовать диапазону [-1; 1]. В качестве аргумента функции TAN недопустимо значение 90. В качестве аргумента функции SQRT недопустимы отрицательные значения. В качестве аргумента функции LN допускаются значения больше нуля.

Функция ROUND в операциях присваивания округляет значения выражений и переменных до целого. Если функция ROUND используется после буквенных адресов, округление выполняется до 3 знаков после точки для метрических единиц и 4 знаков после точки для дюймовых единиц.

Поддерживаемые логические операторы:

- 1) конъюнкция И: #1=#2 AND #3 (альтернативное обозначение: &);
- 2) дизъюнкция ИЛИ: #1=#2 OR #3 (альтернативное обозначение: |).

Поддерживаемые операторы сравнения (применяются в условных выражениях):

- 1) равно: EQ (альтернативное обозначение: ==);
- 2) не равно: NE (альтернативное обозначение: <>);
- 3) больше: GT (альтернативное обозначение: >);
- 4) больше или равно: GE (альтернативное обозначение: >=);
- 5) меньше: LT (альтернативное обозначение: <);
- 6) меньше или равно: LE (альтернативное обозначение: <=).

Переменные могут быть заданы в качестве значений после буквенных адресов, например, X#1. Также в качестве значений после буквенных адресов могут использоваться выражения в скобках, например, Y[#2+SIN[#3]*#4]. Целочисленные значения G/M-кодов также могут задаваться переменными, например, G#4.

При составлении математических или логических выражений необходимо соблюдать порядок расстановки скобок. Если количество открывающих и закрывающих скобок не совпадает в одном выражении, система управления выдаст сообщение об ошибке.

При вычислении выражений порядок математических действий выполняется в соответствии с правилами математики – парсинг алгебраических выражений выполняется методом «Обратной польской нотации».

6.34.2. Безусловный и условный переходы

Оператор безусловного перехода GOTO предназначен для передачи управления определённому блоку программы, указанному после слова GOTO, например, при выполнении оператора GOTO20 осуществится переход к блоку N20. Блок, к которому осуществляется безусловный переход, обязательно должен содержать адрес N.

Ниже представлен пример использования безусловного перехода.

N10 G00 X0 Z0;

N12 GOTO16;

N14 M30;

N16 G01 U100;

В блоке N10 выполняется ускоренное перемещение инструмента в начало активной рабочей системы координат. В блоке N12 осуществляется

безусловный переход к блоку N16, в котором выполняется рабочее линейное перемещение на 100 единиц в положительном направлении оси x.

Условный переход обеспечивает ветвление управляющей программы по критерию выполнения условного выражения. Оператор условного перехода IF задаётся в начале блока, после чего обязательно указывается условное выражение в скобках. Условное выражение должно содержать как минимум один оператор сравнения. Если условное выражение выполняется (значение условного выражения истинно), система управления выполняет правую часть блока (всё, что следует за условным выражением). Если условное выражение не выполняется (значение условного выражения ложно), управление передаётся блоку, следующему за блоком с условным выражением.

Ниже представлены примеры использования условного перехода.

```
N15 IF [#1 GE 360] THEN G00 Z0;
```

В блоке N15 проверяется условие: «если значение переменной #1 больше или равно 360». В случае истинности условного выражения осуществляется ускоренное позиционирование на отметку z=0.

Наличие оператора THEN после условного выражения необязательно.

```
N10 #3=20;
```

```
N12 IF [#1 LE #2 AND #2 GT 45] GOTO#3;
```

В блоке N10 переменной #3 присваивается значение 20. В блоке N12 проверяется условие: «если значение переменной #1 меньше или равно значению переменной #2 И значение переменной #2 больше 45». В случае истинности условного выражения осуществляется переход к блоку N20.

В представленном примере параметр перехода после оператора GOTO задан переменной, однако номер блока не может быть задан переменной, т.е. блок, к которому осуществляется переход, должен содержать адрес N с явным числовым значением (N20).

По аналогии с системой управления «СтанкоМашКомплекс» (г. Тверь) симулятор допускает использование многострочных условий вида:

IF [...] {... ...}	IF [...] {...}	IF [...] {...}
ELIF [...] {... ...}	ELIF [...] {...}	ELIF [...] {...}
ELSE {... ...}	ELSE {...}	ELSE {...}

В данном формате записи количество подусловий ELIF неограниченно. Наличие фигурных скобок обязательно. Между фигурными скобками может быть задано любое выражение или последовательность G-кодов в виде блоков.

Симулятор позволяет выполнять безусловные переходы в тело многострочных условий. При этом выполнение кода осуществляется по аналогии с языком программирования BASIC.

6.34.3. Циклы с предусловием

Цикл с предусловием программируется с помощью синтаксической конструкции WHILE...DO...END. Оператор WHILE задаётся в начале блока, после чего обязательно указывается условное выражение в скобках. Условное выражение должно содержать как минимум один оператор сравнения. После условного выражения указывается оператор начала цикла DO_n, где n – целочисленный индекс вложенного цикла от 1 до 3. После данного блока идут блоки, выполняемые в цикле. Цикл заканчивается оператором END_n, где n – индекс, соответствующий индексу начала цикла.

Если условное выражение выполняется (значение условного выражения истинно), управление передаётся блоку, следующему за блоком начала цикла (DO). Если условное выражение не выполняется (значение условного

выражения ложно), управление передаётся блоку, следующему за блоком окончания цикла (END).

Цикл DO...END выполняется до тех пор, пока условное выражение истинно.

Ниже представлен пример использования цикла с предусловием.

```
N01 #1=300;  
N02 #2=200;  
N03 WHILE [#1 GT #2] DO1;  
N04 #1=#1-10  
N05 U100;  
N06 W-100;  
N07 U-100;  
N08 W100;  
N09 END1;  
N10 M30;
```

В блоке N1 переменной #1 присваивается значение 300. В блоке N2 переменной #2 присваивается значение 200. В блоке N3 начинается цикл с предусловием. После оператора WHILE следует условное выражение: «если значение переменной #1 больше значения переменной #2», в случае истинности которого управление передаётся блоку N4. В блоке N4 значение переменной #1 уменьшается на 10 (это происходит на каждой итерации цикла). В блоках N5, N6, N7 и N8 инструмент перемещается по квадратному контуру в режиме относительного позиционирования в плоскости x-z. Блок N9 завершает цикл оператором END. Цикл DO...END выполняется 10 раз подряд. На последней итерации цикла значение переменной #1 равно 200, следовательно, условие цикла не выполняется, и управление передаётся блоку N10, завершающему выполнение программы кодом M30.

Цикл DO...END может программироваться без предусловия WHILE. Допускается использование до трёх вложенных циклов, при этом следует соблюдать порядок индексации операторов DO и END.

СОДЕРЖАНИЕ

1. Общее описание программного продукта	1
2. Описание графического интерфейса пользователя	3
3. Принцип управления виртуальной камерой с помощью мыши	17
4. Режим построения модели траекторий режущего инструмента или режим отладки управляющей программы	18
5. Режим привязки инструмента с помощью измерительного датчика	20
6. Краткая справочная информация по программированию операций токарной обработки в симуляторе	23
6.1. Команды линейной интерполяции G00 и G01	25
6.2. Команды круговой интерполяции G02 и G03	26
6.3. Программирование фасок и закруглений	27
6.4. Команда задержки выполнения программы G04	28
6.5. Команды выбора системы мер G20, G21	29
6.6. Команда возврата в референтную точку G28	29
6.7. Команды однопроходного нарезания резьбы G32 и G34	30
6.8. Команды управления режимом коррекции на радиус инструмента G40, G41, G42	31
6.9. Команда смещения системы координат/ограничения скорости вращения шпинделя G50	36
6.10. Команда установки локальной системы координат G52	37
6.11. Команда перехода в систему координат станка G53	38
6.12. Команды переключения рабочей системы координат G54-G59	39

6.13. Команда вызова макропрограммы G65	40
6.14. Команды переключения режима вращения шпинделя G96, G97	42
6.15. Команды переключения режима подачи G98, G99	42
6.16. Команды приостановки выполнения программы M00, M01	43
6.17. Команды завершения выполнения программы M02, M30	43
6.18. Команды управления вращением шпинделя M03, M04, M05	44
6.19. Команда переключения инструмента T	44
6.20. Команды управления подачей СОЖ M08, M09	44
6.21. Команды управления подпрограммами M97, M98, M99	45
6.22. Цикл черновой продольной контурной обработки G71	46
6.23. Цикл черновой поперечной контурной обработки G72	48
6.24. Цикл контурной обработки G73	49
6.25. Цикл чистовой контурной обработки G70	50
6.26. Цикл автоматической обработки боковых канавок G75	51
6.27. Цикл автоматической обработки торцевых канавок G74	52
6.28. Цикл автоматического нарезания резьбы G76	53
6.29. Цикл продольной черновой обработки G90	54
6.30. Цикл торцевой черновой обработки G94	55
6.31. Цикл нарезания резьбы G92	56
6.32. Стандартный цикл многопроходного (прерывистого) сверления G83	57
6.33. Команда отмены стандартного цикла G80	57
6.34. Параметрическое программирование	57
6.34.1. Работа с переменными	58
6.34.2. Безусловный и условный переходы	60
6.34.3. Циклы с условием	62