

Разработка станочных циклов нарезания резьбы для фрезерных станков с ЧПУ¹²

П.А. Никитешкин,
доц., к.т.н., pnikishchkin@gmail.com
А.С. Григорьев,
вед. инж., к.т.н., grigorievantan@mail.ru
А.Э. Эрднеев,
техн., and_erd@mail.ru
МГТУ «СТАНКИН», г. Москва

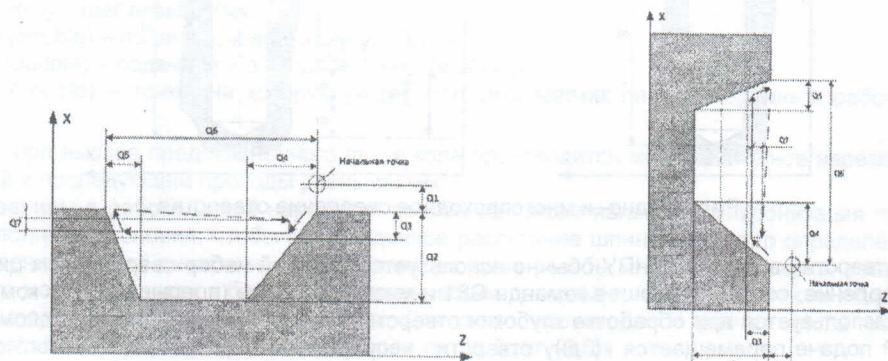
В работе исследованы основные циклы, содержащиеся в современных системах ЧПУ для выполнения различных технологических операций. Рассмотрены практические аспекты разработки станочных циклов в системе ЧПУ АксиОМА Контрол с применением языка ISO 7-bit, а также языка высокого уровня, в частности, реализующих операции нарезания резьбы метчиком.

The article describes the main cycles, which are contained in modern CNC systems for various manufacturing operations. Practical aspects of the development of machine cycles in the CNC system AxiOMA Ctrl using the ISO 7-bit language, as well as high-level language have been considered, in particular, implementing the operation tapping.

В структуре современных АСУ ТП, в особенности в сфере металлообработки, важное место занимают системы числового программного управления, реализующие выполнение одной из основных составляющих технологического процесса – технологических операций механообработки. Современные тенденции развития машиностроительного оборудования и в первую очередь объектов металлообрабатывающей промышленности характеризуются повышением уровня автоматизации, внедрением качественно новых станков и обрабатывающих центров. Современные системы числового программного управления, выпускаемые отечественными и зарубежными производителями, также неизменно развиваются и совершенствуются.

Создание управляющих программ в системах ЧПУ в большинстве систем производится на языке ISO 7 bit, или G- и M- кодах. Язык G- и M-кодов основывается на положениях Международной организации по стандартизации (ISO) и Ассоциации электронной промышленности (EIA). Официально этот язык считается стандартом для американских и европейских производителей оборудования с ЧПУ. Однако, производители систем ЧПУ хоть и придерживаются этих стандартов для описания основных функций, но допускают собственные доработки и отступления от правил, когда речь заходит о каких-либо специальных возможностях своих систем. Системы ЧПУ Fanuc (Япония) были одними из первых, адаптированных под работу с G- и M-кодами ISO и использующими этот стандарт наиболее полно. Системы ЧПУ других известных производителей, например Heidenhain и Sinumerik (Siemens), также имеют возможности по работе с G- и M-кодами, однако у каждого производителя часть кодов могут различаться – так называемые «диалекты» языка. Помимо стандартного языка, во многих системах ЧПУ реализуются языки более высокого уровня (обычно на базе C++), с помощью которых есть возможность реализации более сложных программ, с использованием циклических действий, обработки условий, введения переменных и т.д [1,3].

Одной из характеристик систем ЧПУ, позволяющей реализовывать большое количество технологических операций, является наличие встроенных циклов обработки, – т.е. набора специализированных команд, позволяющих написать УП, вводя только необходимые параметры. На сегодняшний день практически все системы ЧПУ имеют встроенный стандартный набор циклов для различных видов обработки. Они упрощают процесс написания УП и экономят время за счет выполнения множества операций одним кадром. Некоторые циклы стандартизованы, хотя большинство из них разрабатываются производителями станков и систем ЧПУ самостоятельно, без оглядки на форматы других компаний, исходя из собственных возможностей и желаний. Поэтому на разных станках, одинаковые, по сути, циклы могут вызываться по-разному, что, конечно же, затрудняет программирование. В представленной работе мы рассмотрим циклы для обработки отверстий, использующиеся на подавляющем большинстве современных станков с ЧПУ.



а) обработка продольной выточки

б) обработка торцевой выточки

Рис. 1 Базовые циклы для токарной обработки – обработка выточки

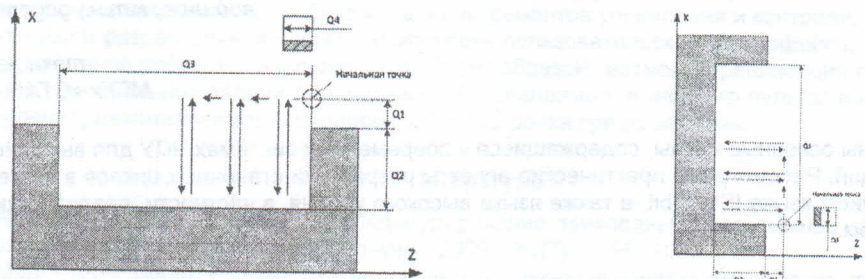
¹ Работа выполнена в рамках Стипендии Президента РФ молодым ученым и аспирантам, договор №СП-2625.2015.5.

² Работа выполнена при финансовой поддержке Министерства образования и науки РФ в рамках государственного задания в сфере научной деятельности.

Циклы представляют собой параметризованные G-команды, выполняющие сложную последовательность операций по обработке детали. Использование циклов во многих случаях позволяет значительно упростить написание управляющей программы для обработки детали, имеющей стандартные геометрические элементы (выточки, последовательности отверстий, фрезерованные карманы).

Постоянными циклами называются специальные макропрограммы, заложенные в УЧПУ для выполнения стандартных операций механической обработки. Глобально циклы можно разделить на основные виды технологических операций: токарные, фрезерные, и циклы обработки отверстий.

Циклы токарной обработки включают в себя обработку выточек (рис. 1): продольной выточки (команда G281), обработку торцевой выточки (команда G282), а также обработку канавок (рис. 2): обработку продольной канавки (команда G288), а также обработку торцевой канавки (команда G289) [2].

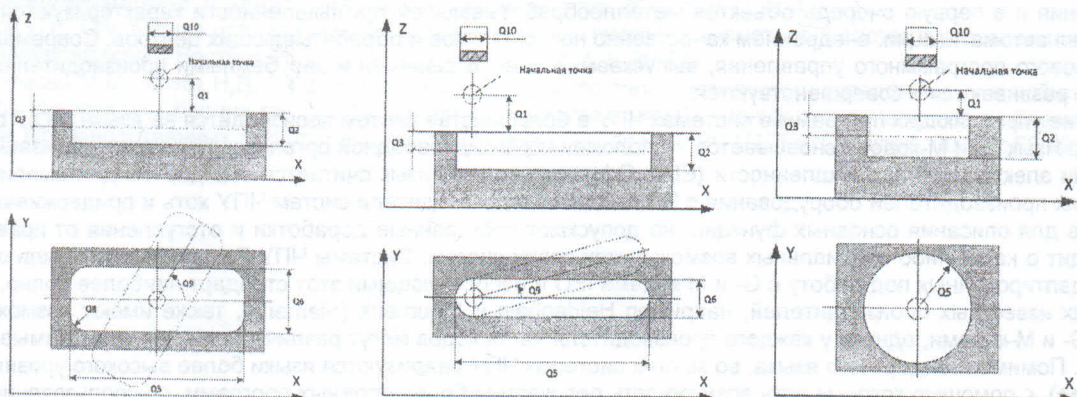


а) обработка продольной канавки

б) обработка торцевой канавки

Рис. 2 Базовые циклы для токарной обработки – обработка канавок

Базовые циклы для фрезерной обработки (рис. 3) обычно включают в себя реализацию операций по фрезерованию прямоугольного кармана (команда G87), фрезерованию паза (команда G88) и фрезерованию круглого кармана (команда G89) [2].



а) фрезерование прямоугольного кармана

б) фрезерование паза

в) фрезерование круглого кармана

Рис. 3 Базовые фрезерные циклы

Создание и обработка отверстий реализуется путем использования циклов сверления, растачивания и нарезания резьбы.

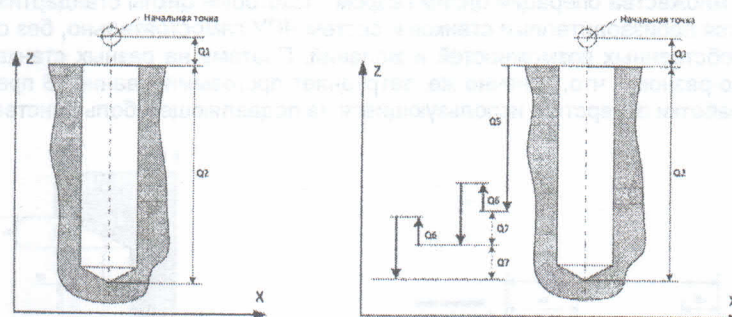


Рис. 4 Одно- и многопроходное сверление отверстия

Для создания отверстий в системах ЧПУ обычно используется базовый набор сверлильных циклов, куда входит однопроходное сверление, соответствующее команде G81 и многопроходное (прерывистое), - команда G83. Прерывистое сверление используется при обработке глубоких отверстий. При обычном (однопроходном) сверлении инструмент на рабочей подаче перемещается ко дну отверстия непрерывно. В цикле прерывистого сверления инструмент поднимается вверх через определенные интервалы для удаления стружки, поскольку, если производится сверление глубокого отверстия (глубина отверстия больше трех диаметров сверла), то есть вероятность, что стружка не успеет выйти из отверстия и произойдет поломка инструмента.

Нарезание резьбы метчиками является самым распространённым способом изготовления резьбы. Для образования резьбы необходимо придать инструменту (метчику) два движения: главное вращательное и поступательное с подачей, равной шагу резьбы за каждый оборот метчика. Нарезают резьбу тремя способами: многопроходное наре-

зание комплектом метчиков; однопроходное машинное нарезание с реверсом метчика; нарезание на проход гаечным метчиком.

Способ многопроходного нарезания резьбы является малопроизводительным, так как обычно комплект метчиков состоит из 2-3 штук. После нарезания резьбы каждым метчиком необходимо осуществлять его реверс. Способ применяется при обработке труднообрабатываемых материалов и резьбы с крупными шагами или малым сбегом, т.е. в тех случаях, когда нарезание резьбы за один проход невозможно или затруднительно.

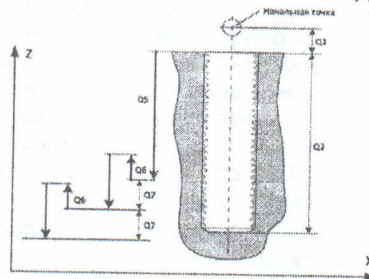


Рис. 5 Многопроходное нарезание резьбы метчиком

Однопроходное нарезание резьбы машинными метчиками является самым распространённым и осуществляется по одной из двух возможных схем обработки - основной и токарной. Основная схема характеризуется наличием главного вращения метчика, а токарная - у детали. Основной схему называют потому, что по ней можно вести обработку заготовок всех типов, практически на любом виде резьбообрабатывающего оборудования, а токарная реализуется только при обработке тел вращения на токарных станках.

По основной схеме проще обеспечить производительную многошпиндельную обработку резьбы. При работе машинными метчиками по основной схеме можно достичь значительной интенсификации процесса путём ускоренного реверсирования движения метчика.

Выбор типа метчика, определение числа метчиков в комплекте, распределение нагрузки между метчиками в комплекте, выбор схемы нарезания и назначение конструктивных элементов выполняются с учётом размеров резьбы, вида отверстия (глухое, сквозное), материала детали и условий производства.

В системе ЧПУ АксиОМА Контрол, на G-коде, с применением языка высокого уровня была создана специальная функция, позволяющая выполнять нарезание резьбы метчиком, как однопроходное так и многопроходное. Разница между двумя циклами заключается в количестве проходов, то есть один и несколько проходов за один цикл соответственно. Многопроходной цикл используется при нарезании глубокой резьбы, а также при нарезании в твёрдом материале.

Функция имеет 13 параметров, с помощью которых можно задать основные характеристики требуемой резьбы.

```
#use "metch_cycles.c"
G90
Tapping("Z", "C", 20, 0, 5, 200, 60, 0, 60, 2, 50, 25, -10);
M02
```

Основные параметры реализованной функции:

1. **strToolAx** (string) – имя оси инструмента.
2. **strSpindleAx** (string) – имя круговой оси.
3. **dRetPlane** (double) – плоскость отвода вдоль оси инструмента (абсолютная система координат).
4. **dRefPlane** (double) – плоскость поверхности начала обработки вдоль оси инструмента (абсолютная система координат).
5. **dSafeDist** (double) – безопасное расстояние до начала резьбы (мм).
6. **dDepth** (double) – длина резьбы (мм).
7. **dStep1** (double) – глубина резания при первом проходе (мм).
8. **dStepBack** (double) – величина отхода метчика перед очередным рабочим проходом. При равенстве параметра нулю отход производится до точки **dExitStep** (мм).
9. **dStep2** (double) – глубина резания при последующих проходах (мм).
10. **dPitch** (double) – шаг резьбы (мм).
11. **dTapFeed** (double) – подача при врезании (мм/мин).
12. **dRetFeed** (double) – подача при выводе метчика (мм/мин).
13. **dExitStep** (double) – точка, на которую будет выходить метчик перед очередным рабочим проходом при **dStepBack = 0**.

Таким образом, при вызове представленного выше кода производится многопроходное нарезание резьбы глубиной 200 мм, первый и последующий проходы равны 60 мм.

Важным требованием в работе цикла нарезания резьбы метчиком является синхронизация подачи и вращения шпинделя, т.е. выполнения условия, чтобы за пройденное расстояние шпиндель делал определённое число оборотов. При несоблюдении данной синхронизации произойдет срыв резьбы или, что ещё хуже, аварийная ситуация и брак в изделии, что может привести к большим экономическим потерям, в особенности если подобных операций на одном изделии выполняется большое количество. Для этого производится расчет числа оборотов шпинделя **dNumRots**, после чего вычисляется расстояние, которое он пройдёт. [5,6]

```
double dNumRots = (dDepth + dSafeDist) / dPitch;
double dSpindlePath = dNumRots * 360.0;
```

Реализация первого прохода нарезания резьбы представляет собой независимый блок, в котором производятся операции подвода инструмента к обрабатываемой поверхности на расстояние **dSafeDist**, нарезание резьбы на глубину **dStep1** при вычисленных параметрах **dSpindlePath** и **dTapFeed**, поднятие инструмента на расстояние **dStepBack** при отрицательном значении **dSpindlePath** и **dRetFeed**.


```

G90
G0 AXSTR(strToolAx)=(dRefPlane + dSafeDist)
G91
G1 AXSTR(strToolAx)=-(dStep1 + dSafeDist) AXSTR(strSpindleAx)=dSpindlePath F=dTapFeed
G1 AXSTR(strToolAx)=(dStepBack) AXSTR(strSpindleAx)=-dSpindlePath F=dRetFeed

```

Реализация последующих проходов представляет собой циклически повторяющиеся действия, в которых производится расчёт оставшихся проходов в зависимости от заданной глубины dDepth и проходов dStep1 и dStep2, вводится переменная dOst, в которое записывается оставшееся расстояние после первого прохода, создаётся цикл for для регулирования числа проходов, которое необходимо сделать станку, после чего выполняется нарезание на глубину dStep2, в dOst записывается новое значение. Далее идёт проверка, имеет ли остаток меньшее значение, чем шаг dStep2. В случае, если нет, то инструмент поднимается на dStepBack, после чего цикл повторяется, если да, то происходит проверка, равен ли остаток нулю. В первом случае инструмент достиг глубины dDepth и поднимается на безопасное расстояние над отверстием на dSafeDist, во втором случае, он поднимается на dStepBack, опускается на величину dOst, после чего поднимается на безопасное расстояние.

```

double dSch = (dDepth - dStep1) / dStep2; //количество оставшихся проходов
double dOst = dDepth - dStep1; //оставшееся расстояние
for (int iSc = 1; iSc <= dSch; iSc++) //определяет, сколько шагов надо сделать
{
    G1 AXSTR(strToolAx)=-(dStep2 + dStepBack) AXSTR(strSpindleAx)=dSpindlePath F=dTapFeed
    dOst = dOst - dStep2;
    dStepBack = dStepBack + dStep2;
    if ((dSch - iSc) < 1) //определяем, есть ли шаг, который меньше шага 2 (остаток)
    {
        if (dOst == 0) //если остаток равен 0
        {
            G1 AXSTR(strToolAx)=(dDepth + dSafeDist) AXSTR(strSpindleAx)=-dSpindlePath F=dRetFeed
        }
        else //если остаток не равен 0
        {
            G1 AXSTR(strToolAx)=(dStepBack) AXSTR(strSpindleAx)=-dSpindlePath F=dRetFeed
            G1 AXSTR(strToolAx)=-(dOst + dStepBack) AXSTR(strSpindleAx)=dSpindlePath F=dTapFeed
            G1 AXSTR(strToolAx)=(dDepth + dSafeDist) AXSTR(strSpindleAx)=-dSpindlePath F=dRetFeed
        }
    }
    else //если нет шага, который меньше шага 2
    {
        G1 AXSTR(strToolAx)=(dStepBack) AXSTR(strSpindleAx)=-dSpindlePath F=dRetFeed
    }
}

```

В случае, если глубина резания при первом проходе dStep1 равна длине резьбы dDepth, то предусмотрен однопроходной цикл. Он является более простым в реализации, так как выполняется нарезание на глубину dStep1, которая равна глубине dDepth, после чего инструмент выводится из отверстия на обратных оборотах шпинделя и отводится на безопасное расстояние dSafeDist.

```

if (dDepth == dStep1) //если глубина равна шагу один
{
    dStepBack = dStep1 + dSafeDist;
    G90
    M902 H1=1
    G0 AXSTR(strToolAx)=(dRefPlane + dSafeDist)
    G91
    G1 AXSTR(strToolAx)=-(dStep1 + dSafeDist) AXSTR(strSpindleAx)=dSpindlePath F=dTapFeed
    G1 AXSTR(strToolAx)=(dStepBack) AXSTR(strSpindleAx)=-dSpindlePath F=dRetFeed
    G90
    G0 AXSTR(strToolAx)=dRetPlane
    M903 H1=1
}

```

Были рассмотрены базовые циклы для различных видов обработки, реализуемые в системах ЧПУ. Рассмотрены основные принципы создания циклов на языке G-кода с применением языка высокого уровня в системе ЧПУ АксиО-МА Контрол. Разработаны и представлены цикл нарезания резьбы. Преимуществами разработанного цикла является рассмотрение различных вариантов задания параметров и выполнение своей задачи в соответствии с требуемыми технологиями обработки.

Литература

1. Рыбников С.В. Руководство программиста по созданию управляющих программ. Руководство по программированию. МГТУ "СТАНКИН".
2. Электронный ресурс. Адрес в интернете: <http://planetacam.ru/college/learn/1-4/>
3. Мартинов Г.М., Мартинова Л.И. Современные тенденции в области числового программного управления станочными комплексами // СТИН. 2010. №7. С. 7-10.
4. Сосонкин, В. Л., Мартинов Г.М. Системы числового программного управления // Учебное пособие. М. : Логос, 2005. - 296 с.
5. Аверьянов О.И., Клепиков В.В. Режущий инструмент. Учебное пособие - М.:МГИУ, 2007. - 144с.