DOCUMENTATION ISG-kernel

Manual
Cycles - Processing

Short Description:
CYCLES-PROC

© Copyright
ISG Industrielle Steuerungstechnik GmbH
STEP, Gropiusplatz 10
D-70563 Stuttgart
All rights reserved
www.isg-stuttgart.de
support@isg-stuttgart.de

Documentation version: 1.03
13/11/2024

Preface

Legal information

This documentation was produced with utmost care. The products and scope of functions described are under continuous development. We reserve the right to revise and amend the documentation at any time and without prior notice.

No claims may be made for products which have already been delivered if such claims are based on the specifications, figures and descriptions contained in this documentation.

Personnel qualifications

This description is solely intended for skilled technicians who were trained in control, automation and drive systems and who are familiar with the applicable standards, the relevant documentation and the machining application.

It is absolutely vital to refer to this documentation, the instructions below and the explanations to carry out installation and commissioning work. Skilled technicians are under the obligation to use the documentation duly published for every installation and commissioning operation.

Skilled technicians must ensure that the application or use of the products described fulfil all safety requirements including all applicable laws, regulations, provisions and standards.

Further information

Links below (DE)

https://www.isg-stuttgart.de/produkte/softwareprodukte/isg-kernel/dokumente-und-downloads

or (EN)

https://www.isg-stuttgart.de/en/products/softwareproducts/isg-kernel/documents-and-downloads

contains further information on messages generated in the NC kernel, online help, PLC libraries, tools, etc. in addition to the current documentation.

Disclaimer

It is forbidden to make any changes to the software configuration which are not contained in the options described in this documentation.

Trade marks and patents

The name ISG®, ISG kernel®, ISG virtuos®, ISG dirigent® and the associated logos are registered and licensed trade marks of ISG Industrielle Steuerungstechnik GmbH.

The use of other trade marks or logos contained in this documentation by third parties may result in a violation of the rights of the respective trade mark owners.

Copyright

© ISG Industrielle Steuerungstechnik GmbH, Stuttgart, Germany.

No parts of this document may be reproduced, transmitted or exploited in any form without prior consent. Non-compliance may result in liability for damages. All rights reserved with regard to the registration of patents, utility models or industrial designs.

General and safety instructions

Icons used and their meanings

This documentation uses the following icons next to the safety instruction and the associated text. Please read the (safety) instructions carefully and comply with them at all times.

Icons in explanatory text

➢ Indicates an action.

⇨ Indicates an action statement.

⚠ DANGER

Acute danger to life!

If you fail to comply with the safety instruction next to this icon, there is immediate danger to human life and health.

⚠ CAUTION

Personal injury and damage to machines!

If you fail to comply with the safety instruction next to this icon, it may result in personal injury or damage to machines.

Attention

Restriction or error

This icon describes restrictions or warns of errors.

Notice

Tips and other notes

This icon indicates information to assist in general understanding or to provide additional information.

Example

General example

Example that clarifies the text.

Programing Example

NC programming example

Programming example (complete NC program or program sequence) of the described function or NC command.

Release Note

Specific version information

Optional or restricted function. The availability of this function depends on the configuration and the scope of the version.

Contents

List of figures

1 Cycles - Introduction

Notice

Cycles are additional options and subject to the purchase of a license.

General information

Cycle call

ISG cycles are called using a cycle call:

```
L CYCLE[ NAME="..." @P1 = .. @P2 = .. ...]
```

A cycle is called by specifying the cycle name. It is also possible to parameterise the cycle input parameters to modify cycle behaviour to a special application.

This documentation contains a separate subsection for each cycle where cycle behaviour is described in greater detail. It also contains a list of the input parameters used for the cycle. Finally, a simple programming example describes how to call the cycle.

Selecting the cycle plane

A cycle is programmed independently of the currently valid plane (G17, G18, G19) and independently of the axis names configured in the NC channel.

In the cycle documentation, the axes are described by the following names for the sake of better legibility:

- The X axis describes the 1st main axis
- The Y axis describes the 2nd main axis
- The Z axis describes the 3rd main axis
- A axis is the designation for the rotary axis about the 1st main axis
- B axis is the designation for the rotary axis about the 2nd main axis
- C axis is the designation for the rotary axis about the 3rd main axis

Notice

Cycles can also be used in offset and rotated coordinate systems. These coordinate systems should only be defined using the #CS command. The #ROTATION command is not suitable for use in combination with cycles.

2 Overview

Notice

Cycles are additional options and subject to the purchase of a license.

Task

Machining cycles are provided for standard machining operations.

They are used to produce:

- drilled holes,
- pockets,
- spigots,
- slots,
- contour pockets and spigots,
- long holes
- and threads.

In addition, the deburring cycle can be used to add a chamfer to milled workpieces.

A machining cycle is called in a part program or subroutine.

3 Drilling cycles

3.1 Overview

Drilling cycles consist of the following functions:

- Centring and simple drilling [▸ 19]
- Deep hole drilling [▸ 23]
- Finish-drilling [▸ 14]
- Helical drill milling [▸ 28]
- Reaming [▸ 31]
- Thread tapping [▸ 34]
- Drill thread milling [▸ 39]
- Drill pattern cycles [▸ 43]

3.1.1 Geometry and machining data

When cycles are called, a distinction can be made between:

- geometry parameters
- machining parameters

Geometry and machining data assign input parameters for each cycle.

Geometry parameters include data to define the:

- reference plane
- retraction plane
- safety clearance
- absolute and relative machining depth required in the same way in all cycles.
- Further machining-specific coordinates

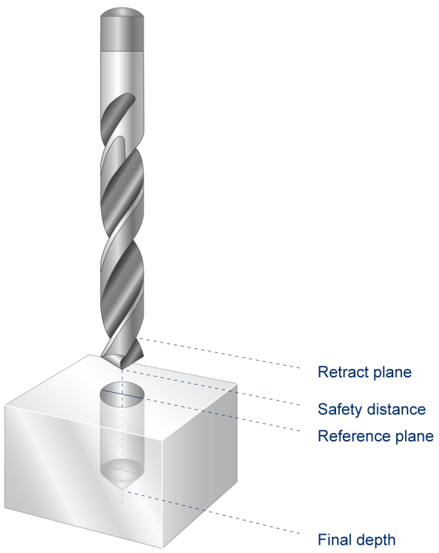

Fig. 1: General geometry parameters

Machining parameters have a specific meaning and effect. They are described in detail in each of the cycles. They mainly include technological information such as dwell times, number of feeds, rotation directions etc., in fact all the data required in the machining process.

3.2 SysDrillBoring - Finish-drilling

3.2.1 Process

When finish-drilling through/centring holes, the tool drills an existing drilled hole to the specified machining depth at the selected spindle speed and feedrate G01. The hole is produced in one operation down to the final drilling depth. When the final drilling depth is reached, an optional dwell time can be programmed. When an optionally programmed dwell time expires, the tool is retracted from the drilled hole depending on the retraction mode.

- Retraction mode 0: The tool moves out of the drilled hole in rapid traverse at speed without lifting off.
- Retraction mode 1: The tool moves out of the drilled hole at feedrate and at speed without lifting off.
- Retraction mode 2: Spindle stops and is positioned. Lift-off first takes place at feedrate by the return clearance in the direction of the hole centre.
- Retraction mode 3: The spindle is stopped and moves at rapid traverse out of the drilled hole without lifting off.

Retraction ends at the safety clearance. Finally, the tool retracts to the specified retraction plane at rapid traverse.

Notice

When used in the finish-drilling cycle, it is technologically essential that the spindle can be operated as a position-controlled spindle since it is positioned internally in the cycle on retraction and is operated as a path axis.

3.2.2 Parameterisation

Required input parameters

Input parameters	Description
@P1	Reference plane (absolute) in [mm, inch]
@P2	Retraction plane (absolute) in [mm, inch]
@P3	Safety clearance (relative to reference plane, unsigned) [mm, inch]
@P4	Final depth (relative to reference plane, unsigned) [mm, inch]

Optional input parameters

Input parameters	Description
@P17	Dwell time at final depth (in seconds)
@P62	0 position of the spindle (+X=0 (default), +Y=90, -X=180, -Y=270)
@P92	Retraction mode 0: Retract at rapid traverse (default) 1: Retract at feedrate 2: Retract at rapid traverse after orienting the spindle and approaching the return clearance (@P93/ @P94) 3: Retract at rapid traverse after spindle has stopped

@P93	Radial return clearance, always relative to drilling centre, incremental, unsigned (default value = 0) (only when @P92 = 2)
@P94	Axial return clearance, in direction of rotation axis, incremental, unsigned (default value = 0)

It is recommended using the Syntax check.to verify whether the input parameters have been correctly assigned.

3.2.3 Syntax

L CYCLE [NAME = SysDrillBoring.ecy @P.. = ..]

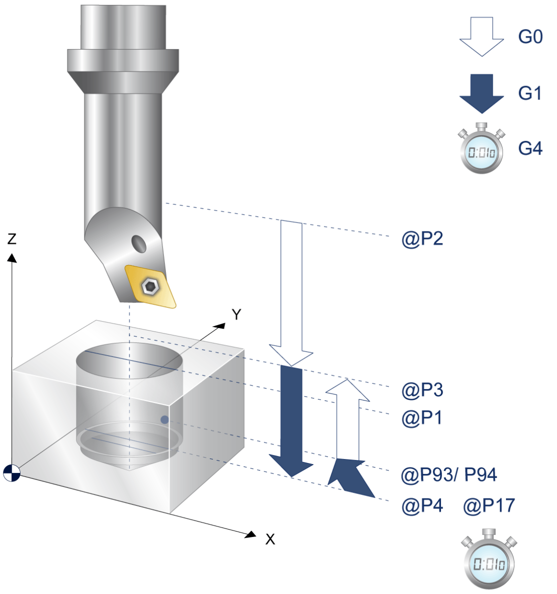

Fig. 2: Finish-drilling process

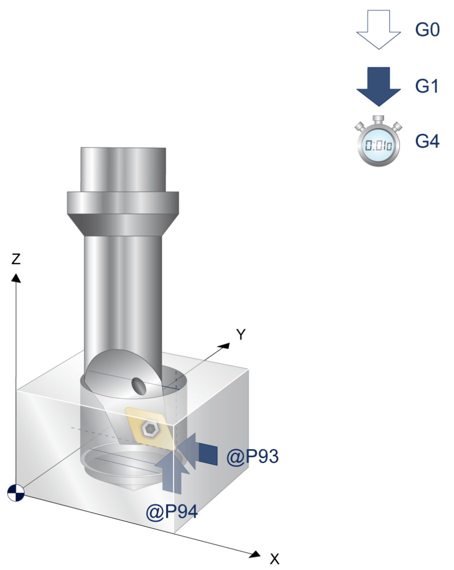

Fig. 3: Parameters 93 and 94

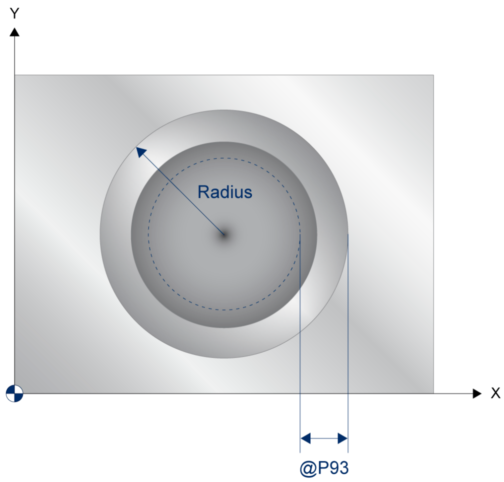

Fig. 4: Parameter 93

3.2.4 Programming example

Programing Example

Finish-drilling

```
T10 D10                              ( Tool data )
M6                                   ( Tool change )

G17 G90 G54 F250 M03 S1200           ( Technology data )
G00 Z100                             ( Go to Z retract plane )
G00 X50 Y50                          ( Go to start position )

#VAR
  ; input parameters:
  V.L.SurfacePosition = 0    ( Z position of workpiece surface )
  V.L.RetractionPlane = 20   ( Z position of retraction plane )
  V.L.SafetyClearance = 5    ( relative value of safety clearance in Z )
  V.L.DrillingDepth   = 20   ( depth )
  V.L.DwellTime       = 2    ( Dwell time )
  V.L.SetSpindel0Pos  = 270  ( 0 pos. of tool(+X=0,+Y=90,-X=180,-Y=270))
  V.L.RetractionMode  = 2    ( Retraction mode )
  V.L.RetractRadial   = 1    ( Radial retraction )
  V.L.RetractAxial    = 1    ( Axial retraction )
#ENDVAR


;cycle call
L CYCLE [NAME=SysDrillBoring.ecy  \
    @P1  = V.L.SurfacePosition    \
    @P2  = V.L.RetractionPlane    \
    @P3  = V.L.SafetyClearance    \
    @P4  = V.L.DrillingDepth      \
    @P17 = V.L.DwellTime          \
    @P62 = V.L.SetSpindel0Pos     \
    @P92 = V.L.RetractionMode     \
    @P93 = V.L.RetractRadial      \
    @P94 = V.L.RetractAxial       \
    ]


;Final position, stop of spindle
G00 Z200 M5

M30
```

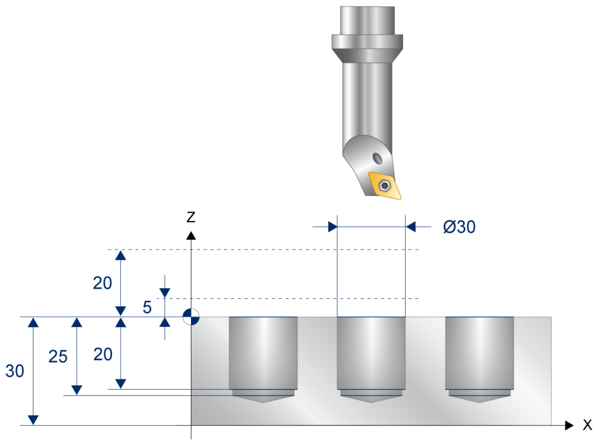

Fig. 5: Programming example

3.3 SysDrillCenterHole - centring and simple drilling

3.3.1 Process

With through/centring hole drilling, the tool drills to the specified machining depth at the selected spindle speed and feed rate G01. The hole is produced in one operation down to the final drilling depth. After that, the tool returns to the specified retraction plane after expiry of an optionally programmed dwell time for chip breaking at rapid traverse G00.

If the centring diameter is specified (@P68), the drilling depth can be automatically calculated. In this case a programmed depth is ignored. The calculation uses the specified diameter and point angle (@P87). The calculated depth is executed depending on the reference plane. If no point angle is specified, an angle of 90° is assumed. If the centring diameter is not specified, a depth must be programmed.

3.3.2 Parameterisation

Required input parameters

Input parameters	Description
@P1	Reference plane (absolute)
@P2	Retraction plane (absolute)
@P3	Safety clearance (relative to the reference plane, unsigned)
@P4	Final drilling depth (relative to the reference plane, unsigned)

Optional input parameters

Input parameters	Description
@P17	Dwell time at final drilling depth (in seconds, default value = 0)
@P68	Centring diameter (if specified, @P4 is not considered)
@P87	Point angle of drilling tool (default value = 90) If @P68 is active, @P87 is used to calculate the centring diameter. If @P68 is not active, @P87 is used to calculate the dimension from the point to the lateral surface of the tool. The calculated dimension is added to the final drilling depth. The drill radius is read from the tool memory to calculate the new final drilling depth. The radius in the tool memory may not be 0.

It is recommended to use the Syntax check.to verify whether the input parameters are correctly assigned.

3.3.3 Syntax

L CYCLE [NAME = SysDrillCenterHole.ecy @P.. = ..]

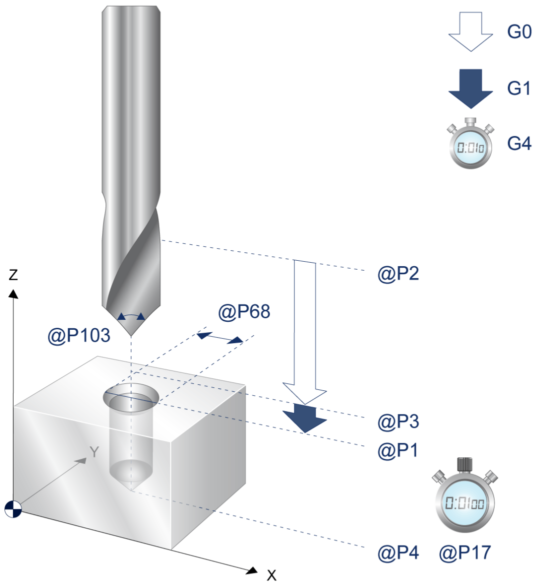

Fig. 6: Centring/drilling cycle process

3.3.4 Programming example

Programing Example

Centring and simple drilling

```
; Centring
#VAR
  ; input parameters:
  V.L.SurfacePosition = 0    ( Z position of workpiece surface )
  V.L.RetractionPlane = 20   ( Z position of retraction plane )
  V.L.SafetyClearance = 5    ( relative value of safety clearance in Z )
  V.L.CenterDiameter  = 8     ( centring diameter )
  V.L.DrillingDepth   = 25    ( drilling depth )

#ENDVAR

(------------------------------ Centring -----------------------------)

T1 D1                         ( Tool data )
M6                            ( Tool change )

G90 G54 F250 M03 S400         ( Technology data )


G00 ZV.L.RetractionPlane      ( Go to Z start position )
G00 X20 Y20                   ( 1st drilling position )

L CYCLE [NAME=SysDrillCenterHole.ecy  \
  @P1  = V.L.SurfacePosition          \
  @P2  = V.L.RetractionPlane          \
  @P3  = V.L.SafetyClearance          \
  @P68 = V.L.CenterDiameter           \
  ]

G00 Z200 M5                   ( Final position, spindle stop )

(------------------------------ Drilling ------------------------------)

T2 D2                         ( Tool data )
M6                            ( Tool change )

G90 G54 F500 M03 S5000        ( Technology data )


G00 ZV.L.RetractionPlane      ( Go to Z start position )
G00 X20 Y20                   ( 1st drilling position )

L CYCLE [NAME=SysDrillCenterHole.ecy  \
  @P1 = V.L.SurfacePosition           \
  @P2 = V.L.RetractionPlane           \
  @P3 = V.L.SafetyClearance           \
  @P4 = V.L.DrillingDepth             \
  ]

G00 Z200 M5                   ( Final position, spindle stop )


M30
```

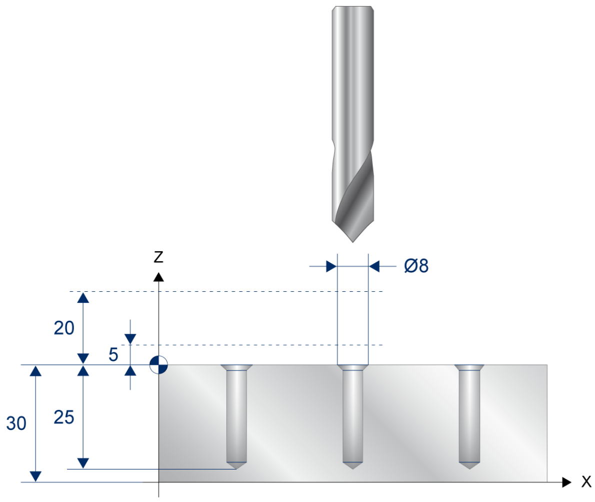

Fig. 7: Programming example for centring and simple drilling

3.4 SysDrillDeepHole - Deep hole drilling

3.4.1 Process

For deep hole drilling, the tool drills to the specified machining depth in multi-step operations at the selected spindle speed and feed rate G01. The number of infeeds is programmable. The resulting feed depth is determined internally in the cycle based on the final drilling depth. A different feed depth can be defined for the first infeed. If the first feed depth (@P10) is specified but not the number of infeeds (@P11), the parameter @P10 acts as the [mm, inch] input for feed depth. When the optional parameter @P173 is used, any Z position can be positioned in an existing pilot hole before the actual drilling process. Please note that for safety reasons the initial speed of the spindle must not exceed 100 rpm. The parameter @P175 is used to define the spindle speed for the actual drilling process. When the optional parameter @P173 is used, the parameters @P174, @P175 become mandatory parameters. They are used to prevent deformation or damage to very long drilling tools. They are always fed into the pilot hole in the opposite direction of spindle rotation (the cycle does this automatically). The parameter @P176 allows you to define a dwell time that gives the machine time to adjust the spindle speed and the direction of spindle rotation. The dwell time is also used at the base of the hole at the end of the drilling operation to achieve a total spindle stop.

3.4.2 Parameterisation

Required input parameters

Input parameters	Description
@P1	Reference plane (absolute)
@P2	Retraction plane (absolute)
@P3	Safety clearance (relative to the reference plane, unsigned)
@P4	Final drilling depth (relative to the reference plane, unsigned)
@P31	Machining mode: 1 = chip breaking 2 = chip evacuation
@P61	Return clearance for chip breaking and holding distance after chip evacuation (unsigned)

Optional input parameters

Input parameters	Description
@P10	First feed depth / depth infeed in mm/inch (unsigned) If the parameter "Number of infeeds" (@P11) is deactivated and the parameter “First feed depth” (@P10) is activated, @P10 specifies a depth infeed in mm/inch.
@P11	Number of infeeds (unsigned) Maximum infeed depth is calculated from [depth (@P4) / number of infeeds (@P11)]
@P17	Dwell time at final drilling depth (in seconds)
@P87	Point angle of the drilling tool If the point angle is specified, the dimension from the point to the lateral surface of the tool is calculation and added to the final drilling depth.

	The drill radius is read from the tool memory to calculate the new final drilling depth. The radius may not be 0.
@P92	Retraction mode: 0 = Retract at rapid traverse (default) 1 = Retract at feed rate
@P132	Feed rate for infeed in Z (values in mm/min, inch/min) Default value = Active feed rate at cycle call
@P172	Feed rate for first infeed in Z (specified in mm/min, inch/min) Default value = Active feed rate at cycle call
@P173	Z start position in pilot hole (relative to reference plane)
@P174	Feed rate of safety clearance to Z start position in pilot hole (specified in mm/min, inch/min) This parameter must be specified if a Z start position in the pilot hole was specified.
@P175	Spindle speed for deep hole drilling This parameter must be specified if a Z start position in the pilot hole was specified.
@P176	Dwell time to adapt spindle speed (specified in seconds, default = 2)
@P177	Feed rate reduction / feed rate increase in every Z infeed (specified as percentage) If the value is positive (e.g. 10), the feed is increased by this percentage value for each Z infeed. If the value is negative (e.g. -10), the feed is reduced by this percentage value for each Z infeed. n total, 50% of the original feed rate is not undershot or exceeded.
@P178	Reduction in the depth infeed (specified as a percentage) Reduces the depth infeed after the first infeed depth as a percentage until the ”Minimum infeed” @P179 is reached. @P11 may not be active.
@P179	Minimum infeed when depth infeed reduction is active (required if @P178 is active)
@P180	Length for through-drilling
@P181	Feed for through-drilling (required if @P180 is active)

It is recommended to use the Syntax check.to verify whether the input parameters are correctly assigned.

3.4.3 Syntax

L CYCLE [NAME=SysDrillDeepHole.ecy @P.. = ..]

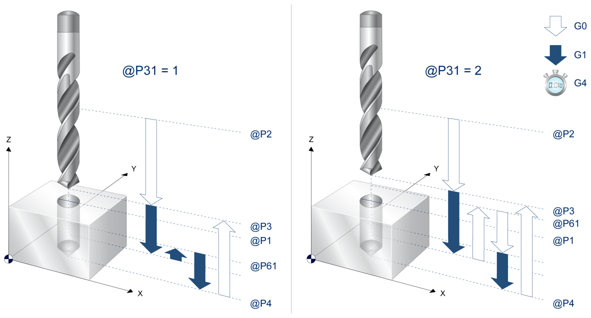

Fig. 8: Deep hole drilling process

3.4.4 Programming example

Programing Example

Deep hole drilling

```
; Deep Hole Drilling

#VAR
  ; input parameters:
  V.L.SurfacePosition = 0   ( Z position of workpiece surface )
  V.L.RetractionPlane = 20  ( Z position of retraction plane )
  V.L.SafetyClearance = 5   ( relative value of safety clearance in Z )
  V.L.DrillingDepth   = 25  ( depth )
  V.L.NumberOfFeeds   = 3   ( number of feeds )
  V.L.MachiningMode   = 2   ( machining mode )
  V.L.ReturnClearance = 0.5 ( return clearance for chip breaking/evac.)
#ENDVAR


T2 D2                       ( Tool data )
M6                          ( Tool change )

G00 G17 G90 G54 F250 M03 S1000  ( Technology data )

G00 Z100                    ( Go to z start position )
G00 X20 Y20                 ( 1st drilling position )

( drill first deep hole in three steps with chip removal )
L CYCLE [NAME=SysDrillDeepHole.ecy  \
  @P1  = V.L.SurfacePosition        \
  @P2  = V.L.RetractionPlane        \
  @P3  = V.L.SafetyClearance        \
  @P4  = V.L.DrillingDepth          \
  @P11 = V.L.NumberOfFeeds          \
  @P31 = V.L.MachiningMode          \
  @P61 = V.L.ReturnClearance        \
  ]

G00 X60 Y20                 ( 2nd drilling position )

( drill second deep hole in two steps with chip break )
```

```
  V.L.NumberOfFeeds   = 2   ( number of feeds )
  V.L.MachiningMode   = 1   ( machining mode )


L CYCLE [NAME=SysDrillDeepHole.ecy  \
  @P1  = V.L.SurfacePosition        \
  @P2  = V.L.RetractionPlane        \
  @P3  = V.L.SafetyClearance        \
  @P4  = V.L.DrillingDepth          \
  @P11 = V.L.NumberOfFeeds          \
  @P31 = V.L.MachiningMode          \
  @P61 = V.L.ReturnClearance        \
  ]


M30

; Deep Hole Drilling with pilot hole
(-----------------------------------------------------------------------)

#VAR
  ; input parameters:
  V.L.SurfacePosition = 0   (Z position of workpiece surface)
  V.L.RetractionPlane = 20  (Z position of retraction plane)
  V.L.SafetyClearance = 5   (relative value of safety clearance in Z)
  V.L.DrillingDepth   = 10  (depth pilot hole)
  V.L.NumberOfFeeds   = 3   (number of feeds)
  V.L.MachiningMode   = 2   (machining mode)
  V.L.DwellTime       = 2   (dwell time)
  V.L.ReturnClearance = 0.5(return clear. for chip breaking/evacuation)

  V.L.PilotHoleZ_Pos     = 5   (start position in Z in pilot hole)
  V.L.PilotHole_Feed     = 50  (feed till start position in pilot hole)
  V.L.Drill_SpindleSpeed = 1000 (spindle speed for drilling)
  V.L.DwellTimeSpindle   = 5   (dwell time for spindle speed/direction)

#ENDVAR

(-----------------------------------------------------------------------)

T2 D2                                     ( Tool data )
M6                                        ( Tool change )

G00 G17 G90 G54 F250 M03 S1000            ( Technology data )

G00 Z100                                  ( Go to Z start position )
G00 X20 Y20                               ( 1st drilling position )

( drill pilot hole )
L CYCLE [NAME=SysDrillDeepHole.ecy  \
  @P1  = V.L.SurfacePosition        \
  @P2  = V.L.RetractionPlane        \
  @P3  = V.L.SafetyClearance        \
  @P4  = V.L.DrillingDepth          \
  @P11 = V.L.NumberOfFeeds          \
  @P31 = V.L.MachiningMode          \
  @P61 = V.L.ReturnClearance        \
  ]

T3 D3                                     ( Tool data )
M6                                        ( Tool change )
```

```
G00 G17 G90 G54 F250 M03 S100          ( Technology data )

G00 Z100                               ( Go to z start position )
G00 X20 Y20                            ( 2nd drilling position )

( drill second deep hole in two steps with chip break )

  V.L.DrillingDepth    = 100           ( depth )
  V.L.NumberOfFeeds    = 2             ( number of feeds )
  V.L.MachiningMode    = 1             ( machining mode )


L CYCLE [NAME=SysDrillDeepHole.ecy  \
  @P1   = V.L.SurfacePosition       \
  @P2   = V.L.RetractionPlane       \
  @P3   = V.L.SafetyClearance       \
  @P4   = V.L.DrillingDepth         \
  @P11  = V.L.NumberOfFeeds         \
  @P17  = V.L.DwellTime             \
  @P31  = V.L.MachiningMode         \
  @P61  = V.L.ReturnClearance       \
  @P173 = V.L.PilotHoleZ_Pos        \
  @P174 = V.L.PilotHole_Feed        \
  @P175 = V.L.Drill_SpindleSpeed    \
  @P176 = V.L.DwellTimeSpindle      \
  ]

(-------------------------------------------------------------------)

M30
```

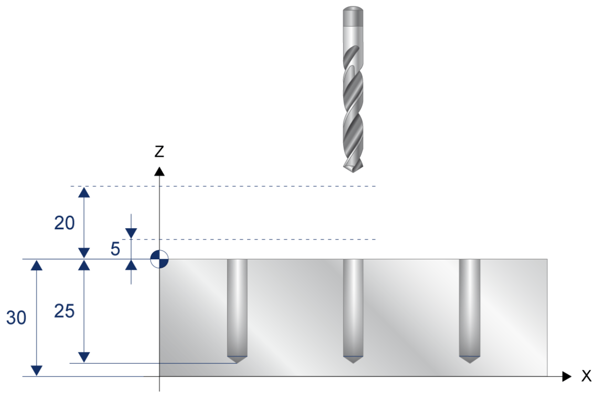

Fig. 9: Programming example: Deep hole drilling

3.5 SysDrillHelicalMilling - Helical drill milling

3.5.1 Process

With helical drill milling, the tool moves to the specified machining depth at the selected spindle speed and feedrate in a helical motion G02/G03. The hole is produced in one operation down to the final drilling depth. The end point on the X and Y axes is calculated internally in the cycle.

Optionally, a flat milled recess can be commanded for the basic hole at the end of the helical drilling operation. This is followed by positioning the tool in the drilling centre without dwell time at rapid traverse G00 and moved to the retraction plane.

3.5.2 Parameterisation

Required input parameters

Input parameters	Description
@P1	Reference plane (absolute)
@P2	Retraction plane (absolute)
@P3	Safety clearance (relative to the reference plane, unsigned)
@P4	Final drilling depth (relative to the reference plane, unsigned)
@P7	Pitch for helical motion (unsigned)
@P68	Drilled hole diameter (unsigned)

Optional input parameters

Input parameters	Description
@P12	Number of radial infeeds
@P30	Machining direction 0: Down milling (default) 1: Up milling
@P33	Machining mode at the base: 0 = Only helical 1 = Helical and base (default)

It is recommended using the Syntax check.to verify whether the input parameters have been correctly assigned.

3.5.3 Syntax

L CYCLE [NAME = SysDrillHelicalMilling.ecy @P.. = ..]

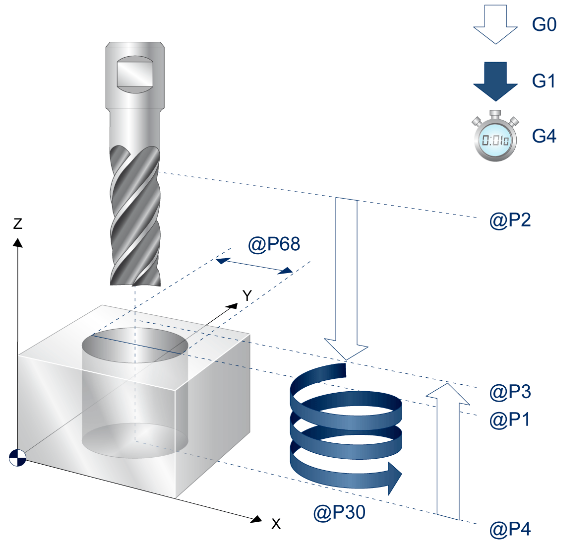

Fig. 10: Helical milling process

3.5.4 Programming example

Programing Example

Helical drill milling

```
%SysDrillHelicalMillingDemo.nc

#COMMENT BEGIN
  CycleName: SysDrillHelicalMilling.ecy
  CycleDescription: Drilling a hole by helical millling
#COMMENT END

#VAR
  ; input parameters:
  V.L.SurfacePosition = 0   ; 1  Z position of workpiece surface
  V.L.RetractionPlane = 20  ; 2  Z position of retraction plane
  V.L.SafetyClearance = 2   ; 3  relative value of safety clear. in Z
  V.L.DrillingDepth   = 15  ; 4  depth
  V.L.HelicalPitch    = 3   ; 7  pitch for helical motion
  V.L.RadialInfeed    = 6   ; 12 infeed radial
  V.L.DrillingDiam    = 30  ; 68 drilling diameter
  V.L.CW_OR_CCW       = 1   ; 30 clockwise or counter-clockwise
  V.L.MachiningMode   = 1   ; 33 machining mode at the base
#ENDVAR

; tool change
T8 D8
M6

; technology data
G17 G90 G54 S1000 M03 F4000

; positioning to the starting point
G00 Z30
G00 X20 Y20

; cycle call
L CYCLE [NAME=SysDrillHelicalMilling.ecy  \
  @P1  = V.L.SurfacePosition              \
  @P2  = V.L.RetractionPlane              \
  @P3  = V.L.SafetyClearance              \
  @P4  = V.L.DrillingDepth                \
  @P7  = V.L.HelicalPitch                 \
  @P12 = V.L.RadialInfeed                 \
  @P68 = V.L.DrillingDiam                 \
  @P30 = V.L.CW_OR_CCW                    \
  @P33 = V.L.MachiningMode                \
  ]


;Final position, stop of spindle
G00 Z150 M5

M30
```

3.6 SysDrillReaming - Reaming

3.6.1 Process

With through/blind hole drilling, the tool drills an existing drilled hole to the specified machining depth at the selected spindle speed and feedrate G01. The hole is produced in one operation down to the final drilling depth.

When the final drilling depth is reached, an optional dwell time can be programmed. Retraction from the hole can then be influenced by the parameter @P92.

Retraction ends at the safety clearance. Finally, the tool retracts to the specified retraction plane at rapid traverse G00.

3.6.2 Parameterisation

Required input parameters

Input parameters	Description
@P1	Reference plane (absolute)
@P2	Retraction plane (absolute)
@P3	Safety clearance (relative to the reference plane, unsigned)
@P4	Final drilling depth (relative to the reference plane, unsigned)

Optional input parameters

Input parameters	Description
@P17	Dwell time at final drilling depth (in seconds)
@P92	Retraction mode (default value = 1) 0: Retract at rapid traverse 1: Retract at feedrate

It is recommended using the Syntax check.to verify whether the input parameters have been correctly assigned.

3.6.3 Syntax

L CYCLE [NAME = SysDrillReaming.ecy @P.. = ..]

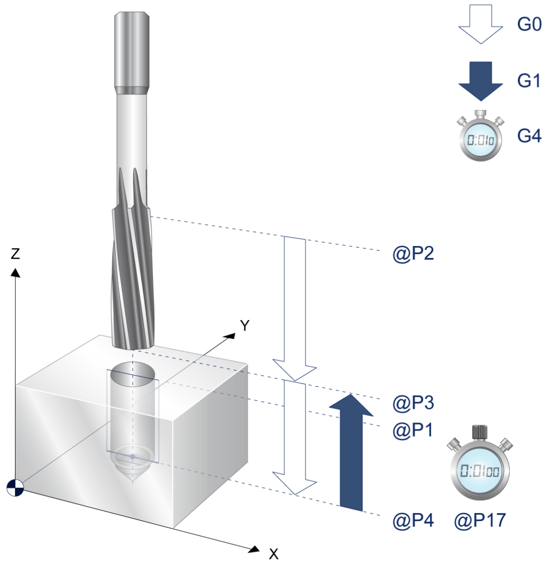

Fig. 11: Reaming process

3.6.4 Programming example

Programing Example

Reaming

```
#VAR
  ; input parameters:
  V.L.SurfacePosition = 0   ( Z position of workpiece surface )
  V.L.RetractionPlane = 20  ( Z position of retraction plane )
  V.L.SafetyClearance = 5   ( relative value of safety clearance in Z )
  V.L.DrillingDepth   = 24   ( depth )
  V.L.RetractMode     = 0    ( Retract mode (default = 1) )
#ENDVAR

T7 D7                        ( Tool data )
M6                           ( Tool change )
G17 G90 G54 F50 M03 S350     ( Technology data )

G00 Z100                     ( Go to z start position )
G00 Y20 X0                   ( 1st position retract G01 (default) )

L CYCLE [NAME=SysDrillReaming.ecy   \
    @P1 = V.L.SurfacePosition       \
    @P2 = V.L.RetractionPlane       \
    @P3 = V.L.SafetyClearance       \
    @P4 = V.L.DrillingDepth         \
    ]

G00 X40 Y20                  ( 2nd position retract G00 )
```

```
L CYCLE [NAME=SysDrillReaming.ecy    \
    @P1 = V.L.SurfacePosition        \
    @P2 = V.L.RetractionPlane        \
    @P3 = V.L.SafetyClearance        \
    @P4 = V.L.DrillingDepth          \
    @P92 = V.L.RetractMode           \
    ]

G00 Z200 M5                 ( Final position, stop of spindle )


M30
```

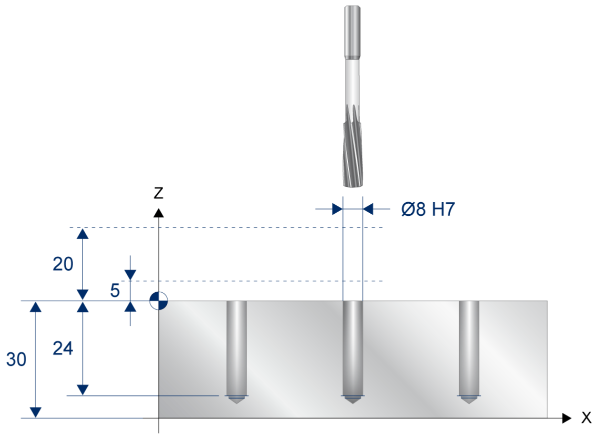

Fig. 12: Programming example: Reaming process

3.7 SysDrillTapping - Thread tapping

3.7.1 Process

In analogy to the section on chip breaking, when the final drilling depth is reached, a reversal of the rotational direction takes place. After reversal of the direction of rotation, the tool is retracted at the same speed and feedrate out of the tapped hole and moved to the specified retraction plane. Reversal of the direction of rotation after reaching the thread depth is internal in the cycle. There are 2 methods for chip breaking and evacuation and they are selectable by using the appropriate parameters.

Chip breaking

In the 1st method @P31=1, the tapping drill is retracted after every infeed depth by a programmed distance @P61 to permit chip breaking. @P11 supplies the number of infeed operations.

Evacuating chips

In the 2nd method @P31=2, the tapping drill is retracted from the hole to the reference plane + safety clearance after every infeed depth to permit chip evacuation. @P11 supplies the number of infeed operations.

Notice

For use of the thread tapping and milling cycle without compensation chuck, it is technologically necessary for the spindle to be operated as a position-controlled spindle since it is positioned internally in the cycle before thread tapping and is operated as a path axis.

3.7.2 Parameterisation

Required input parameters

Input parameters	Description
@P1	Reference plane (absolute)
@P2	Retraction plane (absolute)
@P3	Safety clearance (relative to the reference plane, unsigned)
@P4	Final drilling depth (relative to the reference plane, unsigned)
@P7	Pitch
@P24	Spindle speed for thread tapping

Optional input parameters

Input parameters	Description
@P11	Number of feeds
@P17	Dwell time at final drilling depth (in seconds)
@P18	Thread type: -1 = left-hand thread 1 = right-hand thread (default value)

@P31	Machining mode: 0 = thread tapping default (default value) 1 = thread tapping with chip breaking 2 = thread tapping with chip evacuation
@P61	Return clearance for chip breaking and holding distance after chip evacuation (unsigned) (default: 0)
@P62	Angle for spindle positioning (default: 0)

It is recommended to use the Syntax check.to verify whether the input parameters are correctly assigned.

3.7.3 Syntax

L CYCLE [NAME = SysDrillTapping.ecy @P.. = ..]

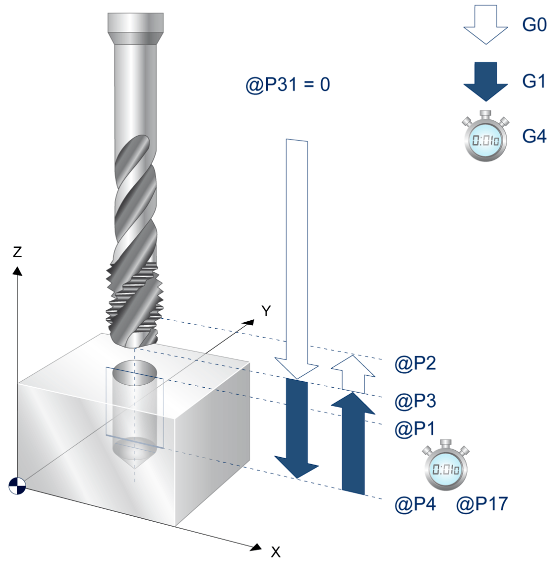

Fig. 13: Thread tapping process

3.7.4 Programming example

Programing Example

Thread tapping

```
#VAR
  ; input parameters:
  V.L.SurfacePosition          = 0   (Z position of workpiece surface )
  V.L.RetractionPlane          = 20  (Z position of retraction plane )
  V.L.SafetyClearance          = 5   (rel. value of safety clear. in Z )
  V.L.DrillingDepth            = 20  (depth )
  V.L.ThreadPitch              = 1   (pitch of the thread )
  V.L.SpindleSpeed             = 200 (spindle rotation speed )
  V.L.NumberOfFeeds            = 2   (number of feeds )
  V.L.DwellTime                = 2   (dwell time )
  V.L.ThreadType               = -1  (thread type (default = 1))
  V.L.MachiningMode            = 1   (machining mode )
  V.L.ReturnClearance          = 3   (return clear. chip breaking/evac.)
  V.L.SpindlePositioningMode   = 0   (angle of pos. (default = 0) )
#ENDVAR

T3 D3                                (Current tool data )
M6                                   (Tool change )
G17 G90 G54 S400 M03                 (Technology data )
Z200                                 (Travel to retraction plane )

( right-hand thread )
X20 Y20                              (Thread tapping position )
L CYCLE [NAME=SysDrillTapping.ecy        \
    @P1  = V.L.SurfacePosition           \
    @P2  = V.L.RetractionPlane           \
    @P3  = V.L.SafetyClearance           \
    @P4  = V.L.DrillingDepth             \
    @P7  = V.L.ThreadPitch               \
    @P24 = V.L.SpindleSpeed              \
    @P62 = V.L.SpindlePositioningMode    \
    ]

T13 D13                              (Current tool data )
M6                                   (Tool change )
G17 G90 G54 S400 M04                 (Technology data )
Z200                                 (Travel to retraction plane )

( left-hand thread )
X40 Y20                              (Thread tapping position )
L CYCLE [NAME=SysDrillTapping.ecy        \
    @P1  = V.L.SurfacePosition           \
    @P2  = V.L.RetractionPlane           \
    @P3  = V.L.SafetyClearance           \
    @P4  = V.L.DrillingDepth             \
    @P7  = V.L.ThreadPitch               \
    @P18 = V.L.ThreadType                \
    @P24 = V.L.SpindleSpeed              \
    @P62 = V.L.SpindlePositioningMode    \
    ]

T3 D3                                (Current tool data )
```

```
M6                                    (Tool change )
G17 G90 G54 S400 M03                  (Technology data )
Z200                                  (Travel to retraction plane )

; Threading with chip breaking

( Thread drilling in two steps, with chip breaking, right-hand thread )
X60 Y20                               (Thread tapping position )
L CYCLE [NAME=SysDrillTapping.ecy         \
    @P1  = V.L.SurfacePosition            \
    @P2  = V.L.RetractionPlane            \
    @P3  = V.L.SafetyClearance            \
    @P4  = V.L.DrillingDepth              \
    @P7  = V.L.ThreadPitch                \
    @P11 = V.L.NumberOfFeeds              \
    @P24 = V.L.SpindleSpeed               \
    @P31 = V.L.MachiningMode              \
    @P61 = V.L.ReturnClearance            \
    @P62 = V.L.SpindlePositioningMode     \
    ]

; Thread tapping with chip evacuation

( Thread tapping in three steps, chip evacuation, right-hand thread )
X80 Y20                               (Thread tapping position )

  V.L.NumberOfFeeds           = 3     (number of feeds )
  V.L.MachiningMode           = 2     (machining mode )

L CYCLE [NAME=SysDrillTapping.ecy         \
    @P1  = V.L.SurfacePosition            \
    @P2  = V.L.RetractionPlane            \
    @P3  = V.L.SafetyClearance            \
    @P4  = V.L.DrillingDepth              \
    @P7  = V.L.ThreadPitch                \
    @P11 = V.L.NumberOfFeeds              \
    @P24 = V.L.SpindleSpeed               \
    @P31 = V.L.MachiningMode              \
    @P61 = V.L.ReturnClearance            \
    @P62 = V.L.SpindlePositioningMode     \
    ]

Z200 M5                               (Parking position, spindle stop )

M30
```

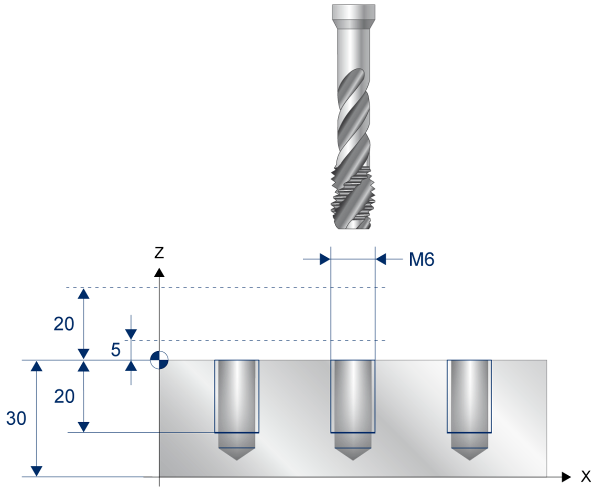

Fig. 14: Programming example: Drill tapping process

3.8 SysDrillThreadMilling - Drill thread milling

3.8.1 Process

With drill thread milling, the combination tool pre-drills at the selected spindle speed and feedrate. The hole is produced in one operation down to the final drilling depth. The end point on the X and Y axes is calculated internally in the cycle. Optionally, a dwell time can be programmed at the end of the drilling operation.

The tool is then retracted at rapid traverse to the safety plane for chip evacuation. The tool is then prepositioned to machining depth less the pitch in rapid traverse. Milling then takes place in a helical motion G02/G03 until the specified machining depth is reached.

This is followed by the tool positioned in the drilling centre without dwell time at rapid traverse G00 and moved to the retraction plane.

3.8.2 Parameterisation

Required input parameters

Input parameters	Description
@P1	Reference plane (absolute)
@P2	Retraction plane (absolute)
@P3	Safety clearance (relative to the reference plane, unsigned)
@P4	Final drilling depth (relative to the reference plane, unsigned)
@P7	Thread pitch (unsigned)
@P30	Clockwise direction (cw) or counter-clockwise (ccw) 0=Clockwise (G02) 1=Counter-clockwise (G03)
@P68	Nominal thread diameter

Optional input parameters

Input parameters	Description
@P17	Dwell time at final drilling depth (in seconds)

It is recommended using the Syntax check.to verify whether the input parameters have been correctly assigned.

3.8.3 Syntax

L CYCLE [NAME = SysDrillThreadMilling.ecy @P.. = ..]

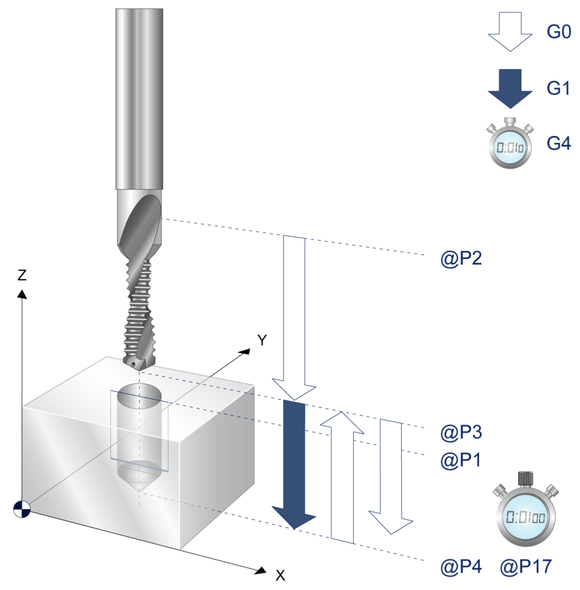

Fig. 15: Drill thread milling process 1 of 2

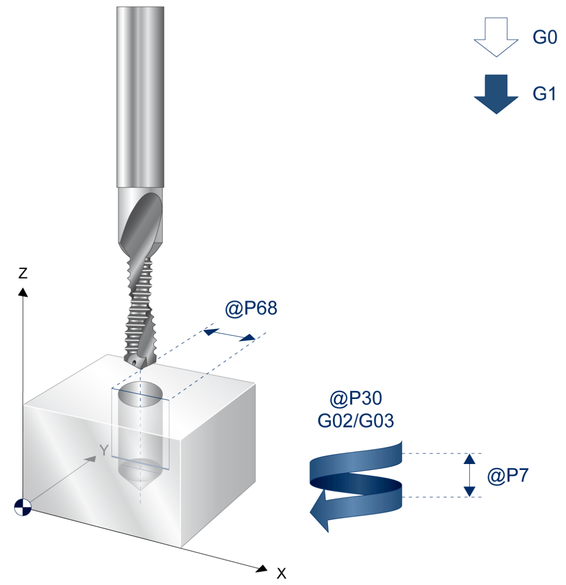

Fig. 16: Drill thread milling process 2 of 2

3.8.4 Programming example

Programing Example

Drill thread milling

```
; Drill thread milling

#VAR
  ; input parameters:
  V.L.SurfacePosition = 0   ( Z position of workpiece surface )
  V.L.RetractionPlane = 20  ( Z position of retraction plane )
  V.L.SafetyClearance = 5   ( relative value of safety clearance in Z )
  V.L.DrillingDepth   = 25   ( depth )
  V.L.ThreadPitch     = 1    ( pitch of the thread )
  V.L.CW_OR_CCW       = 0    ( clockwise or counter-clockwise )
  V.L.ThreadDiameter  = 6    ( diameter of the thread )
#ENDVAR


T5 D5                        ( Act. tool data )
M6                           ( Tool change )
G17 G90 G54 S10000 M03       ( Technology data )

G00 Z100                                ( Rapid move to start posi-
tion z )
G00 X20 Y20 F800             ( start position x y )

L CYCLE [NAME=SysDrillThreadMilling.ecy \
    @P1  = V.L.SurfacePosition          \
    @P2  = V.L.RetractionPlane          \
    @P3  = V.L.SafetyClearance          \
    @P4  = V.L.DrillingDepth            \
    @P7  = V.L.ThreadPitch              \
    @P30 = V.L.CW_OR_CCW                \
    @P68 = V.L.ThreadDiameter           \
    ]

G00 Z150 M5                  ( Park position )


M30
```

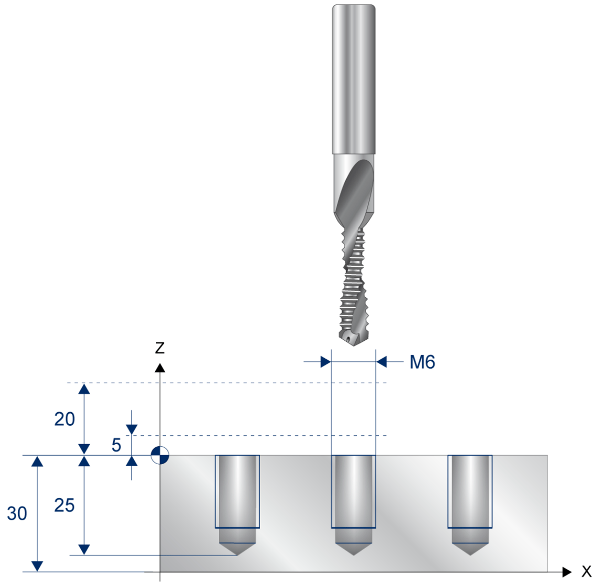

Fig. 17: Programming example: Drill thread milling

3.9 Drill pattern cycles

3.9.1 Introduction

Overview

Drill pattern cycles implement specific arrangements of drilled holes in the 2D plane.

Drill pattern cycles also calculate the drill pattern based on the input parameters. The required drilling cycle can then be called at the calculated positions.

Notice

This function is available as of CNC Build V3.01.3079.28.

Possible applications

The drill pattern cycle can implement the following drill patterns:

- Drilling a row of holes [▷ 43]
- Drilling a hole circle [▷ 47]
- Drilling a point grid [▷ 50]
- Drilling a point frame [▷ 53]

3.9.2 SysDrillPatternLine cycle - Drilling a row of holes

This cycle calculates positions on a linear pattern depending on the input and executes a drilling cycle accordingly.

The following drilling cycles can be called in combination with the cycle:

- SysDrillBoring [▷ 14]
- SysDrillCenterHole [▷ 19]
- SysDrillDeepHole [▷ 23]
- SysDrillHelicalMilling [▷ 28]
- SysDrillReaming [▷ 31]
- SysDrillThreadMilling [▷ 39]
- SysDrillTapping [▷ 34]

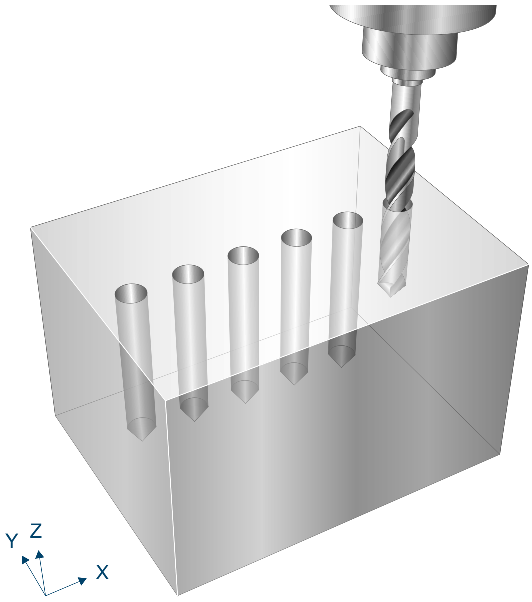

Fig. 18: Drilling a row of holes

3.9.2.1 Process

To ensure that the cycle is executed successfully, the following requirements must be met.

- The spindle speed must be selected before the cycle is started.
- The positions can be reached without collision starting from the current position at the height of the retraction plane.

When the cycle is called, the position data is calculated from the input parameters.

Every calculated position is approached at the retraction plane and the related reference cycle is executed.

3.9.2.2 Parameterisation

The section below described the parameterisation of the position data.

In addition, the parameters of the drilling cycle must also be specified when the cycle is called (see Programming example [▹ 46]). The actual drilling cycle is parameterised as described in the cycle documentation:

- SysDrillBoring [▹ 14]
- SysDrillCenterHole [▹ 19]
- SysDrillDeepHole [▹ 23]
- SysDrillHelicalMilling [▹ 28]
- SysDrillReaming [▹ 31]
- SysDrillThreadMilling [▹ 39]
- SysDrillTapping [▹ 34]

The position parameters here refer to the currently active tool coordinate system (before cycle call).

Required input parameters

Input parameters	Description
@P100	Reference cycle as character string
@P104	Distance between positions in X (incremental)

@P107	Number of positions

Optional input parameters

Input parameters	Description
@P101	Coordinate in X of the first position (absolute) Default value = current position in X
@P102	Coordinate in Y of the first position (absolute) Default value = current position in Y
@P109	Rotation of pattern in relation to the X axis (in degrees) Default value = 0

It is recommended using the Syntax check.to verify whether the input parameters have been correctly assigned.

3.9.2.3 Syntax

L CYCLE [NAME=SysDrillPatternLine.ecy @P .. = ..]

3.9.2.4 Programming example

This example describes a drilling operation along a line pattern using the example of the "Sys-DrillDeepHole" cycle

```
#VAR
  ; input parameters for pattern
  V.L.DiffX            = 10    ( Distance of the positions in X )
  V.L.NumPositions     = 4     ( Number of positions in X )

  ; input parameters for drilling cycle:
  V.L.SurfacePosition = 0    ( Z position of workpiece surface )
  V.L.RetractionPlane = 20   ( Z position of retraction plane )
  V.L.SafetyClearance = 2    ( relative value of safety clearance in Z )
  V.L.DrillingDepth   = 30    ( Depth )
  V.L.NumberOfFeeds   = 3     ( Number of feeds )
  V.L.MachiningMode   = 2     ( Machining mode )
  V.L.ReturnClearance = 0.5   ( Return clearance for chip breaking/evac.)
#ENDVAR

T2 D2                         ( Tool data )
M6                            ( Tool change )

G17 G90 G54 S1000 M03 F800    ( technology data )

G00 Z100                      ( Go to z start position )
G00 X0 Y0                     ( 1st drilling position )

L CYCLE [NAME=SysDrillPatternLine.ecy       \
  ; drill parameter of SysDrillDeepHole.ecy:
  @P1  = V.L.SurfacePosition                \
  @P2  = V.L.RetractionPlane                \
  @P3  = V.L.SafetyClearance                \
  @P4  = V.L.DrillingDepth                  \
  @P11 = V.L.NumberOfFeeds                  \
  @P31 = V.L.MachiningMode                  \
  @P61 = V.L.ReturnClearance                \

  ; pattern parameter and ref cycle:
  @P100 = "SysDrillDeepHole"                \
  @P107 = V.L.NumPositions                  \
  @P104 = V.L.DiffX                         \
  ]

G00 Z200 M5                   ( Final position, spindle stop )

M30
```

3.9.3 SysDrillPatternLine cycle - Drilling a hole circle

This cycle calculates positions on a circular pattern depending on the input and executes a drilling cycle accordingly.

The following drilling cycles can be called in combination with the cycle:

- SysDrillBoring [▸ 14]
- SysDrillCenterHole [▸ 19]
- SysDrillDeepHole [▸ 23]
- SysDrillHelicalMilling [▸ 28]
- SysDrillReaming [▸ 31]
- SysDrillThreadMilling [▸ 39]
- SysDrillTapping [▸ 34]

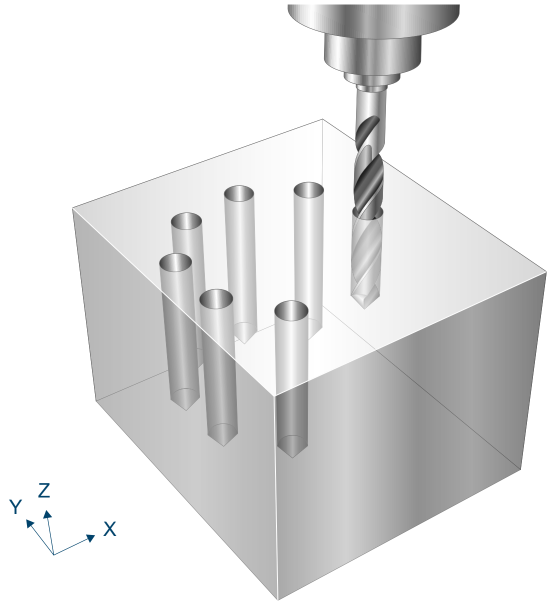

Fig. 19: Drill hole cycle

3.9.3.1 Process

To ensure that the cycle is executed successfully, the following requirements must be met.

- The spindle speed must be selected before the cycle is started.
- The positions can be reached without collision starting from the current position at the height of the retraction plane.

When the cycle is called, the position data is calculated from the input parameters.

Every calculated position is approached at the retraction plane and the related reference cycle is executed.

3.9.3.2 Parameterisation

The section below described the parameterisation of the position data.

In addition, the parameters of the drilling cycle must also be specified when the cycle is called (see Programming example [▸ 49]). The actual drilling cycle is parameterised as described in the cycle documentation:

- SysDrillBoring [▸ 14]
- SysDrillCenterHole [▸ 19]
- SysDrillDeepHole [▸ 23]
- SysDrillHelicalMilling [▸ 28]
- SysDrillReaming [▸ 31]
- SysDrillThreadMilling [▸ 39]
- SysDrillTapping [▸ 34]

The position parameters here refer to the currently active tool coordinate system (before cycle call).

Required input parameters

Input parameters	Description
@P100	Reference cycle as character string
@P107	Number of positions
@P111	Circle radius

Optional input parameters

Input parameters	Description
@P101	Coordinate in X of the circle centre point (absolute) Default value = current position in X
@P102	Coordinate in Y of the circle centre point (absolute) Default value = current position in Y
@P112	Angle of pattern in relation to the X axis (in degrees) Default value = 0
@P113	Incremental angle to the preceding position (in degrees, incremental) Default value = 360 / number of positions

It is recommended using the Syntax check.to verify whether the input parameters have been correctly assigned.

3.9.3.3 Syntax

L CYCLE [NAME=SysDrillPatternCircle.ecy @P .. = ..]

3.9.3.4 Programming example

This example describes a drilling operation along a circular pattern using the example of the "SysDrillDeepHole" cycle.

```
#VAR
  ; input parameters for pattern
  ; RefCycle = "SysDrillDeepHole"
  V.L.NumPositions       = 6    ( number of positions )
  V.L.Radius          = 30   ( circle radius )

  ; input parameters for drilling cycle:
  V.L.SurfacePosition = 0   ( Z position of workpiece surface )
  V.L.RetractionPlane = 20  ( Z position of retraction plane )
  V.L.SafetyClearance = 2   ( relative value of safety clearance in Z )
  V.L.DrillingDepth   = 30   ( depth )
  V.L.NumberOfFeeds   = 3    ( number of feeds )
  V.L.MachiningMode   = 2    ( machining mode )
  V.L.ReturnClearance = 0.5  ( return clearance for chip breaking/evac.)
#ENDVAR

T2 D2                        ( Tool data )
M6                           ( Tool change )

G17 G90 G54 F800 M03 S1000   ( Technology data )

G00 Z100                     ( Go to z start position )
G00 X0 Y0                    ( 1st drilling position )

L CYCLE [NAME=SysDrillPatternCircle.ecy     \
  ; drill parameter of SysDrillDeepHole.ecy:
  @P1  = V.L.SurfacePosition                \
  @P2  = V.L.RetractionPlane                \
  @P3  = V.L.SafetyClearance                \
  @P4  = V.L.DrillingDepth                  \
  @P11 = V.L.NumberOfFeeds                  \
  @P31 = V.L.MachiningMode                  \
  @P61 = V.L.ReturnClearance                \

  ; pattern parameter and ref cycle:
  @P100 ="SysDrillDeepHole"                 \
  @P107 = V.L.NumPositions                     \
  @P111 = V.L.Radius                        \
  ]

G00 Z200 M5                  ( Final position, stop of spindle )

M30
```

3.9.4 SysDrillPatternLine cycle - Drilling a point grid

This cycle calculates positions on a point grid depending on the input and executes a drilling cycle accordingly.

The following drilling cycles can be called in combination with the cycle:

- SysDrillBoring [▷ 14]
- SysDrillCenterHole [▷ 19]
- SysDrillDeepHole [▷ 23]
- SysDrillHelicalMilling [▷ 28]
- SysDrillReaming [▷ 31]
- SysDrillThreadMilling [▷ 39]
- SysDrillTapping [▷ 34]

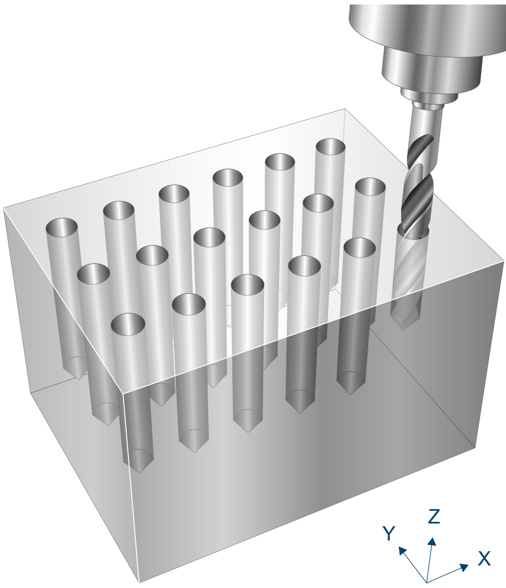

Fig. 20: Drill point grid

3.9.4.1 Process

To ensure that the cycle is executed successfully, the following requirements must be met.

- The spindle speed must be selected before the cycle is started.
- The positions can be reached without collision starting from the current position at the height of the retraction plane.

When the cycle is called, the position data is calculated from the input parameters.

Every calculated position is approached at the retraction plane and the related reference cycle is executed.

3.9.4.2 Parameterisation

The section below described the parameterisation of the position data.

In addition, the parameters of the drilling cycle must also be specified when the cycle is called (see Programming example [▸ 52]). The actual drilling cycle is parameterised as described in the cycle documentation:

- SysDrillBoring [▸ 14]
- SysDrillCenterHole [▸ 19]
- SysDrillDeepHole [▸ 23]
- SysDrillHelicalMilling [▸ 28]
- SysDrillReaming [▸ 31]
- SysDrillThreadMilling [▸ 39]
- SysDrillTapping [▸ 34]

The position parameters here refer to the currently active tool coordinate system (before cycle call).

Required input parameters

Input parameters	Description
@P100	Reference cycle as character string
@P104	Distance between positions in X (incremental)
@P105	Distance between positions in Y (incremental)
@P107	Number of positions along X
@P108	Number of positions along Y

Optional input parameters

Input parameters	Description
@P101	Coordinate in X of the first position (absolute) Default value = current position in X
@P102	Coordinate in Y of the first position (absolute) Default value = current position in Y
@P109	Rotation of pattern in relation to the X axis (in degrees) Default value = 0
@P110	Angle of inclination of pattern in relation to the Y axis (in degrees) Default value = 0

It is recommended using the Syntax check.to verify whether the input parameters have been correctly assigned.

3.9.4.3 Syntax

L CYCLE [NAME=SysDrillPatternGrid.ecy @P .. = ..]

3.9.4.4 Programming example

This example describes a drilling operation along a grid pattern using the example of the "Sys-DrillDeepHole" cycle

```
#VAR
  ; input parameters for pattern
  ; RefCycle           = "SysDrillDeepHole"
  V.L.DiffX          = 5    ( distance of the positions in X )
  V.L.DiffY          = 5    ( distance of the positions in Y )
  V.L.NumPositionsX     = 4    ( number of positions in X )
  V.L.NumPositionsY     = 4    ( number of positions in Y )

  ; input parameters for drilling cycle:
  V.L.SurfacePosition = 0   ( Z position of workpiece surface )
  V.L.RetractionPlane = 20  ( Z position of retraction plane )
  V.L.SafetyClearance = 2   ( relative value of safety clearance in Z )
  V.L.DrillingDepth   = 30   ( depth )
  V.L.NumberOfFeeds   = 3    ( number of feeds )
  V.L.MachiningMode   = 2    ( machining mode )
  V.L.ReturnClearance = 0.5  ( return clearance for chip breaking/evac.)
#ENDVAR

T2 D2                        ( Tool data )
M6                           ( Tool change )

G17 G90 G54 S1000 M03  F800  ( Technology data )

G00 Z100                     ( Go to z start position )
G00 X0 Y0                    ( 1st drilling position )

; cycle call
L CYCLE [NAME=SysDrillPatternGrid.ecy       \
  ; drill parameter of SysDrillDeepHole.ecy:
  @P1  = V.L.SurfacePosition                \
  @P2  = V.L.RetractionPlane                \
  @P3  = V.L.SafetyClearance                \
  @P4  = V.L.DrillingDepth                  \
  @P11 = V.L.NumberOfFeeds                  \
  @P31 = V.L.MachiningMode                  \
  @P61 = V.L.ReturnClearance                \

  ; pattern parameter and ref cycle:
  @P100 = "SysDrillDeepHole"                \
  @P104 = V.L.DiffX                         \
  @P105 = V.L.DiffY                         \
  @P107 = V.L.NumPositionsX                    \
  @P108 = V.L.NumPositionsY                    \
  ]

G00 Z200 M5                  ( Final position, stop of spindle )

M30
```

3.9.5 SysDrillPatternLine cycle - Drilling a point frame

This cycle calculates positions on a frame pattern depending on the input and executes a drilling cycle accordingly.

The following drilling cycles can be called in combination with the cycle:

- SysDrillBoring [▸ 14]
- SysDrillCenterHole [▸ 19]
- SysDrillDeepHole [▸ 23]
- SysDrillHelicalMilling [▸ 28]
- SysDrillReaming [▸ 31]
- SysDrillThreadMilling [▸ 39]
- SysDrillTapping [▸ 34]

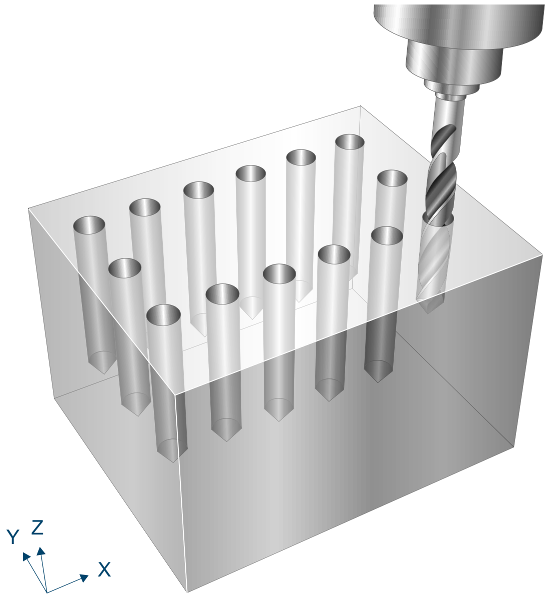

Fig. 21: Drill point frame

3.9.5.1 Process

To ensure that the cycle is executed successfully, the following requirements must be met.

- The spindle speed must be selected before the cycle is started.
- The positions can be reached without collision starting from the current position at the height of the retraction plane.

When the cycle is called, the position data is calculated from the input parameters.

Every calculated position is approached at the retraction plane and the related reference cycle is executed.

3.9.5.2 Parameterisation

The section below described the parameterisation of the position data.

In addition, the parameters of the drilling cycle must also be specified when the cycle is called (see Programming example [▶ 55]). The actual drilling cycle is parameterised as described in the cycle documentation:

- SysDrillBoring [▶ 14]
- SysDrillCenterHole [▶ 19]
- SysDrillDeepHole [▶ 23]
- SysDrillHelicalMilling [▶ 28]
- SysDrillReaming [▶ 31]
- SysDrillThreadMilling [▶ 39]
- SysDrillTapping [▶ 34]

The position parameters here refer to the currently active tool coordinate system (before cycle call).

Required input parameters

Input parameters	Description
@P100	Reference cycle as character string
@P104	Distance between positions in X (incremental)
@P105	Distance between positions in Y (incremental)
@P107	Number of positions along X
@P108	Number of positions along Y

Optional input parameters

Input parameters	Description
@P101	Coordinate in X of the first position (absolute) Default value = current position in X
@P102	Coordinate in Y of the first position (absolute) Default value = current position in Y
@P109	Rotation of pattern in relation to the X axis (in degrees) Default value = 0
@P110	Angle of inclination of pattern in relation to the Y axis (in degrees) Default value = 0

It is recommended using the Syntax check.to verify whether the input parameters have been correctly assigned.

3.9.5.3 Syntax

```
L CYCLE [NAME=SysDrillPatternFrame.ecy @P .. = ..]
```

3.9.5.4 Programming example

This example describes a drilling operation along a frame pattern using the example of the "Sys-DrillDeepHole” cycle

```
#VAR
  ; input parameters for pattern
  ; RefCycle = "SysDrillDeepHole"
  V.L.DiffX      = 5          ( distance of the positions in X )
  V.L.DiffY      = 5          ( distance of the positions in Y )
  V.L.NumPositionsX = 4          ( number of positions in X )
  V.L.NumPositionsY = 4          ( number of positions in Y )

  ; input parameters for drilling cycle:
  V.L.SurfacePosition = 0    ( Z position of workpiece surface )
  V.L.RetractionPlane = 20   ( Z position of retraction plane )
  V.L.SafetyClearance = 2    ( relative value of safety clearance in Z )
  V.L.DrillingDepth   = 30    ( depth )
  V.L.NumberOfFeeds   = 3     ( number of feeds )
  V.L.MachiningMode   = 2     ( machining mode )
  V.L.ReturnClearance = 0.5   ( return clearance for chip breaking/evac.)

#ENDVAR

T2 D2                         ( Tool data )
M6                            ( Tool change )

G17 G90 G54 S1000 M03 F800    ( Technology data )

G00 Z100                      ( Go to z start position )
G00 X0 Y0                     ( 1st drilling position )

L CYCLE [NAME=SysDrillPatternFrame.ecy      \
  ; drill parameter of SysDrillDeepHole.ecy:
  @P1  = V.L.SurfacePosition                \
  @P2  = V.L.RetractionPlane                \
  @P3  = V.L.SafetyClearance                \
  @P4  = V.L.DrillingDepth                  \
  @P11 = V.L.NumberOfFeeds                  \
  @P31 = V.L.MachiningMode                  \
  @P61 = V.L.ReturnClearance                \

  ; pattern parameter and ref cycle:
  @P100 = "SysDrillDeepHole"                \
  @P104 = V.L.DiffX                         \
  @P105 = V.L.DiffY                         \
  @P107 = V.L.NumPositionsX                    \
  @P108 = V.L.NumPositionsY                    \
  ]

G00 Z200 M5                   ( Final position, stop of spindle )

M30
```

3.9.6 SysDrillPatternArbitrary cycle - Drill any position

This cycle executes a drilling cycle at any specified position.

The following drilling cycles can be called in combination with the cycle:

- SysDrillBoring [▷ 14]
- SysDrillCenterHole [▷ 19]
- SysDrillDeepHole [▷ 23]
- SysDrillHelicalMilling [▷ 28]
- SysDrillReaming [▷ 31]
- SysDrillThreadMilling [▷ 39]
- SysDrillTapping [▷ 34]

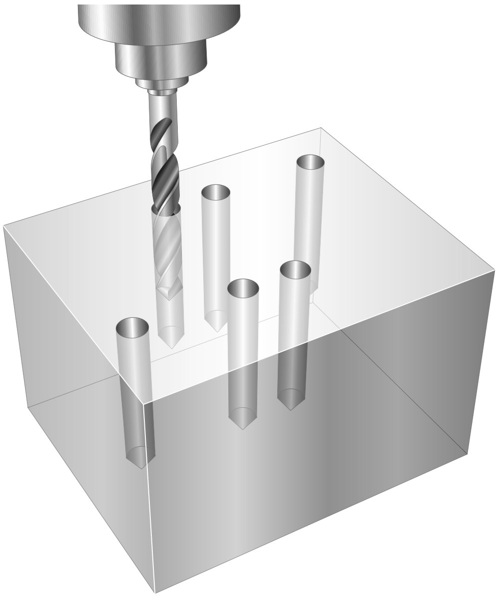

Fig. 22: Drill any position

3.9.6.1 Process

To ensure that the cycle is executed successfully, the following requirements must be met.

- The spindle speed must be selected before the cycle is started.
- The positions can be reached without collision starting from the current position at the height of the retraction plane.

Every calculated position is approached at the retraction plane at cycle call and the related reference cycle is executed.

3.9.6.2 Parameterisation

The section below described the parameterisation of the position data.

In addition, the parameters of the drilling cycle must also be specified when the cycle is called (see Programming example [▷ 58]). The actual drilling cycle is parameterised as described in the cycle documentation:

- SysDrillBoring [▷ 14]
- SysDrillCenterHole [▷ 19]
- SysDrillDeepHole [▷ 23]
- SysDrillHelicalMilling [▷ 28]
- SysDrillReaming [▷ 31]

- SysDrillThreadMilling [39]
- SysDrillTapping [34]

The position parameters here refer to the currently active tool coordinate system (before cycle call).

Required input parameters

Input parameters	Description
@P100	Reference cycle as character string

Optional input parameters

Input parameters	Description
@P101	Position in X of the 1st hole (absolute) Default value = current position in X
@P102	Position in Y of the 1st hole (absolute) Default value = current position in Y
@P103	Position in X of the 2nd hole (absolute)
@P104	Position in Y of the 2nd hole (absolute)
@P105	Position in X of the 3rd hole (absolute)
@P106	Position in Y of the 3rd hole (absolute)
@P107	Position in X of the 4th hole (absolute)
@P108	Position in Y of the 4th hole (absolute)
@P109	Position in X of the 5th hole (absolute)
@P110	Position in Y of the 5th hole (absolute)
@P111	Position in X of the 6th hole (absolute)
@P112	Position in Y of the 6th hole (absolute)
@P113	Position in X of the 7th hole (absolute)
@P114	Position in Y of the 7th hole (absolute)
@P115	Position in X of the 8th hole (absolute)
@P116	Position in Y of the 8th hole (absolute)
@P117	Position in X of the 9th hole (absolute)
@P118	Position in Y of the 9th hole (absolute)
@P119	Position in X of the 10th hole (absolute)
@P120	Position in Y of the 10th hole (absolute)

It is recommended to use the Syntax check.to verify whether the input parameters are correctly assigned.

3.9.6.3 Syntax

L CYCLE [NAME=SysDrillPatternArbitrary.ecy @P .. = ..]

3.9.6.4 Programming example

This example describes a drilling operation at any position using the example of the "SysDrillDeepHole” cycle.

```
#VAR
  ; input parameters for pattern
  ; RefCycle = "SysDrillDeepHole"
  V.L.Pos1X           = 0    ( first drill position in X )
  V.L.Pos1Y           = 10   ( first drill position in Y )
  V.L.Pos2X           = 10   ( second drill position in X )
  V.L.Pos2Y           = 10   ( second drill position in Y )
  V.L.Pos3X           = 10   ( third drill position in X )
  V.L.Pos3Y           = 0    ( third drill position in Y )

  ; input parameters for drilling cycle:
  V.L.SurfacePosition = 0   ( Z position of workpiece surface )
  V.L.RetractionPlane = 20  ( Z position of retraction plane )
  V.L.SafetyClearance = 2   ( relative value of safety clearance in Z )
  V.L.DrillingDepth   = 30   ( depth )
  V.L.NumberOfFeeds   = 3    ( number of feeds )
  V.L.MachiningMode   = 2    ( machining mode )
  V.L.ReturnClearance = 0.5  ( return clearance for chip breaking/evac.)
#ENDVAR

T2 D2                        ( Tool data )
M6                           ( Tool change )

G17 G90 G54 F800 M03 S1000   ( Technology data )
G00 Z100                     ( Go to z start position )

L CYCLE [NAME=SysDrillPatternArbitrary.ecy  \
  ; drill parameter of SysDrillDeepHole.ecy:
  @P1  = V.L.SurfacePosition                \
  @P2  = V.L.RetractionPlane                \
  @P3  = V.L.SafetyClearance                \
  @P4  = V.L.DrillingDepth                  \
  @P11 = V.L.NumberOfFeeds                  \
  @P31 = V.L.MachiningMode                  \
  @P61 = V.L.ReturnClearance                \

  ; pattern parameter and ref cycle:
  @P100 = "SysDrillDeepHole"                \
  @P101 = V.L.Pos1X                         \
  @P102 = V.L.Pos1Y                         \
  @P103 = V.L.Pos2X                         \
  @P104 = V.L.Pos2Y                         \
  @P105 = V.L.Pos3X                         \
  @P106 = V.L.Pos3Y                         \
  ]

G00 Z200 M5                  ( Final position, stop of spindle )

M30
```

4 Milling cycles

4.1 Overview

Mill any contour

The contour cycles [▷ 128] are used to mill any contour as a pocket or spigot.

Milling common geometric objects

Milling cycles consist of the following functions:

- Mill circular pocket [▷ 88]
- Mill circular spigot [▷ 70]
- Mill rectangular pocket [▷ 93]
- Mill rectangular spigot [▷ 76]
- Mill rectangular spigot [▷ 76]
- Mill rectangular spigot [▷ 109]
- Mill circumferential slots [▷ 109]
- Face milling [▷ 59]
- Mill longholes arranged in a circle [▷ 99]
- Mill multi-edge [▷ 82]
- Milling open slot [▷ 116]
- Mill circumferential slots [▷ 104]
- Milling thread [▷ 122]

4.2 SysMillFace - Face milling

4.2.1 Process

The SysMillFace cycle can be used to face mill a surface.

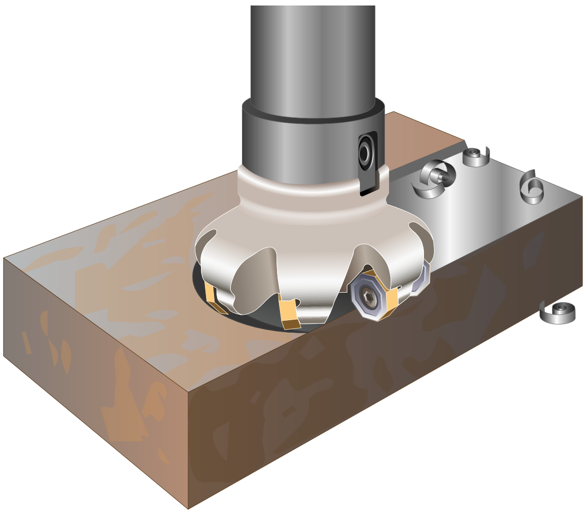

Fig. 23: Face milling process

The face milling cycle permits a number of different machining directions (@P30). In this cycle, the entire tool diameter traverses the edge of the workpiece in the machining direction.

The cycle also provides a central milling mode (@P86). Face milling is then always executed centrally along the workpiece centre. This mode is particularly suitable if the tool diameter exceeds the workpiece width. An oscillating infeed in Z can be activated for central milling (@P32).

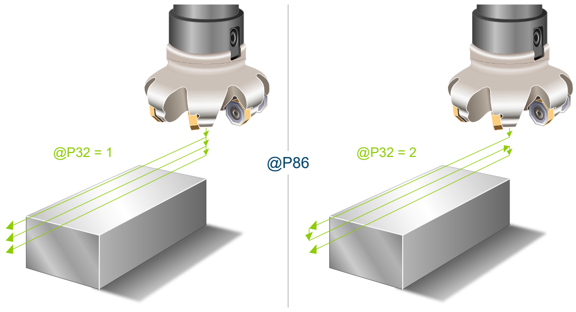

Fig. 24: Description of central milling parameters

To avoid jerky movements in the milling path and thus relieve the load on the machine tool, it can be useful to activate polynomial contouring. This also leads to an accelerated execution of the milling cycle.

The milling cycle with polynomial smoothing can be called as follows:

```
#CONTOUR MODE [DEV, PATH_DEV = V.G.WZ_AKT.R / 10 ]  ( Parameterisation )
G261                              ( Activation of polynomial contouring )
L CYCLE [NAME = SysMill... ]                                  ( Milling )
G260                            ( Deactivation of polynomial contouring )
M30
```

4.2.2 Parameterisation

Required input parameters

Input parameters	Description
@P1	Reference plane default value = 0
@P2	Retraction plane (relative to the reference plane, unsigned)
@P3	Safety clearance (relative to the reference plane, unsigned)
@P5	Maximum feed Z (unsigned)
@P30	Machining direction: 1=from left to right 2=from right to left 3=from front to rear 4=from rear to front 5=alternating left/right 6=alternating front/rear
@P31	Machining mode: " Roughing" or 1: Roughing " Finishing" or 2: Finishing (default value) "Roughing+Finishing" or 3: Roughing and finishing
@P80	1st corner X position start point
@P81	1st corner Y position start point
@P82	1st corner Z position start point
@P83	2nd corner X position end point
@P84	2nd corner Y position end point
@P85	2nd corner Z position end point

Optional input parameters

Input parameters	Description
@P6	Maximum feed XY in mm (unsigned) default value=60 percent of current tool diameter
@P15	Finishing allowance (unsigned) default value = 0.2 mm
@P32	Feed mode in Z 1: Vertical (default) 2: Oscillating (central milling only in combination with @P86)
@P42	Rotate machining by 180 degrees about zero point 0 = no rotation (default) 1 = rotate by 180 degrees about zero point
@P86	Central milling 0 = defined machining surface (default) 1 = central to the defined machining surface

It is recommended using the Syntax check.to verify whether the input parameters have been correctly assigned.

Machining directions 1-6

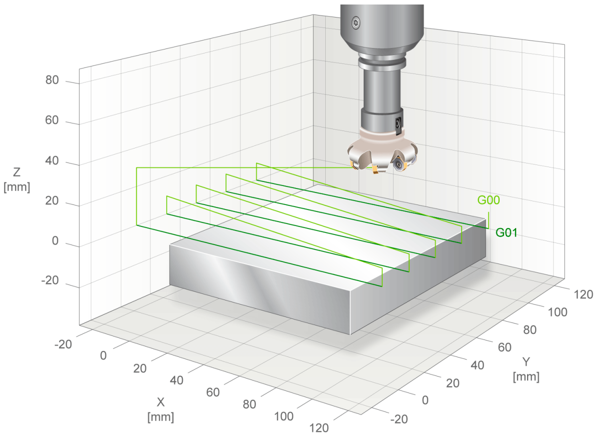

Fig. 25: Machining direction parameter 1 of 6

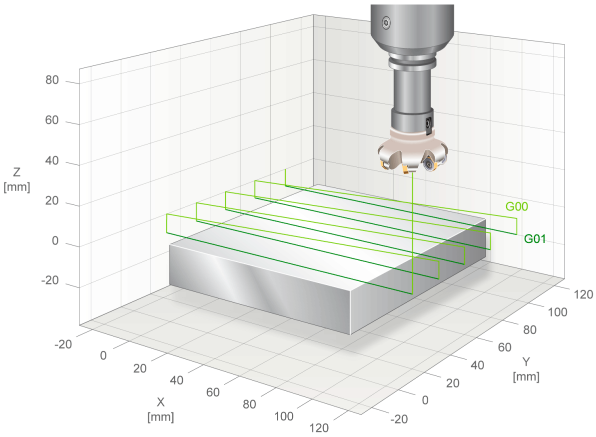

Fig. 26: Machining direction parameter 2 of 6

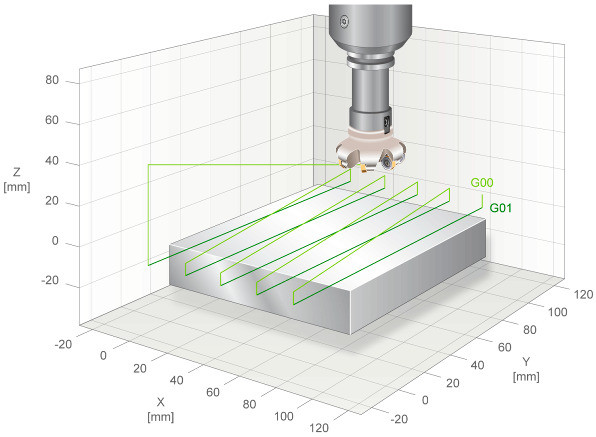

Fig. 27: Machining direction parameter 3 of 6

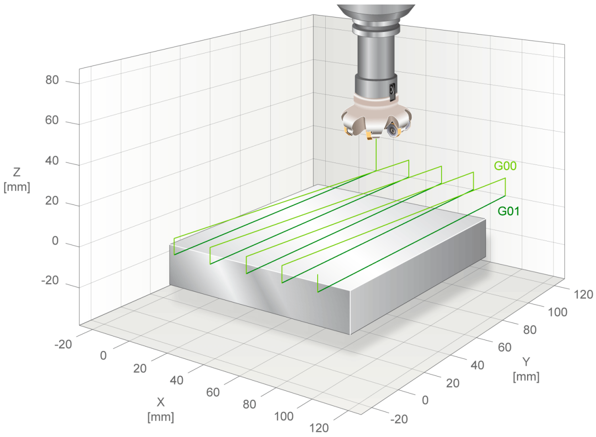

Fig. 28: Machining direction parameter 4 of 6

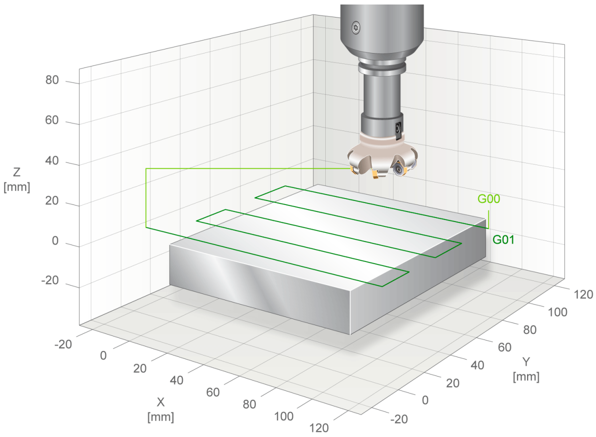

Fig. 29: Machining direction parameter 5 of 6

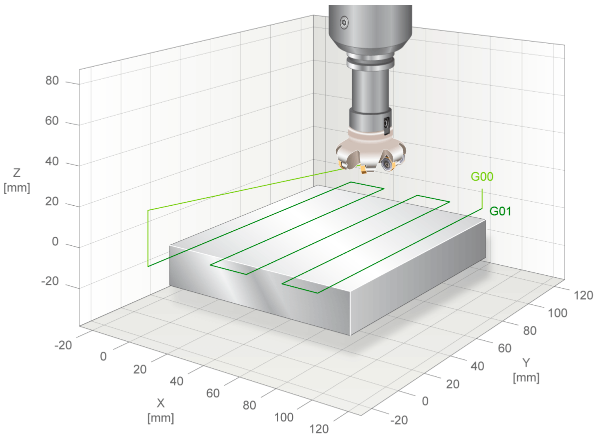

Fig. 30: Machining direction parameter 6 of 6

No.	Machining direction	Description
1	From left to right	Feed always takes place on the left-hand side outside the workpiece in the Z direction and machining is always to the right.
2	From right to left	Feed always takes place on the right-hand side outside the workpiece in the Z direction and machining is always to the left.
3	From front to rear	Feed always takes place in front of the workpiece in the Z direction and machining is always to the rear.

4	From rear to front	Feed always takes place behind the workpiece in the Z direction and machining is always to the front.
5	Alternating left and right	Feed takes place from the front left outside the workpiece in the Z direction. Machining is to the right until the tool diameter passes over the workpiece edge. The surface is then machined in zig-zag.
6	Alternating front to rear	Feed takes place from the front left outside the workpiece in the Z direction. Machining is to the rear until the tool diameter passes over the workpiece edge. The surface is then machined in zig-zag.

4.2.3 Syntax

Programing Example

Face milling

L CYCLE [NAME = SysMillFace.ecy @P.. = ..]

4.2.4 Programming example

```
; Face milling
T17 D17                         ( Tool data )
M6                              ( Tool change )
G17 G90 G54 F2000 M03 S6000     ( Technology data )
G00 Z50                         ( Go to Z start position )
G00 X-50 Y0    ( position near workpiece mill tool is outside the part )

#VAR
  V.L.SurfacePositionZ = 0     ( Z position of workpiece surface )
  V.L.RetractionPlane  = 20    ( Z position of retraction plane )
  V.L.SafetyClearance  = 5     ( relative value of safety clearance in Z)
  V.L.MaxIncrementZ    = 2     ( maximum increment of Z )
  V.L.MaxIncrementXY   = V.G.WZ_AKT.R*1.2 ( maximum increment of XY )
  V.L.FinishingOffsetZ = 0.2   ( finishing offset )
  V.L.MachiningMode    = 1     ( machining mode )
  V.L.PlungingModeZ    = 2     ( plunging mode Z (default = 1) )
  V.L.MachineDirection = 1     ( machining direction )
  V.L.Rotation         = 1     ( machining rot. 180 deg. (default = 0) )
  V.L.Corner_Start_X   = 0     ( start point X )
  V.L.Corner_Start_Y   = 0     ( start point X )
  V.L.Corner_Start_Z   = 5     ( start point Z )
  V.L.Corner_End_X     = 100   ( end point X )
  V.L.Corner_End_Y     = 100   ( end point Y )
  V.L.Corner_End_Z     = 0     ( end point Z )
  V.L.CenterPlane      = 1     ( machining on centre of workp.(def.=0))
#ENDVAR

; polynomial contouring for smooth movements
#CONTOUR MODE [DEV, PATH_DEV = V.G.WZ_AKT.R / 100]
G261

; Roughing Direction 1
L CYCLE [NAME=SysMillFace.ecy   \
@P1  = V.L.SurfacePositionZ     \
@P2  = V.L.RetractionPlane      \
@P3  = V.L.SafetyClearance      \
@P5  = V.L.MaxIncrementZ        \
@P6  = V.L.MaxIncrementXY       \
@P15 = V.L.FinishingOffsetZ     \
@P30 = V.L.MachineDirection     \
@P31 = V.L.MachiningMode        \
@P80 = V.L.Corner_Start_X       \
@P81 = V.L.Corner_Start_Y       \
@P82 = V.L.Corner_Start_Z       \
@P83 = V.L.Corner_End_X         \
@P84 = V.L.Corner_End_Y         \
@P85 = V.L.Corner_End_Z         \
]

; Roughing Direction 2

V.L.MachineDirection = 2

L CYCLE [NAME=SysMillFace.ecy   \
@P1  = V.L.SurfacePositionZ     \
@P2  = V.L.RetractionPlane      \
@P3  = V.L.SafetyClearance      \
@P5  = V.L.MaxIncrementZ        \
@P6  = V.L.MaxIncrementXY       \
@P15 = V.L.FinishingOffsetZ     \
```

```
@P30 = V.L.MachineDirection      \
@P31 = V.L.MachiningMode         \
@P80 = V.L.Corner_Start_X        \
@P81 = V.L.Corner_Start_Y        \
@P82 = V.L.Corner_Start_Z        \
@P83 = V.L.Corner_End_X          \
@P84 = V.L.Corner_End_Y          \
@P85 = V.L.Corner_End_Z          \
]

; Roughing Direction 3

V.L.MachineDirection = 3

L CYCLE [NAME=SysMillFace.ecy    \
@P1  = V.L.SurfacePositionZ      \
@P2  = V.L.RetractionPlane       \
@P3  = V.L.SafetyClearance       \
@P5  = V.L.MaxIncrementZ         \
@P6  = V.L.MaxIncrementXY        \
@P15 = V.L.FinishingOffsetZ      \
@P30 = V.L.MachineDirection      \
@P31 = V.L.MachiningMode         \
@P80 = V.L.Corner_Start_X        \
@P81 = V.L.Corner_Start_Y        \
@P82 = V.L.Corner_Start_Z        \
@P83 = V.L.Corner_End_X          \
@P84 = V.L.Corner_End_Y          \
@P85 = V.L.Corner_End_Z          \
]

; Roughing Direction 4

V.L.MachineDirection = 4

L CYCLE [NAME=SysMillFace.ecy    \
@P1  = V.L.SurfacePositionZ      \
@P2  = V.L.RetractionPlane       \
@P3  = V.L.SafetyClearance       \
@P5  = V.L.MaxIncrementZ         \
@P6  = V.L.MaxIncrementXY        \
@P15 = V.L.FinishingOffsetZ      \
@P30 = V.L.MachineDirection      \
@P31 = V.L.MachiningMode         \
@P80 = V.L.Corner_Start_X        \
@P81 = V.L.Corner_Start_Y        \
@P82 = V.L.Corner_Start_Z        \
@P83 = V.L.Corner_End_X          \
@P84 = V.L.Corner_End_Y          \
@P85 = V.L.Corner_End_Z          \
]

; Roughing Direction 5

V.L.MachineDirection = 5

L CYCLE [NAME=SysMillFace.ecy    \
@P1  = V.L.SurfacePositionZ      \
@P2  = V.L.RetractionPlane       \
@P3  = V.L.SafetyClearance       \
@P5  = V.L.MaxIncrementZ         \
@P6  = V.L.MaxIncrementXY        \
@P15 = V.L.FinishingOffsetZ      \
```

```
@P30 = V.L.MachineDirection     \
@P31 = V.L.MachiningMode        \
@P80 = V.L.Corner_Start_X       \
@P81 = V.L.Corner_Start_Y       \
@P82 = V.L.Corner_Start_Z       \
@P83 = V.L.Corner_End_X         \
@P84 = V.L.Corner_End_Y         \
@P85 = V.L.Corner_End_Z         \
]

; Roughing Direction 6

V.L.MachineDirection = 6

L CYCLE [NAME=SysMillFace.ecy   \
@P1  = V.L.SurfacePositionZ     \
@P2  = V.L.RetractionPlane      \
@P3  = V.L.SafetyClearance      \
@P5  = V.L.MaxIncrementZ        \
@P6  = V.L.MaxIncrementXY       \
@P15 = V.L.FinishingOffsetZ     \
@P30 = V.L.MachineDirection     \
@P31 = V.L.MachiningMode        \
@P80 = V.L.Corner_Start_X       \
@P81 = V.L.Corner_Start_Y       \
@P82 = V.L.Corner_Start_Z       \
@P83 = V.L.Corner_End_X         \
@P84 = V.L.Corner_End_Y         \
@P85 = V.L.Corner_End_Z         \
]

; Finishing Direction 1

V.L.MachineDirection = 1
V.L.MachiningMode    = 2

L CYCLE [NAME=SysMillFace.ecy   \
@P1  = V.L.SurfacePositionZ     \
@P2  = V.L.RetractionPlane      \
@P3  = V.L.SafetyClearance      \
@P5  = V.L.MaxIncrementZ        \
@P6  = V.L.MaxIncrementXY       \
@P15 = V.L.FinishingOffsetZ     \
@P30 = V.L.MachineDirection     \
@P31 = V.L.MachiningMode        \
@P80 = V.L.Corner_Start_X       \
@P81 = V.L.Corner_Start_Y       \
@P82 = V.L.Corner_Start_Z       \
@P83 = V.L.Corner_End_X         \
@P84 = V.L.Corner_End_Y         \
@P85 = V.L.Corner_End_Z         \
]

; Roughing + Finishing Direction 1

V.L.MachineDirection = 1
V.L.MachiningMode    = 3

L CYCLE [NAME=SysMillFace.ecy   \
@P1  = V.L.SurfacePositionZ     \
@P2  = V.L.RetractionPlane      \
@P3  = V.L.SafetyClearance      \
@P5  = V.L.MaxIncrementZ        \
```

```
@P6  = V.L.MaxIncrementXY        \
@P15 = V.L.FinishingOffsetZ      \
@P30 = V.L.MachineDirection      \
@P31 = V.L.MachiningMode         \
@P80 = V.L.Corner_Start_X        \
@P81 = V.L.Corner_Start_Y        \
@P82 = V.L.Corner_Start_Z        \
@P83 = V.L.Corner_End_X          \
@P84 = V.L.Corner_End_Y          \
@P85 = V.L.Corner_End_Z          \
]

; Roughing Direction 1 Rotated 180 degrees

V.L.MachineDirection = 1
V.L.MachiningMode    = 1

L CYCLE [NAME=SysMillFace.ecy    \
@P1  = V.L.SurfacePositionZ      \
@P2  = V.L.RetractionPlane       \
@P3  = V.L.SafetyClearance       \
@P5  = V.L.MaxIncrementZ         \
@P6  = V.L.MaxIncrementXY        \
@P15 = V.L.FinishingOffsetZ      \
@P30 = V.L.MachineDirection      \
@P31 = V.L.MachiningMode         \
@P42 = V.L.Rotation              \
@P80 = V.L.Corner_Start_X        \
@P81 = V.L.Corner_Start_Y        \
@P82 = V.L.Corner_Start_Z        \
@P83 = V.L.Corner_End_X          \
@P84 = V.L.Corner_End_Y          \
@P85 = V.L.Corner_End_Z          \
]

; Roughing Direction 1 Centre Lane

V.L.MachineDirection = 1
V.L.MachiningMode    = 1

L CYCLE [NAME=SysMillFace.ecy    \
@P1  = V.L.SurfacePositionZ      \
@P2  = V.L.RetractionPlane       \
@P3  = V.L.SafetyClearance       \
@P5  = V.L.MaxIncrementZ         \
@P6  = V.L.MaxIncrementXY        \
@P15 = V.L.FinishingOffsetZ      \
@P30 = V.L.MachineDirection      \
@P31 = V.L.MachiningMode         \
@P80 = V.L.Corner_Start_X        \
@P81 = V.L.Corner_Start_Y        \
@P82 = V.L.Corner_Start_Z        \
@P83 = V.L.Corner_End_X          \
@P84 = V.L.Corner_End_Y          \
@P85 = V.L.Corner_End_Z          \
@P86 = V.L.CenterPlane           \
]

; Roughing Direction 1 Centre Lane + oscillating depth feed

V.L.MachineDirection = 1
V.L.MachiningMode    = 1
```

```
L CYCLE [NAME=SysMillFace.ecy   \
@P1  = V.L.SurfacePositionZ     \
@P2  = V.L.RetractionPlane      \
@P3  = V.L.SafetyClearance      \
@P5  = V.L.MaxIncrementZ        \
@P6  = V.L.MaxIncrementXY       \
@P15 = V.L.FinishingOffsetZ     \
@P30 = V.L.MachineDirection     \
@P31 = V.L.MachiningMode        \
@P32 = V.L.PlungingModeZ        \
@P80 = V.L.Corner_Start_X       \
@P81 = V.L.Corner_Start_Y       \
@P82 = V.L.Corner_Start_Z       \
@P83 = V.L.Corner_End_X         \
@P84 = V.L.Corner_End_Y         \
@P85 = V.L.Corner_End_Z         \
@P86 = V.L.CenterPlane          \
]

G260

M30
```

4.3 SysMillCircularSpigot - Mill circular spigot

This cycle is used to mill a circular spigot.

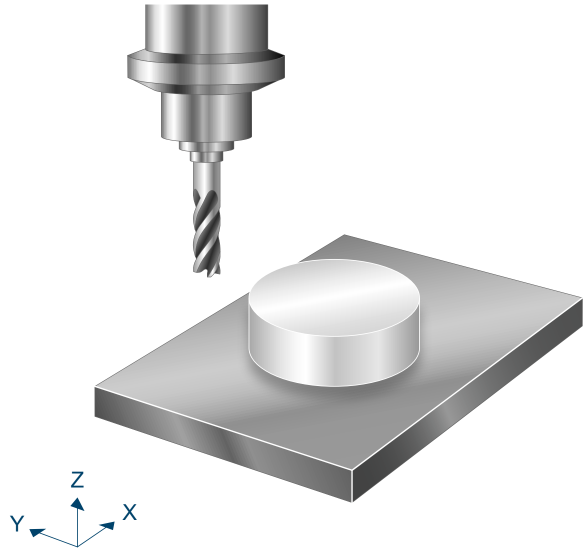

Fig. 31: Circular spigot

4.3.1 Process

To ensure that the cycle is executed successfully, the following requirements must be met.

- The tool radius is defined by V.G.WZ_AKT.R
- Tool length compensation is active.
- The spindle speed must be selected before the cycle is started.
- The start point in the XY plane (see below) can be reached without collision starting from the current position at the height of the retraction plane.
- A maximum of 3 systems may be active or defined when the cycle is called to ensure that sufficient machining coordinate systems (#CS) are available for the cycle.

Within the cycle, the tool is fed incrementally along the Z axis. The starting point here is a start point which has a safety clearance to the blank set by @P13 in the XY plane and is vertically parallel to the Y axis above the spigot centre. The blank is repeatedly approached in a circular movement for each machining height. When the machining direction is clockwise, the approach is counter-clockwise and departure is clockwise. When the machining direction is counter-clockwise, the approach is clockwise and departure is counter-clockwise.

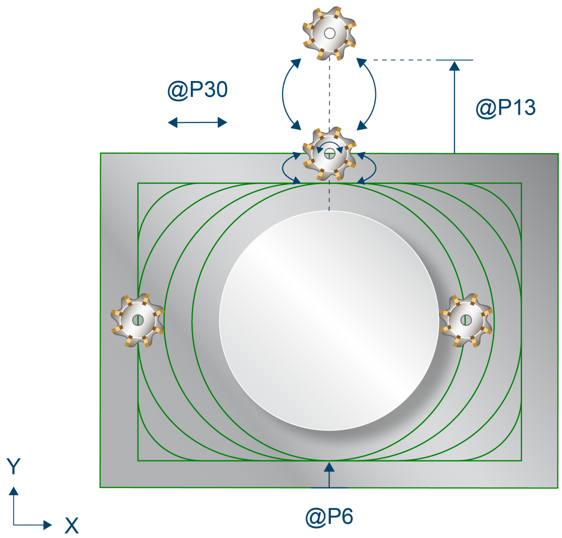

Fig. 32: Machining strategy for circular spigot

Roughing process

At the start, the start point in the XY plane is approached at the height of the retraction plane. The first feed is to the safety plane at rapid traverse along the Z axis followed by a feed to the machining height at the feed rate defined by @P21. At this start point, an interative feed takes place at the new machining height until the spigot depth @P4 is reached with the addition of the finishing allowance at the base (@P15). The maximum infeed per pass is the maximum infeed in Z at the maximum value defined in @P5.

The blank is first milled linearly for every machining height, i.e. without considering the spigot diameter (@P68). The tool then approaches closer to the centre of the spigot in a circular movement each circumferential pass without exceeding the maximum feed in XY (@P6). Linear removal continues until the linear outer dimensions of the circle with the addition of the finishing allowance at the edge (@P16) are removed.

When linear machining is completed, the spigot diameter is milled (@P68). Corner removal continues every circumferential pass around the spigot until the correct diameter (spigot diameter + finishing allowance in XY) is reached.

To avoid jerky movements in the milling path and thus relieve the machine tool, it can be useful to activate polynomial contouring during roughing. This also leads to an accelerated execution of the milling cycle. The parameterisation of polynomial smoothing should be made dependent on the selected finishing allowance in order to avoid damage to the outer contour. To avoid residue, the maximum infeed in XY must then be reduced.

The milling cycle with polynomial smoothing can be called as follows:

```
#CONTOUR MODE [DEV, PATH_DEV = V.G.WZ_AKT.R / 10 ] ( Parameterisation )
G261                                ( Activation of polynomial contouring )
L CYCLE [NAME = SysMill... ]                                   ( Roughing )
G260                               ( Deactivation of polynomial contouring )
L CYCLE [NAME = SysMill... ]                                  ( Finishing )
M30
```

Finishing process

The start point mentioned for roughing is also approached as the starting position for finishing. From there, an interative feed takes place up to the new machining height until the spigot depth @P4 is reached. The maximum infeed per pass is the maximum infeed in Z at the maximum value defined in @P5.

A distinction is made between two cases for the milling operation at each machining height:

- If no finishing allowance at the base needs to be removed at the current machining height, only the finishing allowance around the spigot is removed. As for roughing, linear removal first takes place (as required) before the correct spigot diameter is removed.
- If the current machining height also involves a removal of the finishing allowance at the base, the complete blank must be again machined at this height until the correct linear dimensions are reached. Here too, removal is first linear before the spigot diameter is removed.

4.3.2 Parameterisation

Required input parameters

Input parameters	Description
@P1	Reference plane (absolute)
@P2	Retraction plane (absolute)
@P3	Safety clearance in Z (relative to reference plane/machining height)
@P4	Depth (relative to reference plane)
@P5	Maximum infeed in Z
@P6	Maximum infeed in XY (maximum 2*V.G.WZ_AKT.R)
@P68	Spigot diameter
@P69	Length of blank (X axis)
@P70	Width of blank (Y axis)

Optional input parameters

Input parameters	Description
@P13	Safety clearance to blank in the XY plane (relative) Default value is the tool radius
@P15	Finishing allowance at base Default value = 0
@P16	Finishing allowance at edge Default value = 0
@P20	Feed rate for milling operation (specified in mm/min, inch/min)
@P21	Feed rate in Z (values in mm/min, inch/min)
@P22	Feed rate for finishing for milling in the XY plane (specified in mm/min, inch/min) The default value is the feed rate for roughing
@P23	Feed rate for finishing in Z (specified in mm/min, inch/min) The default value is the feed rate for finishing for milling in the XY plane
@P25	Spindle speed for finishing (specified in rpm)
@P30	Machining direction "ClimbMilling" or 0: Down milling (default) "UpCutMilling" or 1: Up milling
@P31	Machining mode:

	"Roughing" or 1: Roughing "Finishing" or 2: Finishing "Roughing+Finishing" or 3: Roughing and finishing (default value)
@P40	Position of centre in X Default value is the current position
@P41	Position of centre in Y Default value is the current position
@P42	Angle of inclination of blank (in degrees) Default value = 0

The function of the parameters acting in the XY plane results from the graphic below:

It is recommended to use the Syntax check.to verify whether the input parameters are correctly assigned.

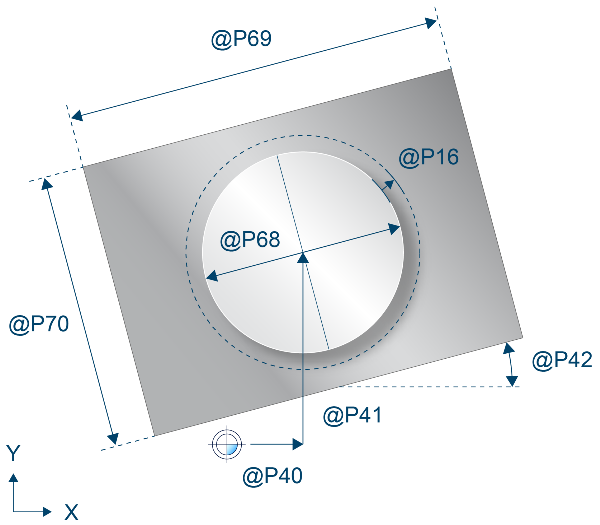

Fig. 33: Top view - circular spigot

Parameters which mainly refer to the Z axis are displayed in the graphic below:

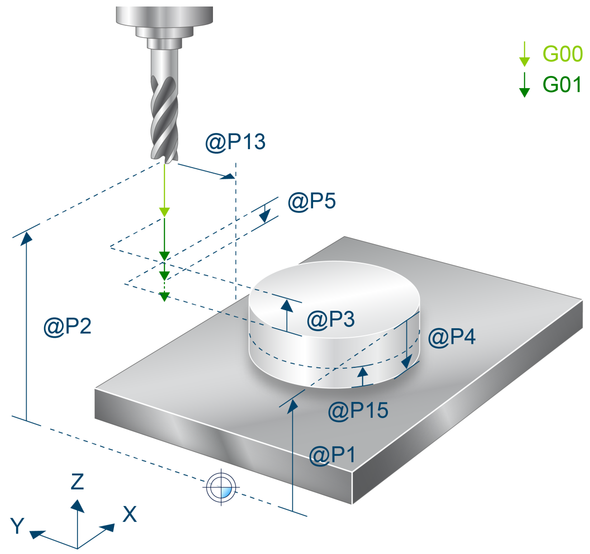

Fig. 34: 3D view - circular spigot

4.3.3 Syntax

```
L CYCLE [ NAME = SysMillCircularSpigot.ecy @P.. = .. ]
```

4.3.4 Programming example

Programing Example

Mill circular spigot

```
T8 D8                                ( Tool data )
M6                                   ( Tool change )

G90 G54 S6000 M03 F1500          ( Technology data )

#VAR
  ; input parameters:
  V.L.SurfacePosition = 0            ( Z position of workpiece surface )
  V.L.RetractionPlane = 50           ( Z position of retraction plane )
  V.L.SafetyClearance = 2            ( safety clearance in Z )
  V.L.DepthOfSpigot   = 10           ( depth of spigot )
  V.L.MaxIncrementZ   = V.L.DepthOfSpigot    ( maximum infeed in Z )
  V.L.MaxIncrementXY  = V.G.WZ_AKT.R         ( maximum infeed in XY )
  V.L.FeedRateXY      = 6000         ( machining feedrate in XY )
  V.L.FeedRateZ       = 4000         ( plunging feedrate )
  V.L.SpigotDiameter  = 10           ( diameter of spigot )
  V.L.BlankLength     = 50           ( length of the blank )
  V.L.BlankWidth      = 50           ( width of the blank )

#ENDVAR

G00 Z60
G00 X50 Y80

; polynomial contouring for smooth movements
#CONTOUR MODE [DEV, PATH_DEV = V.G.WZ_AKT.R / 1000]
G261

L CYCLE [NAME=SysMillCircularSpigot.ecy \
    @P1  = V.L.SurfacePosition          \
    @P2  = V.L.RetractionPlane          \
    @P3  = V.L.SafetyClearance          \
    @P4  = V.L.DepthOfSpigot            \
    @P5  = V.L.MaxIncrementZ            \
    @P6  = V.L.MaxIncrementXY           \
    @P20 = V.L.FeedRateXY               \
    @P21 = V.L.FeedRateZ                \
    @P68 = V.L.SpigotDiameter           \
    @P69 = V.L.BlankLength              \
    @P70 = V.L.BlankWidth               \
    ]

G260
M05
M30
```

4.4 SysMillRectangularSpigot - Mill rectangular spigot

This cycle is used to mill a rectangular spigot.

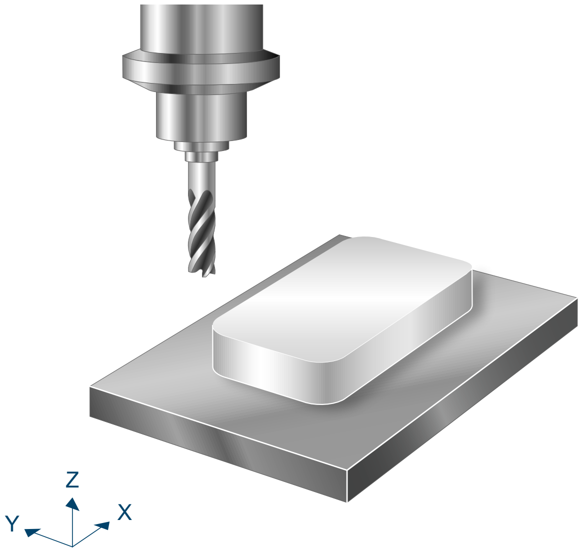

Fig. 35: Rectangular spigot

4.4.1 Process

To ensure that the cycle is executed successfully, the following requirements must be met.

- The tool radius is defined by V.G.WZ_AKT.R
- Tool length compensation is active.
- The spindle speed must be selected before the cycle is started.
- The start point in the XY plane (see below) can be reached without collision starting from the current position at the height of the retraction plane.
- A maximum of 3 systems may be active or defined when the cycle is called to ensure that sufficient machining coordinate systems (#CS) are available for the cycle.

Within the cycle, the tool is fed incrementally along the Z axis. The starting point here is a start point which has a safety clearance to the blank set by @P13 in the XY plane and is vertically parallel to the Y axis above the spigot centre. The blank is repeatedly approached in a circular movement for each machining height. With a clockwise machining direction (@P30=0), approach is counter-clockwise and departure is clockwise. With a counter-clockwise machining direction (@P30=1), approach is clockwise and departure is counter-clockwise.

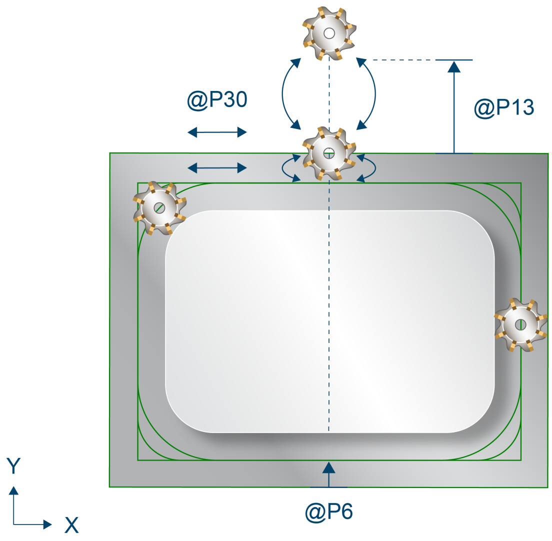

Fig. 36: Machining strategy for rectangular spigot

Roughing process

At the start, the start point in the XY plane is approached at the height of the retraction plane. The first feed is to the safety plane at rapid traverse along the Z axis followed by a feed to the machining height at the feed rate defined by @P21. Starting at this start point, an iterative feed takes place at the new machining height until the spigot depth @P4 is reached with the addition of the finishing allowance at the base (@P15). The maximum infeed per pass is the maximum infeed in Z at the maximum value defined in @P5.

The blank is first milled linearly for every machining height, i.e. without considering the corner radius (@P65). The tool then approaches closer to the centre of the spigot in a circular movement each circumferential pass without exceeding the maximum feed in XY (@P6). Linear removal continues until the linear outer dimensions of the spigot (@P72 and @P73) with the addition of the finishing allowance at the edge (@P16) are removed.

When linear machining is completed, the corner radius is milled (@P65). Corner removal continues every circumferential pass around the spigot until the correct radius (corner radius + finishing allowance in XY) is reached.

To avoid jerky movements in the milling path and thus relieve the machine tool, it may be useful to activate polynomial smoothing during roughing. This also leads to an accelerated execution of the milling cycle. The parameterisation of polynomial smoothing should be made dependent on the selected finishing allowance in order to avoid damage to the outer contour. To avoid residue, the maximum infeed in XY must then be reduced.

The milling cycle with polynomial smoothing can be called as follows:

```
#CONTOUR MODE [DEV, PATH_DEV = V.G.WZ_AKT.R / 10 ]   ( Parameterisation )
G261                                 ( Activation of polynomial contouring )
L CYCLE [NAME = SysMill... ]                                    ( Roughing )
G260                               ( Deactivation of polynomial contouring )
L CYCLE [NAME = SysMill... ]                                   ( Finishing )
M30
```

Finishing process

The start point mentioned for roughing is also approached as the starting position for finishing. From there, an interative feed takes place up to the new machining height until the spigot depth @P4 is reached. The maximum infeed per pass is the maximum infeed in Z at the maximum value defined in @P5.

A distinction is made between two cases for the milling operation at each machining height:

- If no finishing allowance at the base needs to be removed at the current machining height, only the finishing allowance around the spigot is removed. As for roughing, linear removal first takes place (as required) before the correct corner radius is removed.
- If the current machining height also involves a removal of the finishing allowance at the base, the complete blank must be again machined at this height until the correct spigot dimensions are reached. Here too, removal is first linear before the corner radius is removed.

4.4.2 Parameterisation

Required input parameters

Input parameters	Description
@P1	Reference plane (absolute)
@P2	Retraction plane (absolute)
@P3	Safety clearance in Z (relative to reference plane/machining height)
@P4	Depth (relative to reference plane)
@P5	Maximum infeed in Z
@P6	Maximum infeed in XY (maximum 2*V.G.WZ_AKT.R)
@P69	Length of blank
@P70	Width of blank
@P72	Length of spigot
@P73	Width of spigot

Optional input parameters

Input parameters	Description
@P13	Safety clearance to blank in the XY plane (relative) Default value is the tool radius
@P15	Finishing allowance at base Default value = 0
@P16	Finishing allowance at edge Default value = 0
@P20	Feed rate for milling operation (specified in mm/min, inch/min)
@P21	Feed rate in Z (values in mm/min, inch/min)
@P22	Feed rate for finishing for milling in the XY plane (specified in mm/min, inch/min) The default value is the feed rate for roughing
@P23	Feed rate for finishing in Z (specified in mm/min, inch/min) The default value is the feed rate for finishing for milling in the XY plane
@P25	Spindle speed for finishing (specified in rpm)
@P30	Machining direction "ClimbMilling" or 0: Down milling (default) "UpCutMilling" or 1: Up milling

@P31	Machining mode: "Roughing" or 1: Roughing "Finishing" or 2: Finishing "Roughing+Finishing" or 3: Roughing and finishing (default value)
@P40	Position of centre in X Default value is the current position
@P41	Position of centre in Y Default value is the current position
@P42	Angle of inclination of pocket (in degrees) Default value = 0
@P65	Corner radius of spigot Default value = 0

The function of the parameters acting in the XY plane results from the graphic below:

It is recommended to use the Syntax check.to verify whether the input parameters are correctly assigned.

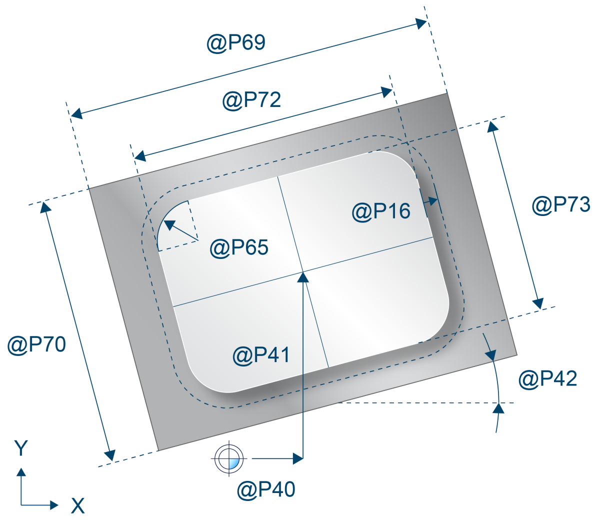

Fig. 37: Top view - rectangular spigot

Parameters which mainly refer to the Z axis are displayed in the graphic below:

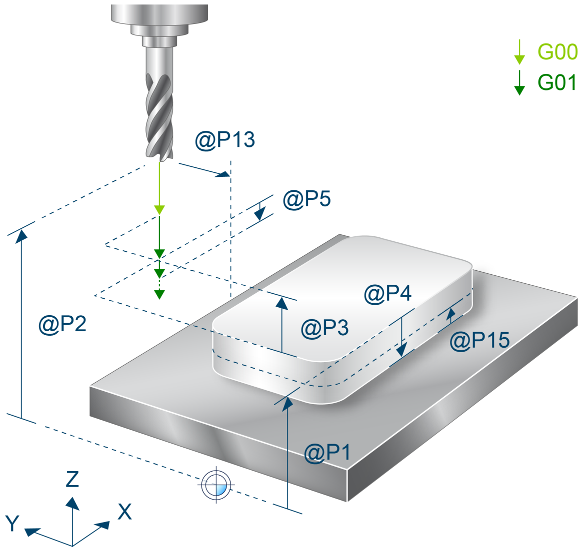

Fig. 38: 3D view - rectangular spigot

4.4.3 Syntax

L CYCLE [NAME = SysMillRectangularSpigot.ecy @P.. = ..]

4.4.4 Programming example

Programing Example

Mill rectangular spigot

```
T8 D8                               ( Tool data )
M6                                  ( Tool change )


G90 G54 S6000 M03 F2000             ( Technology data )


#VAR
  ; input parameters:
  V.L.SurfacePosition   = 0         ( Z position of workpiece surface )
  V.L.RetractionPlane   = 50        ( Z position of retraction plane )
  V.L.SafetyClearance   = 2         ( safety clearance in Z )
  V.L.DepthOfSpigot     = 4         ( depth of spigot )
  V.L.MaxIncrementZ     = V.L.DepthOfSpigot  ( maximum infeed in Z )
  V.L.MaxIncrementXY    = V.G.WZ_AKT.R ( maximum infeed in XY )
  V.L.FeedRateXY        = 6000      ( machining feedrate in XY )
  V.L.FeedRateZ         = 4000      ( plunging feedrate )
  V.L.BlankLength       = 50        ( length of the blank )
  V.L.BlankWidth        = 50        ( width of the blank )
  V.L.SpigotLength      = 35        ( length of the spigot )
  V.L.SpigotWidth       = 15        ( width of the spigot )
  #ENDVAR


G00 Z60
G00 X50 Y25                         ( Positioning to the starting point )

; polynomial contouring for smooth movements
#CONTOUR MODE [DEV, PATH_DEV = V.G.WZ_AKT.R / 100]
G261

L CYCLE [NAME=SysMillRectangularSpigot.ecy  \
  @P1  = V.L.SurfacePosition                \
  @P2  = V.L.RetractionPlane                \
  @P3  = V.L.SafetyClearance                \
  @P4  = V.L.DepthOfSpigot                  \
  @P5  = V.L.MaxIncrementZ                  \
  @P6  = V.L.MaxIncrementXY                 \
  @P20 = V.L.FeedRateXY                     \
  @P21 = V.L.FeedRateZ                      \
  @P69 = V.L.BlankLength                    \
  @P70 = V.L.BlankWidth                     \
  @P72 = V.L.SpigotLength                   \
  @P73 = V.L.SpigotWidth                    \
  ]

G260
M05
M30
```

4.5 SysMillMultiEdge – Mill multi-edge

This cycle is for milling a regular polygon, i.e. an object with equal sides and equal inner angles

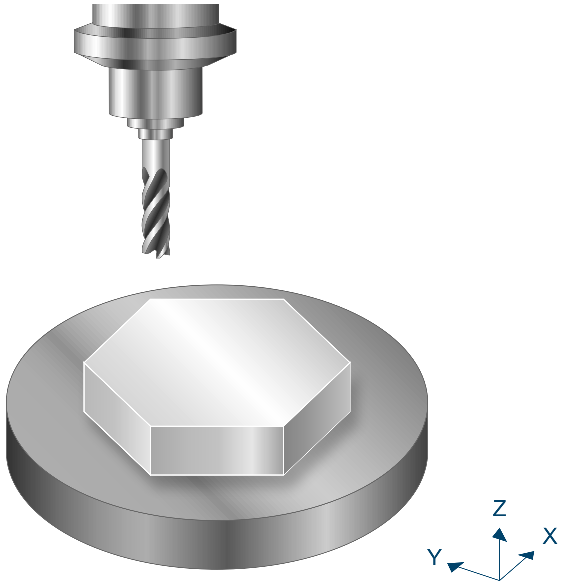

Fig. 39: Mill multi-edge

The shapes depicted below can be milled with this cycle from a cylindrical blank. Shapes with more than six corners are also possible.

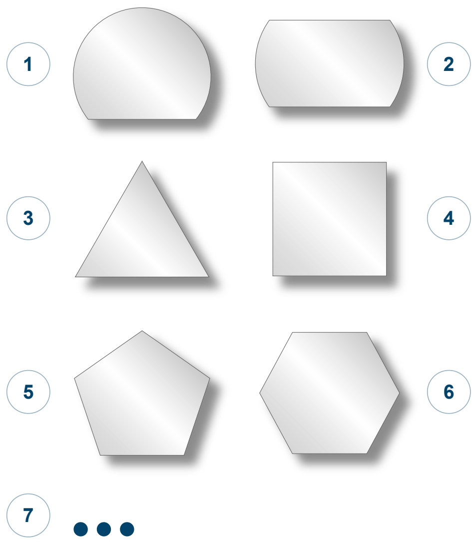

Fig. 40: Overview of multi-edge shapes

4.5.1 Process

To ensure that the cycle is executed successfully, the following requirements must be met.

- The tool radius is defined by V.G.WZ_AKT.R
- Tool length compensation is active.
- The spindle speed must be selected before the cycle is started.
- The blank can be circled at machining height without collision (see below).
- A maximum of 3 systems may be active or defined when the cycle is called to ensure that sufficient machining coordinate systems (#CS) are available for the cycle.

Within the cycle, the tool is fed incrementally along the Z axis. The starting point here is a start point which has a safety clearance to the blank set by @P13 in the XY plane and is vertically parallel to the Y axis under the spigot centre. The blank is approached in a circular or linear movement along the outer edge for each machining height. When the machining direction is clockwise (@P30=0 -> down milling), the approach is counter-clockwise and departure is clockwise. When the machining direction is counter-clockwise (@P30=1 -> up milling), the approach is clockwise and departure is counter-clockwise.

The milling operation has a helical shape for a multi-edge which has more than two edges. The milling operation has a linear shape for a multi-edge with one or two edges. The blank is continuously circled in the direction set in @P30 during the milling operation.

Attention

Collision detection by minimum clearance to blank

To permit a circular approach to the blank, it must be possible to circle the blank safely at machining height at a clearance D = [@P13 + 2 * tool radius]. The green dashed line in the graphic below describes the area in which no collision may occur.

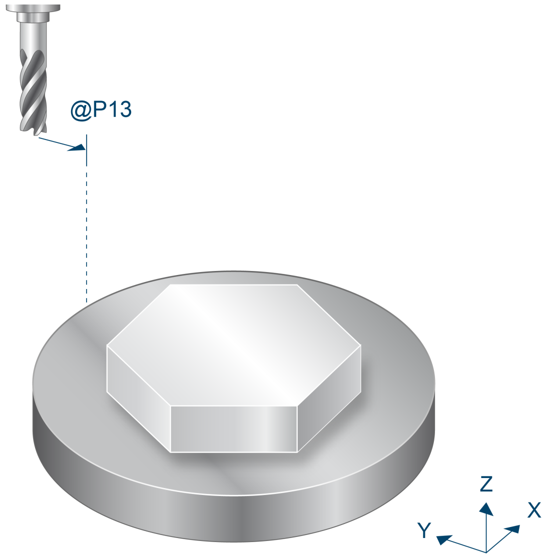

Fig. 41: Minimum clearance - multi-edge

Roughing process

At the start, the start point in the XY plane is approached at the height of the retraction plane. The infeed is first to the safety plane at rapid traverse along the Z axis followed by an infeed to the machining height at the feedrate defined by @P21. At this start point, an interative feed takes place up to the new machining height until the spigot depth @P4 is reached with the addition of the finishing allowance at the base (@P15). The maximum infeed per pass is the maximum infeed in Z at the maximum value defined in @P5.

For each machining height, the tool then approaches closer to the centre of the spigot in a circular movement each circumferential pass without exceeding the maximum feed in XY (@P6). Linear removal continues until the linear outer dimensions of the multi-edge with the addition of the finishing allowance at the edge (@P16) are removed.

When linear machining is completed, the corner radius (@P44) or the chamfer (@P66) is milled. Corner removal continues every circumferential pass around the spigot until the correct radius is reached.

To avoid jerky movements in the milling path and thus relieve the machine tool, it may be useful to activate polynomial contouring during roughing. This also leads to an accelerated execution of the milling cycle. The parameterisation of polynomial smoothing should be made dependent on the selected finishing allowance in order to avoid damage to the outer contour. To avoid residue, the maximum infeed in XY must then be reduced.

The milling cycle with polynomial smoothing can be called as follows:

```
#CONTOUR MODE [DEV, PATH_DEV = V.G.WZ_AKT.R / 10 ]  ( Parameterisation )
G261                                  ( Activation of polynomial contouring )
L CYCLE [NAME = SysMill... ]                                      ( Roughing )
G260                                ( Deactivation of polynomial contouring )
L CYCLE [NAME = SysMill... ]                                     ( Finishing )
M30
```

Finishing process

The start point mentioned for roughing is also approached as the starting position for finishing. From there, an interative feed takes place up to the new machining height until the spigot depth @P4 is reached. The maximum infeed per pass is the maximum infeed in Z at the maximum value defined in @P5.

A distinction is made between two cases for the milling operation at each machining height:

- If no finishing allowance at the base needs to be removed at the current machining height, only the finishing allowance around the spigot is removed.
- If the current machining height also involves a removal of the finishing allowance at the base, the complete blank must be again machined at this height until the correct spigot height is reached.

4.5.2 Parameterisation

Required input parameters

Input parameters	Description
@P1	Reference plane (absolute)
@P2	Retraction plane (absolute)
@P3	Safety clearance in Z (relative to reference plane/machining height)
@P4	Depth (relative to reference plane)
@P5	Maximum infeed Z (unsigned)
@P6	Maximum infeed in XY (maximum 2*V.G.WZ_AKT.R)
@P44	Number of edges (corners)
@P71	Radius of round blank
@P72	Edge length

Optional input parameters

Input parameters	Description
@P13	Safety clearance to blank in the XY plane (relative) Default value is the tool radius

@P15	Finishing allowance at base Default value = 0
@P16	Finishing allowance at edge Default value = 0
@P20	Feed rate for milling operation (specified in mm/min, inch/min)
@P21	Feed rate for infeed in Z (values in mm/min, inch/min)
@P22	Feed rate for finishing for milling in the XY plane (specified in mm/min, inch/min) The default value is the feed rate for roughing
@P23	Feed rate for finishing for infeed in Z (specified in mm/min, inch/min) The default value is the feed rate for finishing for milling in the XY plane
@P30	Machining direction "ClimbMilling" or 0: Down milling (default) "UpCutMilling" or 1: Up milling
@P31	Machining mode: "Roughing" or 1: Roughing "Finishing" or 2: Finishing "Roughing+Finishing" or 3: Roughing and finishing (default value)
@P40	Position of centre in X Default value is the current position
@P41	Position of centre in Y Default value is the current position
@P42	Angle of inclination of spigot Default value = 0
@P65	Corner radius of spigot Default = 0
@P66	Chamfer length of spigot (Alternative to corner radius) default value = 0
@P73	Width across flats (alternative to corner length)

The function of the parameters acting in the XY plane results from the graphic below:

It is recommended to use the Syntax check.to verify whether the input parameters are correctly assigned.

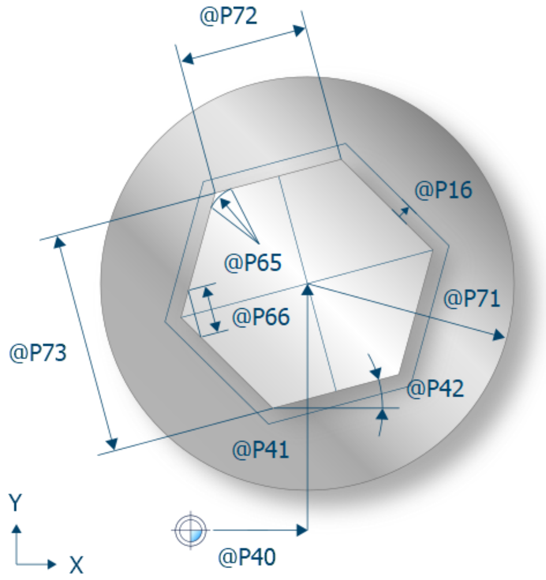

Fig. 42: Top view - multi-edge

The majority of parameters which mainly refer to the Z axis are displayed in the graphic below:

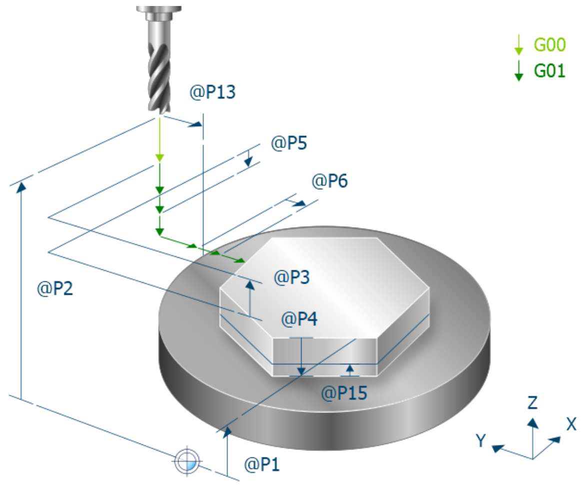

Fig. 43: 3D view - multi-edge

4.5.3 Syntax

L CYCLE [NAME = SysMillMultiEdge.ecy @P.. = ..]

4.5.4 Programming example

Programing Example

Multi-edge milling

```
T8 D8                        ( Tool data )
M6                           ( Tool change )


G90 G54 S6000 M03 F2000      ( Technology data )

#VAR
  ; input parameters:
  V.L.SurfacePosition = 0   ( Z position of workpiece surface )
  V.L.RetractionPlane = 50  ( Z position of retraction plane )
  V.L.SafetyClearance = 2   ( relative value of safety clearance in Z )
  V.L.Depth           = 5    ( depth of multiedge )
  V.L.MaxIncrementZ   = V.L.Depth      ( maximum infeed in Z )
  V.L.MaxIncrementXY  = V.G.WZ_AKT.R   ( maximum infeed in XY )
  V.L.FeedRateXY      = 6000 ( machining feedrate in XY )
  V.L.FeedRateZ       = 4000 ( plunging feedrate )
  V.L.NumberOfEdges   = 6    ( number of edges )
  V.L.BlankRadius     = 25   ( radius of the blank )
  V.L.EdgeLength      = 10   ( length of the edges )
#ENDVAR

G00 Z60                      ( Go to Z start position )
G00 X25 Y25                  ( Go to centre of the multi edge )

; polynomial contouring for smooth movements
#CONTOUR MODE [DEV, PATH_DEV = V.G.WZ_AKT.R / 100]
G261

L CYCLE [NAME=SysMillMultiEdge.ecy  \
    @P1  = V.L.SurfacePosition      \
    @P2  = V.L.RetractionPlane      \
    @P3  = V.L.SafetyClearance      \
    @P4  = V.L.Depth                \
    @P5  = V.L.MaxIncrementZ        \
    @P6  = V.L.MaxIncrementXY       \
    @P20 = V.L.FeedRateXY           \
    @P21 = V.L.FeedRateZ            \
    @P31 = "Roughing"               \
    @P44 = V.L.NumberOfEdges        \
    @P71 = V.L.BlankRadius          \
    @P72 = V.L.EdgeLength           \
    ]

G260

M05
M30
```

4.6 SysMillCircularPocket - Mill circular pocket

This cycle is used to mill a circular pocket.

In this cycle, the tool is fed incrementally along the Z axis and the pocket is also milled from the inside out along a circular or helical path plane-by-plane for each machining height. To mill the circular pocket, the tool is always lowered over the centre of the circular pocket. If the milling cutter does not cut above centre, the circular pocket can be predrilled in the centre before the cycle.

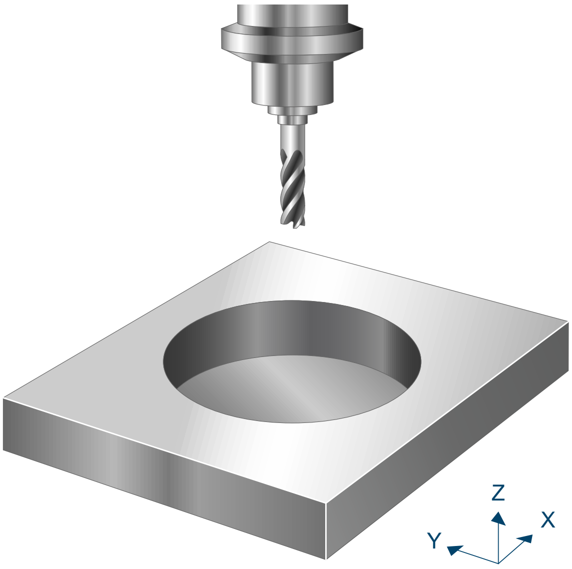

Fig. 44: Mill circular pocket

4.6.1 Process

To ensure that the cycle is executed successfully, the following requirements must be met.

- The tool radius is defined by V.G.WZ_AKT.R
- Tool length compensation is active.
- The spindle speed must be selected before the cycle is started.
- The pocket centre can be reached without collision starting from the current position at the height of the retraction plane.
- A maximum of 3 systems may be active or defined when the cycle is called to ensure that sufficient machining coordinate systems (#CS) are available for the cycle.

Within the cycle, the tool is first fed incrementally along the Z axis. The starting point is a start point located above the pocket. The pocket is then milled from the inside out plane-by-plane and for every machining height in either a circular or helical movement. When the pocket is fully milled at the current machining height, the tool is retracted to the safety clearance in XY (@P13, relative) and Z (@P3, relative) at feedrate in a helical movement in a semicircle and repositioned back at the centre (XY plane) at rapid traverse. It is then positioned at feedrate at the next machining height.

Roughing process

At the start, the start point in the XY plane is approached at the height of the retraction plane for roughing. The infeed is first to the safety plane at rapid traverse along the Z axis followed by an infeed to the machining height at the feedrate defined by @P20. At this start point, an interative feed takes place up to the new machining height until the pocket depth @P4 is reached with the addition of the finishing allowance at the base (@P15). The maximum infeed per pass is the maximum infeed in Z at the maximum value defined in @P5.

At every machining height, the circular pocket is again milled from the centre outwards in a helical (@P34=1) or circular movement at infeed in a semicircle (@P34=2). The radius is obtained from the pocket radius (@P68) minus the finishing allowance at the edge (@P16). The clearance of the helical or circular path between each circumferential pass never exceeds the value of the maximum infeed in XY defined in @P6.

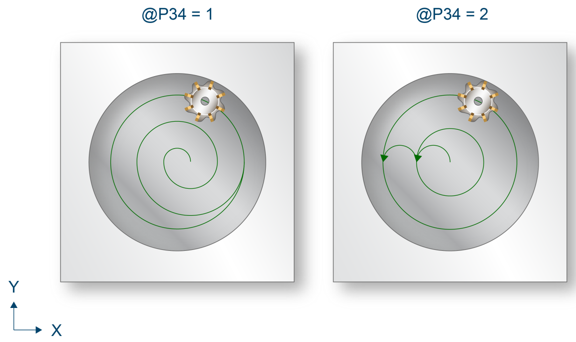

Fig. 45: Infeed in XY plane

To avoid jerky movements in the milling path and thus relieve the machine tool, it may be useful to activate polynomial smoothing during roughing. This also leads to an accelerated execution of the milling cycle. The parameterisation of polynomial smoothing should be made dependent on the selected finishing allowance in order to avoid damage to the outer contour. To avoid residue, the maximum infeed in XY must then be reduced.

The milling cycle with polynomial smoothing can be called as follows:

```
#CONTOUR MODE [DEV, PATH_DEV = V.G.WZ_AKT.R / 10 ] ( Parameterisation )
G261                            ( Activation of polynomial contouring )
L CYCLE [NAME = SysMill... ]                                ( Roughing )
G260                          ( Deactivation of polynomial contouring )
L CYCLE [NAME = SysMill... ]                               ( Finishing )
M30
```

Finishing process

The start point mentioned for roughing is also approached as the starting position for finishing. From there, an interative feed takes place up to the new machining height until the spigot depth @P4 is reached. The maximum infeed per pass is the maximum infeed in Z at the maximum value defined in @P5.

A distinction is made between two cases for the milling operation at each machining height:

If no finishing allowance at the base needs to be removed at the current machining height, only the finishing allowance at the edge of the pocket is removed. If the current machining height also involves a removal of the finishing allowance at the base, the complete circular pocket must be again machined at this height until the correct dimensions are reached.

4.6.2 Parameterisation

Required input parameters

Input parameters	Description
@P1	Reference plane (absolute)
@P2	Retraction plane (absolute)
@P3	Safety clearance in Z (relative to reference plane/machining height)
@P4	Depth (relative to reference plane)

@P5	Maximum infeed Z (unsigned)
@P6	Maximum infeed in XY (Maximum 2*V.G.WZ_AKT.R)
@P68	Pocket diameter

Optional input parameters

Input parameters	**Description**
@P7	Pitch of plunge helix with helical infeed in Z Default value = 10% of tool radius
@P8	Radius of plunge helix with helical infeed in Z Default value = 10% of tool radius
@P13	Safety clearance in the XY plane (relative to the pocket edge) Default value is 1/4 of the tool radius
@P15	Finishing allowance at base Default value = 0
@P16	Finishing allowance at edge Default value = 0
@P20	Feed rate for roughing for milling in the XY plane (specified in mm/min, inch/min)
@P21	Feed rate for roughing for infeed in Z (specified in mm/min, inch/min)
@P22	Feed rate for finishing for milling in the XY plane (specified in mm/min, inch/min) The default value is the feed rate for roughing
@P23	Feed rate for finishing for infeed in Z (specified in mm/min, inch/min) The default value is the feed rate for finishing for milling in the XY plane
@P30	Machining direction "ClimbMilling" or 0: Down milling (default) "UpCutMilling" or 1: Up milling
@P31	Machining mode: "Roughing" or 1: Roughing "Finishing" or 2: Finishing "Roughing+Finishing" or 3: Roughing and finishing (default value)
@P32	Feed mode in Z 1: Vertical 2: Helical (default value)
@P34	Machining path in XY 1: Helical (default value) 2: Circular with infeed in semicircle
@P40	Position of centre in X Default value is the current position
@P41	Position of centre in Y

	Default value is the current position

The function of the parameters acting in the XY plane is derived largely from the graphic below:

It is recommended to use the Syntax check.to verify whether the input parameters are correctly assigned.

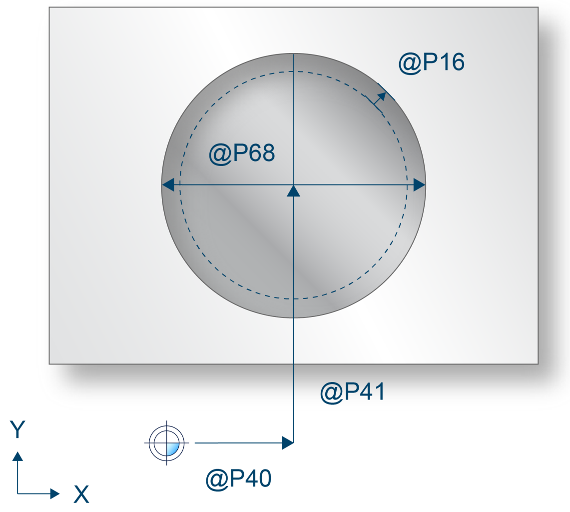

Fig. 46: Top view - circular pocket

Parameters which mainly refer to the Z axis are displayed in the graphic below:

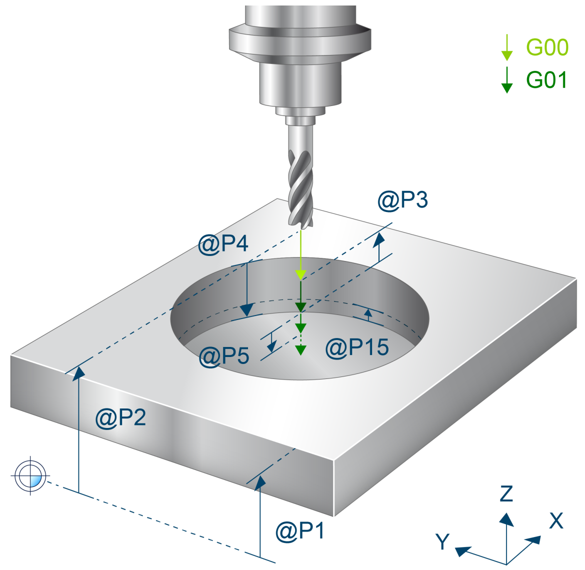

Fig. 47: 3D view - circular pocket

4.6.3 Syntax

L CYCLE [NAME = SysMillCircularPocket.ecy @P.. = ..]

4.6.4 Programming example

Programing Example

Mill circular pocket

```
T8 D8                       ( Tool data )
M6                          ( Tool change )

G90 G54 S6000 M03 F1500     ( Technology data )

#VAR
  ; input parameters:
  V.L.SurfacePosition = 0   ( Z position of workpiece surface )
  V.L.RetractionPlane = 50  ( Z position of retraction plane )
  V.L.SafetyClearance = 2   ( relative value of safety clearance in Z )
  V.L.DepthOfPocket   = 20  ( depth of pocket )
  V.L.MaxIncrementZ   = 3   ( maximum infeed in Z )
  V.L.MaxIncrementXY  = V.G.WZ_AKT.R  ( maximum infeed in XY )
  V.L.FeedRateXY      = 600 ( machining feedrate in XY )
  V.L.FeedRateZ       = 400 ( plunging feedrate )
  V.L.PocketDiameter  = 20  ( diameter of pocket )
#ENDVAR

G00 Z60
G00 X50 Y50

; polynomial contouring for smooth movements
#CONTOUR MODE [DEV, PATH_DEV = V.G.WZ_AKT.R / 100]
G261

L CYCLE [NAME=SysMillCircularPocket.ecy \
    @P1  = V.L.SurfacePosition          \
    @P2  = V.L.RetractionPlane          \
    @P3  = V.L.SafetyClearance          \
    @P4  = V.L.DepthOfPocket            \
    @P5  = V.L.MaxIncrementZ            \
    @P6  = V.L.MaxIncrementXY           \
    @P20 = V.L.FeedRateXY               \
    @P21 = V.L.FeedRateZ                \
    @P31 = "Roughing"                   \
    @P68 = V.L.PocketDiameter           \
    ]

G260
M05
M30
```

4.7 SysMillRectangularPocket – Mill rectangular pocket

This cycle is used to mill a rectangular pocket.

In this cycle, the tool is fed incrementally along the Z axis and the pocket is also milled from the inside out plane-by-plane and incremental or in a helical trajectory for each machining height. To mill the rectangular pockets, the tool is always lowered over the centre of the rectangular pocket. If the milling cutter does not cut above centre, the rectangular pocket can be predrilled in the centre before the cycle.

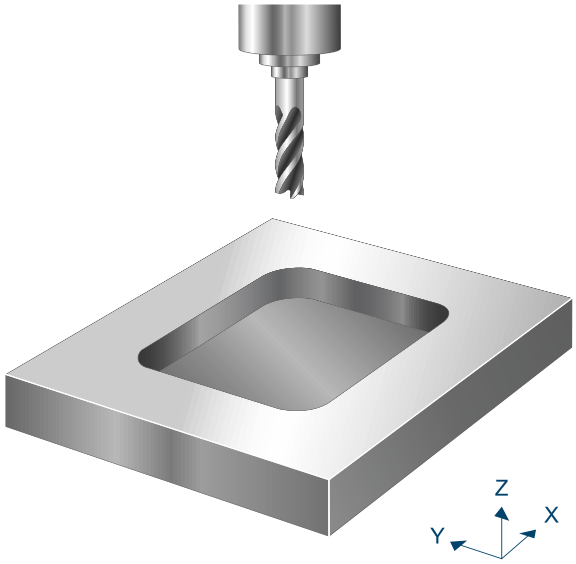

Fig. 48: Rectangular pocket

4.7.1 Process

To ensure that the cycle is executed successfully, the following requirements must be met.

- The tool radius is defined by V.G.WZ_AKT.R
- Tool length compensation is active.
- The spindle speed must be selected before the cycle is started.
- The pocket centre can be reached without collision starting from the current position at the height of the retraction plane.
- A maximum of 3 systems may be active or defined when the cycle is called to ensure that sufficient machining coordinate systems (#CS) are available for the cycle.

Within the cycle, the tool is first fed incrementally along the Z axis. The starting point is a start point located above the pocket. The pocket is then again milled plane-by-plane from the inside out for every machining height either incremental or in a helical movement. When the pocket is fully milled at the current machining height, the tool is retracted to the safety clearance in XY (@P13, relative) and Z (@P3, relative) at feedrate in a helical movement in a semicircle and repositioned back at the centre (XY plane) at rapid traverse. It is then positioned at feedrate at the next machining height.

Roughing process

At the start, the start point in the XY plane is approached at the height of the retraction plane for roughing. The infeed is first to the safety plane at rapid traverse along the Z axis followed by an infeed to the machining height at the feedrate. At this start point, an interative feed takes place up to the new machining height at the feedrate defined in @P21 until the pocket depth @P4 is reached with the addition of the finishing allowance at the base (@P15). The maximum feed in Z per pass is the maximum feed in Z at the maximum value defined in @P5.

At every machining height, the pocket is again milled from the centre outwards in a helical movement (@P34=1) or incrementally at infeed in a semicircle (@P34=2). The clearance of the paths between one pass around never exceeds the maximum value of the maximum feed in XY defined in @P5.

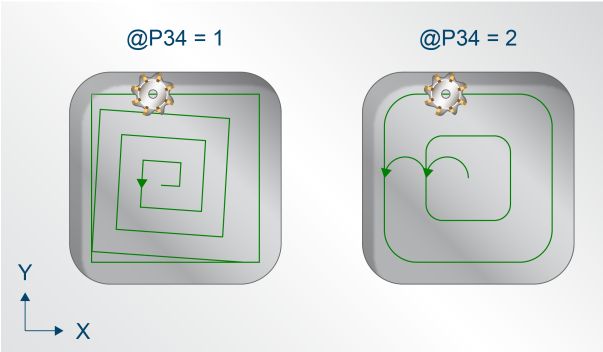

Fig. 49: Infeed in XY plane

To avoid jerky movements in the milling path and thus relieve the machine tool, it may be useful to activate polynomial contouring during roughing. This also leads to an accelerated execution of the milling cycle. The parameterisation of polynomial smoothing should be made dependent on the selected finishing allowance in order to avoid damage to the outer contour. To avoid residue, the maximum infeed in XY must then be reduced.

The milling cycle with polynomial smoothing can be called as follows:

```
#CONTOUR MODE [DEV, PATH_DEV = V.G.WZ_AKT.R / 10 ]  ( Parameterisation )
G261                                  ( Activation of polynomial contouring )
L CYCLE [NAME = SysMill... ]                                      ( Roughing )
G260                                ( Deactivation of polynomial contouring )
L CYCLE [NAME = SysMill... ]                                     ( Finishing )
M30
```

Finishing process

The start point mentioned for roughing is also approached as the starting position for finishing. From there, infeed is repeated up to the new machining height until the pocket depth @P4 is reached. The maximum infeed per pass is the maximum infeed in Z at the maximum value defined in @P65.

A distinction is made between two cases for the milling operation at each machining height:

- If no finishing allowance at the base needs to be removed at the current machining height, only the finishing allowance at the edge of the pocket is removed.
- If the current machining height also involves a removal of the finishing allowance at the base, the complete rectangular pocket must be again machined at this height until the correct dimensions are reached.

4.7.2 Parameterisation

Required input parameters

Input parameters	Description
@P1	Reference plane (absolute)
@P2	Retraction plane (absolute)
@P3	Safety clearance in Z (relative to reference plane/machining height)
@P4	Depth (relative to reference plane)

@P5	Maximum infeed in Z
@P6	Maximum infeed in XY (Maximum 2*V.G.WZ_AKT.R)
@P72	Length of the pocket in X
@P73	Width of the pocket in Y

Optional input parameters

Input parameters	Description
@P7	Pitch of plunge helix with helical infeed in Z Default value = 10% of tool radius
@P8	Radius of plunge helix with helical infeed in Z Default value = 10% of tool radius
@P13	Safety clearance in the XY plane (relative to the pocket edge) Default value is 1/8 of the pocket radius
@P15	Finishing allowance at base Default value = 0
@P16	Finishing allowance at edge Default value = 0
@P20	Feed rate for roughing for milling in the XY plane (specified in mm/min, inch/min)
@P21	Feed rate for roughing for infeed in Z (specified in mm/min, inch/min)
@P22	Feed rate for finishing for milling in the XY plane (specified in mm/min, inch/min) The default value is the feed rate for roughing
@P23	Feed rate for finishing for infeed in Z (specified in mm/min, inch/min) The default value is the feed rate for finishing for milling in the XY plane
@P30	Machining direction "ClimbMilling" or 0: Down milling (default) "UpCutMilling" or 1: Up milling
@P31	Machining mode: "Roughing" or 1: Roughing "Finishing" or 2: Finishing "Roughing+Finishing" or 3: Roughing and finishing (default value)
@P32	Feed mode in Z 1: Vertical 2: Helical (default value)
@P34	Infeed mode in XY 1: Helical 2: Incremental with infeed in semicircle (default value)
@P40	Position of centre in X Default value is the current position

@P41	Position of centre in Y Default value is the current position
@P42	Angle of inclination of pocket (in degrees) Default value = 0
@P65	Corner radius of the pocket Default value is the tool radius Only possible with incremental infeed in XY

The function of the parameters acting in the XY plane results from the graphic below:

It is recommended to use the Syntax check.to verify whether the input parameters are correctly assigned.

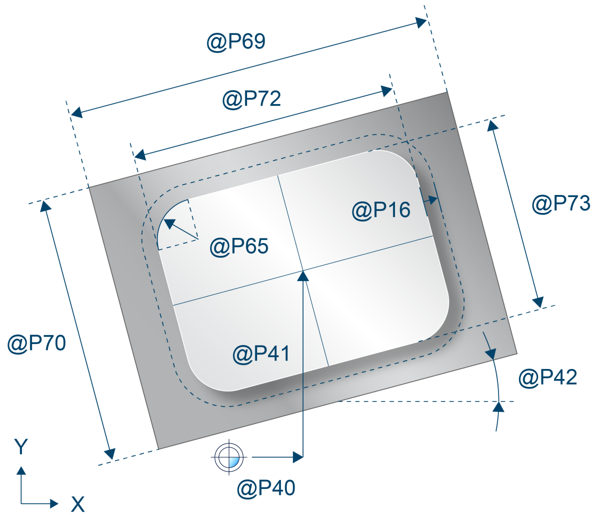

Fig. 50: Top view - rectangular pocket

Parameters which mainly refer to the Z axis are displayed in the graphic below:

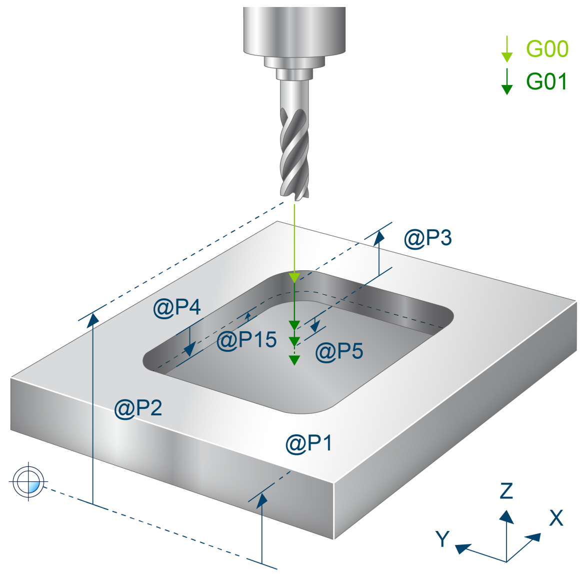

Fig. 51: 3D view - rectangular pocket

4.7.3 Syntax

L CYCLE [NAME = SysMillRectangularPocket.ecy @P.. = ..]

4.7.4 Programming example

Programing Example

Mill rectangular pocket

```
T8 D8                        ( Tool data )
M6                           ( Tool change )


G54 G90 S6000 M03 F5000      ( Technology data )

#VAR
  ; input parameters:
  V.L.SurfacePosition   = 0    ( Z position of workpiece surface )
  V.L.RetractionPlane   = 50   ( Z position of retraction plane )
  V.L.SafetyClearance   = 2    ( relative value of safety clearance in Z )
  V.L.DepthOfPocket     = 12   ( depth of pocket )
  V.L.MaxIncrementZ     = 4    ( maximum infeed in Z )
  V.L.MaxIncrementXY    = V.G.WZ_AKT.R ( maximum infeed in XY )
  V.L.FeedRateZ         = 1000         ( plunging feed rate )
  V.L.PocketWidth       = 20           ( width of the pocket )
  V.L.PocketLength      = 20   ( length of the pocket )
#ENDVAR


G00 Z60
G00 X50 Y50                  ( Positioning to the starting point )

; polynomial contouring for smooth movements
#CONTOUR MODE [DEV, PATH_DEV = V.G.WZ_AKT.R / 100]
G261

L CYCLE [NAME=SysMillRectangularPocket.ecy  \
    @P1  = V.L.SurfacePosition              \
    @P2  = V.L.RetractionPlane              \
    @P3  = V.L.SafetyClearance              \
    @P4  = V.L.DepthOfPocket                \
    @P5  = V.L.MaxIncrementZ                \
    @P6  = V.L.MaxIncrementXY               \
    @P21 = V.L.FeedRateZ                    \
    @P31 = "Roughing"                       \
    @P72 = V.L.PocketLength                 \
    @P73 = V.L.PocketWidth                  \
    ]

G260
M05
M30
```

4.8 SysMillLonghole cycle - Milling longholes arranged in a circle

This cycle is used to mill longholes arranged in a circle. As opposed to slots, long holes always have a width corresponding to the diameter of the milling cutter.

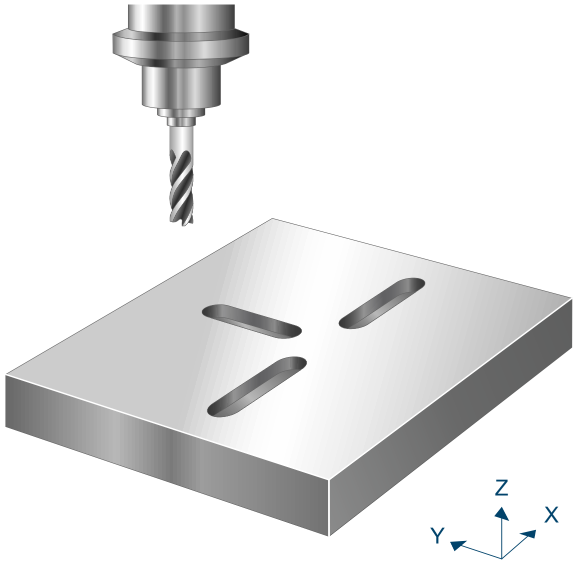

Fig. 52: Mill longholes arranged in a circle

4.8.1 Process

To ensure that the cycle is executed successfully, the following requirements must be met.

- The tool radius is defined by V.G.WZ_AKT.R
- Tool length compensation is active.
- The spindle speed must be selected before the cycle is started.
- Both ends of longholes are reachable without collision at the height of the retraction plane
- The tool cuts across centre or the workpiece was predrilled (both ends of the longholes)

During the cycle, the longholes are approached at rapid traverse at the height of the retraction plane. The feed is first to the safety plane at rapid traverse along the Z axis followed by a feed to the machining height at the feed rate. Starting at this start point, an iterative feed takes place up to the new machining height at the feed defined in @P21 and the longhole is machined until the longhole depth @P4 is reached. The iterative infeed in Z per pass never exceeds the value of the maximum infeed in Z defined in @P5.

If “oscillating” is selected as infeed mode in Z defined in @P32, iterative infeeds in Z are executed by a lowering approach to the other end of the longhole at the feedrate defined in @P20. To completely remove the machining height, the longhole is machined horizontally at the feedrate defined in @P20 after an inclined plunge.

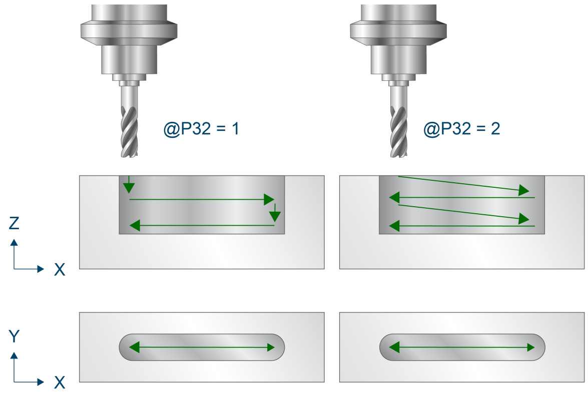

Fig. 53: Infeed in Z direction

4.8.2 Parameterisation

Required input parameters

Input parameters	Description
@P1	Reference plane (absolute)
@P2	Retraction plane (absolute)
@P3	Safety clearance in Z (relative to reference plane/machining height)
@P4	Depth of longhole (relative to reference plane)
@P5	Maximum infeed Z (unsigned)
@P44	Number of longholes
@P72	Length of longhole
@P74	Circle radius of centre point positions

Optional input parameters

Input parameters	Description
@P20	Feed rate for finishing for milling in the XY plane (values in mm/min, inch/min)
@P21	Feed rate for infeed in Z (values in mm/min, inch/min)
@P32	Feed mode in Z 1: Vertical 2: Oscillating (default)
@P40	Position of circle centre point in X Default value is the current position
@P41	Position of circle centre point in Y Default value is the current position
@P45	Starting angle of first slot (in degrees) Default value = 0
@P46	Incremental angle (in degrees) Default value = 360°/number of longholes

The function of the parameters acting in the XY plane is derived largely from the graphic below:

It is recommended to use the Syntax check.to verify whether the input parameters are correctly assigned.

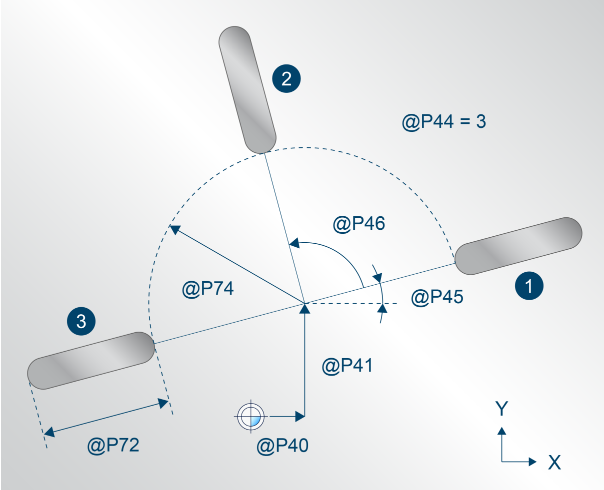

Fig. 54: Top view – longholes arranged in a circle

Parameters which mainly refer to the Z axis are displayed in the graphic below:

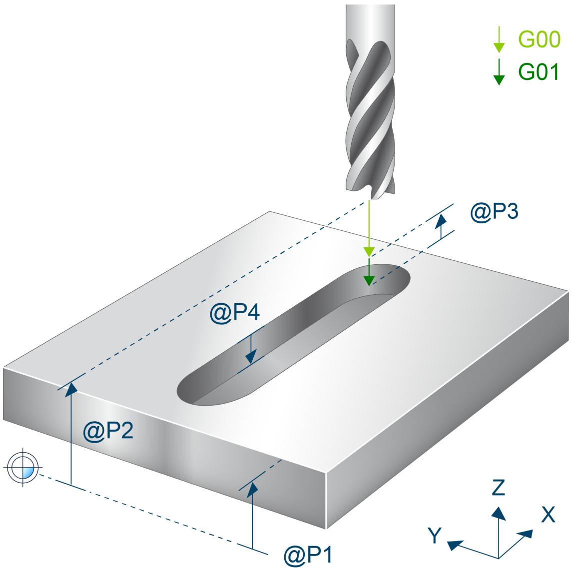

Fig. 55: 3D view – longholes arranged in a circle

4.8.3 Syntax

L CYCLE [NAME = SysMillLonghole.ecy @P.. = ..]

4.8.4 Programming example

Programing Example

Mill longholes arranged in a circle

```
T8 D8                          ( Tool data )
M6                             ( Tool change )
G90 G54 F200 S2000 M03         ( Technology data )
G00 Z100                       ( Go to Z start position )
G00 X50 Y50                    ( Go to the center of the slot )

#VAR
  ; input parameters:
  V.L.SurfacePosition   = 0    ( Z position of workpiece surface )
  V.L.RetractionPlane   = 40   ( Z position of retraction plane )
  V.L.SafetyClearance   = 10   ( relative value of safety clearance in Z )
  V.L.DepthOfHole       = 10   ( depth of hole )
  V.L.MaxIncrementZ     = 3    ( maximum infeed in Z )
  V.L.NumberOfHoles     = 4    ( number of holes )
  V.L.HoleLength        = 50   ( length of the hole )
  V.L.Radius            = 20   ( radius of the hole centres )
  ; optional parameters:
  V.L.FeedRateXY       = 1000  ( machining feed rate in XY )
  V.L.FeedRateZ        = 500   ( plunging feed rate )
  V.L.PlungingModeZ    = 1     ( plunging mode in Z )
#ENDVAR

; Oscillating infeed in Z
L CYCLE[NAME=SysMillLonghole.ecy  \
    @P1  = V.L.SurfacePosition    \
    @P2  = V.L.RetractionPlane    \
    @P3  = V.L.SafetyClearance    \
    @P4  = V.L.DepthOfHole        \
    @P5  = V.L.MaxIncrementZ      \
    @P20 = V.L.FeedRateXY         \
    @P21 = V.L.FeedRateZ          \
    @P44 = V.L.NumberOfHoles      \
    @P72 = V.L.HoleLength         \
    @P74 = V.L.Radius             \
    ]

; Vertical infeed in Z
L CYCLE[NAME=SysMillLonghole.ecy  \
     @P1  = V.L.SurfacePosition   \
     @P2  = V.L.RetractionPlane   \
     @P3  = V.L.SafetyClearance   \
     @P4  = V.L.DepthOfHole       \
     @P5  = V.L.MaxIncrementZ     \
     @P20 = V.L.FeedRateXY        \
     @P21 = V.L.FeedRateZ         \
     @P32 = V.L.PlungingModeZ     \
     @P44 = V.L.NumberOfHoles     \
     @P72 = V.L.HoleLength        \
     @P74 = V.L.Radius            \
    ]

M05
M30
```

4.9 SysMillSlot - Milling slots arranged in circle

This cycle is used to mill slots arranged in a circle.

In this cycle, the tool is fed incrementally along the Z axis per slot and the slot is also milled from the inside out plane-by-plane and incrementally for each machining height. To mill the slots, the tool is always lowered over the centre of the slot. If the milling cutter does not cut above centre, the slots can be predrilled in the centre before the cycle is executed.

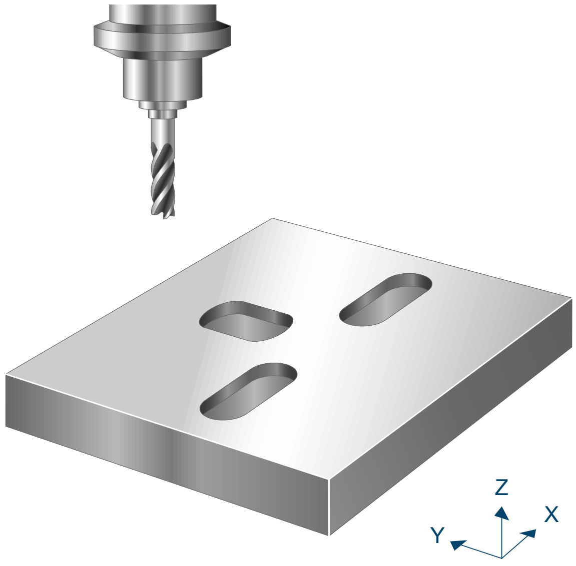

Fig. 56: Mill circumferential slots

4.9.1 Process

To ensure that the cycle is executed successfully, the following requirements must be met.

- The tool radius is defined by V.G.WZ_AKT.R
- Tool length compensation is active.
- The spindle speed must be selected before the cycle is started.
- The slot centres can be reached without collision starting from the current position at the height of the retraction plane.
- A maximum of 3 systems may be active or defined when the cycle is called to ensure that sufficient machining coordinate systems (#CS) are available for the cycle.

Within the cycle, the centre of each slot is approached and the tool is fed incrementally along the Z axis. The slot is then again milled plane-by-plane from the inside out for every machining height incrementally. When the slot is fully milled at the current machining height, the tool is retracted to the safety clearance in XY (@P13, relative) at the feedrate in a semicircle and repositioned back at the centre (XY plane) at rapid traverse. It is then positioned at feedrate at the next machining height.

Roughing process

At the start of roughing, the start point for every slot is approached via the centre of the slot in the XY plane at the height of the retraction plane. The infeed is first to the safety plane at rapid traverse along the Z axis followed by a vertical (@P32=1) or helical (@P32=2) or oscillating (@P32=3) infeed to the machining height at the feedrate.

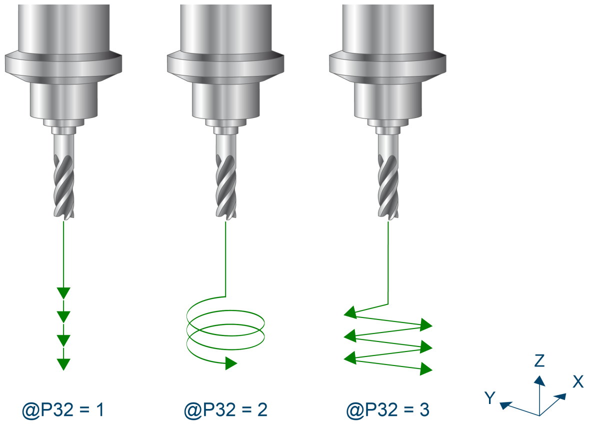

Fig. 57: Infeed variants

The beginning of the slot is started with the oscillating infeed. When the required infeed depth is reached, the milling cutter is positioned at the centre of the slot. At this start point, an iterative feed takes place up to the new machining height at the feedrate defined in @P21 until the slot depth @P4 is reached with the addition of the finishing allowance at the base (@P15). The maximum feed in Z per pass is the maximum feed in Z at the maximum value defined in @P5.

At every machining height, the slot is again milled from the centre outwards incrementally at infeed in a semicircle. The clearance of the paths between one pass around never exceeds the maximum value of the maximum feed in XY defined in @P6.

To avoid jerky movements in the milling path and thus relieve the machine tool, it may be useful to activate polynomial smoothing during roughing. This also leads to an accelerated execution of the milling cycle. The parameterisation of polynomial smoothing should be made dependent on the selected finishing allowance in order to avoid damage to the outer contour. To avoid residue, the maximum infeed in XY must then be reduced.

The milling cycle with polynomial smoothing can be called as follows:

```
#CONTOUR MODE [DEV, PATH_DEV = V.G.WZ_AKT.R / 10 ]  ( Parameterisation )
G261                                 ( Activation of polynomial contouring )
L CYCLE [NAME = SysMill... ]                                     ( Roughing )
G260                               ( Deactivation of polynomial contouring )
L CYCLE [NAME = SysMill... ]                                    ( Finishing )
M30
```

Finishing process

The start point mentioned for roughing is also approached as the starting position for finishing. From there, an interative feed takes place up to the new machining height until the slot depth @P5 is reached. The maximum infeed per pass is the maximum infeed in Z at the maximum value defined in @P5.

A distinction is made between two cases for the milling operation at each machining height:

- If no finishing allowance at the base needs to be removed at the current machining height, only the finishing allowance at the edge of the slot is removed. Infeed is vertical.
- If the current machining height also involves a removal of the finishing allowance at the base, the complete slot must be again machined at this height until the correct dimensions are reached. Infeed in Z is vertical (@P32=1) or helical (@P32=2) or oscillating (@P32=3).

4.9.2 Parameterisation

Required input parameters

Input parameters	Description

@P1	Reference plane (absolute)
@P2	Retraction plane (absolute)
@P3	Safety clearance in Z (relative to reference plane/machining height)
@P4	Depth of slot (relative to reference plane)
@P5	Maximum infeed in Z
@P6	Maximum infeed in XY plane (Maximum 2*V.G.WZ_AKT.R)
@P44	Number of slots
@P72	Length of slot
@P73	Width of slot
@P74	Circle radius of centre point positions

Optional input parameters

Input parameters	Description
@P7	Pitch of plunge helix with helical infeed in Z Default value = 10% of tool radius
@P8	Radius of plunge helix with helical infeed in Z Default value = 10% of tool radius
@P9	Plunge angle for oscillating infeed in Z Default value = 2 degrees
@P13	Safety clearance on retraction in XY plane (relative to milled edge) Default value is 1/8 of slot width
@P15	Finishing allowance at base Default value = 0
@P16	Finishing allowance at edge Default value = 0
@P20	Feed rate for roughing for milling in the XY plane (specified in mm/min, inch/min)
@P21	Feed rate for roughing for infeed in Z (specified in mm/min, inch/min)
@P22	Feed rate for finishing for milling in the XY plane (specified in mm/min, inch/min) The default value is the feed rate for roughing
@P23	Feed rate for finishing for infeed in Z (specified in mm/min, inch/min) The default value is the feed rate for finishing for milling in the XY plane
@P30	Machining direction "ClimbMilling" or 0: Down milling (default) "UpCutMilling" or 1: Up milling
@P31	Machining mode: "Roughing" or 1: Roughing "Finishing" or 2: Finishing

	"Roughing+Finishing" or 3: Roughing and finishing (default value)
@P32	Feed mode in Z 1: Vertical 2: Helical 3: Oscillating (default value)
@P40	Position of circle centre point in X (absolute) Default value is the current position
@P41	Position of circle centre point in Y (absolute) Default value is the current position
@P45	Starting angle of first slot (in degrees) Default value = 0
@P46	Incremental angle (in degrees) Default value = 360°/number of slots

The function of the parameters acting in the XY plane is derived largely from the graphic below:

It is recommended to use the Syntax check.to verify whether the input parameters are correctly assigned.

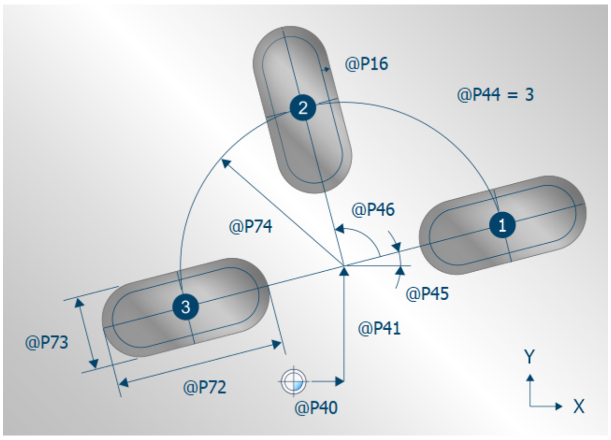

Fig. 58: Top view – slots arranged in a circle

Parameters which mainly refer to the Z axis are displayed in the graphic below:

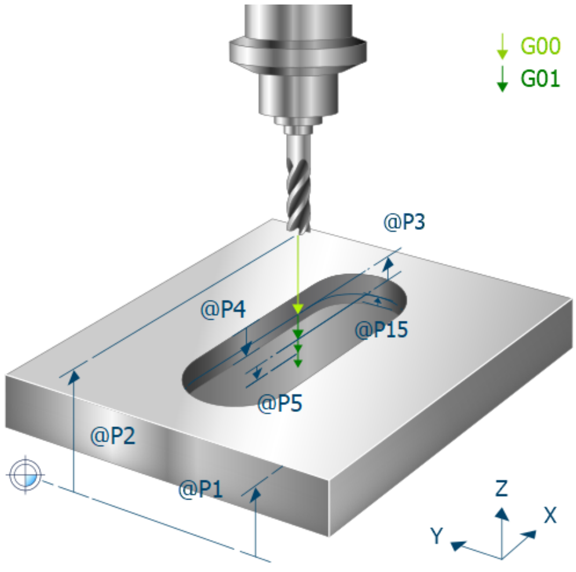

Fig. 59: 3D view – slots arranged in a circle

4.9.3 Syntax

L CYCLE [NAME = SysMillSlot.ecy @P.. = ..]

4.9.4 Programming example

Programing Example

Mill circumferential slots

```
T8 D8                          ( D5 Milling Tool data )
M6                             ( Tool change )

G90 G54 M03 S6000 F5000        ( Technology data )

G00 Z50                        ( Go to Z start position )
G00 X50 Y50                    ( Go to start position )

#VAR
  ; input parameters:
  V.L.SurfacePosition    = 0   ( Z position of workpiece surface )
  V.L.RetractionPlane   = 50  ( Z position of retraction plane )
  V.L.SafetyClearance    = 2   ( relative value of safety clearance in Z)
  V.L.DepthOfSlot        = 8   ( depth of slot )
  V.L.MaxIncrementZ      = 4   ( maximum infeed in Z )
  V.L.MaxIncrementXY     = V.G.WZ_AKT.R  ( maximum infeed in XY )
  V.L.FinishingOffsetZ   = 0.2 ( 11 finishing offset Z )
  V.L.FinishingOffsetXY  = 0.2 ( finishing offset XY )
  V.L.FeedRateZ          = 500 ( plunging feedrate )
  V.L.NumberOfSlots      = 4   ( number of slots )
  V.L.SlotLength         = 30  ( length of the slot )
  V.L.SlotWidth          = 12  ( width of the slot )
  V.L.Radius             = 25  ( radius of the centres )
#ENDVAR
```

```
; polynomial contouring for smooth movements
#CONTOUR MODE [DEV, PATH_DEV = 0.01]
G261

L CYCLE[NAME=SysMillSlot.ecy         \
    @P1  = V.L.SurfacePosition       \
    @P2  = V.L.RetractionPlane       \
    @P3  = V.L.SafetyClearance       \
    @P4  = V.L.DepthOfSlot           \
    @P5  = V.L.MaxIncrementZ         \
    @P6  = V.L.MaxIncrementXY        \
    @P15 = V.L.FinishingOffsetZ      \
    @P16 = V.L.FinishingOffsetXY     \
    @P21 = V.L.FeedRateZ             \
    @P31 = "Roughing"                \
    @P44 = V.L.NumberOfSlots         \
    @P72 = V.L.SlotLength            \
    @P73 = V.L.SlotWidth             \
    @P74 = V.L.Radius                \
    ]

G260
M05
M30
```

4.10 SysMillCircumferentialSlot cycle - Milling slots arranged in a circle

This cycle is used to mill circumferential slots.

In this cycle, the tool is fed incrementally along the Z axis per slot and the slot is also milled from the inside out plane-by-plane and incrementally for each machining height. To mill the slots, the tool is always lowered over the centre of the slot. If the milling cutter does not cut above centre, the slots can be predrilled in the centre before the cycle is executed.

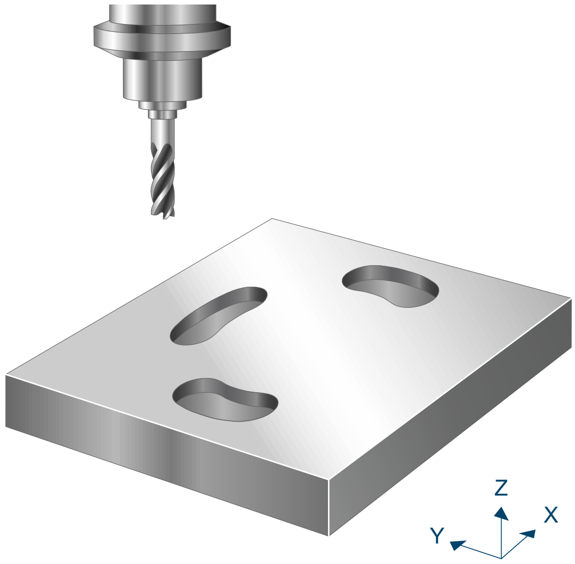

Fig. 60: Mill circumferential slots

4.10.1 Process

To ensure that the cycle is executed successfully, the following requirements must be met.

- The tool radius is defined by V.G.WZ_AKT.R
- Tool length compensation is active.
- The spindle speed must be selected before the cycle is started.
- The slot centres are reachable without collision at the height of the retraction plane
- A maximum of 3 systems may be active or defined when the cycle is called to ensure that sufficient machining coordinate systems (#CS) are available for the cycle.

Within the cycle, the centre of each slot is approached and the tool is fed incrementally along the Z axis. The slot is then again milled plane-by-plane from the inside out for every machining height incrementally. When the slot is fully milled at the current machining height, the tool is retracted to the safety clearance in XY (@P13, relative) and Z (@P3, relative) at feedrate in a helical movement in a semicircle and repositioned back at the centre (XY plane). It is then positioned at feedrate at the next machining height.

Roughing process

At the start of roughing, the starting point in the XY plane is approached for each slot at the height of the retraction plane. First, the infeed moves to the safety plane at rapid traverse along the Z axis followed by a vertical (@P32=1) or helical (@P32=2) or oscillating (@P32=3) infeed to the machining height at the feedrate.

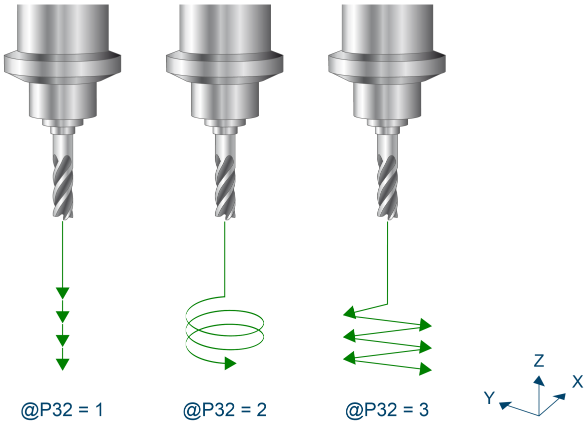

Fig. 61: Infeed variants

The beginning of the slot is started with the oscillating infeed. When the required infeed depth is reached, the milling cutter is positioned at the centre of the slot. At this start point, an iterative feed takes place up to the new machining height at the feedrate defined in @P21 until the slot depth @P4 is reached with the addition of the finishing allowance at the base (@P15). The maximum feed in Z per pass is the maximum feed in Z at the maximum value defined in @P5.

At every machining height, the slot is again milled from the centre outwards incrementally at infeed in a semicircle. The clearance of the paths between one pass around never exceeds the maximum value of the maximum feed in XY defined in @P6.

To avoid jerky movements in the milling path and thus relieve the machine tool, it may be useful to activate polynomial contouring during roughing. This also leads to an accelerated execution of the milling cycle. The parameterisation of polynomial smoothing should be made dependent on the selected finishing allowance in order to avoid damage to the outer contour. To avoid residue, the maximum infeed in XY must then be reduced.

The milling cycle with polynomial smoothing can be called as follows:

```
#CONTOUR MODE [DEV, PATH_DEV = V.G.WZ_AKT.R / 10 ] ( Parameterisation )
G261                          ( Activation of polynomial contouring )
L CYCLE [NAME = SysMill... ]                                ( Roughing )
G260                        ( Deactivation of polynomial contouring )
L CYCLE [NAME = SysMill... ]                               ( Finishing )
M30
```

Finishing process

The start point mentioned for roughing is also approached as the starting position for finishing. From there, an interative feed takes place up to the new machining height until the slot depth is reached. The maximum infeed per pass is the maximum infeed in Z at the maximum value defined in @P5.

A distinction is made between two cases for the milling operation at each machining height:

- If no finishing allowance at the base needs to be removed at the current machining height, only the finishing allowance at the edge of the slot is removed. Infeed is vertical.
- If the current machining height also involves a removal of the finishing allowance at the base, the complete circumferential slot must be again machined at this height until the correct dimensions are reached. Infeed in Z is vertical (@P32=1) or helical (@P32=2) or oscillating (@P32=3).

4.10.2 Parameterisation

Required input parameters

Input parameters	Description
@P1	Reference plane (absolute)
@P2	Retraction plane (absolute)
@P3	Safety clearance in Z (relative to reference plane/machining height)
@P4	Depth of slot (relative to reference plane)
@P5	Maximum infeed in Z
@P6	Maximum infeed in XY plane (Maximum 2*V.G.WZ_AKT.R)
@P43	Opening angle of the slot in X
@P44	Number of slots
@P73	Width of slot
@P74	Circle radius of centre point positions

Optional input parameters

Input parameters	Description
@P7	Pitch of plunge helix with helical infeed in Z Default value = 10% of tool radius
@P8	Radius of plunge helix with helical infeed in Z Default value = 10% of tool radius
@P9	Plunge angle for oscillating infeed in Z Default value = 2 degrees
@P13	Safety clearance on retraction in XY plane (relative to milled edge) Default value is half the slot width
@P15	Finishing allowance at base Default value = 0
@P16	Finishing allowance at edge Default value = 0
@P20	Feed rate for roughing for milling in the XY plane (specified in mm/min, inch/min)
@P21	Feed rate for roughing for infeed in Z (specified in mm/min, inch/min)
@P22	Feed rate for finishing for milling in the XY plane (specified in mm/min, inch/min) The default value is the feed rate for roughing
@P23	Feed rate for finishing for infeed in Z (specified in mm/min, inch/min) The default value is the feed rate for finishing for milling in the XY plane
@P30	Machining direction "ClimbMilling" or 0: Down milling (default)

	"UpCutMilling" or 1: Up milling
@P31	Machining mode: "Roughing" or 1: Roughing "Finishing" or 2: Finishing "Roughing+Finishing" or 3: Roughing and finishing (default value)
@P32	Feed mode in Z 1: Vertical 2: Helical 3: Oscillating (default value)
@P40	Position of circle centre point in X (absolute) Default value is the current position
@P41	Position of circle centre point in Y (absolute) Default value is the current position
@P45	Starting angle of first slot (in degrees) Default value = 0
@P46	Incremental angle (in degrees) Default value = 360°/number of slots

The function of the parameters acting in the XY plane is derived largely from the graphic below:

It is recommended to use the Syntax check.to verify whether the input parameters are correctly assigned.

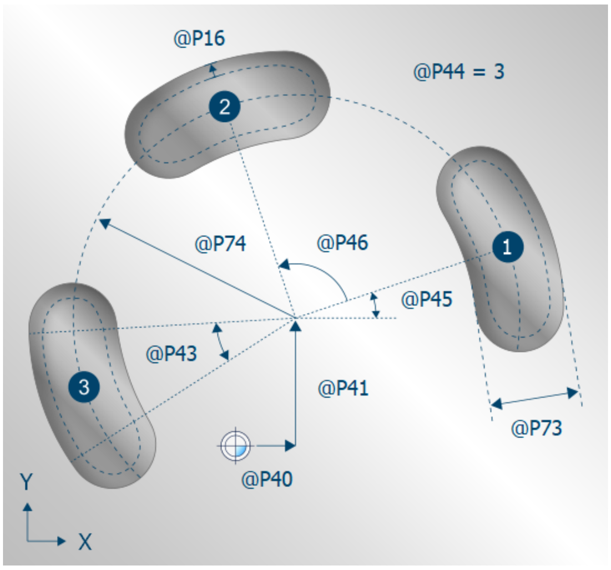

Fig. 62: Top view- circumferential slots

Parameters which mainly refer to the Z axis are displayed in the graphic below:

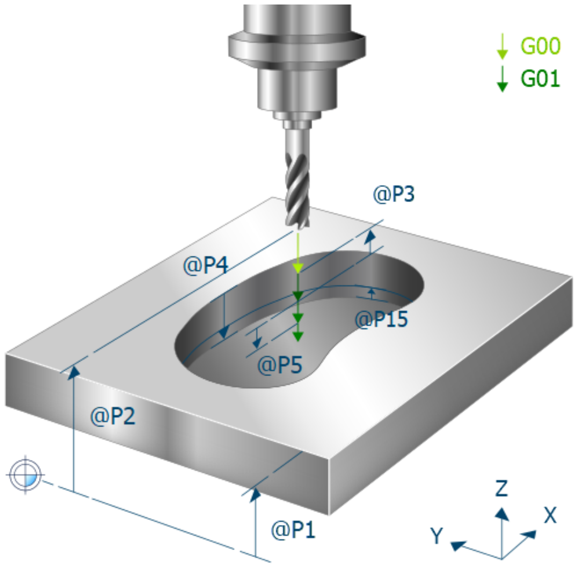

Fig. 63: 3D view - circumferential slots

4.10.3 Syntax

L CYCLE [NAME = SysMillCircumferentialSlot.ecy @P.. = ..]

4.10.4 Programming example

Programing Example

Mill circumferential slots

```
T8 D8                        ( D5 Milling tool data )
M6                           ( Tool change )

G90 G54 S6000 M03            ( Technology data )

G00 Z50                      ( Go to Z start position )
G00 X50 Y50                  ( Go to start position )

#VAR
V.L.SurfacePosition   = 0    ( Z position of workpiece surface )
V.L.RetractionPlane   = 50   ( Z position of retraction plane )
V.L.SafetyClearance   = 2    ( relative value of safety clearance in Z )
V.L.DepthOfSlot       = 10   ( depth of slot )
V.L.MaxIncrementZ     = 4    ( maximum infeed in Z )
V.L.MaxIncrementXY    = V.G.WZ_AKT.R  ( maximum infeed in XY )
V.L.FeedRateXY        = 600  ( machining feedrate in XY )
V.L.FeedRateZ         = 500  ( plunging feedrate )
V.L.SlotAngle         = 30   ( opening angle of the slot )
V.L.NumberOfSlots     = 4    ( number of slots )
V.L.SlotWidth         = 12   ( width of the slot )
V.L.Radius            = 30   ( radius of the slot centres )
#ENDVAR


; polynomial contouring for smooth movements
#CONTOUR MODE [DEV, PATH_DEV = 0.01]
G261

L CYCLE[NAME=SysMillCircumferentialSlot.ecy \
        @P1  = V.L.SurfacePosition          \
        @P2  = V.L.RetractionPlane          \
        @P3  = V.L.SafetyClearance          \
        @P4  = V.L.DepthOfSlot              \
        @P5  = V.L.MaxIncrementZ            \
        @P6  = V.L.MaxIncrementXY           \
        @P20 = V.L.FeedRateXY               \
        @P21 = V.L.FeedRateZ                \
        @P43 = V.L.SlotAngle                \
        @P44 = V.L.NumberOfSlots            \
        @P73 = V.L.SlotWidth                \
        @P74 = V.L.Radius                   \
        ]

G260
M05
M30
```

4.11 SysMillOpenSlot - Milling open slot

This cycle is used to mill an open slot. As opposed to trochoidal milling, plunge milling requires that the slot may not be wider than double the tool diameter.

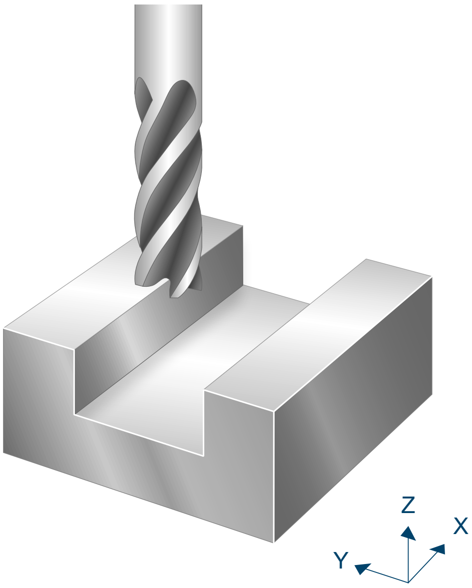

Fig. 64: Mill open slot

4.11.1 Process

To ensure that the cycle is executed successfully, the following requirements must be met.

- The tool radius is defined by V.G.WZ_AKT.R
- Tool length compensation is active.
- The spindle speed must be selected before the cycle is started.
- The slot can be reached without collision starting from the current position at the height of the retraction plane.
- The slot must be approachable without collision along its opening with the clearance [2*V.G.WZ_AKT.R+@P13].
- A maximum of 3 systems may be active or defined when the cycle is called to ensure that sufficient machining coordinate systems (#CS) are available for the cycle.

At the start, the start point in the XY plane is approached at the height of the retraction plane. At this start point, the milling cutter has the safety clearance to the blank as set in @P13. The starting point lies in the direction of the negative X axis relative to the reference point (at orientation angle 0).

After executing the cycle, the milling cutter retracts to its position when the cycle was called.

The slot must be approachable without collision along its opening with the clearance [2*V.G.WZ_AKT.R+@P13]. The dashed area in the graphic below is travelled during the cycle at machining height.

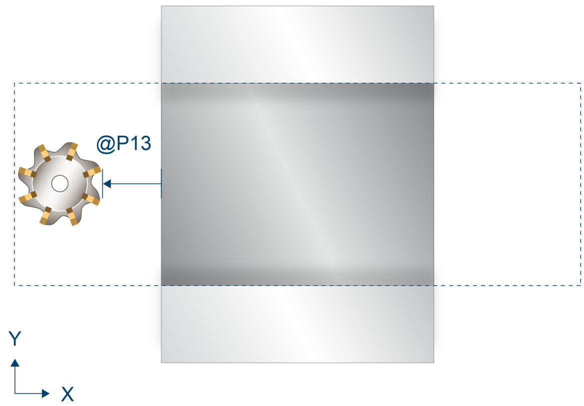

Fig. 65: Minimum clearance/security zone – open slot

Roughing process

A distinction is made with roughing between trochoidal milling (@P32=1) and plunge milling (@P32=2).

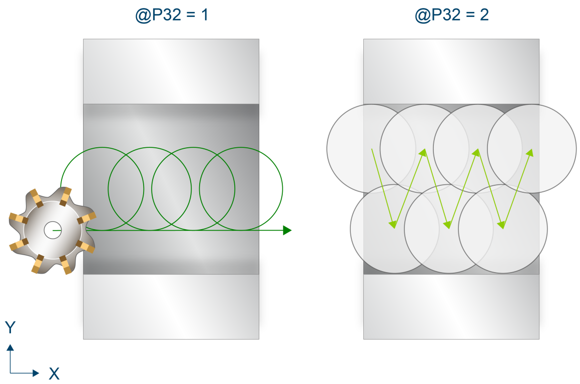

Fig. 66: Trochoidal milling/plunge milling

Trochoidal milling (@P32=1):

The feed is first to the safety plane at rapid traverse along the Z axis followed by a feed to the machining height at the feed rate. At this start point, an iterative feed takes place to the new machining height at the feed rate defined in @P40 until the slot depth @P4 is reached with the addition of the finishing allowance at the base (@P15). The maximum feed in Z per pass is the maximum feed in Z at the maximum value defined in @P5.

The slot is repeatedly milled incrementally in a circular movement for each machining height. The clearance of the paths between each circumferential pass never exceeds the value of the maximum feed in XY defined in @P6.

To avoid jerky movements in the milling path and thus relieve the machine tool, it may be useful to activate polynomial smoothing during roughing. This also leads to an accelerated execution of the milling cycle. The parameterisation of polynomial smoothing should be made dependent on the selected finishing allowance in order to avoid damage to the outer contour. To avoid residue, the maximum infeed in XY must then be reduced.

The milling cycle with polynomial smoothing can be called as follows:

```
#CONTOUR MODE [DEV, PATH_DEV = V.G.WZ_AKT.R / 10 ]   ( Parameterisation )
G261                                         ( Activation of polynomial contouring )
L CYCLE [NAME = SysMill... ]                                         ( Roughing )
G260                                       ( Deactivation of polynomial contouring )
L CYCLE [NAME = SysMill... ]                                        ( Finishing )
M30
```

Plunge milling (@P32=2):

The milling cutter is fed incrementally in the XY plane at rapid traverse at the height of the safety clearance. The clearance between two feeds never exceeds the value of the maximum feed in XY defined in @P6. Milling takes place along the Z axis for every iteration until the slot depth @P4 is reached with the addition of the finishing allowance at the base (@P15).

Prefinishing process

Prefinishing is used to remove residue from the roughing process. The finishing allowances are not removed by milling. The milling cutter is iteratively fed up to the new machining height at the feedrate defined in @P40. The feed in Z per pass is the maximum feed in Z at the maximum value defined in @P5. The slot edge is milled linearly at the feed rate defined in @P20 for every new machining height.

Finishing process

As with prefinishing, however, the finishing allowances (@P15 and @P16) are removed here. The feed rates for finishing are defined in @P22 and @P23.

4.11.2 Parameterisation

Required input parameters

Input parameters	Description
@P1	Reference plane (absolute)
@P2	Retraction plane (absolute)
@P3	Safety clearance in Z (relative to reference plane/machining height)
@P4	Depth of slot (relative to reference plane)
@P5	Maximum infeed in Z (relevant for trochoidal milling)
@P6	Maximum infeed in XY plane (Maximum 2*V.G.WZ_AKT.R)
@P31	Machining mode: 1: Roughing 2: Finishing 3: Roughing and finishing 4: Prefinishing 5: Roughing, prefinishing and finishing
@P43	Orientation angle of slot (in degrees)
@P72	Length of slot
@P73	Width of slot (for plunge milling maximum 4*V.G.WZ_AKT.R)

Optional input parameters

Input parameters	Description
@P13	Safety clearance from blank in XY plane (relative to milled edge) Default value = V.G.WZ_AKT.R

@P15	Finishing allowance at base Default value = 0
@P16	Finishing allowance at edge Default value = 0
@P20	Feed rate for roughing for milling in the XY plane (specified in mm/min, inch/min)
@P21	Feed rate for roughing for infeed in Z (specified in mm/min, inch/min)
@P22	Feed rate for finishing for milling in the XY plane (specified in mm/min, inch/min) The default value is the feed rate for roughing
@P23	Feed rate for finishing for infeed in Z (specified in mm/min, inch/min) The default value is the feed rate for finishing for milling in the XY plane
@P30	Machining direction (relevant for trochoidal milling) "ClimbMilling" or 0: Down milling (default) "UpCutMilling" or 1: Up milling
@P32	Infeed mode 1: Trochoidal milling (default) 2: Plunge milling
@P40	Reference point in X (absolute) Default value is the current position
@P41	Reference point in Y (absolute) Default value is the current position

The function of the parameters acting in the XY plane is derived largely from the graphic below:

It is recommended to use the Syntax check.to verify whether the input parameters are correctly assigned.

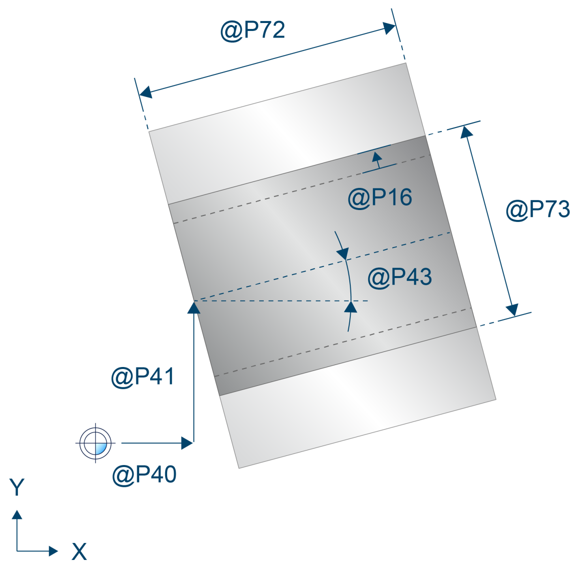

Fig. 67: Top view - open slot

Parameters which mainly refer to the Z axis are displayed in the graphic below:

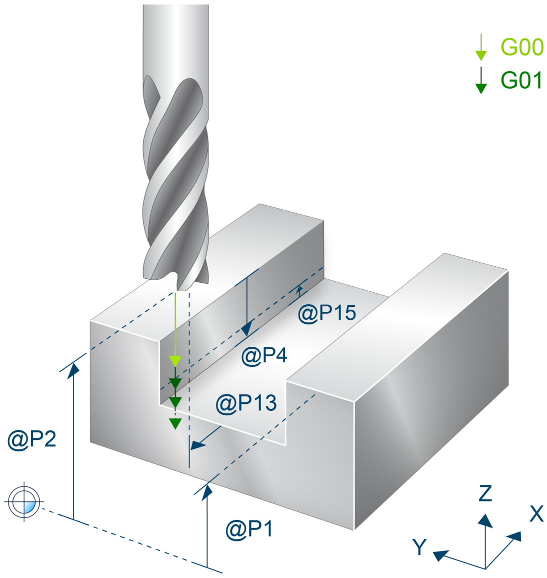

Fig. 68: 3D view - open slot

4.11.3 Syntax

L CYCLE [NAME = SysMillOpenSlot.ecy @P.. = ..]

4.11.4 Programming example

Programing Example

Milling open slot

```
T8 D8                         ( Tool data )
M6                            ( Tool change )


G90 G54 S6000 M03 F4000       ( Technology data )

#VAR
  ; input parameters:
  V.L.SurfacePosition    = 0    ( Z position of workpiece surface )
  V.L.RetractionPlane   = 50   ( Z position of retraction plane )
  V.L.SafetyClearance    = 10   ( relative value of safety clearance in Z)
  V.L.DepthOfSlot        = 4    ( depth of slot )
  V.L.MaxIncrementZ      = 20   ( maximum infeed in Z )
  V.L.MaxIncrementXY     = 2    ( maximum infeed in XY )
  V.L.FinishingOffsetXY = 0.2 ( finishing offset in XY )
  V.L.FeedRateXY         = 6000   ( machining feedrate in XY )
  V.L.FeedRateZ          = 4000   ( plunging feedrate )
  V.L.MachiningMode      = 5    ( machining mode )
  V.L.InfeedMode         = 1    ( infeed mode )
  V.L.OrientationAngle   = 0    ( orientation angle of slot )
  V.L.SlotLength         = 100  ( length of the slot )
  V.L.SlotWidth          = 9    ( width of the slot )
#ENDVAR


G00 Z100
G00 X0 Y50                    ( Positioning to the starting point )

; polynomial contouring for smooth movements
#CONTOUR MODE [DEV, PATH_DEV = V.G.WZ_AKT.R / 100]
G261

L CYCLE[NAME=SysMillOpenSlot.ecy  \
    @P1  = V.L.SurfacePosition    \
    @P2  = V.L.RetractionPlane    \
    @P3  = V.L.SafetyClearance    \
    @P4  = V.L.DepthOfSlot        \
    @P5  = V.L.MaxIncrementZ      \
    @P6  = V.L.MaxIncrementXY     \
    @P16 = V.L.FinishingOffsetXY  \
    @P20 = V.L.FeedRateXY         \
    @P21 = V.L.FeedRateZ          \
    @P31 = V.L.MachiningMode      \
    @P32 = V.L.InfeedMode         \
    @P43 = V.L.OrientationAngle   \
    @P72 = V.L.SlotLength         \
    @P73 = V.L.SlotWidth          \
    ]

G260
M05
M30
```

4.12 SysMillThread - Thread milling cycle

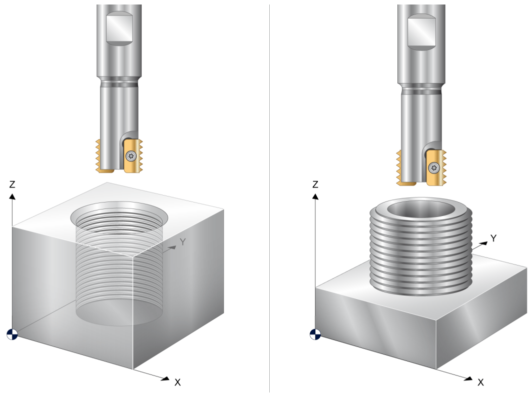

Fig. 69: Milling thread

4.12.1 Process

The SysMillThread cycle can be used to mill inner and outer threads. The path for thread milling is based on a helical interpolation. The centre point of the thread is the starting point. This is where the tool must be positioned when the cycle is called. At the end of the cycle, the cycle retracts to the starting point.

There are three tool types available:

- Tool type 1: The thread is milled once to final depth by the full profile thread milling cutter.
- Tool type 2: The indexable insert thread milling cutter is offset by the cutting width of the indexable inserts until full depth is reached.
- Tool type 3: The single-flute thread milling cutter traverses the thread pitch from top to final depth in a continuous helical movement.

4.12.2 Parameterisation

Required input parameters

Input parameters	Description
@P1	Reference plane (absolute)
@P2	Retraction plane (absolute)
@P3	Safety clearance (relative to the reference plane, unsigned)
@P4	Depth in Z axis (relative to reference plane, unsigned)
@P7	Thread pitch (unsigned)
@P68	Nominal thread diameter

Notice

The safety clearance @P3 may be corrected slightly upwards in the cycle to achieve the correct thread pitch at the specified drilled hole depth.

Optional input parameters

Input parameters	Description
@P6	Maximum infeed in XY plane (default = removed in an infeed)
@P16	Finishing allowance Default value = 0
@P30	Machining direction: 0 = Clockwise (default value) 1 = Counter-clockwise
@P31	Machining mode: "Roughing" or 1: Roughing (default value) "Finishing" or 2: Finishing "Roughing+Finishing" or 3: Roughing and finishing
@P37	Machining mode: 1 = Inner (default value) 2 = Outer
@P63	Tool type 1 = Full profile thread milling cutter (default value) 2 = Indexable insert thread milling cutter 3 = Single-flute thread milling cutter
@P64	Indexable insert cutting width (when indexable insert thread milling cutter is used)

It is recommended to use the Syntax check.to verify whether the input parameters are correctly assigned.

Machining modes:

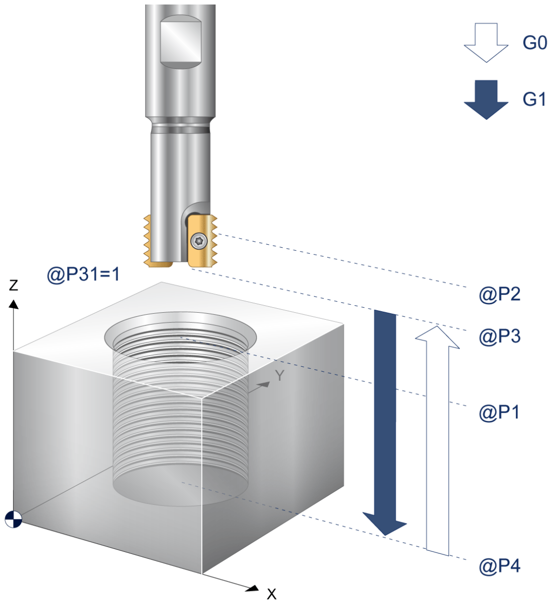

Fig. 70: Parameter for inner machining mode

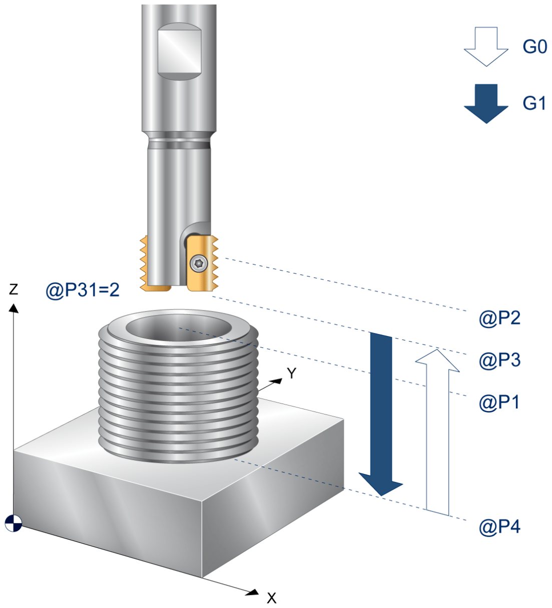

Fig. 71: Parameter for outer machining mode

Tool types

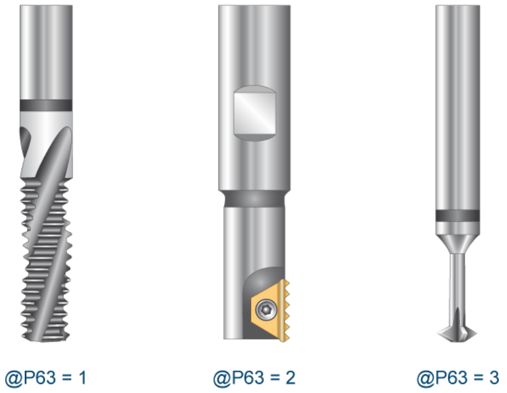

Fig. 72: Thread milling cutter types

4.12.3 Syntax

L CYCLE [NAME = SysMillThread.ecy @P.. = ..]

4.12.4 Programming example

Programing Example

Thread milling

```
; Thread milling
T4 D4                         (Tool data)
M6                            (Tool change)
G17 G90 G54 F500 M03 S1200    (Technology data)
G00 Z50                       (Go to z start position)
G00 X50 Y50   (position near workpiece mill tool is outside the part)

#VAR
  ; input parameters:
  V.L.SurfacePosition     = 0   (Z position of workpiece surface)
  V.L.RetractionPlane     = 20  (Z position of retraction plane)
  V.L.SafetyClearance     = 4   (rel. value of safety clearance in Z)
  V.L.DepthOfThread       = -15 (depth of thread)
  V.L.MaxIncrementXY      = 0.5 (maximal increment in XY)
  V.L.ThreadPitch         = 1   (pitch of the thread)
  V.L.FinishingOffsetXY   = 0.2 (finishing offset in XY)
  V.L.MachiningMode       = 3   (machining mode)
  V.L.ToolTyp             = 3   (tool type)
  V.L.ThreadDiameter      = 20  (diameter of the thread)
#ENDVAR

L CYCLE [NAME=SysMillThread.ecy  \
    @P1  = V.L.SurfacePosition   \
    @P2  = V.L.RetractionPlane   \
    @P3  = V.L.SafetyClearance   \
    @P4  = V.L.DepthOfThread     \
    @P6  = V.L.MaxIncrementXY    \
    @P7  = V.L.ThreadPitch       \
    @P16 = V.L.FinishingOffsetXY \
    @P31 = V.L.MachiningMode     \
    @P63 = V.L.ToolTyp           \
    @P68 = V.L.ThreadDiameter    \
    ]

G00 Z100
M05
M30
```

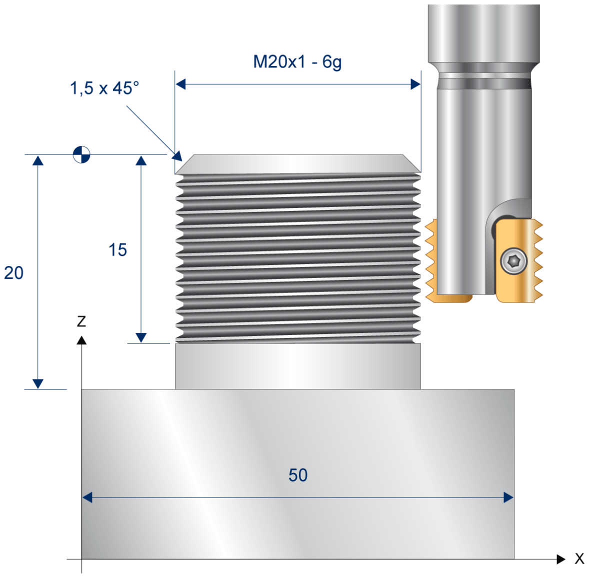

Fig. 73: Programming example

5 Contour milling

5.1 Overview

Notice

This function is available as of CNC Build V3.01.3079.25.

Contour milling cycles are used to mill geometrical contours consisting of linear and circular segments, The user can specify them in almost any combination. When machining, it is possible to distinguish between pocket contours, blank contours and island contours. This allows the combination of already defined contours.

5.1.1 Possible applications

The following application are possible for contour milling:

- Milling a contour pocket including residual material removal
- Milling a contour spigot including residual material removal
- Milling along a defined contour
- Preparatory predrilling before contour milling

5.1.2 Definition of a contour in NC program code

Time of definition

No contours are predefined when the controller starts up. A definition in the configuration lists is not possible. Contours are defined directly in the NC program in a sequence of path motions embedded in plain text commands. The contours used must then be defined before a machining cycle is called. The contour definition is valid until it is overwritten, deleted or up to program end.

Start of a contour definition

```
# CONTOUR BEGIN [ID<expr>]
```

ID <expr> Identification number of the contour.

A contour definition is activated by #CONTOUR BEGIN [ID<expr>]. The freely selectable identification number is then transferred. If a contour already exists with the required identification number, it is overwritten by the new contour.

End of a contour definition

```
# CONTOUR END
```

Each contour definition must be closed by the command #ONTOUR END. Only at the end of the contour definition can standard commands be used again.

Description of programming - Description of a contour

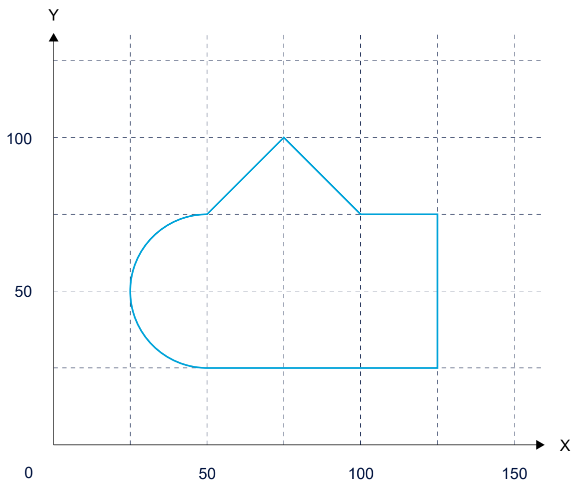

Fig. 74: Example 1 – Contour definition

```
#CONTOUR BEGIN [ID1]
G1 G90 X125 Y25 (start point)
G1 X50
G2 G161 X50 Y75 I50 J50
G1 X75 Y100
G1 X100 Y75 F2000
G1 X125 Y75
G1 Y25
#CONTOUR END
```

Every control area definition begins with #CONTROL BEGIN and must be terminated with #CONTROL END. Between these commands, the geometrical form of the contour is defined by DIN 66025 move commands (G01, G02, G03).

The end point of the first motion block in the contour definition described the start point of the contour. The start point must be defined by a linear motion block (G1). It is read in as absolute irrespective of the dimension system (G90/91). After defining the start point, it is possible to switch between a absolute and a relative description (G90/G91, G161/G162). The use of full circles is not permitted in the contour description..

In addition, it is possible to provide individual contour elements with a feed using the F word. Within the contour definition, this is non modal but must be defined separately for each element as required. Depending on the machining cycle used for the defined contour, there is a difference in the meaning of the defined feed, See the individual cycle descriptions for more detailed information.

Attention

Active Cartesian transformations and offsets are not considered in the definition of the contour. The contour geometry is always specified in the PCS coordinate system. .

Programming example - Adding chamfers and roundings

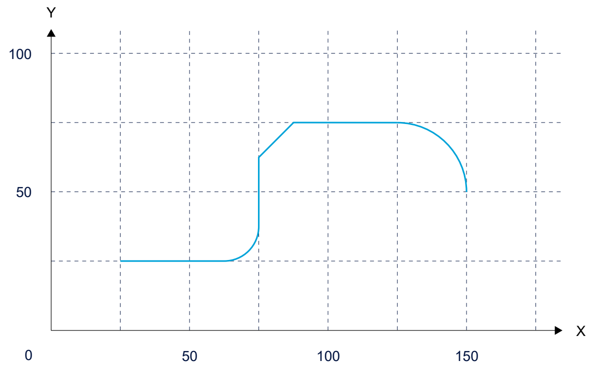

Fig. 75: Example 2 – Contour definition

```
# CONTOUR BEGIN[ID1]
G1 G90 X25 Y25  (start point)
G1 G91 X50
G302 I12.5
G1 G91 Y50
G301 I12.5 J12.5
G1 G91 X50
G02 G91 X25 Y-25 I0 J-25
# CONTOUR END
```

Chamfers and roundings can be added to the contour by using G301/G302. Here, a feed definition with #FRC. is not possible.

5.1.3 Delete contours

The command to delete a contour contains the following syntax elements:

```
# CONTOUR DELETE [ID<expr>]
```

ID <expr> Identification number of the contour

It is also possible to delete currently defined contours and clear the memory location for new definitions.

```
# CONTOUR DELETE ALL
```

Information on deleted contours is irrevocably lost. The assigned memory location is then released for new contour definitions. Only already defined contours can be deleted.

Programming example - Delete contours

```
# PATTERN DELETE [ID3]     ( Delete specific contour with ID 3)
# PATTERN DELETE ALL       ( Delete all contours)
```

5.2 SysMillContourPocket - Mill contour pocket

This cycle is used to mill a contour pocket.

The geometrical description of the contour pocket in the X and Y axes is provided by a contour definition Contour definition [▶ 128].

Up to 10 island contours can be added to the contour pocket. To facilitate use of contour milling, the contour of the pocket can overlap the island regions. The relevant areas can be calculated from this by using an internal intersection point calculation.

Island contours may not overlap. If island contours overlap, they must be programmed as one island contour.

It is possible to machine a contour pocket with different milling cutter radii. For more information on this, see the related Cycle description [▶ 148].

To mill the contour pocket, the tool is lowered over a calculated plunge point. With complex contour pockets it may be necessary to perform a second plunge during surface machining. If the milling cutter does not cut over centre, the plunge points can be calculated in advance and machined in a predrilling cycle [▶ 154].

In the cycle, the tool is fed incrementally with a helical or vertical movement along the Z axis. The pocket is milled out in parallel to the contour at each machining height. To avoid residue, infeed in the XY plane is limited to the tool radius.

Attention

If the contour pocket contains islands or complex geometrical shapes, it is possible that the maximum infeed in the XY plane is violated and the milling cutter enters the material at full circumference. This must be considered when the milling feed is selected.

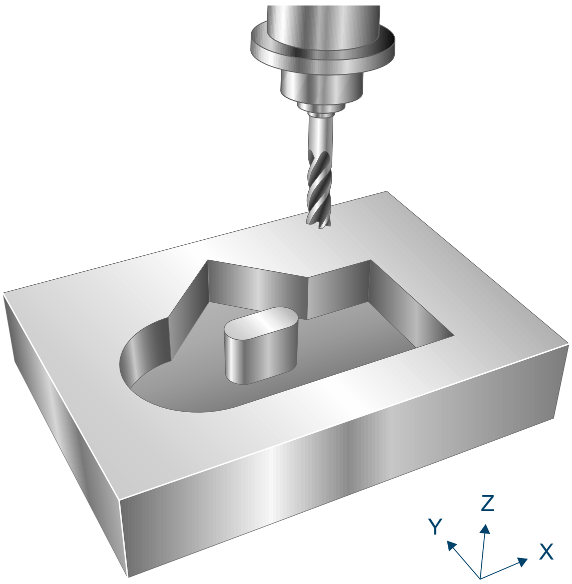

Fig. 76: SysMillContourPocket

5.2.1 Process

To ensure that the cycle is executed successfully, the following requirements must be met.

- The tool radius is defined by V.G.WZ_AKT.R
- Tool length compensation is active.
- The spindle speed must be selected before the cycle is started.

- The contour pocket can be reached without collision starting from the current position at the height of the retraction plane.
- The contours used were already programmed using #CONTOUR BEGIN / #CONTOUR END.
- The defined contours for pocket and islands are closed, i.e. start and end points are identical.

Within the cycle, the tool is incrementally fed helically or vertically along the Z axis. The starting point is a start point located above the contour pocket. For a helical plunge, the system checks whether the defined radius of the plunging helix leads to a contour violation.

The pocket is then milled out again parallel to the contour at each machining height. When the pocket is fully milled at the current machining height, the tool is retracted at feedrate in a helical movement in a semicircle in XY (maximum infeed in XY, relative) and Z (@P3, relative) and re-positioned at the matching plunge point at rapid traverse. It is then positioned at feedrate at the next machining height.

If individual contour elements were specified at a feedrate during the contour definition, it is activated for the particular element during edge finishing.

When complex contours are milled, there may be milling movements containing no material removal. If the length of an empty section exceeds a certain value, the cycle will attempt internally to lift the cutter at certain points and replunge at a position that has already been removed.

Roughing process

When roughing, the contour pocket is milled out again parallel to the contour at each machining height. The specified finishing allowance is added to the edge and base of the contour.

To avoid jerky movements in the milling path and thus relieve the machine tool, it may be useful to activate polynomial smoothing during roughing. This also leads to an accelerated execution of the contour milling cycle. The parameterisation of polynomial smoothing should be made dependent on the selected finishing allowance in order to avoid damage to the outer contour. To avoid residue, the maximum deviation from the calculated milling path (PATH_DEV) must not exceed the tool radius.

The contour milling cycle with polynomial contouring can be called as follows:

```
#CONTOUR MODE [DEV, PATH_DEV = V.G.WZ_AKT.R / 2]     ( Parameterisation )
G261                                  ( Activation of polynomial contouring )
L CYCLE [NAME = SysMillContourPocket.ecy @P31 = 1, @P... ]   ( Roughing )
G260                                ( Deactivation of polynomial contouring )
L CYCLE [NAME = SysMillContourPocket.ecy @P31 = 2, @P... ] ( Finishing )
M30
```

Finishing process

When finishing the contour pocket, the maximum infeed in Z is also iteratively fed at the new machining height until the pocket depth is reached.

A distinction can be made between two cases for the milling process during finishing:

- If no finishing allowance at the base needs to be removed at the current machining height, only the finishing allowance at the edge of the contour (island) is removed. The milling cutter is fed slightly offset to the edge of the contour to ensure a faster plunge.
- If the current machining height also involves a removal of the finishing allowance at the base, the complete contour pocket must be again machined at this height until the correct dimensions are reached. The cutter is lowered above the calculated plunge position for this purpose.

5.2.2 Parameterisation

Required input parameters

Input parameters	Description

@P1	Reference plane (absolute)
@P2	Retraction plane (absolute)
@P3	Safety clearance in Z (relative to reference plane/machining height)
@P4	Pocket depth (relative to reference plane)
@P5	Maximum infeed in Z
@P6	Maximum infeed in XY (Maximum V.G.WZ_AKT.R is recommended to avoid residual material)
@P50	Identification number of the pocket contour

Optional input parameters

Input parameters	Description
@P7	Pitch of plunge helix Default value = 10% of tool radius
@P8	Radius of plunge helix Default value = 50% of tool radius
@P15	Finishing allowance at base Default value = 0
@P16	Finishing allowance at edge Default value = 0
@P20	Feed rate for roughing for milling in the XY plane (specified in mm/min, inch/min)
@P21	Feed rate for roughing in Z (specified in mm/min, inch/min)
@P22	Feed rate for finishing for milling in the XY plane (specified in mm/min, inch/min) The default value is the feed rate for roughing
@P23	Feed rate for finishing for infeed in Z (specified in mm/min, inch/min) The default value is the feed rate for finishing for milling in the XY plane
@P25	Spindle speed for finishing (specified in rpm)
@P30	Machining direction "ClimbMilling" or 0: Down milling (default) "UpCutMilling" or 1: Up milling
@P31	Machining mode: "Roughing" or 1: Roughing "Finishing" or 2: Finishing "Roughing+Finishing" or 3: Roughing and finishing (default value)
@P32	Feed mode in Z 1: Vertical 2: Helical (default)
@P135	Maximum infeed in XY for edge finishing Default value is the maximum infeed in XY for roughing

@P170	Maximum infeed in XY for base finishing Default value is the maximum infeed in XY for roughing
@P171	Feed rate for plunging in previously milled areas Default value is the feed rate for roughing for the infeed in Z
@P51	Identification number of an island contour
@P52	Identification number of an island contour
@P53	Identification number of an island contour
@P54	Identification number of an island contour
@P55	Identification number of an island contour
@P56	Identification number of an island contour
@P57	Identification number of an island contour
@P58	Identification number of an island contour
@P59	Identification number of an island contour
@P60	Identification number of an island contour

The function of the parameters acting in the XY plane is derived largely from the graphic below:

It is recommended to use the Syntax check.to verify whether the input parameters are correctly assigned.

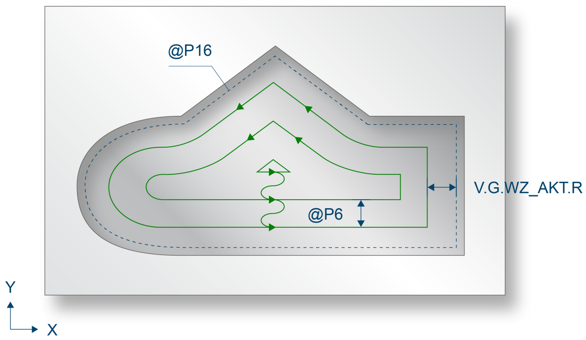

Fig. 77: SysMillContourPocket - 2D view

Parameters which mainly refer to the Z axis are displayed in the graphic below:

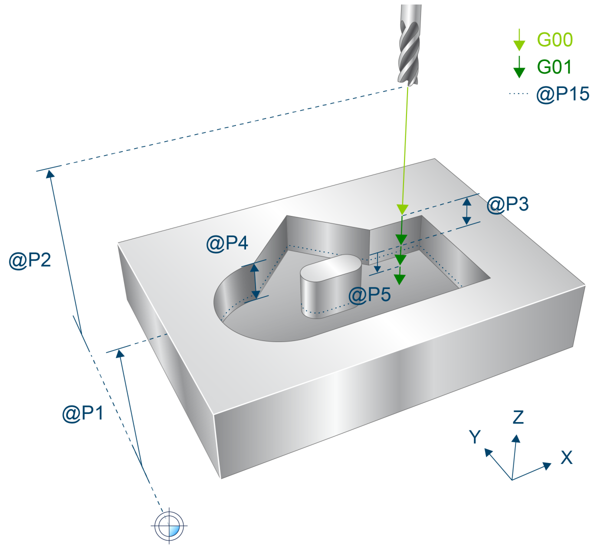

Fig. 78: SysMillContourPocket - 3D view

5.2.3 Syntax

L CYCLE [NAME=SysMillContourPocket.ecy @P.. = ..]

5.2.4 Programming example

Programing Example

Mill contour pocket

This example describes milling of a contour pocket defined in the NC program. This results in the following cycle call:

```
T8 D8                              ( Tool data )
M6                                 ( Tool change )

G90 G54 M03 S6000 F5000            ( Technology data )
G00 Z100                           ( Go to z start position )
G00 X0 Y0                          ( Go to start position )

#VAR
  V.L.SurfacePosition = 0          ( Z position of workpiece surface )
  V.L.RetractionPlane = 100        ( Z position of retraction plane )
  V.L.SafetyClearance = 5          ( relative value of safety clearance )
  V.L.DepthOfPocket   = 20         ( depth of pocket )
  V.L.MaxIncrementZ   = V.L.DepthOfPocket  / 2  ( maximum infeed in Z )
  V.L.MaxIncrementXY  = V.G.WZ_AKT.R*0.9        ( maximum infeed in XY )
  V.L.FeedRateZ       = 150        ( plunging feedrate )
  V.L.ContourID       = 1          (identification number pocket contour)
#ENDVAR

; contour definition:
#CONTOUR BEGIN[ID = 1]
G1 G90 X90 Y20
G1 G91 Y40
G1 G91 X-20
G1 G90 X50 Y80
G1 X30 Y60
G161 G03 X30 Y20 I30 J40
G1 X90
#CONTOUR END

G0 ZV.L.RetractionPlane

; polynomial contouring for smooth movements
#CONTOUR MODE [DEV, PATH_DEV = V.G.WZ_AKT.R / 100]
G261

L CYCLE [NAME = SysMillContourPocket.ecy  \
    @P1  = V.L.SurfacePosition            \
    @P2  = V.L.RetractionPlane            \
    @P3  = V.L.SafetyClearance            \
    @P4  = V.L.DepthOfPocket              \
    @P5  = V.L.MaxIncrementZ              \
    @P6  = V.L.MaxIncrementXY             \
    @P21 = V.L.FeedRateZ                  \
    @P31 = "Roughing"                     \
    @P50 = V.L.ContourID                  \
    ]

G260
M30
```

5.3 SysMillContourSpigot - Mill contour spigot

This cycle is used to mill a contour spigot.

To carry out contour spigot milling, the contours of the blank and the individual spigots are transferred to the cycle. The cycle removes the area of the blank contour with the exception of the defined spigots.

The geometrical description of the blank and the spigot contours in the X and Y axes is provided by a contour definition Contour definition [▸ 128].

Up to 10 spigot contours can be added to the blank. They are also described by a contour definition. For spigot milling, they may not intersect with the blank contour.

It is possible to machine a contour spigot with different milling cutter radii. For more information on this, see the related Cycle description [▸ 148].

To mill the contour spigot, the tool is lowered to the side of the blank. With complex contour spigots it may be necessary to perform a second plunge into the raw material during surface machining. Here you can activate a helical plunging operation. The plunge radius is 50% of the tool radius if not otherwise specified but this can be changed by adding an input parameter. This cycle continues the vertical feed in Z after helical plunge is activated. Only then can the helical variant be used if a plunge takes place in the blank. If the milling cutter does not cut over centre, the plunge points can be calculated in advance and machined in a predrilling cycle [▸ 154].

In the cycle, the spigot is milled out incrementally parallel to the contour at each machining height. To avoid residue, infeed in the XY plane is limited to the tool radius.

Attention

For contour spigot milling, it is possible that the maximum infeed in the XY plane is violated and the milling cutter enters the material at full circumference. This must be considered when the milling feed is selected.

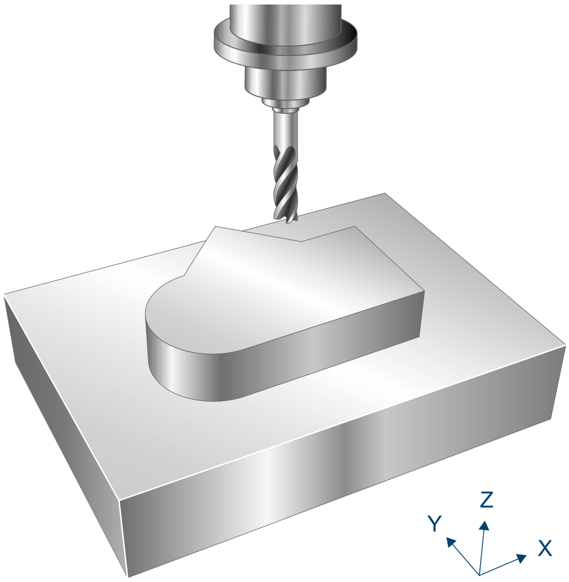

Fig. 79: SysMillContourSpigot

5.3.1 Process

To ensure that the cycle is executed successfully, the following requirements must be met.

- The tool radius is defined by V.G.WZ_AKT.R
- Tool length compensation is active.
- The spindle speed must be selected before the cycle is started.

- The contour spigot can be reached without collision starting from the current position at the height of the retraction plane.
- The blank can be circumvented at machining height at a distance of 2*V.G.WZ_AKT.R without collision.
- The contours used were already programmed using #CONTOUR BEGIN / #CONTOUR END.
- The defined contours for the blank and spigots are closed, i.e. start and end points are identical.

Within the cycle, the tool is fed incrementally along the Z axis. The starting point is a start point located above the blank. The blank is then milled out again parallel to the contour at each machining height. When the blank is fully machined at the current machining height, the tool is retracted and positioned again at rapid traverse. It is then positioned at feedrate at the next machining height.

If individual contour elements of the spigot were specified at a feedrate during the contour definition, it is activated for the particular element during edge finishing.

When complex contours are milled, there may be milling movements containing no material removal. If the length of an empty section exceeds a certain value, the cycle will attempt internally to lift the cutter at certain points and replunge at a position that has already been removed.

Roughing process

When roughing, the contour spigot is milled out again parallel to the contour at each machining height. The specified finishing allowance is added to the edge of the contour spigot and to the base of the blank contour.

To avoid jerky movements in the milling path and thus relieve the machine tool, it may be useful to activate polynomial smoothing during roughing. This also leads to an accelerated execution of the contour milling cycle. The parameterisation of polynomial smoothing should be made dependent on the selected finishing allowance in order to avoid damage to the outer contour. To avoid residue, the maximum deviation from the calculated milling path (PATH_DEV) must not exceed the tool radius.

The contour milling cycle with polynomial contouring can be called as follows:

```
#CONTOUR MODE [DEV, PATH_DEV = V.G.WZ_AKT.R / 2]     ( Parameterisation )
G261                             ( Activation of polynomial contouring )
L CYCLE [NAME = SysMillContourSpigot.ecy @P31 = 1, @P... ]   ( Roughing )
G260                           ( Deactivation of polynomial contouring )
L CYCLE [NAME = SysMillContourSpigot.ecy @P31 = 2, @P... ] ( Finishing )
M30
```

Finishing process

When finishing the contour spigot, the maximum infeed in Z is also iteratively fed up to the new machining height until the spigot depth is reached.

A distinction can be made between two cases for the milling process during finishing:

- If no finishing allowance at the base needs to be removed at the current machining height, only the finishing allowance at the edge of the spigot contours is removed. The milling cutter is fed slightly offset to the edge of the contour to ensure a faster plunge. - If the current machining height also involves a removal of the finishing allowance at the base, the complete contour spigot must be again machined at this height until the correct dimensions are reached.

5.3.2 Parameterisation

Required input parameters

Input parameters	Description

@P1	Reference plane (absolute)
@P2	Retraction plane (absolute)
@P3	Safety clearance in Z (relative to reference plane/machining height)
@P4	Spigot depth (relative to reference plane)
@P5	Maximum infeed in Z
@P6	Maximum infeed in XY (Maximum V.G.WZ_AKT.R is recommended to avoid residual material)
@P31	Machining mode: "Roughing" or 1: Roughing "Finishing" or 2: Finishing "Roughing+Finishing" or 3: Roughing and finishing (default value)
@P50	Identification number of the blank contour

Optional input parameters

Input parameters	Description
@P7	Pitch of plunge helix Default value = 10% of tool radius
@P8	Radius of plunge helix Default value = 50% of tool radius
@P15	Finishing allowance at base Default value = 0
@P16	Finishing allowance at edge Default value = 0
@P20	Feed rate for roughing for milling in the XY plane (specified in mm/min, inch/min)
@P21	Feed rate for roughing for infeed in Z (specified in mm/min, inch/min)
@P22	Feed rate for finishing for milling in the XY plane (specified in mm/min, inch/min) The default value is the feed rate for roughing
@P23	Feed rate for finishing for infeed in Z (specified in mm/min, inch/min) The default value is the feed rate for finishing for milling in the XY plane
@P25	Spindle speed for finishing (specified in rpm)
@P30	Machining direction "ClimbMilling" or 0: Down milling (default) "UpCutMilling" or 1: Up milling
@P32	Infeed mode in Z (if repeated plunge into blank is necessary due to bottlenecks) 1: Vertical (default value) 2: Helical
@P135	Maximum infeed in XY for edge finishing Default value is the maximum infeed in XY for roughing

@P170	Maximum infeed in XY for base finishing Default value is the maximum infeed in XY for roughing
@P171	Feed rate for plunging in previously milled areas Default value is the feed rate for roughing for the infeed in Z
@P51	Identification number of an island contour
@P52	Identification number of an island contour
@P53	Identification number of an island contour
@P54	Identification number of an island contour
@P55	Identification number of an island contour
@P56	Identification number of an island contour
@P57	Identification number of an island contour
@P58	Identification number of an island contour
@P59	Identification number of an island contour
@P60	Identification number of an island contour

The function of the parameters acting in the XY plane is derived largely from the graphic below:

It is recommended to use the Syntax check.to verify whether the input parameters are correctly assigned.

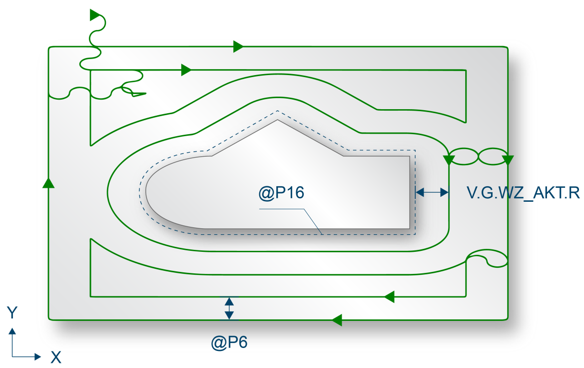

Fig. 80: SysMillContourSpigot - 2D view

Parameters which mainly refer to the Z axis are displayed in the graphic below:

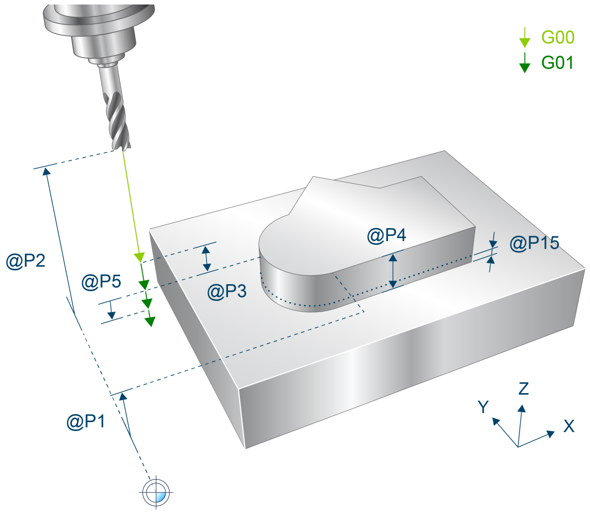

Fig. 81: SysMillContourSpigot - 3D view

5.3.3 Syntax

L CYCLE [NAME = SysMillContourSpigot.ecy @P.. = ..]

5.3.4 Programming example

Programing Example

Mill contour spigot

This example describes milling of a blank contour pocket with an island region (spigot) in the NC program. This results in the following cycle call:

```
T20 D20                           ( Tool data )
M6                                ( Tool change )

G90 G54 S6000 M03 F5000           ( Technology data )
G00 Z100                          ( Go to Z start position )
G00 X0 Y0                         ( Go to start position )

#VAR
  V.L.SurfacePosition = 0         ( Z position of workpiece surface )
  V.L.RetractionPlane = 100       ( Z position of retraction plane )
  V.L.SafetyClearance = 10        ( relative value of safety clearance)
  V.L.Depth           = 10        ( depth of pocket )
  V.L.MaxIncrementZ   = V.L.Depth/2        ( maximum infeed in Z )
  V.L.MaxIncrementXY  = V.G.WZ_AKT.R*0.8   ( maximum infeed in XY )
  V.L.FeedRateZ       = 1000      ( plunging feedrate )
  V.L.ContourID       = 1         ( identification number raw contour)
  V.L.IsleID1         = 2         ( identification number spigot contour)
#ENDVAR

; raw contour definition:
#CONTOUR BEGIN[ID = 1]
G1 G90 X0 Y0
G91 X100
G91 Y100
```

```
G91 X-100
G91 Y-100
#CONTOUR END

; spigot contour definition:
#CONTOUR BEGIN[ID = 2]
G1 G90 X90 Y20
G1 G91 Y40
G1 G91 X-20
G1 G90 X50 Y80
G1 X30 Y60
G161 G03 X30 Y20 I30 J40
G1 X90
#CONTOUR END


; polynomial contouring for smooth movements
#CONTOUR MODE [DEV, PATH_DEV = V.G.WZ_AKT.R / 100]
G261

L CYCLE [NAME = SysMillContourSpigot.ecy  \
    @P1  = V.L.SurfacePosition            \
    @P2  = V.L.RetractionPlane            \
    @P3  = V.L.SafetyClearance            \
    @P4  = V.L.Depth                      \
    @P5  = V.L.MaxIncrementZ              \
    @P6  = V.L.MaxIncrementXY             \
    @P21 = V.L.FeedRateZ                  \
    @P31 = "Roughing"                     \
    @P50 = V.L.ContourID                  \
    @P51 = V.L.IsleID1                    \
    ]

G260
M05
M30
```

5.4 SysMillContourPath - Path milling

This cycle is for path milling.

The path milling cycle can remove a 2D contour path defined in advance. The path need not be closed (identical start and end). For machining, tool radius compensation with optional finishing allowance at the edge and base can also be selected. In addition, various approach and withdrawal movements can be added to the milling process.

The geometrical description of the contour in the X and Y axes is provided by a Contour definition [▸ 128].

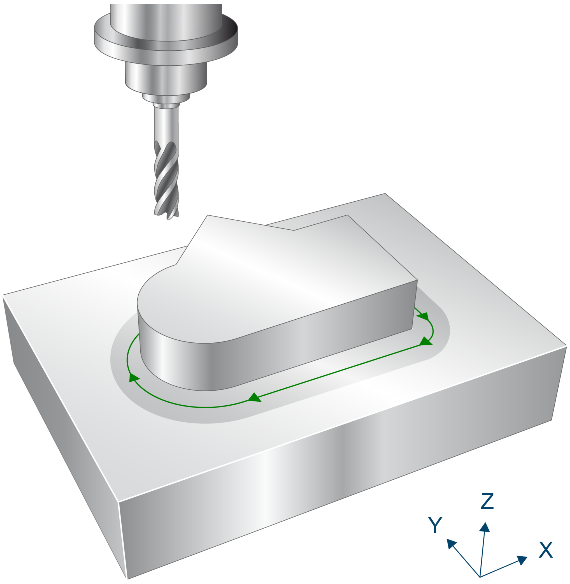

Fig. 82: SysMillContourPath

5.4.1 Process

To ensure that the cycle is executed successfully, the following requirements must be met.

- The tool radius is defined by V.G.WZ_AKT.R
- Tool length compensation is active.
- The spindle speed must be selected before the cycle is started.
- The start and end points must be reachable collision-free, starting from the current position at the height of the retraction plane.
- The contours used were already programmed using #CONTOUR BEGIN / #CONTOUR END.

Infeed in Z takes place according to the parameterisation either vertically along the Z axis at the beginning of the approach movement (contour) or obliquely in space during the approach movement. Retraction in Z takes place analogously.

The tool is moved along the contour at the specified feedrate at each machining height. After the contour is removed, the tool is retracted and, if necessary, repositioned at the starting point for the next pass.

The use of finishing allowance at the edge is only allowed when tool radius compensation is active (@P75). In this case, the offset to the contour during roughing is increased by the finishing allowance. During finishing, the finishing allowance at the edge is removed first and then the finishing allowance at the base.

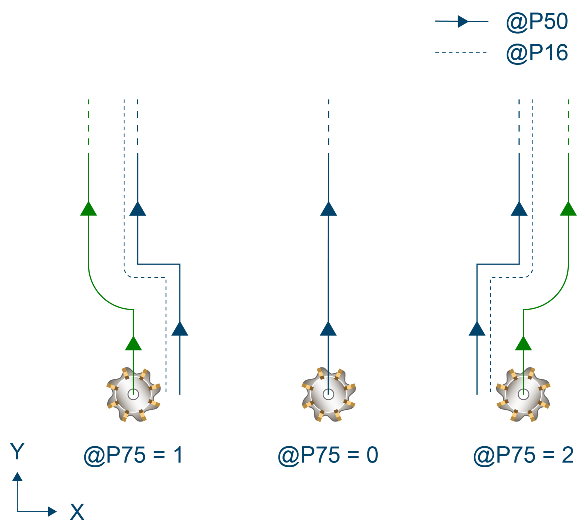

Fig. 83: Use of finishing allowance with active tool radius compensation

After the finishing allowance an extra roughing allowance (@P95) can be added. When it is activated, material is removed in parallel to the contour at the infeed defined in @P6.

If individual contour elements were specified at a feedrate during the contour definition, it is activated for the particular element (independently from @ P23). If a finishing allowance with tool radius compensation was specified, the selected feedrate is only activated during the finishing pass.

5.4.2 Parameterisation

Required input parameters:

Input parameters	Description
@P1	Reference plane (absolute)
@P2	Retraction plane (absolute)
@P3	Safety clearance in Z (relative to reference plane/machining height)
@P4	Depth (relative to reference plane)
@P5	Maximum infeed in Z
@P50	Identification number of the path contour

Optional input parameters:

Input parameters	Description
@P6	Infeed in XY to remove the additional roughing allowance (@P95). The default value is the current tool radius (Maximum V.G.WZ_AKT.R is recommended to avoid residual material)
@P15	Finishing allowance at base Default value = 0
@P16	Finishing allowance at edge (only when tool radius compensation is active @P75) Default value = 0
@P20	Feed rate for roughing for milling in the XY plane (specified in mm/min, inch/min)
@P21	Feed rate for infeed in Z (values in mm/min, inch/min)

@P22	Feed rate for finishing for milling in the XY plane (specified in mm/min, inch/min) The default value is the feed rate for roughing
@P23	Feed rate for finishing for infeed in Z (specified in mm/min, inch/min) The default value is the feed rate for finishing for milling in the XY plane
@P25	Spindle speed for finishing
@P30	Machining direction "ClimbMilling" or 0: Down milling (default) "UpCutMilling" or 1: Up milling
@P31	Machining mode: "Roughing" or 1: Roughing "Finishing" or 2: Finishing "Roughing+Finishing" or 3: Roughing and finishing (default value)
@P32	Feed mode in Z 1: Vertical at start and end of machining movement (default value) 2: In space during approach and withdrawal movements
@P75	Tool radius compensation "TrcOff" or 0: Off (default value) "TrcLeft" or 1: Left of path contour "TrcRight" or 2: Right of path contour
@P90	Approach movement "MotionNone" or 0: No approach movement (default value) "MotionLinear" or 1: Straight line: "MotionQuarterCircle" or 2: Quadrant "MotionHalfCircle" or 3: Semicircle
@P91	Length (straight line) / radius (quadrant/semicircle) of the approach movement Default value = 0
@P92	Withdrawal movement "MotionNone" or 0: No withdrawal movement (default value) "MotionLinear" or 1: Straight line: "MotionQuarterCircle" or 2: Quadrant "MotionHalfCircle" or 3: Semicircle
@P93	Length (straight line) / radius (quadrant/semicircle) of the withdrawal movement Default value = 0
@P95	Additional roughing allowance (results in removal of the contour path multiple times) Default value = 0
@P135	Maximum infeed in XY for edge finishing The default value is the current tool radius

Most of the parameter functions which mainly refer to the Z axis are displayed in the graphic below:

It is recommended to use the Syntax check.to verify whether the input parameters are correctly assigned.

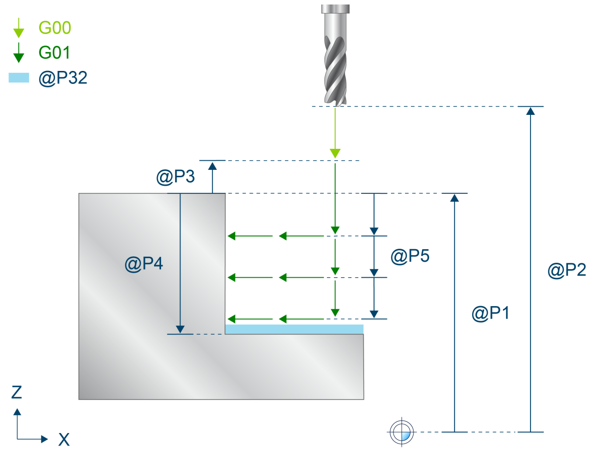

Fig. 84: SysMillContourPath - 2D view

5.4.3 Syntax

L CYCLE [NAME = SysMillContourPath.ecy @P.. = ..]

5.4.4 Programming example

Programing Example

Path milling

This example describes path milling of a contour path defined in the NC program.

Here, a tool radius compensation on the left-hand side is defined. A quadrant was also defined as the approach movement and this is used to feed the tool in Z during the approach.

```
T12 D12                     ( Tool data )
M6                          ( Tool change )
G90 G54 S6000 M03 F5000     ( Technology data )

#VAR
  V.L.SurfacePosition = 0   ( Z position of workpiece surface )
  V.L.RetractionPlane = 10  ( Z position of retraction plane )
  V.L.SafetyClearance = 10  ( relative value of safety clearance )
  V.L.Depth           = 20  ( depth of path milling )
  V.L.MaxIncrementZ   = V.L.Depth/2   ( maximum infeed in Z )
  V.L.FeedRateZ       = 100 ( plunging feedrate in Z )
  V.L.PlungingModeZ   = 1   ( plunging during approaching )
  V.L.TrcMode         = 2   ( trc on left side )
  V.L.ApprRad         = 10  ( approaching with radius 10 )
  V.L.ContourID       = 1   ( 50 identification number pocket contour )
#ENDVAR

; contour definition:
#CONTOUR BEGIN[ID = 1]
G1 G90 X25 Y25  (start point)
G1 G91 X50
G302 I12.5
G1 G91 Y50
G301 I12.5 J12.5
G1 G91 X50
```

```
G02 G91 X25 Y-25 I0 J-25
#CONTOUR END

G0 ZV.L.RetractionPlane

; polynomial contouring for smooth movements
#CONTOUR MODE [DEV, PATH_DEV = V.G.WZ_AKT.R / 100]
G261

L CYCLE [NAME = SysMillContourPath.ecy  \
    @P1  = V.L.SurfacePosition          \
    @P2  = V.L.RetractionPlane          \
    @P3  = V.L.SafetyClearance          \
    @P4  = V.L.Depth                    \
    @P5  = V.L.MaxIncrementZ            \
    @P21 = V.L.FeedRateZ                \
    @P31 = "Roughing"                   \
    @P32 = V.L.PlungingModeZ            \
    @P75 = V.L.TrcMode                  \
    @P90 = "MotionLinear"               \
    @P91 = V.L.ApprRad                  \
    @P50 = V.L.ContourID                \
    ]

G260
M30
```

5.5 SysMillContourResidual - Remove residual material with contour milling

This cycle is used to remove residual material during contour milling. The cycle can be used both in combination with the pocket milling cycle (reference cycle = SysMillContourPocket) and with the spigot milling cycle (reference cycle = SysMillContourSpigot).

When residual material is removed, it is assumed that the contour was already machined in a first step with the reference cycle. When residual material is removed, a smaller tool radius can now be used to remove the residual material that could not be removed in the first pass due to the size of the previous tool radius (@P36) without having to re-mill the entire contour. The finishing allowance is not considered as residual material.

For example, a typical process that results here when the contour pocket milling cycle is used can look as follows:

- Predrilling (SysMillContourPreDrill.ecy)
- Mill contour pocket with the reference cycle - Roughing (SysMillContourPocket.ecy)
- Tool change
- Remove residual material (SysMillContourResidual.ecy)
- Mill contour pocket - Finishing (SysMillContourPocket.ecy)

If several contour pockets are to be milled with the same tool radius, it may be practical to first mill the two pockets before removing the residual material. This avoids unnecessary tool changes.

- Mill contour pocket 1 with the reference cycle - Roughing (SysMillContourPocket.ecy)
- Mill contour pocket 2 with the reference cycle - Roughing (SysMillContourPocket.ecy)
- Tool change
- Remove residual material pocket 1 (SysMillContourResidual.ecy)
- Mill contour pocket 1 - Finishing (SysMillContourPocket.ecy)
- Remove residual material pocket 2 (SysMillContourResidual.ecy)
- Mill contour pocket 2 - Finishing (SysMillContourPocket.ecy)

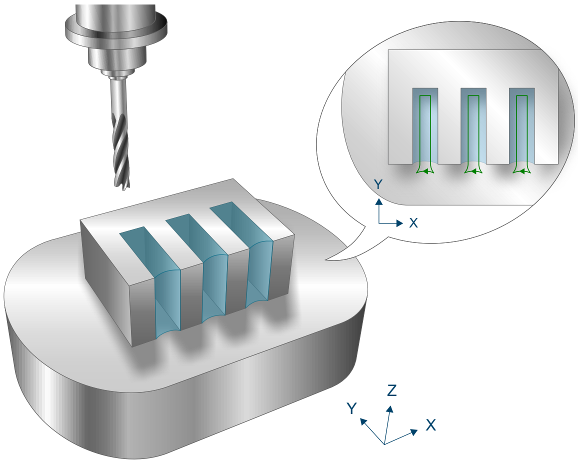

Fig. 85: SysMillContourResidual

5.5.1 Process

To ensure that the cycle is executed successfully, the following requirements must be met.

- The tool radius is defined by V.G.WZ_AKT.R
- Tool length compensation is active.
- The spindle speed must be selected before the cycle is started.

- The contour pocket can be reached without collision starting from the current position at the height of the retraction plane.
- The contours used were already programmed using #CONTOUR BEGIN / #CONTOUR END.
- The defined contours for the edge contour and islands are closed, i.e. start and end points are identical.

In the cycle, the tool is fed vertically along the Z axis, if possible slightly offset from the residual material in order to allow increased feedrates. The tool then plunges into the residual material with its full circumference. Once the centre of the residual material is reached, it is removed parallel to the contour. Infeed in the XY plane is specified by the input parameters. However, it is internally limited to the tool radius to avoid residual material.

If there is residual material in several places, the tool is retracted several times along Z and then re-approached. If a slightly offset infeed to the residual material is not possible, helical plunging is used provided it was previously activated by the input parameters.

If individual contour elements were specified with a feedrate during contour definition, this has no influence on the milling feedrate.

Attention

For contour spigot milling, it is possible that the maximum infeed in the XY plane is violated and the milling cutter enters the material at full circumference. This must be considered when the milling feed is selected.

5.5.2 Parameterisation

Required input parameters

Input parameters	Description
@P1	Reference plane (absolute)
@P2	Retraction plane (absolute)
@P3	Safety clearance in Z (relative to reference plane/machining height)
@P4	Depth (relative to reference plane)
@P5	Maximum infeed in Z
@P6	Maximum infeed in XY (Maximum V.G.WZ_AKT.R is recommended to avoid residual material)
@P35	Reference cycle 1: SysMillContourPocket.ecy 2: SysMillContourSpigot.ecy
@P36	Tool radius used in the reference cycle
@P50	Identification number of the pocket or blank contour

Optional input parameters

Input parameters	Description
@P7	Pitch of plunge helix Default value = 10% of tool radius
@P8	Radius of plunge helix Default value = 50% of tool radius
@P15	Finishing allowance at base in reference cycle Default value = 0
@P16	Finishing allowance at edge in reference cycle Default value = 0
@P20	Feed rate for roughing for milling in the XY plane (specified in mm/min, inch/min)
@P21	Feed rate for roughing in Z (specified in mm/min, inch/min)
@P30	Machining direction "ClimbMilling" or 0: Down milling (default) "UpCutMilling" or 1: Up milling
@P32	Feed mode in Z 1: Vertical (default value) 2: Helical
@P51	Identification number of an island contour
@P52	Identification number of an island contour
@P53	Identification number of an island contour

@P54	Identification number of an island contour
@P55	Identification number of an island contour
@P56	Identification number of an island contour
@P57	Identification number of an island contour
@P58	Identification number of an island contour
@P59	Identification number of an island contour
@P60	Identification number of an island contour

Parameters which mainly refer to the Z axis are displayed in the graphic below:

It is recommended to use the Syntax check.to verify whether the input parameters are correctly assigned.

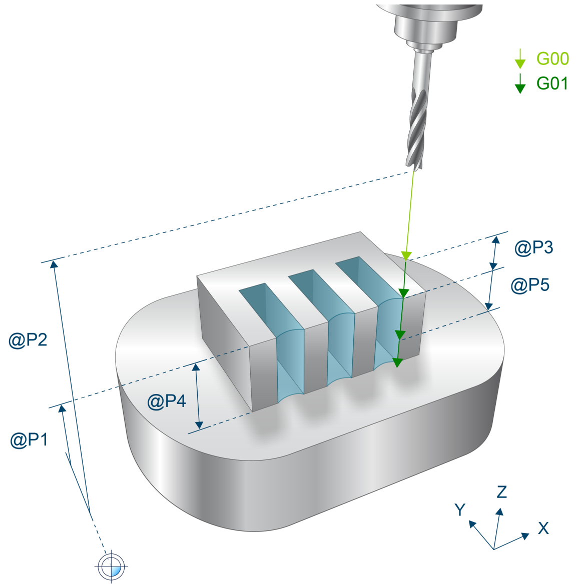

Fig. 86: Parameterisation

5.5.3 Syntax

L CYCLE [NAME = SysMillContourResidual.ecy @P.. = ..]

5.5.4 Programming example

Programing Example

Remove residual material

```
T20 D20                             ( Tool data )
M6                                  ( Tool change )

G90 G54 S6000 M03 F5000             ( Technology data )
G00 Z100                            ( Go to z start position )

#VAR
  V.L.SurfacePosition      = 0    ( Z position of workpiece surface )
  V.L.RetractionPlane      = 50   ( Z position of retraction plane )
  V.L.SafetyClearance      = 2    ( relative value of safety clearance )
  V.L.Depth                = 10   ( depth of pocket )
  V.L.MaxIncrementZ        = V.L.Depth  ( maximum infeed in Z )
  V.L.FeedRateZ            = 80   ( plunging feedrate )
  V.L.MaxIncrementXY       = 2.2  ( maximum infeed in XY )
  V.L.ReferenceCycle       = 1    ( reference cycle (1/2) )
  V.L.RadiusReferenceCycle = 4    ( radius of tool in reference cycle )
  V.L.ContourID            = 1    ( identification number pocket contour)
  V.L.FinishingOffsetXY    = 0.2  ( finishing offset XY (default = 0) )
  V.L.IsleID1              = 2    ( identification number isle 1 )
#ENDVAR

; circular pocket definition:
#CONTOUR BEGIN[ID = 1]
G1 G90 X30 Y0
G02 G161 X-30 Y0 I0 J0
G02 G161 X30 Y0 I0 J0
#CONTOUR END

; island (with residual material) definition:
#CONTOUR BEGIN[ID = 2]
G1 G90 X-10 Y-10
G1 X10 Y-10
G1 X10 Y-3
G1 X-5 Y-3
G1 X-5 Y3
G1 X10 Y3
G1 X10 Y10
G1 X-10 Y10
G1 X-10 Y-10
#CONTOUR END

G0 ZV.L.RetractionPlane

; polynomial contouring for smooth movements
#CONTOUR MODE [DEV, PATH_DEV = V.G.WZ_AKT.R / 100]
G261

; pocket milling cycle call:
L CYCLE [NAME = SysMillContourPocket.ecy   \
    @P1   = V.L.SurfacePosition            \
    @P2   = V.L.RetractionPlane            \
    @P3   = V.L.SafetyClearance            \
```

```
    @P4  = V.L.Depth                       \
    @P5  = V.L.MaxIncrementZ               \
    @P6  = V.L.MaxIncrementXY              \
    @P16 = V.L.FinishingOffsetXY           \
    @P21 = V.L.FeedRateZ                   \
    @P31 = "Roughing"                      \
    @P50 = V.L.ContourID                   \
    @P51 = V.L.IsleID1                     \
    ]

T13 D13                                   ( Tool data )
M6                                        ( Tool change )
G00 G90 M03 S8000 F5000                   ( Technology data )

; residual milling cycle call:

V.L.MaxIncrementXY = V.G.WZ_AKT.R         ( adapt increment to new tool )
V.L.MaxIncrementZ  = 3                    ( maximal infeed in Z )

L CYCLE [NAME = SysMillContourResidual.ecy  \
    @P1  = V.L.SurfacePosition              \
    @P2  = V.L.RetractionPlane              \
    @P3  = V.L.SafetyClearance              \
    @P4  = V.L.Depth                        \
    @P5  = V.L.MaxIncrementZ                \
    @P6  = V.L.MaxIncrementXY               \
    @P16 = V.L.FinishingOffsetXY            \
    @P21 = V.L.FeedRateZ                    \
    @P35 = V.L.ReferenceCycle               \
    @P36 = V.L.RadiusReferenceCycle         \
    @P50 = V.L.ContourID                    \
    @P51 = V.L.IsleID1                      \
    ]

G260
M05
M30
```

5.6 SysMillContourPreDrilling - Predrilling with contour milling

This cycle is used to predrill for contour milling.

The cycle can be used both in combination with the pocket milling cycle (reference cycle = SysMillContourPocket) and with the spigot milling cycle (reference cycle = SysMillContourSpigot). The plunge points required for roughing are considered for predrilling.

For example, a typical process that results here when the contour pocket milling cycle is used can look as follows:

- Predrilling (SysMillContourPreDrilling.ecy)
- Tool change
- Mill contour pocket (= SysMillContourPocket.ecy)

If several contours are to be milled, it may be practical to make multiple calls of the predrilling cycle before the contour is machined. This avoids unnecessary tool changes.

- Predrill contour pocket 1 (SysMillContourPreDrilling.ecy)
- Predrill contour pocket 2 (SysMillContourPreDrilling.ecy)
- Tool change
- Mill contour pocket 1 (= SysMillContourPocket.ecy)
- Mill contour pocket 2 (= SysMillContourPocket.ecy)

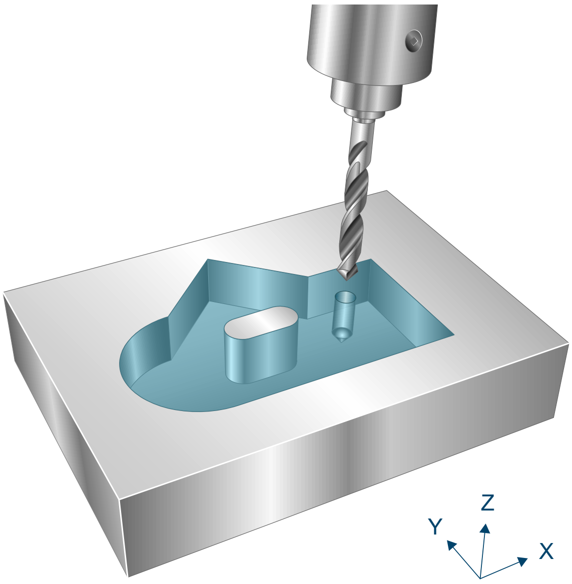

Fig. 87: SysMillContourPreDrilling

5.6.1 Process

To ensure that the cycle is executed successfully, the following requirements must be met.

- Tool length compensation is active.
- The spindle speed must be selected before the cycle is started.
- The contour can be reached without collision starting from the current position at the height of the retraction plane.
- The contours used were already programmed using #CONTOUR BEGIN / #CONTOUR END.
- The defined contours for the edge contour and islands are closed, i.e. start and end points are identical.

The necessary plunge positions in the cycle are calculated and approached one after the other on the retraction plane. For each plunge position, the tool drills to the specified machining depth in multi-step operations at the selected spindle speed and feedrate G01. The number of infeeds is programmable. The resulting feed depth is determined internally in the cycle based on the final drilling depth. A different feed depth can be defined for the first infeed.

5.6.2 Parameterisation

Required input parameters

Input parameters	Description
@P1	Reference plane (absolute)
@P2	Retraction plane (absolute)
@P3	Safety clearance in Z (relative to reference plane/machining height)
@P4	Pocket depth (relative to reference plane)
@P6	Maximum infeed in XY in reference cycle (Maximum V.G.WZ_AKT.R)
@P11	Number of infeeds (unsigned) Maximum infeed depth is calculated from [depth (@P4) / number of infeeds (@P11)]
@P31	Machining mode: 1 = chip breaking 2 = chip evacuation
@P35	Reference cycle 1: SysMillContourPocket.ecy 2: SysMillContourSpigot.ecy
@P61	Return clearance for chip breaking and holding distance after chip evacuation (unsigned)
@P130	Milling cutter radius in reference cycle
@P50	Identification number of the pocket or blank contour for reference cycle

Optional input parameters

Input parameters	Description
@P16	Finishing allowance at edge in reference cycle Default value = 0
@P30	Machining direction for reference cycles "ClimbMilling" or 0: Down milling (default) "UpCutMilling" or 1: Up milling
@P92	Retraction mode: 0 = Retract at rapid traverse (default) 1 = Retract at feed rate
@P131	Drilling depth Default value = pocket depth

@P132	Feed rate for drilling (values in mm/min, inch/min) Default value = active feed rate
@P133	Direction of spindle rotation for reference cycle 3: Clockwise (default) 4: Counter-clockwise
@P51	Identification number of an island contour for reference cycle
@P52	Identification number of an island contour for reference cycle
@P53	Identification number of an island contour for reference cycle
@P54	Identification number of an island contour for reference cycle
@P55	Identification number of an island contour for reference cycle
@P56	Identification number of an island contour for reference cycle
@P57	Identification number of an island contour for reference cycle
@P58	Identification number of an island contour for reference cycle
@P59	Identification number of an island contour for reference cycle
@P60	Identification number of an island contour for reference cycle

It is recommended to use the Syntax check.to verify whether the input parameters are correctly assigned.

5.6.3 Syntax

L CYCLE [NAME = SysMillContourPreDrilling.ecy @P.. = ..]

5.6.4 Programming example

Programing Example

Predrill contour

This example describes predrilling of a contour pocket defined in the NC program.

The first cycle call activates predrilling. In addition to the feedrate for drilling, the parameters used for pocket milling must be transferred to this cycle in order to calculate the plunge points of the milling cycle.

After predrilling, a tool change takes place before the contour pocket is milled.

```
#VAR
  V.L.SurfacePosition = 0         ( Z position of workpiece surface )
  V.L.RetractionPlane = 100       ( Z position of retraction plane )
  V.L.SafetyClearance = 10        ( relative value of safety clearance )

  ; Parameter pre drilling cycle   (SysMillContourPreDrilling)
  V.L.DrillFeed              = 200  ( drilling feedrate in Z )
  V.L.RadiusReferenceCycle = 2.5  ( tool radius of reference cycle (D5))
  V.L.ReferenceCycle         = 1    ( reference cycle = pocket milling )
  V.L.NumberOfFeeds          = 3    ( number of feeds )
  V.L.DrillMachiningMode     = 2    ( drill machining mode )
```

```
  V.L.ReturnClearance      = .5   ( return clr. for chip breaking/evac.)

  ; Parameter reference cycle     (SysMillContourPocket)
  V.L.FeedRateZ       = 1000      ( plunging feedrate )
  V.L.Direction       = 0         ( up cut milling )
  V.L.DepthOfPocket   = 20        ( depth of pocket )
  V.L.MaxIncrementZ   = 10        ( maximum infeed in Z )
  V.L.MaxIncrementXY  = 2.2       ( maximum infeed in XY )
  V.L.PlungingModeZ   = 1         ( plunging mode 1 )
  V.L.ContourID       = 1         (identification number pocket contour)
#ENDVAR

G0 G90 ZV.L.RetractionPlane F2000

; contour definition:
#CONTOUR BEGIN[ID = 1]
G1 G90 X60 Y0
G1 G91 Y40
G1 G91 X-20
G1 G90 X20 Y60
G1 X0 Y40
G161 G03 X0 Y0 I0 J20
G1 X60
#CONTOUR END

T6 D6                             ( Tool data )
M6                                ( Tool change )
G90 G54 S1000 M03 F5000       ( Technology data )

G00 Z100                          ( Go to z start position )
G00 X0 Y0                         ( Go to centre of circle )

; pre drilling call:
L CYCLE [NAME = SysMillContourPreDrilling.ecy    \
        @P1   = V.L.SurfacePosition              \
        @P2   = V.L.RetractionPlane              \
        @P3   = V.L.SafetyClearance              \
        @P4   = V.L.DepthOfPocket                \
        @P6   = V.L.MaxIncrementXY               \
        @P11  = V.L.NumberOfFeeds                \
        @P31  = V.L.DrillMachiningMode           \
        @P35  = V.L.ReferenceCycle               \
        @P61  = V.L.ReturnClearance              \
        @P130 = V.L.RadiusReferenceCycle         \
        @P132 = V.L.DrillFeed                    \
        @P50  = V.L.ContourID                    \
        ]

T8 D8                             ( Tool data, radius = 2.5 )
M6                                ( Tool change )
G90 S6000 M03 F5000               ( Technology data )

; polynomial contouring for smooth movements
#CONTOUR MODE [DEV, PATH_DEV = V.G.WZ_AKT.R / 100]
G261

; reference cycle call:
L CYCLE [NAME = SysMillContourPocket.ecy         \
        @P1  = V.L.SurfacePosition               \
        @P2  = V.L.RetractionPlane               \
        @P3  = V.L.SafetyClearance               \
        @P4  = V.L.DepthOfPocket                 \
        @P5  = V.L.MaxIncrementZ                 \
```

```
        @P6  = V.L.MaxIncrementXY                \
        @P21 = V.L.FeedRateZ                     \
        @P31 = "Roughing"                        \
        @P32 = V.L.PlungingModeZ                 \
        @P50 = V.L.ContourID                     \
        ]

G260
M30
```

6 Engraving

Notice

This function is available as of CNC Build V3.01.3079.12.

6.1 Overview

The engraving cycle is an aid to milling a text in the workpiece.

Possible applications

When engraving texts, variable values such as times and quantities can be included in the text.

The following additional modifications are possible:

- Orienting the text on a circle with variable opening angle
- Adjusting font size and total text width
- Text mirroring
- Modifying the engraving depth
- etc.

6.2 SysMillEngrave cycle - Engrave text

6.2.1 Process

Enter the parameters for workpiece surface, engraving depth, feedrates, text, font size etc. The cycle then engraves the specified text onto the workpiece.

Programing Example

Engrave call sequence

```
; Call engrave cycle
L CYCLE [NAME  = "SysMillEngrave.ecy"                    \
          @P1  = 0                ( workpiece surface ) \
          @P2  = 2                ( retraction plane       ) \
          @P3  = 1                ( safety height   ) \
          @P4  = 0.2              ( engraving depth   ) \
          @P11 = 2000             ( feedrate infeed ) \
          @P12 = 1000             ( engraving feedrate ) \
          @P21 = "ISG kernel"     ( Text                 ) \
          @P53 = 2                ( font size         ) \
        ]
```

This cycle call engraves the text “ISG kernel” on the workpiece surface. The line width of the letters is given by the radius of the tool used.

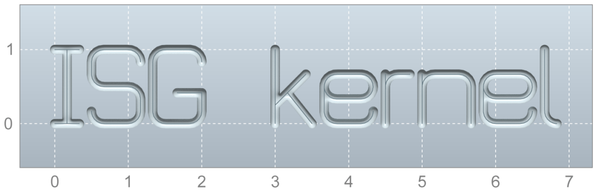

Fig. 88: A simple script.

A maximum of 4 systems may be active or defined when the cycle is called to ensure that sufficient machining coordinate systems (#CS) are available for the cycle.

6.2.2 Parameters

Required input parameters

Input parameters	Description
@P1	Workpiece surface, Z position (absolute)
@P2	Retraction plane, Z position (absolute)
@P3	Safety clearance above workpiece (unsigned), Z distance (relative to the reference plane/ machining height)
@P4	Engraving depth in workpiece (unsigned), Z distance (relative to the reference plane)
@P21	to engraving text
@P53	Font size

Optional input parameters

Input parameters	Description
@P11	Feed rate for Z infeeds, F word
@P12	Feed rate for engraving movements, F word
@P13	Subroutine executed at start of engraving
@P14	Subroutine executed at end of engraving
@P15	Subroutine executed at start of each character
@P16	Subroutine executed at end of each character
@P17	Subroutine executed at start of each path
@P18	Subroutine executed at end of each path
@P23	Invert text (default value = `FALSE`)
@P24	Use placeholders (default value = `FALSE`), see also `@P101` to `@P103`
@P31	Arrangement (default value = 0) 0 = rectangle 1 = arc, circle centre point below text 2 = arc, circle centre point above text
@P32	Arrangement parameter 1 if `@P31` = 1 or 2, then X position of circle centre point

@P33	Arrangement parameter 2 if `@P31` = 1 or 2, then Y position of circle centre point
@P41	Reference point (default value = 6) 0 = top left 1 = top centre 2 = top right 3 = centre left 4 = centre centre 5 = centre right 6 = baseline left 7 = baseline centre 8 = baseline right 9 = bottom left 10 = bottom centre 11 = bottom right
@P42	Reference point, X position (default value = `V.A.ACT_POS.X` = current position)
@P43	Reference point, Y position (default value = `V.A.ACT_POS.Y` = current position)
@P52	Determining font size (default value = 0) 0 = height of letter "X" 1 = width of letter "X" 2 = height of entire text 3 = width of entire text
@P54	Font weight (default = calculation from engraving depth) Can be specified instead of the engraving depth together with the point angle of the engraving tool.
@P55	Point angle of the engraving tool Can be specified instead of the engraving depth together with the font weight.
@P61	non-proportional font (default text = `FALSE`)
@P62	mirror each letter in X direction (default value = `FALSE`)
@P63	mirror each letter in Y direction (default value = `FALSE`)
@P64	additional X offset for each letter (default value = 0)
@P65	additional Y offset for each letter (default value = 0)
@P71	Type of output (default value = 0) 0 = G code, machine is moved 1 = no output, for calculations, size requests, etc.
@P101	free parameter
@P102	free parameter
@P103	free parameter

It is recommended to use the Syntax check.to verify whether the input parameters are correctly assigned.

Encoding the NC program

The text “ISG kernel” consists of only ASCII characters. Nothing else needs to be considered when encoding the NC program that calls up the engraving cycle.

However, if you want to use diacritical marks or other special characters, the NC program must be saved in the Windows-1252 encoding.

Programing Example

Engraving special characters

```
; call engraving cycle for text with special characters
; !!Save the NC program in Windows-1252 encoding!!
L CYCLE [NAME  = "SysMillEngrave.ecy"                        \
          @P1  = 0                                            \
          @P2  = 2                                            \
          @P3  = 1                                            \
          @P4  = 0.2                                          \
          @P11 = 2000                                         \
          @P12 = 1000                                         \
          @P21 = "Brühe €1.95"  ( Text with special characters ) \
          @P53 = 2                                            \
        ]
M30
```

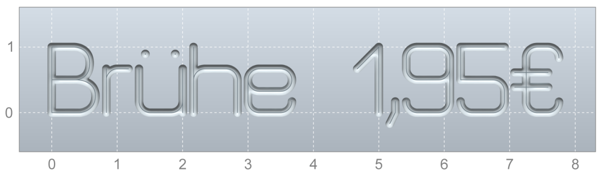

Fig. 89: Use of diacritical marks and special characters

The supplied standard font supports all printable characters in the Windows-1252 character set.

Subroutines

The parameters `@P13` to `@P18` can specify subroutines that execute specific results.

- Subroutine `@P13` is executed before the start of text engraving.
- Subroutine `@P14` is executed after the end of text engraving.
- Subroutine `@P15` is executed before engraving each character.
- Subroutine `@P16` is executed after engraving each character.
- Subroutine `@P17` is executed each time before leaving the retraction plane to start a new milling section.
- Subroutine `@P18` is executed each time after a milling section ends and the retraction plane is reached.

Subroutines enable various technologies during engraving,

e.g. the engraving cycle can be used for milling machines as well as for laser or plasma cutting machines. Subroutines `@P17` and `@P18` could enable or disable the laser.

Invert text

If `@P23` is set to `TRUE`, the input text is output inverted.

Programing Example

Invert text

```
; call engraving cycle for inverted text
L CYCLE [NAME  = "SysMillEngrave.ecy"            \
         @P1   = 0                               \
         @P2   = 2                               \
         @P3   = 1                               \
         @P4   = 0.2                             \
         @P11  = 2000                            \
         @P12  = 1000                            \
         @P21  = "ISG kernel"                    \
         @P23  = TRUE            ( invert text ) \
         @P53  = 2                               \
        ]
M30
```

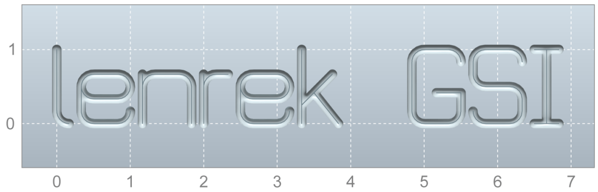

Fig. 90: Inverted text

Placeholders

If `@P24` is set to `TRUE`, placeholders can be used in the transferred text. For example, to engrave the current date, the placeholder `${date}` can be used:

Programing Example

Placeholders

```
; use placeholders
L CYCLE [NAME  = "SysMillEngrave.ecy"                          \
         @P1   = 0                                             \
         @P2   = 2                                             \
         @P3   = 1                                             \
         @P4   = 0.2                                           \
         @P11  = 2000                                          \
         @P12  = 1000                                          \
         @P21  = "heute ${date}"  ( engrave current date ) \
         @P24  = TRUE              ( use placeholders     ) \
         @P53  = 2                                             \
        ]
M30
```

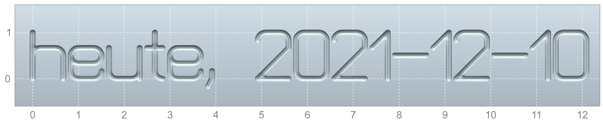

Fig. 91: Using a placeholder

Placeholders supported include

Placeholders	Example	Description
`${date}`	03/12/2021	current date, YYYY-MM-DD
`${time}`	17:15:43	current time, hh:mm:ss
`${now}`	2021-12-03 17:15:43	current date and time, YYYY-MM-DD hh:mm:ss
`${year}`	2021	current year, YYYY
`${month}`	12	current month, MM
`${day}`	03	current day, DD
`${hour}`	17	current hour, hh
`${min}`	15	current minute, mm
`${sec}`	43	current second, ss
`${file}`	SysMillEngraveDemo.nc	Filename of the calling NC main program
`${prog}`	EngraveDemo	Name (%…) of the calling NC main program
`${1}`	11	Value of cycle parameter `@ P101`
`${2}`	22	Value of cycle parameter `@ P102`
`${3}`	33	Value of cycle parameter `@ P103`

Programing Example

Placeholders

```
; use placeholders for free parameters
L CYCLE [NAME   = "SysMillEngrave.ecy"                \
          @P1    = 0                                   \
          @P2    = 2                                   \
          @P3    = 1                                   \
          @P4    = 0.2                                 \
          @P11   = 2000                                \
          @P12   = 1000                                \
          @P21   = "${1} + ${2} = ${3}"                \
          @P24   = TRUE      ( use placeholders )  \
          @P53   = 1                                   \
          @P101  = 11        ( free parameter      ) \
          @P102  = 22        ( free parameter      ) \
          @P103  = 33        ( free parameter      ) \
        ]
M30
```

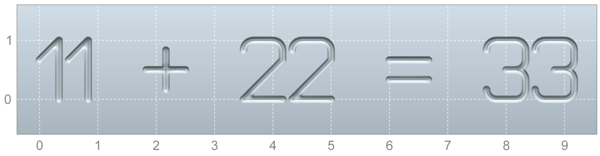

Fig. 92: Use placeholders for free parameters

Layout and reference point

The parameter `@P31` determines the layout of the engraved text.

Linear (default value)

By default `@P31 = 0` the text is output in one line in the usual form. The surrounding character box is positioned by a reference point. The following intersections are possible as a reference point.

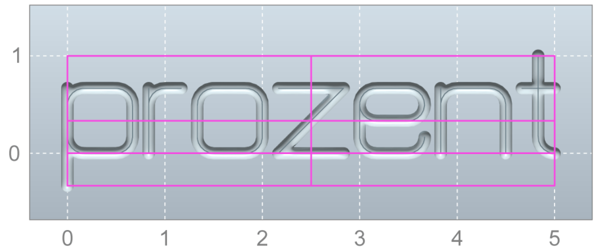

Fig. 93: Character box with auxiliary lines

The reference point is specified by the parameter `@P41` and can assume the following values.

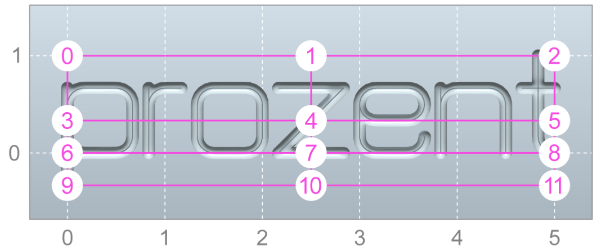

Fig. 94: Possible reference points

The reference point coordinates `(X, Y)` are specified by `(@P42, @P43)`. By default these parameters are pre-assigned the current axis positions.

Programing Example

Engrave linear

```
; reference point (centre, centre) where (1, -0.25)
L CYCLE [NAME  = "SysMillEngrave.ecy"                           \
         @P1   = 0                                              \
         @P2   = 2                                              \
         @P3   = 1                                              \
         @P4   = 0.2                                            \
         @P11  = 2000                                           \
         @P12  = 1000                                           \
         @P21  = "ISG kernel"                                   \
         @P41  = 4           ( reference point, centre, centre ) \
         @P42  = 1           ( reference point X coordinate ) \
         @P43  = -0.25       ( reference point Y coordinate ) \
```

```
          @P53 = 2                                            \
        ]
M30
```

Fig. 95: Example of reference point (centre, centre) where (1, -0.25)

Arc

Where `@P31 = 1` and `@P31 = 2`, the characters can be arranged in the form of an arc.

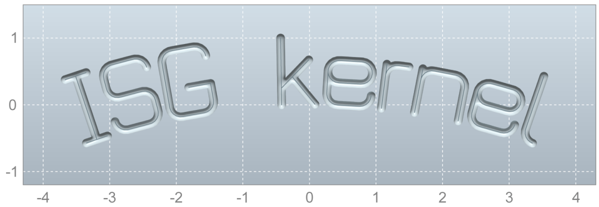

Fig. 96: Example of an arc layout, circle centre point is below the text

In addition to the reference point `R`, the parameters `(@P32, @P33)` specify the circle centre point `M`.

Programing Example

Engrave in arc

```
; reference point (centre, bottom) and circle centre point (0, -2)
L CYCLE [NAME  = "SysMillEngrave.ecy"                         \
          @P1  = 0                                            \
          @P2  = 2                                            \
          @P3  = 1                                            \
          @P4  = 0.2                                          \
          @P11 = 2000                                         \
          @P12 = 1000                                         \
          @P21 = "abcde"                                      \
          @P31 = 1            ( circle layout              ) \
          @P32 = 0            ( circle centre point X       ) \
          @P33 = -2           ( circle centre point Y       ) \
          @P41 = 10           ( reference point centre, bottom ) \
          @P42 = 0            ( reference point X           ) \
          @P43 = 0            ( reference point Y           ) \
          @P53 = 2                                            \
        ]
M30
```

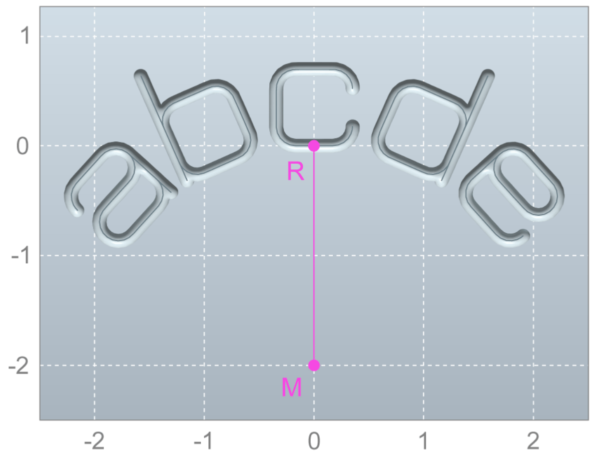

Fig. 97: Reference point (centre, bottom) and circle centre point (0, -2)

When reference point and circle centre point are identical, an error is output.

Example: varied circle centre point

Programing Example

Engrave in arc

```
; reference point (centre, bottom) and circle centre point (1, -1)
L CYCLE [NAME  = "SysMillEngrave.ecy",                              \
         @P1   = 0                                                  \
         @P2   = 2                                                  \
         @P3   = 1                                                  \
         @P4   = 0.2                                                \
         @P11  = 2000                                               \
         @P12  = 1000                                               \
         @P21  = "abcde"                                            \
         @P31  = 1              ( circle layout                ) \
         @P32  = 1              ( circle centre point X         ) \
         @P33  = -1             ( circle centre point Y         ) \
         @P41  = 10             ( reference point centre, bottom ) \
         @P42  = 0              ( reference point X             ) \
         @P43  = 0              ( reference point Y             ) \
         @P53  = 2                                                  \
        ]
M30
```

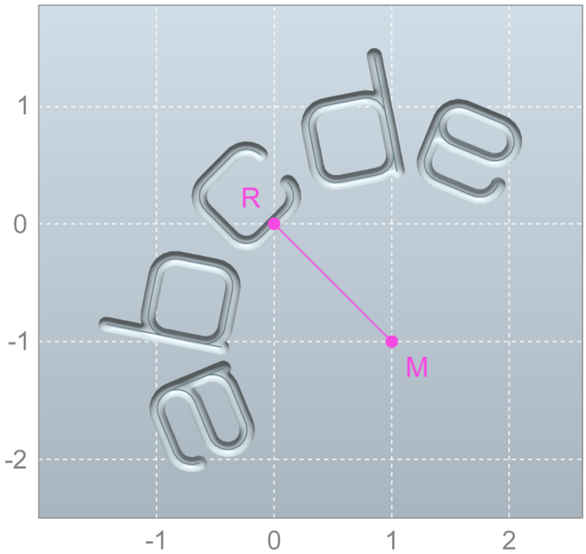

Fig. 98: Reference point (centre, bottom) and circle centre point (1, -1)

Example: Reference point (left, bottom)

Programing Example

Reference point (left, bottom)

```
; reference point (left, bottom) and circle centre point (0, -2)
L CYCLE [NAME  = "SysMillEngrave.ecy"                              \
         @P1   = 0                                                 \
         @P2   = 2                                                 \
         @P3   = 1                                                 \
         @P4   = 0.2                                               \
         @P11  = 2000                                              \
         @P12  = 1000                                              \
         @P21  = "abcde"                                           \
         @P31  = 1              ( circle layout                 ) \
         @P32  = 0              ( circle centre point X          ) \
         @P33  = -2             ( circle centre point Y          ) \
         @P41  = 9              ( reference point left, bottom ) \
         @P42  = 0              ( reference point X              ) \
         @P43  = 0              ( reference point Y              ) \
         @P53  = 2                                                 \
        ]
M30
```

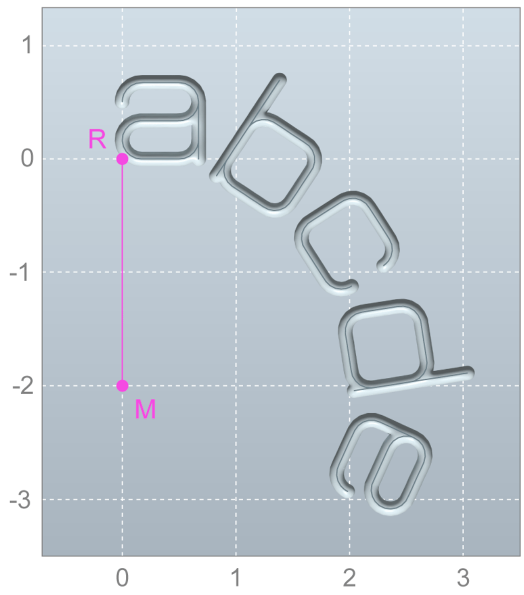

Fig. 99: Reference point (left, bottom) and circle centre point (0, -2)

Example: Circle centre point is above text

In the above examples, the circle centre point was always below the text. Therefore, the text start and end were both with “curved downwards”. Use `@P31 = 2` to achieve the opposite effect. The circle centre point is above the text and the text ends are “curved upwards”.

Programing Example

Circle centre point above text

```
; circle centre point is above text
L CYCLE [NAME  = "SysMillEngrave.ecy"                          \
          @P1  = 0                                             \
          @P2  = 2                                             \
          @P3  = 1                                             \
          @P4  = 0.2                                           \
          @P11 = 2000                                          \
          @P12 = 1000                                          \
          @P21 = "abcde"                                       \
          @P31 = 2              ( circle layout 2            ) \
          @P32 = 0              ( circle centre point X     ) \
          @P33 = 2             ( circle centre point Y     ) \
          @P41 = 1              ( reference point, centre, top ) \
          @P42 = 0              ( reference point X            ) \
          @P43 = 0              ( reference point Y            ) \
          @P53 = 2                                             \
        ]
M30
```

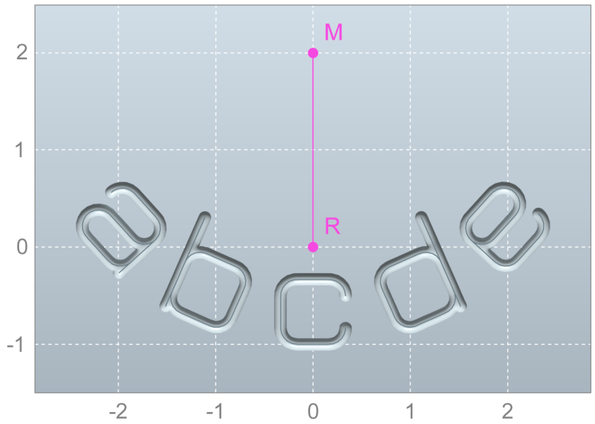

Fig. 100: Circle centre point is above text

Font size

The font size can either be specified directly or derived from another size. The parameter @P52 determines how the font size is calculated. The following options are available.

Value of @P52	Meaning
0 (default)	height of letter "X"
1	width of letter "X"
2	height of entire text (in linear layout)
3	width of entire text (in linear layout)

The corresponding size is then specified in @P53.

Example: height of letter "X"

Programing Example

height of letter X

```
; font size via the height of "X"
L CYCLE [NAME   = "SysMillEngrave.ecy"            \
         @P1    = 0                               \
         @P2    = 2                               \
         @P3    = 1                               \
         @P4    = 0.2                             \
         @P11 = 2000                              \
         @P12 = 1000                              \
         @P21 = "Xylophone"                      \
         @P52 = 0              ( height of “X”    ) \
         @P53 = 2             ( font size 1 ) \
        ]
M30
```

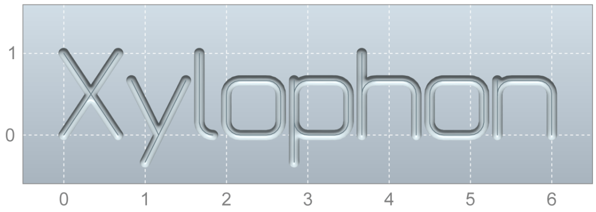

Fig. 101: Font size via the height of "X"

Example: width of letter "X"

Programing Example

width of letter X

```
; font size via the width of "X"
L CYCLE [NAME   = "SysMillEngrave.ecy"            \
          @P1   = 0                               \
          @P2   = 2                               \
          @P3   = 1                               \
          @P4   = 0.2                             \
          @P11  = 2000                            \
          @P12  = 1000                            \
          @P21  = "Xylophone"                    \
          @P52  = 1             ( width of "X"     ) \
          @P53  = 2            ( font size 1 ) \
        ]
M30
```

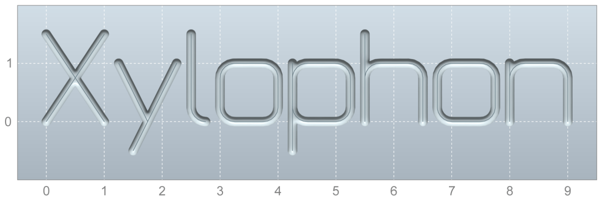

Fig. 102: Font size via the width of "X"

Example: height of entire text

Programing Example

height of entire text

```
; font size via the height of the entire text
L CYCLE [NAME   = "SysMillEngrave.ecy"                      \
          @P1   = 0                                         \
          @P2   = 2                                         \
          @P3   = 1                                         \
          @P4   = 0.2                                       \
          @P11  = 2000                                      \
          @P12  = 1000                                      \
          @P21  = "Xylophone"                                \
          @P52  = 2            ( height of entire text ) \
```

```
          @P53 = 2            ( font size 1 )   \
        ]
M30
```

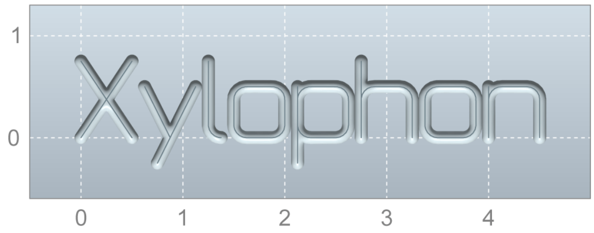

Fig. 103: Font size via the height of the entire text

Example: width of entire text

Programing Example

width of entire text

```
; font size via the width of the entire text
L CYCLE [NAME  = "SysMillEngrave.ecy"                  \
          @P1  = 0                                     \
          @P2  = 2                                     \
          @P3  = 1                                     \
          @P4  = 0.2                                   \
          @P11 = 2000                                  \
          @P12 = 1000                                  \
          @P21 = "Xylophone"                          \
          @P52 = 3            ( width of entire text ) \
          @P53 = 4            ( font size 4 )          \
        ]
M30
```

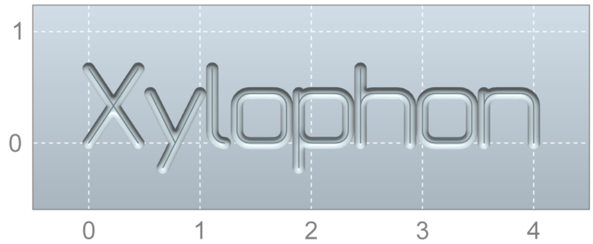

Fig. 104: Font size via the width of the entire text

Non-proportional font

The parameter `@P61` switches over to non-proportional font. In this mode, all characters have the same width.

Programing Example

Non-proportional font

```
; proportional font
L CYCLE [NAME  = "SysMillEngrave.ecy"               \
         @P1   = 0                                  \
         @P2   = 2                                  \
         @P3   = 1                                  \
         @P4   = 0.2                                \
         @P11 = 2000                                \
         @P12 = 1000                                \
         @P21 = "floor"                             \
         @P42 = 0        ( reference point X        ) \
         @P43 = 1.5      ( reference point Y        ) \
         @P53 = 2                                   \
         @P63 = FALSE    ( proportional font ) \
        ]

; non-proportional font
L CYCLE [NAME  = "SysMillEngrave.ecy"                    \
         @P1   = 0                                       \
         @P2   = 2                                       \
         @P3   = 1                                       \
         @P4   = 0.2                                     \
         @P11 = 2000                                     \
         @P12 = 1000                                     \
         @P21 = "floor"                                  \
         @P42 = 0        ( reference point X         ) \
         @P43 = 0        ( reference point Y             ) \
         @P53 = 2                                        \
         @P61 = TRUE    ( non-proportional font ) \
        ]
M30
```

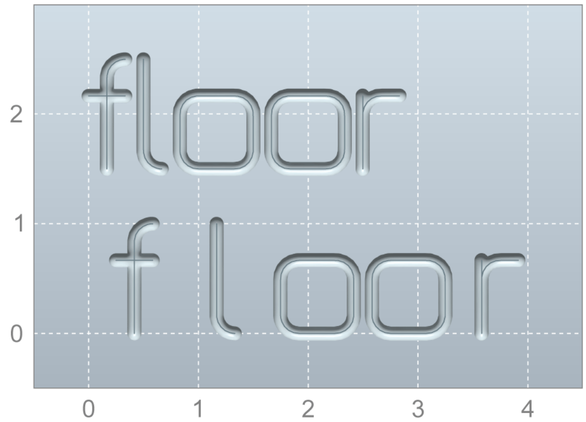

Fig. 105: Comparison between proportional font (top) and non-proportional font (bottom)

Mirror font

The parameters `@P62` and `@P63` can mirror the characters in the X and Y directions.

Example: Mirror in X, character by character

Programing Example

Mirror in X - character by character

```
; mirror in X, character by character
L CYCLE [NAME  = "SysMillEngrave.ecy"                   \
          @P1  = 0                                      \
          @P2  = 2                                      \
          @P3  = 1                                      \
          @P4  = 0.2                                    \
          @P11 = 2000                                   \
          @P12 = 1000                                   \
          @P21 = "ISG kernel"                           \
          @P53 = 2                                      \
          @P62 = TRUE            ( mirror text in X ) \
        ]
M30
```

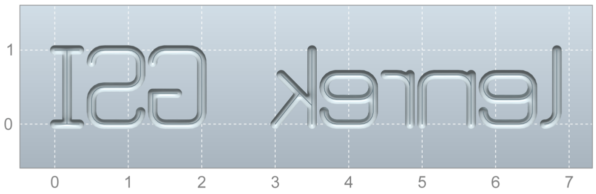

Fig. 106: Mirror text character by character in X

Example: Mirror in X, complete

Programing Example

Mirror in X - mirror

If only `@P62` is used to mirror in X, only each character is mirrored separately, but the order of the characters remains unchanged. To mirror the text completely, use `@P23` in addition for text inversion.

```
; mirror in X, complete
L CYCLE [NAME  = "SysMillEngrave.ecy"                   \
          @P1  = 0                                      \
          @P2  = 2                                      \
          @P3  = 1                                      \
          @P4  = 0.2                                    \
          @P11 = 2000                                   \
          @P12 = 1000                                   \
          @P21 = "ISG kernel"                           \
          @P23 = TRUE            ( invert text      ) \
          @P53 = 2                                      \
          @P62 = TRUE            ( mirror text in X ) \
        ]
M30
```

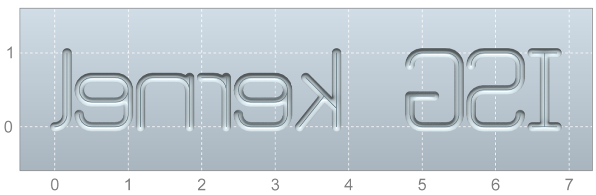

Fig. 107: Mirror complete text in X

Example: Mirror in Y

Programing Example

Mirror in Y

```
; mirror in Y
L CYCLE [NAME  = "SysMillEngrave.ecy"                      \
          @P1  = 0                                          \
          @P2  = 2                                          \
          @P3  = 1                                          \
          @P4  = 0.2                                        \
          @P11 = 2000                                       \
          @P12 = 1000                                       \
          @P21 = "ISG kernel"                               \
          @P53 = 2                                          \
          @P63 = TRUE              ( mirror text in Y )  \
        ]
M30
```

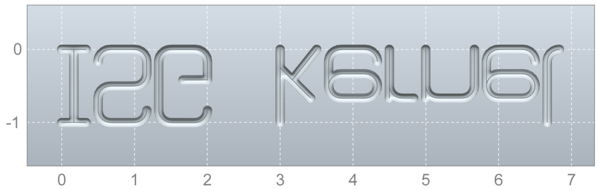

Fig. 108: Mirror text in Y

Letter spacing

Character spacing can be influenced by the parameters `@P64` in X and `@P65` in Y.

Example: X spacing with large tool radius

It may be advantage to increase the spacing, especially in X, if large tool radii are used

Programing Example

X spacing with large tool radius

```
; normal spacing in X
L CYCLE [NAME  = "SysMillEngrave.ecy"                    \
          @P1  = 0                                        \
          @P2  = 2                                        \
          @P3  = 1                                        \
          @P4  = 0.2                                      \
          @P11 = 2000                                     \
          @P12 = 1000                                     \
          @P21 = "ISG kernel"                             \
          @P42 = 0            ( reference point X      )  \
          @P43 = 1.5          ( reference point Y      )  \
          @P53 = 2                                        \
          @P64 = 0            ( normal X spacing )  \
        ]
```

```
; adapt spacing in X
L CYCLE [NAME  = "SysMillEngrave.ecy"                             \
          @P1  = 0                                                \
          @P2  = 2                                                \
          @P3  = 1                                                \
          @P4  = 0.2                                              \
          @P11 = 2000                                             \
          @P12 = 1000                                             \
          @P21 = "ISG kernel"                                     \
          @P42 = 0           ( reference point X               ) \
          @P43 = 0           ( reference point Y               ) \
          @P53 = 2                                                \
          @P64 = 0.1         ( increase X spacing by 0.1 ) \
        ]
M30
```

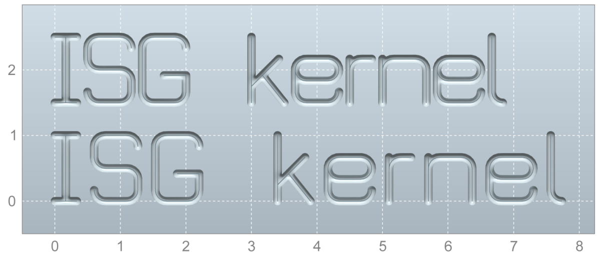

Fig. 109: Text with normal spacing (top) and slightly increased spacing (bottom)

Example: Y spacing for staircase effect

A staircase effect is obtainable with a Y offset.

Programing Example

Y spacing for staircase effect

```
; staircase effect in Y
L CYCLE [NAME  = "SysMillEngrave.ecy"           \
          @P1  = 0                              \
          @P2  = 2                              \
          @P3  = 1                              \
          @P4  = 0.2                            \
          @P11 = 2000                           \
          @P12 = 1000                           \
          @P21 = "ISG kernel"                   \
          @P53 = 2                              \
          @P65 = 0.1          ( Y spacing ) \
        ]
M30
```

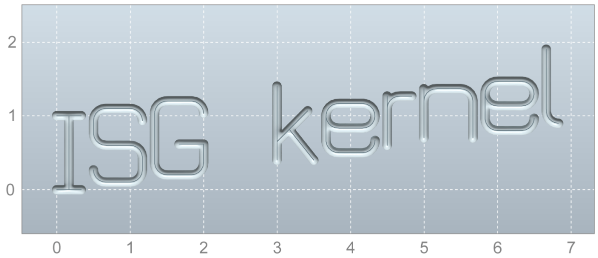

Fig. 110: Staircase effect

6.2.3 Syntax

L CYCLE [NAME = SysMillEngrave.ecy" , \ = ..]

6.2.4 Output variables

Variable	Value
V.CYC.SysRetBBoxMinX	X coordinate of the bottom left corner of the character box
V.CYC.SysRetBBoxMinY	Y coordinate of the bottom left corner of the character box
V.CYC.SysRetBBoxMaxX	X coordinate of the top right corner of the character box
V.CYC.SysRetBBoxMaxY	Y coordinate of the top right corner of the character box

Type of output

The parameter `@P71` determines the type of output. The output variables are assigned independent of this parameter in every case.

By default, `@P71 = 0` outputs the cycle G code and the machine executes the corresponding movements.

If `@P71 = 1` is set, only the output variables are assigned and every further output is suppressed. In particular, the machine does not move. This mode can be used to execute only one size request, for example. How big is the letter "Ä" in font size 1? How big is the character box for the text "ISG kernel" in font size 1?

Programing Example

Output

```
; create output variables
#VAR
  V.CYC.SysRetBBoxMinX
  V.CYC.SysRetBBoxMinY
  V.CYC.SysRetBBoxMaxX
  V.CYC.SysRetBBoxMaxY
#ENDVAR

; no output
L CYCLE [NAME  = "SysMillEngrave.ecy"      \
         @P1   = 0                         \
         @P2   = 2                         \
         @P3   = 1                         \
         @P4   = 0.2                       \
```

```
            @P11 = 2000                    \
            @P12 = 1000                    \
            @P21 = "ISG kernel"            \
            @P53 = 2        ( font size  ) \
            @P71 = 1        ( no output ) \
          ]

; output character box
#MSG SAVE EXCLUSIVE ["X coordinate of bottom left corner = "]
#MSG SAVE EXCLUSIVE ["%f",                V.CYC.SysRetBBoxMinX]
#MSG SAVE EXCLUSIVE ["Y coordinate of bottom left corner = "]
#MSG SAVE EXCLUSIVE ["%f",                V.CYC.SysRetBBoxMinY]
#MSG SAVE EXCLUSIVE ["X coordinate of top right corner = "]
#MSG SAVE EXCLUSIVE ["%f",                V.CYC.SysRetBBoxMaxX]
#MSG SAVE EXCLUSIVE ["Y coordinate of top right corner = "]
#MSG SAVE EXCLUSIVE ["%f",                V.CYC.SysRetBBoxMaxY]

M30
```

6.2.5 Programming example

This cycle call engraves the text “ISG kernel” on the workpiece surface. The line width of the letters is given by the radius of the tool used.

Programing Example

Engraving

```
T15 D15                    ; Tool data
M6                         ; Tool change
G90 G54 S1500 M3 F1000     ; Technology data

; cycle call parameter:
#VAR
  V.L.SurfacePositionZ      = 0    ; Z position of workpiece surface
  V.L.RetractionPlane       = 20   ; Z position of retraction plane
  V.L.SafetyClearance       = 2    ; relative value of safety clearance
  V.L.Depth                 = 0.2  ; depth of path milling
  V.L.TextSize              = 2    ; size of text
#ENDVAR

G00 X10 Y10        ; position of text

L CYCLE [NAME   = "SysMillEngrave.ecy"       \
  @P1   = V.L.SurfacePositionZ               \
  @P2   = V.L.RetractionPlane                \
  @P3   = V.L.SafetyClearance                \
  @P4   = V.L.Depth                          \
  @P21  = "ISG kernel"                       \
  @P53  = V.L.TextSize                       \
  ]

M05

M30
```

7 Deburring

7.1 Overview

Overview

The deburring cycle can be used to add a chamfer to the machined workpiece after a milling operation.

Possible applications

The cycle can be called in combination with the following milling cycles:

- Mill contour pocket [▸ 131]
- Mill contour spigot [▸ 137]
- Mill circular pocket [▸ 88]
- Mill circular spigot [▸ 70]
- Mill rectangular pocket [▸ 93]
- Mill rectangular spigot [▸ 76]
- Mill rectangular spigot [▸ 76]
- Mill rectangular spigot [▸ 109]
- Mill circumferential slots [▸ 109]
- Mill longholes arranged in a circle [▸ 99]
- Mill multi-edge [▸ 82]
- Milling open slot [▸ 116]
- Mill circumferential slots [▸ 104]

7.2 SysMillChamfer cycle - Deburring

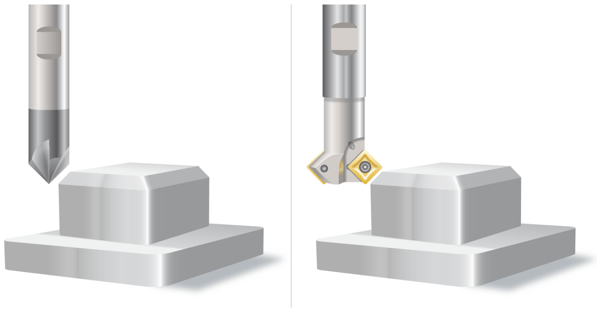

Fig. 111: SysMillChamfer

7.2.1 Process

To ensure that the cycle is executed successfully, the following requirements must be met.

- The tool radius is defined by V.G.WZ_AKT.R **(the tool radius is always specified at the face of the tool = 0)**
- Tool length compensation is active.
- The spindle speed must be selected before the cycle is started.
- The workpiece edge to be deburred is reachable without collision at the height of the retraction plane
- All reference cycle parameters containing geometric descriptions of the milled object are transferred.

The workpiece is prepositioned within a cycle at the retraction plane. Then the tool is fed in Z with a slight offset to the workpiece edge before the required chamfer is produced by moving along the workpiece edge. To create a slight offset to the edge of the workpiece, you must either specify the machining depth (@P4) at a slightly greater depth than the bevel depth to be milled (@P101) or specify a radial tool offset to the edge of the workpiece (@P104).

The operation may vary slightly depending on the reference cycle used.

7.2.2 Parameters

All reference cycle parameters containing geometrical descriptions of the object to be milled must be transferred for deburring (see Programming example [▸ 181]).

Required input parameters

Input parameters	Description
@P100	Reference cycle as character string

Only one parameter must be defined from the following list. The other parameter is calculated from the specified parameter and the chamfer geometry data. We recommend specifying the tool radius of the milling cutter used in the reference cycle (@P104). The resulting uniform tool offset to the workpiece edge results in a uniform chamfer. However, if the machining depth is specified, it can lead to an uneven chamfer since the radius of the previous milling cutter is not included.

Input parameters	Description
@P1	Reference plane (absolute)
@P2	Retraction plane (absolute)
@P3	Safety clearance in Z (relative to reference plane/machining height)
@P4	Machining depth If this parameter is specified, it can lead to an uneven chamfer since the radius of the previous milling cutter is not included.
@P104	Tool radius of the milling cutter used in the reference cycle (corresponds to radial tool offset to the workpiece edge during chamfer milling) If the chamfer milling cutter radius is smaller than this tool radius, the parameters must be defined in @P4.

In addition, only one parameter must be defined from the following list. The other parameter is calculated internally from the specified parameter and the point angle of the chamfer cutter (@P103).

Input parameters	Description
@P101	Chamfer depth
@P102	chamfer width

Optional input parameters

Input parameters	Description
@P103	Tool point angle Default value = 90 degrees
@P105	Feed rate for machining

	Default value = Active feed rate at cycle call
@P106	Feed rate for infeed in Z Default value = Active feed rate at cycle call

It is recommended to use the Syntax check.to verify whether the input parameters are correctly assigned.

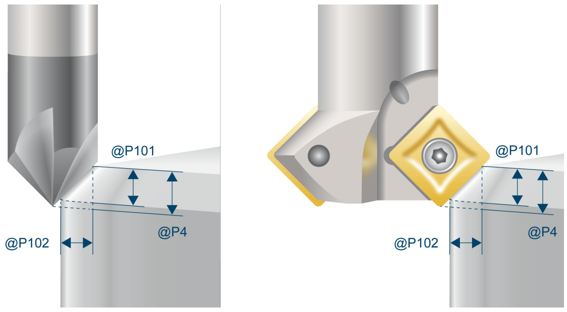

Fig. 112: Parameter definition of variant 1

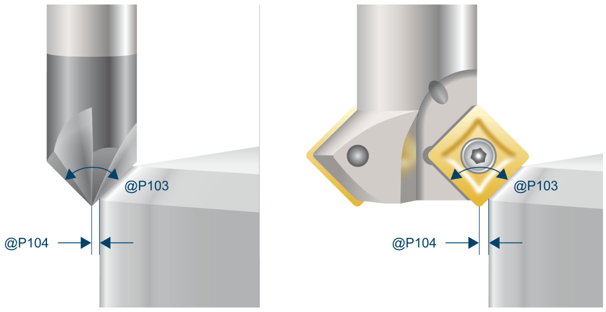

Fig. 113: Parameter definition of variant 2

7.2.3 Syntax

L CYCLE [NAME = SysMillChamfer.ecy @P.. = ..]

7.2.4 Programming example

Programing Example

Mill circular pocket

This example describes the milling and deburring operations for a circular pocket.

```
#VAR

  ; input parameters: pocket
  V.L.SurfacePositionZ = 0    (Z position of workpiece surface)
  V.L.RetractionPlane  = 50  (Z position of retraction plane)
  V.L.SafetyClearance   = 2    (relative value of safety clearance in Z)
  V.L.Depth             = 10   (depth of pocket)
  V.L.MaxIncrementZ     = 10   (maximum infeed in Z)
  V.L.MaxIncrementXY    = 2.5  (maximum infeed in XY)
  V.L.FeedZ             = 2000 (plunging feedrate)
  V.L.MachiningMode     = 1    (machining mode)
  V.L.Diameter          = 20   (diameter of pocket)
```

```
  ; input parameters: chamfer
  V.L.ChamferDepth       = 1    (depth of chamfer)
  V.L.ChamferFeedZ       = 3000 (machining feedrate in Z)
  V.L.ChamferMachDepth   = 2    (machining depth for chamfer)

#ENDVAR

G54

; polynomial contouring for smooth movements
#CONTOUR MODE [DEV, PATH_DEV = 0.01]
G261

; positioning to the starting point
G00 X50 Y50

; tool change
T8 D8 (endmill 5mm)
M6
; technology data
G90 M03 S6000 F4000

; cycle call for milling
L CYCLE [NAME=SysMillCircularPocket.ecy    \
  @P1  = V.L.SurfacePositionZ              \
  @P2  = V.L.RetractionPlane               \
  @P3  = V.L.SafetyClearance               \
  @P4  = V.L.Depth                         \
  @P5  = V.L.MaxIncrementZ                 \
  @P6  = V.L.MaxIncrementXY                \
  @P21 = V.L.FeedZ                         \
  @P31 = V.L.MachiningMode                 \
  @P68 = V.L.Diameter                      \
  ]

; tool change
T9 D9 (chamfering mill)
M6
; technology data
G90 M03 S6000 F2000

; cycle call for milling
L CYCLE [NAME=SysMillChamfer.ecy     \
  @P1   = V.L.SurfacePositionZ       \
  @P2   = V.L.RetractionPlane        \
  @P3   = V.L.SafetyClearance        \
  @P4   = V.L.ChamferMachDepth       \
  @P68  = V.L.Diameter               \
  @P100 = "SysMillCircularPocket"    \
  @P101 = V.L.ChamferDepth           \
  @P106 = V.L.ChamferFeedZ           \
  ]

G260
M05
M30
```

8 Appendix

8.1 Suggestions, corrections and the latest documentation

Did you find any errors? Do you have any suggestions or constructive criticism? Then please contact us at documentation@isg-stuttgart.de. The latest documentation is posted in our Online Help (DE/EN):

QR code link: https://www.isg-stuttgart.de/documentation-kernel/

The link above forwards you to:

https://www.isg-stuttgart.de/fileadmin/kernel/kernel-html/index.html

Notice

Change options for favourite links in your browser;

Technical changes to the website layout concerning folder paths or a change in the HTML framework and therefore the link structure cannot be excluded.

We recommend you to save the above "QR code link" as your primary favourite link.

PDFs for download:

DE:
https://www.isg-stuttgart.de/produkte/softwareprodukte/isg-kernel/dokumente-und-downloads

EN:
https://www.isg-stuttgart.de/en/products/softwareproducts/isg-kernel/documents-and-downloads

E-Mail: documentation@isg-stuttgart.de

Index

© Copyright
ISG Industrielle Steuerungstechnik GmbH
STEP, Gropiusplatz 10
D-70563 Stuttgart
All rights reserved
www.isg-stuttgart.de
support@isg-stuttgart.de