RZESZOW UNIVERSITY OF TECHNOLOGY

THE FACULTY OF MECHANICAL ENGINEERING AND AERONAUTICS
RZESZOW UNIVERSITY OF TECHNOLOGY

Programming of CNC machine tools

dr inż. Roman Wdowik

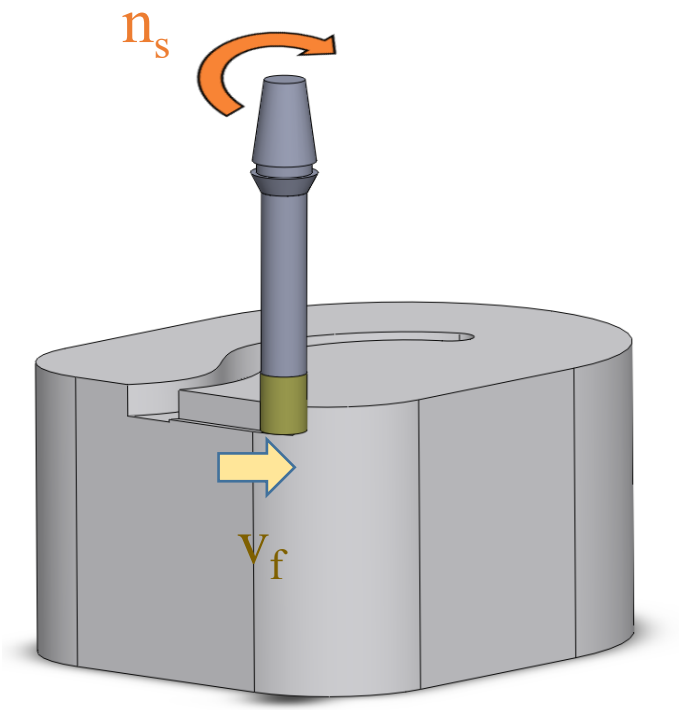
e-mail: rwdowik@prz.edu.pl

What does „programming” mean in general?

CNC machine tool

- To create the tool/the head path

- To set the machining parameters (e.g. feed, spindle speed)



Programming must be supported by specific language which is „clear and understandable for the specific CNC machine tool with specific CNC controller (e.g. Sinumerik, Fanuc, Heidenhain, etc.) ”

- (LENGTH AND DIAMETER MEASURE)
- N130 (SHELL MILL)
- N140 (END MILL)
- N145 (LENGTH AND DIAMETER MEASURE)
- G00 G90
- G65 P9023 A13. T#20 D#7 H#8
- GOTO800
- N170 (PROBE)
- G65 P9023 A21. T#20
- GOTO800
- (MANUALLY CHECK TOOLS)
- N300 (MANUALLY)
- G00 G90
- G65 P9023 A25. T#20
- GOTO800
- (CHECK TOOLS FOR BREAKAGE AND WEAR)

Language

- **Standard DIN/ISO code according to standards: DIN66025, ISO 6983-1:2009**
- **Specific CODE created by the producer of CNC controller (e.g. Heidenhain company)**

Important remark!

- If you have several machine tools which can be programmed using G-code it does not mean that every one of your machine tools may be programmed the same way! It is always easier to learn programming of the next CNC controller if you know at least one, so do not worry about the differences because the basics are very similar!

Conclusion:

- This G-code may differ depending on the producer of the controller of the machine tool!

Methods of Programming of CNC Machine Tools

Method 1

Direct programming

using special „words”

e.g. G1, G2, G90, etc.

- TEXT EDITOR, NC CODE EDITOR
- A SHEET OF PAPER
- EDIT MODE IN CNC CONTROLLER

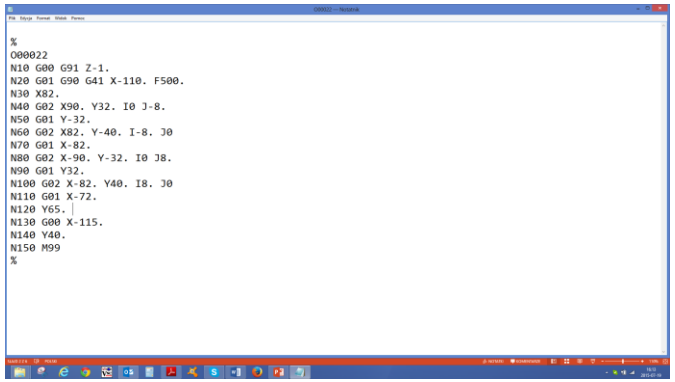


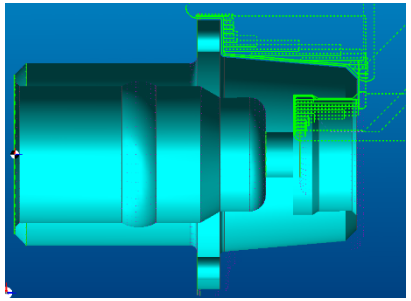
Fig. 1. Example of direct programming in text editor

Method 2

CAD/CAM programming

using CAD/CAM programs

INDIRECT DATA



POSTPROCESSING

NC CODE

Fig. 2. CAD/CAM programming (example of programming in Edgecam software)

Method 3

Dedicated workshop or DIALOG PROGRAMMING

at the CNC machine tool or PC

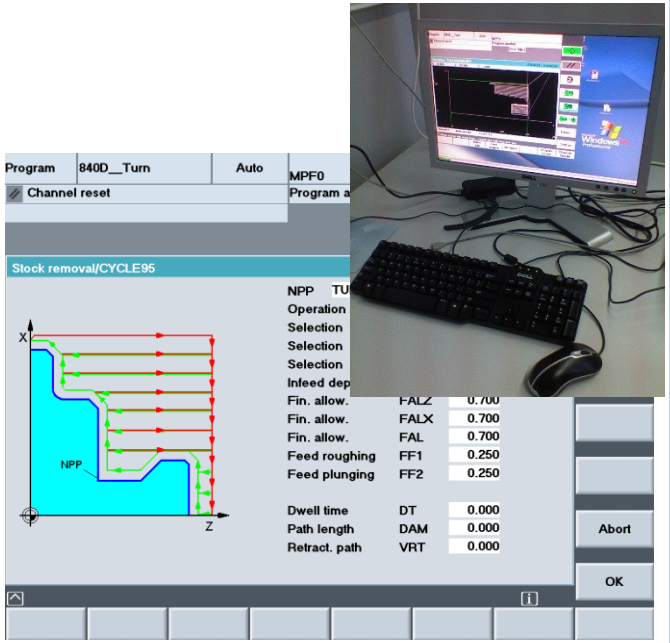


Fig. 3. Dedicated programming using dialog windows

(dialog window in Sinutrain Software from **Siemens** company)

I would like to explain in details these 3 methods.

Method 1

Direct Programming

Parametric programming

Macros programming

also called

Simple programming

It uses simple G-codes (or other standard codes) and standard cycles manufactured by the producer of CNC controller

Advanced direct programming

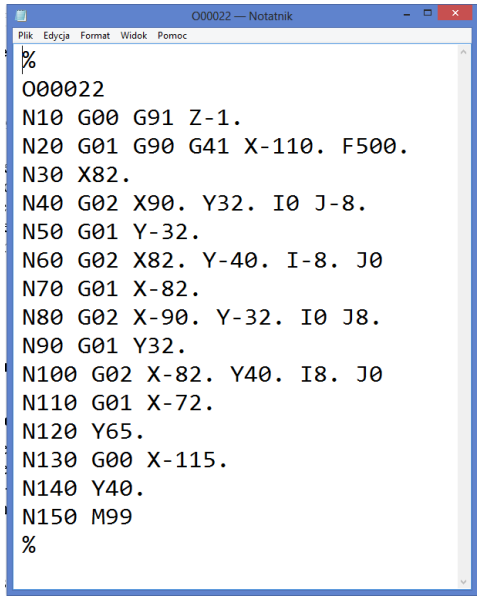
It uses simple G-codes (or other standard codes) and standard cycles manufactured by the producer of CNC controller + **advanced functions and instructions of CNC controller (VARIABLES, CONDITIONS, LOOPS, „JUMPS”, REPETITIONS, ETC.)**

Drawing of a part

Catalogues of cutting tools

Manuals of CNC controllers

Manuals of machine tools



Editing the NC programs or subprograms files

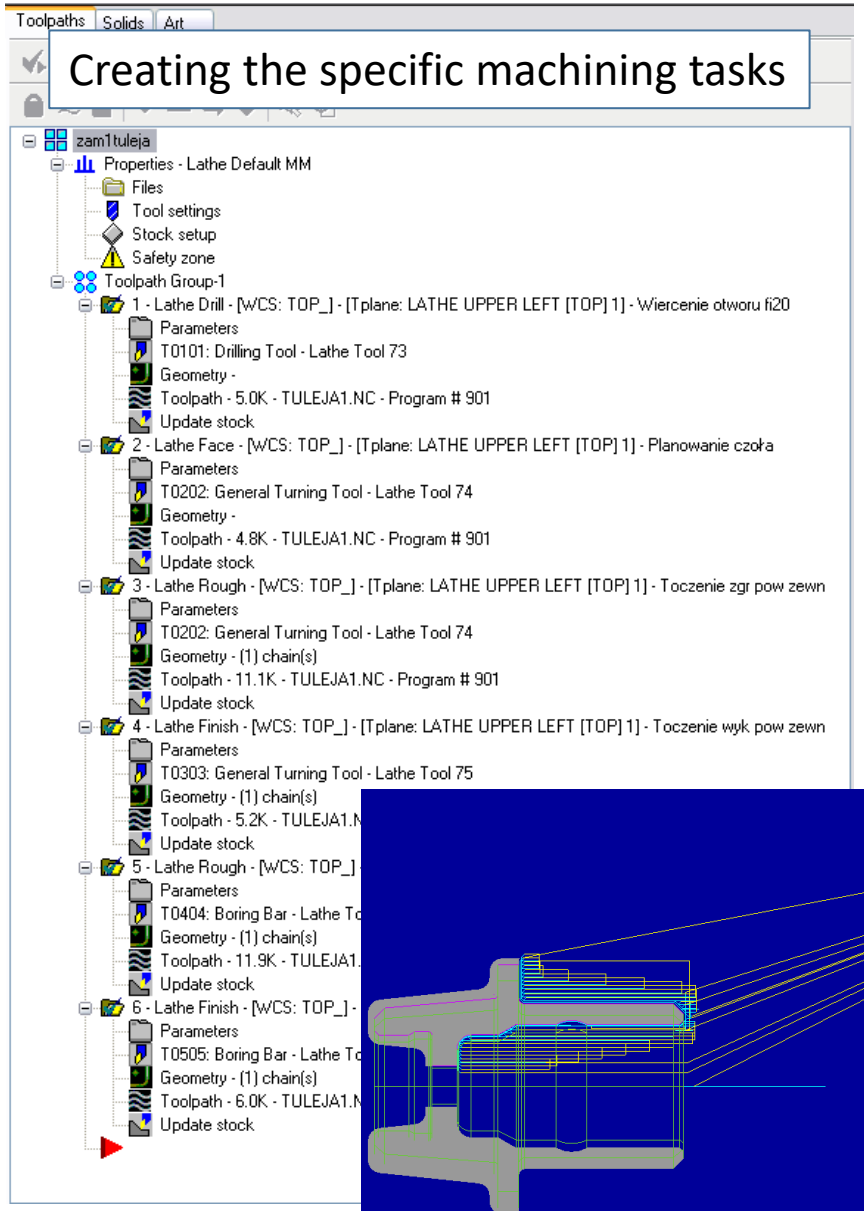
Sending or writing it in the memory of CNC controller

Testing program and starting production

Method 2

CAD/CAM Programming

Creating the specific machining tasks



Indirect data

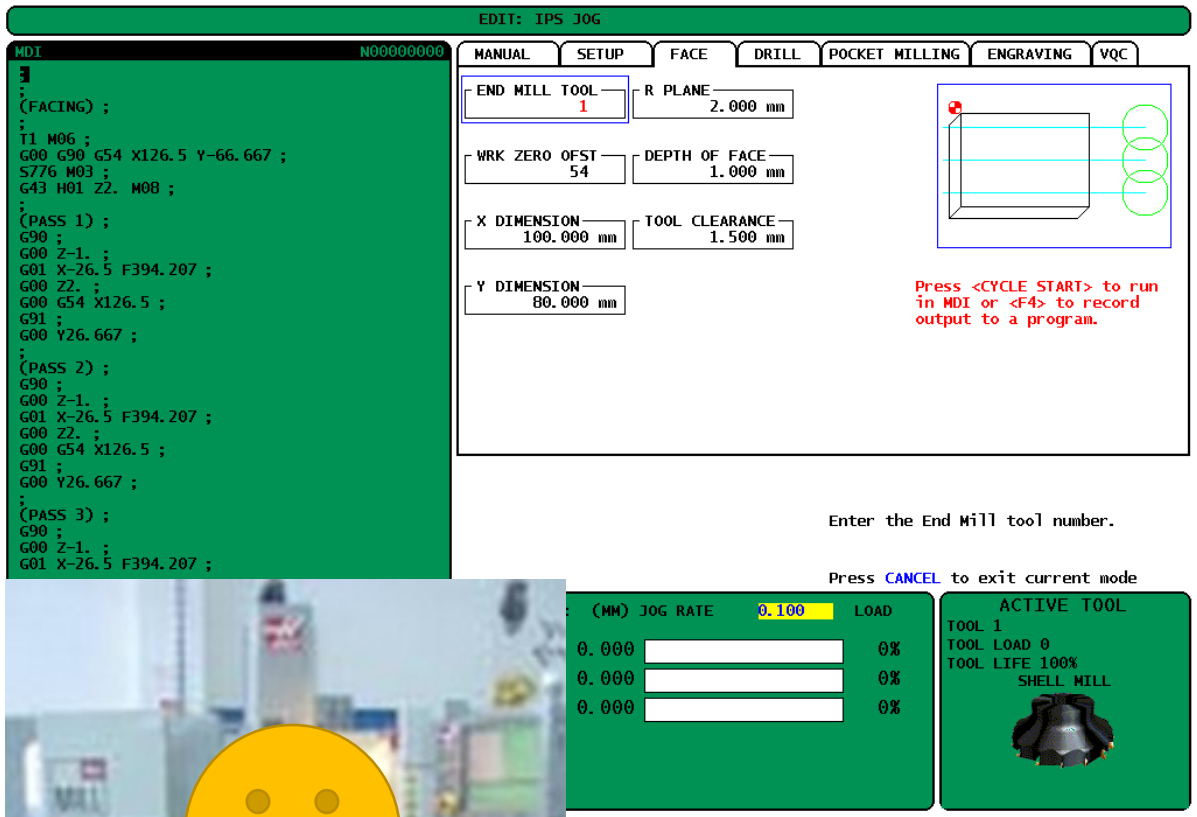
Postprocesing

```
%MPF0901
N20 G90 G95 G71 G40
N30 G54 D200
N31 G0 X0. Z150.
N32 T01 D01 M5 M9
N40 G0 Z150. S1250 M3 M8
N50 G0 X0.
N60 Z5.
N70 Z-58.627
N80 G1 Z-130.627 F.36
N90 G0 Z-58.627
N100 Z5.
N110 Z150.
N120
N130 G0 Z150. S0100
N131 T02 D02 M4 M8
N140 G0 X91.187
N150 Z5.05
N160 G92 S2000
N170 G96 F.25 S0245
N180 Z.8
N190 G1 X53.857
N200 G0 Z2.8
N210
N220 X121.727
N230 Z-70.804
N240 G1 Z-72.804
N250 Z-78.5
N260 X125.6
N270 X128.428 Z-77.086
N280 G0 Z-70.079
N290 X117.855
N300 G1 Z-72.079
N310 Z-78.5
N320 X122.127
N330 X124.956 Z-77.086
N340 G0 Z-70.
N350 X113.982
```

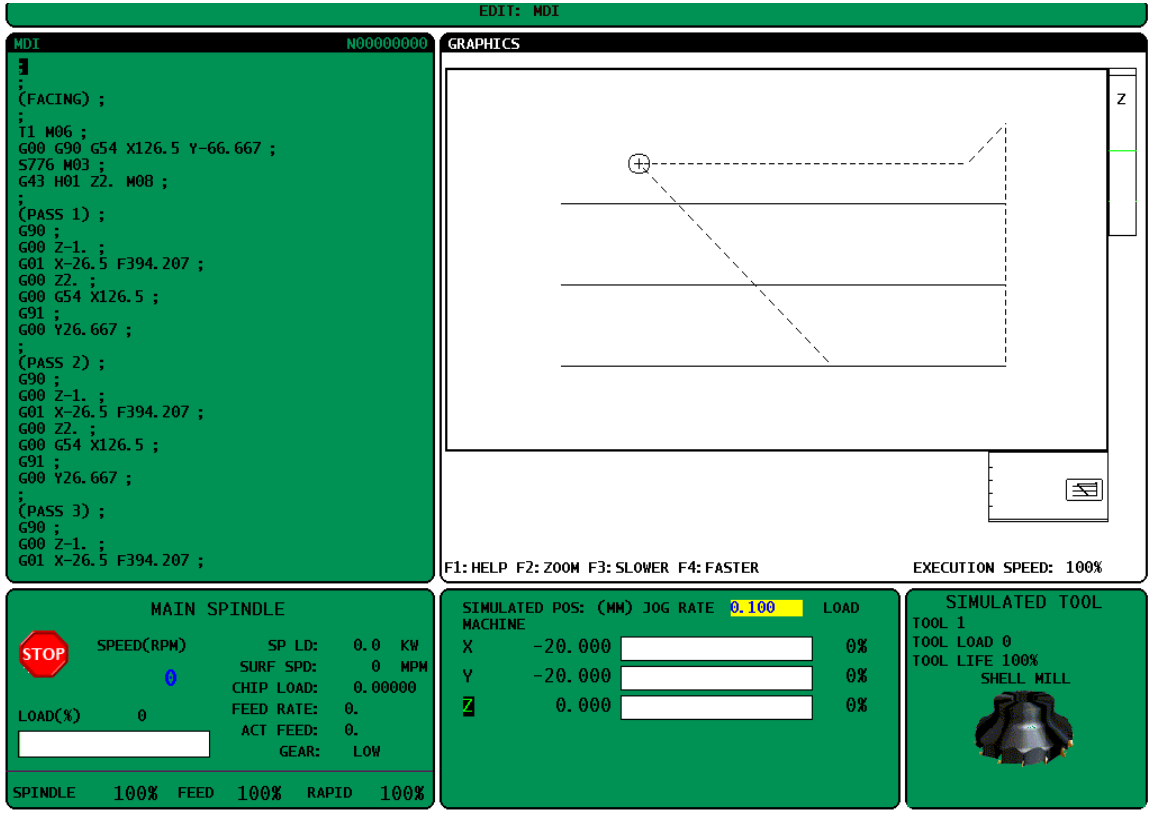
Source: R. Wdowik's master thesis

Method 3

Dialog Programming



Dialog window



Simulation

Testing the program
and start of machining

The examples of controllers which allow to use dialog windows for workshop programming

- **Heidenhain**
- **Sinumerik**
- **Haas**
- **Fanuc**
- **Etc.**

Sometimes producers of CNC machine tools create their own dialog windows for specific controllers! It means that they are not the products of controller's producer!

Exemplary webpages of CNC controls producers

CNC 8065 for lathes

Designed to control high-production turning centers, vertical lathes, slanted-bed lathes, parallel lathes, dual-turret (TT) lathes, lathes with several turrets and spindles and dual-purpose (mill-lathe) machines.

Ergonomic and robust design.

- New line of keyboards with touch screen, integrated mouse and USB connector.
- Very visual browsing.
- Mounting design and component technology that meets sealing standard IP65 (NEMA12).

High speed machining

- Polynomial interpolation (splines) that provides an excellent part surface finish and high machining speed.
- More accurate finish, adapting the machining conditions to the dynamics of the machine evaluating in advance the programmed tool path changes.
- HSSA (High Speed Surface Accuracy) machining system that reduces mechanical strain on the machine and achieves smoother movements.

More information

First name:

Company:

E-mail *:

Country *:

Select country

Comment *:

* Required fields

I would like to receive updates on FAGOR Automation by email..

I have read and accept the terms of use and privacy statement.

CNC Learning and Support Tool for Customization

FANUC CNC GUIDE
FANUC CNC Simulator

Product Brochures

FANUC CNC GUIDE / CNC Simulator (English | Chinese) (PDF file)

FANUC CNC GUIDE

FANUC CNC GUIDE is a PC software by which you can learn how to operate CNC, check the machining program, and confirm the operation of customization features.

Training
- Calendars
- Class Offerings
- Certification Programs
- Workforce Development
- Educational Resources
  - CNC Simulators
  - PLC Simulators
  - VFD Simulators
  - Robot Simulators
  - Simulation Software
  - Demo Inquiry
- Online Training (e-Learning)
- Locations
- My Training Profile

CNC Simulators

Mitsubishi Electric CNC simulators are designed to prepare the next generation of students and workers with the tools they need to succeed in the industrial automation industry. From desktop units to fully functional CNC mills and lathes, Mitsubishi Electric's broad offering of classroom simulators allow students the ability to develop practical skills in a safe manner. Students utilizing Mitsubishi Electric simulators will be trained on controls and programming methods common throughout the industrial automation industry, ensuring they are prepared to become valuable TEAM members and leaders in their respective fields.

EDU M80 SIM | EDU MILL NO-3D | EDU MILL W-3D | EDU PORTABLE MILL | EDU PORTABLE LATHE

Inquiries
Contact Us
- Shop Now
- Find a Distributor
- Find an OEM
- Systems Integrators
- Support
- Careers
- Locations
- En Español

DIAMOND PARTNER

The Examples of CAM or CAD/CAM Programs

More on: https://en.wikipedia.org/wiki/Computer-aided_manufacturing

www.dptechnology.com

SIEMENS

Registered Trademarks of Their Producers – here only for teaching

Adventages and disadventages of programming methods

Method 1

Direct programming:

- A programmer decides about the structure of NC program
- A programmer uses the functions (e.g. G-codes) the way he/she wants
- The programmer has to know functions of CNC controller
- Time consuming
- Programming of complex parts (e.g. 5-axis machining) is difficult and time consuming
- Parametric programming is possible

Method 2

CAD/CAM programming:

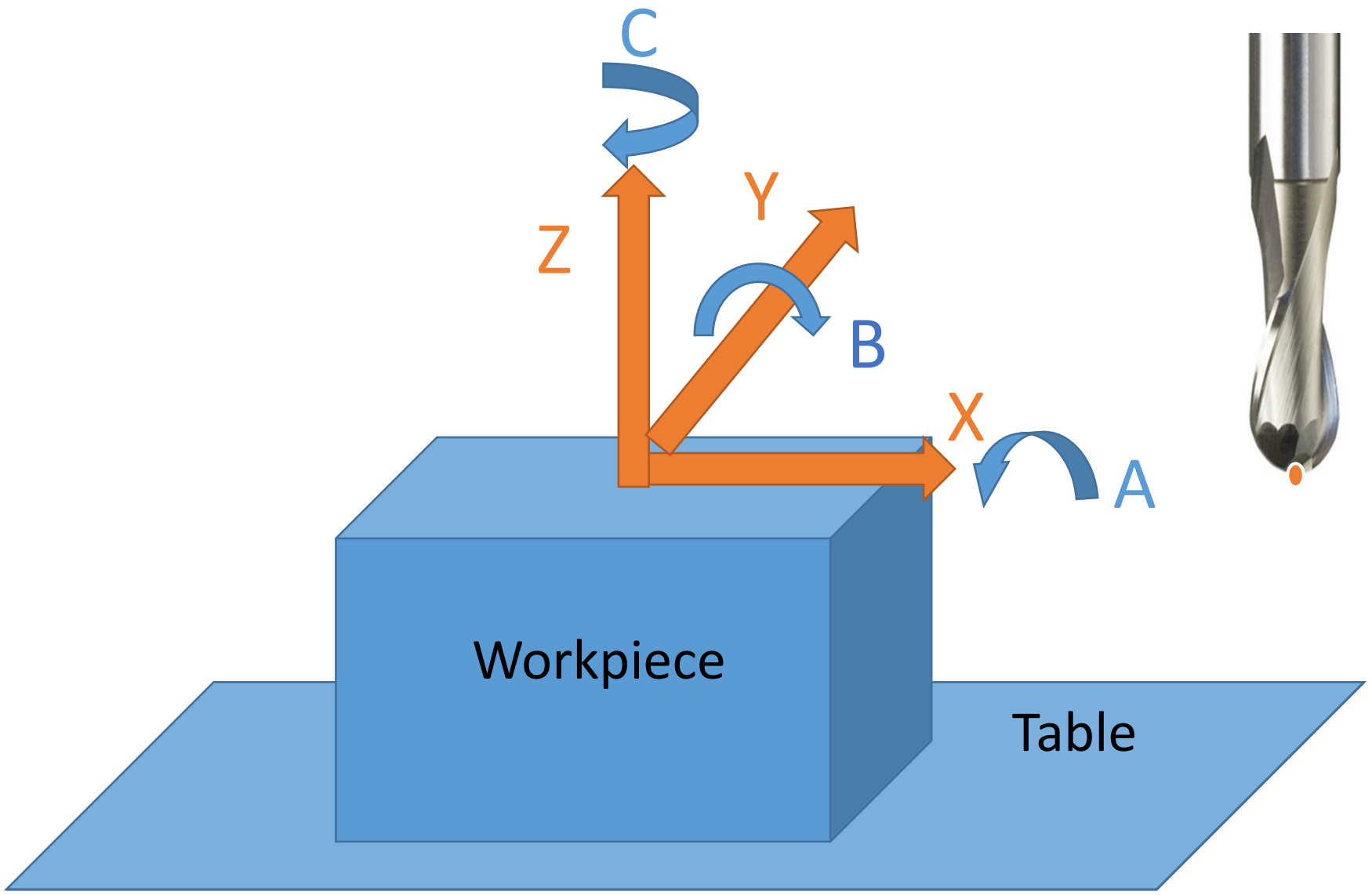
- The structure of NC program depends on postprocesor
- Changing G-codes in the final program is limited
- The programmer do not have to know G-codes if he/she has a good postprocesor
- The method is „fast” if the part is complex
- Programming of complex parts (e.g. 5-axis machining) is possible

Method 3

Workshop or PC dedicated programming:

- The programmer do not have to know functions (G-codes) of CNC controller or this knowledge may be very limited (That’s why this method is good for operators)
- Good for simple parts
- Programming is easy
- Programming of complex parts (e.g. 5-axis machining) is possible if specific cycles are avaible

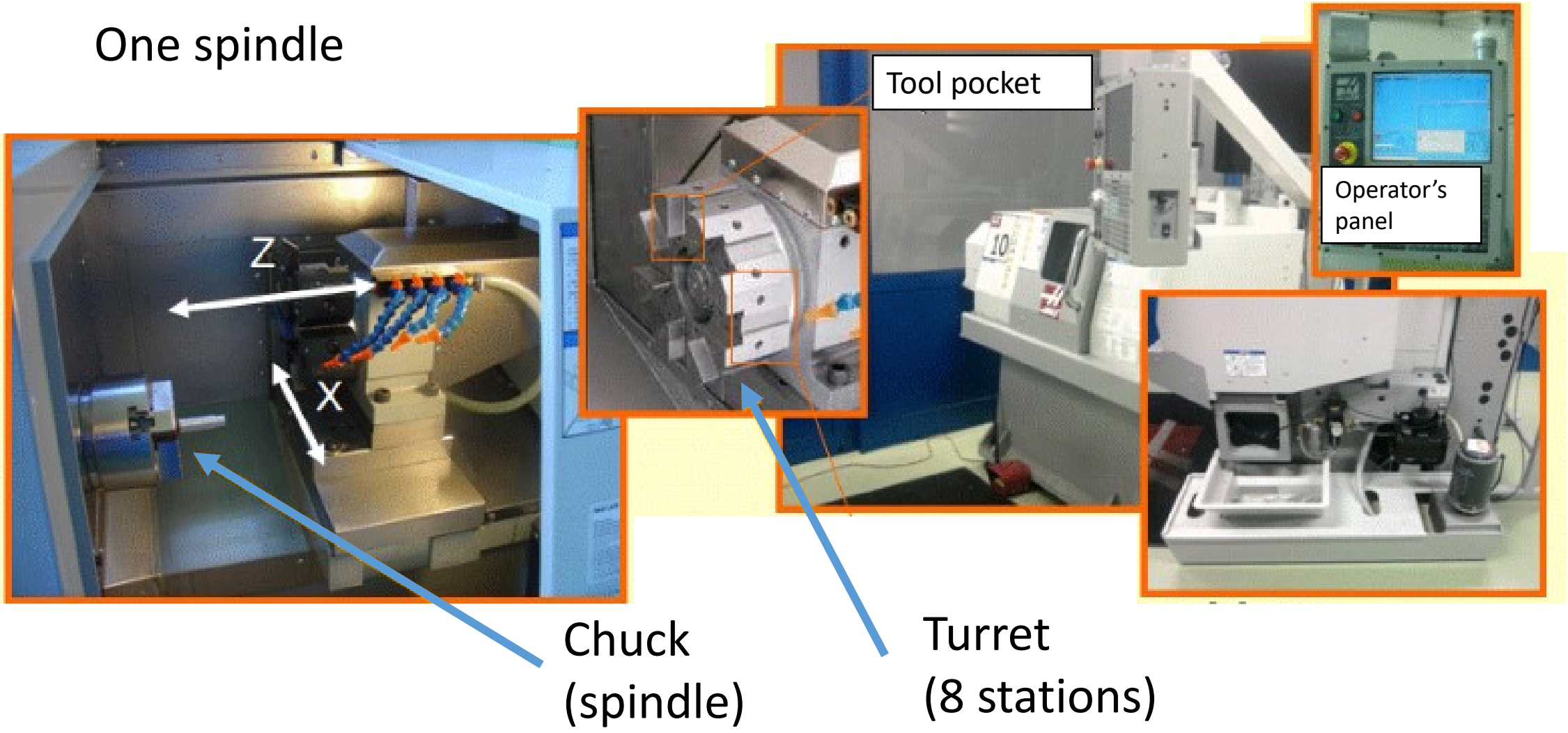
How to define the directions?



GT10 (Haas)

Two linear axes

One spindle



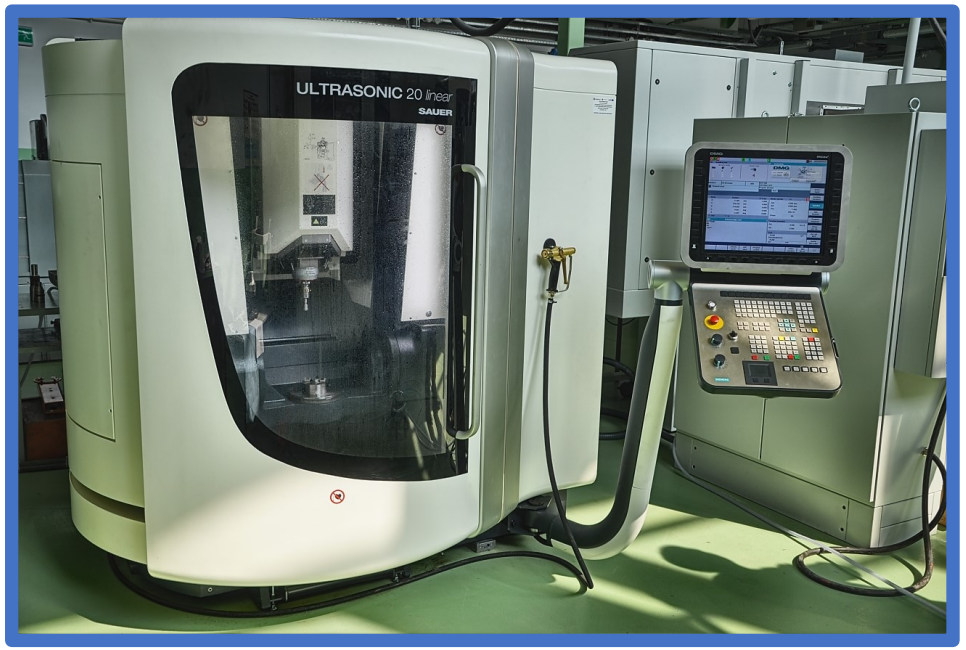
Ultrasonic 20 linear (Sauer)

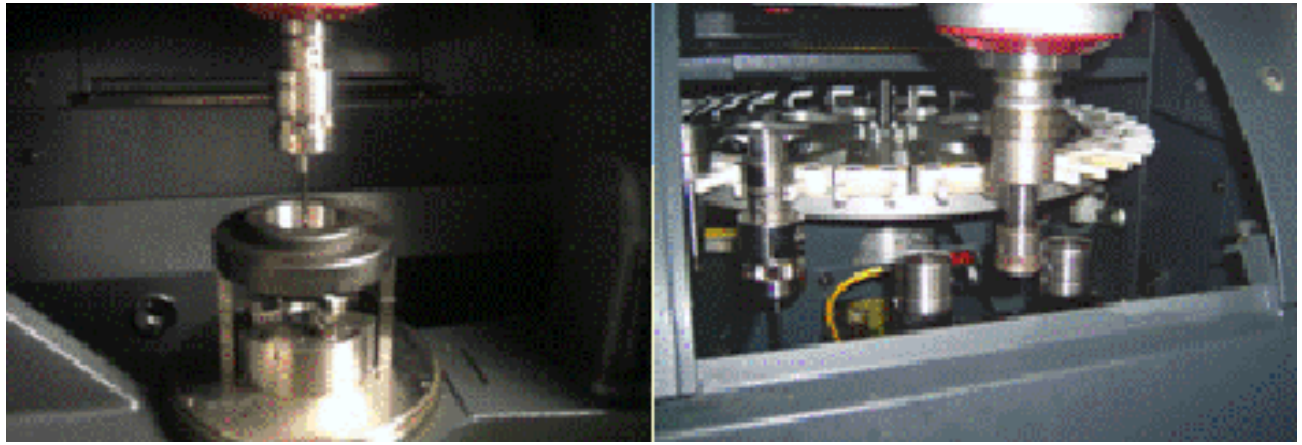
Three linear axes

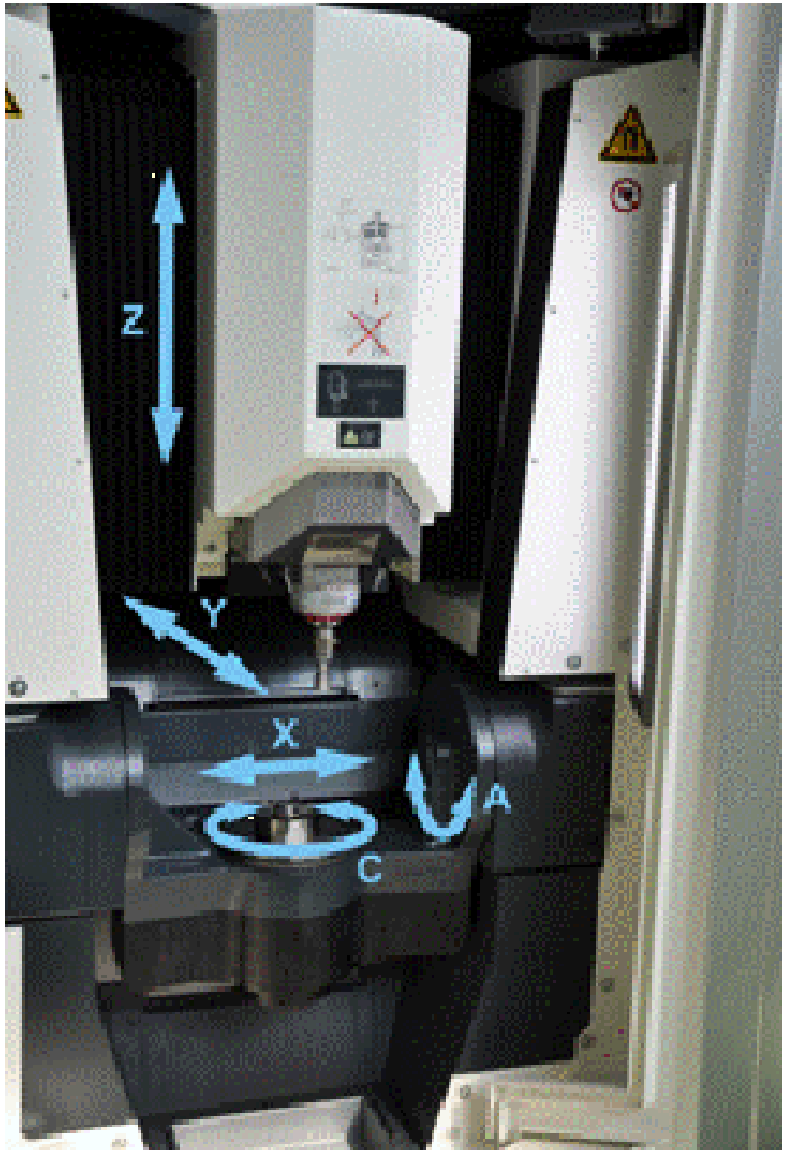
Two rotational axes

One spindle

Two measuring heads

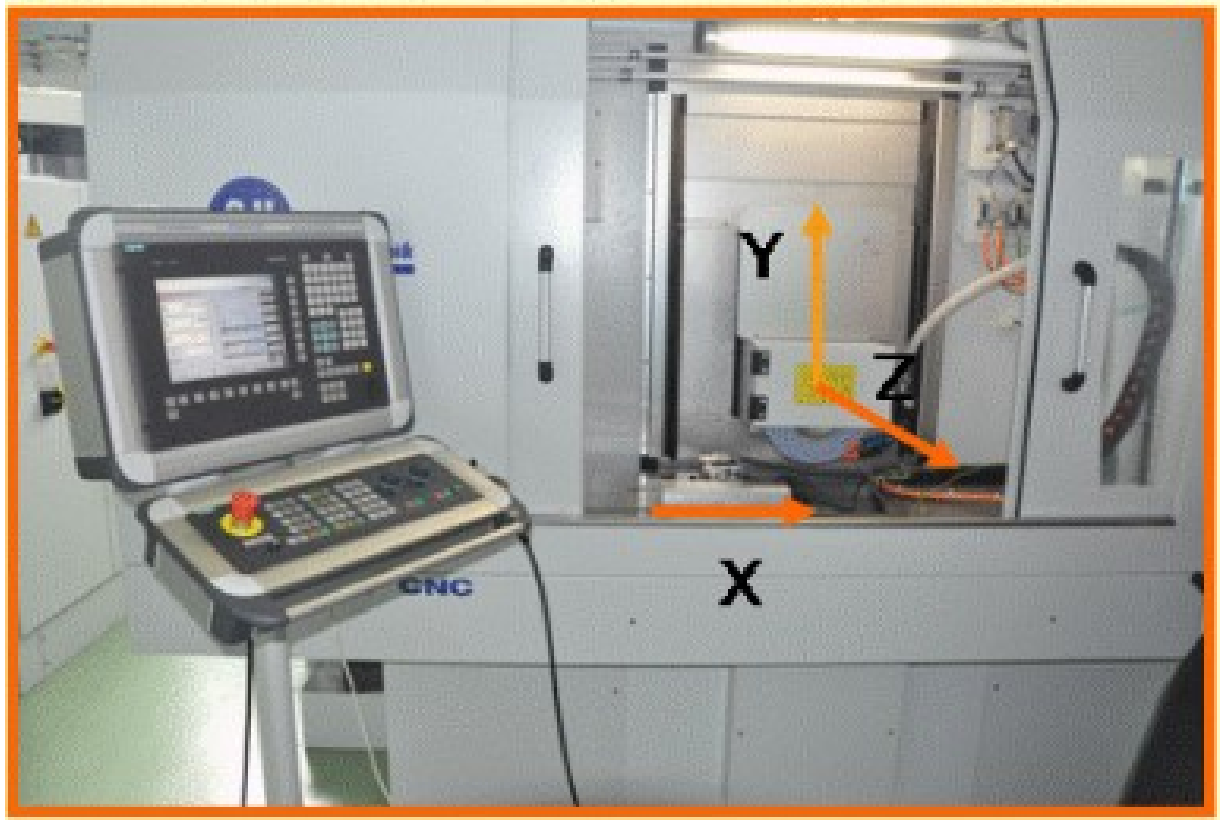






FS 640 C CNC (Geibel&Hotz)

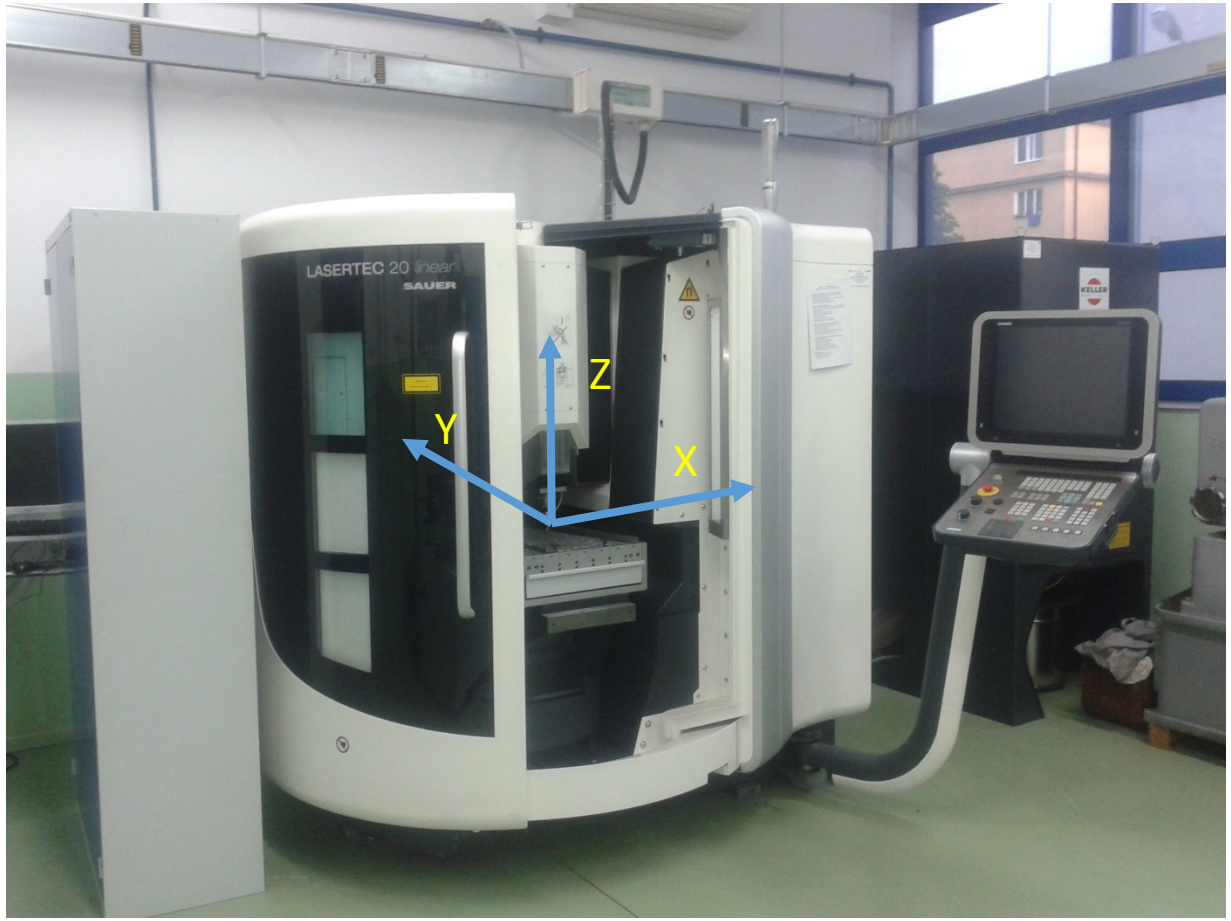
Three linear axes
One spindle
One roller dresser unit



Lasertec 20 linear (Sauer)

Three linear axes

Laser generator



Direct Programming

The general structure of NC program

The program consists of blocks.
The block consists of words.
The word consists of an address and its value.

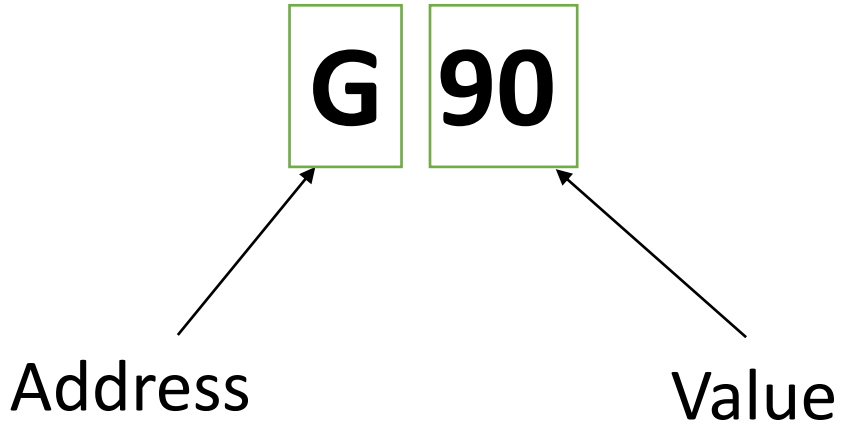
This is a program.

```
o1111
N10  G90 G54 G17
N11 T1 M6
N11 M98 P1112 L15
N12 M97 P1112 L15
N12 G0 X50. Y50.
N13 G0 Z0.
N14 G1 X68. Y45.
N15 M30 (The end of program)
N1112 G1 X50. Y20. F500.
N16 G1 G91 Z-1.
N17 M99
```

This is a block.

N10 G90 G54 G17

This is a word.



The general structure of NC program

This is a program.

```
o1111
N10  G90 G54 G17
N11 T1 M6
N11 M98 P1112 L15
N12 M97 P1112 L15
N12 G0 X50. Y50.
N13 G0 Z0.
N14 G1 X68. Y45.
N15 M30 (The end of program)
N1112 G1 X50. Y20. F500.
N16 G1 G91 Z-1.
N17 M99
```

The names of addresses (the names of codes):

- G - preparatory codes,
- M - miscellaneous codes,
- Address codes X, Y, Z, etc.,
- T, F, P, R - other codes.

The general structure of NC program

G-codes (preparatory codes) tell the CNC machine tool what type of action to do, e.g. do:

- rapid moves,
- move in a straight line or arc,
- set tool information,
- define axis beginning and ending positions,
- pre-set series of moves that bore a hole, cut a specific dimension, or a contour (canned cycles)*

G-code commands are either modal or non-modal. A modal G-code stays in effect until the end of the program or until you command another G-code from the same group. A non-modal G-code affects only the line it is in; it does not affect the next program line.*

* Reference: www.haascnc.com (Haas Mill Operator's Manual)

The general structure of NC program

The examples of G-codes:

G0 – RAPID MOTION
G1 – LINEAR INTERPOLATION MOTION
G2/G3 – CIRCULAR INTERPOLATION MOTION
G17/G18/G19 – INTERPOLATION PLANE
G40/G41/G42 – CUTTER COMPENSATION COMMANDS
G90 – ABSOLUTE POSITIONING
G91 – INCREMENTAL POSITIONING
G94 – FEED PER MINUTE MODE

The general structure of NC program

M-codes are miscellaneous machine commands that do not command axis motion. The format for an M-code is the letter M followed by digits; for example M03. Only one M-code is allowed per line of code in Haas controller but e.g. in the case of Sinumerik you can use several codes. All M-codes take effect at the end of the block in the case of Haas.*

The examples of M-codes:

M2 OR M30 – END OF PROGRAM
M3 – SPINDLE ON (CW)
M8 – COOLANT ON

* Reference: www.haascnc.com (Haas Mill Operator's Manual)

The general structure of NC program

Other codes are machine commands that are used to set machining parameters, axes, values of parameters of cycles, etc.

The examples of other codes:

T1 – SETTING THE TOOL NUMBER 1
F500. – SETTING THE FEED 500 MM/MIN
S2000 – SETTING THE SPINDLE SPEED 2000 REV/MIN
X50. – SETTING THE AXIS POSITION 50
(DEPENDING ON THE DIMENSIONING **G90 OR G91**)

The general structure of NC program

Main program

o1111 (The head block)
N10 G90 G54 G17
N11 T1 M6
N11 M98 P1112 L15
N12 M97 P1112 L15
N12 G0 X50. Y50.
N13 G0 Z0.
N14 G1 X68. Y45.
N15 M30 (The end of program)
N1112 G1 X50. Y20. F500.
N16 G1 G91 Z-1.
N17 M99 (The end of subprogram)

Subprogram (Internal)

Subprogram (External)

o1112 (The head block)

N10 G1 X50. Y20. F500.

N11 G1 G91 Z-1.

N15 M99 (The end of subprogram)

Simulation and software – webpages of selected producers

SOFTWARE

With EMCO Industrial Training Software the student experiences professional training for precise programming on modern CNC machines.
The result: smooth industrial production.

SinuTrain for SINUMERIK Operate

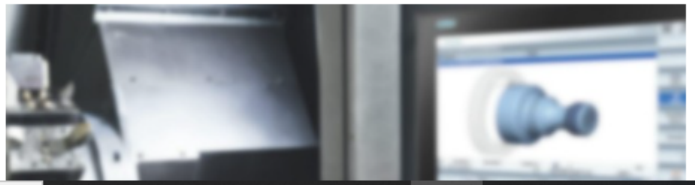
Overview and version choice

SinuTrain for SINUMERIK Operate 4.95 is available!

Read more here:

SinuTrain for SINUMERIK Operate V4.95

The control identical programming station for production planning, CNC program creation, and training



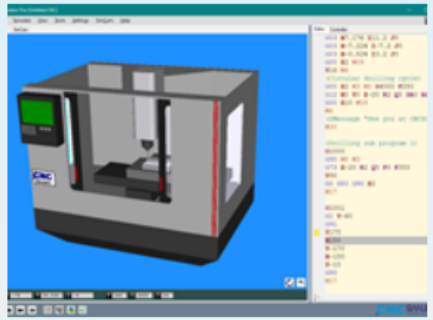
Overview

- CNC programming on the PC as on the CNC - same operation and programming
- Work preparation even while the machine is still cutting: test, run in and simulate NC programs

Download and install CNC Simulator Pro for Windows. See this page for what is needed to be able to run the program.
Downloading and installing our software means that you agree to our license agreement.

Program name: **CNC Simulator Pro**
Version: **3.2.0.0**
Variant: **STABLE**
Microsoft .NET Framework: **Version 4.5 or higher**
Windows Version: **7 or higher (32 bit)**

DOWNLOAD



Program name: **CNC Simulator Pro**
Version: **4.0.0.0**
Variant: **PUBLIC BETA 2* (Oct 22, 2021)**
Microsoft .NET Framework: **Version 4.8 or higher**
Windows Version: **10 or higher (64 bit)**
Video preview: CNC SIMULATOR PRO - YouTube

DOWNLOAD

VERICUT 9.2.2 – Increase Productivity and Sustainability

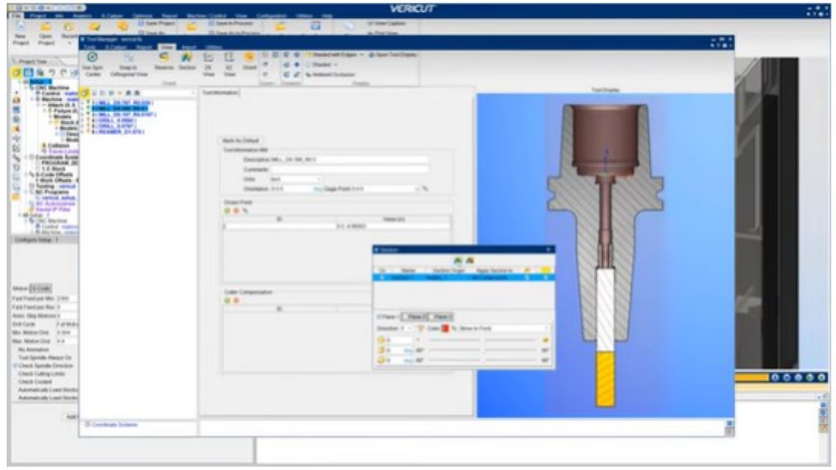
Latest VERICUT Release

Release Notes
Download

9.2 Flyer Download
Download

Request Latest VERICUT Release
Request

Previous VERICUT Release



Collision & Performance Improvements

VERICUT 9.2 brings significant speed increases and improved accuracy to collision checking and overall performance. Users gain substantial speed increases for deep concave collision penetration, turning operations (especially inside diameter work on large parts), and collision checking between highly detailed models such as those having high triangle counts. Simulate material removal at tight cutting resolutions up to 30% faster.

New 3DLive™ Interface

Simplify the VERICUT machine building process and create more realistic setups in a matter of seconds by importing 3DLive data. Import GDML format files containing 3D geometry, including colors for CNC machines, fixture components and cutting tool holders- along with kinematic information, travel limits, min/max feed rates for axes and initial machine position.

*3DLive™ is a registered trademark of MachiningCloud, Inc.

NCSIMUL Machine

The smart, high-performance CNC machining verification / simulation software experience

With over 20 years of R&D, the NCSIMUL G-code verification technology has become a combination of the **highest performance simulation software** available on the market and the standard for "ease of use" in today's software offerings.

Available for **turning**, **drilling**, **milling** (3 to 5 axis), **multi-tasking** or even more complex machining, NCSIMUL Machine is the most advanced machining verification software for simulating, verifying and optimizing CNC programs. Based on the **real characteristics of your CNC machine**, the result is a dynamic verification software that includes the exact environment for all machines, tools and materials.

> *"Countless man and machine hours are saved, since we don't have to verify step by step on the machine"*
>
> *Dean Dancer, Manufacturing Manager at Hutchinson Inc.*

While the practical interface demonstrates CNC collision detection using crisp 3D graphics in real time, the powerful attributes of the software provide **optimization** of cutting tool **feeds and speeds to reduce the parts' cycle time**.

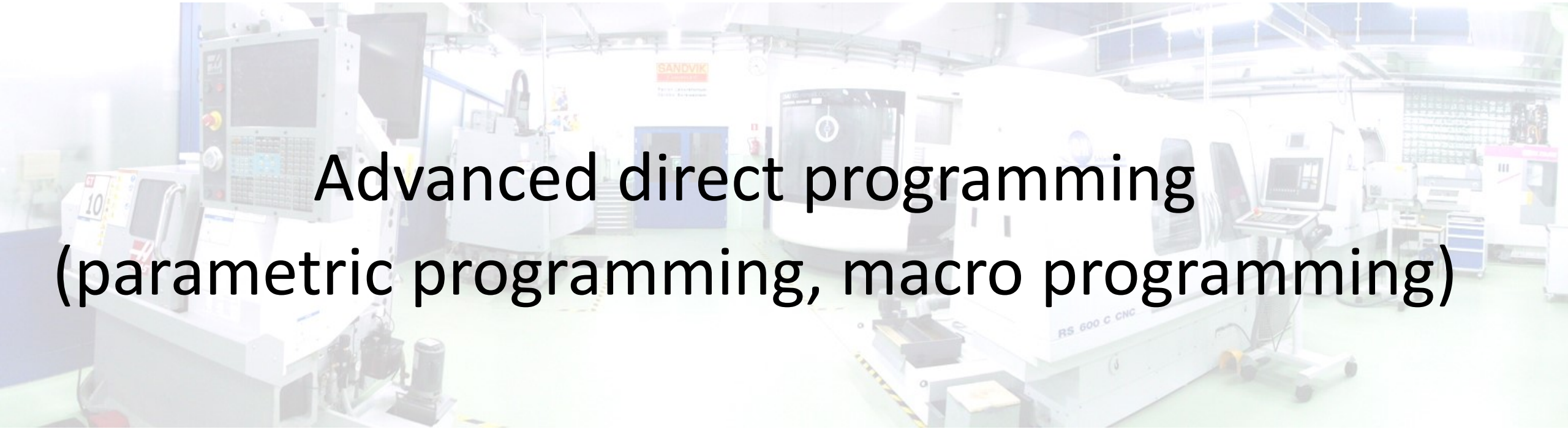
Collaboration in the workshop is also possible thanks to an embedded function that generates **CNC technical documents**, which can be shared and reviewed, along with the simulation, on **NCSIMUL player**.

NCSIMUL Machine is available as a standalone software, scalable to your needs. It facilitates your CNC verification process and tightly integrates your IT system through **CAD/CAM interfaces**. One of the key benefits, for example, is that any existing CAM data can be seamlessly imported into the software, eliminating the need to rebuild tool libraries.

If your company runs CNC machines, such as 5-axis or more complex machines, seriously consider NCSIMUL Machine to save time and money with better quality results:

RZESZOW UNIVERSITY OF TECHNOLOGY

THE FACULTY OF MECHANICAL ENGINEERING AND AERONAUTICS
RZESZOW UNIVERSITY OF TECHNOLOGY



Advanced direct programming (parametric programming, macro programming)

Method 1

Direct Programming

Parametric programming

Macros programming

also called

Simple programming

It uses simple G-codes (or other standard codes) and standard cycles manufactured by the producer of CNC controller

Advanced direct programming

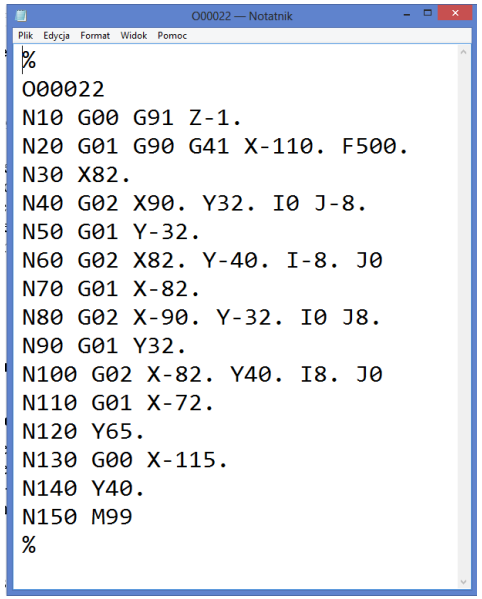
It uses simple G-codes (or other standard codes) and standard cycles manufactured by the producer of CNC controller + **advanced functions and instructions of CNC controller (VARIABLES, CONDITIONS, LOOPS, „JUMPS”, REPETITIONS, ETC.)**

Drawing of a part

Catalogues of cutting tools

Manuals of CNC controllers

Manuals of machine tools

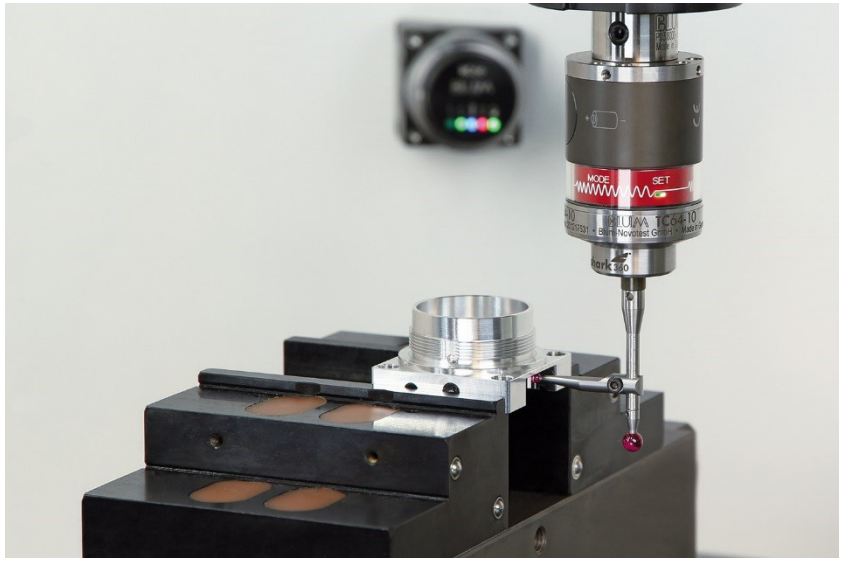


Editing the NC programs or subprograms files

Sending or writing it in the memory of CNC controller

Testing program and starting production

Applications of parametric programming

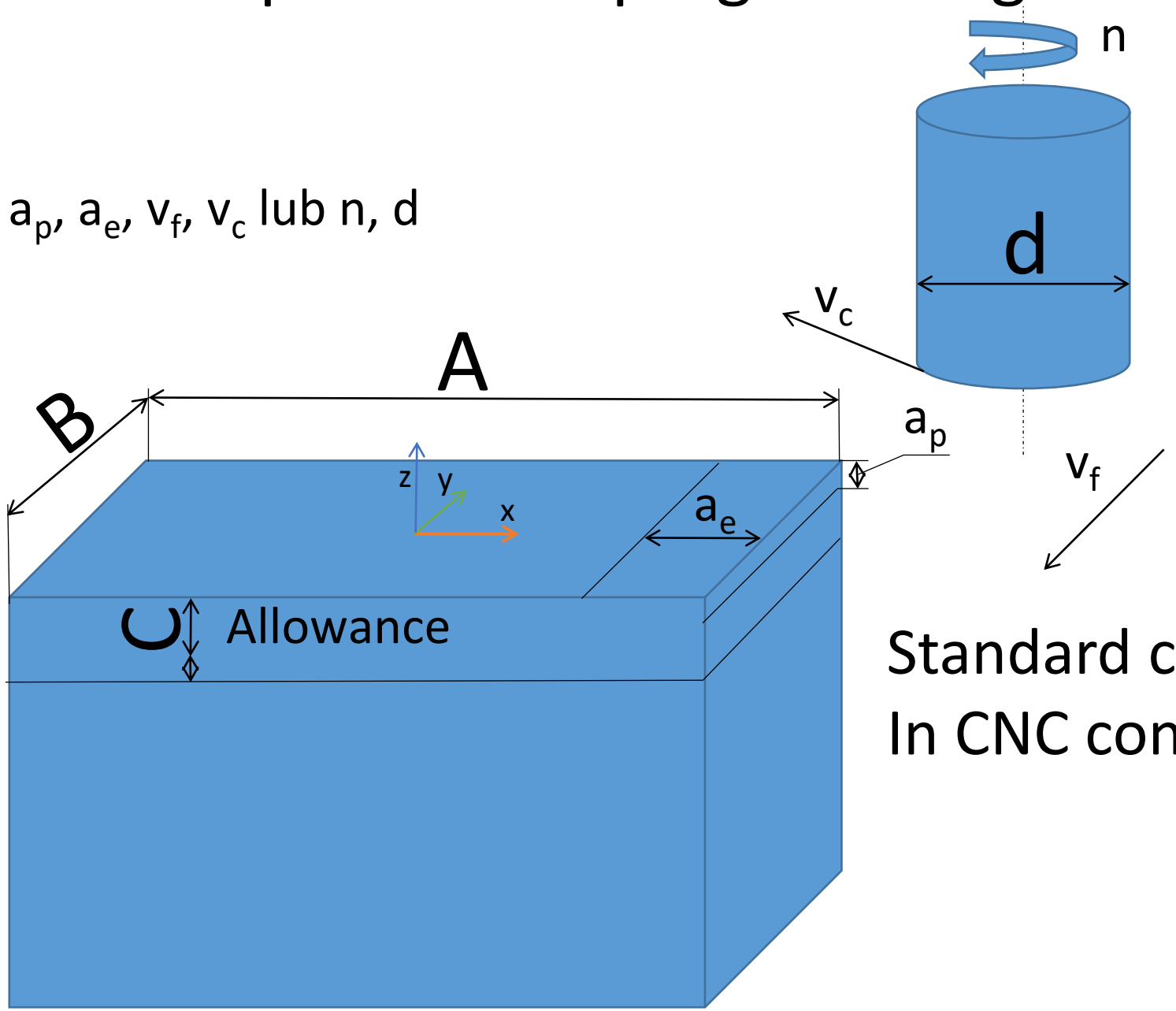


Programming of probes

Applications of parametric programming

Data:

A, B, C, a_p, a_e, v_f, v_c lub n, d



Standard cycles
In CNC controllers

Applications of parametric programming

#3=sin(#2+#1)

Calculations inside programs

What does parametric programming mean?

- If you add:

a) the variables, b) loops, c) conditional expressions, d) jumps

to the standard program (G-code) you can say that it is advanced direct programming (**parametric programming, macros programming**).

The areas for using this type of programming

- Programming of special cycles dedicated for measuring probes
- Creating programs for parts which have the similar geometry
- Creating a standard cycles of machine controllers
- Setting and changing the tool offsets, work offsets, recording machine data
- Creating the programs for research activity
- For calculations

The types of variables:

- Local variables
- General purpose
- System variables

Examples of Macro Variables

Variables	Usage
#0	Not a number (read only)
#1-#33	Macro call arguments
#100-#155	General-purpose variables saved on power off
#156-#199	Used by the probe (if installed)
#500-#549	General-purpose variables saved on power off
#556-#599	Probe calibration data (if installed)
#600-#699	General-purpose variables saved on power off
#700-#749	Hidden variables for internal use only
#800-#999	General-purpose variables saved on power off

* Reference: www.haascnc.com (Haas Mill Operator's Manual)

Iteration/Looping (WHILE DO END)

Essential to all programming languages is the ability to execute a sequence of statements a given number of times or to loop through a sequence of statements until a condition is met. Traditional G coding allows this with the use of the L address. A subprogram can be executed any number of times by using the L address.

```
M98 P2000 L5 ;
```

This is limited since you cannot terminate execution of the subprogram on condition. Macros allow flexibility with the WHILE-DO-END construct. For example:

```
%
WHILE [<conditional expression>] DOn ;
<statements> ;
ENDn ;
%
```

* Reference: www.haascnc.com (Haas Mill Operator's Manual)

Conditional Expressions

In the Haas control, all expressions set a conditional value. The value is either 0.0 (FALSE) or the value is nonzero (TRUE). The context in which the expression is used determines if the expression is a conditional expression. Conditional expressions are used in the `IF` and `WHILE` statements and in the `M99` command. Conditional expressions can make use of Boolean operators to help evaluate a `TRUE` or `FALSE` condition.

The `M99` conditional construct is unique to the Haas control. Without macros, `M99` in the Haas control has the ability to branch unconditionally to any line in the current subprogram by placing a `P` code on the same line. For example:

```
N50 M99 P10 ;
```

branches to line `N10`. It does not return control to the calling subprogram. With macros enabled, `M99` can be used with a conditional expression to branch conditionally. To branch when variable `#100` is less than 10 we could code the above line as follows:

```
N50 [#100 LT 10] M99 P10 ;
```

In this case, the branch occurs only when `#100` is less than 10, otherwise processing continues with the next program line in sequence. In the above, the conditional `M99` can be replaced with

```
N50 IF [#100 LT 10] GOTO10 ;
```

* Reference: www.haascnc.com (Haas Mill Operator's Manual)

Function	Argument	Returns	Notes
SIN[ ]	Degrees	Decimal	Sine
COS[ ]	Degrees	Decimal	Cosine
TAN[ ]	Degrees	Decimal	Tangent
ATAN[ ]	Decimal	Degrees	Arctangent Same as FANUC ATAN[ ]/[1]
SQRT[ ]	Decimal	Decimal	Square root
ABS[ ]	Decimal	Decimal	Absolute value
ROUND[ ]	Decimal	Decimal	Round off a decimal
FIX[ ]	Decimal	Integer	Truncate fraction
ACOS[ ]	Decimal	Degrees	Arc cosine
ASIN[ ]	Decimal	Degrees	Arcsine
#[ ]	Integer	Integer	Variable Indirection
DPRNT[ ]	ASCII text	External Output	

* Reference: www.haascnc.com (Haas Mill Operator's Manual)

Logical Operators

Logical operators are operators that work on binary bit values. Macro variables are floating point numbers. When logical operators are used on macro variables, only the integer portion of the floating point number is used. The logical operators are:

`OR` - logically OR two values together

`XOR` - Exclusively OR two values together

`AND` - Logically AND two values together

* Reference: www.haascnc.com (Haas Mill Operator's Manual)

Boolean Operators

Boolean operators always evaluate to 1.0 (TRUE) or 0.0 (FALSE). There are six Boolean operators. These operators are not restricted to conditional expressions, but they most often are used in conditional expressions. They are:

EQ - Equal To

NE - Not Equal To

GT - Greater Than

LT - Less Than

GE - Greater Than or Equal To

LE - Less Than or Equal To

* Reference: www.haascnc.com (Haas Mill Operator's Manual)

Arithmetic Expressions

An arithmetic expression is any expression using variables, operators, or functions. An arithmetic expression returns a value. Arithmetic expressions are usually used in assignment statements, but are not restricted to them.

Examples of Arithmetic expressions:

```
%
#101=#145*#30 ;
#1=#1+1 ;
X[#105+COS[#101]] ;
#[#2000+#13]=0 ;
%
```

* Reference: www.haascnc.com (Haas Mill Operator's Manual)

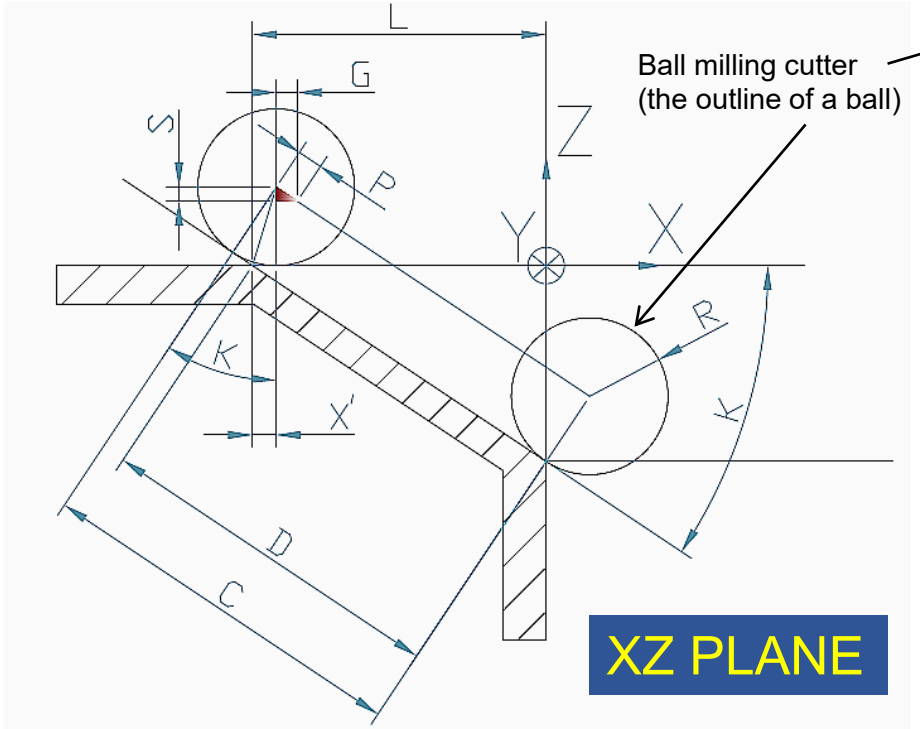
Examples of Macro Variables

Variables	Usage
#0	Not a number (read only)
#1-#33	Macro call arguments
#100-#155	General-purpose variables saved on power off
#156-#199	Used by the probe (if installed)
#500-#549	General-purpose variables saved on power off
#556-#599	Probe calibration data (if installed)
#600-#699	General-purpose variables saved on power off
#700-#749	Hidden variables for internal use only
#800-#999	General-purpose variables saved on power off

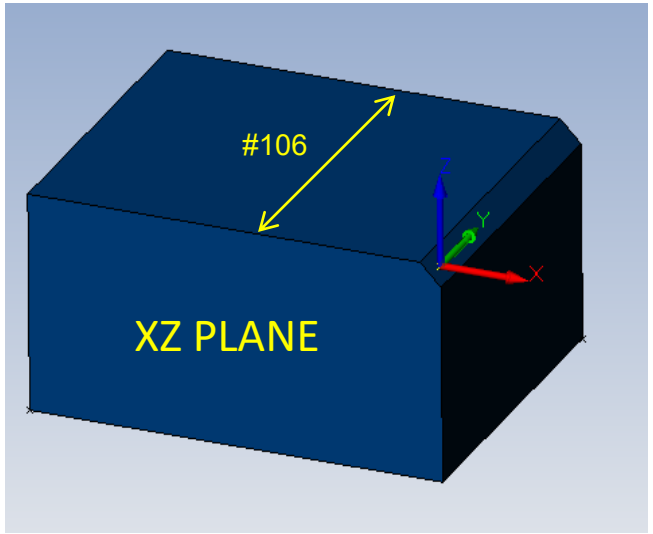
* Reference: www.haascnc.com (Haas Mill Operator's Manual)

Chamfer milling – something more advanced and very interesting

Example of macro program



Ball milling cutter
(the outline of a ball)



Calculations:

X'/R=TAN(K/2) --> X=R*TAN(K/2)

x_{ps}= 0-L+X'=R*tan(K/2)-L- starting point in X axis
z_{ps}=R- starting point for machining in Z axis
y_{ps}=-R- starting point for machining in Y axis

x_{pe}=R*COS(90-K)=R*SIN(K)- the end point of machining in X axis
z_{pe}=R*SIN(90-K)=R*COS(K)- the end point of machining in Z axis

L/D=COS(K)--> D=L/COS(K)
C=D+X'=L/COS(K) + R*TAN(K/2)

S=P*SIN(K) --> P=S/SIN(K)
G=P*COS(K) --> P=G/COS(K)

```
%
o22
(CHAMFER MILLING)
#101=4 (RADIUS OF CUTTER R)
#102=30.(K ANGLE)
#103=25.(L LENGTH)
#105=0.1 (P INCREMENT)
#106=50. (CHAMFER LENGHT IN Y AXIS)
N10 G90 G17 G94
N20 T1 M06
N30 G43 H01
N40 S2000 M3
N50 G0 X[#101*TAN[#102/2]-#103] Z[#101+0.5] Y-#101
N60 G1 Z#101 F100.
N70 #104=0
N80 WHILE[#104 LT #103/COS[#102]+#101*TAN[#102/2]] DO1
G91 G1 X[#105*COS[#102]] Z-[#105*SIN[#102]]
G1 Y[#106+2*#101]
G91 G1 X[#105*COS[#102]] Z-[#105*SIN[#102]]
G1 Y-[#106+2*#101]
#104=#104+2*#105
END1
N90 G0 Z100.
N100 M30
%
```

WHILE INSTRUCTION
Example of a loop

RZESZOW UNIVERSITY OF TECHNOLOGY

THE FACULTY OF MECHANICAL ENGINEERING AND AERONAUTICS
RZESZOW UNIVERSITY OF TECHNOLOGY

Thank you for your attention

- dr inż. Roman Wdowik
- e-mail: rwdowik@prz.edu.pl