Marian KRÁLIK
Dariusz PROSTAŃSKI
Peter MICHALIK
Witold BIAŁY

PROGRAMMING OF PRODUCTION AND MANIPULATION TECHNOLOGY

MONOGRAPH

KOMAG

2024

Marian KRÁLIK
Dariusz PROSTAŃSKI
Peter MICHALIK
Witold BIAŁY

PROGRAMMING of PRODUCTION and MANIPULATION TECHNOLOGY

MONOGRAPH

KOMAG 2024

Marian KRÁLIK
Dariusz PROSTAŃSKI
Peter MICHALIK
Witold BIAŁY

PROGRAMMING of PRODUCTION and MANIPULATION TECHNOLOGY

MONOGRAPH

GLIWICE 2024

Reviewers

Pavol BOŽEK, Slovak University of Technology in Bratislava, Slovakia

Jarosław BRODNY, Silesian University of Technology, Poland

Cover design

Michał ZASADZIEŃ

Publisher

KOMAG Institute of Mining Technology
ul. Pszczyńska 37, 44-101 Gliwice, Poland

ISBN 978-83-65593-37-5

DOI:
https://doi.org/10.32056/KOMAG/Monograph2024.1

Contents

Foreword **9**
1. Flexible Manufacturing Systems – FMS **11**
2. Machining Processes **13**
2.1 Material Removal Processes 13
2.2 Types of Machining Process 13
2.3 Material Removal Processes 13
2.4 Machine Tool Motions 14
3. Turning Operations **16**
3.1 Workpieces made by turning 19
3.2 Lathe Components 20
Bed 20
Headstock assembly 20
Tailstock assembly 20
Carriage 21
Cross slide 21
Compound 21
Three-Jaw Chuck 21
3.3 Cutting Tools 22
3.4 Identification and application of cutting tools for turning 22
Anatomy of a turning tool 22
Insert Shape 23
Clearance 24
Tolerance 24
Type 24
Size 25
Thickens 25
Corner 25
Insert Materials 26
Seat 27
Tool holders 27
Feeds and Speeds 27
Feed Rate 29
Selection of Tools, Feeds, and Speeds 29
4. CNC Machine Tools **30**
4.1 Introduction to Computer Numerical Control 30
Definition 30
4.2 Numerical control definition, its concepts and advantages 31
4.3 Definition of Computer Numerical Control and its components 32
Scheme of CNC machine tool 33
Advantages of CNC compared with NC 34
Special Requirements for Utilizing CNC 35
Schemes of work CNC machining machine 35
Features of the machine control unit (MCU) – machining centers 36
Coordinate system of the machine 36
5. CNC Machining Centers and Turning Centers **38**
6. Modern Machine Tool Controls **40**
6.1 Point-to-Point Tool Movements 40
6.2 Continuous Path Tool Movements 40

6.3 Interpolation 41
Linear Interpolation 41
The Principle of Linear Interpolation 42
Circular Interpolation 43
The Principle of Circular Interpolation 43
6.4 Loop Systems for Controlling Tool Movement 44
Open Loop Systems 44
Closed Loop Systems 44
Precision Position Feedback Detector 46
6.5 Controlling Backlash in CNC Systems 49
6.6 Types of Tool Positioning Modes 49
Incremental Positioning 50
Absolute Positioning 50
7. Programming format, interpolation 51
7.1 Programming Format 51
7.2 Word Address Format 51
7.3 Codes 51
8. Tooling Systems 56
8.1 Tools for machining centers 56
8.2 Tool identify 57
8.3 Automatic Tool Changer Systems 59
8.4 Pallet Loading Systems 61
9. Programming of CNC Lathe – EMCO WinNC SINUMERIK 810/820 T 64
Machine Control Keys 64
Screen with Softkeys 65
Important Points on a CNC Machine Tool 66
Zero Offset 66
Coordinate System 67
Tool Data Measuring 68
Cutter position (Type) 69
Input of the Tool Data 69
Tool Data Measuring with the Optical Presetting Device 71
Survey Modes 71
Input of Programs 71
10. Programming 73
Program Structure 73
Addresses 73
Survey of G Commands 73
Survey of M Commands 74
Survey of Cycles 74
10.1 Description of G Commands 75
G00 Positioning (Rapid Traverse) 75
G01 Linear Interpolation 75
G02 Circular Interpolation Clockwise 76
G03 Circular Interpolation Counterclockwise 76
G04 Dwell 76
G33 Thread Cutting 77
Cutter Radius Compensation 77
G40 Cancel Cutter Radius Compensation 78
G41 Cutter Radius Compensation Left 78
G42 Cutter Radius Compensation Right 78

G70 Measuring in Inches 79
G71 Measuring in Millimeters 79
G90 Absolute Programming 79
G91 Incremental Programming 79
G92 Spindle Speed Limit 80
G94 Feed per Minute 80
G95 Feed per Revolution 80
G96 Constant Cutting Speed 80
G97 Constant Spindle Speed 80
10.2 Description of M Commands 80
M00 Programmed Stop Unconditional 80
M01 Programmed Stop Conditional 80
M02 Main Program End 81
M03 Main Spindle ON Clockwise 81
M04 Main Spindle ON Counterclockwise 81
M05 Main Spindle Off 81
M08 Coolant ON 81
M09 Coolant OFF 81
M17 Subroutine End 81
M30 Main Program End 81
10.3 Description of Cycles 81
L93 Cut-in Cycle 81
L95 Stock Removal with Back Pockets 82
L96 Stock Removal without Back Pockets 82
L97 Thread Cutting Cycle 84
L98 Deep Hole Drilling Cycle 84
10.4 Subroutines 85
10.5 Illustrative example 86
11. CNC Programming For Milling 90
11.1 Machine Basics 90
Machine Axes 90
Machine Zero Point 90
Work Part Zero Point 91
Axis During Machining 91
11.2 Tool Compensation for CNC Milling 92
11.3 NC Programming Basics 92
Standard Formatting 92
M-Codes 93
G-CODES 94
The Absolute System (G90) and Incremental System (G91) 95
Coordinate System Setting G92, G54-G59 96
Tool Length Offset - G43 97
12. TNC 640 Heidenhain 98
12.1 TNC 640 98
12.2 Selecting the Correct Operating Mode 98
12.3 Machine Switch – on 99
12.4 Controls and displays 99
12.5 Programming of the First Part 102
Definition a Workpiece Blank 102
Program Layout 103
Programming a Simple Contour 104

Fundamental of Path Functions 106
Creating a Cycle Program 110
Graphically Testing the First Part 112
Entering Tool – Related Data 114
Requirements for Tool Compensation 116
Tool compensation 118
Programming Contours 123
Tool movements 123
12.6 Approaching and Departing a Contour 124
12.7 Overview of Part Functions 130
12.8 Subprograms and Program Section Repeats 135
12.9 An example of manufacturing a part 138
13. Robotics in manufacturing process 140
13.1 Introduction 140
13.2 Industrial robot structure 140
Classification of robots 141
Degree of freedom 141
Kinematic pairs (KP) 141
13.3 Classification of kinematic structures according to structural layout 142
Accuracy and repeatability 148
Kinematics of robots 149
Homogeneous Transforms 149
Mathematical description of objects 151
Mathematical description of objects – illustrative example 153
Transformation with homogeneous matrix – illustrative example 155
Transformation along the kinematic chain 156
The Denavit-Hartenberg Convention 157
Mechanical section 159
A robot control system 160
Programming unit 161
13.4 Kinematics 161
13.5 Control of robots 162
13.6 Motion control of industrial robots 163
13.7 Interpolations 165
13.8 Robots Programming 167
On-line programming 167
Coordinate system 170
Basics of motion programming 172
13.9 PTP movement 173
Approximation of PTP movement 173
Programming the movement of PTP in an inline form 173
13.10 LIN movement 174
Approximation of movement LIN 175
LIN motion programming in inline form 175
13.11 Movement of the CIR 176
Approximation of CIR movement 176
Programming the movement of the CIR in the inline form 177
Conclusion 179
Literature 181

Foreword

This study material is an introduction to CNC Machining and Programming is intended to support the essentially practical activity of preparing and proving computer numerical control (CNC) part programs for turning, milling, and drilling. It will be of value to students in a wide range of courses dealing with CNC programming and calculations of all forms, tooling for CNC, and fixturing for CNC whether in a major or related course in a college, university, or industrial organization.

Students undertaking a course of study devoted to part programming will therefore find it necessary to attend training center. The student must also have a good understanding of basic machining techniques, and should ideally have previous experience in turning, milling, and drilling operations. In preparing this text, these fundamental requirements have been borne in mind.

CNC part programming is an absorbing and time-consuming activity it is one of the few areas of study where students complain that time has passed too quickly! Thus, a primary objective of this book is to ensure that limited course time can be used to the best advantage by providing the opportunity to devote as much time as possible to preparing programs and using the associated equipment. Accordingly, an attempt has been made to include sufficient information to provide the student with much of the theoretical knowledge needed to support the more practical elements of study, thereby reducing the time spent on formal lectures and unnecessary note taking. The text also provides the student with the opportunity to study specific aspects of interest or needs.

This text is essentially practical in nature and is intended to provide adequate material for course work. It contains a series of assignments that provide the student with a practical understanding of CNC tooling, processing, and programming by various means.

The second part of this study material is a welcome addition to the increasing volume of literature on robotics too. Although the subject is still in its relative infancy, there has been an almost explosive growth in the application of robots in recent years ranging from relatively simple pick-and-place devices to complex multi-degree-of- freedom machines with computer assisted programming of their functions.

Thus, this book is being published at a particularly opportune time, and it is likely to become a standard reference book for programmer of CNC machine machines and robots.

1

Flexible Manufacturing Systems – FMS

A **flexible manufacturing system** (FMS) is an arrangement of machines.... interconnected by a transport system (Fig. 1.1). The transporter carries work to the machines on pallets or other interface units so that work-machine registration is accurate, rapid, and automatic. A central computer controls both machines and transport system... The key idea in FMS is that the co-ordination of the flow of work is carried out by a central control computer. This computer performs functions such as:

- Scheduling jobs onto the CNC machine tools.
- Downloading part-programs (giving detailed instructions on how to produce a part) to the CNC machine tools.
- Sending instructions to the automated vehicle system for transportation.

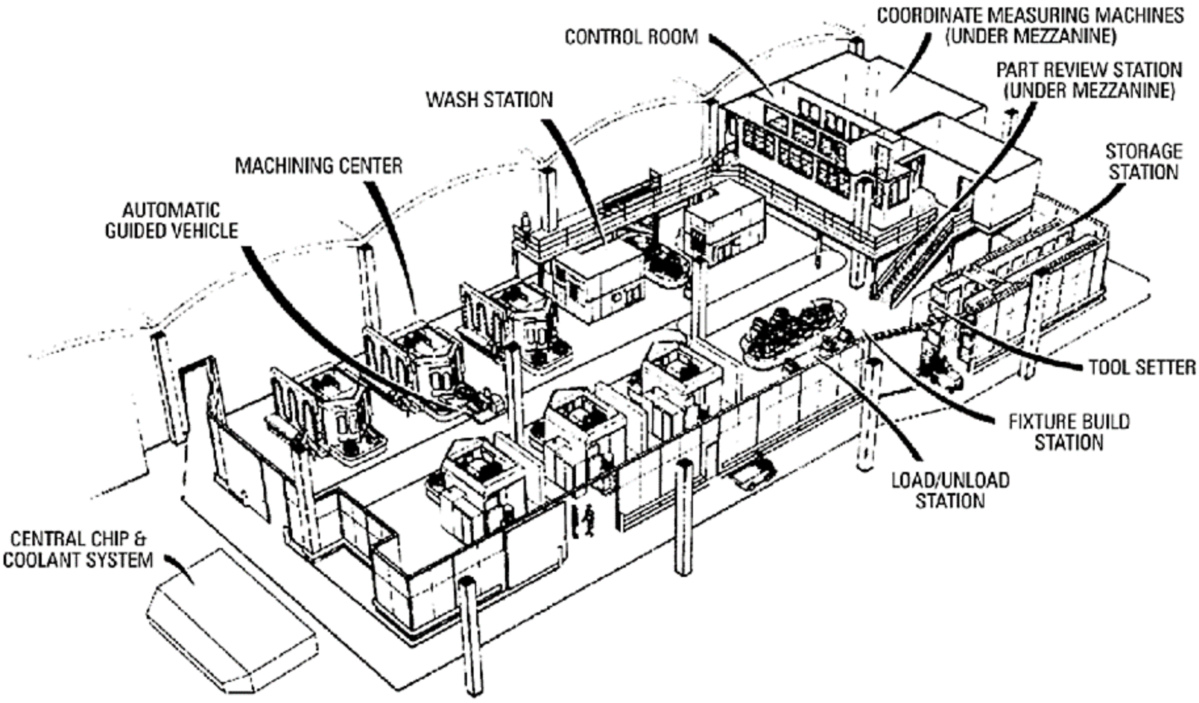


Fig. 0.1 Flexible manufacturing system

Demands FMS:

- Efficiency
- Flexibility (could be reprogrammed for different parts)
- Capability (could process parts requiring many operations from many machines)
- Scope (could make many kinds of parts)
- Automation (could be programmed remotely and operated without people)

In Industry it is not efficient or profitable to make everyday products by hand. On a CNC machine it is possible to make hundreds or even thousands of the same items in a day. First a design is drawn using design software, and then it is processed by the computer

and manufactured using the CNC machine. https://www.youtube.com/watch?v=Br2eEpiiwvU
https://www.youtube.com/watch?v=YGtg4OPSFhc

The scheduling problem is one of the most important problems that largely affects the efficiency of the production systems. As the current trend in manufacturing technology is towards the use of flexible manufacturing systems with CNC machine tools, the scheduling problem in these systems was given a focus through the last few years.

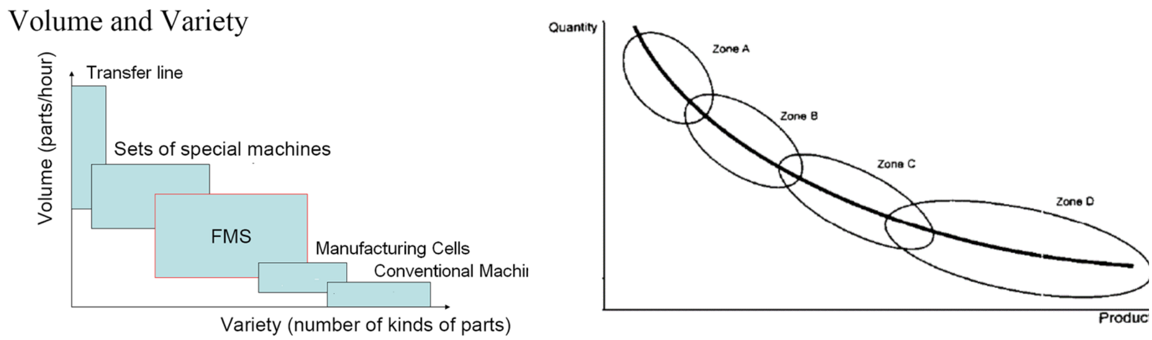


Fig. 0.2 Volume-variety relationship in different production systems

Four different zones are identified in the analysis:

1 - **Zone A** is where a few products of large quantities are produced. This suggests the use of single product line for the products.
2 - **Zone B**, when compared with Zone A, has more variety but the production quantities for individual items are relatively less. Single product lines cannot be justified hence items having similar operation sequences are grouped together in a mullti-product line set up.
3 - In **Zone D**. a wide variety of products are manufactured, and the quantities required of each are much less. Different components follow totally different flow paths and process layout is the preferred choice.
4 - **Zone C** is in the intermediary position between Zone B and Zone D. The parts produced might have similar operations, but the operation sequences could be different. Group technology is the probable solution. **There are used FMS with the CNC machine tools**.

2

Machining Processes

Machining is any of various processes in which a piece of raw material is cut into a desired final shape and size by a controlled material-removal process.

2.1 Material Removal Processes

- Machining is the broad term used to describe removal of material from a workpiece.
- Includes Cutting, Abrasive Processes (grinding), Advanced Machining Processes (electrical, chemical, thermal, hydrodynamic, lasers)
- Automation began when lathes were introduced in 1700s.
- Now have computer numerical control (CNC) machines
- Machining operations are a system consisting of:
    - Workpiece – material, properties, design, temperature.
    - Cutting tool – shape, material, coatings, condition.
    - Machine tool – design, stiffness & damping, structure.
    - Fixture – workpiece holding devices.
    - Cutting parameters – speed, feed, depth of cut.

Independent variables

The three principal machining processes are classified as **turning**, **drilling** and **milling**. Other operations falling into miscellaneous categories include shaping, planing, boring, broaching and sawing.
Turning operations are operations that rotate the workpiece as the primary method of moving metal against the cutting tool. Lathes are the principal machine tool used in turning.
Milling operations are operations in which the cutting tool rotates to bring cutting edges to bear against the workpiece. Milling machines are the principal machine tool used in milling.
Drilling operations are operations in which holes are produced or refined by bringing a rotating cutter with cutting edges at the lower extremity into contact with the workpiece. Drilling operations are done primarily in drill presses but sometimes on lathes or mills [12].

2.2 Types of Machining Process

- Single Cutting Edge (Point) Processes
- Multi-Cutting Edge (Point) Processes
- Random Point Cutting Processes –Abrasive Machining
- Within each category the basic motions (kinematics) differentiate one process from another

2.3 Material Removal Processes

- Material removal processes are often required after casting or forming to:
    - Improve dimensional accuracy.

- Produce external and internal geometric features, sharp corners, or flatness not possible with forming or shaping.
  - Obtain final dimensions and surfaces with finishing operations.
  - Obtain special surface characteristics or textures.
  - Provide the most economical means of producing a particular part.
- Limitations, because material removal processes:
  - Inevitably waste material.
  - Generally, require more energy, capital, and labor than forming or shaping operations.
  - Can have adverse effects on the surface quality and properties, unless carried out properly,
  - Generally, take longer than shaping a product with other processes

2.4 Machine Tool Motions

Turning machines, typically referred to as lathes, can be found in a variety of sizes and designs. While most lathes are horizontal turning machines, vertical machines are sometimes used, typically for large diameter workpieces. Turning machines can also be classified by the type of control that is offered (Fig. 2.1, 2.2).

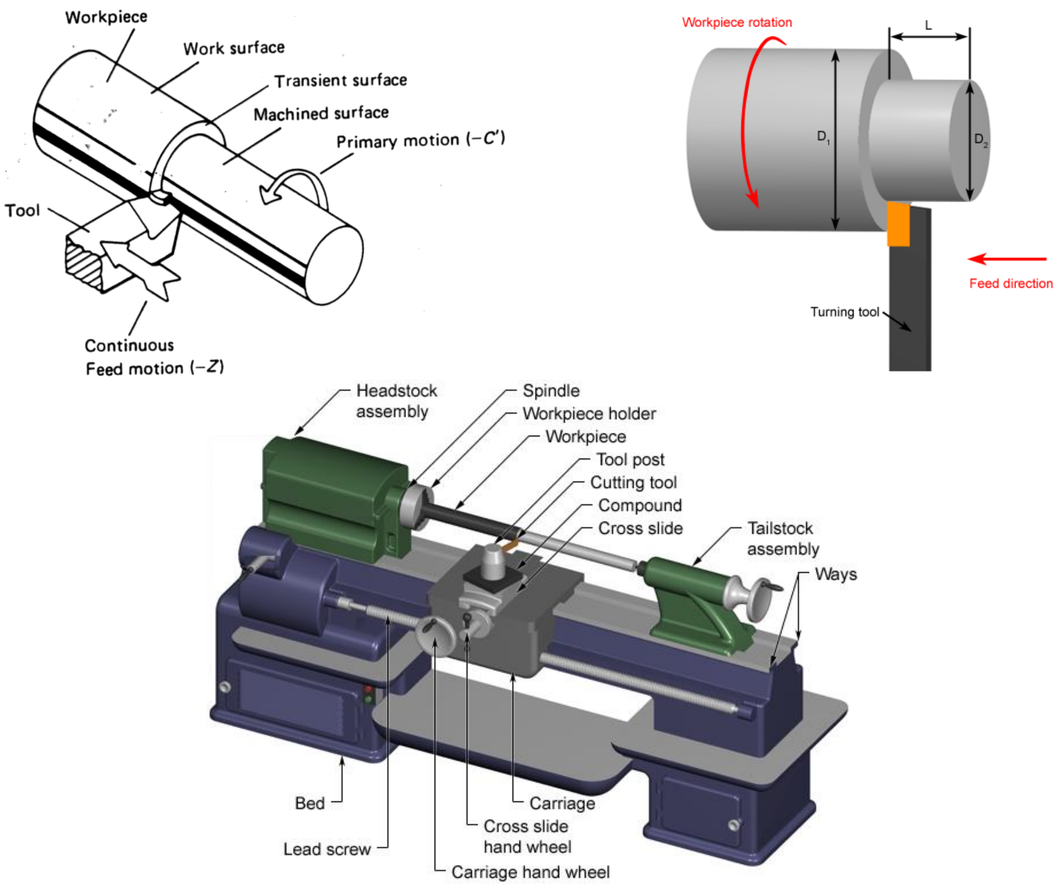


Fig. 0.1 Machine tool motions by turning and lathe (turning machine tool)

A manual lathe requires the operator to control the motion of the cutting tool during the turning operation. Turning machines are also able to be computer controlled, in which case they are referred to as a computer numerical control (CNC) lathe. CNC lathes rotate the workpiece and move the cutting tool based on commands that are preprogrammed and offer very high precision. In this variety of turning machines, the main components that enable the workpiece to be rotated and the cutting tool to be fed into the workpiece remain the same [1], [12].

- Primary motion that causes cutting to take place.
- Feed motion that causes more of the part surface to be machined.
- Rotations and/or translations of the workpiece or cutting tool [3].

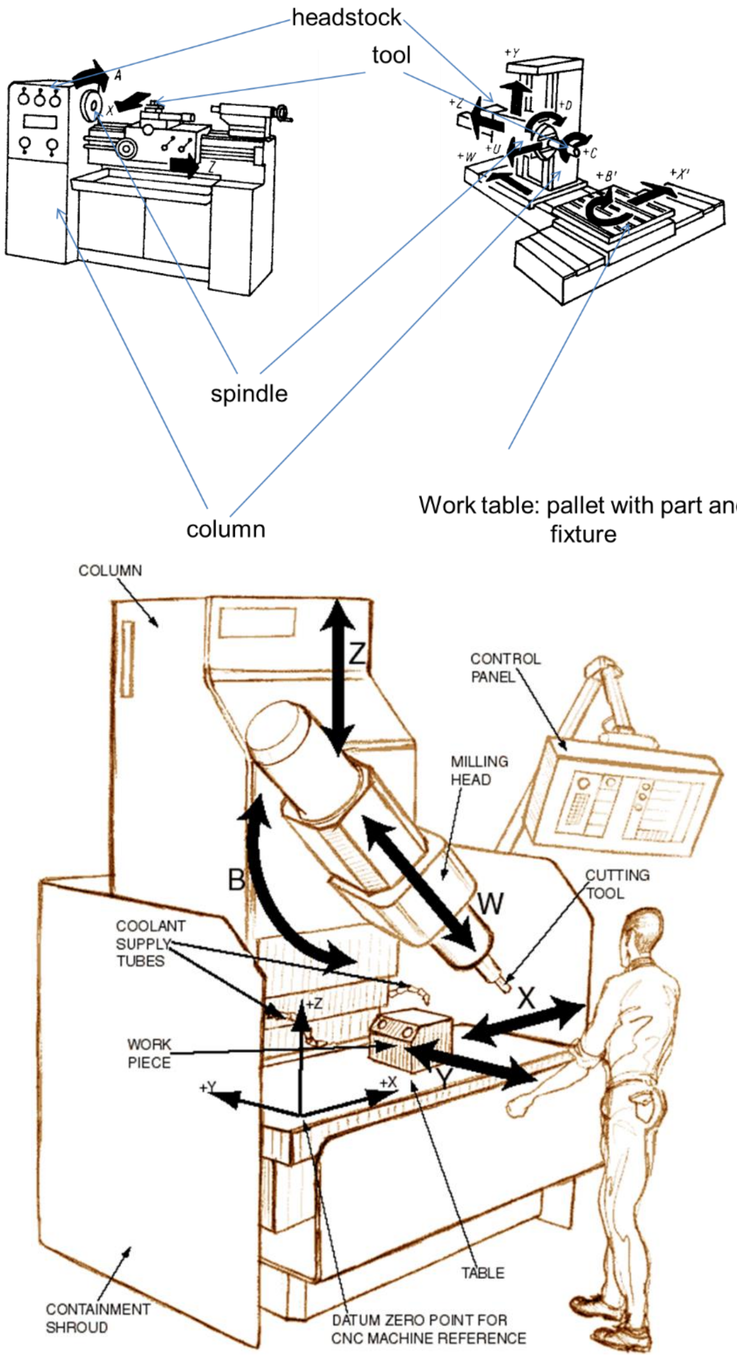
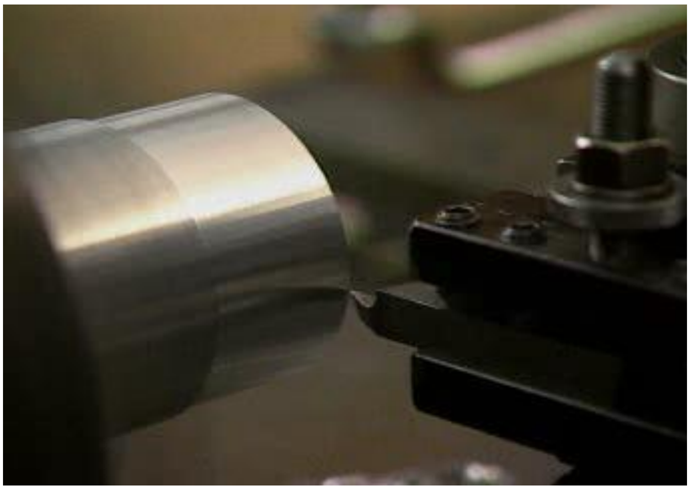


Fig. 0.2 Motions in Axes on the CNC Machine Tools

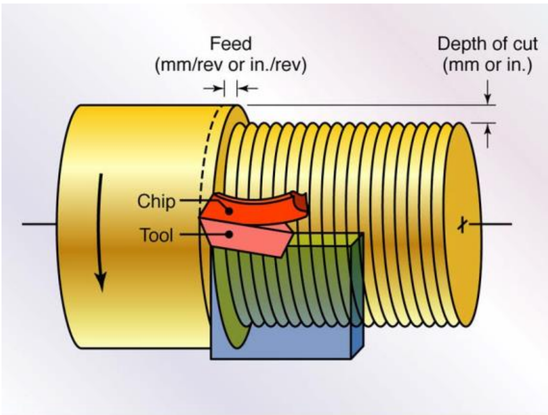
3

Turning Operations

This operation is one of the most basic machining processes. That is, the part is rotated while a single point cutting tool is moved parallel to the axis of rotation. Turning can be done on the external surface of the part as well as internally (boring). The starting material is generally a workpiece generated by other processes such as casting, forging, extrusion, or drawing.
Primary motion is rotation of the workpiece. Feed motion is translation of the cutting tool – continuous driven by spindle rotation.



https://www.youtube.com/watch?v=8EsAxOnzEms,
https://en.wikipedia.org/wiki/Speeds_and_feeds



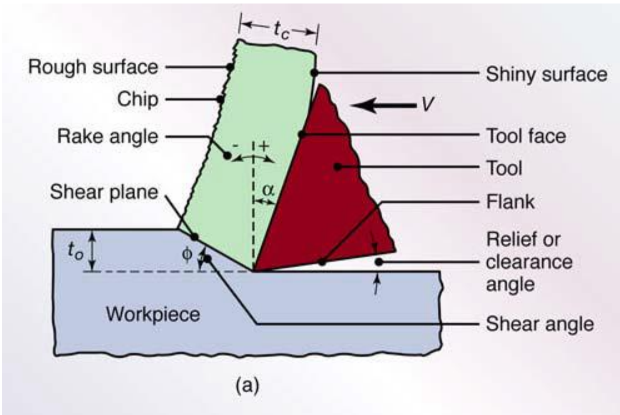


Fig. 0.1 Principles of metal cutting [12]

To machine, the tool or workpiece must rotate or move in a linear direction [2].
To remove material, the cutting tool must engage the workpiece. The rate of speed at which this engagement occurs is called ***Feed-Rate***. Feed-Rate controls the thickness of the chip and is a major factor in the type of chip that is formed [3].

Depth of cut, also known as in-feed is the depth of material set to be removed by the cutting tool.

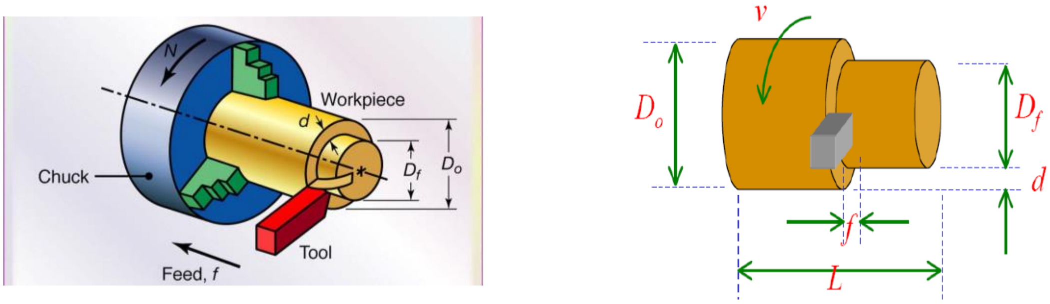


Fig. 0.2 Schematic illustration of the basic turning operation, showing depth-of cut (d) [mm] or [in]; feed, (f); [mm/rev] or [in/rev] and spindle rotational speed N [rev/min]. Cutting speed is the surface speed of the workpiece at the tool tip V [m/min] or [ft/min]

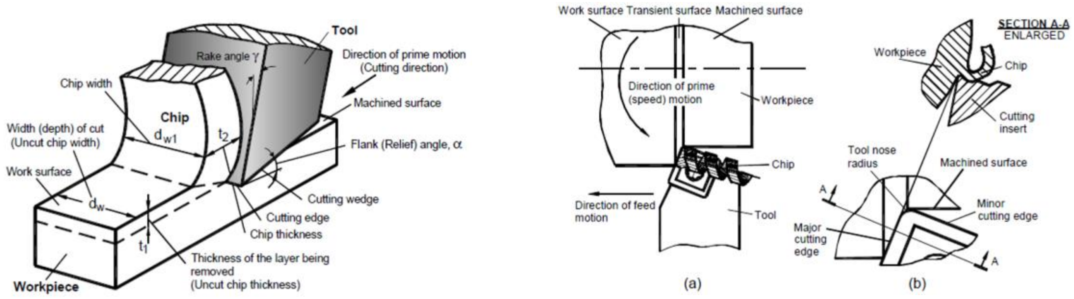


Fig. 0.3 Visualization of basic terms in orthogonal cutting and visualization of basic terms in turning: (a) general view and (b) enlarged cutting portion.

Captions for the Figure 3.4:

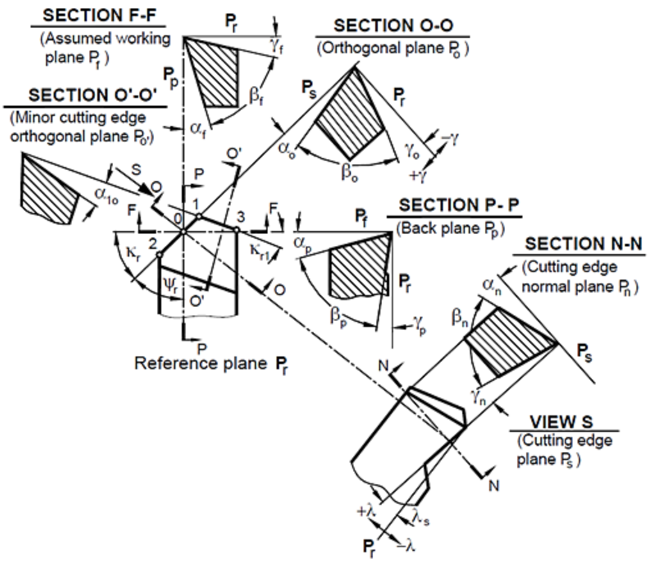


Fig. 0.4 System of tool angles in the defined planes in T-hand-S [12]

A_α – flank, A_γ- face, P_r – tool reference plane, P_f – assumed working plane, P_s – tool cutting edge plane, P_n – cutting edge normal plane, P_o – tool orthogonal plane, S – major cutting edge, S´ – minor cutting edge, corner, selected point of the cutting edge

Tool angles and parameters:

α_o –tool orthogonal clearance, γ_o – tool orthogonal rake, χ_r – tool cutting edge angle, χ'_r – tool approach angle, ε_r – tool included angle, β_o – tool orthogonal wedge angle, λ_s – tool cutting edge inclination, r_ε – corner radius, r_n – rounded cutting-edge radius.
https://www.youtube.com/watch?v=bUrp8JMRwx4,https://www.youtube.com/watch?v=Mn9jpqI8rao, https://www.youtube.com/watch?v=BHEYrGrvp6U,
https://www.youtube.com/watch?v=J63dZsw7Ia4

Summary of Turning Parameters and Formulas

N = Rotational speed of the workpiece, rpm
f = Feed, mm/rev or in./rev
v = Feed rate, or linear speed of the tool along workpiece length, mm/min or in./min
$= fN$
V = Surface speed of workpiece, m/min or ft/min
$= \pi D_o N$ (for maximum speed)
$= \pi D_{avg} N$ (for average speed)
l = Length of cut, mm or in.
D_o = Original diameter of workpiece, mm or in.
D_f = Final diameter of workpiece, mm or in.
D_{avg} = Average diameter of workpiece, mm or in.
$= (D_o + D_f)/2$
d = Depth of cut, mm or in.
$= (D_o - D_f)/2$
t = Cutting time, s or min
$= l/fN$
MRR = mm^3/min or in^3/min
$= \pi D_{avg} d f N$
Torque = N•m or lb•ft
$= F_c D_{avg}/2$
Power = kW or hp
= (Torque)(ω), where $\omega = 2\pi N$ rad/min

Note: The units given are those that are used commonly; however, appropriate units must be used and checked in the formulas.

Fig. 3.5 Turning formulas

The main formulas:

Rotational speed:	$N = \frac{V}{\pi D_0}$	*[rpm]*
Feed rate:	$f_r = Nf$	*[mm/min]*
Time of machining:	$T_m = \frac{L}{f_r}$	*[min]*
Depth of cut:	$d = (D_0 - D_f)/2$	*[mm]*

Material removal rate: $MRR = Vfd$ $[mm^3/min]$

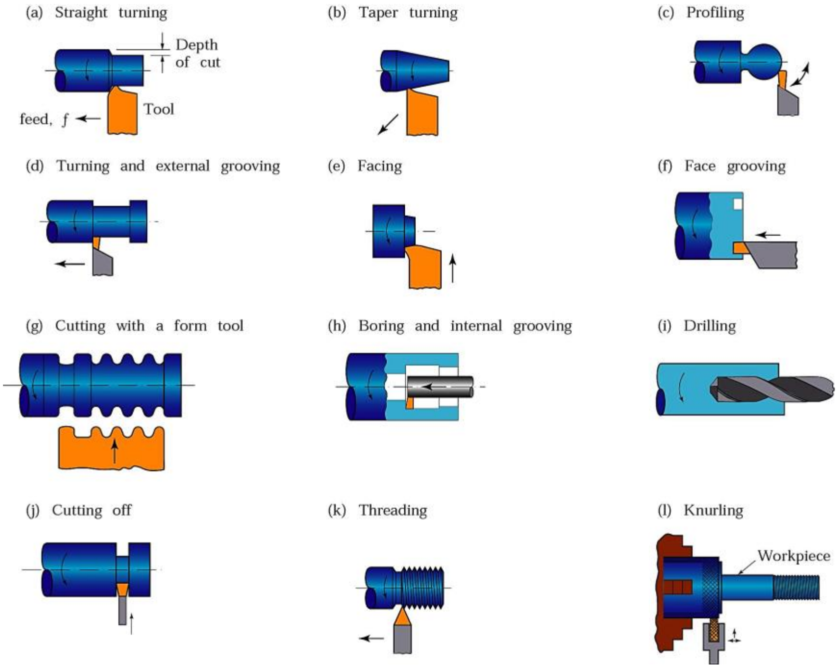


Fig. 3.6 Turning is the process for machining round workpieces on a lathe

3.1 Workpieces made by turning

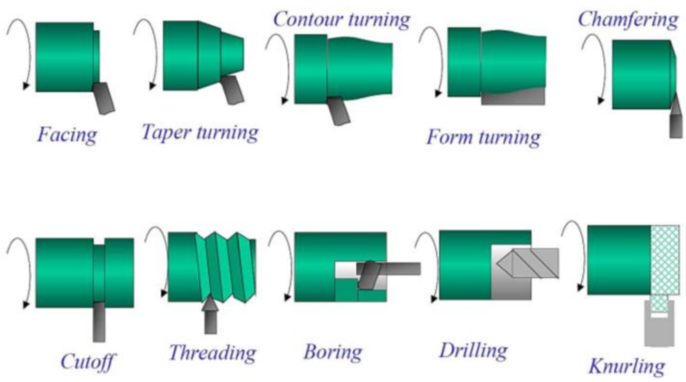


Fig. 3.7 Turning motions

Fig. 3.8 Workpieces made by turning

3.2 Lathe Components

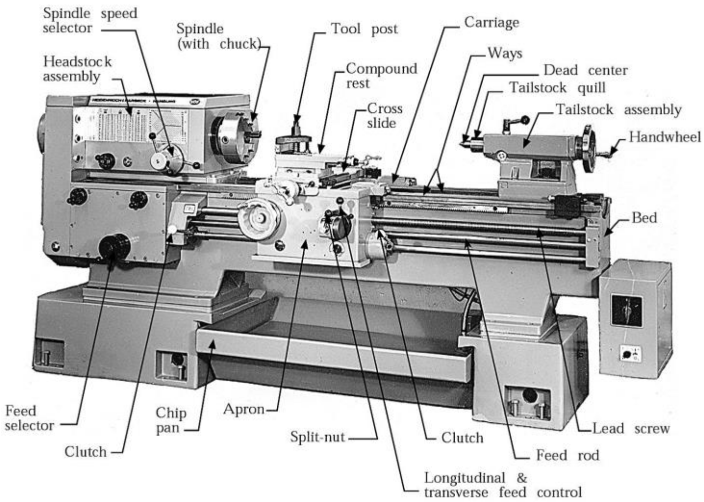
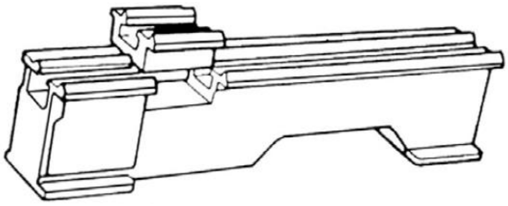


Fig. 3.9 Engine Lathe-the main part of the structure of the machine tool (696) Construction Details and Operation of Different Parts of a Lathe Machine. – YouTube

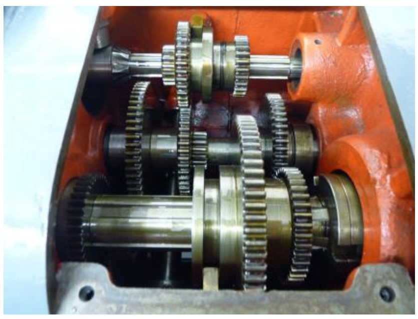
Bed

The bed of the turning machine is simply a large base that sits on the ground or a table and supports the other components of the machine.



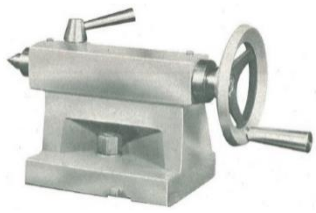
Headstock assembly

The headstock assembly is the front section of machine that is attached to the bed. This assembly contains the motor and drive system which powers the spindle. The spindle supports and rotates the workpiece, which is secured in a workpiece holder or fixture, such as a chuck or collet.

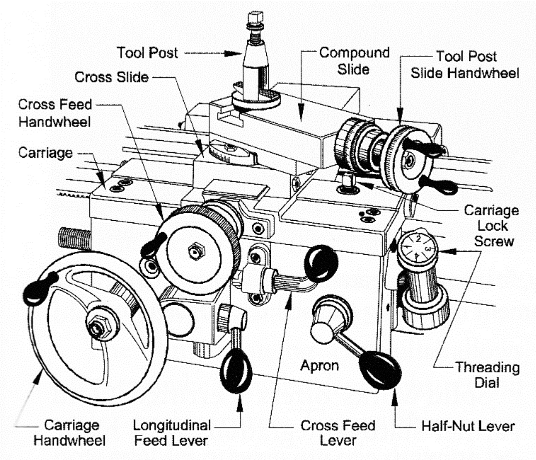


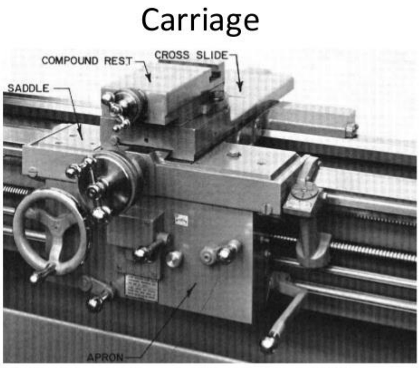
Tailstock assembly

The tailstock assembly is the rear section of the machine that is attached to the bed. The purpose of this assembly is to support the other end of the workpiece and allow it to rotate, as it's driven by the spindle.



Carriage
The carriage is a platform that slides alongside the workpiece, allowing the cutting tool to cut away material as it moves.

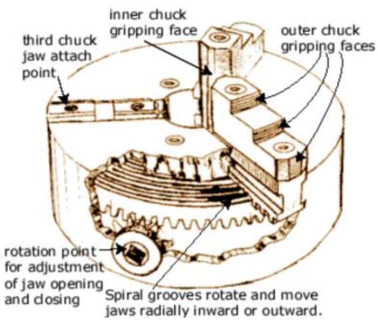




Cross slide
The cross slide is attached to the top of the carriage and allows the tool to move towards or away from the workpiece, changing the depth of cut.

Compound
The compound is attached on top of the cross slide and supports the cutting tool. The cutting tool is secured in a tool post which is fixed to the compound. The compound can rotate to alter the angle of cutting tool relative to the workpiece.

Three-Jaw Chuck
The chuck is integral to a lathe's functioning because it fixtures the part to the spindle axis of the machine. Below is shown a three-jaw chuck with jaws that are all driven by the same chuck key. This arrangement provides convenience in that parts can be mounted and dismounted quickly.



3.3 Cutting Tools

All cutting tools that are used in turning can be found in a variety of materials, which will determine the tool's properties and the workpiece materials for which it is best suited. These properties include the tool's hardness, toughness, and wear resistance. The most common tool materials that are used include the following [12]:

- High-speed steel (HSS)
- Carbide
- Carbon steel
- Cobalt high speed steel

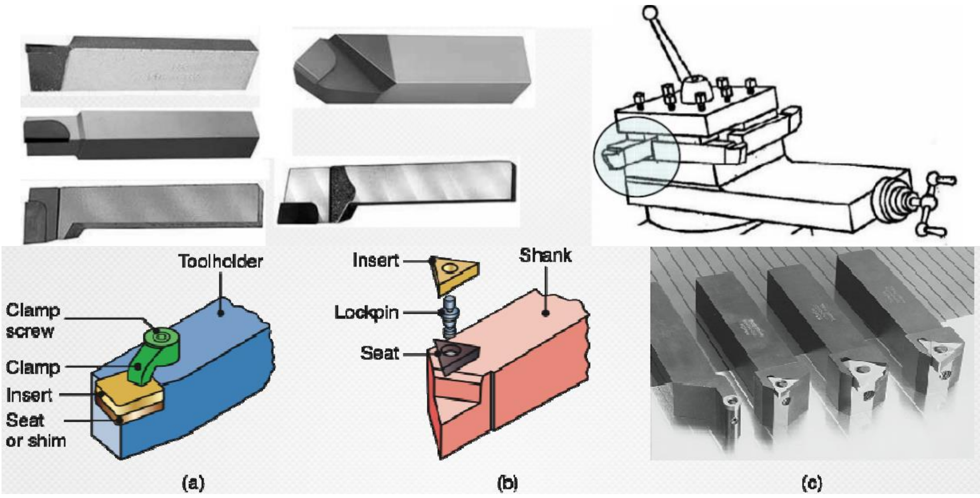


Fig. 3.10 Cutting tool for turning
https://www.youtube.com/watch?v=KSl3dLeeJ5g

Methods of mounting inserts on tool holders:

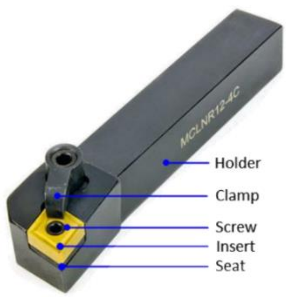
a) Clamping, and
b) Wing lock pins.
c) Examples of inserts mounted using threadless lock pins, which are secured with side screws.

3.4 Identification and application of cutting tools for turning.

The variety of cutting tools available for modern CNC turning centers makes it imperative for machine operators to be familiar with different tool geometries and how they are applied to common turning processes [2], [12].

Anatomy of a turning tool

Most turning is done using a replaceable insert that is gripped in a turning tool body, which is then mounted on the lathe turret.



Standard Inserts

Turning inserts employ highly engineered composite structures, coatings, and geometry features to achieve great accuracy and high material removal rates.

The benefits of using replaceable inserts for turning tools include:

- Some inserts can be indexed to use other edges when one becomes worn.
- Inserts are quickly and easily replaced at the machine.

ANSI Insert Designations

The American National Standards Institute (ANSI) has developed a coding system of numbers and letters to describe the shape, dimensions, and important parameters of turning inserts.

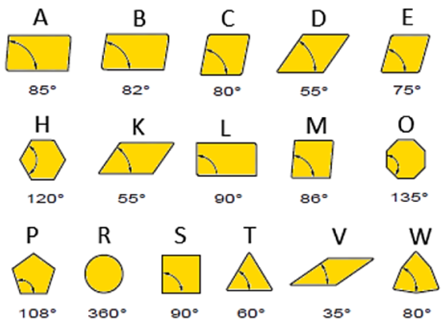
C	N	M	G	-	4	3	2
1	2	3	4		5	6	7
Shape	Clearance	Tolerance	Type		I.C. Size	Thickness	Nose Radius

Fig. 3.11 ANSI indexable inserts identification system

Insert Shape

Turning inserts are manufactured in a variety of shapes, sizes, and thicknesses. The shape can be round to maximize edge strength, diamond-shaped to allow a sharp point to cut fine features, square, or even octagonal to increase the number of separate edges that can be applied as one edge after another wears out.

- A = parallelogram 85°
- B = parallelogram 82°
- C = diamond 80°
- D = diamond 55°
- E = diamond 75°
- H = hexagon
- K = diamond 55°
- L = rectangle
- M = diamond 86°
- O = octagon
- P = pentagon
- R = round
- S = square
- T = triangle
- V = diamond 35°
- W - trigon 80°



C and W type turning inserts are often used for rough machining due to their larger point angle, which makes them more rigid. Inserts with a smaller point angle, such as D and V, are often used for finish machining. Although they have less strength, the smaller angle can reach more part details.

Large point angle:

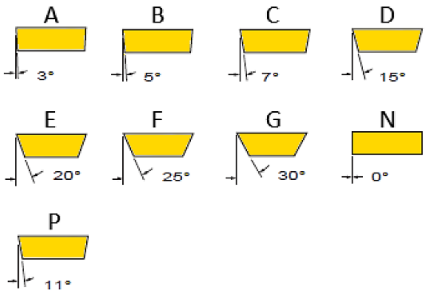
- Stronger cutting edge
- Higher feed rates
- Increased cutting forces.
- Increased vibration

Small point angle:

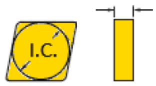
- Weaker cutting edge
- Increased access to part details
- Decreased cutting forces.
- Decreased vibration

Clearance

Most inserts have drafted faces on the walls. Clearance prevents the walls of the insert from rubbing against the part, which will give poor machining. However, a turning insert with a 0° clearance angle is mostly used for rough machining.



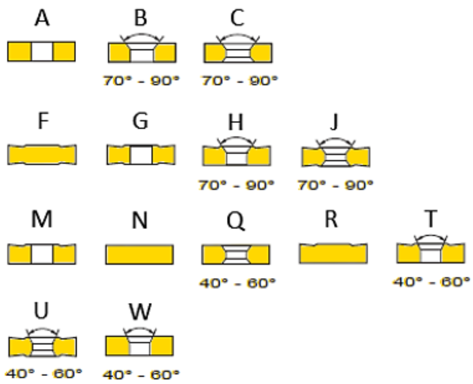
Tolerance

Letter	Inscribed Circle (I.C.)	Thickness	
A	±.0002	±.001	
B	±.0002	±.005	
C	±.0005	±.001	
D	±.0005	±.005	
E	±.001	±.001	
F	±.002 to ±.004	±.002	
G	±.001	±.005	
M	±.002 to ±.010	±.005	
U	±.005 to ±.012	±.005	

Type

The turning inserts hole shape and chip breaker type.

- A = Cylindrical hole
- B = 70-90° countersink hole
- C = Double countersink
- F = No hole; Double-sided chip breaker
- G = Cylindrical; Double-sided chip breaker
- H = 70-90° countersink; Single-sided
- J = Double countersink; Double-sided
- M = Cylindrical; Single-sided
- N = No hole; No chip breaker
- P = Cylindrical; Hi-Double-Positive Chip breaker
- Q = 40-60° Double Countersink; No chip breaker
- R = No hole; Single-sided
- S = Cylindrical; Hi-double positive
- T = 40-60° double countersink; Single-sided
- U = 40-60° double countersink; Double-sided
- W = 40-60° double countersink
- X = Special Design



A chip breaker is a feature in the face of the insert that disrupts the flow of chips such that they break into short segments, rather than forming a long, stringy chip (Fig. 3.12).

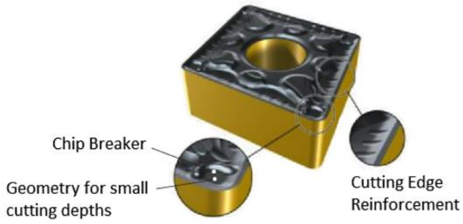


Fig. 3.12 Chip breaker on the insert

Size

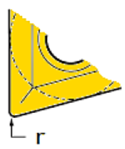
This numeric value tells us the cutting edge length of the turning insert.  	
For equal sided inserts 1/4" I.C. or over: Size = Number of 1/8" increments.	Examples: 2 = 1/4" 3 = 3/8" 4 = 1/2"
For equal sided inserts less than 1/4" I.C.: Size = Number of 1/32" increments.	Examples: 2 = 1/16" 3 = 3/32" 4 = 1/8"
For rectangles and parallelograms, 2 digits are necessary (Width and Length). Digit 1: Width in 1/8" increments Digit 2: Length in 1/4" increments	Example: 12 = 1/8" x 1/2"
Important: The depth-of-cut for roughing should never exceed 1/2 the inscribed circle of the insert.	

Thickens

This value tells us the thickness of the insert.
For inserts 1/4" I.C. or over: Thickness in 1/16" increments.
For inserts less than 1/4" I.C.: Thickness in 1/32" increments.

Corner

Form on the corner in 1/16" increments for inserts with a corner radius. To reduce vibration, it is often an advantage to choose a nose radius that is smaller than the depth of cut.

0 = Sharp corner (.005R) 1 = 1/64" radius 2 = 1/32 3 = 3/64 4 = 1/16 5 = 5/64	6 = 3/32 7 = 7/64 8 = 1/8 10 = 5/32 12 = 3/16 14 = 7/32 16 = 1/4	
Large nose radius: • Increased feed rates • Large depths of cut • Strong edge security • Increased radial pressures	Small nose radius: • Small cutting depths • Reduced vibration • Weak cutting edge	

Insert Materials

Insert material is typically carbide, though ceramic, cermet or diamond inserts can be applied to more demanding applications. A variety of protective coatings also help these insert materials cut faster and last longer.

Insert Material	Characteristics
Cemented carbide (HW, HC) • HW: Uncoated • HC: Coated	The most common material used in the industry today. It is offered in several "grades" containing different proportions of tungsten carbide and binder (usually cobalt). High resistance to abrasion.
Cermets (HT, HC) Cermet containing primarily titanium carbides (TiC) or titanium nitrides (TiN) or both • HT: Uncoated • HC: Coated	Another cemented material, based on titanium carbide (TiC). Binder is usually nickel. It provides higher abrasion resistance compared to tungsten carbide at the expense of some toughness. Extremely high resistance to abrasion.
Ceramics (CA, CM, CN, CC) • CA Oxide ceramics containing primarily aluminum oxide (Al_2O_3) • CM Mixed ceramics containing primarily aluminum oxide (Al_2O_3) but containing components other than oxides • CN Nitride ceramics containing primarily silicon nitride (Si_3N_4) • CC Nitride ceramics containing primarily silicon nitride (Si_3N_4), but coated	Chemically inert and extremely resistant to heat, ceramics are usually desirable in high speed applications, the only drawback being their high fragility. The most common ceramic materials are based on alumina (aluminium oxide), silicon nitride and silicon carbide.
Cubic boron nitrides (BN)	The second hardest substance. It offers extremely high resistance to abrasion at the expense of much toughness. It is generally used in a machining process called "hard machining", which involves running the tool or the part fast enough to melt it before it touches the edge, softening it considerably.
Polycrystalline diamonds (DP, HC) • DP : Uncoated • HC : Coated	The hardest substance. Superior resistance to abrasion but also high chemical affinity to iron which results in being unsuitable for steel machining. It is used where abrasive materials would wear anything else.

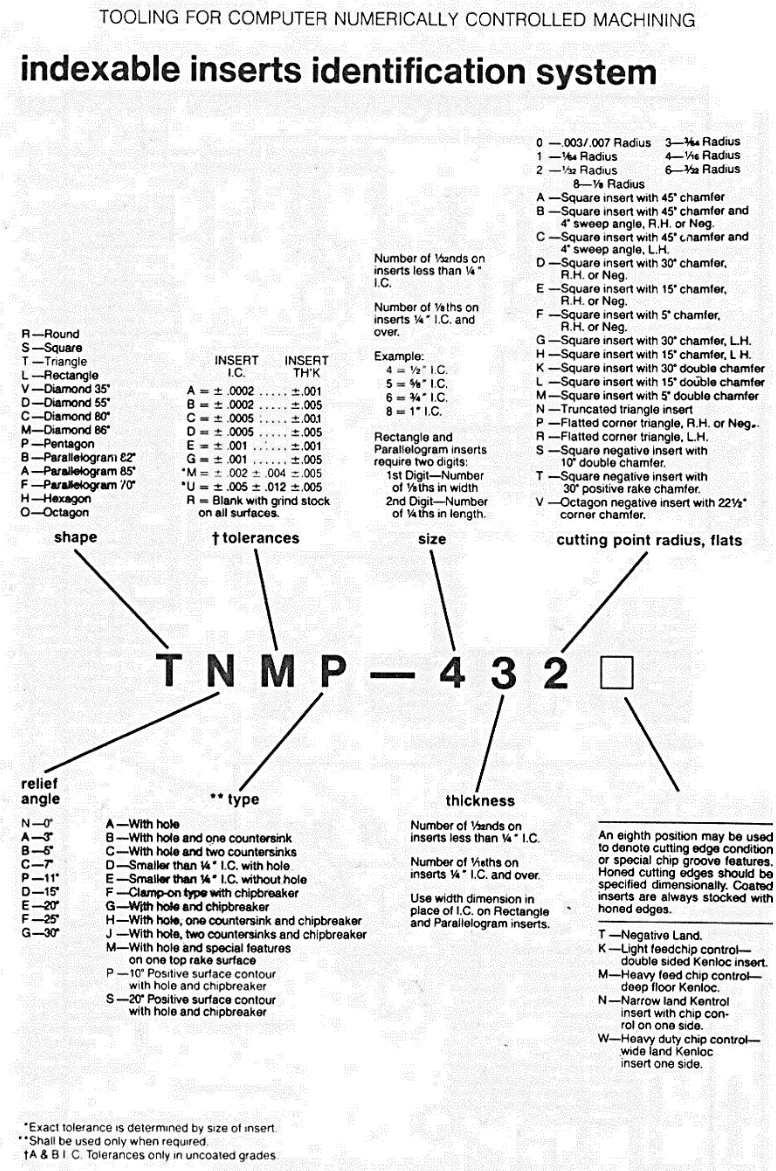


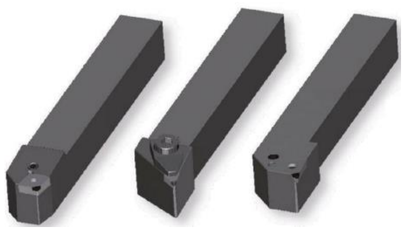
Fig. 3.13 ANSI indexable insert identification system

Seat

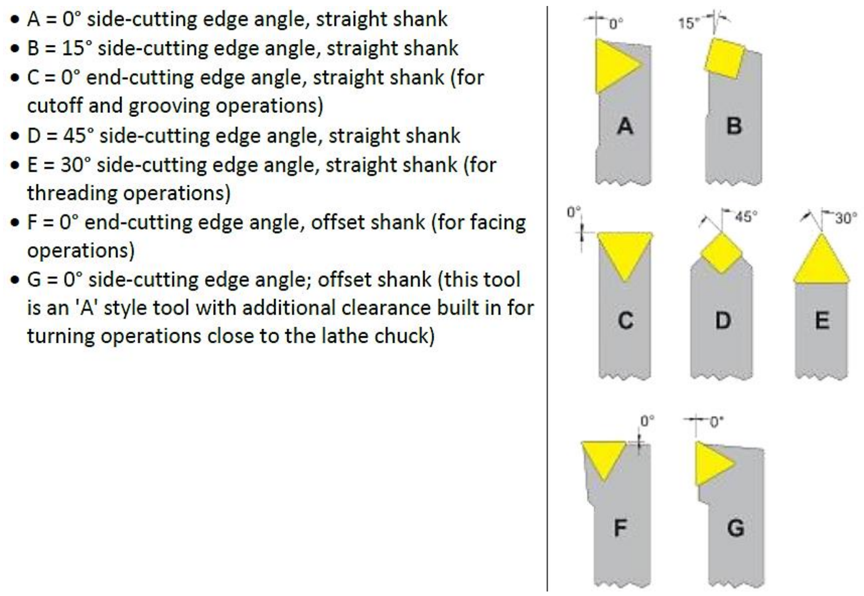
A piece of carbide, the same size as the insert it supports, placed between the insert and the bottom of the pocket where the insert fits in the tool holder.

Tool holders

It is essential that the insert be supported in a strong, rigid manner to minimize deflection and possible vibration. Consequently, turning tools are supported in various types of heavy, forged steel tool holders.



The turning tool body generally does not feature quite so much engineering as the insert, but even here there are a range of choices for fine-tuning the process. Quick-change tools involve modular bodies that allow replacement tool bodies to be swapped in and out and locked in place quickly to minimize setup time. The turning tool body can also channel high-pressure coolant more effectively to the cutting edge of the tool.
The ANSI numbering system for turning tool holders has assigned letters to specific geometries in terms of lead angle and end cutting edge angle. The primary lathe machining operations of turning, facing, grooving, threading and cutoff are covered by one of the seven basic tool styles outlined by the ANSI system.
The designations for the seven primary tool styles are A, B, C, D, E, F and G.



Tip: The most used insert/holder combination for O.D. rough turning and facing is the C type 80° diamond insert with a 3-5° negative lead tool holder. It is often selected because it is the best compromise between strength of insert and end-angle clearance.

Feeds and Speeds

Feeds and speeds refer to two separate velocities for machine tools: feed rate and cutting speed. They are used together because of their combined effect on the cutting process.

- Cut speed is the speed at the outside edge of the part as it is rotating, also known as surface speed.
- Feed rate is the velocity at which the cutter is advanced along the spinning workpiece.

Cutting Speed

Cutting speed is the speed that the material moves past the cutting edge of the tool. Cut speed can be defined as revolutions per minute (RPM) or as surface feet per minute (SFM).
Revolutions Per Minute (RPM) relates directly to the speed, or velocity, of the spindle. It annotates the number of turns completed in one minute around a fixed axis. RPM maintains the same revolutions per minute throughout the entire operation.
RPM mode is useful for:

- Center cutting operations (drilling)
- When the diameter at the beginning and end of a cut only differs slightly

- During threading to allow the perfect synchronization between spindle revolution and Z-axis motion to allow precise threads Surface Feet Per Minute (SFM) is a combination of the cut diameter and RPM. The faster the spindle turns, and/or the larger the part diameter, the higher the SFM.

If two round pieces of different sizes are turning at the same revolutions per minute, the larger piece has a greater surface speed because it has a larger circumference and has more surface area. As the tool plunges closer to the center of a workpiece, the same spindle speed will yield a decreasing surface speed. This is because each revolution represents a smaller circumferential distance but takes the same amount of time.

Most CNC lathes have CSS (constant surface speed) to counteract the natural decrease in surface speed, which speeds up the spindle as the tool moves closer to the turning axis. CSS adjusts the revolutions per minute to maintain a constant surface speed at every distance from the center.

CSS is useful for:

- A uniform surface finishes.
- When the diameter at the beginning a cut will differ significantly from the diameter at the end of the cut.
- Better tool life and machining time because tools will always cut at the appropriate speed.

Materials will run better at specific SFMs. SFM is a constant, with RPM as a variable based upon cut diameter.

Feed Rate

Feed rate is the velocity at which the cutter is advanced along the workpiece. Feed rate is expressed as units of distance (inch) per minute or per single revolution.

Feed rate can be defined as inch per minute (IPM) or inch per revolution (IPR). IPR is more commonly used.

Selection of Tools, Feeds, and Speeds

Cutting tool selection has a direct impact on the proper programming of feeds and speeds at the machine.

However, many other variables that affect feeds and speeds are:

- Workpiece material class and condition
- Workpiece diameter
- Cutter material
- Cutter geometry
- Type of cut
- Depth of cut
- Condition of the machine

Cutting tool manufacturers publish the general feeds and speeds and recommended usage for the application. Cutting tool manufacturers are often a good place to start for recommendations on tool selection and feeds/speeds since they rely on customer loyalty. The customer (or potential) should select an insert and grade based on the vendor’s recommendation.

4

CNC Machine Tools

Computer numerical control (CNC) is the automation of machine tools by means of computers executing pre-programmed sequences of machine control commands. This contrasts with machines that are manually controlled by hand wheels or levers, or mechanically automated by cams alone.

In Industry it is not efficient or profitable to make everyday products by hand. On a CNC machine it is possible to make hundreds or even thousands of the same items in a day. First a design is drawn using design software, and then it is processed by the computer and manufactured using the CNC machine.

Motion is controlled along multiple axes, normally at least two (X and Y), and a tool spindle that moves in the Z (depth). The position of the tool is driven by direct-drive stepper motor or servo motors to provide highly accurate movements, or in older designs, motors through a series of step-down gears. **Open-loop** control works as long as the forces are kept small enough and speeds are not too great. On commercial metalworking machines, **closed loop** controls are standard and required to provide the accuracy, speed, and repeatability demanded.

CNC means Computer Numerical Control. This means a computer converts the design into numbers which the computer uses to control the cutting and shaping of the material.

Computer Numerical Control (CNC) is one in which the functions and motions of a machine tool are controlled by means of a prepared program containing coded alphanumeric data.

CNC can control the motions of the work piece or tool, the input parameters such as **feed, depth of cut, speed**, and the functions such as turning spindle on/off, turning coolant on/off.

Computer Numerical Control has been around since the early 1970's. Prior to this, it was called NC, for Numerical Control. (In the early 1970's computers were introduced to these controls, hence the name change)

People in most walks of life have never heard of this term, CNC has touched almost every form of manufacturing process in one way or another. If you'll be working in manufacturing, it's likely that you'll be dealing with CNC on a regular basis.

4.1 Introduction to Computer Numerical Control

The basic concepts of numerical control (NC) and computer numerical control (CNC) technology are discussed. Traditional NC and contemporary CNC hardware configurations are described. The important benefits to be derived from CNC operations are listed and explained. The different types of media used for storage and input of CNC programs are then explored. The reader is introduced to different formats for punched tape, and machining centers with automatic tool changers, the latest development in CNC, are considered.

Definition

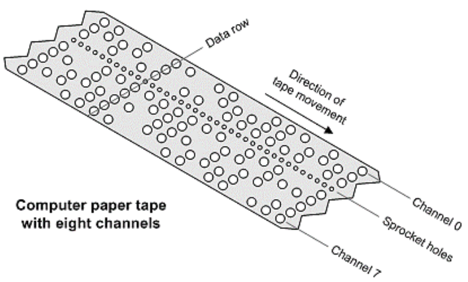
Computer numeric control production machines (CNC) are characterized by central control system, driving main and auxiliary machine functions by program. Information is in a program written by alphanumeric characters. The program itself is given by sequence of separate group of characters called blocks or sentences. The program is designed to manage power elements of the machine and guarantees the manufacturing of the workpiece [2].

Term CNC (Computer Numerical Control) means: Computer (numerically) controlled machine.

Machines are flexible, can quickly adapt for different production and work in automatic cycle which is managed by numerical control. CNC machines applied in all sectors of industrial production (machining, forming, assembling, measuring). Their typical representatives are lathes and milling machines used for training programmers and operators.
The information in the program can be divided into:
Geometric – describe the tool path, which are given by workpiece dimensions, machining methods and describe feeding of the tool to the workpiece and from it. It is a description of the tool tracks in Cartesian coordinates, when the creation of the program needs the dimensions of workpice blueprint. Description in program is in X, Z axis for the lathe, in axis X, Y, Z for milling machine (and also by other axes depending on construction of machine and workpiece complexity), by given functions which provides ISO and the individual producers of control systems.
Technological – provide machining technology in terms of cutting conditions (especially the speed or cutting speed, feed or depth).
Auxiliary – it's information, commands for the machine for some auxiliary functions (such as switching on coolant pump, direction of spindle rotation etc.)

4.2 Numerical control definition, its concepts and advantages

Numerical control has been used in industry for more than 40 years. Simply put, numerical control is a method of automatically operating a manufacturing machine based on a code of letters, numbers, and special characters. A complete set of coded instructions for executing an operation is called a program. The program is translated into corresponding electrical signals for input to motors that run the machine. Numerical control machines can be programmed manually. If a computer is used to create a program, the process is known as computer-aided programming. The approach taken in this text will be in the form of manual programming.
Traditionally, numerical control systems have been composed of the following components:
Tape punch: converts written instructions into a corresponding hole pattern. The hole pattern is punched into tape which is passed through the tape punch. Much older units used a typewriter device called a Flexowriter, and later devices included a microcomputer coupled with a tape punch unit.



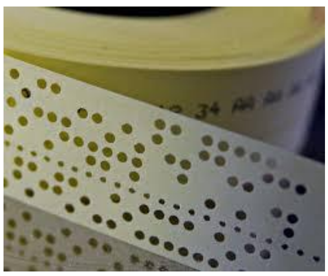


Fig. 4.1 The tape punch with the hole pattern

Tape reader: reads the hole pattern on the tape and converts the pattern to a corresponding electrical signal code.
Controller: receives the electrical signal code from the tape reader and subsequently causes the NC machine to respond.
NC machine: responds to programmed signals from the controller. Accordingly, the machine executes the required motions to manufacture a part (spindle rotation on/off, table and or spindle movement along programmed axis directions, etc.). See Fig. 4.2.

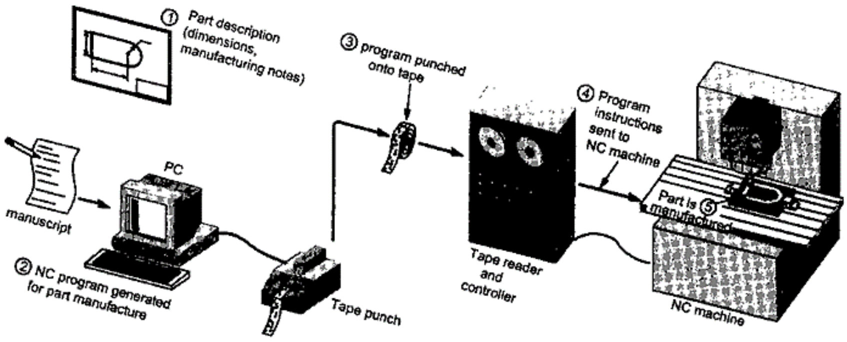


Fig. 4.2 Components of traditional NC systems

NC systems offer some advantages over manual production methods:
1. Better control of tool motions under optimum cutting conditions.
2. Improved part quality and repeatability.
3. Reduced tooling costs, tool wear, and job setup time.
4. Reduced time to manufacture parts.
5. Reduced scrap.
6. Better production planning and placement of machining operations in the hands of engineering.

4.3 Definition of Computer Numerical Control and its components

A computer numerical control (CNC) machine is an NC machine with the added feature of an onboard computer. The onboard computer is often referred to as the machine control unit or MCU. Control units for NC machines are usually hardwired, which means that all machine functions are controlled by the physical electronic elements that are built into the controller. The onboard computer, on the other hand, is "soft" wired, which means the machine functions are encoded into the computer at the time of manufacture, and they will not be erased when the CNC machine is turned off Computer memory that holds such information is known as ROM or read-only memory.
The MCU usually has an alphanumeric keyboard for direct or manual data input (MDI) of part programs. Such programs are stored in RAM or the random-access memory portion of the computer. They can be played back, edited, and processed by the control. All programs residing in RAM, however, are lost when the CNC machine is turned off. These programs can be saved on auxiliary storage devices such as punched tape, magnetic tape, or magnetic disk. Newer MCU units have graphics screens that can display not only the CNC program, but the cutter paths generated and any errors in the program.
The components found in many CNC systems are shown in Fig. 4.3.

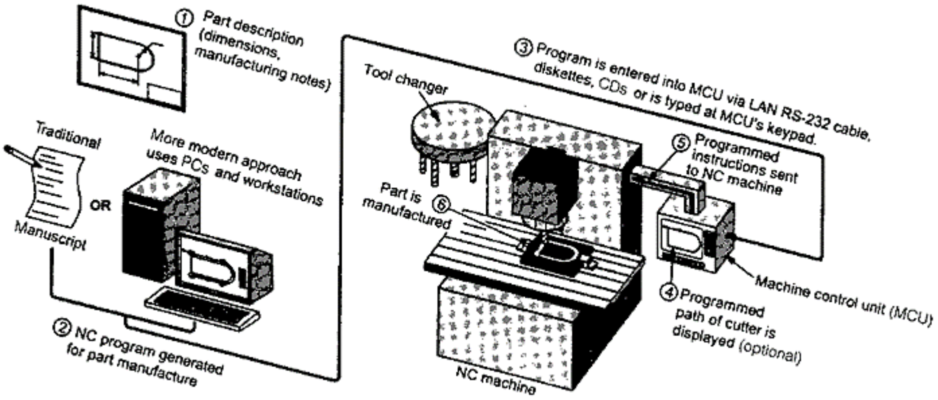


Fig. 4.3 Components of modern CNC systems

Machine control unit generates, stores, and processes CNC programs. The machine control unit also contains the machine motion controller in the form of an executive software program. See Fig. 4.4.

Fig. 4.4 A modern machine control unit (MCU) or CNC machine control panel

NC machine: responds to programmed signals from the machine control unit and manufactures the part.

Scheme of CNC machine tool

- Computer – it is an industrial computer with pre-recorded control system, which is part of the machine. Is given by a screen and control panel. Via control panel is possible to perform requested steps during manual servicing, for adjusting CNC machining and work in other modes of machine. The computer enables, by using the software control system, to create the required CNC program.
- Control circuits – In these circuits the logic signals are converted to high-voltage electrical signals, which directly control each part of the machine - the spindle and feed motor, valves, etc.
- Interpolator – Managing the tool path, which is given by geometry, calculations of the length and radius corrections of tool. So, it calculated equidistant movement, which is shifted from the calculated correction of the geometric contour. Guarantees the geometric accuracy of the product.
- Comparative circuit – The machine must be equipped with the feedback (with exceptions for simple CNC machines for staff training), which transmits information about the geometric values in coordinate axes at various points within the range of motion. These coordinates are compared with the values that are given by the program (and modified

in interpolator). If a difference appears, feed drives receive the command to achieve the desired coordinate values. The machine must be equipped with a transducer, which is used to obtain the coordinates.

Control panel – (can be solved as a figure 4.4) divided into several parts, which differ with their meanings:

- Data input – alpha numeric section, which is used to manually write for example a program, data instruments, adjusted for machinery, etc.
- Machine control – special section used to tool or workpiece motion, triggers the spindle speed, affects the size of the hand-feeds, speeds, etc.
- Choice of operation mode – you can choose the manual mode, automatic mode, workshop programming, etc.
- Memory activation – induce different types of memory.
- Tests activating – calling tests of programs and test of machine, simulation of programs.
- Screen – used to control process.
- Portable panel – for controlling the basic physical functions of the machine as a basic part of the keypad. Allows the operator to go to places that offer the possibility of more accurate and more complete visual inspection.

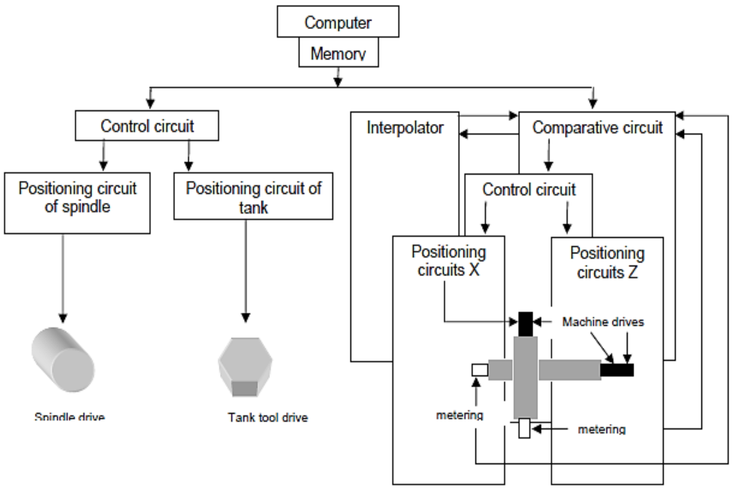


Fig. 4.5 Block diagram of the CNC machine tool – simplified

Advantages of CNC compared with NC

Computer numerical control opens new possibilities and advantages not offered by older NC machines.

1. Reduction in the hardware necessary to add a machine function. New functions can be programmed into the MCU as software.
2. The CNC program can be written, stored, and executed directly at the CNC machine.
3. Any portion of an entered CNC program can be played back and edited at will. Tool motions can be electronically displayed upon playback.
4. Many different CNC programs can be stored in the MCU.
5. Several CNC machines can be linked together to a main computer. Programs written via the main computer can be downloaded to any CNC machine in the network. This is known as direct numerical control or DNC. See Fig. 4.6 a.
6. Several **DNC** systems can also be networked to form a large distributive numerical control system. Refer to Fig. 4.6 b.

7. The CNC program can be input from zip or floppy disks or downloaded from local area networks.

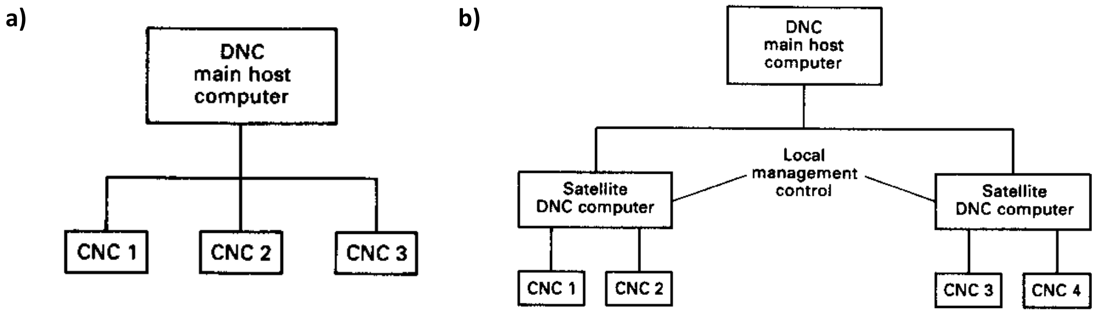


Fig. 4.6 a) Direct numerical control and b) distributive numerical control

Special Requirements for Utilizing CNC

Computer numerical control machines can dramatically boost productivity. The CNC manager, however, can only ensure such gains by first addressing several critical issues, such as:

1. Sufficient capital must be allocated for purchasing quality CNC equipment.
2. CNC equipment must be maintained on a regular basis by obtaining a full-service contract or by hiring an in-house technician.
3. Personnel must be thoroughly trained in the operation of CNC machines. In particular, many jobs require setups for machining parts to comply with tolerances of form and function.
4. Careful production planning must be studied because the hourly cost of operation of a CNC machine is usually higher than that for conventional machines.

Schemes of work CNC machining machine

During operations, we may encounter several types of operations of the machine or only types of machine control system. They can be set on the control panel by buttons. Typically, control systems have these schemes:

- Manual mode – is used for resetting tool or measuring equipment in the desired position, tool change, approaching to the workpiece, start-up speeds, etc.
- Automatic mode – a smooth implementation of the program. After the block processing machine reads and processes the next block automatically – smooth machining process.
- Mode B – B (block by block) – the machine stops after processing the block and after re-start reads and processes next block. BB scheme is one of the options for checking correctness of CNC program.
- Setting (impact speed, work shift, fast feed) – The amount of movement can affect by hand control, by potentiometer, where you can adjust range usually between 5-150% of the value set in manual or automatic mode.
- Tool memory mode (the tool data memory) – you can save and recall tools data, including corrections.
- Teach in mode ("learning" or "lead-in and storage") – the machine could learn. Operator manually (via keyboard) perform required movements to manufacture workpiece.
- EDIT mode program – a program for processing is entered directly into the editor of the machine or is "loaded" into the machine control system externally. In the editor of the machine programs can be repaired as required.

- Diagnostic mode – reports, locates, diagnoses defects for quick removal. It also allows remote service.

Features of the machine control unit (MCU) – machining centers

Specific details and features of machine control units vary from manufacturer to manufacturer. What is given in this section is a generic presentation of what is found on most machine control units for machining centers. The reader is reminded to consult the machine tool builder's manual for the detailed information relating to a particular machine control unit. In larger companies, it is the CNC setup person, not the programmer, who sets up tooling, loads the job, and runs the first piece on the CNC machine. In smaller operations, the programmer can be expected to get involved in some or all these tasks. To be more versatile, the programmer needs to acquire a basic knowledge of the features of the machine control unit.
The machine control unit is divided into two types of operations panels: the control panel and the machine panel (see Fig. 4.7).

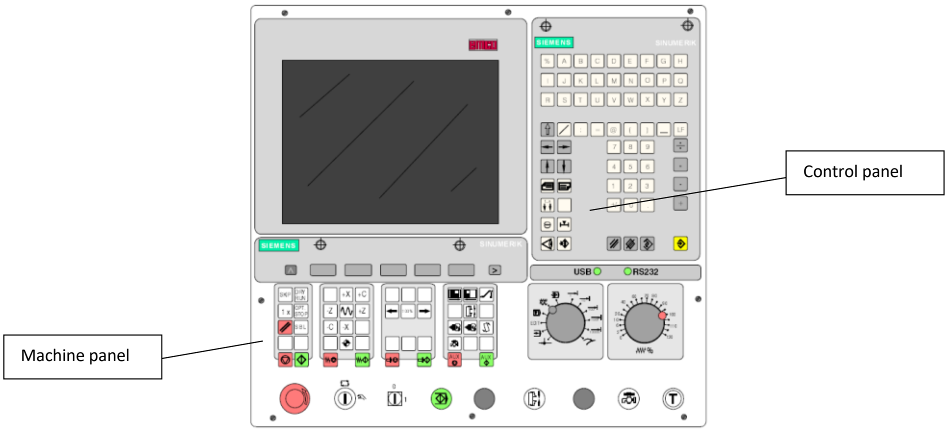


Fig. 4.7 Machine control unit (Sinumerik 820 T) is divided into two types of operations panels: the control panel and the machine panel

The control panel is designed and built by the control manufacturer. It contains a keyboard for entering data and a CRT display. CNC programs can be entered into memory, edited, and displayed on the CRT display. Data needed to set up a job, such as tool offsets, are also entered. Other important information such the axis positions of the machine and the spindle's speeds and feeds are also displayed on the control panel's CRT.
The machine panel is designed and built by the machine tool builder. It contains buttons and switches for controlling the physical behavior of the CNC machine tool. Power buttons turn the CNC machining center on or off. An emergency stop button is used to stop all machine motions. A jog wheel enables the operator to move a machine axis manually by turning a hand wheel. Dial controls can override programmed spindle speeds and feeds, etc.

Coordinate system of the machine

Production machines use a Cartesian coordinate system. The system is right-handed, rectangular with axes X, Y, Z, rotary motion, whose axes are parallel to the axes X, Y, Z, marked A, B, C – Fig. 4.8. True that the Z axis is parallel to the axis of the work spindle, and a positive sense takes place from the workpiece to the tool. Values are present also in the negative field of coordinates.

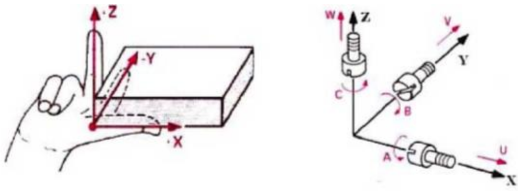


Fig. 4.8 Definition of Cartesian coordinates – right-handed system

Cartesian coordinate system is necessary for the machine control and for the measuring of the tools. Tool, in the machine, moves according to orders from the CNC machine control panel or under the CNC program commands. If necessary, coordinate system can move and rotate. In the case of measurement instruments (surveys corrections) is a Cartesian system placed in a point of exchange tools or tool tip.

The coordinate system is placed in the machine according to the following rules:

1) Start from a stationary workpiece.
2) Always must be defined X-axis.
3) The X axis lies in the plane of the fixture or the workpiece or is parallel to its plane.
4) Z-axis is identical or parallel to the axis of work spindle, which grants main cutting movement.
5) A positive axis sense goes from the workpiece to a tool in the direction of workpiece growing.
6) If there are other additional movements in the axes X, Y, Z on the machine, these are called U, V, W.
7) If the workpiece is moved against the tool, coordinates are called X ', Y' and Z '.

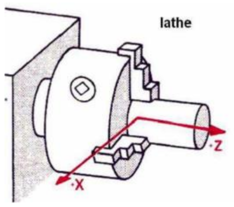


Fig. 4.9 Lathe coordinate system (spindle without driven tools)

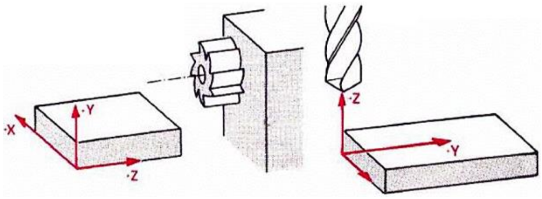


Fig. 4.10 Coordinate system for CNC horizontal milling and drilling

5

CNC Machining Centers and Turning Centers

Machining centers are the latest development in CNC technology. These systems come equipped with automatic tool changers capable of changing 90 or more tools. Many are also fitted with movable rectangular worktables called pallets. The pallets are used to automatically load and unload workpieces. At a single setup, machining centers can perform such operations as milling, drilling, tapping, boring, counterboring, and so on. Additionally, by utilizing indexing heads, some centers are capable of executing these tasks on many different faces of a part and at specified angles. Machining centers save production time and cost by reducing the need for moving a part from one machine to another. Two types of machining centers for milling are shown in Fig. 5.1 a and b.

a) **b)**

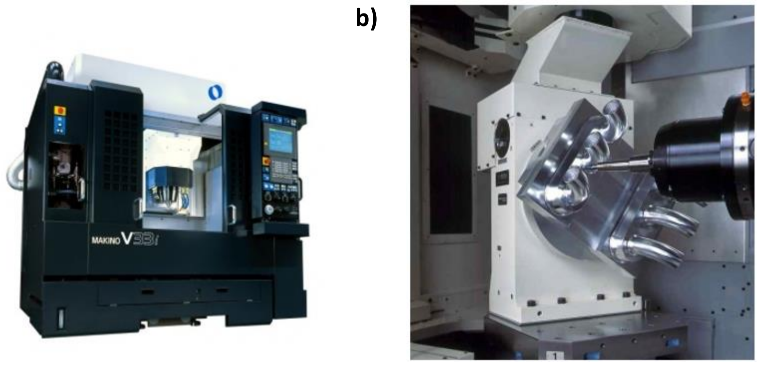


Fig. 5.1 a) Vertical Machining Center “V33i” and
b) 5-Axis Horizontal Machining Center “a1-5XR” of the firm MAKINO

Fig. 5.2 CNC turning center TNA 500 TRAUB with and kinematic of machining

Turning centers with increased capacity tool changers are also making a strong appearance in modern production shops. These CNC machines are capable of executing many different types of lathe cutting operations simultaneously on a rotating part. A modern turning center, milling center and modern machining center are shown in Fig 5.3-5.6.

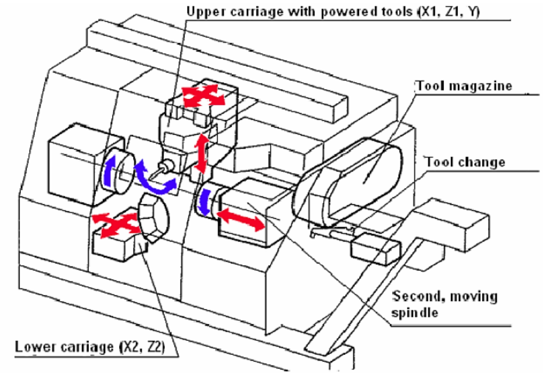


Fig. 5.3 Multi – axes CNC lathe

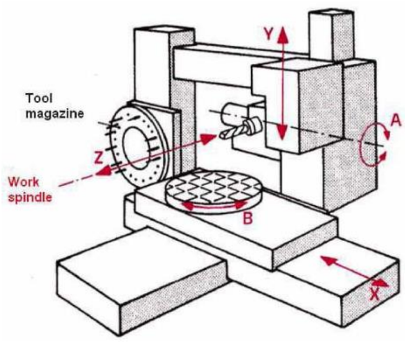


Fig. 5.4 Coordinate system of 5-axes horizontal milling center

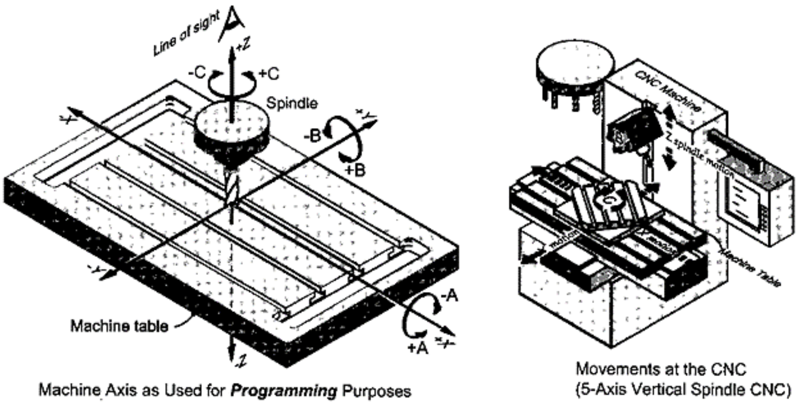


Fig. 5.5 Coordinate system of 5-axes vertical milling center

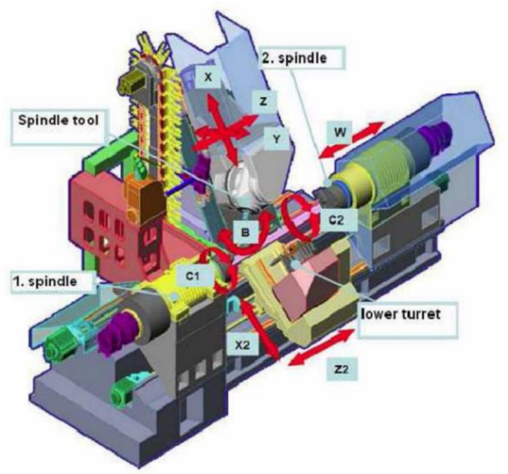


Fig. 5.6 Coordinate system of machining center
https://en.dmgmori.com/products/machines/turning/universal-turning/nef

6

Modern Machine Tool Controls

The machine control unit (MCU) contains a machine motion controller for controlling tool movement. Many different types of controllers are available today, including Fanuc, Allen-Bradley, GE, Okuma, Bendix, Mazak, and others. The physical appearance of these controllers is somewhat similar and each respond to a slightly different set of programmed codes. All control systems, however, fall into two major categories: point-to-point and continuous path [1].

6.1 Point-to-Point Tool Movements

Point-to-point control systems cause the tool to move to a point on the part and execute an operation at that point only. The tool is not in continuous contact with the part while it is being moved to a working location. Some point-to-point operations are drilling, reaming, boring, tapping, and punching. See Fig. 6.1.

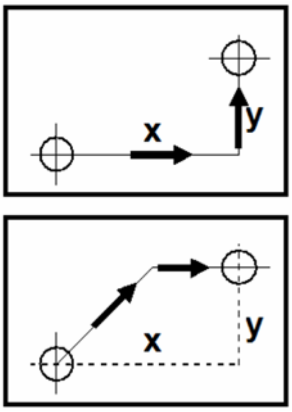


Fig. 6.1 Point-to-point tool movement

6.2 Continuous Path Tool Movements

Continuous path controllers are so named because they cause the tool to maintain continuous contact with the part as the tool cuts a contour shape. Continuous path operations include milling along lines at any angle, milling arcs, and lathe turning. See Fig. 6.2.

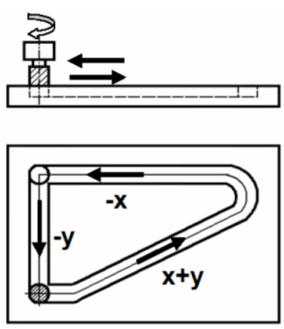


Fig. 6.2 Continuous path tool movement

6.3 Interpolation

It should be noted that continuous path controllers output motion by interpolating each position of the tool. Interpolation is a mathematical method of approximating the true or exact positions required to follow a precalculated path. The interpolated positions are determined such that they differ from the exact positions within an acceptable tolerance. Many continuous path controllers interpolate curves as a series of straight-line segments. Very high accuracy can be achieved by making the line segments smaller and smaller. These concepts are illustrated in Fig. 6.3.

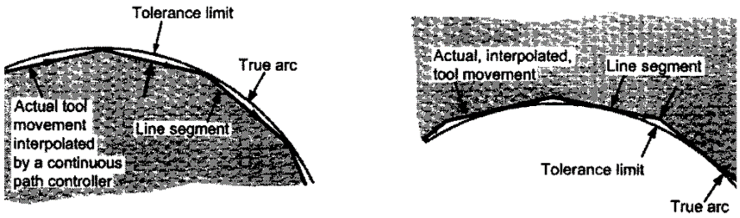


Fig. 6.3 Interpolation used for continuous path

The method by which contouring machine tools move from one programmed point to the next is called interpolation. This ability to merge individual axis points into a predefined tool path is built into most of today's MCUs.

There are five methods of interpolation: linear, circular, helical, parabolic, and cubic. All contouring controls provide linear interpolation, and most controls are capable of both linear and circular interpolation. Helical, parabolic, and cubic interpolation are used by industries that manufacture parts which have complex shapes, such as aerospace parts and dies for car bodies.

Information about track media of tool are processed in the interpolator. Interpolator is an arithmetic unit which calculates the path elements in each coordinate axis so that the resulting movement between two given points is:

- linear – linear interpolation
- around circular arc – circular interpolation
- parabola or a general curve

Interpolator generates signal of the desired path. Measuring device generates a signal on the actual track. Both signals are compared in a differential element – their difference is the regulation divergence, which after amplification and transformation creates a action quantity. In other words – differential member sends impulses to the motor until rest reached the desired position. Measuring devices work after some non-zero "jumping" – increments. Increment is the smallest measurable and therefore programmable path. At present is widely used increment 0.001 mm.

Linear Interpolation

Linear interpolation consists of any programmed points linked together by straight lines, whether the points are close together or far apart (Fig. 6.4). Curves can be produced with linear interpolation by breaking them into short, straight-line segments. This method has limitations, because a very large number of points would have to be programmed to describe the curve in order to produce a contour shape.

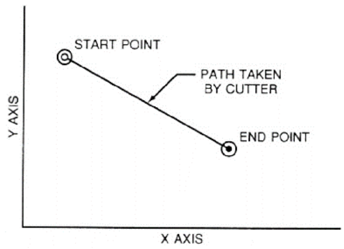


Fig. 6.4 An example of two-axis linear interpolation

A contour programmed in linear interpolation requires the coordinate positions (XY positions in two-axis work) for the start and finish of each line segment. Therefore, the end point of one line or segment becomes the start point for the next segment, and so on, throughout the entire program.

The Principle of Linear Interpolation

Fig. 6.5 shows the principle of linear interpolation of first quadrant.

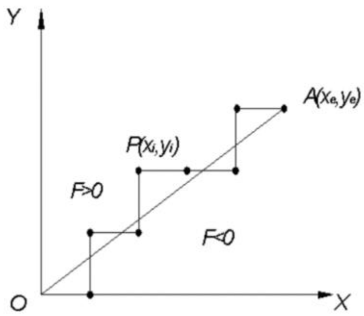


Fig. 6.5 The principle of linear interpolation

Assume that starting point of line $\overline{OA}$ is the origin of coordinates, and the coordinate of terminal point *A* is *(x_e ,y_e)* , and *, P(x_i, y_i)* is machining point. If *P* happens to be the point on the line $\overline{OA}$, the following condition holds:

$x_e y_j$ - $x_i y_e = 0$.

If arbitrary point *P(x_i, y_i)* is above the line $\overline{OA}$, (Seriously, within the range of the angle between the line $\overline{OA}$, and *y* axis), then the eq.1 holds:

$$\frac{y_j}{x_i} > \frac{y_e}{x_e}$$

That is: $x_e y_j - x_i y_e > 0$.

Then the error discriminant function F_{ij} is expressed as $F_{ij} = x_e y_j - x_i y_e$.

The relative position between the point P and line $\overline{OA}$, can be differentiated from the value of F_{ij}, that is:

When $Fij = 0$, the point P happens to be in the line $\overline{OA}$,. When $Fij > 0$, the point *P* is above the line $\overline{OA}$, When $Fij <0$, the point *P* is under the line$\overline{OA}$.

Interpolator generates signal of the desired path. Measuring device generates a signal on the actual track. Both signals are compared in a differential element – their difference is the regulation divergence, which after amplification and transformation creates an action quantity. In other words – differential member sends impulses to the motor until rest reached the desired position. Measuring devices work after some non-zero "jumping" – increments. Increment is the smallest measurable and therefore programmable path. At present is widely used increment 0.001 mm.

Circular Interpolation

The development of MCUs capable of *circular interpolation* has greatly simplified the process of programming arcs and circles. To program an arc (Fig. 6.6), the MCU requires only the coordinate positions (the XY axes) of the circle center, the radius of the circle, the start point, and end point of the arc being cut, and the direction in which the arc is to be cut (clockwise or counterclockwise). See Fig. 6.6.

The information required may vary with different MCUs.

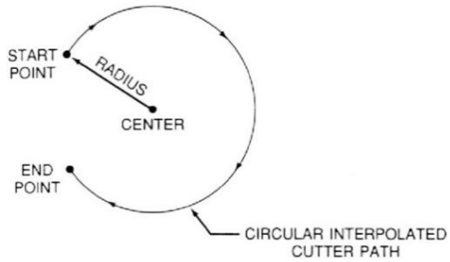


Fig. 6.6 For two-dimensional circular interpolation the MCU must be supplied with the XY axis, radius, start point, end point, and direction of cut

The Principle of Circular Interpolation

Traditionally, continuous path controllers were more expensive than point-to-point systems. Advancements in microelectronics, however, have reduced the cost and today most CNC machines come equipped with continuous path controllers.

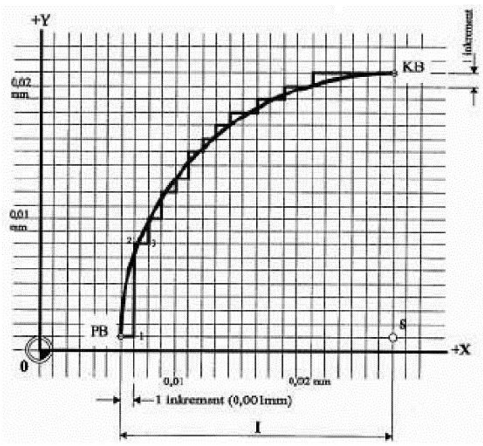


Fig. 6.7 Circular interpolation – interpolar function

Operation principle of interpolator of circular interpolation clockwise (G02) is on the previous figure (Fig. 6.7):

1. Creates an equation of a circle in the XY plane $(X-0{,}026)^2 + (Y-0{,}001)^2 = 0{,}022$
2. Send a unit impulse in the direction + X
3. Substituting the coordinates of point 1 in the equation of a circle and find that the left side of the equation is smaller than the right. This means that the point 1 lies within the arc.
4. Change the direction of movement and sends unit pulses until it finds that the point lies outside the arc, see section 2)
5. Repeats the previous paragraphs, until it gets to the end point.

6.4 Loop Systems for Controlling Tool Movement

A loop system sends electrical signals to drive motor controllers and receives some form of electrical feedback from the motor controllers. One of the important factors that determine the tolerance to which a part can be cut is the loop system type.
There are two main systems in use today for controlling CNC machine movements: the open loop system and the closed loop system.

Open Loop Systems

An open loop system utilizes stepping motors to create machine movements (see Fig. 6.8).

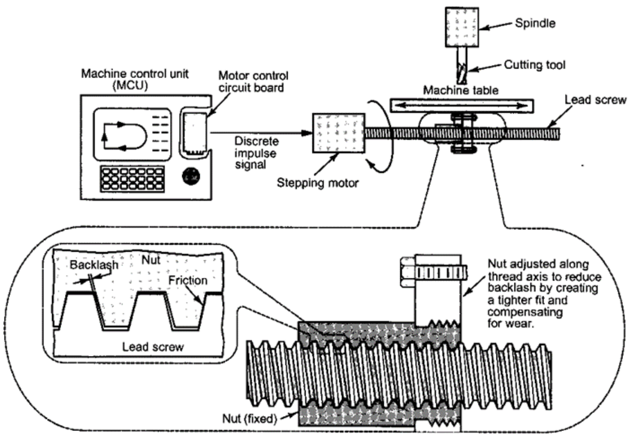


Fig. 6.8 Configuration for an open loop system that uses a conventional Acme lead screw mechanism

These motors rotate a fixed amount, usually 1.8°, for each pulse received. Stepping motors are driven by electrical signals coming from the MCU. The motors are connected to the machine table lead screw and spindle. Upon receiving a signal, they move the table and/or spindle a fixed amount. The motor controller sends signals back indicating the motors have completed the motion. The feedback, however, is not used to check how close the actual machine movement comes to the exact movement programmed. Also, the lead screws used in these systems tend to generate friction and backlash. Backlash can cause positioning errors if the motions required to machine a part require a reversal in axis direction.

Closed Loop Systems

Special motors called servos are used for executing machine movements in closed loop systems. Motor types include AC servos, DC servos, and hydraulic servos. Hydraulic servos, being the most powerful, are used on large CNC machines. AC servos are next in strength and are found on many machining centers.
A servo does not operate like a pulse counting stepper motor. The speed of an AC or DC servo is variable and depends upon the amount of current passing through it. The speed of a hydraulic servo depends upon the amount of fluid passing through it. The strength of current coming from the Motor Drive Unit determines the speed at which a servo rotates. A device called a tachometer is mounted on the backside of each servo to measure and feedback its angular velocity or RPM. The feedback is sent to the Motor Drive Unit. The unit compares the motion command from MCU and the voltage feedback from the tachometer (actual motor speed) and outputs back to the motor the difference between the two values or error. Velocity and acceleration control are important in ensuring both tool load and surface finish are acceptable.

The servos are connected to the spindle. They are also connected to the machine table through the ball lead screw (Fig. 6.9, 6.10 and 6.11). The ball lead screw is the heart of the drive system. It overcomes the problems of lead screws by greatly reducing friction and backlash.

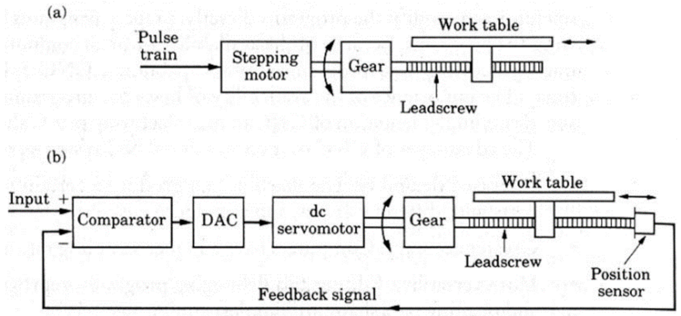


Fig. 6.9 Open loop vs. closed loop controls

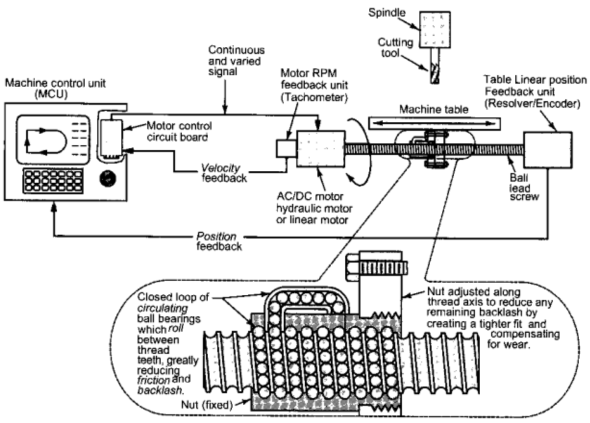


Fig. 6.10 Configuration for a closed loop system that uses a ball lead screw mechanism

This is accomplished by creating a rolling motion design in which precision ball bearings roll between the nut and the screw, which allows for higher precision positioning with greater repeatability, and positioning at higher speeds with less wear.

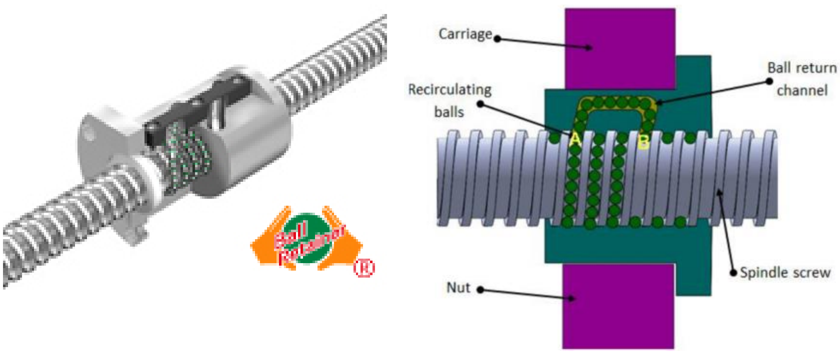


Fig. 6.11 The ball lead screw

A device called a resolver or encoder continuously monitors the distance by which the table and/or spindle has moved and sends this information back to the MCU. The MCU can then adjust its signal as the actual table and/or spindle position approaches the programmed position. Systems that provide feedback signals of this type are called servo systems or servomechanisms. They can position tools with a very high degree of accuracy even when driving motors with high-horsepower ranges. A typical CNC closed loop system has guaranteed positioning accuracies of 0.001 or 0.0001 in. See Fig. 6.10.

There has been renewed interest in open loop systems for CNC applications. Improvements in stepping motor accuracy and power have, in some cases, eliminated the need for expensive feedback system hardware and its associated circuitry. These newer systems represented substantial savings in machine and maintenance costs.

Precision Position Feedback Detector

General

Mitsubishi Precision Scale (MP SCALE) is a non-physical contact, ultra-precision position detector used for detecting linear or angular position utilizing the inductive coupling principle. The manufacture and sale of MP Scale has been started under the technical cooperation with INDUCTOSYN Co. in U.S. since 1970.

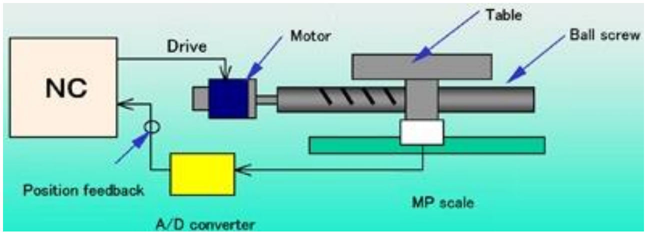


Fig. 6.12 Full closed control composition

MP Scale detects the amount of movement of a machine and outputs an analog signal. This signal is converted into a digital signal by an A/D converter and fed back to the NC system.

Thus the machine can be controlled accurately by the fully closed-loop NC system based on the position feedback data from MP SCALE.

Note: Full closed-loop NC system (Fig. 6.12): The control method which detects the last position (or angle) of the machine by Scale, and return detected data to NC equipment in order to move the machine to the target position (or angle).

Linear Feedback Detector

Slider of Linear MP Scale is mounted on the movable part, and Scale is mounted on the fixed part. The position of the linear axis is detected with high precision by Linear MP Scale (Fig. 6.13).

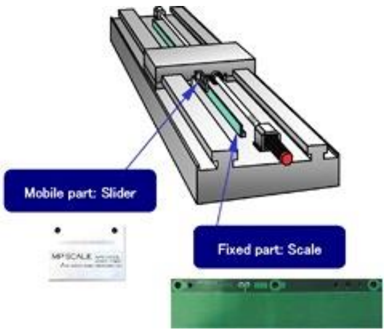


Fig.6.13 Linear position detector (Linear MP Scale)

Rotary Feedback Detector

Rotor of Rotary MP Scale is mounted on the rotation shaft of the rotating axis like the rotary table or the angle head, and Stator is mounted at the fixed part. The rotation angle is detected with high precision by Rotary MP Scale (Fig. 6.14).

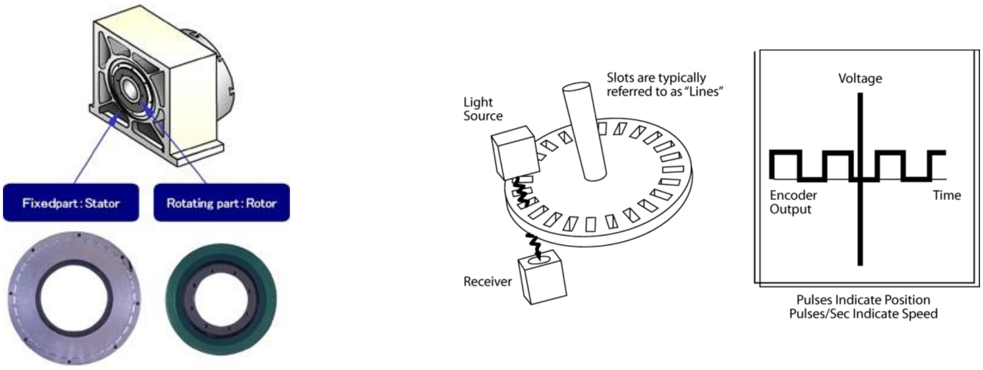


Fig. 6.14 Rotation angle detector (Rotary MP Scale)

Principle

1. The induction pattern, in the rectangular form, is printed on both Slider and Scale in the linear type and Stator and Rotor in the Rotary type of MP Scale.
2. When the alternating current is fed to the Slider (Stator) patter, the voltage is energized to the Scale(Rotor) pattern due to the inductive coupling action.

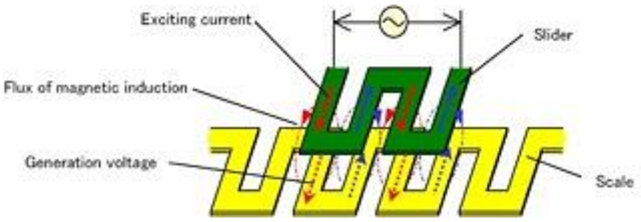


Fig. 6.15 Inductive coupling action

3. The energized voltage will vary as the relative position of Scale (Rotor) and Slider (Stator) changes. The position of the machine is detected based on the variation of this voltage.

Incremental Encoders

An encoder is an electrical mechanical device that converts linear or rotary displacement into digital or pulse signals. The most popular type of encoder is the optical encoder, which consists of a rotating disk, a light source, and a photo detector (light sensor). The disk, which is mounted on the rotating shaft, has patterns of opaque and transparent sectors coded into the disk (Fig. 6.16). As the disk rotates, these patterns interrupt the light emitted onto the photo detector, generating a digital or pulse signal output.

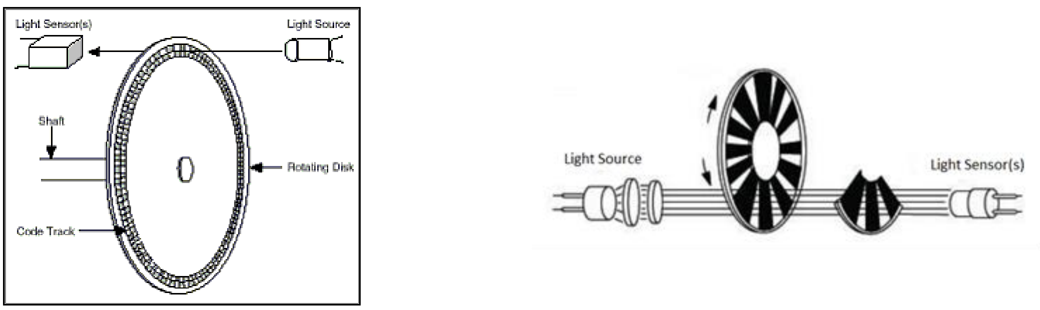


Fig. 6.16 Construction of Incremental encoders

An incremental encoder generates a pulse for each incremental step in it's rotation. Although the incremental encoder does not output absolute position, it can provide high resolution at an acceptable price. For example, an incremental encoder with a single code track, referred to as a tachometer encoder, generates a pulse signal whose frequency indicates the velocity of displacement. However, the output of the single-channel encoder does not indicate direction. To determine direction, a two-channel, or quadrature, encoder uses two detectors and two code tracks (Fig. 6.17).

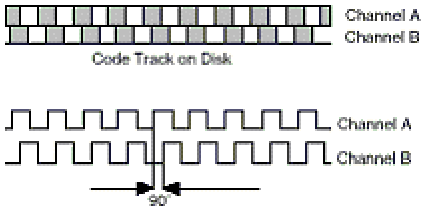


Fig. 6.17 Principles of incremental encoders

The most common type of incremental encoder uses two output channels (A and B) to sense position. Using two code tracks with sectors positioned 90° out of phase, the two output channels of the quadrature encoder indicate both position and direction of rotation. If A leads B, for example, the disk is rotating in a clockwise direction. If B leads A, then the disk is rotating in a counterclockwise direction. Therefore, by monitoring both the number of pulses and the relative phase of signals A and B, you can track both the position and direction of rotation.

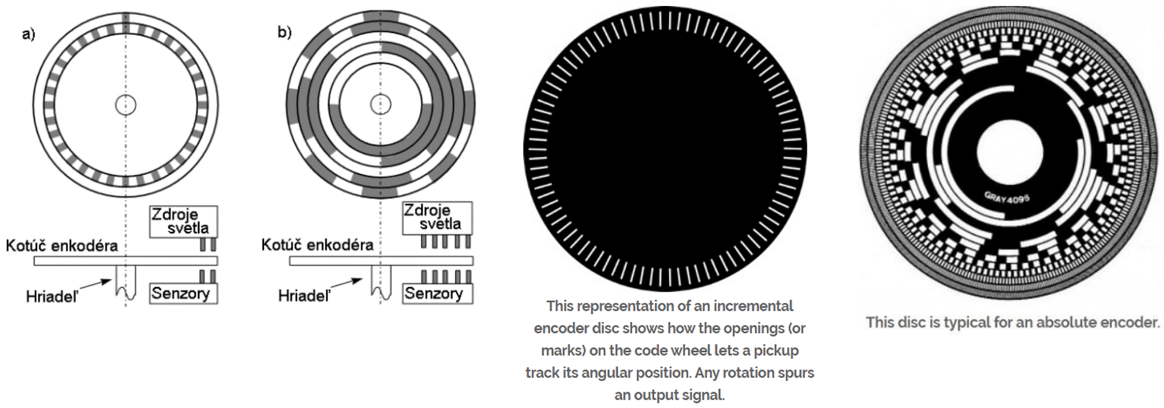


Fig. 6.18 Incremental encoders a) and absolute encoders b)

Rotary **incremental encoders** work by generating a series of pulses during movement. The encoder disc (sporting marks or slots) attaches to a power-transmission shaft, and a stationary pickup device mounts nearby. When the shaft and disc turn, the pickup tracks the motion to output the relative position. Such encoders generally supply square-wave signals in two channels that are offset from each other by 90° — in other words, out of phase by 90°. Each increment of rotation spurs an output signal. Incremental encoders must always come back to a reference point. This must occur both when the machine initially starts and whenever something interrupts its power supply. Here, battery backups can help eliminate the need for re-homing after shutdowns. Incremental encoders are generally simpler and cheaper than absolute encoders.

Absolute encoders have an encoder disc (sporting marks or slots) on a power-transmission shaft and a stationary pickup, but the disc marks output a unique code for each shaft position. Absolute encoders are either single-turn or multi turn encoders. Single turn absolute encoders can verify position within a single turn of the encoder shaft. This makes them useful for short

travel situations. In contrast, multiple-turn absolute encoders are better for more complex or longer positioning situations.
In addition, some quadrature detectors include a third output channel, called a zero or reference signal, which supplies a single pulse per revolution. This single pulse can be used for precise determination of a reference position.

Fig. 6.19 Construction of absolute encoder

6.5 Controlling Backlash in CNC Systems

All ballscrews have some "slop" or backlash at assembly. As stated before, this backlash causes errors when the screw reverses direction and the nut lags behind. Modern machine tool laser calibration equipment is used to precisely measure the amount of pitch error and backlash in the CNC positioning system. This data is input into a backlash compensation program installed in the MCU. Backlash compensation is then added to all motor commands. The backlash parameter values should normally be checked about every 3-6 months, because as the machine wears, the value will increase. Normal wear might have 0.005 "-0.010" adjustment in a ballscrew. A number higher than 0.010" is cause for a closer look. A good tip is to change the location of setups on the machine table to spread even wear on the screws. Care should also be taken to keep the screws away from excessive chip buildups that can be forced into the mechanism and cause premature wear.
A good tip is to change the location of setups on the machine table to spread even wear on the screws. Care should also be taken to keep the screws away from excessive chip buildups that can be forced into the mechanism and cause premature wear.

6.6 Types of Tool Positioning Modes

Within a given machine axes coordinate system, a CNC can be programmed to locate tool positions in the following modes: incremental, absolute, or mixed (incremental and absolute). Two types of programming modes, the incremental system and the absolute system, are used for CNC. Both systems have applications in CNC programming, and no system is either right or wrong all the time. Most controls on machine tools today are capable of handling either incremental or absolute programming (Fig. 6.20).

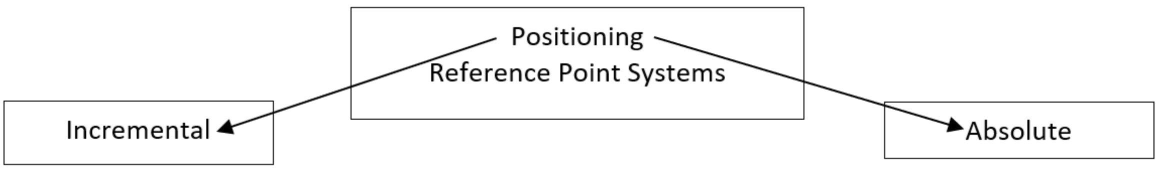
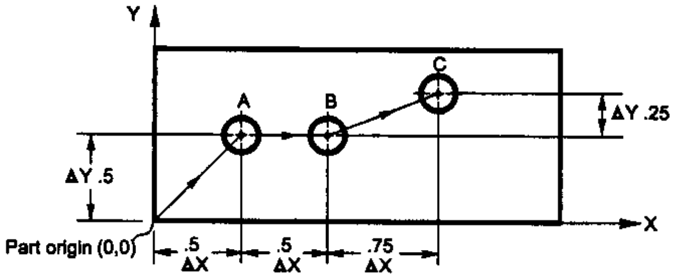


Fig. 6.20 Types of tool positioning

Incremental Positioning

Machines operating in incremental positioning mode locate each new tool position by measuring from the last tool position established. See Fig. 6.21 for an illustration of incremental positioning.



Tool position	Location	
	ΔX	ΔY
A	.5	.5
B	.5	0
C	.75	.25

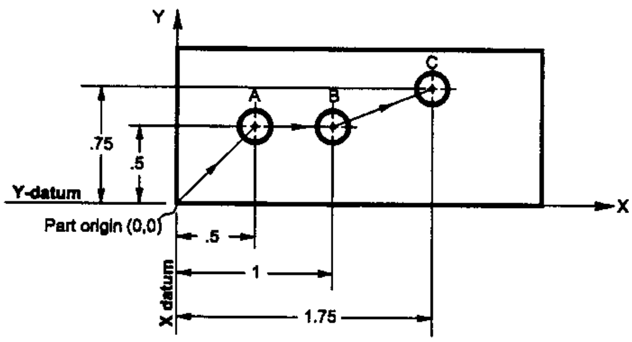
Fig. 6.21 Delta dimensioning for incremental mode positioning

Incremental positioning has some drawbacks. The most notable is that if one incremental movement is in error, all subsequent movements will also be incorrect.

Note: With delta dimensioning, each new dimension is specified by measuring it relative to the dimension previously entered. Delta dimensioning is well suited to incremental position programming.

Absolute Positioning

When operating in absolute positioning mode, the machine determines each new tool position from a fixed home or specified origin (0, 0). Refer to Figure 6.22.



Tool position	Location	
	ΔX	ΔY
A	.5	.5
B	1	.5
C	1.75	.75

Fig. 6.22 Datum dimensioning for absolute mode positioning

Many modern controllers can operate in either incremental or absolute positioning mode. The programmer can switch from one to the other by inputting a single code.

Note: With datum dimensioning, datum or zero reference line is established. All linear dimensions are then taken relative to the datum. Datum dimensioning is well suited to absolute position programming.

7

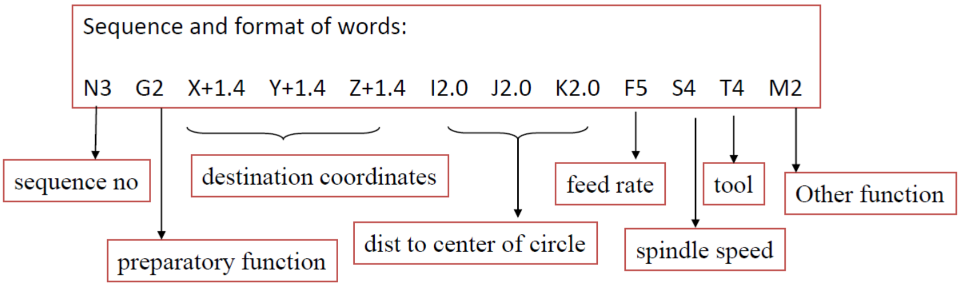
Programming format, interpolation

7.1 Programming Format

Word address is the most common programming format used for CNC programming systems. This format contains a large number of different codes (preparatory and miscellaneous) that transfers program information from the part print to machine servos, relays, micro-switches, etc., to manufacture a part. These codes, which conform to EIA (Electronic Industries Association) standards, are in a logical sequence called a *block of information*. Each block should contain enough information to perform one machining operation.

7.2 Word Address Format

Every program for any part to be machined, must be put in a format that the machine control unit can understand. The format used on any CNC machine is built in by the machine tool builder and is based on the type of control unit on the machine. A variable-block format which uses words (letters) is most commonly used. Each instruction word consists of an address character, such as X, Y, Z, G, M, or S. Numerical data follows this address character to identify a specific function such as the distance, feed rate, or speed value.
The address code G90 in a program, tells the control that all measurements are in the absolute mode. The code G91, tells the control that measurements are in the incremental mode.



7.3 Codes

The most common codes used when programming CNC machines tools are G-codes (preparatory functions), and M codes (miscellaneous functions). Other codes such as F, S, D, and T are used for machine functions such as feed, speed, cutter diameter offset, tool number, etc.
G-codes are sometimes called cycle codes because they refer to some action occurring on the X, Y, and/or Z axis of a machine tool, Fig. 7.1.
The G-codes are grouped into categories such as Group 01, containing codes G00, G01, G02, and G03 which cause some movement of the machine table or head. Group 03 includes either absolute or incremental programming, while Group 09 deals with canned cycles.
A G00 code rapidly positions the cutting tool while it is above the workpiece from one point to another point on a job. During the rapid traverse movement, either the X or Y axis can be moved individually or both axes can be moved at the same time. Although the rate of rapid

travel varies from machine to machine, it ranges between 200 and 800 in./min (5 and 20 m/min).

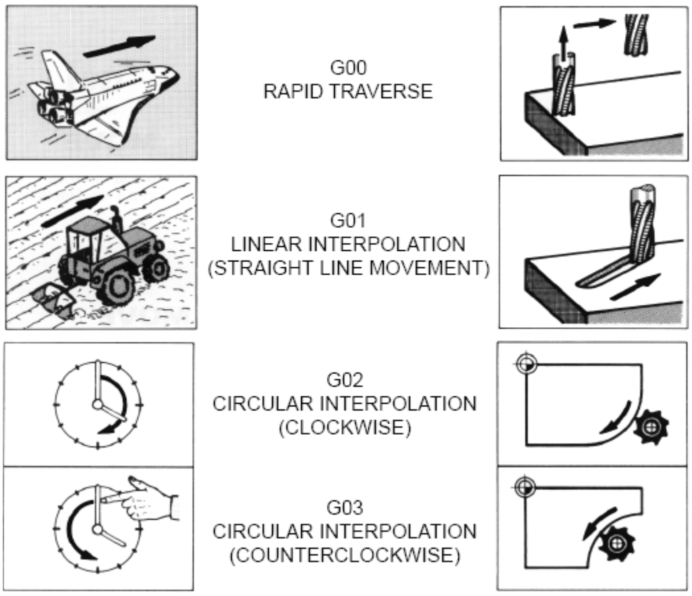


Fig. 7.1 The functions of a few common G – codes used in CNC programming

The G01, G02, and G03 codes move the axes at a controlled feedrate.

G01 is used for straight-line movement (linear interpolation).

G02 (clockwise) and G03 (counterclockwise) are used for arcs and circles (circular interpolation).

M or miscellaneous codes are used to either turn ON or OFF different functions which control certain machine tool operations, Fig. 7.2

M-codes are not grouped into categories, although several codes may control the same type of operations such as M03, M04, and M05 which control the machine tool spindle.

M03 turns the spindle on clockwise

M04 turns the spindle on counterclockwise

M05 turns the spindle off.

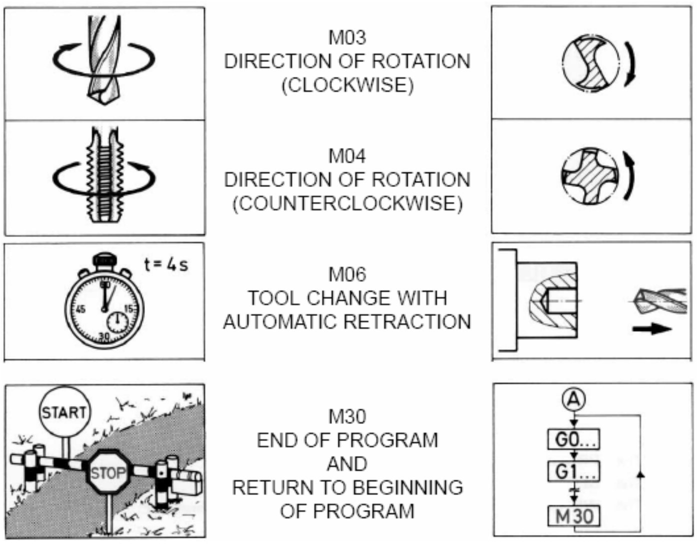


Fig. 7.2 The functions of a few common M-codes (Deckel Maho, Inc.)

Code	Function
M00	Program stop
M02	End of program
M03	Spindle start (forward CW)
M04	Spindle start (reverse CCW)
M05	Spindle stop
M06	Tool change
M08	Coolant on
M09	Coolant off
M10	Chuck - clamping (**)
M11	Chuck - unclamping (**)
M12	Tailstock spindle out (**)
M13	Tailstock spindle in (**)
M17	Toolpost rotation normal (**)
M18	Toolpost rotation reverse (**)
M30	End of tape and rewind
M98	Transfer to subprogram
M99	End of subprogram

(**) - refers only to CNC lathes and turning centers.

Fig. 7.3 Some of the most common M-codes used in CNC programming

Variable	Description	Corollary info
A	Absolute or incremental position of A axis (rotational axis around X axis)	Positive rotation is defined as a counterclockwise rotation looking from X positive towards X negative.
B	Absolute or incremental position of B axis (rotational axis around Y axis)	
C	Absolute or incremental position of C axis (rotational axis around Z axis)	
D	Defines diameter or radial offset used for cutter compensation. D is used for depth of cut on lathes. It is used for aperture selection and commands on photoplotters.	G41: left cutter compensation, G42: right cutter compensation
E	Precision feedrate for threading on lathes	
F	Defines feed rate	Common units are distance per time for mills (inches per minute, IPM, or millimeters per minute, mm/min) and distance per revolution for lathes (inches per revolution, IPR, or millimeters per revolution, mm/rev)
G	Address for preparatory commands	G commands often tell the control what kind of motion is wanted (e.g., rapid positioning, linear feed, circular feed, fixed cycle) or what offset value to use.
H	Defines tool length offset; Incremental axis corresponding to C axis (e.g., on a turn-mill)	G43: Negative tool length compensation, G44: Positive tool length compensation
I	Defines arc center in X axis for G02 or G03 arc commands. Also used as a parameter within some fixed cycles.	The arc center is the relative distance from the current position to the arc center, not the absolute distance from the work coordinate system (WCS).
J	Defines arc center in Y axis for G02 or G03 arc commands. Also used as a parameter within some fixed cycles.	Same corollary info as I above.

K	Defines arc center in Z axis for G02 or G03 arc commands. Also used as a parameter within some fixed cycles, equal to L address.	Same corollary info as I above.
L	Fixed cycle loop count; Specification of what register to edit using G10	Fixed cycle loop count: Defines number of repetitions ("loops") of a fixed cycle at each position. Assumed to be 1 unless programmed with another integer. Sometimes the K address is used instead of L. With incremental positioning (G91), a series of equally spaced holes can be programmed as a loop rather than as individual positions. G10 use: Specification of what register to edit (work offsets, tool radius offsets, tool length offsets, etc.).
M	Miscellaneous function	Action code, auxiliary command; descriptions vary. Many M-codes call for machine functions, which is why people often say that the "M" stands for "machine", although it was not intended to.
N	Line (block) number in program; System parameter number to change using G10	Line (block) numbers: Optional, so often omitted. Necessary for certain tasks, such as M99 P address (to tell the control which block of the program to return to if not the default) or GoTo statements (if the control supports those). N numbering need not increment by 1 (for example, it can increment by 10, 20, or 1000) and can be used on every block or only in certain spots throughout a program. System parameter number: G10 allows changing of system parameters under program control.[6]
O	Program name	For example, O4501. For many years it was common for CNC control displays to use slashed zero glyphs to ensure effortless distinction of letter "O" from digit "0". Today's GUI controls often have a choice of fonts, like a PC does.
P	Serves as parameter address for various G and M codes	• With G04, defines dwell time value. • Also serves as a parameter in some canned cycles, representing dwell times or other variables. • Also used in the calling and termination of subprograms. (With M98, it specifies which subprogram to call; with M99, it specifies which block number of the main program to return to.)
Q	Peck increment in canned cycles	For example, G73, G83 (peck drilling cycles)
R	Defines size of arc radius, or defines retract height in milling canned cycles	For radii, not all controls support the R address for G02 and G03, in which case IJK vectors are used. For retract height, the "R level", as it's called, is returned to if G99 is programmed.
S	Defines speed, either spindle speed or surface speed depending on mode	Data type = integer. In G97 mode (which is usually the default), an integer after S is interpreted as a number of rev/min (rpm). In G96 mode (CSS), an integer after S is interpreted as surface speed – sfm (G20) or m/min (G21). See also Speeds and feeds. On multifunction (turn-mill or mill-turn) machines, which spindle gets the input (main spindle or subspindles) is determined by other M codes.
T	Tool selection	To understand how the T address works and how it interacts (or not) with M06, one must study the various methods, such as lathe turret programming, ATC fixed tool selection, ATC random memory tool selection, the concept

		of "next tool waiting", and empty tools.[3] Programming on any particular machine tool requires knowing which method that machine uses.[3]
U	Incremental axis corresponding to X axis (typically only lathe group A controls) Also defines dwell time on some machines (instead of "P" or "X").	In these controls, X and U obviate G90 and G91, respectively. On these lathes, G90 is instead a fixed cycle address for roughing.
V	Incremental axis corresponding to Y axis	Until the 2000s, the V address was very rarely used, because most lathes that used U and W didn't have a Y-axis, so they didn't use V. (Green et al. 1996[5] did not even list V in their table of addresses.) That is still often the case, although the proliferation of live lathe tooling and turn-mill machining has made V address usage less rare than it used to be (Smid 2008[3] shows an example). See also G18.
W	Incremental axis corresponding to Z axis (typically only lathe group A controls)	In these controls, Z and W obviate G90 and G91, respectively. On these lathes, G90 is instead a fixed cycle address for roughing.
X	Absolute or incremental position of X axis. Also defines dwell time on some machines (instead of "P" or "U").	
Y	Absolute or incremental position of Y axis	
Z	Absolute or incremental position of Z axis	The main spindle's axis of rotation often determines which axis of a machine tool is labeled as Z.

Fig. 7.4 The most common codes for programming CNC machines

8

Tooling Systems

8.1 Tools for machining centers

The production of a machined component invariably involves the use of a variety of cutting tools, and the machine has to cater for their use. The way in which a range of cutting tools can be located and securely held in position is referred to as a tooling system and is usually an important feature of the machine tool manufacturers' advertising literature.
The tooling system for a machining center is illustrated in Fig. 8.1. Note the use of tool holders with standard tapers, a feature that can be very helpful in keeping tooling costs to a minimum.

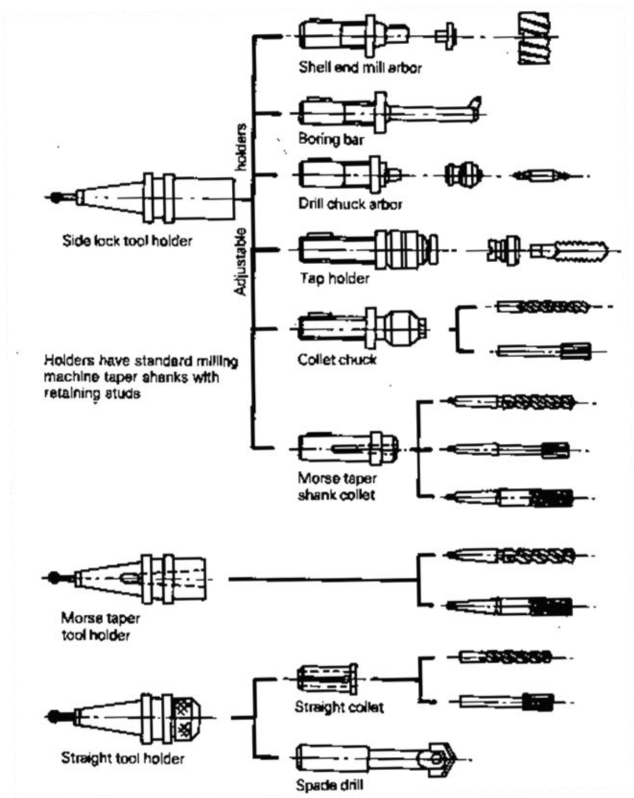


Fig. 8.1 Tooling system for a machining center

The types of tool holders shown in Fig. 8.2 are retained in and released from the machine spindle by a hydraulic device, an arrangement that lends itself to automation, since it is relatively simple to control hydraulic systems using electrically activated solenoid valves which themselves can be controlled via the machine control system.
The hydraulic force retaining the holder is supplemented by a mechanical force exerted by powerful disk springs, as illustrated in Fig. 8.2 for added safety. It should be noted that less expensive machines may use a mechanical device in conjunction with pneumatics or hydraulics for tool changes.

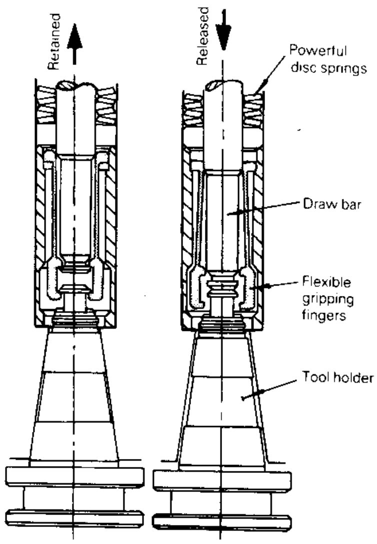


Fig. 8.2 Hydraulic – mechanical draw bar assembly used on machining centers

Not all machines have automatic tool-changing arrangements and when manual tool changing is involved, mechanical retaining devices are used. Conventional tool holders for milling situations use the tried and tested screwed drawbar arrangement, but unfortunately their use is not in keeping with modern machining techniques, where the accent is on speed. Because of this, several machine tool manufacturers have introduced tool holders of their own design that have dispensed with the need to undo a drawbar each time a tool holder is changed and as a result they have greatly speeded up the replacement process.
As with milling, a tooling system for a turning center will indicate the range of tooling which can be accommodated on the machine. One such system is illustrated in Fig. 8.3.

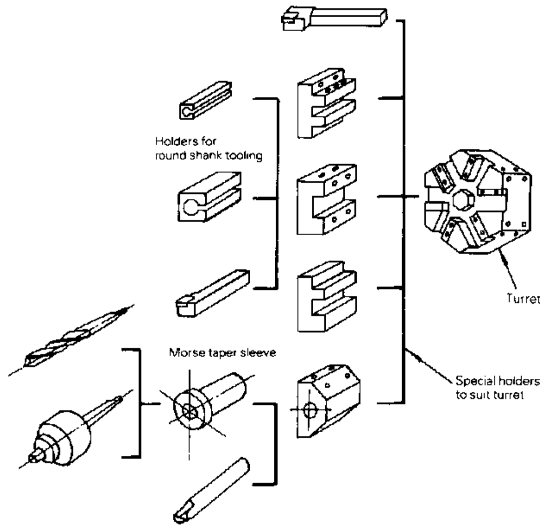


Fig. 8.3 Tooling system for a turning center

8.2 Tool identify

The automatic selection and presentation of a cutting tool to the workpiece is a prime function of computer numerically controlled machining. To achieve this there must be a link between programming and machine setup. Tool stations are numbered according to the tooling stations available, and when writing a program, the programmer will provide each tool with a corresponding numerical identity, usually in the form of the letter T followed by two digits:

T01, T02, T03 and so on (Fig. 8.4). The machine setter will need to know the type of tool required and will set it in its allocated position.

Fig. 8.4 Tooling system of FANUC Robodrill [3]

Tool encoding using chips

The memory module of each tool is contactlessly scanned by the scanning head, when the magazine is rotated (Fig. 8.5). The memory module contains not only the numeric code of the tool, but also the tool's durability information and its usage time.

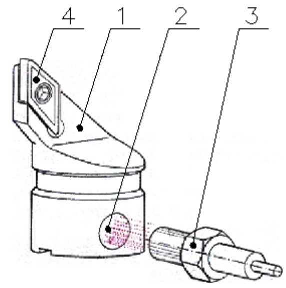


Fig. 8.5 Tool encoding using chips
1 – tool holder, 2 – memory module, 3 – sensing/write head, 4 – replaceable cutting insert

Binary encoding of tools

It is coded so that the identification number is broken down by the binary system (Fig. 8.6). The coefficient 0 is expressed by means of a coding ring with a smaller outer diameter and the coefficient 1 is expressed by a coding ring with a larger outer diameter.

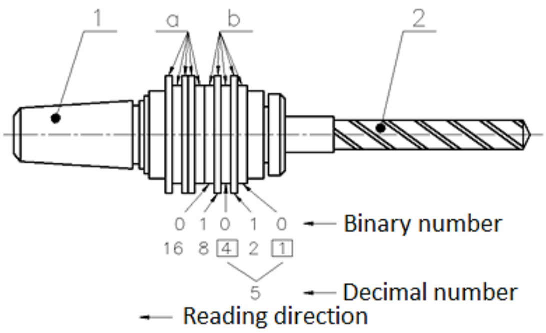


Fig. 8.6 Binary encoding of tools using coding rings
1 – standard tool holder, MORSE cone, 2 – tool that allows pre-ordering outside the machine

Encryption with a plastic sticker (card)
Tool labels use plastic labels in which the tool code is marked with holes. The encoding method does not allow writing and rewriting information on the tools and the tool can be positioned at any point of the tool magazine (Fig. 8.7).

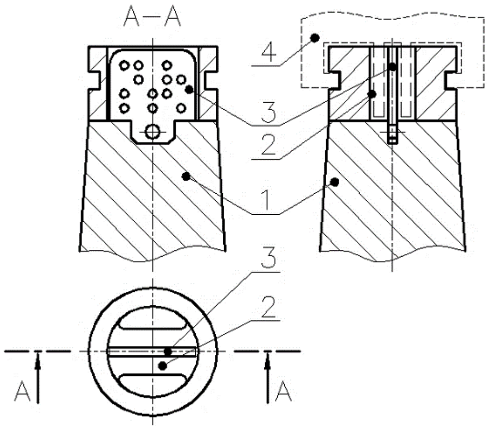


Fig. 8.7 Coding of tools with plastic processing card
1 – the tool holder, 2 – slot shank, 3 – plastic coding element, 4 – reading head

8.3 Automatic Tool Changer Systems
Many different types of mechanisms have been designed for storing and changing tools. The three most important are turret head, carousel storage with spindle direct changing, and matrix magazine storage with pivot insertion tool changer. Tool storage magazines may be horizontal or vertical.

Turret Head
This type of system is found on older NC drilling machines. The tools are stored in the spindles of a device called a turret head. When a tool is called by the program, the turret rotates (indexes) it into position. The tool can be used immediately without having to be inserted into a spindle. Thus, turret head designs provide for very fast tool changes. The main disadvantage of turret head changers is the limit on the number of tool spindles that can be used. See Fig. 8.8.

Fig. 8.8 A turret head tool changing system by FANUC ROBODRILL α-T21iF

Carousel Storage with Spindle Direct Tool Changer

Systems of this type are usually found in vertical machining centers. Tools are stored in a coded drum called a carousel. The drum rotates to the space where the current tool is to be stored. It moves up and removes the current tool, then rotates the new tool into position and places it into the spindle. On larger systems, the spindle moves to the carousel during a tool change. See Fig. 8.9, 8.10.

Fig. 8.9 The system carousel storage with spindle direct tool changer on milling center (Laboratory at the Faculty of Mechanical Engineering, Slovak University of Technology Bratislava)

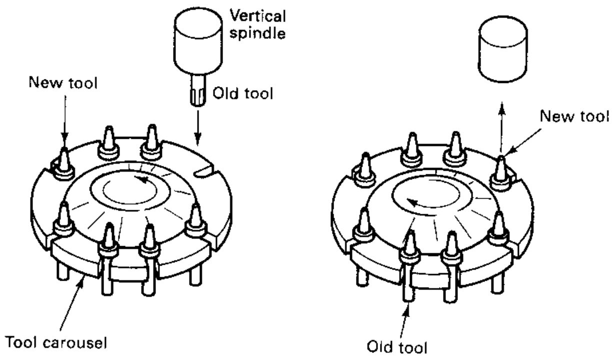


Fig. 8.10 A carousel storage system with spindle direct tool changer

Horizontal Storage Matrix Magazine with Pivot Insertion Tool Changer

Chain-type storage matrix magazines have been popular in machining centers since early 1972. This type of system permits an operator to load many tools in a relatively small space. The chain may be located on the side or the top of the CNC machine. These positions enable tools to be stored away from the spindle and work. This will ensure a minimum of chip interference with the storage mechanism and a maximum of tool protection.

Upon entering a programmed tool change, the system advances to the proper tool via the chain mechanism. The pivot arm rotates and picks up both the new tool in the magazine and the old tool in the spindle. The magazine then advances to the space where the old tool is to be stored. The arm executes a rotation again and inserts the new tool into the spindle and the old tool into the magazine. A final rotation returns the arm back to its parked position. These steps are illustrated in Fig. 8.11.

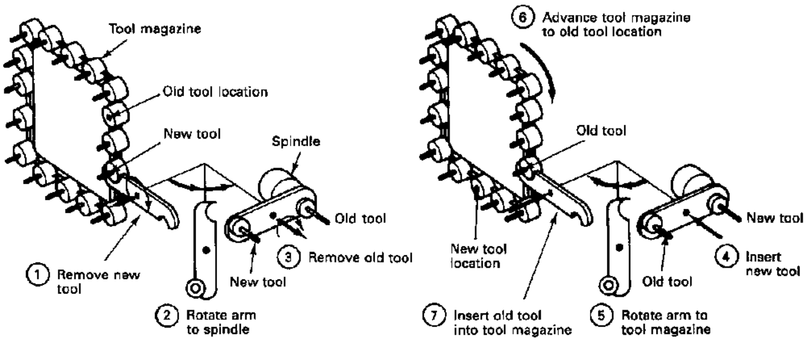


Fig. 8.11 A storage matrix magazine and pivot insert tool changer

From time to time, owing to wear or breakage, cutting tools have to be replaced. Such changes need to be rapid, with the minimum loss of machining time.

If the machining program is to remain valid, one of two requirements must be met:

1. The replacement tool must be dimensionally identical to the original.
2. The program must be capable of temporary modification to accommodate the tool variations.

Identical replacement tooling can be achieved by using qualified or preset tooling. Temporary program modifications are achieved by offsetting the tool from its original datum. See Fig. 8.12.

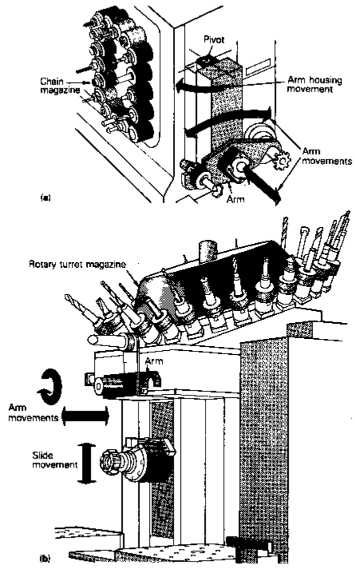


Fig. 8.12 Auto tool change

Two methods of tool identification are currently in use. One is the bar code designation. The code is imprinted and fastened to the tool. When the program calls for a specific tool, the controller looks for a particular tool code, not a specific location. Another tool identification system uses a computer microchip that is part of the tool or tool holder. The microchip contains the tool identification number and information related to the parameters of the tool. A special sensor reads the data and transfers it to the machine controller.

8.4 Pallet Loading Systems

Pallet loading systems represent another means of cutting down on machine idle time due to setup operations. One of the main goals with pallets is to keep the machine tool running. A setup person can set up a job on an idle pallet. It can then be automatically loaded into the machine as soon as the part that is running is finished. See Fig. 8.13.

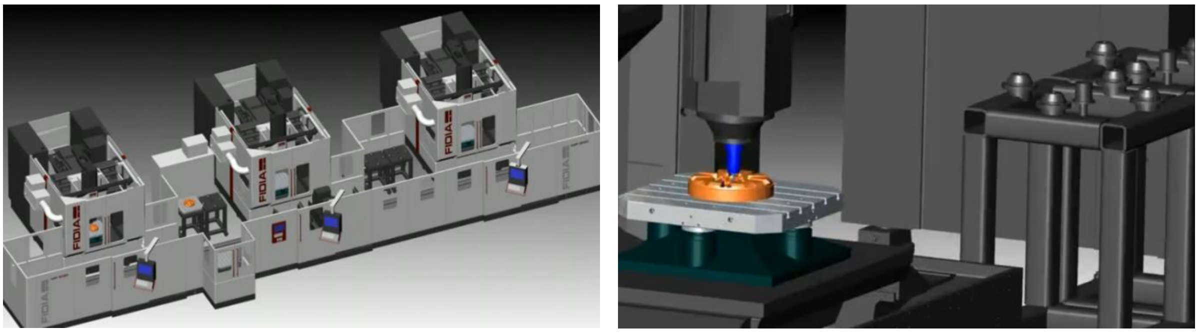


Fig. 8.13 Flexible manufacturing systems 3-630 Island – 3 Machines G996 handling 6 Pallets MP630 Fidia CNC[4]

https://www.youtube.com/watch?v=0WjZWNy4iwY
A basic pallet system is normally composed of the following elements:
The pallet: holds the work directly or by means of a fixture.
The pallet loader: moves the pallet from the load/unload station to the pallet receiver/holder.
The pallet receiver/holder: grips and positions the pallet in the CNC machine.

Pallets are manufactured in a variety of shapes and sizes. These include round, square, rectangular, and three- and four-jaw chuck holding. The size and shape of the part to be manufactured will determine the type of pallet used.
A typical two-pallet operation is described below. Refer to Fig. 8.14 for a corresponding illustration.

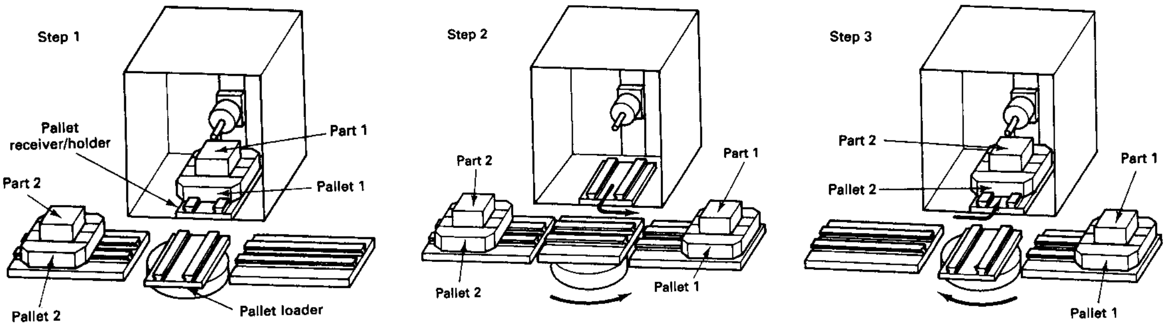


Fig. 8.14 The operations involved in a two-pallet work changing system

Automatic Pallet Changer System:
https://www.youtube.com/watch?v=z6CoFuLyMqc
https://www.youtube.com/watch?v=HeWSkR8sW3A

Step 1: Assume pallet 2 is on the loader that is currently off-line. The operator sets up a job by mounting the work to a fixture bolted to pallet 2. The operator then enters information into the MCU indicating that this pallet is ready.
Step 2: After completing the part program for the part on pallet 1, the MCU directs the machine to pull it from the receiver onto the loader. Pallet 1 is removed and placed at an off-line holding location.

Step 3: Pallet 2 is then loaded into the receiver and locked into position. If the new job is identical to the last job run, the same part program is retained. If, however, a different type of job is to be run, the machine must be signaled to load the corresponding part program. This information can be entered manually or automatically read from a microchip or bar code encoded on pallet 2. The part program is run for the work on pallet 2. The completed part on pallet 1 is removed. A new job is set up on pallet 1, thereby maintaining a continuous work cycle.

The two pallets are rotated in and out of the machining center until the entire job run for all parts is completed.

Pallet loading systems have been adapted to all types of CNC machines including vertical and horizontal machining centers.

9

Programming of CNC Lathe – EMCO WinNC SINUMERIK 810/820 T

The software EMCO WinNC SINUMERIK 810/820 T Turning is a part of the EMCO education concept on PC basis. Target of this concept is learning to operate and program the original control at the PC. The turning machines of the EMCO PC TURN und CONCEPT TURN series can be directly controlled via PC by means of the EMCO WinNC for the EMCO TURN [5].

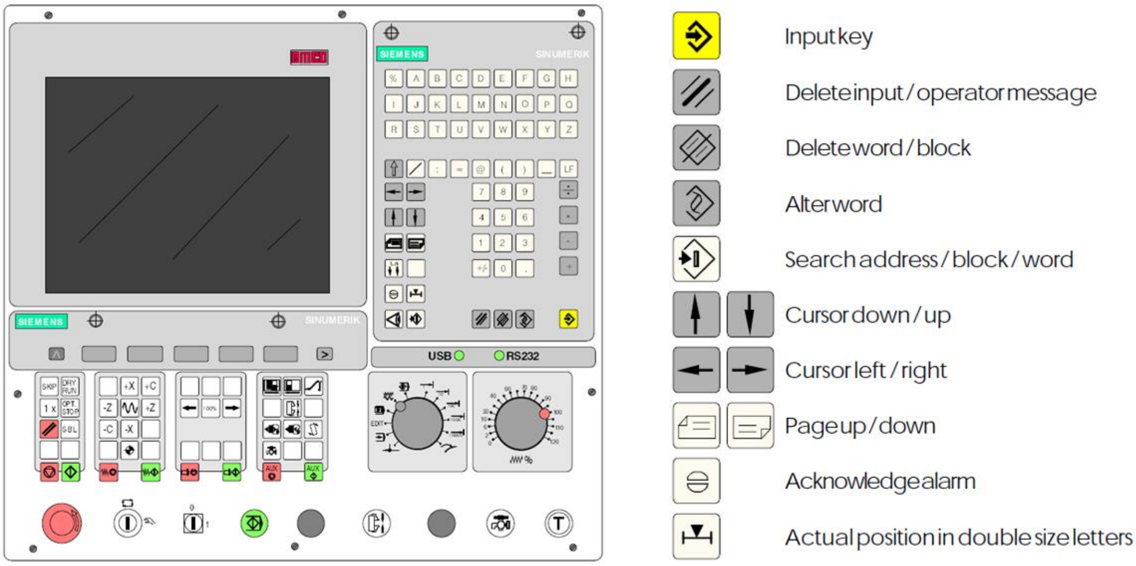


Fig. 9.1 Control keyboard and key functions

Machine Control Keys

The machine control keys are in the lower block of the control keyboard resp. the digitizer overlay. Depending on the used machine and the used accessories not all functions may be active.

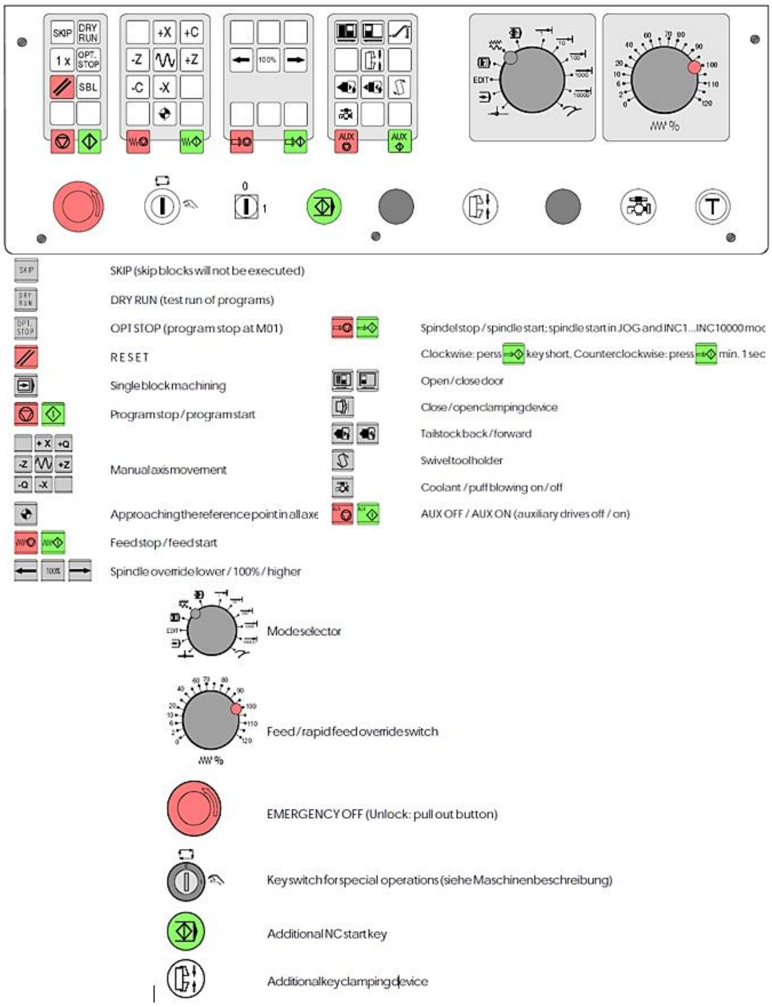


Fig. 9.2 Machine control keys

Screen with Softkeys

At the operating field the following parts are defined (Fig. 9.3):

1 - Display of the mode
2 - Display of the operating conditions
3 - Display of the alarm number, text(comment)
4 - Display of the notes to the operator
5 - Display of inputs from the keyboard
6 - Display of the channel number
7 - Display of the softkey functions
8 - Key "jump back to a higher-level menu" (key F2 at the PC)
9 - Softkeys (keys F3-F7 at the PC)
10 - Key "Further functions in the same menu" (key F11 at the PC)

Softkeys (9) are keys with multiple meaning. The valid meaning will be displayed at the bottom line (7) of the screen.

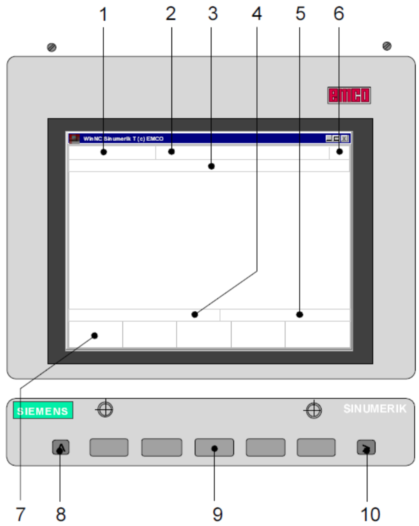


Fig. 9.3 Screen with softkeys

Important Points on a CNC Machine Tool

M = Machine Zero Point
An unchangeable reference point established by the machine manufacturer. Proceeding from this point the entire machine is measured. At the same time "M" is the origin of the coordinate system.

R = Reference Point
A position in the machine working area which is determined exactly by limit switches. The slide positions are reported to the control by the slides approaching the "R". Required after every power failure.

N = Tool Mount Reference Point
Starting point for the measurement of the tools. "N" lies at a suitable point on the tool holder system and is established by the machine manufacturer.

W = Workpiece Zero Point
Starting point for the dimensions in the part program. Can be freely established by the programmer and moved as desired within the part program.

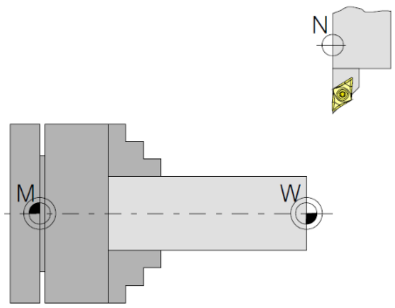


Fig. 9.4 Reference points in the working area

Zero Offset

With EMCO lathes the machine zero "M" lies on the rotating axis and on the end face of the spindle flange. This position is unsuitable as a starting point for dimensioning. With the so-called zero offset the coordinate system can be moved to a suitable point in the working area of the machine.

In the setting data zero offset (Fig. 9.5), are four adjustable zero offsets available [5].

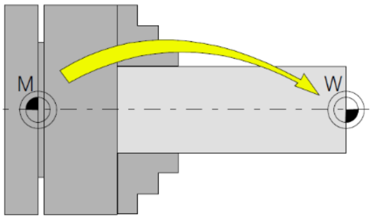


Fig. 9.5 Zero offset from the machine zero point to the workpiece zero point

When you define a value in the offset register, this value will be considered with call up in program (G54-G57) and the coordinate zero point will be shifted from the machine zero M to the workpiece zero W.
The workpiece zero point can be shifted within a program with "G58, G59 – programmable zero offset" in any number.
More information see in command description G58, G59.

Coordinate System
The X coordinate lies in the direction of the cross slide, the Z coordinate lies in the direction of the longitudinal slide. Coordinate values in minus direction describe movements of the tool system towards the workpiece, values in plus direction away from the workpiece.
Coordinate System with Absolute Programming
The origin of the coordinate system lies at the machine zero "M" or at the workpiece zero "W" following a programmed zero offset.
All target points are described from the origin of the coordinate system by the indication of the respective X and Z distances.
X distances are indicated as the diameter (as dimensioned on the drawing)
Coordinate System with Incremental Programming
The origin of the coordinate system lies at the tool mount reference point "N" or at the cutting tip after a tool call-up.
With incremental programming the actual paths of the tool (from point to point) are described.
X will be entered as radius.

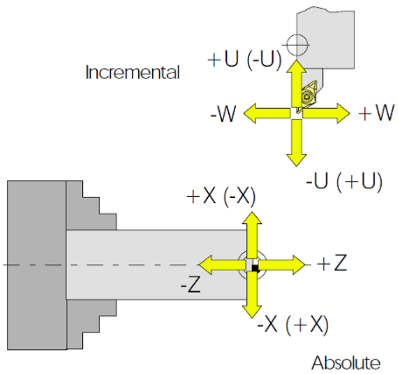


Fig. 9.6 Absolute coordinates relate to a fixed position, incremental coordinates to the tool position

Input of the Zero Offset (Fig. 9.7).
Four zero offsets can be entered (e.g. for four different clamping devices):
- Press the softkey SETTING DATA in any mode.

- Press the softkey ZERO OFFSET.
- The screen shows the input pattern for the zero offset G54. The particular offsets G54-G57 can be selected with softkeys.
- Below ZERO OFFSET the measured values (e.g.: X=0, Z=length of chuck) are entered.
- Corrections to this value can be entered below ZO ADDIT. These corrections will be added.

Move the cursor to the value to be altered with the keys .

Enter the new value and press the key .

The inverse input mark jumps to the next input field.

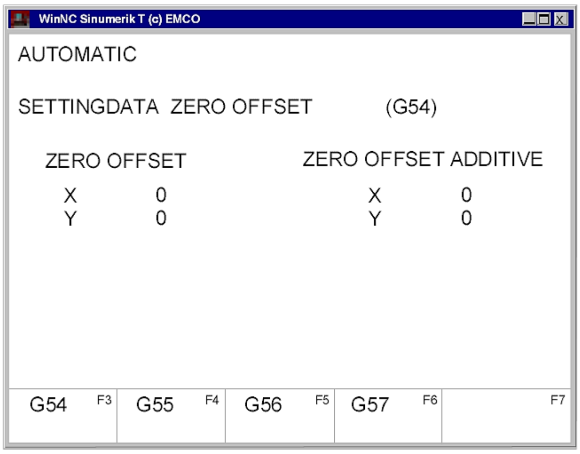


Fig. 9.7 Input pattern for Zero Offset G54

Tool Data Measuring

Aim of the tool data measuring:

The CNC should use the tool tip for positioning, not the tool mount reference point.

Every tool which is used for machining must be measured. The distances in both axis directions between tool mount reference point and tool tip are to be measured.

In the so-called tool register the measured length corrections, the cutter radius and the cutter position can be stored.

Every tool offset number D1-D99 is related to a tool.

The correction number can be any register number but must be considered with tool call in program.

Example

The length corrections of a tool in the tool turret station 4 have been stored as correction number 41. Tool call in program: **T4 D41**

The address **T** marks the position in the tool turret, the address **D** marks the correction number belonging to the tool.

The length corrections can be measured half-automatically, cutter radius and cutter position must be inserted manually.

The cutter position must always be inserted! Inserting the cutter radius is only necessary for using the cutter radius compensation with this tool.

The tool data measuring occurs for type 1-9:

L1 : in -X direction absolute from point "N" in radius

L2 : in -Z direction absolute from point "N"

R: cutter radius

Type: cutter position (1-9)

The tool data measuring occurs for type 10:
L1 : in -Z direction absolute from point "N"
Type: drilling tool (10)

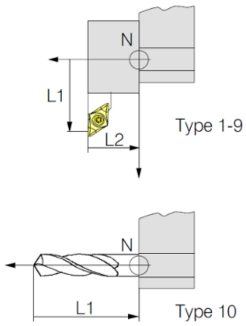


Fig. 9.8 Direction of the length correction for the tool types

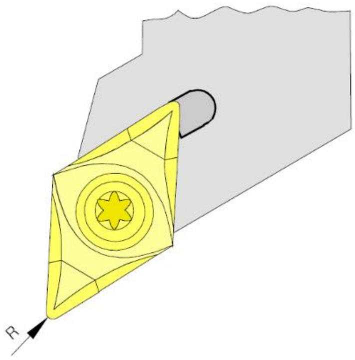


Fig. 9.9 Tool radius R

Cutter position (Type)
Look at the tool like it is clamped at the machine to determine the tool type [5].
With "**wear**" occurs the correction of not exact measured tool data or worn tools after several machining runs. The inserted length corrections will be added to or subtracted from the geometry of the tool incrementally.
X+/- incremental in diameter
Z+/- incremental
R+/- incremental

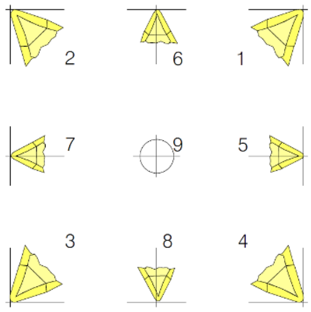


Fig. 9.10 Cutter position

Input of the Tool Data
Select the softkey TOOL OFFSET in any mode. The screen shows the input pattern for tool data.
The screen shows the input pattern for tool data.

• Select the desired tool offset number with the keys and or by entry of the correction number and the key "search" (e.g. _ 5).

• Position the cursor (inverse mark) with the keys and to the desired input field. Enter the desired value with the numeric keyboard.
The entered value will be shown at the input line of the screen.

• Store the correction value with the key in the tool offset register.
The cursor jumps to the next input position resp. after input of the last value to the first value of the next tool offset number.
Additive input with , delete with .

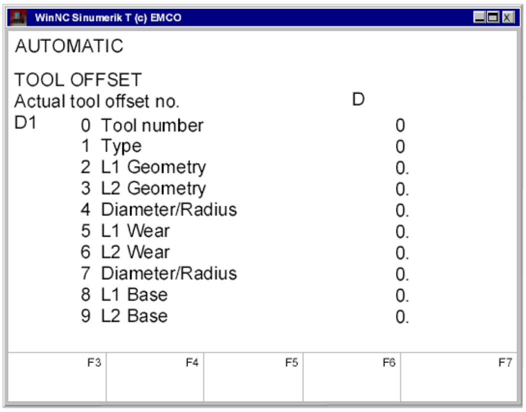


Fig. 9.11 Input pattern for tool data

Tool Data Measuring with the Optical Presetting Device
The optical way is more precise because touching will be avoided and the tool is displayed enlarged in the optics.
Manual Calculation

- Mount the optical presetting device in the working area in a way that the measuring point can be reached with the reference tool and with all tools to be measured.
- Select the mode JOG
- Mount the reference tool at station 1 of the tool turret
- Swivel in station 1
- Traverse the tip of the reference tool into the reticule of the optics.

Note: An object viewed through the optics is mirrored in the X and Z axis.

- Read and note actual slide position displayed at the screen.
- The tip of the reference tool is at X height of the tool mount reference point N and in Z direction 22 mm (PC TURN 120/125): 20 mm tool length + 2mm outstanding length of the holder) in front of N.
  $Z_N = Z-22$ $X_N = X$
- Swivel tool turret and traverse with the first tool to be measured into the reticule.
- The difference between the old position of the tool mount reference point(X_N, Z_N) and the new position are the tool offsets L1, L2.
  The X values are in diameter and must be halved because L1 is a radius value.
- Clamp next tool etc.

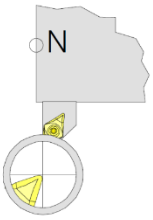


Fig. 9.12 Traverse into the graticule with the tool

Survey Modes

AUTOMATIC

For working off a part program the control calls up block after block and interprets them.
The interpretation considers all corrections which are called up by the program.
The so-handled blocks will be worked off one by one.

JOG

With the JOG keys the tool can be traversed manually. In the submode OVERSTORE (softkey) you can switch on the spindle and swivel the tool turret.

MDI-AUTOMATIC

You can enter blocks of a part program in the intermediate store.
The control works off the inserted blocks and deletes the intermediate store for new entries.

REFPOINT

This mode is used to approach the reference point. With reaching the reference point the actual position store is set to the value of the reference point coordinates. By that the control acknowledges the position of the tool in the working area.
With the following situations the reference point has to be approached:

- After switching on the machine
- After mains interruption
- After alarm "Approach reference point" or "Ref. point not reached".
- After collisions or if the slides stucked because of overload.

I 1 ... I 10 000 1 ... 10000

In this mode the slides can be traversed for the desired increment (1...10000 in µm/10^{-4} inch) with means of the JOG keys -X +X -Z +Z.
The selected increment (1, 10, 100, ...) must be larger than the machine resolution (smallest possible traverse path), otherwise no movement will occur.

Approach the Reference Point

By approaching the reference point the control will be synchronized with the machine.

- Select the mode REFPOINT.
- Press the JOG keys -X or +X resp. -Z or +Z to approach the reference point in the respective direction.
- With the key REF ALL both axes will be approached automatically (PC keyboard).

Input of Programs

Part programs and subroutines can be entered in the modes JOG, AUTOMATIC, INC 1 ... INC 10 000 and REFPOINT.

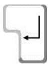
Call up an existing or new program

- Press softkey PART PROGRAM
- Press softkey EDIT
- Enter program number %... or L...

Press softkey SELECT PROGRAM. Blocks in an existing program will be displayed.

Input of a block

Example:
Block number (not necessary)
1. word
2. word

- LineFeed – block end (with PC  keyboard)

Insert Block
Position the cursor before the block that should follow the inserted block and enter the block to be inserted.

Delete Block
Position the cursor before the block, enter block number (if no block number; N0) and press .

key

Insert Word
Position the cursor before the word, that should follow the inserted word and enter the word and value) to be inserted and . press

Alter Word
Position the cursor before the word to be altered, enter word and .
press

Delete  **Word**
Position the cursor before the word to be deleted, enter the address (e.g. X) and press the key.

10

Programming

Program Structure

NC programming for machine tools according to DIN 66025 is used. The NC program is a sequence of program blocks which are stored in the control. With machining of workpieces these blocks will be read and checked by the computer in the programmed order. The corresponding control signals will be sent to the machine.

The NC program consists of:

- Program number
- NC blocks
- Words
- Address
- Number combinations (partly with sign)

Addresses

% program number 1 - 9999
L subroutine number 1 - 9999
N block number 1 - 9999
G path function
M miscellaneous function
A angle
B radius (pos. sign), chamfer (neg. sign), circle radius
D tool offset 1 - 49 feed,
F feed, dwell
I, K circle parameter, thread pitch
P number of subroutine runs, scale factor
R parameter for cycles
S spindle speed, cutting speed
T tool call (tool turret position)
X, Z position data (X also dwell)
LF block end

Survey of G Commands

G00 Positioning (rapid traverse)
G01[11] Linear interpolation
G02 Circular interpolation clockwise
G03 Circular interpolation counterclockwise
G04[2] Dwell
G09[2] Exact stop
G10 Polar coordinate interpolation, rapid traverse
G11 Polar coordinate interpolation, linear interpolation
G12 Polar coordinate interpolation, circular interpolation clockwise
G13 Polar coordinate interpolation, circular interpolation counterclockwise
G16 Plane selection with free axis selection
G18 Plane ZX
G25 Minimum working area limitation
G26 Maximum working area limitation
G33 Thread cutting
G40[1] Cancel cutter radius compensation
G41 Cutter radius compensation left
G42 Cutter radius compensation right
G48[2] Leave as approached
G50[1] Cancel scale modification
G51 Scale modification
G53[22] Cancel zero offset blockwise
G54[1] Zero offset 1
G55 Zero offset 2
G56 Zero offset 3
G57 Zero offset 4
G58 Programmable zero offset 1
G59 Programmable zero offset 2
G60 Exact stop mode
G62 Deselection exact stop mode
G63 Thread tapping with compensation chuck
G64[1] Deselection exact stop mode

[1] Initial status

[2] Effective blockwise

G70 Measuring in inches

G71 Measuring in millimeters

G80 Deselection of drilling cycles

G90[1] Absolute programming

G91 Incremental programming

G92[2] Spindle speed limit

G92 P Cylindrical interpolation ON

G92 P1 Cylindrical interpolation OFF

G94 Feed per minute

G95[1] Feed per revolution

G96 Constant cutting speed, feed per revolution

G972 Constant spindle speed

G130 Transmit OFF

G131 Transmit ON

G147[2] Soft approach to contour with linear

G148[2] Soft leaving with linear

G247[2] Soft approach to contour with quarter circle

G248[2] Soft leaving with quarter circle

G347[2] Soft approach to contour with semicircle

G348[2] Soft leaving with semicircle

Survey of M Commands

M00 Programmed stop unconditional

M01 Programmed stop conditional

M02 Main program end

M03 Spindle ON clockwise

M04 Spindle ON counterclockwise

M05[1] Spindle OFF

M08 Coolant ON

M09[1] Coolant OFF

M17 Subroutine end

M20 Tailstock BACK

M21 Tailstock FORWARD

M25 Open clamping device

M26 Close clamping device

M30 Main program end

M52 C- axis ON

M53 C- axis OFF

M71 Puff blowing ON

M721 Puff blowing OFF

Survey of Cycles

L93 Cut-in cycle

L94 Undercut cycle

L95 Stock removal with back pockets

L96 Stock removal without back pockets

L97 Thread cutting cycle

L98 Deep hole drilling cycle

L99 Thread draft

L971 Longitudinal thread

10.1 Description of G Commands

G00 Positioning (Rapid Traverse)

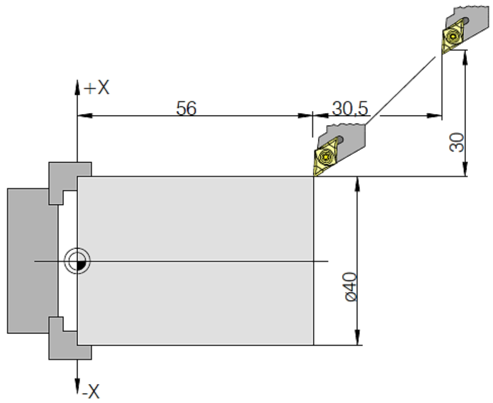


Fig. 10.1 Absolute and incremental measures

Format

N.... G00 X... Z...

The slides are traversed with maximum speed to the programmed target point (tool change position, start point for following machining) [5].

Note:

- A programmed feed F is suppressed while G01
- The maximum feed is defined by the producer of the machine
- The feed override switch is active.

Example:

Absolute G90

N50 G90 G00 X40 Z56

Incremental G91

N50 G91 G00 X-30 Z-30.5

G01 Linear Interpolation

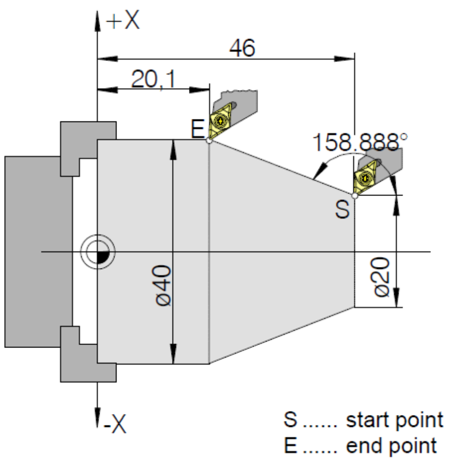


Fig. 10.2 Absolute and incremental measures [5]

Format

N... G01 X... Z F

Straight movements with programmed feed in mm/ rev (initial status).

Example:

Absolute G90

N20 G01 X40 Z20.1 F0.1 or

N20 G01 X40 A158.888 F0.1

Incremental G91

N20 G01 X10 Z-25.9 F0.1

G02 Circular Interpolation Clockwise

G03 Circular Interpolation Counterclockwise

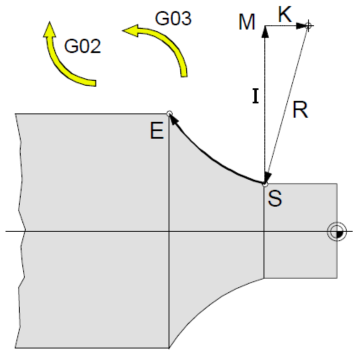


Fig. 10.3 Parameters and rotational direction

Format

N... G02/G03 X... Z... I... K... F... or

N... G02/G03 X... Z... B... F...

X, Z End point of the arc (absolute or incremental)

I, K Incremental circle parameter (Distance from the start point to center of arc, I is related to X, K to Z-axis)

B Radius of the arc (arc smaller than semicircle with +B, larger than semicircle with -B), can be entered instead of I, K

The tool will be traversed to the end point along the defined arc with the programmed feed F.

Note:

Programming the value 0 for I or K can be omitted. The position of the circle end point will be checked, a tolerance of 100 pm (computing and rounding errors) is allowed.

According to DIN 66025 the observation of G02, G03 is always behind the turning axis, no matter whether machining occurs behind or in front of the turning axis.

G04 Dwell

Format

N... G04 X/F... [sec]

The tool movements will be stopped for a time defined by X or F (in the last reached position) - sharp edges – transitions, cleaning cut-in ground, exact stop

Note:

The dwell time starts at the moment when the tool movement speed is zero.

Example

N75 G04 X2.5 (dwell time = 2.5sec)

G33 Thread Cutting

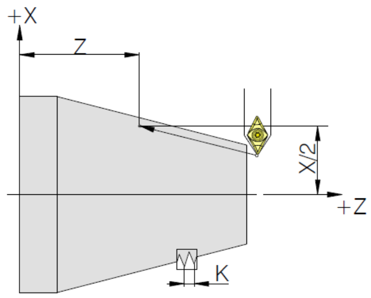


Fig. 10.4 Measures for thread cutting

Format

N... G33 X... Z... I/K...

I/K Thread pitch [mm]
I in X direction (longitudinal)
K in Z direction (face)
Straight, tapered and scroll threads can be cut. For thread pitch I or K has to be entered according to the main direction of the thread (longitudinal or face). Machining routines like knurling are also possible.
Note:
Feed and spindle override are not active with G33 (100%).
A fair sized undercut has to be machined first.

Cutter Radius Compensation

During tool measurement the tool tip is merely measured at two points (touching the X and Z axes).
The tool offset therefore only describes a theoretical cutter tip.
This point is traversed on the workpiece in the programmed path. With movements in the axis directions (longitudinal and face turning) the points on the tool tip touching the axes are used. No dimensional errors are produced on the workpiece.
With simultaneous movements in both axis directions (tapers, radii) the position of the theoretical cutting point no longer coincides with the point on the tip actually cutting. Dimensional errors occur on the workpiece.
Maximum contour error without cutter radius compensation with 45° movements: Cutter radius 0,4 mm ≅ 0,16 mm path distance ≅ 0,24 mm distance in X and Z.
If cutter radius compensation is used, these dimensional errors are automatically calculated and compensated by the control.

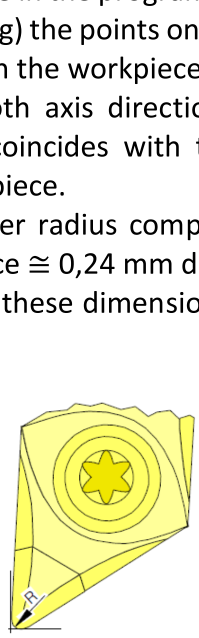


Fig. 10.5 Tip radius and theoretical cutter tip

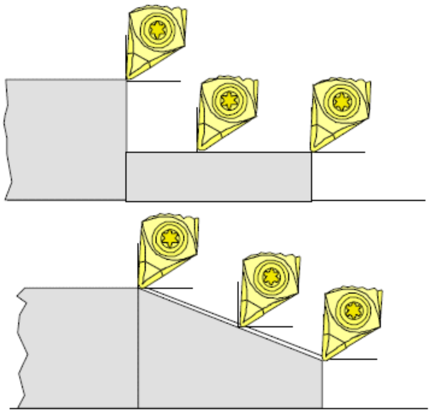


Fig. 10.6 Cutting movements parallel to the axes and slant [5]

G40 Cancel Cutter Radius Compensation

The cutter radius compensation will be cancelled by G40.
Cancellation is only permitted with a linear traversing command (G00, G01).
G00 or G01 can be programmed in the same block or as the first traversing movement after cancellation.
G40 is programmed frequently in the block with return to the tool change position.

G41 Cutter Radius Compensation Left

If the programmed tool path (viewed in the direction of machining) is on the **left** of the material to be machined, the cutter radius is to be selected with G41.

Note:

- No direct change between G41 and G42 – cancel with G40 previously.
- Cutter radius R and cutter position (tool type) must be defined.
- Selection is only permitted in conjunction with G00 or G01.
- Change of the tool correction is not possible while cutter radius compensation.

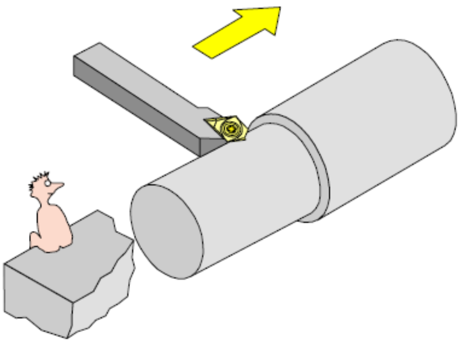


Fig. 10.7 Definition G41 cutter radius compensation left [5]

G42 Cutter Radius Compensation Right

If the programmed tool path (viewed in the direction of machining) is on the right of the material to be machined, the cutter radius is to be selected with G42.

Note: see G41!

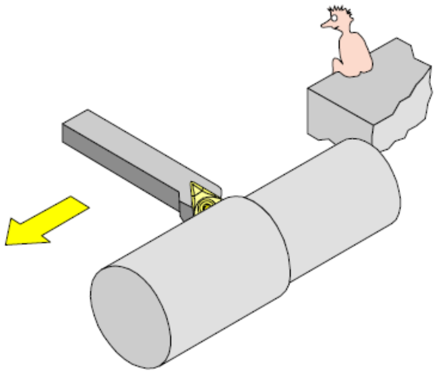


Fig. 10.8 Definition G42 cutter radius compensation right

G70 Measuring in Inches
Format

N5 G70

By programming the following data will be transformed into the inch system:
- Path information X, Z
- Interpolation parameter I, K
- Chamfers, radii +B, -B

Notes:
- For clearness G70 should be defined in the first block of the program.
- A change between G70 and G71 within a program is allowed.
- A steady setting of the measuring system mm/inch will be proceeded in DIAGNOSIS, NC-MD. This setting is relevant for all values and will be kept also with power off/on.

G71 Measuring in Millimeters
Format

N5 G71

Comment and notes like G70!

G90 Absolute Programming
Format

N... G90

The addresses have to be programmed as following:
X..........diameter
Z+/- ......absolute (referred to the workpiece zero point)
Note
• Direct switchover within a block between G90 and G91 is not possible.
• G90 (G91) may also be programmed with some other G functions (N... G90 G00 X... Z...).

G91 Incremental Programming
Format

N... G91

The addresses have to be programmed as following:
X..........radius
Z ....Incremental (real) traverse path with sign
Note see G90.

G92 Spindle Speed Limit
Format

N... G92 S... [U/min]

With G92 a maximum spindle speed (rev/min) can be determined for a part program (effective only in connection with G96). Enter the desired maximum speed with address S. This command is used in conjunction with constant cutting speed.

G94 Feed per Minute
By G94 all F (feed) values are in mm/min (inch/min).

G95 Feed per Revolution
By G95 all F (feed) values are in mm/revolution (inch/revolution).

G96 Constant Cutting Speed
Unit: m/min feet/min
The control computes the spindle speed for the actual diameter continuously. With diameters decreasing to 0 the speed should increase to infinite. In fact it increases to the maximum speed of the machine and the program runs on without alarm.
If the maximum speed is to high (e.g.: limited chuck speed, unbalanced work pieces, ...) G92 has to be programmed additionally to delimit the speed.
Feed is automatically set to G95 (mm/rev, inch/rev). No zero offset in X direction must be active.

G97 Constant Spindle Speed
Unit: rev/min
G96 will be deselected and the spindle speed will be held at the last valid value. Afterwards S will be programmed in rev/min.

10.2 Description of M Commands

M Commands are switching or additional functions (miscellaneous). The M commands can stand alone in a program block or together with other commands. Commands of the same group cancel each other that means the M command programmed last cancels the previously programmed M command of the same group.
Remark
The following pages describe the standard M commands. Whether these M commands are executable depends on the type of the machine and the used accessories

M00 Programmed Stop Unconditional
These command effects a stop in the execution of the part program.
Main spindle, feed and coolant will be switched off. The chip protection door can be opened without triggering an alarm.
With "NC start" the program run can be continued. After that the main drive will be switched on with all values which were valid before.

M01 Programmed Stop Conditional
M01 works like M00, but only if the function PRO-GRAMMED STOP YES was switched on by softkey in the menu PROGRAM CONTROL.

With “NC start” the program run can be continued. After that the main drive will be switched on with all values which were valid before.

M02 Main Program End
M02 works like M30.

M03 Main Spindle ON Clockwise
The spindle is switched on provided that a spindle speed or cutting speed has been programmed, the chip protection door is closed and a workpiece correctly clamped.
M03 must be used for all right-hand cutting or overhead clamped tools, if machining occurs behind the turning center.

M04 Main Spindle ON Counterclockwise
The same conditions as described under M03 apply here.
M04 must be used for all left-hand cutting or normal clamped tools, if machining occurs behind the turning center.

M05 Main Spindle Off
The main drive is braked electrically.
At the program end the main spindle is automatically switched off.

M08 Coolant ON
Only for EMCO PC Turn 120. The coolant will be switched on.

M09 Coolant OFF
Only for EMCO PC Turn 120/125. The coolant will be switched off.

M17 Subroutine End
M17 will be written in the last block of a subroutine. It can stand alone in this block or with other functions. The call-up of a subroutine and M17 must not stand in the same block (nesting).

M30 Main Program End
With M30 all drives are switched off and the control is returned to the start of the program. Moreover, the counter level is increased by 1.

10.3 Description of Cycles

Cycles will be programmed in the program in a manner that first the R parameters will be written into the program and then the cycle call with the number of program runs.
Example
N... R20=... R21=... R22=... R24=... R25=... R26=... R27=... R28=... R29=... R30=... L95 P2
That means that the cycle L95 with the programmed parameters will run 2 times.

L93 Cut-in Cycle
The cut-in cycle allows machining of grooves longitudinal, face, outside and inside.
Note:

Both cutting edges of the cut-in tool have to be measured in **neighboring** tool data registers (e.g.: D21 and D22).

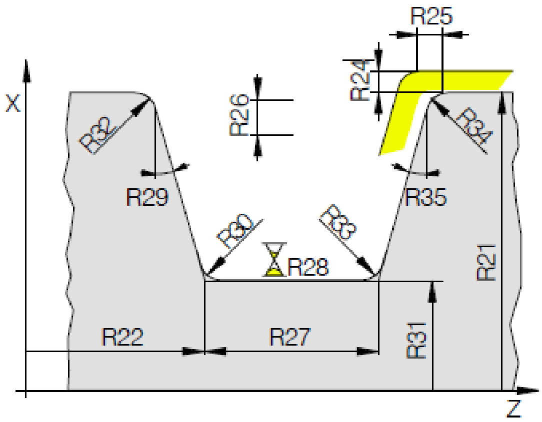


Fig. 10.9 Dimensions for cut – in

Programming the parameter:

R10 0 for longitudinal, 1 for face cut-in
R21 Outer resp. inner diameter
R22 Start point in Z
R23 Determine the start point
Longitudinal cut-in
1 inside/outside right
-1 inside/outside left
Face cut-in
1 inside right/left
-1 outside right left
R24 Finishing offset in X
R25 Finishing offset in Z
R26 Infeed depth
R27 Width of cut-in
R28 Dwell of the bottom of the cut-in
R29 Angle left flank (0° - 89°)
R30 Radius (+) or chamfer (-) at the bottom left
R31 Cut-in diameter
R32 Radius or chamfer at edge of cut-in left
R33 Radius or chamfer at bottom of cut-in right
R34 Radius or chamfer at edge of cut-in right
R35 Angle right flank

L95 Stock Removal with Back Pockets

L96 Stock Removal without Back Pockets

Machining can be programmed longitudinal, on face, inside, outside.

For L95 max. 10 back pockets (in machining direction decreasing diameters) may be programmed. The first contour point will be determined in the circle. The contour draft has to be described in a subroutine (G1, G2, G3).

The last point in the subroutine determines the start diameter of the roughing sequence. For face operation this point must be the highest point in X, for longitudinal operation this must be the lowest point in Z.
The first block in the subroutine must be programmed in absolute coordinates.

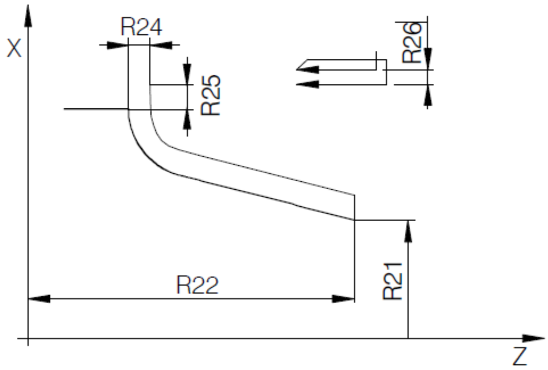


Fig. 10.10 Dimensions for the stock removal cycle

Programming the parameter:

R20 Selection of the desired contour (subroutine number)
R21 Start point of the contour in X
R22 Start point of the contour in Z
R24 Finishing offset in X
R25 Finishing offset in Z
R24 and R25 are valid for roughing and finishing, with R24=0 and R25=0 the finish dimension will be reached
R26 Roughing infeed (X or Z), will be omitted with finishing
R27 Selection cutter radius compensation (40, 41, 42), the control activates the cutter radius compensation automatically, correct selection and deselection automatically
R28 Feed
R29 Machining procedure, see Fig. 10.11
R30 Feed factor with back pockets
With this number (e.g.: 0,7) the feed will be multiplied for dive in movements with back pockets (feed reduction)

R29=XX	1X	2X	3X	4X
X1	Roughing axis parallel longitudinal outside	Finishing to finishing offset longitudinal outside	Roughing axis parallel and one roughing cut contour parallel (rest edges) longitudinal outside	Complete machining (roughing, rest eedges, finishing) longitudinal outside
X2	Roughing axis parallel face outside	Finishing to finishing offset face outside	Roughing axis parallel and one roughing cut contour parallel (rest edges) face outside	Complete machining (roughing, rest eedges, finishing) face outside
X3	Roughing axis parallel longitudinal inside	Finishing to finishing offset longitudinal inside	Roughing axis parallel and one roughing cut contour parallel (rest edges) longitudinal inside	Complete machining (roughing, rest eedges, finishing) longitudinal inside
X4	Roughing axis parallel face inside	Finishing to finishing offset face inside	Roughing axis parallel and one roughing cut contour parallel (rest edges) face inside	Complete machining (roughing, rest eedges, finishing) face inside

Fig. 10.11 Machining possibilities with R29

L97 Thread Cutting Cycle

Longitudinal, face, inside and outside threads can be programmed. The infeed of the tool occurs automatically and is digressively quadratic. This keeps the cross section of the chip constant.

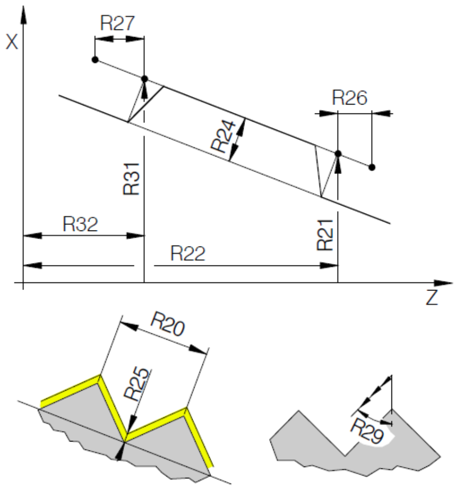


Fig. 10.12 Thread cutting cycle

Programming the parameter:

R20 Thread pitch (always the value parallel to axis)
R21 Start point of the thread in X
R22 Start point of the thread in Z
R21 and R22 describe the real start point at the contour.
R23 Number of idle cuts
R24 Thread depth (positive value = inside thread, negative value = outside thread)
R25 Finishing offset
After automatic cut dividing with roughing occurs a finishing cut with the programmed cutting depth.
R26 Thread let-in, incremental without sign
R27 Thread let-out, incremental without sign
The values R26 and R27 always will be entered parallel to axis without sign. With taper threads the correct starting and end points will be calculated automatically.
R28 Number of roughing cuts
R29 Infeed angle (half flank angle)
Flank infeed is only possible with longitudinal or face threads (no taper threads)
R31 End point of the thread in X (absolute)
R32 End point of the thread in Z (absolute)
R31 and R32 describe the real end point at the contour.

L98 Deep Hole Drilling Cycle

This cycle is for drilling deep holes or for drilling in materials with bad cutting property. With the parameter R11 the retraction movement can be determined.

Chip breaking (R11=0)

The drill dives into the work piece to the first drilling depth (R25), dwells (duration R27), retracts 1 mm and dives in again.

Retracting (R11=1)

The drill dives into the work piece to the first drilling depth (R25), dwells (duration R27), retracts out of the drilling hole (to R22), dwells (duration R28), and dives in again.

The following infeed is always smaller for the value R24 than the previous. The sequence infeed – retraction will be repeated until the end depth is reached. If the calculated infeed is smaller than R24, it will be constant with the value R24.
If the remaining infeed down to end depth R26 is smaller than the double degression value (2xR24), the remaining infeed will be halved and worked off in two infeeds. Therefore the smallest infeed never can be less than R24/2.

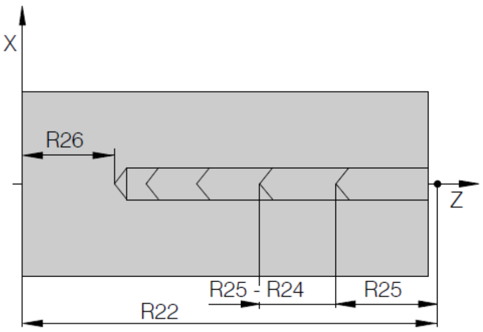


Fig. 10.13 Deep hole drilling cycle

Programming the parameter:

R11 0 – chip breaking
    1 – retracting
R22 Start point in Z (absolute)
R24 Degression value (incremental, without sign)
R25 First drilling depth (incremental, without sign)
R26 End drilling depth (absolute)
R27 Dwell at start point (for retracting only)
R28 Dwell at drilling depth (chip breaking and retracting)

10.4 Subroutines

Function sequences which are repeated multiple can be programmed as subroutines. Contour descriptions for cycles also will be entered as subroutine.
The numbers L90-L100 are reserved for cycles and must not be used for subroutines.

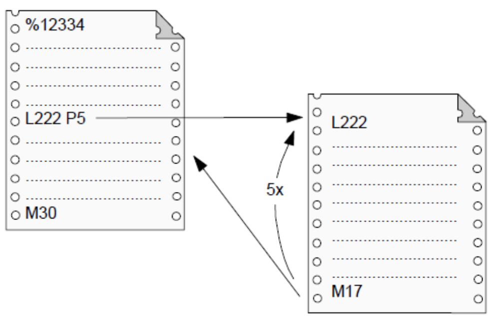


Fig. 10.14 Multiple call of a subroutine

Subroutine Call in Part Program

e.g.: L123 P1 LF

L subroutine

123 subroutine number

P1 number of subroutine runs (max. 99)

Subroutine End with M17

e.g.: N150 M17 LF

10.5 Illustrative example

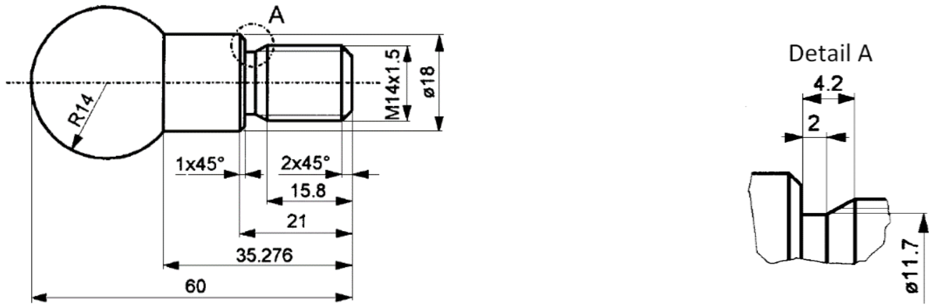


Fig. 10.15 Ball pin

Notes on the technological process of component manufacturing

The workpiece is made from duralumin, cutting speeds are about 200 m/min for roughing and about 220 m/min for finishing with feeds of 0.2 mm/rev resp. 0.1 mm/rev. We produce the component for two clamps in the universal chuck. When creating a program, we use cycles to draw a contour of the component in the subroutines.

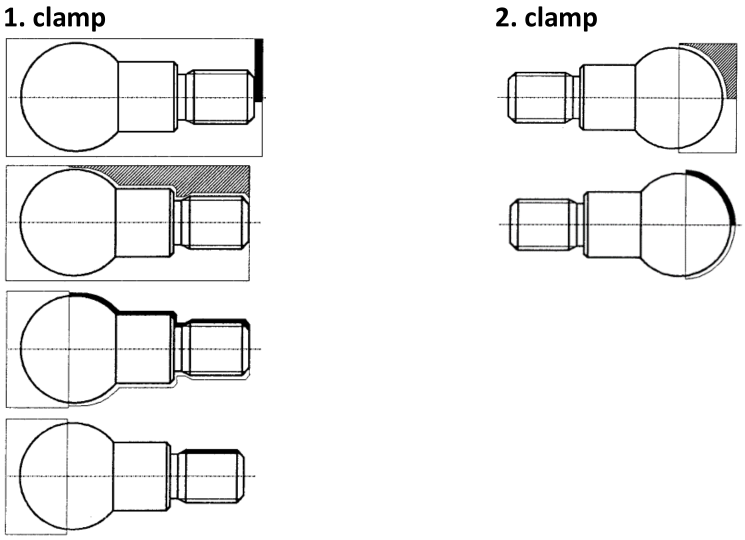


Fig. 10.16 Process of machining of the two sides of workpieces

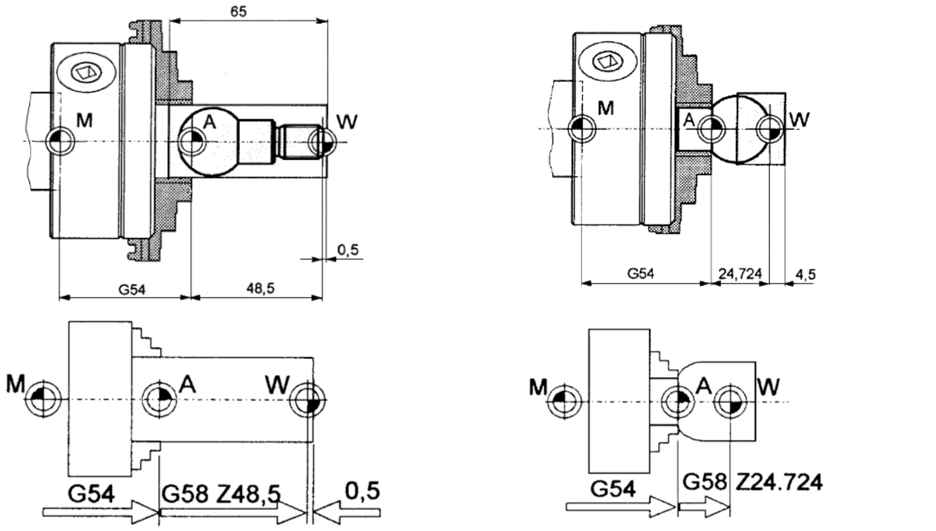
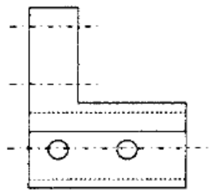
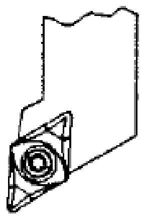
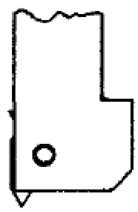


Fig. 10.17 Situation at first and second clamping with marked zero point displacements

When designing a technological process, it is necessary to determine the tools to produce individual surfaces. The tools must be clamped into the tool turret, set correction numbers, determine the correction sizes in both axes and the position of the cutting tool tips.

Type of the tool	**Parameters**
Stop for rod material (tool holder) 	Position in the tool turret: T1 Number of correction: D1 Position of the tool tip (cutter position): 7
Left – hand facing tool 	Position in the tool turret: T2 Number of correction: D2 Cutter radius: 0,4 Position of the tool tip (cutter position): 3 Cutting speed: 200 resp. 220 m/min Feed: 0,2 resp. 0,1 mm/rev
Treading tool 	Position in the tool turret: T3 Number of correction: D3 Cutter radius: 0 Position of the tool tip (cutter position): 8 spindle rotational speed 800 rev/min

After these steps, we can proceed to writing the CNC program itself. For clarity, we will introduce the program and the description of some activities.

Program	**Description**
%100 N0005 G54 N0010 G58 Z48.5 N0015 T1 D1 N0020 G94 F1000 N0025 G1 X0 Z0.5 N0030 M00 N0035 G0 X80 Z20 N0040 T2 D2 N0045 G95 S2500 F0.1 M4 N0050 G0 X32 Z0 N0055 G1 X-1 N0060 G0 X32 Z2 N0065 G96 S200 F0.2 M4 N0070 R20 = 100 R21 = 10 R22 = 0 R24 = 0.3 R25 = 0.1 R26 = 1.5 R27 = 40 R29 = 31 R28 = 0.2 R30 = 0.5 L95 P1 N0075 G96 S220 F0.1 M4 N0080 R20 = 100 R21 = 10 R22 = 0 R24 = 0 R25 = 0 R26 = 1.5 R27 = 42 R29 = 21 R28 = 0.1 R30 = 0.5 L95 P1 N0085 G0 X50 Z30 M5 N0090 T3 D3 N0095 G95 S800 M3 N0100 G0 X20 Z5 N0105 R20 = 1.5 R21 = 14 R22 = 0 R23 = 1 R24 = - 0,92 R25 = 0.05 R26 = 1 R27 = 1 R28 = 6 R29 = 30 R31 = 14 R32 = - 17 L97 P1 N0110 G0 X50 Z50 M5 N0115 M30	**1. clamping** - zero offset from the machine - program zero offset to the workpiece - tool Nr. 1 - feed rate 1000 mm/min - start to the position with the tool 1 - programming stop unconditional, put the product to touch - position for the tool change - left – hand facing tool - rotational speed (revolutions) 2500 rev/min, feed 0.1 mm/rev, revolution of the spindle - left - facing (turning of the face) - cutting speed 200 m/min, feed 0.2 mm/rev, turning the spindle on the left - roughing cycle according to the contour of the component (workpiece) - cutting speed 220 m/min, feed 0.1 mm/rev, turning the spindle on the left - finishing cycle according to the contour of the component - position for the tool change, stop of the spindle - Treading tool (revolution 800 rev/min, revolution of the spindle - right - cycle of thread cutting - position for the tool change, stop of the spindle - end of the program
L100 N0005 G1 X14 Z – 2 N0010 Z – 15.8 N0015 X11.7 Z – 18 N0020 Z – 20 N0025 X18 B – 1 N0030 Z35.276 N0035 G3 X28 Z – 46 I – 9 K – 10.724 N0040 G1 Z – 47 N0045 X30 N0050 M17	**Subroutine** – contour of the workpiece 1. clamp

%101 N0005 G54 N0010 G58 Z24.724 N0015 T2 D2 N0020 G96 S200 F0.2 M4 N0025 G0 X32 Z10 N0030 R20 = 101 R21 = -1 R22 = 6 R24 = 0.3 R25 = 0.1 R26 = 1.5 R27 = 40 R29 = 31 R28 = 0.2 R30 = 0.5 L96 P1 N0035 G96 X220 F0.1 M4 N0040 R20 = 101 R21 = 0 R22 = 6 R24 = 0 R25 = 0 R26 = 1.5 R27 = 42 R29 = 21 R28 = 0.1 R30 = 0.5 L96 P1 N0045 G0 X50 Z40 M5 N0050 M30	**2. clamping** - zero offset from the machine - program zero offset to the workpiece - left – hand facing tool - cutting speed 200 m/min, feed 0.2 mm/rev, turning the spindle on the left - roughing cycle according to the contour of the component (workpiece) - cutting speed 220 m/min, feed 0.1 mm/rev, turning the spindle on the left - finishing cycle according to the contour of the component - position for the tool change, stop of the spindle - end of the program
L101 N0005 G1 Z0 N0007 X0 N0010 G3 X28 Z-14 B14 N0015 G1 Z-16 N0020 X30 N0025 M17	**Subroutine** – contour of the workpiece 2. clamp

11

CNC Programming For Milling

11.1 Machine Basics

Machine Axes
CNC Milling machine has 3, 4 and 5 axes. For the Spinner VC810, there are only 3 axes.

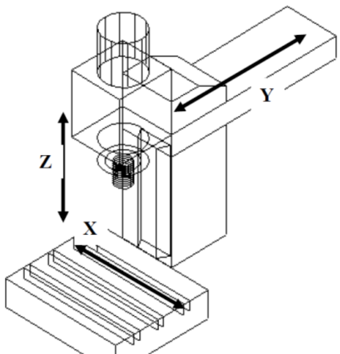


Fig. 11.1 3 axes of CNC Milling

Machine Zero Point
The Machine Zero Point, often referred to as the HOME position, or sometimes the Machine Origin is located at the top, away from the working table. This position has been set by the machine manufacturer. At this position, the reading for X, Y and Z axis for the machine position equals "0" ie. X = 0, Y = 0, and Z = 0.
There are 2 ways of returning the Spindle to its Home position.

- By pressing the origin button / switch , followed by the Start Switch. All the three axis, X,Y and Z, moves automatically to the Home position .
- By using a program: G91 G28 Z0;
  G28 X0 Y0;

All machining processes have to start from the Machine Zero Point. It is good to remember the two lines of the program because all programs that will be written will contain the two lines, one at the beginning of a program and another at the end.

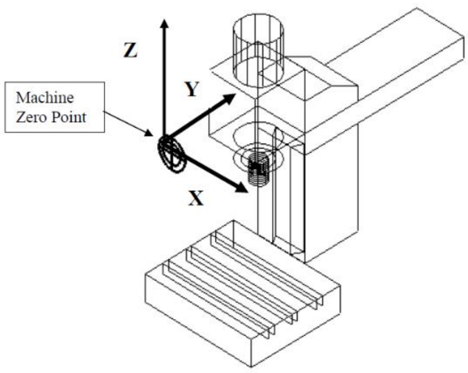


Fig. 11.2 Machine Zero Point

Work Part Zero Point

The Reference Point of a workpiece or sometimes referred to as the Work part Zero Point is where the X, Y and Z coordinates equals "0". The distance from the machine zero point (Home) to the work part zero point is called the Workshift and the Tool Length Offset.

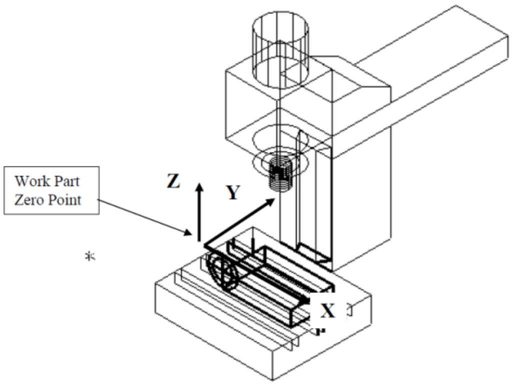


Fig. 11.3 Work Zero Point

Axis During Machining

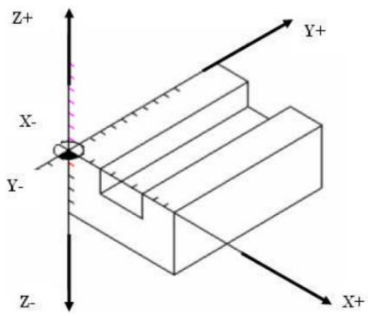


Fig. 11.4 Axis during machining

Symbols used for 'points' identification:

M	machine zero point
W	work part zero point
R	reference point
E	tool reference point
B	tool setup point
A	tool shank point
N	tool change point

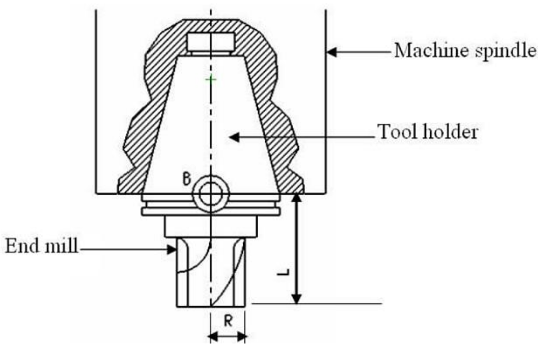


Fig. 11.5 Tool Length Compensation
B – Tool Setup point, L – Tool length compensation, R – Radius compensation

11.2 Tool Compensation for CNC Milling

a) Types of tools available in CNC milling

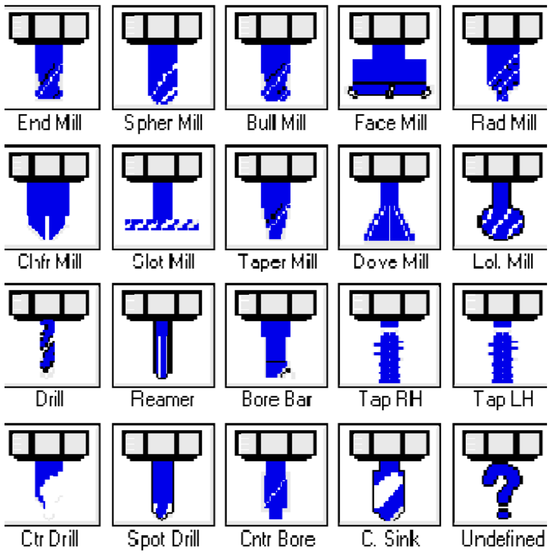


Fig. 11.6 Tool Compensation for CNC Milling

11.3 NC Programming Basics

Standard Formatting

A NC program will be written in a standard format and consist of the following basic structure [10]:

O 1234;	Program Number: Letter O followed by 4 digits.
N10 ;	
N20 ;	
N30 ;	
N40 ;	
N50 ;	
N60 ;	
N70 ;	
N80 ;	
N90 ;	

"N" refers to the Block Numbers

All blocks ends with a ";" or often referred to as **EOB** or End Of Block

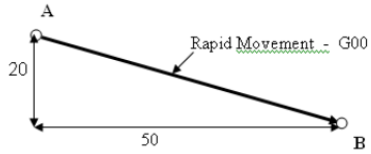
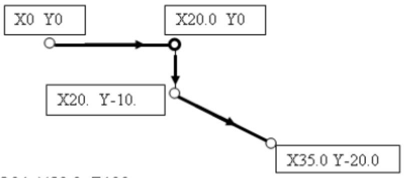
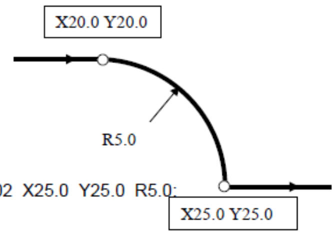
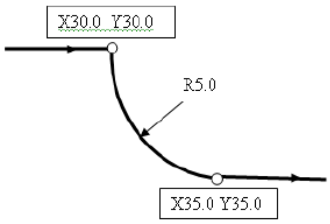
The sequence of the words in an NC-block is as follows:

	Address	Defination
1	N	block numbers
2	G	G-functions
3	X,Y,Z	Coordinates
4	I,J,K	Interpolation parameter
5	F	Feed
6	S	Speed
7	T	Tool position
8	M	Additional functions

M-Codes

Code	Task	Functions
M00	Program stop	This code is used to stop a program that is being executed.
M01	Optional Program Stop	These codes performed the same task as the M00 code, but the programmer/machine operator still have a choice of whether to stop or not at the desired stopping position. Selections are done by pressing the Optional Stop Button. If the optional stop button is depressed or not chosen, machine will not stop even when the M01 code is displayed or reached.
M02	Program End	This code is placed at the end of a program. This code is also used to stop all axis movement, spindle rotation, coolant pump and able to cut off the machine power supply automatically. Control will be reset to program start.
M03	Spindle Clockwise Direction	Starts spindle rotation in the clockwise direction. When it is needed to rotate the spindle at 1500 rpm in the clockwise direction, program is written as: M03 S1500;
M04	Spindle Counter Clockwise Direction	Starts spindle rotation in the counterclockwise direction. Program is written as: M04 S1500;
M05	Spindle Stop	Stops spindle rotation either in M03 or M04.
M06	Tool Change	This code is used to change to the tool required for cutting. Program is written as: M06 T02;
M08	Coolant Pump “On”	To turn on the coolant pump
M09	Coolant Pump “Off”	To turn off the coolant pump
M30	Program End	Has the same function as M02.

G-CODES

Code	Task	Functions
G00	Rapid Traverse.	This code is used to move the spindle head or the cutting tool to its destination from one point A, to another point, B with rapid movement of 15000mm/min. However, this code cannot be used for cutting purposes.  G00 X50.0 Y20.0 ;
G01	Linear Interpolation	This code is used when cutting in a straight direction from one point to another. Applies to vertical, horizontal or at an angle (slope) movement. The feedrate, F must be stated in the program.  G01 X20.0 F100.; Y-10.0; X35.0 Y-20.0;
G02	Circular Interpolation - Clockwise	This code is used when cutting arcs or circles in the clockwise direction.  G02 X25.0 Y25.0 R5.0;
G03	Circular Interpolation -Counter-Clockwise	This code is used when cutting arcs or circles in the counter-clockwise direction.  G03 X35.0 Y35.0 R5.0 ;
G04	Dwell	This code is used when to stop the cutting movement for a few seconds in order for better finishing or in the case of drilling, a better hole surface. Although the axis movement of the cutting tool has stopped, the rotation of the cutting tool remains active. Duration of the stop time can be define by using either X or P. G04 X4.0; (stops for 4 sec.) Or G04 P3000; (stops for 3 sec.)

By using only code G01, G02 and G03, write a program for Exercise 1 and Exercise 2 shown below, on the program sheet provided. You are now at point A. Proceed to the next point B, then C, D,....... following in alphabetical order until you reach your start point A.

Exercise 1:

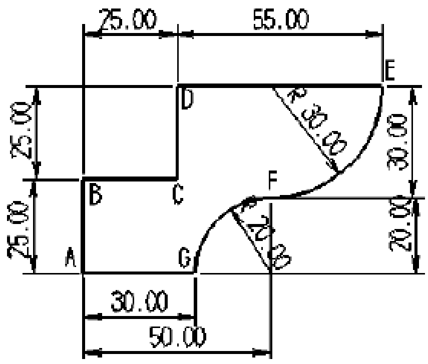


Figure 1

Exercise 2:

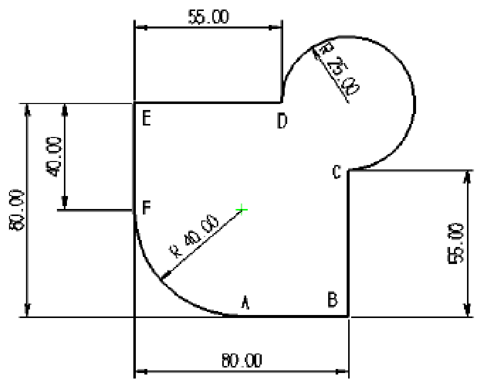


Figure 2

The Absolute System (G90) and Incremental System (G91)

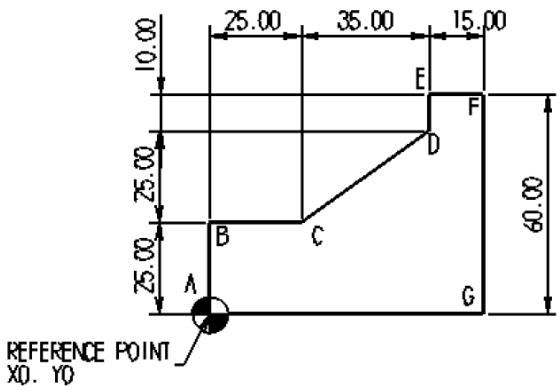
When writing a program, it is necessary to choose either one of the following coordinate systems:

- the absolute system, G90 or
- the incremental system, G91

Even though a program may consist of both the systems, it must be separated at different sections, and not mixed.

The Absolute System – G90

When using this system, the distant from one point to another in the X or Y axis must be taken from the reference point and does not depend on the distance of travel. The reference point is where the X, Y and Z coordinates equals zero, ie. X0 ,Y0 and Z0 (see example below).



Program:

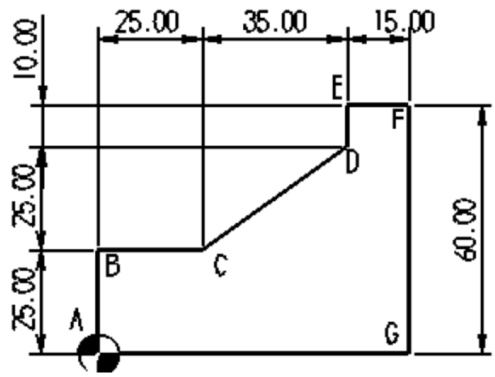
The Absolute System – G90

G90	(Absolute System)
G01 Y25.0	(Point B)
X25.0	(Point C)
X60.0 Y50.0	(Point D)
Y60.0	(Point E)

X75.0	(Point F)
Y0	(Point G)
X0	(Back to Point A)

The Incremental System – G91

Reference point is not necessary when using this system. When programming, it refers to the distance of travel either in the X or Y direction or in both. In the X -axis, movement to the right is positive and movement to the left is negative. In the Y-axis, movement upwards is positive and movement downwards is negative (see example below).



G91	(Incremental System)
G01 Y25.0	(Point B)
X25.0	(Point C)
X35.0 Y25.0	(Point D)
Y10.0	(Point E)
X15.0	(Point F)
Y-60.0	(Point G)
X-75.0	(Back to Point A)

Coordinate System Setting G92, G54-G59

The work coordinate system refers to a point on the workpiece where the reference point or the work part zero point is located, and is measured from the Home position (machine zero point). These distances, consisting of the X-axis and Y-axis are referred to as the Work shift and the Z-axis is referred to as the Tool Length Offset. Since the machine is unable to trace the location of the work part visually, these 3 values of X, Y and Z are guidance for the machine to know exactly where the work part has been placed.

To obtain the values for X, Y and Z, the machine operator has to first clamp the work part in position and to return the tool to its origin or the home position. Using the Jog mode, the tool is brought slowly to the work part zero point, and its position is taken at the machine axis position from the monitor.

For example, the reading taken is:

X 120.333
Y 24.789
Z 139.225

There are two ways of notifying the machine or the computer the location of the work part.

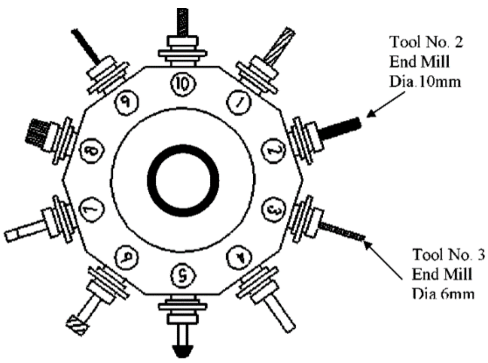
- Using code G92 and writing the X,Y and Z values in the program. This method is suitable when programming with a single cutting tool.
- In the program it will be written as: **G92 X120.333 Y24.789 Z139.225**
- Using code G54 and G43 in the program and installing the X and Y values in the Work shift (G54 – G59) and Z values in the Tool Offset (G43) on the monitor. This method is suitable when programming with a single cutting tool or using more than one cutting tool.

  In the program it will be written as:

G54
G43 H01

Tool Length Offset – G43

A CNC machine with Automatic Tool Change (ATC) facilities, ranging from 8 tools to 200 tools are normally equipped with different types of tools such as End Mills, Ball Ends, Drills, Taps, Reamers, Fly Cutters, Edge Finders, Slot Cutters, etc. Each of these tools has different lengths and diameters and there must be a way of recording their values.
The diagram below shows an automatic tool change (ATC) device that holds 10 tools of different sizes and shapes. They are many designs for the ATC, depending on the machine manufacturer. Some ATC have tools placed vertically, while others are placed horizontally. No matter what the design could be, these machine manufacturers always try to reduce the time taken for a tool change. Every second counts.



The Automatic Tool Change (ATC) with 10 Tools mounted on a circular turret

12

TNC 640 Heidenhain

12.1 TNC 640

Besides milling, the TNC 640 from HEIDENHAIN is also capable of combined milling and turning operations. It is particularly well suited for milling-turning, HSC and 5-axis machining on machines with up to 18 axes. The workshop oriented and versatile control features numerous functions. It is especially attractive for the following areas of application [8]:

Universal milling machines
Combined milling-turning machines
High speed milling
Five-axis machining with swivel head and rotary table
Five-axis machining on very large machines
Boring mills
Machining centers and automated machining
The optimized user interface of the TNC 640 gives you a fast overview: various color coding, standardized table editors and smart Select – the dialog-guided fast selection of functions – aid you at your work.

The TNC 640 features optimized motion control, short block processing times and special control strategies. Together with its uniform digital design and its integrated digital drive control including inverters, it enables you to reach very high machining speeds and the best possible contour accuracy – particularly when machining 3-D contours.

You can program turning contours with the TNC 640 in the familiar HEIDENHAIN plain language. Beyond this, you have typical contour elements for turning (recesses, undercuts, thread undercuts) as well as cycles for complex turning operations.

The TNC 640, HEIDENHAIN's high performance mill-turn control, is popular with users thanks to its workshop-oriented operational design. Now the TNC 640 is, and will only be, available with groundbreaking touch technology that supplements the TNC 640's field-proven cycles and functions. It allows the user to operate the control screen with gestures, like smartphones or tablets [4].

12.2 Selecting the Correct Operating Mode

This chapter is intended to help users quickly learn to handle the most important procedures on the control.
The following topics are included in this chapter:
- Machine switch-on
- Programming the first part
- Graphically testing the first part
- Setting up tools
- Workpiece setup

- Running the first program

12.3 Machine Switch – on

After starting the program, such a screen can also be seen with the control keyboard.

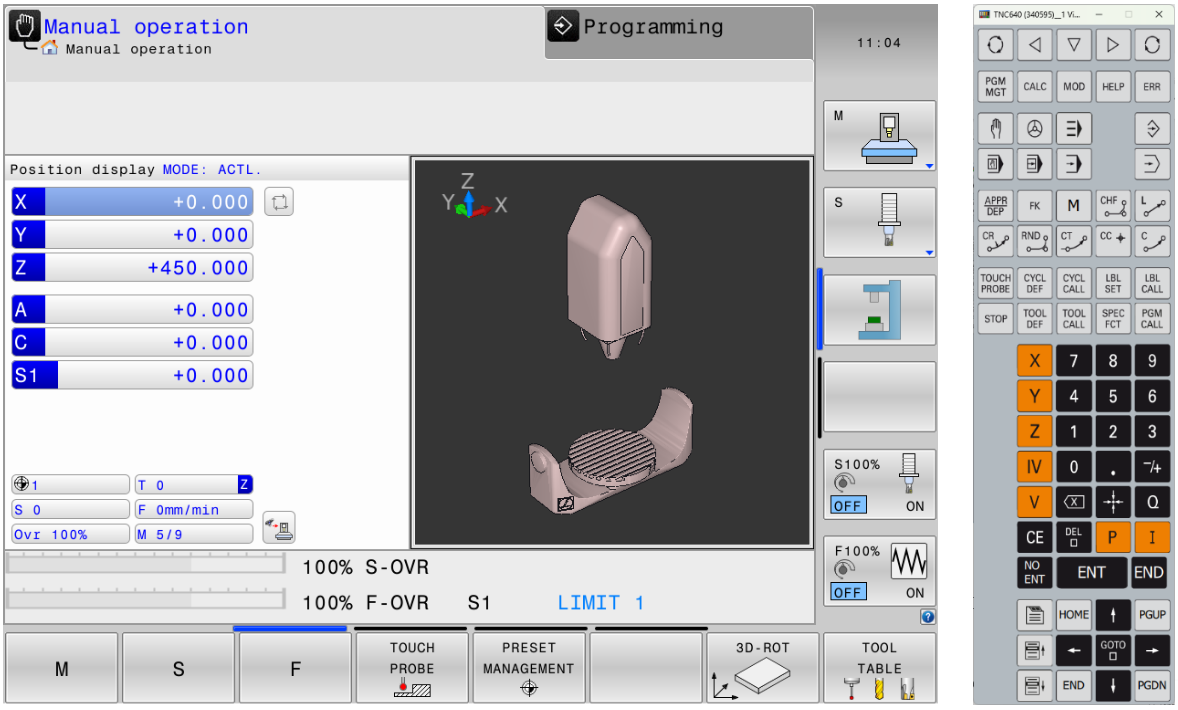


Fig. 12.1 Initial control panel with control keyboard [13]

- Switch on the power supply for control and machine
  - The control starts the operating system. This process may take several minutes.
  - The control will then display the "Power interrupted" message in the screen header.
- Press the CE key CE
  - The control compiles the PLC program.
- Switch on the machine control voltage I
  - The control checks operation of the emergency stop circuit and goes into Reference Run mode.
- Cross the reference point manually in the prescribed sequence: For each axis press the START key. If you have absolute linear and angle encoders on your machine, there is no need for a reference run
  - The control is now ready for operation in the Manual operation mode.

12.4 Controls and displays

If you are using a TNC 640 with touch control, you can replace some keystrokes with hand-to-screen contact [4].

Keys on visual display unit

Key	Function
	Selecting the screen layout
	Toggle the display between machine operating mode, programming mode, and a third desktop

	Soft keys for selecting functions on screen
◁ ▷ △	Shifting between soft-key rows

Alphanumeric keyboard

Key	Function
Q W E	File names, comments
G F S	DIN/ISO programming

Machine operating modes

Key	Function
	Manual operation
	Electronic handwheel
	Positioning with manual data input
	Program run, single block
	Program run, full sequence

Programming modes

Key	Function
	Programming
	Test run

Entering and editing coordinate axes and numbers

Key	Function
X V	Select coordinate axes or enter them in a program
0 9	Numbers
. -/+	Decimal separator/Reverse algebraic sign
P I	Polar coordinate entry/Incremental values
Q	Q parameter programming/Q parameter status
	Capture actual position
NO ENT	Skip dialog questions, delete words
ENT	Confirm entry and resume dialog
END	Conclude block and exit entry
CE	Clear entries or error message
DEL	Abort dialog, delete program section

Tool functions

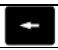
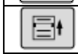
Key	Function
TOOL DEF	Define tool data in the program
TOOL CALL	Call tool data

Managing programs and files, control functions

Key	Function
PGM MGT	Select or delete programs and files, external data transfer
PGM CALL	Define program call, select datum and point tables
MOD	Select MOD functions

HELP	Display help text for NC error messages, call TNC guide
ERR	Display all current error messages
CALC	Show calculator
SPEC FCT	Show special functions
	Open the batch process manager

Navigation keys

Key	Function
 	Position the cursor
GOTO	Go directly to blocks, cycles and parameter functions
HOME	Navigate to the program start or table start
END	Navigate to the program end or end of a table line
PG UP	Navigate up one page
PG DN	Navigate down one page
	Select the next tab in forms
 	Up/down one dialog box or button

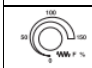
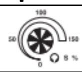
Up/down one dialog box or button

Key	Function
TOUCH PROBE	Define touch probe cycles
CYCL DEF CYCL CALL	Define and call cycles
LBL SET LBL CALL	Enter and call labels for sub programming and program section repeats
STOP	Enter program stop in a program

Programming path movements

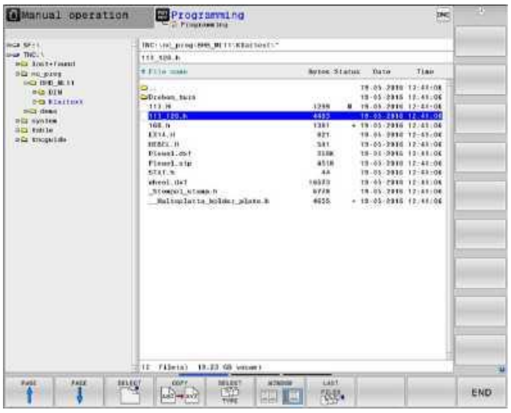
Key	Function
APPR DEP	Approach/depart contour
FK	FK free contour programming
L	Straight line
CC	Circle center/pole for polar coordinates
C	Circular arc with center
CR	Circle with radius
CT	Circular arc with tangential connection
CHF RND	Chamfer/rounding arc

Potentiometer for feed rate and spindle speed

Feed rate	Spindle speed
	

12.5 Programming of the First Part

Opening a new program/file management



- Press the PGM MGT key

The control opens the file manager. The file management of the control is arranged much like the file management on a PC with Windows Explorer. The file management enables you to manage data in the control's internal memory.
Use the arrow keys to select the folder.
If you want to open the new file you have to choose NEW

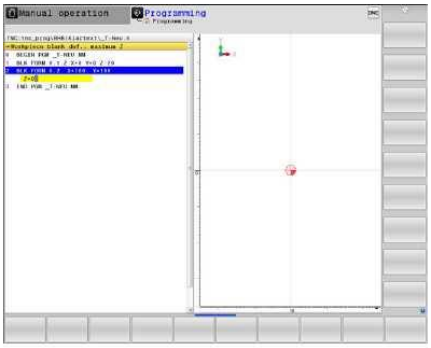
- Enter any desired file name with the extension .H

- Press the ENT key

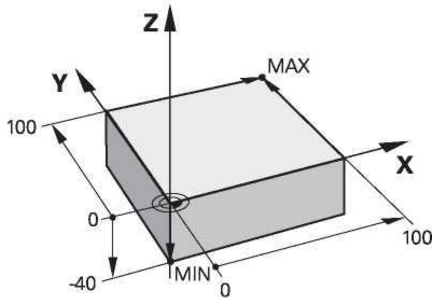
The control asks you for the unit of measure for the new program.

- Select the unit of measure: Press the MM or INCH soft key

Definition a Workpiece Blank



After you have created a new program you can define a workpiece blank. For example, define a cuboid by entering the MIN and MAX points, each with reference to the selected preset.
After you have selected the desired blank form via soft key, the control automatically initiates the workpiece blank definition and asks for the required data:



- **Working plane in graphic: XY:** Enter the active spindle axis. Z is saved as default setting. Accept with the **ENT** key
- **Workpiece blank def.: Minimum X**: Enter the smallest X coordinate of the workpiece blank with respect to the preset, e.g. 0, confirm with the **ENT** key
- **Workpiece blank def.: Minimum Y**: Enter the smallest Y coordinate of the workpiece blank with respect to the preset, e.g. 0, confirm with the **ENT** key
- **Workpiece blank def.: Minimum Z**: Enter the smallest Z coordinate of the workpiece blank with respect to the preset, e.g. -40, confirm with the **ENT** key
- **Workpiece blank def.: Maximum X**: Enter the largest X coordinate of the workpiece blank with respect to the preset, e.g. 100, confirm with the **ENT** key
- **Workpiece blank def.: Maximum Y**: Enter the largest Y coordinate of the workpiece blank with respect to the preset, e.g. 100, confirm with the **ENT** key
- **Workpiece blank def.: Maximum Z**: Enter the largest Z

coordinate of the workpiece blank with respect to the preset, e.g. 0, confirm with the **ENT** key
- The control ends the dialog.

Example:

```
0  BEGIN PGM NEW MM
1  BLK FORM 0.1 Z X+0 Y+0 Z-40
2  BLK FORM 0.2 X+100 Y+100 Z+0
3  END PGM NEW MM
```

Program Layout

NC programs should be arranged consistently in a similar manner. This makes it easier to find your place, accelerates programming and reduces errors.

Recommended program layout for simple, conventional contour machining
Example:

```
0 BEGIN PGM BSPCONT MM
1 BLK FORM 0.1 Z X... Y... Z...
2 BLK FORM 0.2 X... Y... Z...
3 TOOL CALL 5 Z S5000
4 L Z+250 R0 FMAX
5 L X... Y... R0 FMAX
6 L Z+10 R0 F3000 M13
7 APPR ... X... Y...RL F500
...
16 DEP ... X... Y... F3000 M9
17 L Z+250 R0 FMAX M2
18 END PGM BSPCONT MM
```

The program contains the following instructions:

1 Call tool, define tool axis
2 Retract the tool
3 Pre-position the tool in the working plane near the contour starting point
4 In the tool axis, position the tool above the workpiece, or preposition immediately to workpiece depth. If required, switch on the spindle/coolant
5 Contour approach
6 Contour machining
7 Contour departure
8 Retract the tool, end program

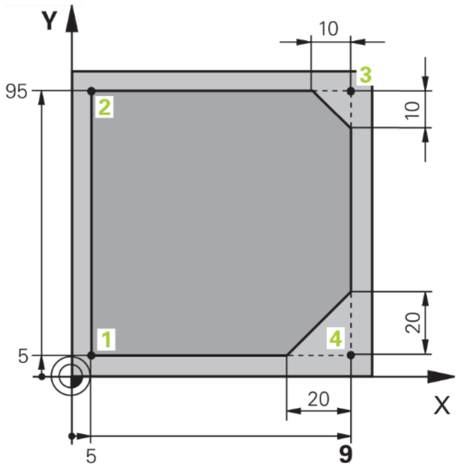
Recommended program layout for simple cycle programs
Example:

0 BEGIN PGM BSBCYC MM
1 BLK FORM 0.1 Z X... Y... Z...
2 BLK FORM 0.2 X... Y... Z...
3 TOOL CALL 5 Z S5000
4 L Z+250 R0 FMAX
5 PATTERN DEF POS1(X... Y... Z...) ...
6 CYCL DEF...
7 CYCL CALL PAT FMAX M8
8 L Z+250 R0 FMAX M2
9 END PGM BSBCYC MM

The program contains the following instructions:
1 Call tool, define tool axis
2 Retract the tool
3 Define the machining positions
4 Define the fixed cycle
5 Call the cycle, switch on the spindle/coolant
6 Retract the tool, end program

Programming a Simple Contour

The contour shown to the right is to be milled once to a depth of 5 mm. You have already defined the workpiece blank. After you have initiated a dialog through a function key, enter all the data requested by the control in the screen header [13].



- Call the tool: Enter the tool data. Confirm the entry in each case with the **ENT** key, and do not forget the **Z** 

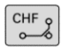
 tool axis

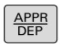
- Retracting tool: Press the orange axis key **Z** and enter the value for the position to be approached, e.g. 250. 

 Press the **ENT** key

- Confirm Tool radius comp: RL/RR/no comp?
- with the **ENT** key: Do not activate radius compensation
- Confirm **Feed F=?** with the **ENT** key: Rapid traverse (**FMAX**)
- Enter **Miscellaneous Function M?** and confirm with the **END** key
- The control stores the entered positioning block.

- Preposition the tool in the working plane: Press the orange **X** axis key and enter the value for the position  to be approached, e.g. -20
- Press the orange axis key **Y** and enter the value for the position to be approached, e.g. -20. Press the **ENT** key 

- Confirm Tool radius comp: RL/RR/no comp?
- with the **ENT** key: Do not activate radius compensation
- Confirm **Feed F=?** with the **ENT** key: Rapid traverse (**FMAX**)
- Confirm Miscellaneous Function M? with the END key
- The control stores the entered positioning block.
- Move tool to working depth: Press the orange axis key **Z** and enter the value for the position to be approached, e.g. -5. Press the  **ENT** key
- Confirm **Tool radius comp: RL/RR/no comp?** with the **ENT** key: Do not activate radius compensation 

- **Feed rate F=?** Enter the positioning feed rate, e.g. 3000 mm/min, confirm with the **ENT** key
- **Miscellaneous function M?** Switch on the spindle and coolant, e.g. **M13**, and confirm with the **END** key
- The control stores the entered positioning block.
- Move to the contour: Press the **APPR DEP**  key

- The control displays a soft-key row with approach and departure functions. 

- Press the approach function soft key **APPR CT**: Enter the coordinates of the contour starting point 1 in X and Y, e.g. 5/5, confirm with the **ENT** key
- **Center angle?** Enter the approach angle, e.g. 90°, confirm with the **ENT** key
- **Circle radius?** Enter the circular radius, e.g. 8 mm, confirm with the **ENT** key
- Confirm **Tool radius comp: RL/RR/no comp?** with the **RL** soft key: Activate the radius compensation to the left of the programmed contour
- **Feed rate F=?** Enter the machining feed rate, e.g. 700 mm/min, save your entry with the END key
- Machine the contour and move to contour point 2: You only need to enter the information that changes. In other words, enter the Y coordinate 95 and save your entry with the **END** key
- Move to contour point 3: Enter the X coordinate 95 and save your entry with the **END** key 
- Define the chamfer at contour point 3: Enter the chamfer width 10 mm and save with the **END** key
- Move to contour point 4: Enter the Y coordinate 5 and save your entry with the **END** key 
- Define the chamfer at contour point **4**: Enter the chamfer width 20 mm and save with the **END** key
- Move to contour point **1**: Enter the X coordinate 5 and save your entry with the **END** key 

- Depart contour: Press the **APPR DEP** key
- Departure function: Press the **DEP CT** soft key

- Center angle? Enter the departure angle, e.g. 90°, confirm with the **ENT** key
- Circle radius? Enter the departure radius, e.g.
- 8 mm, confirm with the **ENT** key
- Feed rate F=? Enter the positioning feed rate, e.g. 3000 mm/min, confirm with the **ENT** key
- Miscellaneous function M? Switch off the coolant, e.g. **M9**, and confirm with the **END** key
- The control stores the entered positioning block.
- Retracting tool: Press the orange axis key **Z** and enter the value for the position to be approached, e.g. 250. Press the **ENT** key

- Confirm **Tool radius comp: RL/RR/no comp**?
- with the **ENT** key: Do not activate radius compensation
- Confirm **Feed F=?** with the **ENT** key: Rapid traverse (FMAX)
- **Miscellaneous function M?** Enter **M2** to end the
- program, then confirm with the **END** key
- The control stores the entered positioning block.

Fundamental of Path Functions

Programming tool movements for workpiece machining

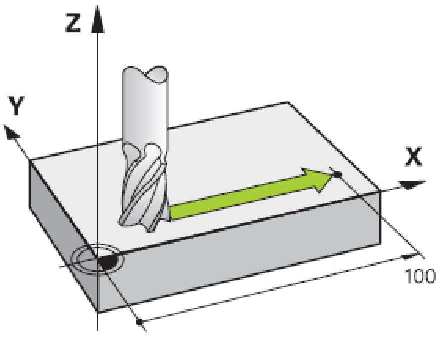
You create a part program by programming the path functions for the individual contour elements in sequence. You do this by entering the coordinates of the end points of the contour elements given in the production drawing. The control calculates the actual path of the tool from these coordinates, and from the tool data and radius compensation.

The control moves all machine axes programmed in the NC block of a path function simultaneously.

Movement parallel to the machine axes

If the NC block contains one coordinate, the control moves the tool parallel to the programmed machine axis.

Depending on the individual machine tool, the part program is executed by movement of either the tool or the machine table on which the workpiece is clamped. Path contours are programmed as if the tool were moving.



Example

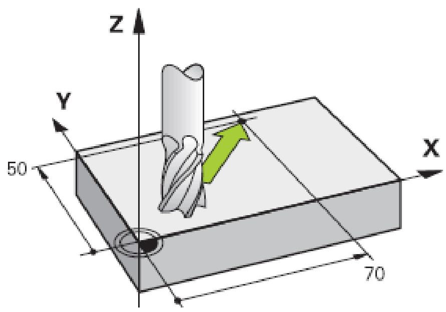
```
50 L  X+100
```

50	Block number
L	Path function **straight line**
X+100	Coordinate of the end point

The tool retains the Y and Z coordinates and moves to the position X=100.

Movement in the main planes

If the NC block contains two coordinates, the control moves the tool in the programmed plane.



Example

```
L X+70 Y+50
```

The tool retains the Z coordinate and moves on the XY plane to the position X=70, Y=50.

Three-dimensional movement

If the NC block contains three coordinates, the control moves the tool spatially to the programmed position.

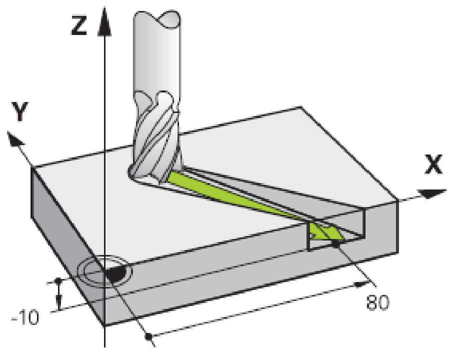
Example

```
L X+80  Y+0 Z-10
```

You can program up to six axes in a straight line block according to the kinematics of your machine.

Example

```
L X+80 Y+0 Z-10 A+15 B+0 C-45
```

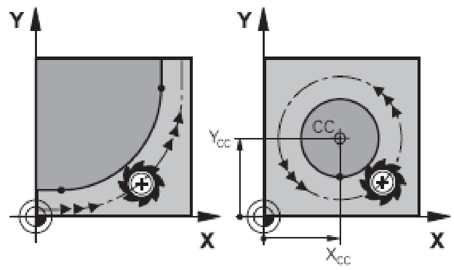


Circles and circular arcs

The control moves two machine axes simultaneously on a circular path relative to the workpiece. You can define a circular movement by entering the circle center **CC** .

When you program a circle, the control assigns it to one of the main planes. This plane is defined automatically when you set the spindle axis during a **TOOL CALL**:

Spindle axis	Main plane
Z	**XY**, also UV, XV, UY
Y	**ZX**, also WU, ZU, WX
X	**YZ**, also VW, YW, VZ



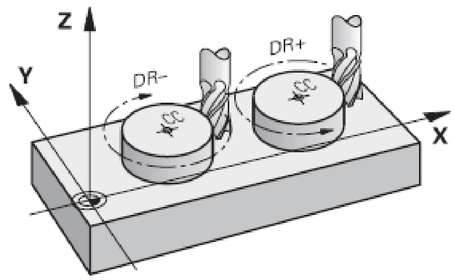
> You can program circles that do not lie parallel to a main plane by using the function for **Tilt working plane** or with Q parameters.
>
> **Further information:** "The PLANE function: Tilting the working plane (option 8)", page 587
>
> **Further information:** "Principle and overview of functions", page 376

Direction of rotation DR for circular movements

When a circular path has no tangential transition to another contour element, enter the direction of rotation as follows:

Clockwise direction of rotation: **DR-**

Counterclockwise direction of rotation: **DR+**



Radius compensation

The radius compensation must be in the block in which you move to the first contour element. You cannot activate radius compensation in a circle block. It must be activated beforehand in a straight-line block.

Further information: "Path contours Cartesian coordinates", page 298

Further information: "Approaching and departing a contour", page 288

Pre-positioning

NOTICE
Danger of collision! The control does not automatically check whether collisions can occur between the tool and the workpiece. Incorrect pre-positioning can also lead to contour damage. There is danger of collision during the approach movement! ▶ Program a suitable pre-position ▶ Check the sequence and contour with the aid of the graphic simulation

Creating the NC blocks with the path function keys

The gray path function keys initiate the dialog. The control asks you successively for all the necessary information and inserts the NC block into the part program.

Example – programming a straight line

- Initiate the programming dialog, e.g. for a straight line

COORDINATES?

X
- Enter the coordinates of the straight-line end point, e.g. -20 in X

COORDINATES?

Y
- Enter the coordinates of the straight-line end point, e.g. 30 in Y, and confirm with the **ENT** key

Radius comp.: RL/RR/no comp.?

R0
- Select the radius compensation (here, press the **R0** soft key—the tool moves without compensation)

Feed rate F=? / F MAX = ENT

ENT
- Enter **100** (feed rate e.g. 100 mm/min; for programming in inches: an input of 100 corresponds to a feed rate of 10 inches/min) and confirm your entry with the **ENT** key, or

F MAX
- Move at rapid traverse: Press the **FMAX** soft key, or

F AUTO
- Traverse with the feed rate defined in the **TOOL CALL** block: Press the **F AUTO** soft key.

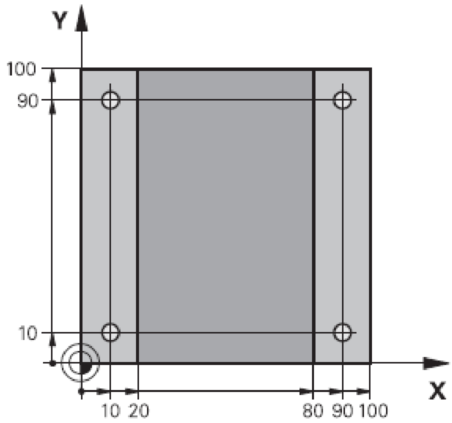
MISCELLANEOUS FUNCTION M?

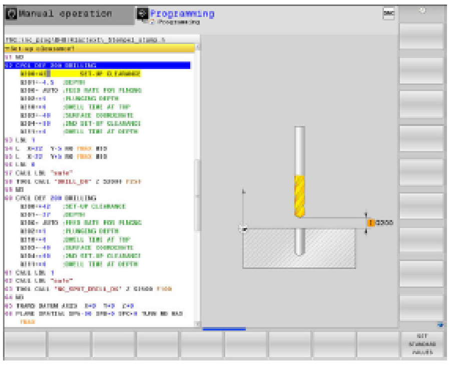
END
- Enter **3** (miscellaneous function e.g. M3) and terminate the dialog with the **END** key

Example

```
L X-20 Y+30 R0 FMAX M3
```

Creating a Cycle Program





The holes (depth of 20 mm) shown in the figure at right are to be drilled with a standard drilling cycle. You have already defined the workpiece blank.

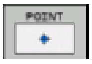
- Call the tool: Enter the tool data. Confirm the entry in each case with the **ENT** key, do not forget the tool axis

- Press the **L** key to open an NC block for a linear movement
- Retract tool: Press the orange axis key **Z** and enter the value for the position to be approached, e.g. 250. Press the **ENT** key
- Confirm **Radius comp.: RL/RR/no comp.?** by pressing the **ENT** key: Do not activate radius compensation
- Confirm **Feed rate F=?** with the **ENT** key: Move at rapid traverse (**FMAX**)
- **Miscellaneous function M?** Confirm with the **END** key
- > The control stores the entered positioning block.

- Call the menu for special functions: Press the **SPEC FCT** key

- Display the functions for point machining

- Select the pattern definition



- Select point entry: Enter the coordinates of the 4 points and confirm each with the **ENT** key. After entering the fourth point, save the block with the **END** key

- Call the cycle menu: Press the **CYCL DEF** key

- Display the drilling cycles

- Select standard drilling cycle 200
- > The control starts the dialog for cycle definition.
- Enter all parameters requested by the control step by step and conclude each entry with the **ENT** key
- > In the screen to the right, the control also displays a graphic showing the respective cycle parameter

- Display the menu for defining the cycle call: Press the **CYCL CALL** key

- Run the drilling cycle on the defined pattern:
- Confirm **Feed rate F=?** with the **ENT** key: Move at rapid traverse (**FMAX**)
- **Miscellaneous function M?** Switch on the spindle and coolant, e.g. **M13**, and confirm with the **END** key

> The control stores the entered positioning block.

- Enter Retract tool: Press the orange axis key **Z** and enter the value for the position to be approached, e.g. 250. Press the **ENT** key
- Confirm **Radius comp.: RL/RR/no comp.?** by pressing the **ENT** key: Do not activate radius compensation
- Confirm **Feed rate F=?** with the **ENT** key: Move at rapid traverse (**FMAX**)
- **Miscellaneous function M?** Enter **M2** to end the program, then confirm with the **END** key

> The control stores the entered positioning block.

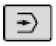
Example

0 BEGIN PGM C200 MM	
1 BLK FORM 0.1 Z X+0 Y+0 Z-40	Workpiece blank definition
2 BLK FORM 0.2 X+100 Y+100 Z+0	
3 TOOL CALL 5 Z S4500	Tool call
4 L Z+250 R0 FMAX	Retract the tool
5 PATTERN DEF POS1 (X+10 Y+10 Z+0) POS2 (X+10 Y+90 Z+0) POS3 (X+90 Y+90 Z+0) POS4 (X+90 Y+10 Z+0)	Define the machining positions
6 CYCL DEF 200 DRILLING	Define the cycle
Q200=2 ;SET-UP CLEARANCE	
Q201=-20 ;DEPTH	
Q206=250 ;FEED RATE FOR PLNGNG	
Q202=5 ;PLUNGING DEPTH	
Q210=0 ;DWELL TIME AT TOP	
Q203=-10 ;SURFACE COORDINATE	
Q204=20 ;2ND SET-UP CLEARANCE	
Q211=0.2 ;DWELL TIME AT DEPTH	
Q395=0 ;DEPTH REFERENCE	
7 CYCL CALL PAT FMAX M13	Spindle and coolant on, call the cycle
8 L Z+250 R0 FMAX M2	Retract the tool, end program
9 END PGM C200 MM	

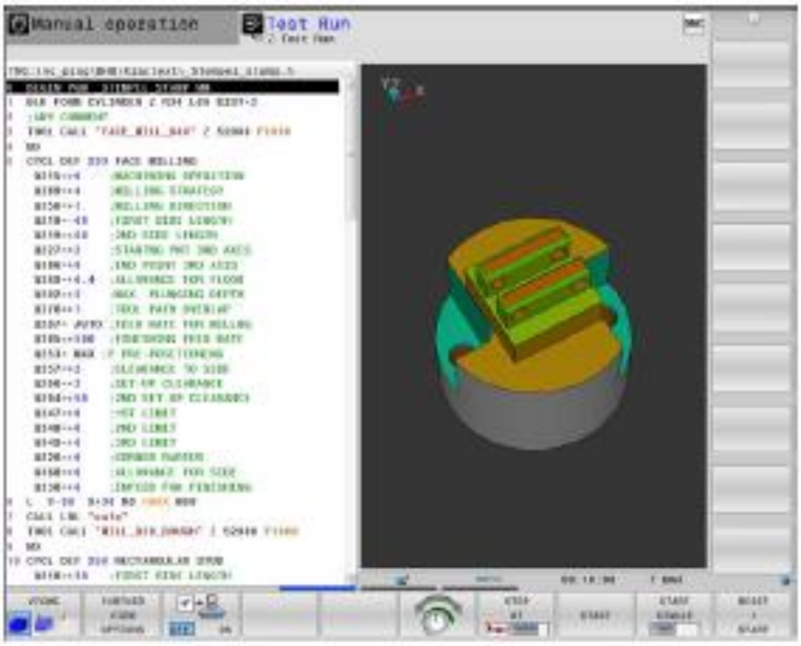
Graphically Testing the First Part

Selecting the correct operating mode

You can test programs in the **Test Run** operating mode:



- ▶ Press the operating mode key
- \> The control switches to the **Test Run** mode of operation.

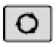


Choosing the program you want to test

- ▶ Press the **PGM MGT** key
- \> The control opens the file manager.

- ▶ Press the **LAST FILES** soft key
- \> The control opens a pop-up window with the most recently selected files.
- ▶ Use the arrow keys to select the program that you want to test. Load with the **ENT** key

Selecting the screen layout and the view



- ▶ Press the key for selecting the screen layout
- > The control displays all available alternatives in the soft-key row.

- ▶ Press the **PROGRAM + GRAPHICS** soft key
- > In the left half of the screen the control shows the program; in the right half it shows the workpiece blank.

The control features the following views:

Soft keys	Function
	Volume view
	Volume view and tool paths
	Tool paths

Starting the test run

- ▶ Press the **RESET + START** soft key
- > The control resets the previously active tool data
- > The control simulates the active program up to a programmed break or to the program end
- ▶ While the simulation is running, you can use the soft keys to change views

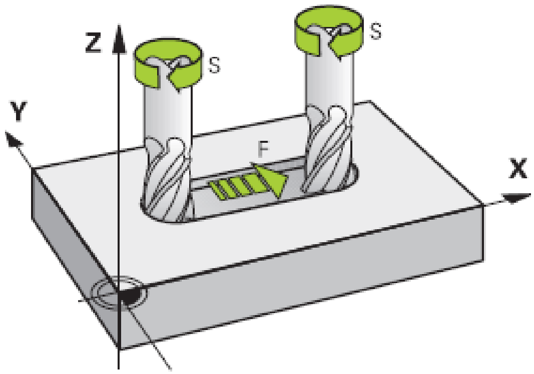
- ▶ Press the **STOP** soft key
- > The control interrupts the test run

- ▶ Press the **START** soft key
- > The control resumes the test run after a break

Entering Tool – Related Data

Feed rate F

The feed rate **F** is the speed at which the tool center point moves. The maximum feed rates can be different for the individual axes and are set in machine parameters.



Input

You can enter the feed rate in the **TOOL CALL** block and in every positioning block.

Further information: "Creating the NC blocks with the path function keys", page 286

You enter the feed rate **F** in mm/min in millimeter programs, and in 1/10 inch/min in inch-programs, for resolution reasons. Alternatively, with the corresponding soft keys, you can also define the feed rate in mm per revolution (mm/1) **FU** or in mm per tooth (mm/tooth) **FZ**.

Rapid traverse

If you wish to program rapid traverse, enter **F MAX.** To enter **FMAX,** press the **ENT** key or the **FMAX** soft key when the dialog question **FEED RATE F = ?** appears on the control's screen.

> To move your machine at rapid traverse, you can also program the corresponding numerical value, e.g. **F30000**. Unlike **FMAX**, this rapid traverse remains in effect not only in the individual block but in all blocks until you program a new feed rate.

Duration of effect

A feed rate entered as a numerical value remains in effect until a block with a different feed rate is reached. **FMAX** is only effective in the block in which it is programmed. After the block with **F MAX** is executed, the feed rate will return to the last feed rate entered as a numerical value.

Changing during program run

You can adjust the feed rate during the program run with the feed rate potentiometer F.

The feed rate potentiometer lowers the programmed feed rate, not the feed rate calculated by the control.

Spindle speed S

The spindle speed S is entered in revolutions per minute (rpm) in a **TOOL CALL** block (tool call). Instead, you can also define the cutting speed Vc in meters per minute (m/min).

Programmed change

In the NC program, you can change the spindle speed in a **TOOL CALL** block by entering the spindle speed only:

TOOL CALL

- Program a tool call: Press the **TOOL CALL** key
- Ignore the dialog question for **Tool number ?** with the **NO ENT** key
- Ignore the dialog question for **Working spindle axis X/Y/Z ?** with the **NO ENT** key
- Enter the new spindle speed for the dialog question **Spindle speed S= ?,** and confirm with **END,** or switch via the **VC** soft key to entry of the cutting speed.

If the number of the already inserted tool is entered in the **TOOL CALL** block without specifying the tool axis, then only the spindle speed will change.

If the tool axis is also entered in the **TOOL CALL** block, the control will insert a replacement tool if a replacement tool was defined.

Changing during program run

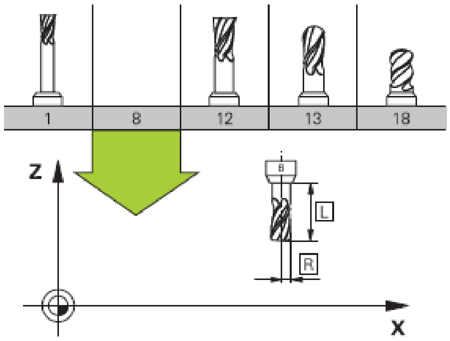
You can adjust the spindle speed during program run with the spindle speed potentiometer S.

Requirements for Tool Compensation

Requirements for tool compensation

You usually program the coordinates of path contours as they are dimensioned in the workpiece drawing. To allow the control to calculate the tool center path (i.e. the tool compensation) you must also enter the length and radius of each tool you are using.

Tool data can be entered either directly in the part program with **TOOL DEF** or separately in a tool table. In a tool table, you can also enter additional data for the specific tool. The control will consider all the data entered for the tool when executing the part program.



Tool number, tool name

Each tool is identified by a number between 0 and 32767. If you are working with tool tables, you can also enter a tool name for each tool. Tool names can have up to 32 characters.

Permitted special characters: # $ % & , - _ . 0 1 2 3 4 5 6 7 8 9 @ A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

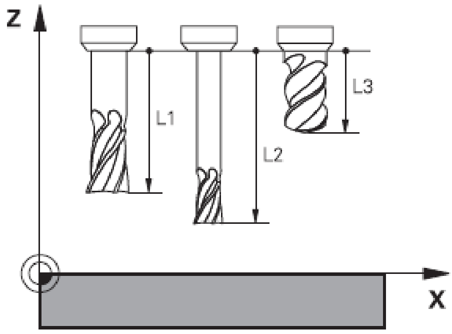
The control automatically replaces lowercase letters with corresponding uppercase letters during saving.

Impermissible characters: <blank space> ! " ' ( ) * + : ; < = > ? [/] ^ ` { | } ~

The tool number 0 is automatically defined as the zero tool with the length L=0 and the radius R=0. In tool tables, tool T0 should also be defined with L=0 and R=0.

Tool length L

You should always enter the tool length L as an absolute value based on the tool reference point. The entire tool length is essential for the control in order to perform numerous functions involving multi-axis machining.



Delta values for lengths and radii

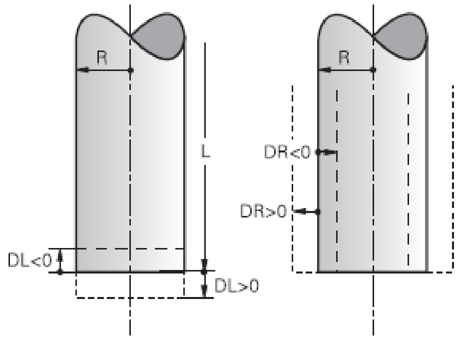
Delta values are offsets in the length and radius of a tool.

A positive delta value describes a tool oversize (**DL**, **DR**>0). If you are programming the machining data with an allowance, enter the oversize value in the **TOOL CALL**.

A negative delta value describes a tool undersize (**DL**, **DR**<0). An undersize is entered in the tool table for wear.

Delta values are usually entered as numerical values. In a **TOOL CALL** block, you can also assign the values to Q parameters.

Input range: You can enter a delta value with up to ± 99.999 mm.



> Delta values from the tool table influence the graphical representation of the clearing simulation.
>
> Delta values from the **TOOL CALL** block do not change the represented size of the **tool** during the simulation. However, the programmed delta values move the **tool** by the defined value in the simulation.

> Delta values from the **TOOL CALL** block influence the position display depending on the optional machine parameter **progToolCallDL** (no. 124501).

Entering tool data into the NC program

> Refer to your machine manual.
>
> The machine tool builder determines the scope of functions of the **TOOL DEF** function.

The number, length and radius of a specific tool is defined in the **TOOL DEF** block of the part program:

- Select the tool definition: Press the **TOOL DEF** key

TOOL DEF

- **Tool number**: Each tool is uniquely identified by its tool number
- **Tool length**: Compensation value for the tool length
- **Tool radius**: Compensation value for the tool radius

> In the programming dialog, you can transfer the value for tool length and tool radius directly into the input line by pressing the desired axis soft key.

Example

```
4 TOOL DEF 5 L+10 R+5
```

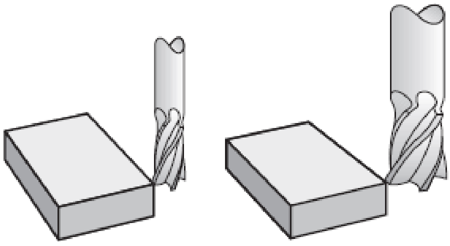
Tool compensation

Introduction

The control adjusts the tool path by the compensation value for the tool length in the spindle axis. In the machining plane, it compensates the tool radius.

If you are writing the part program directly on the control, the tool radius compensation is effective only in the working plane.

The control accounts for the compensation value in up to six axes including the rotary axes.



Tool length compensation

Length compensation becomes effective automatically as soon as a tool is called. To cancel length compensation, call a tool with the length L=0 (e.g. **TOOL CALL 0**).

NOTICE

Danger of collision!

The control uses the defined tool lengths for tool length compensation. Incorrect tool lengths will result in an incorrect tool length compensation. The control does not perform a length compensation and a collision check for tools with a length of **0** and after **TOOL CALL 0**. Danger of collision during subsequent tool positioning movements!

- Always define the actual tool length of a tool (not just the difference)
- Use **TOOL CALL 0** only to empty the spindle

For tool length compensation, the control takes the delta values from both the **TOOL CALL** block and the tool table into account:

Compensation value = **L** + **DL** $_{\text{TOOL CALL}}$ + **DL** $_{\text{TAB}}$ with

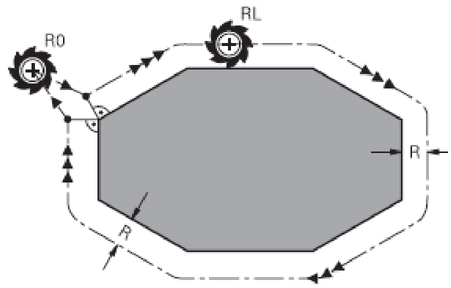
L:	Tool length **L** from **TOOL DEF** block or tool table
DL $_{\text{TOOL CALL}}$:	Oversize for length **DL** in the **TOOL CALL** block
DL $_{\text{TAB}}$:	Oversize for length **DL** in the tool table

Tool radius compensation

The block for programming a tool movement contains:

- **RL** or **RR** for radius compensation
- **R0**, if there is no radius compensation

The radius compensation is effective as soon as a tool is called and traversed with a straight-line block in the working plane with **RL** or **RR**.



> The control automatically cancels radius compensation in the following cases:
> - Straight-line block with **R0**
> - **DEP** function for departing from the contour
> - Selection a new program via **PGM MGT**

For radius compensation, the control takes the delta values from both the **TOOL CALL** block and the tool table into account:

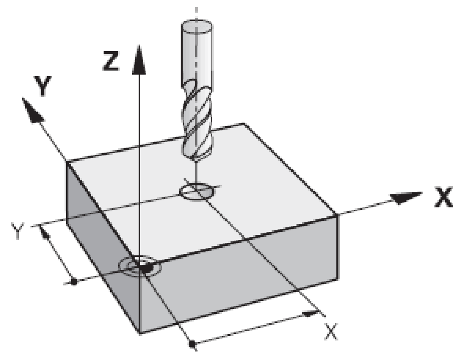
Compensation value = **R** + **DR**$_{\text{TOOL CALL}}$ + **DR**$_{\text{TAB}}$ with

R:	Tool radius **R** from **TOOL DEF** block or tool table
DR $_{\text{TOOL CALL}}$:	Oversize for radius **DR** in the **TOOL CALL** block
DR $_{\text{TAB}}$:	Oversize for radius **DR** in the tool table

Contouring without radius compensation: R0

The tool center moves in the working plane along the programmed path, or to the programmed coordinates.

Applications: Drilling and boring, pre-positioning

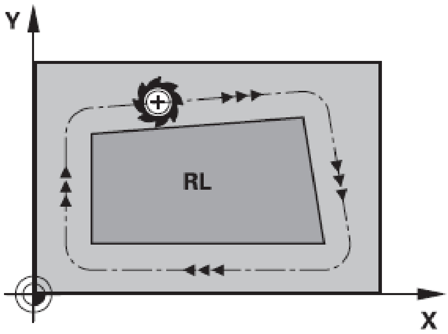


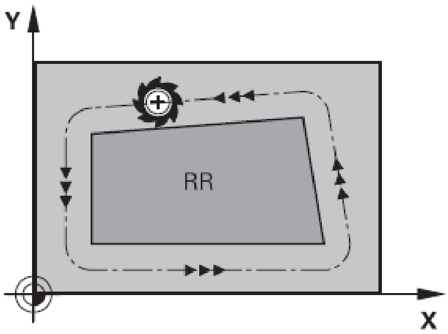
Contouring with radius compensation: RR and RL

RR: The tool moves to the right of the programmed contour

RL: The tool moves to the left of the programmed contour

The tool center moves along the contour at a distance equal to the radius. **Right** or **left** are to be understood as based on the direction of tool movement along the workpiece contour.





> Between two NC blocks with different radius compensations **RR** and **RL** you must program at least one traversing block in the working plane without radius compensation (that is, with **R0**).
>
> The control does not put radius compensation into effect until the end of the block in which it is first programmed.
>
> When radius compensation is activated with **RR/RL** or canceled with **R0** the control always positions the tool perpendicular to the programmed starting or end position. Position the tool at a sufficient distance from the first or last contour point to prevent the possibility of damaging the contour.

Entering radius compensation

Radius compensation is entered in an **L** block. Enter the coordinates of the target point and confirm your entry with the **ENT** key.

Radius comp.: RL/RR/no comp.?

RL
- Select tool movement to the left of the contour: Press the **RL** soft key, or

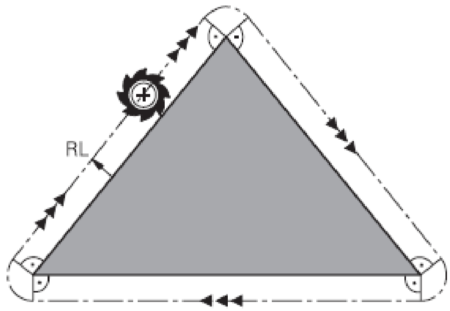
RR
- Select tool movement to the right of the contour: Press the **RR** soft key, or

ENT
- Select tool movement without radius compensation or cancel radius compensation: Press the **ENT** key

END
- Terminate the block: Press the **END** key

Radius compensation: Machining corners

- Outside corners:
  If you program radius compensation, the control moves the tool around outside corners on a transitional arc. If necessary, the control reduces the feed rate at outside corners to reduce machine stress, for example at very great changes of direction
- Inside corners:
  The control calculates the intersection of the tool center paths at inside corners under radius compensation. From this point it then starts the next contour element. This prevents damage to the workpiece at the inside corners. The permissible tool radius, therefore, is limited by the geometry of the programmed contour.

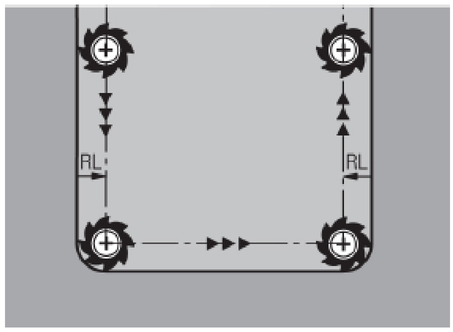


NOTICE

Danger of collision!

The control needs safe positions for contour approach and departure. These positions must enable the control to perform compensating movements when radius compensation is activated and deactivated. Incorrect positions can lead to contour damage. Danger of collision during machining!

- Program safe approach and departure positions at a sufficient distance from the contour
- Consider the tool radius
- Consider the approach strategy



Tool table: Standard tool data

Abbr.	Inputs	Dialog
T	Number by which the tool is called in the program (e.g. 5, indexed: 5.2)	-
NAME	Name by which the tool is called in the program (max. 32 characters, all capitals, no spaces)	**Tool name?**
L	Tool length L	**Tool length?**
R	Tool radius R	**Tool radius?**
R2	Tool radius R2 for toroid cutters (only for 3-D radius compensation or graphical representation of a machining operation with spherical or toroid cutters)	**Tool radius 2?**
DL	Delta value for tool length L	**Tool length oversize?**
DR	Delta value for tool radius R	**Tool radius oversize?**
DR2	Delta value for tool radius R2	**Tool radius oversize 2?**
TL	Set tool lock (**TL** for **T**ool **L**ocked	**Tool locked? Yes=ENT/ No=NOENT**
RT	Number of a replacement tool – if available – as replacement tool (**RT**: for **R**eplacement **T**ool) An empty field or input **0** means no replacement tool has been defined.	**Replacement tool?**
TIME1	Maximum tool life in minutes. This function can vary depending on the individual machine tool. Your machine manual provides more information	**Maximum tool age?**
TIME2	Maximum tool life in minutes during a tool call: If the current tool age reaches or exceeds this value, the control inserts the replacement tool during the next **TOOL CALL** (if the tool axis is specified)	**Max. tool age for TOOL CALL?**
CUR_TIME	Current age of the tool in minutes: The control automatically counts the current tool life (**CUR_TIME**: For **CUR**rent **TIME**) A starting value can be entered for used tools	**Current tool age?**

Available tool types

The tool management displays the various tool types with an icon. The following tool types are available:

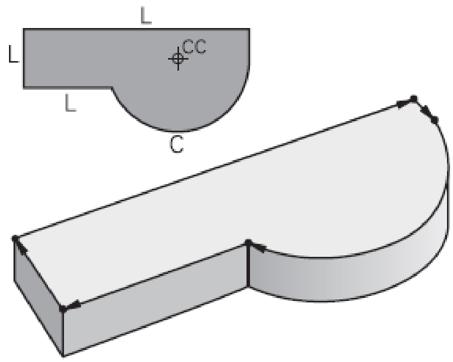
Icon	Tool type	Tool type number
	Undefined,****	99
	Milling cutter,MILL	0
	Drill,DRILL	1
	Tap,TAP	2
	Center drill,CENT	4
	Turning Tool,TURN	29
	Touch probe,TCHP	21
	Ream,REAM	3
	Countersink,CSINK	5
	Piloted counterbore(TSINK),TSINK	6
	Boring tool,BOR	7
	Back boring tool,BCKBOR	8
	Thread mill,GF	15
	Thread mill w/ countersink,GSF	16
	Thread mill w/ single thread,EP	17
	Thread mill w/ indxbl insert,WSP	18
	Thread milling drill,BGF	19
	Circular thread mill,ZBGF	20
	Roughing cutter (MILL_R),MILL_R	9
	Finishing cutter (MILL_F),MILL_F	10
	Rough/finish cutter,MILL_RF	11
	Floor finisher(MILL_FD),MILL_FD	12
	Side finisher (MILL_FS),MILL_FS	13
	Face milling cutter,MILL_FACE	14

Programming Contours

Tool movements

Path functions

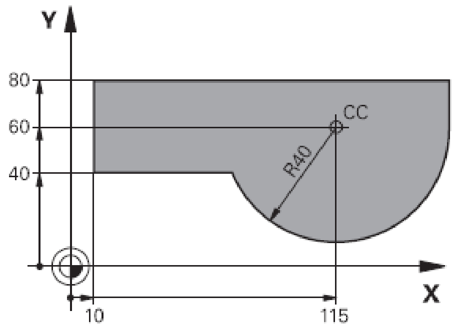
A workpiece contour is usually composed of several contour elements such as straight lines and circular arcs. With the path functions, you can program the tool movements for **straight lines** and **circular arcs**.



FK free contour programming

If a production drawing is not dimensioned for NC and the dimensions given are not sufficient for creating a part program, you can program the workpiece contour with the FK free contour programming. The control calculates the missing data.

With FK programming, you also program tool movements for **straight lines** and **circular arcs**.



Miscellaneous functions M

With the control's miscellaneous functions you can affect

- the program run, e.g., a program interruption
- the machine functions, such as switching spindle rotation and coolant supply on and off
- the path behavior of the tool

Subprograms and program section repeats

If a machining sequence occurs several times in a program, you can save time and reduce the chance of programming errors by entering the sequence once and then defining it as a subprogram or program section repeat. If you wish to execute a specific program section only under certain conditions, you also define this machining sequence as a subprogram. In addition, you can have a part program call a separate program for execution.

Programming with Q parameters

Instead of programming numerical values in a machining program, you enter markers called Q parameters. You assign the values to the Q parameters separately with the Q parameter functions. You can use the Q parameters for programming mathematical functions that control program execution or describe a contour.

In addition, programming with Q parameters enables you to measure with the 3-D touch probe during the program run.

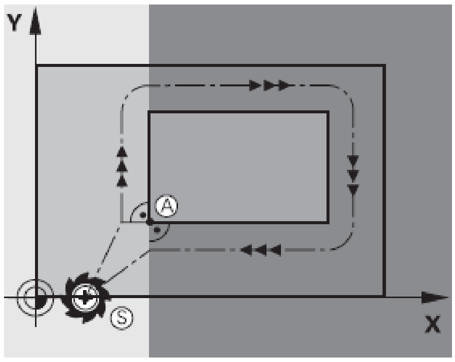
12.6 Approaching and Departing a Contour

Starting point and end point

The tool approaches the first contour point from the starting point. The starting point must be:

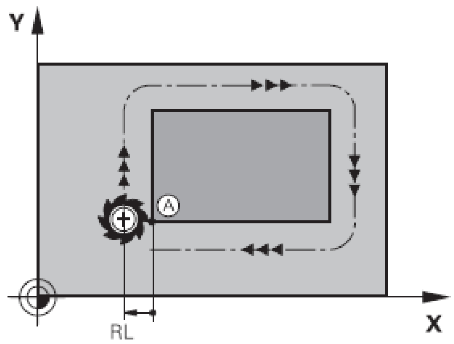
- Programmed without radius compensation
- Approachable without danger of collision
- Close to the first contour point

Example in the figure on the right:
If you set the starting point in the dark gray area, the contour will be damaged when the first contour element is approached.



First contour point

You need to program a radius compensation for the tool movement to the first contour point.

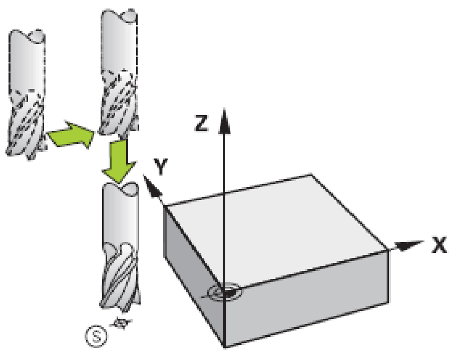


Approaching the starting point in the spindle axis

When the starting point is approached, the tool must be moved to the working depth in the spindle axis. If danger of collision exists, approach the starting point in the spindle axis separately.

Example

30 L Z-10 R0 FMAX	
31 L X+20 Y+30 RL F350	

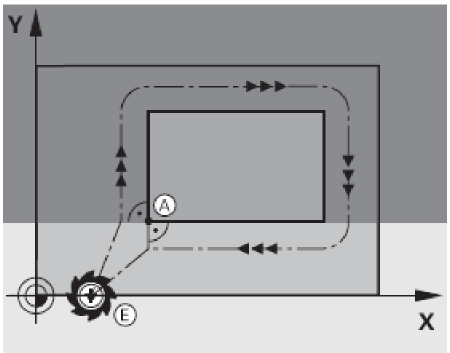


End point

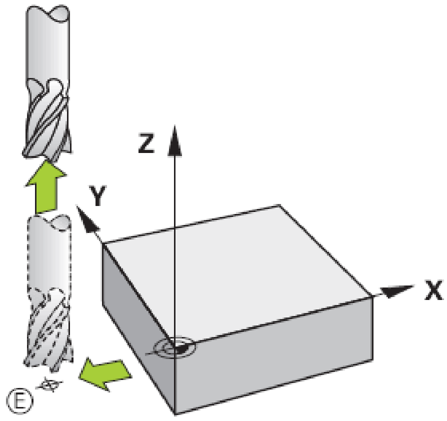
The end point should be selected so that it is:

- Approachable without danger of collision
- Near to the last contour point
- In order to make sure the contour will not be damaged, the optimal ending point should lie on the extended tool path for machining the last contour element

Example in the figure on the right:
If you set the end point in the dark gray area, the contour will be damaged when the end point is approached.



Departing the end point in the spindle axis:
Program the departure from the end point in the spindle axis separately.



Example

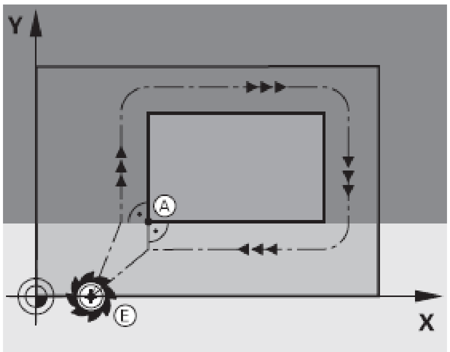
50 L X+60 Y+70 R0 F700	
51 L Z+250 R0 FMAX	

Common starting and end points

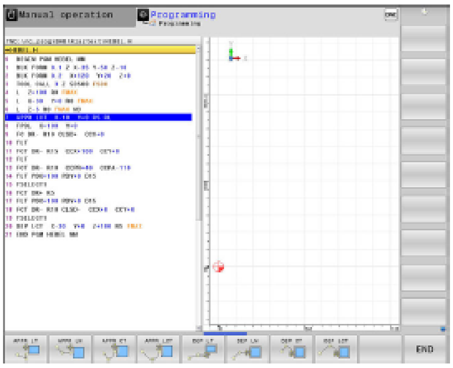
Do not program any radius compensation if the starting point and end point are the same.

In order to make sure the contour will not be damaged, the optimal starting point should lie between the extended tool paths for machining the first and last contour elements.

Example in the figure on the right:
If you set the end point in the dark gray area, the contour will be damaged when the contour is approached/departed.



Overview: Types of paths for contour approach and departure



The functions for contour approach **APPR** and departure **DEP** are activated with the **APPR/DEP** key. You can then select the following path forms with the corresponding soft keys:

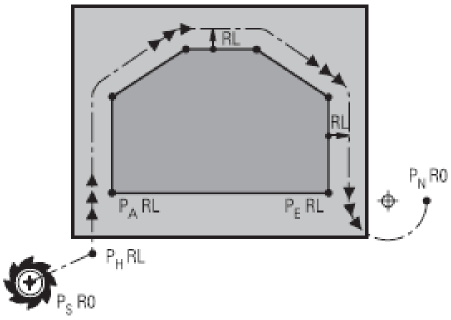
Approach	Departure	Function
		Straight line with tangential connection
		Straight line perpendicular to a contour point
		Circular arc with tangential connection
		Circular arc with tangential connection to the contour. Approach and departure to an auxiliary point outside the contour on a tangentially connecting line

Approaching and departing a helix

The tool approaches and departs a helix on its extension by moving in a circular arc that connects tangentially to the contour. You program helical approach and departure with the **APPR CT** and **DEP CT** functions.

Important positions for approach and departure

- Starting point P_S
  You program this position in the block before the APPR block. P_S lies outside the contour and is approached without radius compensation (R0).
- Auxiliary point P_H
  Some of the paths for approach and departure go through an auxiliary point P_H that the control calculates from your input in the APPR or DEP block. The control moves from the current position to the auxiliary point P_H at the feed rate last programmed. If you have programmed **FMAX** (positioning at rapid traverse) in the last positioning block before the approach function, the control also approaches the auxiliary point P_H at rapid traverse
- First contour point P_A and last contour point P_E
  You program the first contour point P_A in the APPR block. The last contour point P_E can be programmed with any path function. If the APPR block also includes the Z coordinate, the control moves the tool simultaneously to the first contour point P_A.
- End point P_N
  The position P_N lies outside of the contour and results from your input in the DEP block. If the DEP block also includes the Z coordinate, the control moves the tool simultaneously to the end point P_N.



Abbreviation	Meaning
APPR	Approach
DEP	Departure
L	Line
C	Circle
T	Tangential (smooth connection)
N	Normal (perpendicular)

NOTICE

Danger of collision!

The control does not automatically check whether collisions can occur between the tool and the workpiece. Incorrect pre-positioning and incorrect auxiliary points P_H can also lead to contour damage. There is danger of collision during the approach movement!

- ▶ Program a suitable pre-position
- ▶ Check the auxiliary point P_H, the sequence and the contour with the aid of the graphic simulation

With the **APPR LT**, **APPR LN** and **APPR CT** functions, the control moves the tool to the auxiliary point P_H at the last programmed feed rate (which can also be **FMAX**). With the **APPR LCT** function, the control moves to the auxiliary point P_H at the feed rate programmed with the APPR block. If no feed rate is programmed yet before the approach block, the control generates an error message.

Polar coordinates

You can also program the contour points for the following approach/ departure functions over polar coordinates:

- APPR LT becomes APPR PLT
- APPR LN becomes APPR PLN
- APPR CT becomes APPR PCT
- APPR LCT becomes APPR PLCT
- DEP LCT becomes DEP PLCT

Select an approach or departure function with the soft key, then press the orange **P** key.

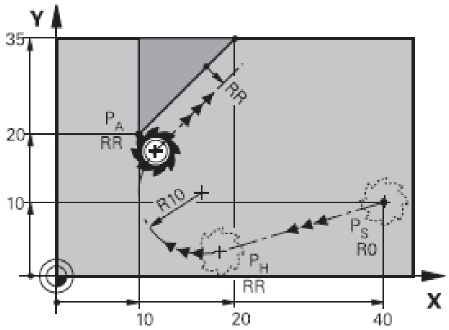
Radius compensation

The tool radius compensation is programmed together with the first contour point P_A in the APPR block. The DEP blocks automatically discard the tool radius compensation.

If you program **APPR LN** or **APPR CT** with **R0**, the control stops the machining/simulation with an error message.

This method of function differs from the iTNC 530 control!

Approaching on a circular path with tangential connection from a straight line to the contour: APPR LCT



The tool moves on a straight line from the starting point P_S to an auxiliary point P_H. It then moves to the first contour point P_A on a circular arc. The feed rate programmed in the APPR block is effective for the entire path that the control traversed in the approach block (path P_S to P_A).

If you have programmed the coordinates of all three principal axes X, Y and Z in the approach block, the control moves the tool from the position defined before the APPR block to the auxiliary point P_H on all three axes simultaneously. Then the connect goes from P_H to P_A only on the working plane.

The arc is connected tangentially both to the line P_S - P_H as well as to the first contour element. Once these lines are known, the radius then suffices to completely define the tool path.

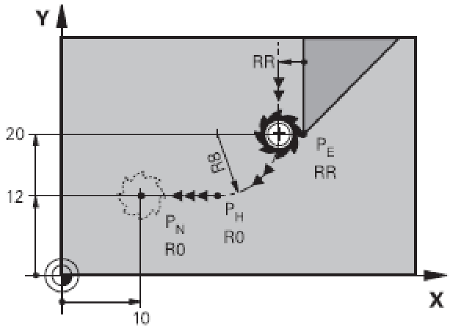
- Use any path function to approach the starting point P_S.
- Initiate the dialog with the **APPR DEP** key and **APPR LCT** soft key:

- Coordinates of the first contour point P_A
- Radius R of the circular arc. Enter R as a positive value
- Radius compensation **RR/RL** for machining

Example

7 L X+40 Y+10 R0 FMAX M3	Approach PS without radius compensation
8 APPR LCT X+10 Y+20 Z-10 R10 RR F100	PA with radius compensation RR, radius R=10
9 L X+20 Y+35	End point of the first contour element
10 L ...	Next contour element

Departing on a circular arc tangentially connecting the contour and a straight line: DEP LCT



The tool moves on a circular arc from the last contour point P_S to an auxiliary point P_H. It then moves on a straight line to the end point P_N. The arc is tangentially connected both to the last contour element and to the line from P_H to P_N. Once these lines are known, the radius R suffices to unambiguously define the tool path.

- Program the last contour element with the end point P_E and radius compensation
- Initiate the dialog with the **APPR/DEP** key and **DEP LCT** soft key

- Enter the coordinates of the end point P_N
- Radius R of the circular arc. Enter R as a positive value

Example

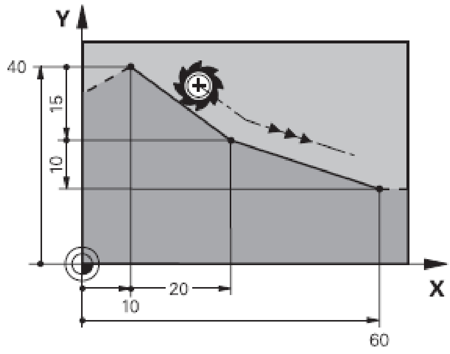
23 L Y+20 RR F100	Last contour element: PE with radius compensation
24 DEP LCT X+10 Y+12 R+8 F100	Coordinates PN, arc radius=8 mm
25 L Z+100 FMAX M2	Retract in Z, return to block 1, end program

12.7 Overview of Part Functions

Path function key	Function	Tool movement	Required input
L	Straight line **L**	Straight line	Coordinates of the end point of the straight line
CHF	Chamfer: **CHF**	Chamfer between two straight lines	Chamfer side length
CC	Circle center **CC**	None	Coordinates of the circle center or pole
C	Circular arc **C**	Circular arc around a circle center CC to an arc end point	Coordinates of the arc end point, direction of rotation
CR	Circular arc **CR**	Circular arc with a certain radius	Coordinates of the arc end point, arc radius, direction of rotation
CT	Circular arc **CT**	Circular arc with tangential connection to the preceding and subsequent contour elements	Coordinates of the arc end point
RND	Corner rounding **RND**	Circular arc with tangential connection to the preceding and subsequent contour elements	Rounding radius R
FK	**FK** free contour programming	Straight line or circular path with any connection to the preceding contour element	"Path contours – FK free contour programming ", page 317

Straight line L

The control moves the tool in a straight line from its current position to the straight-line end point. The starting point is the end point of the preceding block.



- Press the **L** key to open a program block for a linear movement
- **Coordinates** of the end point of the straight line, if necessary
- **Radius compensation RL/RR/R0**
- **Feed rate F**
- **Miscellaneous function M**

Example

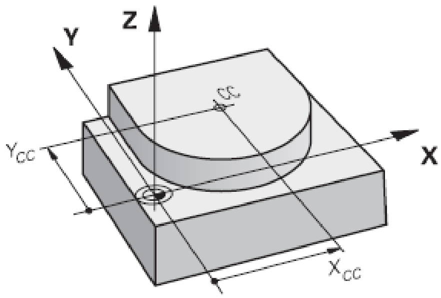
7 L X+10 Y+40 RL F200 M3
8 L IX+20 IY-15
9 L X+60 IY-10

Actual position capture

You can also generate a straight-line block (**L** block) by using the **actual position capture** key:

- In the Manual Operation mode, move the tool to the position you want to capture
- Switch the screen display to programming.
- Select the NC block after which you want to insert the straight line block
- Press the **actual position capture** key
- > The control generates a straight-line block with the actual position coordinates.

Circle center CC



You can define a circle center for circles that you have programmed with the C key (circular path C) . This is done in the following ways:

- Entering the Cartesian coordinates of the circle center in the working plane, or
- Using the circle center defined in an earlier block, or
- Capturing the coordinates with the **Actual-position capture** key

CC ✦ ▶ Enter coordinates for the circle center or, if you want to use the last programmed position, enter no coordinates

Example

```
5 CC  X+25 Y+25
```

or

```
10 L X+25 Y+25
11 CC
```

The program lines 10 and 11 do not refer to the illustration.

Validity

The circle center definition remains in effect until a new circle center is programmed.

Entering the circle center incrementally

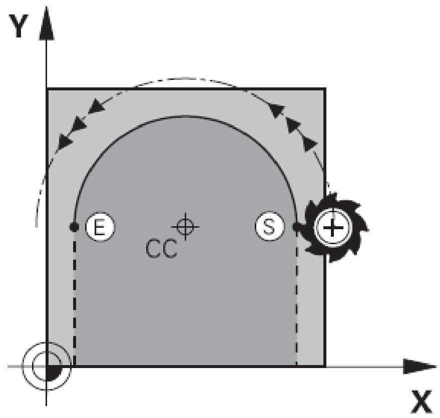
If you enter the circle center with incremental coordinates, you have programmed it relative to the last programmed position of the tool.

> The only effect of **CC** is to define a position as circle center: The tool does not move to this position.
>
> The circle center is also the pole for polar coordinates.

Circular path C around circle center CC

Before programming a circular arc, you must first enter the circle center **CC**. The last programmed tool position will be the starting point of the arc.

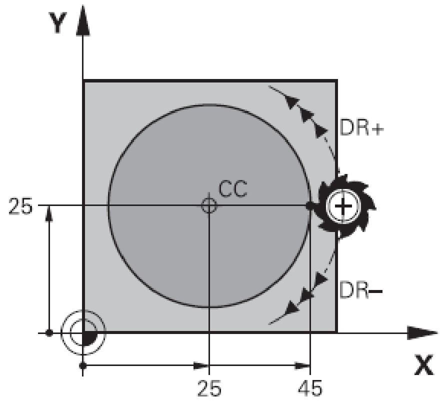
- Move the tool to the circle starting point
  - Enter the **coordinates** of the circle center
  - Enter the **coordinates** of the arc end point, and if necessary:
  - **Direction of rotation DR**
  - **Feed F**
  - **Miscellaneous function M**



> The control normally makes circular movements in the active working plane. However, you can also program circular arcs that do not lie in the active working plane. By simultaneously rotating these circular movements you can create spatial arcs (arcs in three axes), e.g. **C Z... X... DR+** (with tool axis Z).

Example

5 CC X+25 Y+25
6 L X+45 Y+25 RR F200 M3
7 C X+45 Y+25 DR+



Full circle

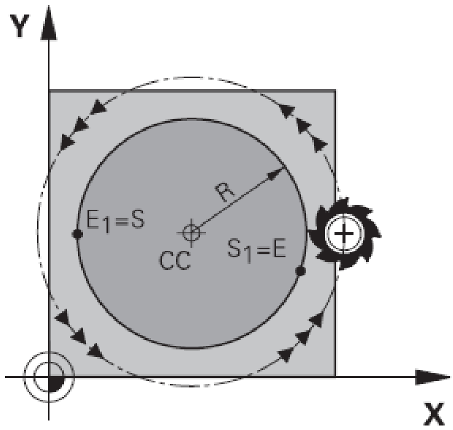
For the end point, enter the same point that you used for the starting point.

> The starting and end points of the arc must lie on the circle.
>
> The maximum value for input tolerance is 0.016 mm. Set the input tolerance in the machine parameter **circleDeviation** (no. 200901).
>
> Smallest possible circle that the control can traverse: 0.016 mm.

Circle CR with defined radius

The tool moves on a circular path with the radius R.

- **Coordinates** of the arc end point
- **Radius R** (the algebraic sign determines the size of the arc)
- **Direction of rotation DR** Note: The algebraic sign determines whether the arc is concave or convex.
- **Miscellaneous function M**
- **Feed F**



Full circle

For a full circle, program two blocks in succession:

The end point of the first semicircle is the starting point of the second. The end point of the second semicircle is the starting point of the first.

Central angle CCA and arc radius R

The starting and end points on the contour can be connected with four arcs of the same radius:

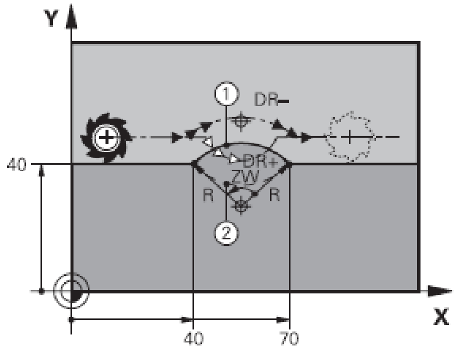
Smaller arc: CCA<180°
Enter the radius with a positive sign R>0

Larger arc: CCA>180°
Enter the radius with a negative sign R<0

The direction of rotation determines whether the arc is curving outward (convex) or curving inward (concave):

Convex: Direction of rotation **DR-** (with radius compensation **RL**)

Concave: Direction of rotation **DR+** (with radius compensation **RL**)



> The distance from the starting and end points of the arc diameter cannot be greater than the diameter of the arc.
>
> The maximum radius is 99.9999 m.
>
> You can also enter rotary axes A, B and C.
>
> The control normally makes circular movements in the active working plane. However, you can also program circular arcs that do not lie in the active working plane. By simultaneously rotating these circular movements you can create spatial arcs (arcs in three axes).

Example

```
10 L X+40 Y+40 RL F200 M3
11 CR  X+70 Y+40  R+20 DR-  (arc 1)
```

or

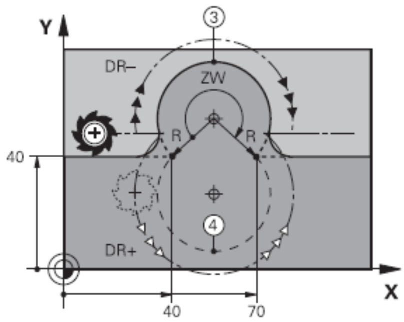
```
11 CR  X+70 Y+40  R+20 DR+  (arc 2)
```

or

```
11 CR X+70 Y+40 R-20 DR- (arc 3)
```

or

```
11 CR X+70 Y+40 R-20 DR+ (arc 4)
```



12.8 Subprograms and Program Section Repeats

Labeling subprograms and program section repeats

Subprograms and program section repeats enable you to program a machining sequence once and then run it as often as necessary.

Label

The beginnings of subprograms and program section repeats are marked in a part program by labels **(LBL)**.

A LABEL is identified by a number between 1 and 65535 or by a name you define. Each LABEL number or LABEL name can be set only once in the program with the **LABEL SET** key. The number of label names you can enter is only limited by the internal memory.

> Do not use a label number or label name more than once!

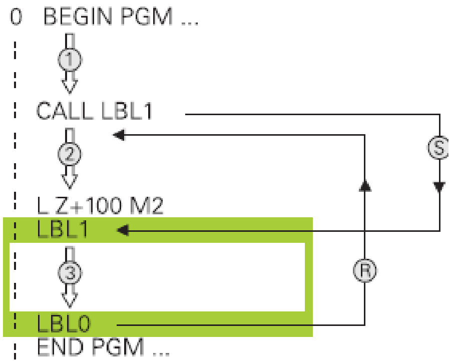
Label 0 (**LBL 0**) is used exclusively to mark the end of a subprogram and can therefore be used as often as desired.

Operating sequence

1 The control executes the part program up to the **CALL LBL** command for calling a subprogram

2 The subprogram is then executed until the subprogram end **LBL 0**

3 The control then resumes the part program from the block after the subprogram call **CALL LBL**



Programming notes

- A main program can contain any number of subprograms
- You can call subprograms in any sequence and as often as desired
- A subprogram cannot call itself
- Write subprograms after the block with M2 or M30
- If subprograms are located before the block with M2 or M30 in the part program, they will be executed at least once even if they are not called

Programming the subprogram

LBL SET

- To mark the beginning: Press the **LBL SET** key
- Enter the subprogram number. If you want to use a label name, press the **LBL NAME** soft key to switch to text entry.
- Enter the text
- Mark the end: Press the **LBL SET** key and enter the label number **0**

Calling a subprogram

LBL CALL

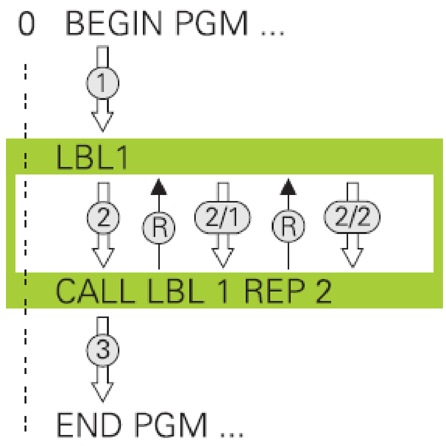
- Call a subprogram: Press the **LBL CALL** key
- Enter the subprogram number of the subprogram you wish to call. If you want to use a label name, press the **LBL NAME** soft key to switch to text entry.
- If you want to enter the number of a string parameter as target address, press the QS soft key
- > The control then jumps to the label name that is specified in the string parameter defined.
- Ignore repeats **REP** by pressing the **NO ENT** key. Repeat **REP** is used only for program section repeats

> **CALL LBL 0** is not permitted (Label 0 is only used to mark the end of a subprogram).

Program-section repeats

Label

The beginning of a program section repeat is marked by the label **LBL**. The end of a program section repeat is identified by **CALL LBL n REPn**.



Operating sequence

1 The control executes the part program up to the end of the program section (**CALL LBL n REPn**)
2 Then the program section between the called LABEL and the label call **CALL LBL n REPn** is repeated the number of times entered after **REP**
3 The control then continues with the part program

Programming notes

- You can repeat a program section up to 65 534 times in succession
- The total number of times the program section is executed is always one more than the programmed number of repeats, because the first repeat starts after the first machining process.

Programming a program section repeat

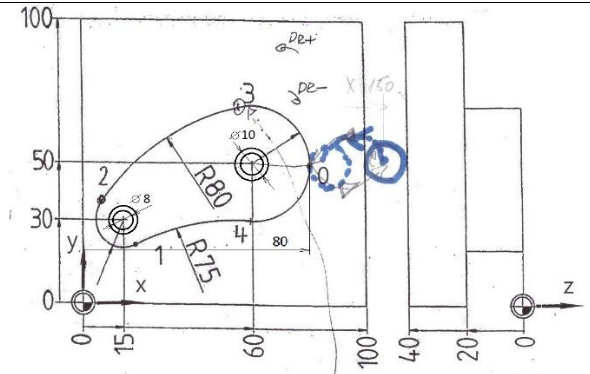
- To mark the beginning, press the **LBL SET** key and enter a LABEL NUMBER for the program section you wish to repeat. If you want to use a label name, press the **LBL NAME** soft key to switch to text entry.
- Enter the program section

Calling a program section repeat

- Call a program section: Press the **LBL CALL** key
- Enter the program section number of the program section to be repeated. If you want to use a LABEL name, press the **LBL NAME** soft key to switch to text entry
- Enter the number of repeats **REP** and confirm with the **ENT** key.

12.9 An example of manufacturing a part

The task is to mill arcs, drill a center hole with an outer diameter of 10 mm and drill a hole with a diameter of 8 mm.
We will also present the program with screen outputs.



	X	Y
1	19,732	21,191
4	58,995	30,025

	X	Y
3	55,505	69,488
2	6,645	35,495

```
0  BEGIN PGM EXCERCISE2 MM
1  BLK FORM 0.1 Z X+0 Y+0 Z-40
2  BLK FORM 0.2  X+100  Y+100  Z+0
3  TOOL CALL 51 Z S2000 F800
4  L  Z+100 R0 FMAX M3
5  L  X+150  Y+50 R0 FMAX
6     Z+0 R0 FMAX
7  LBL 1
8  L IZ-5 R0 F1000
9  APPR LCT  X+80  Y+50 R10 RL F AUTO M8
10 CC  X+60  Y+50
11 C  X+58.995  Y+30.025 DR-
12 CR  X+19.732  Y+21.191 R+75 DR+
13 CC  X+15  Y+30
14 C  X+6.645  Y+35.495 DR-
15 CR  X+55.505  Y+69.488 R+80 DR-
16 CC  X+60  Y+50
17 C  X+80  Y+50 DR- F AUTO
18 DEP LCT  X+150  Y+50 R10 F AUTO
19 CALL LBL 1 REP3
20 L  Z+100 R0 FMAX M5 M9
21 LBL 0
22 TOOL CALL 206 Z S3024 F300
23 CYCL DEF 240 CENTERING ~
    Q200=+2    ;SET-UP CLEARANCE ~
    Q343=+1    ;SELECT DIA./DEPTH ~
    Q201=-2    ;DEPTH ~
    Q344=-10   ;DIAMETER ~
    Q206= FU+0.1  ;FEED RATE FOR PLNGNG ~
    Q211=+3    ;DWELL TIME AT DEPTH ~
    Q203=+0    ;SURFACE COORDINATE ~
    Q204=+50   ;2ND SET-UP CLEARANCE
24 CALL LBL 2
25 TOOL CALL 230 Z S7000 FU0.1
26 CYCL DEF 200 DRILLING ~
    Q200=+2    ;SET-UP CLEARANCE ~
    Q201=-45   ;DEPTH ~
    Q206= FU+0.1  ;FEED RATE FOR PLNGNG ~
    Q202=+5    ;PLUNGING DEPTH ~
    Q210=+1    ;DWELL TIME AT TOP ~
    Q203=+0    ;SURFACE COORDINATE ~
    Q204=+50   ;2ND SET-UP CLEARANCE ~
    Q211=+2    ;DWELL TIME AT DEPTH ~
    Q395=+1    ;DEPTH REFERENCE
27 LBL 2
28 L  X+15  Y+30 R0 FMAX
29 CYCL CALL M3 M8
30 L  X+60  Y+50 R0 FMAX M99
31 M5 M9
32 LBL 0
33 L  Z+100 R0 FMAX M5 M9
34 L  X+250 R0 FMAX
35 M2
36 END PGM EXCERCISE2 MM
```

Fig. 12.2 Drawing of the component and part program for machining

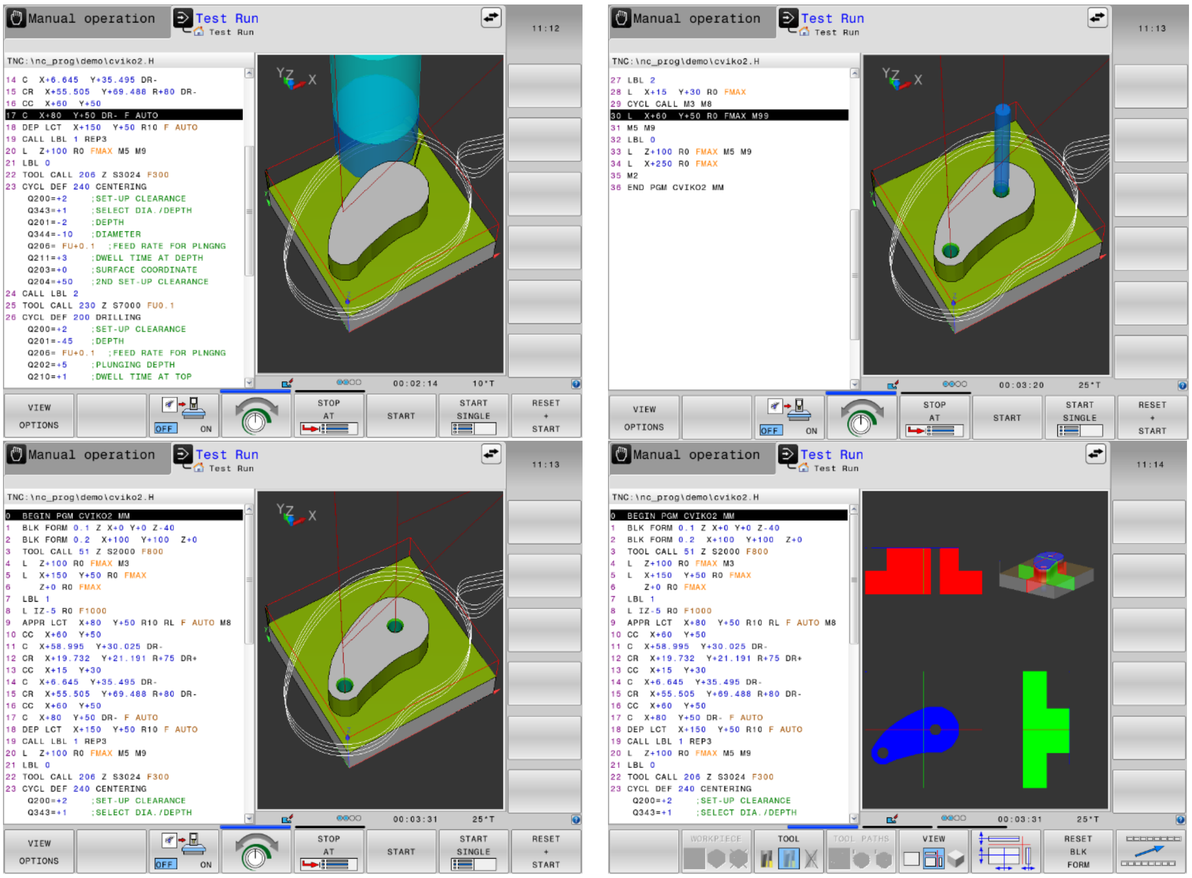


Fig. 12.3 Simulation of the program and inspection of the part in different views

13

Robotics in manufacturing process

13.1 Introduction

Word robot was coined by a Czech novelist Karel Capek in a 1920 play titled Rassum's Universal Robots (RUR)

- Robot in Czech is a word for worker or servant

Definition of robot:

- Any machine made by one our members: Robot Institute of America
- A robot is a reprogrammable, multifunctional manipulator designed to move material, parts, tools, or specialized devices through variable programmed motions for the performance of a variety of tasks: Robot Institute of America, 1979

In literature, the term industrial robot means equipment which could solve independently a variety of handling tasks. At present, although an industrial robot is defined by the ISO, there is several other definitions and different interpretations, but they all have the same essence. "Industrial Robot" is officially defined by ISO 8373:1994 as:

"Automatically controlled, programmable, multipurpose manipulator for action in three or more axes" Biggs and MacDonald [2003] classified programming of industrial robots in two categories: manual- or automatic programming. In manual programming a text- or graphical interface control the robot. "Automatic programming" means to automatically create the program that controls the motion executed by the robot, thus, the user affects the robot's behavior instead of the program. The most common way to program an industrial robot is to guide the robot manually using a teach pendant (an advanced joystick) or a similar device. It is also possible to move the manipulator manually by hand to record joint positions and trajectories. This approach belongs to manual programming since the programmer typically needs to manually (e.g., using a text editor) modify the final trajectory, which allows the programmer to have extensive control over the task.

13.2 Industrial robot structure

The structure of an industrial robot can be divided into mechanical, control and programming section, Fig. 13.1.

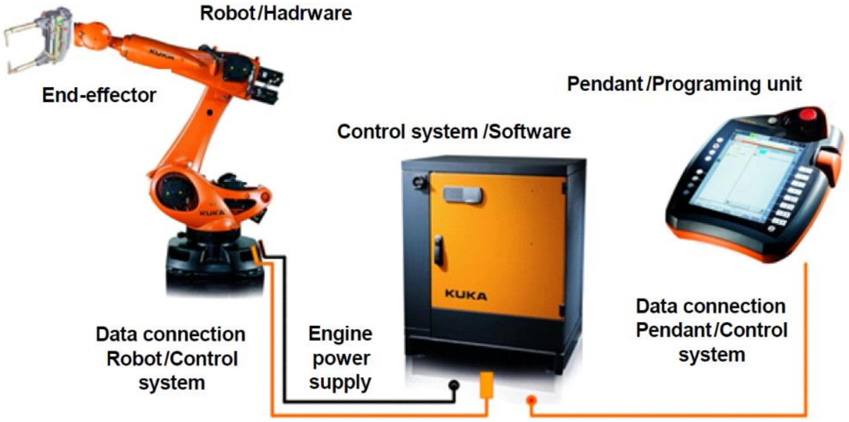


Fig. 13.1 The structure of the industrial robot

Classification of robots

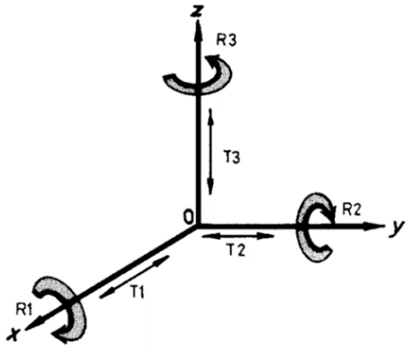
Robot classification may be considered on the following basis:

- Structural configuration and robot motion,
- Trajectories based on motion control,
- Performance characteristics of the robots.

Degree of freedom

The number of independent movements that an object performs in a three-dimensional space is called the number of degrees of freedom (DF). A rigid body when moving freely in space has six degrees of freedom, three for location and three for orientation. These two kinds of independent movement, shown in figure, are

1. the three translations T1,T2 and T3, which represent the linear motions along the x,y and z axes respectively,
2. the three rotations R1,R2 and R3, which represent the angular motions about the x,y and z axes respectively.



By using three orthogonal translations and three rotations about the orthogonal axes, the state of an object, i.e., its location and orientation anywhere in the workspace of the robot, can be completely defined.

Kinematic pairs (KP)

KPs are defined as two members of action mechanism movably connected. Mobility of one of the members to another is limited, the pair usually has one degree of freedom (pairs with more degrees of freedom in the design of robots are not often used). The members, which are connected by sliding and rotating dynamic duos, it is possible to construct any kinematic chains.

The essence of a rotary kinematic pairs (KP) (Fig. 13.2) determines the potential curve only as a circular arc centered on the rotary body of KP with the A and the B bodies with a radius given by the length of arms of these bodies from the axis of rotary KD.

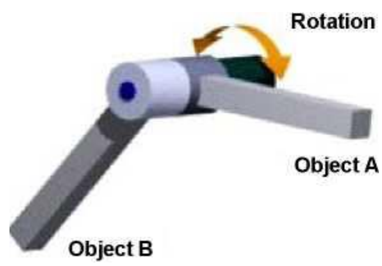


Fig. 13.2 Rotary kinematic pair

The essence of sliding kinematic pairs (KP) (Fig. 13.3) defines itself on the linear trajectory or a section between the solid body A and a movable body B.

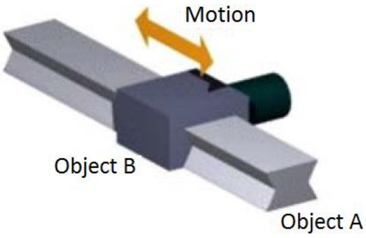


Fig. 13.3 Translational kinematic pair

13.3 Classification of kinematic structures according to structural layout

Kinematic pairs layout:

A) Creating of kinematic chain with serial (consecutive) involving kinematic pairs – robot in Fig. 13.4

a) based on the serial mechanism principle

B) The establishment of a parallel kinematic chain (adjacent) involving kinematic pairs – robot in Fig. 13.4.

b) the principle of parallel mechanism

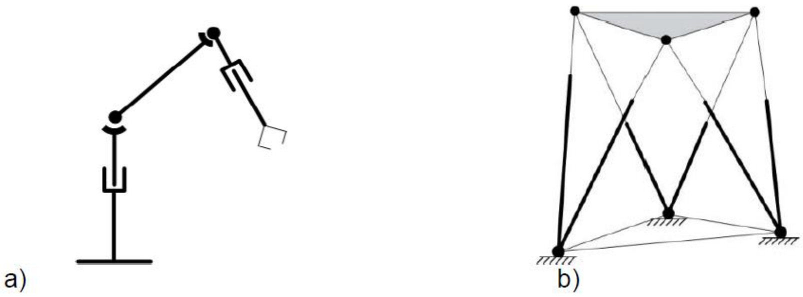
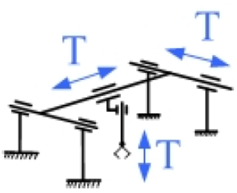
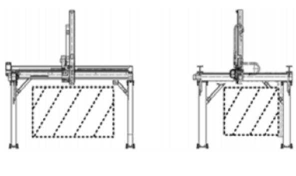


Fig. 13.4 Types of kinematic chains

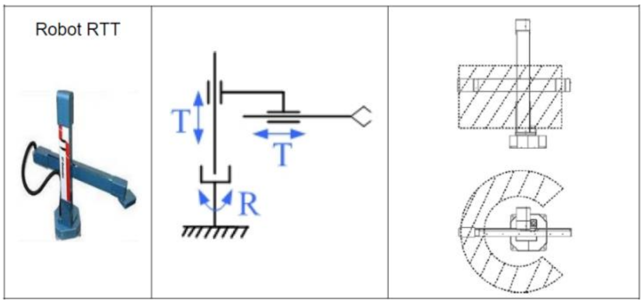
The design of the robot is given by the kinematic structure, which is the type and sequence of arrangement of kinematic pairs in the kinematic chain. The most widespread concepts of an open kinematic chain are those which contains rotational and translational kinematic pairs. Based on the structure of a serial kinematic chain of the main drive system, existing robots can be classified into four basic groups:

Kinematic structure – TTT – Cartesian coordinate system/robot/gantry Cartesian, Gantry robot

Principle	Scheme of kinematic structure	Working area
Robot TTT 		

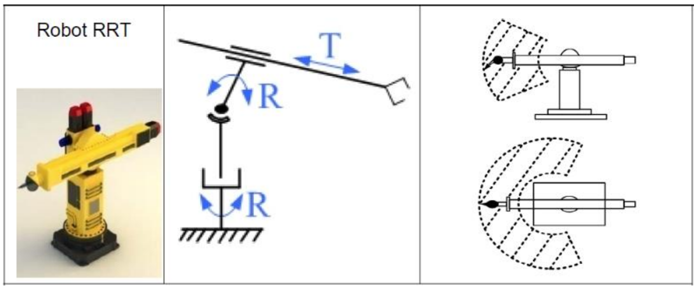
A kinematic chain composed of three, mutually perpendicular (orthogonal) translational kinematic pairs (sliding motion unit). It uses a rectangular coordinate system. This kinematic structure is very stable and in terms of kinematic analysis, it is the most accurate kinematic structure. It is easy to control it. The disadvantage is lower spatial mobility. It is mainly used for large spaces handling areas. Working space of robot consists of a cubic body, namely the prism or cube.

Kinematic structure – RTT – cylindrical coordinate system/robot/Cylindric



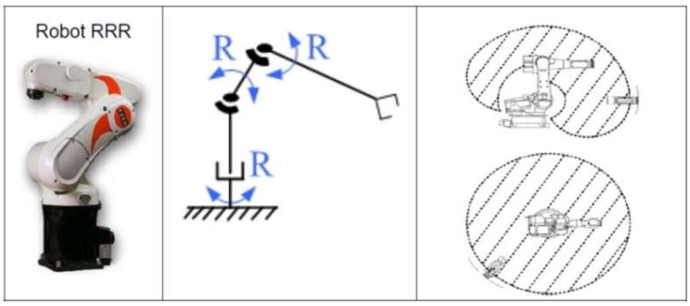
A kinematic chain composed of one rotational kinematic pair (rotational motion unit) and two mutually perpendicular, to each other, translational kinematic pairs (sliding motion unit). It is characterized by its robustness and easy control. Working space of robot consists of a cylindrical body, namely the cylinder, or a part of it.

Kinematic structure – RRT – spherical coordinate system/robot/Spherical



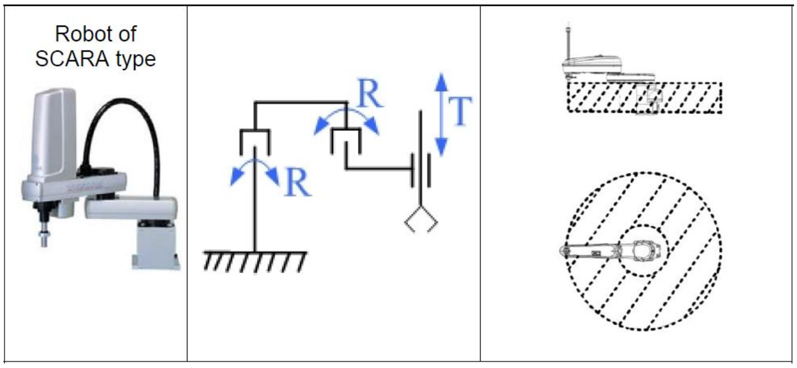
A kinematic chain composed of two rotational kinematic pairs (rotary motion unit) and one translational kinematic pairs (sliding motion unit). The kinematic structure has been proposed as one of the first configuration. The workspace is bounded by spherical surface.

Kinematic structure – RRR – angular coordinate system/robot/Angular



A kinematic chain composed of three rotational kinematic pairs (rotary motion unit). Kinematic structure is characterized by good manipulative skills and thereby it avoids obstacles well. This structure is recently the most common in the construction of robots. Working space of robot consists angular or multiangle body.

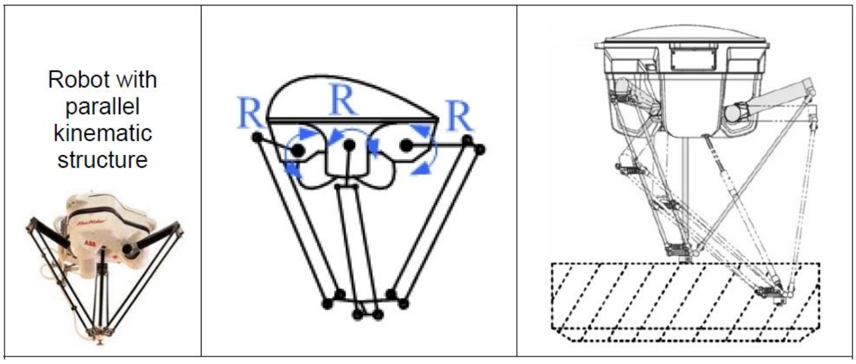
Kinematic structure of SCARA type



A kinematic chain composed of two rotational kinematic pairs (rotary motion unit) and one translational kinematic pair (sliding motion unit). The advantage of this type of kinematic structure is well-positioned service area and higher mobility. It has, however, a smaller working space and more complex control. This structure is destined for the operations performed vertically from above and it is applied for the printed circuit assembly. It reaches high velocity and high acceleration. Working space of robot consists of a ring.

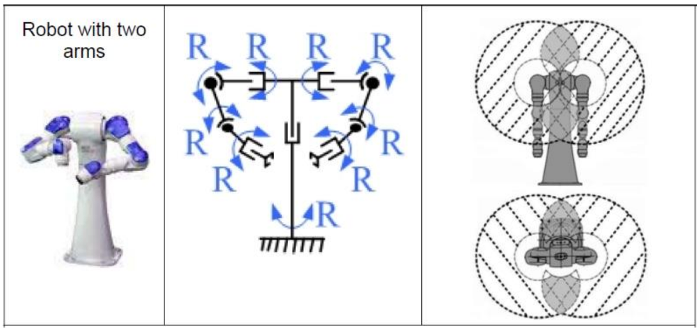
Parallel kinematic structure

Mechanisms in a parallel kinematic structure (Hexapod, 3-pod), have three to six parallel members (arms) that are connected between the base and platform, or output member. Parallel mechanisms generally comprise two platforms, one of which is controlled by a length variable arms, working in parallel. Actuators is defined as a mobile platform, which has three to six degrees of freedom to the other platform – the base.



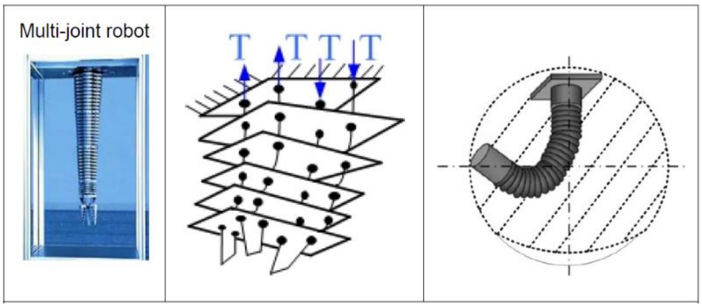
It can be moved individually in each of the three linear and three angular directions or in any combination. The resulting movement of the platform is the current movement and control of these arms. Workspace of parallel kinematic robot structure is not fixed and it needs to be calculated. The length of each joint and rotation of joints must be considered.

Kinematic structure with two arms



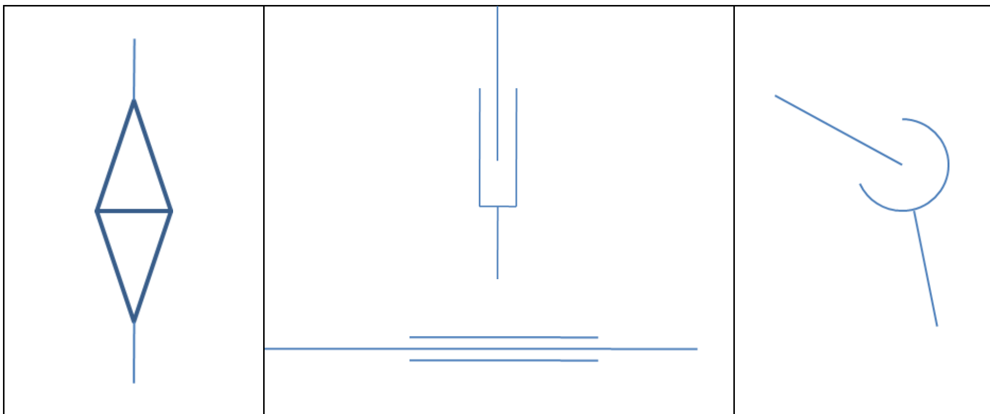
An industrial robot with two arms is equipped with 13 degrees of freedom of movement, each arm has 6 degrees of freedom and the rotation around the vertical axis of the robot is added as well. The parameters of the robot determine it for mounting or handling applications with high level of manipulation, like humans.

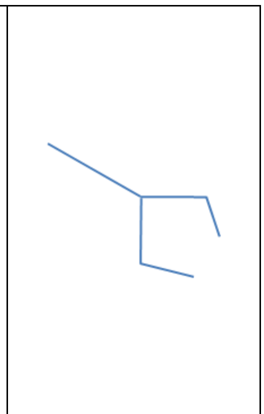
Multi-joint structure



A kinematic structure of multi-joint arrangement offers an excellent flexibility. It differs primarily in the fact that it contains no classical translational or rotational kinematic pair. The structure uses a system of steel wires for perfect control of arm, which are interwoven through a series of plates arranged according to the structure of the spine of a human, to create a workspace of a ball shape with a flat bottom. Robot workspace is characterized by very good manipulative skills in difficult to reach areas such as enclosed areas of car bodies, etc.

Symbols for industrial robots:





Rotary axes	Translatory axes	Joint	Gripper

According to a robot's joint movements there are the following well distinguished basic robot configurations:

- Cylindrical robot,
- Polar (spherical) robot
- Revolute (jointed arm) robot
- Cartesian (rectangular) coordinate robot (sliding type and gantry type)
- SCARA type robot (Selective Compliance Assembly Robot Arm)

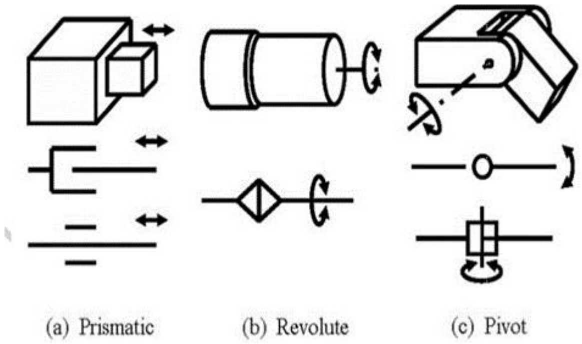


Fig. 13.5 One DOF joints and conventional representation

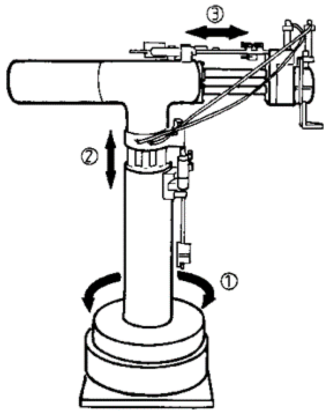
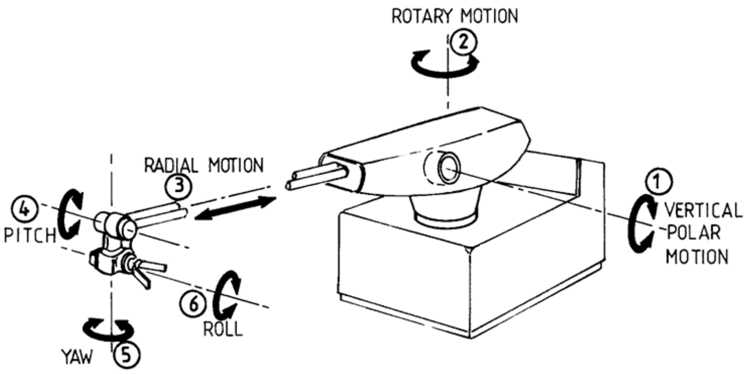


Fig. 13.6 Cylindrical robot

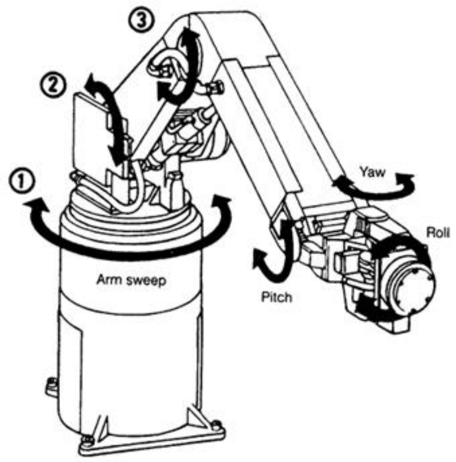
This robot has two prismatic motions and one rotation about its vertical axis.



Roll = 180 ↔ 0 + 90° = 270°
Pitch = -90 ↔ 0 ↔ + 50° = 140°
Yaw = - 45 ↔0 ↔+15° = 60°

Fig. 13.7 Polar robot

The body rotates about longitudinal axis in a vertical direction. The shoulder rotates about its transverse axis and produces vertical polar motion. The arm link extends in its axial direction, i.e. (that *is) radial motion.*



Roll = 180 ↔ 0 + 90° = 270°
Pitch = -90 ↔ 0 ↔ + 50° = 140°
Yaw = - 45 ↔0 ↔+15° = 60°

Fig. 13.8 Revolute robot

The body rotates about longitudinal axis in a vertical direction. The shoulder and elbow joints have transverse axes, and their movements are in the vertical plane. Revolute robot: The body rotates about longitudinal axis in a vertical direction. The shoulder and elbow joints have transverse axes and their movements are in the vertical plane.

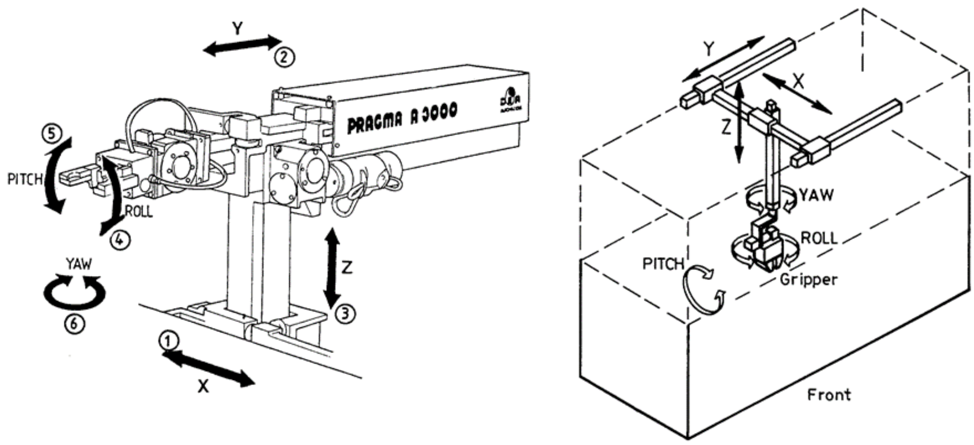


Fig. 13.9 Cartesian coordinate robot

These robots have three longitudinal motions in x, y and z directions (sliding type and gantry type)

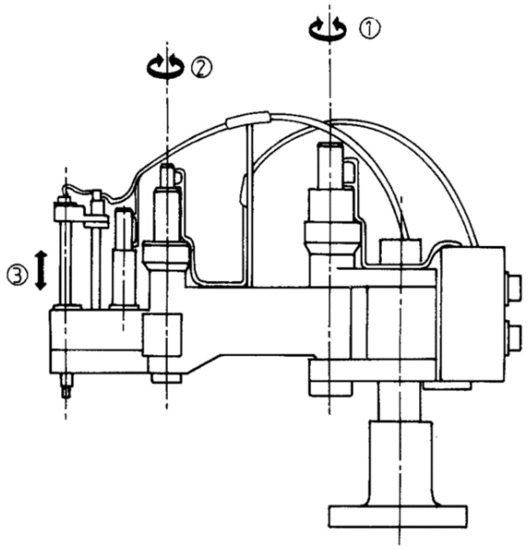


Fig. 13.10 SCARA type robot

(Selective Compliance Assembly Robot Arm) The shoulder and elbow joints can rotate in the horizontal plane about their vertical axes. At the end of the robot there is a vertical axis for lifting.

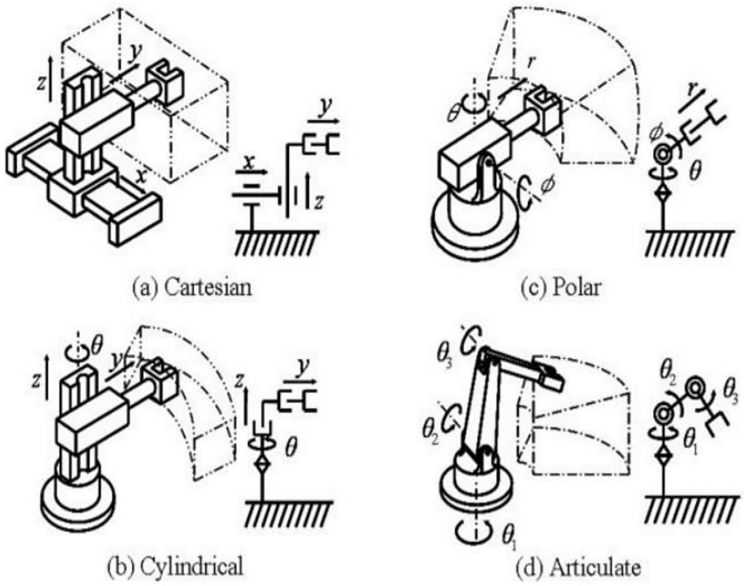


Fig. 13.11 Kinematical structures of robot configurations

Accuracy and repeatability
Accuracy relates to a robot's ability to move to a command position at specified velocity within its established working area. The position and/or velocity are usually measured at the end of the arm [4]

Accuracy is the difference between the measured value and command value at a specified position in the robot's workspace.

Repeatability is a measure of the spread positions in a series of attempts to position the manipulator at a fixed location (Fig. 13.12).

Good repeatability is the ability to repeat the same position several times within specified tolerance.

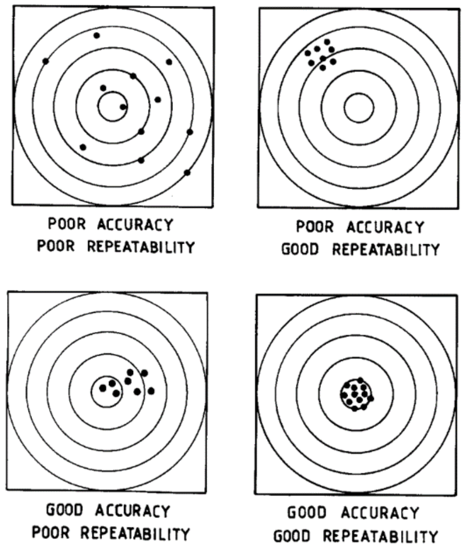


Fig. 13.12 Principles accuracy and repeatability

Kinematics of robots

The purpose of a manipulator is to manipulate its end effector. Some other names for end effector are hand, gripper, and tool. It is that part of the manipulator which physically interfaces with its environment. To perform a task the robot must know where the object to be worked on is located and what the location of the end effector should be with respect to that object. For this purpose one needs a kinematic model of the manipulator. This section presents this model and shows how it is used to define the position of each link coordinate and the end effector.

Homogeneous Transforms

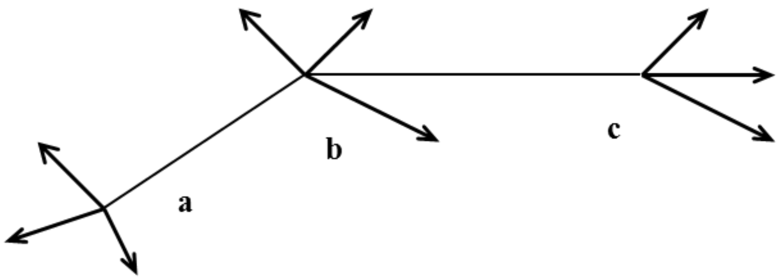


Figure shows three coordinate systems. The position and orientation of system **c** is known with respect to system **b**, and the position and orientation of system **b** is known with respect to system **a**. The problem is to determine the position and orientation of system **c** with respect to system **a.**

The orientation is defined in terms of the direction cosine matrix. This is a 3x3 matrix whose columns are three unit vectors that represent the x,y, and z axes of the right – handed orthogonal coordinate system.

A convenient way of transforming both the orientation and position from one coordinate system to another is with the use of homogeneous transforms. A homogeneous transform is a 4x4 matrix of the form:

$$A(a,b) = \begin{bmatrix} x & y & z & p \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} & D & & p \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

where:
A(a,b) is the transform of coordinate system **b** with respect to **a**,
D is the 3x3 direction cosine matrix of coordinate system **b** expressed in coordinate system **a**
p is the 3x1 vector denoting the position of coordinate system **b** expressed in **a**.
The transform for **c** expressed in **b** is denoted by **A(b,c)**. Given **A(a,b)** and **A(b,c)** the transform **A(a,c)** is computed by simply multiplying these two transforms together:

$$A(a,c) = A(a,b) \cdot A(b,c)$$

Using this technique, one has a simple notation and also an easy technique of computing the position and orientation of any coordinate system with respect to any other coordinate system.
Any homogeneous transform can be defined in terms of the product of six special transforms. This is because there are only three independent elements in the *direction cosine matrix* and three for the position. These special transforms are as follows [4]:

A rotation about the ***x*** axis at angle *θ*

$$Rot\ (x, \theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

A rotation about the ***y*** axis at angle ***θ***

$$Rot\ (y, \theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

A rotation about the **z** axis at angle ***θ***

$$Rot\ (z, \theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

A translation along the ***x*** axis a distance ***a***

$$Trans(a,0,0) = \begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

A translation along the ***y*** axis a distance ***b***

$$Trans(0,b,0) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

A translation along the ***z*** axis a distance ***c***

$$Trans\ (0,0,c) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Thus, the rotation matrix R is an orthogonal 3 x 3 matrix, which consist of the components of the unitary vectors projected to the coordinates of the base coordinate system. The translatory part is a column vector pointing from the origin to another origin of the displaced coordinate system.

Mathematical description of objects

The aim is now to show how to use homogeneous transformations for describing objects and their location in a form understandable to the robot's computers. Objects surrounded by planar surfaces are described by 4xN matrices where N designates the number of vertices of the object. Each vertex appears as a position vector of the form.

$$[x\ \ y\ \ z\ ]^T$$

where:

the superscript T stands for transpose.

For the description of objects with planar surfaces there are two options:

1. The origin of an object's coordinate system can be positioned independent of any of its features, and the orientation of its frame coordinates may also be chosen arbitrarily (see Fig. 13.13 a)

The general matrix presentation of an object with N vertices is of the form:

$$[\text{object}] = \begin{bmatrix} x_0 & x_1 & \cdots & x_{N-1} \\ y_0 & y_1 & \cdots & y_{N-1} \\ z_0 & z_1 & \cdots & z_{N-1} \\ 1 & 1 & \cdots & 1 \end{bmatrix}$$

2. The origin of an object coordinate system can be fixed to a feature of the object (see Fig. 13.13 b).

Let the object be a cuboid represented by its vertices **P_0 – P_7** in Cartesian coordinates as shown in Fig. 13.13 (b). The origin of the fixed coordinate system is chosen at P_0. Such a choice makes the description of the object quite easy.

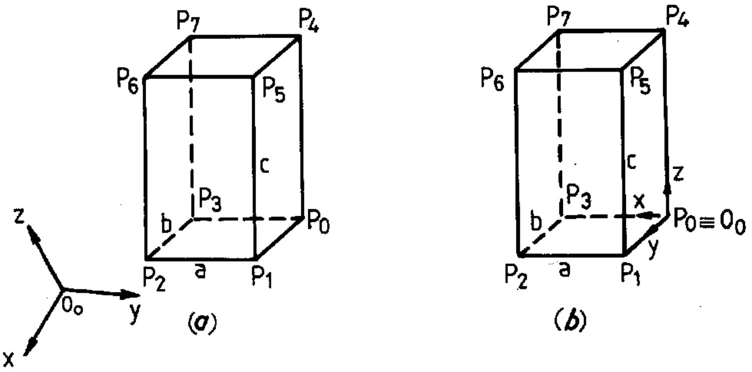


Fig. 13.13 Object's coordinate system – 2 options for positioning coordinate systems [4]

Suppose the cuboid has ends a x b units with height c units in the x,y and z directions respectively.
This leads to the description matrix of the cuboid as given in Equation:

$$[\text{cuboid}] = \begin{array}{c} \begin{bmatrix} 0 & 0 & a & a & 0 & 0 & a & a \\ 0 & b & b & 0 & 0 & b & b & 0 \\ 0 & 0 & 0 & 0 & c & c & c & c \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \\ \begin{matrix} \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow \\ \mathbf{P}_0 & \mathbf{P}_1 & \mathbf{P}_2 & \mathbf{P}_3 & \mathbf{P}_4 & \mathbf{P}_5 & \mathbf{P}_6 & \mathbf{P}_7 \end{matrix} \end{array}$$

Let us now perform a translation and a rotation on the object. This procedure will be described by a 4 x 4 transformation matrix **H**. The relation between the new and starting position of an object is given as:

$$[\text{object}]_n = H \cdot [\text{ object }]_s$$

where:
n denotes the new position,
s the starting position of the object, i.e., in this case the new position refers to the base coordinate frame and the starting position to the displaced coordinate system.
The expanded form of equation becomes.

$$\begin{bmatrix} x_{0,n} & x_{1,n} & \cdots & x_{N-1,n} \\ y_{0,n} & y_{1,n} & \cdots & y_{N-1,n} \\ z_{0,n} & z_{1,n} & \cdots & z_{N-1,n} \\ 1 & 1 & \cdots & 1 \end{bmatrix} = \left[\begin{array}{ccc|c} & \boldsymbol{R} & & \boldsymbol{k} \\ \hline 0 & 0 & 0 & 1 \end{array}\right] * \begin{bmatrix} x_{0,s} & x_{1,s} & \cdots & x_{N-1,s} \\ y_{0,s} & y_{1,s} & \cdots & y_{N-1,s} \\ z_{0,s} & z_{1,s} & \cdots & z_{N-1,s} \\ 1 & 1 & \cdots & 1 \end{bmatrix}$$

The matrix on the left – hand side represents the vertices of the cuboid in their new position after transformation. Matrices on the right – hand represent the transformation matrix and the vertices of the cuboid in the starting coordinate frame respectively.

Mathematical description of objects – illustrative example

(Description of the wedge by transformation matrices.)

Specification

A wedge is shown on both its initial and overturned position in Fig. 13.14 (a) and (b) respectively. A Cartesian coordinate system is fixed to one of the wedge's arbitrarily chosen vertices. In this case the origin is assigned to the vertex at A(0,0,0) shown in Fig.13.14 (a).

Objectives

1. Construct the description matrix W_0 of the wedge.
2. Describe a homogeneous transformation which translates the wedge by 2 units along x axis and -3 units along the **z** axis with zero translation along the **y** axis, see Fig. 13.14 (c).
3. Derive the coordinate transformation which moves the same edge from its original position into a new, overturned position, as shown in Fig. 13.14 (b) and (d)

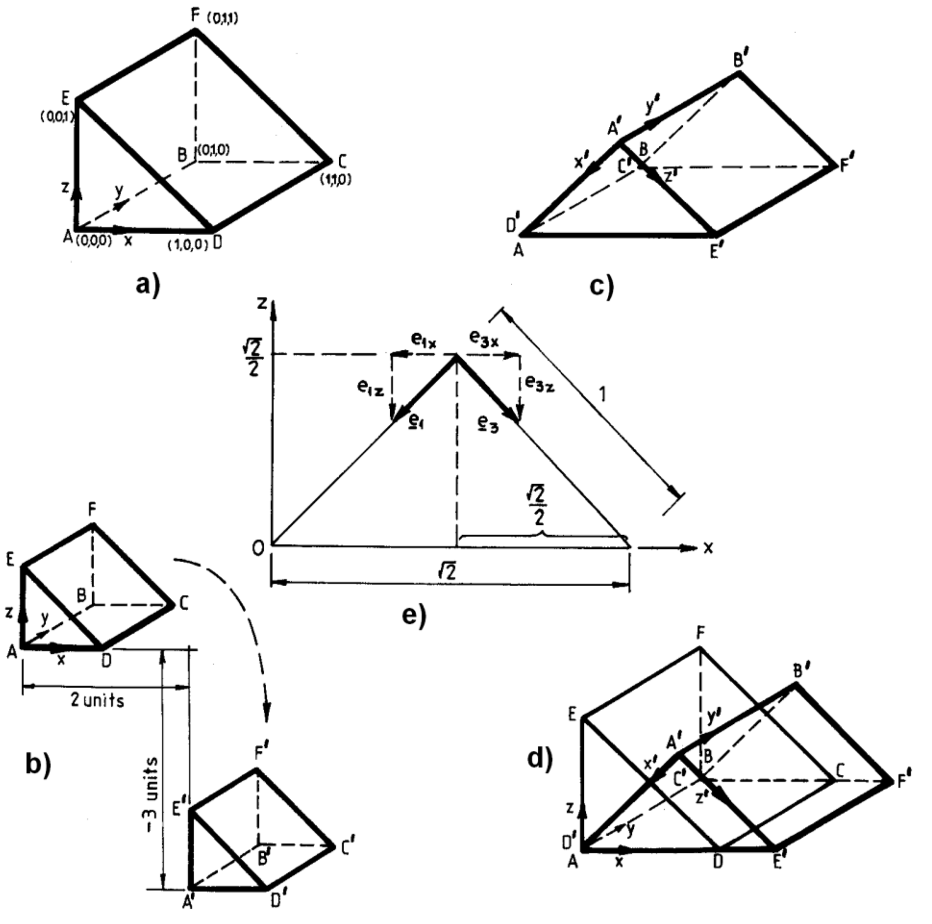


Fig. 13.14 Wedge in different positions – after translation and rotation [4]

Solutions:

1. The description matrix $\boldsymbol{W_0}$ of the wedge being in its initial state is derived as:

$$W_0 = \begin{pmatrix} 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

Note that the sequence of the vertices in the matrix $\boldsymbol{W_0}$ is arbitrary.

2. The shift of the wedge by 2 units along $\boldsymbol{x}$ axis, -3 units along $\boldsymbol{z}$ axis and no translation along the $\boldsymbol{y}$ axis is described by homogeneous transformation matrix $\boldsymbol{H_1}$:

$$H_1 = \begin{pmatrix} 1 & 0 & 0 & 2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -3 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

The matrix $\boldsymbol{W_1}$ describing the wedge in its translated position is given by the product of matrices $\boldsymbol{H_1}$ and $\boldsymbol{W_0}$ as

$$W_1 = H_1 \,.\, W_0$$

$$W_1 = \begin{pmatrix} 1 & 0 & 0 & 2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -3 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} = \begin{pmatrix} 2 & 2 & 3 & 3 & 2 & 2 \\ 0 & 1 & 1 & 0 & 0 & 1 \\ -3 & -3 & -3 & -3 & -2 & -2 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

3. When the wedge is turned over from its original position as shown in Fig. 13.14 (a) into another state as shown in Fig. 13.14 (b) and Fig.13.14 (d), the homogeneous transformation is constructed as given below. By looking at this new state of the wedge we can see how its position and orientation have been changed. The coordinates of the vertex of the new origin at A' can be from Fig. 13.14 (e) as

$$x_0 = \frac{\sqrt{2}}{2}, \qquad y_0 = 0, \qquad z_0 = \frac{\sqrt{2}}{2}$$

The new (x, y, z) axes, corresponding to the unit vector triad (e1, e2, e3) are as follows

$$\overline{e_1} = \begin{pmatrix} e_{1x} \\ e_{1y} \\ e_{1z} \end{pmatrix} = \begin{pmatrix} -\frac{\sqrt{2}}{2} \\ 0 \\ -\frac{\sqrt{2}}{2} \end{pmatrix}, \qquad \overline{e_2} = \begin{pmatrix} e_{2x} \\ e_{2y} \\ e_{2z} \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \qquad \overline{e_3} = \begin{pmatrix} e_{1x} \\ e_{1y} \\ e_{1z} \end{pmatrix} = \begin{pmatrix} \frac{\sqrt{2}}{2} \\ 0 \\ -\frac{\sqrt{2}}{2} \end{pmatrix}$$

Thus, the transformation matrix $\boldsymbol{H_2}$ is given as

$$H_2 = \begin{pmatrix} -\frac{\sqrt{2}}{2} & 0 & \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \\ 0 & 1 & 1 & 1 \\ -\frac{\sqrt{2}}{2} & 0 & -\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

The description matrix $\boldsymbol{W_2}$ of the wedge in its overturned position shown in Fig. 13.14 (d) becomes:

$$W_2 = H_2 \,.\, W_0$$

$$W_2 = \begin{pmatrix} -\frac{\sqrt{2}}{2} & 0 & \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \\ 0 & 1 & 1 & 1 \\ -\frac{\sqrt{2}}{2} & 0 & -\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} = \begin{pmatrix} \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 & 0 & \sqrt{2} & \sqrt{2} \\ 0 & 1 & 1 & 0 & 0 & 1 \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

The equation provides the new coordinates of the wedge's vertices in its overturned position.

Transformation with homogeneous matrix – illustrative example

Since the position of the element in the space can generally be achieved by translational and rotational motion, the subject of further analysis will be the issue of homogeneous transformations for translational and rotational movement. Each transformation in the space will be described by a square degree 4th matrix. The transformation corresponding to the vector translation $a\bar{i} + b\bar{j} + c\bar{k}$ is given by a matrix.

$$Trans(a,b,c) = \begin{pmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

where:

$\bar{i}, \bar{j}, \bar{k}$ are the unit vectors.

The transformation corresponding to the rotation around the **x, y** and **z** axes with the angle **θ** is given by the matrix:

$$Rot(x,\theta) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad Rot(y,\theta) = \begin{pmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad Rot(z,\theta) = \begin{pmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Based on these homogeneous matrices, we will solve an illustrative example. Let the point determined by the vector $\bar{u} = 7\bar{i} + 3\bar{j} + 2\bar{k}$ gradually rotate around the axis **z** 90^0, then around the **y**-axis by 90^0 and finally move it by the vector $4\bar{i} - 3\bar{j} + 7\bar{k}$ (Fig. 13.15).

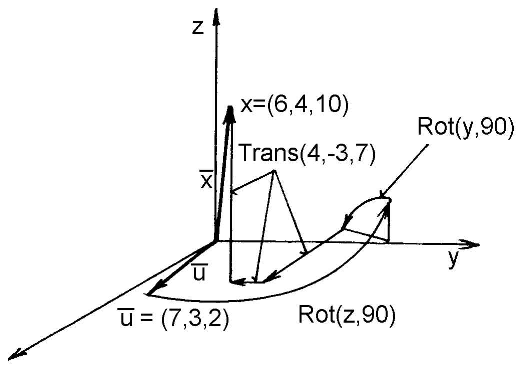


Fig. 13.15 The sequence of transformations according to the illustrative example

Shorthand notation transformation will be:

$$Trans(4,-3,7) \cdot Rot(y,90) \cdot Rot\,(z, 90)$$

Based on the above, we will first calculate:

$$Rot(y,90).Rot(z,90) = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} . \begin{pmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

In another may then be calculated:

$$\begin{pmatrix} 1 & 0 & 0 & 4 \\ 0 & 1 & 0 & -3 \\ 0 & 0 & 1 & 7 \\ 0 & 0 & 0 & 1 \end{pmatrix} . \begin{pmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 & 4 \\ 1 & 0 & 0 & -3 \\ 0 & 1 & 0 & 7 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

This is how we get a transform matrix that describes a defined motion. Then the point determined by the vector $\overline{u} = 7\overline{i} + 3\overline{j} + 2\overline{k}$ is expressed in the form of the column matrix and the result is expressed as the product of the two matrices:

$$\begin{pmatrix} 6 \\ 4 \\ 10 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 & 4 \\ 1 & 0 & 0 & -3 \\ 0 & 1 & 0 & 7 \\ 0 & 0 & 0 & 1 \end{pmatrix} . \begin{pmatrix} 7 \\ 3 \\ 2 \\ 1 \end{pmatrix}$$

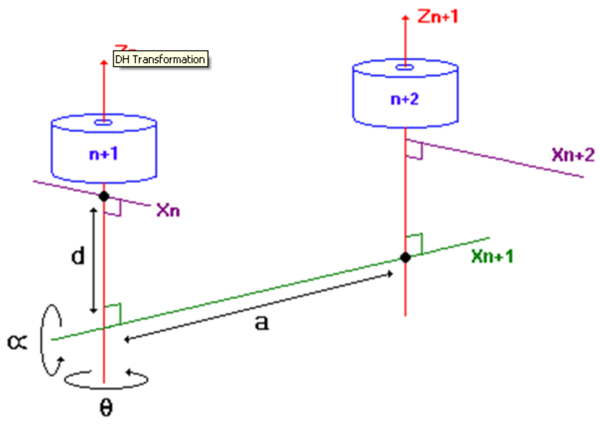
The transformed vector can be expressed $\overline{x} = 6\overline{i} + 4\overline{j} + 10\overline{k}$. The outcome and progress of transformation is expressed in Fig. 13.15 above.

Transformation along the kinematic chain

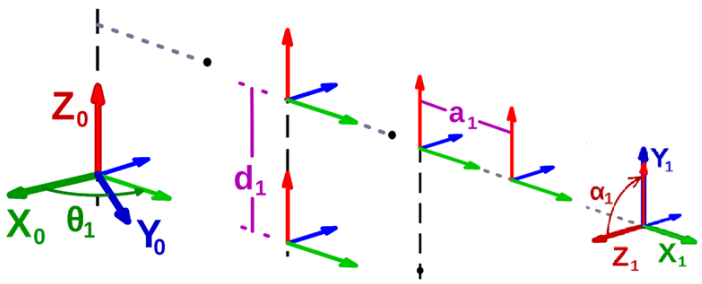
An anthropomorphic robot arm consists of a sequence of links connected by activated joint. A robot manipulator of ***n*** degrees of freedom is considered to have ***n*** joints connected by ***n*** links. The assembly of sequential links and joints makes up a kinematic chain. The function of

the links is to maintain a fixed relationship between the manipulator joints at each end of the link.
In order to describe the relationship between the elements of kinematic chain, we will assign a coordinate system to each joint.
http://www.youtube.com/watch?v=rA9tm0gTln8
http://en.wikipedia.org/wiki/Denavit-Hartenberg_Parameters
http://www.ee.unb.ca/tervo/ee4353/dhxform.htm

The Denavit-Hartenberg Convention
In this chapter we develop the forward or configuration kinematic equations for rigid robots. The forward kinematics problem is concerned with the relationship between the individual joints of the robot manipulator and the position and orientation of the tool or end-effector. The Denavit-Hartenberg representation has become the standard way of representing robots and modeling their motions. The method begins with a systematic approach to assigning and labeling an orthonormal (x,y,z) coordinate system to each robot joint. It is then possible to relate one joint to the next and ultimately to assemble a complete representation of a robot's geometry. This webpage illustrates the second step that of describing the transformations necessary to relate one joint to the next.



- The origins of the coordinate frames at link **n+1** and link **n+2** are marked with a black dot in the figure. Four transformations in a specific order will bring these frames coincident with each other. Coordinate axis $\boldsymbol{Z_n}$ passes though link ***n+1***. In summary:



- A rotation $\boldsymbol{\theta_{n+1}}$ about the $\mathbf{Z_n}$ axis to bring $\mathbf{X_n}$ parallel with $\mathbf{X_{n+1}}$;
- A translation $\mathbf{d_{n+1}}$ along the $\mathbf{Z_n}$ axis to make the **x**-axes collinear;
- A translation $\mathbf{a_{n+1}}$ (or **r**) along the **X** axis to make the **z**-axes coincide, and;
- A rotation $\boldsymbol{\alpha_{n+1}}$ about the $\mathbf{X_n}$ axis to bring $\mathbf{Z_n}$ parallel with $\mathbf{Z_{n+1}}$;

Together, these four transformations in the above order led to a unique homogeneous transformation matrix with four variables representing the relationship between these two links (Fig. 13.16).

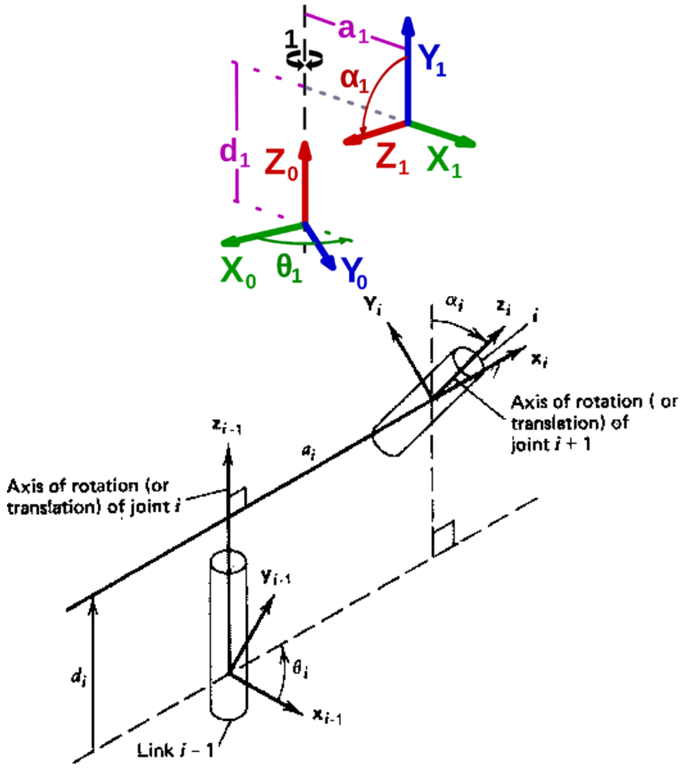


Fig. 13.16 Parameters relating adjacent link coordinate systems

The position and orientation of the i – coordinate system (Fig. 13.17) of the corresponding joint of the robot kinematic chain can be transformed into a *(i - 1)* coordinate system by means of a matrix as a result of the product of four homogeneous transformation matrices, denominated as the Denavit – Hartenberg matrix, referred to in the world literature as D – H matrix, respectively. A – matrix. Can write.

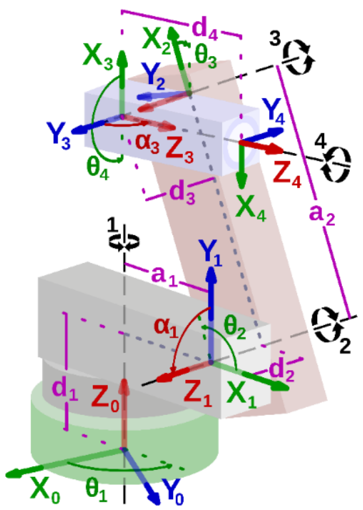


Fig. 13.17 Determination of coordinate systems in robot joints with parameters according to Denavit-Hartenberg

$$A_{i-1,i} = Rot(z, \theta_i) \,.\, Trans(0,0, d_i) \,.\, Trans(a_i,0,0) \,.\, Rot(x, \alpha_i)$$

and after adding the appropriate parameters into the transformation matrices, we get the resulting Denavit-Hartenberg transformation matrix

$$A_{i-1,i} = \begin{pmatrix} \cos\theta_i & -\sin\theta_i \cos\alpha_i & \sin\theta_i \sin\alpha_i & a_i \cos\theta_i \\ \sin\theta_i & \cos\theta_i \cos\alpha_i & -\cos\theta_i \sin\alpha_i & a_i \sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

If we choose the n - the coordinate system of the robot effector, by multiplying A – matrices

$$T_e = A_{0,1} \,.\, A_{1,2} \,.\, A_{2,3} \,.\, \dots \,.\, A_{n-1,n}$$

we get a known matrix that includes the position and orientation of the robot effector relative to the robot coordinate system (Fig. 13.18).

$$T_e = \begin{pmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

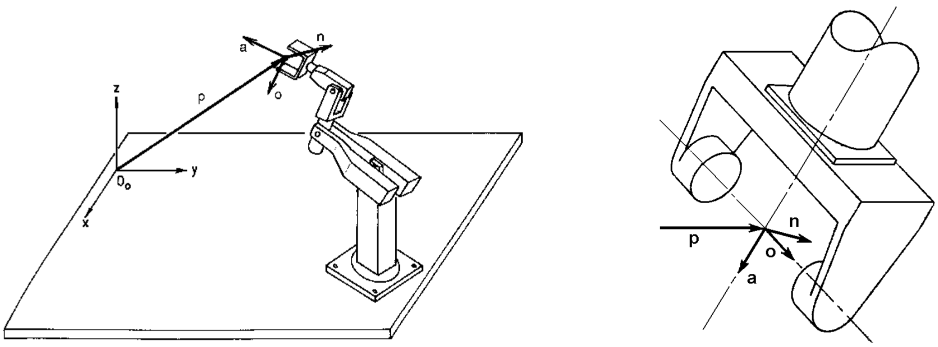


Fig. 13.18 The position and orientation of the robot effector relative to the robot coordinate system

Mechanical section

Mechanical section of an industrial robot consists of links and joints, and joints are used to implement a robot motion and solids are links between them. Each joint provides a degree of freedom. Most robots have 5 possibly 6 degrees of freedom. The mechanical section of a robot consists of a base, carousel and arms, such as depicted in Fig. 13.19

The end effector is a separate part of a robot, which serves for accommodating objects – griper or a technological head, such as a welding torch, Fig. 13.20. Together with industrial robots it is involved in the implementation of positioning and orientation of carried items. According to their intended use these are divided into gripers, head, integrated effectors and tools.

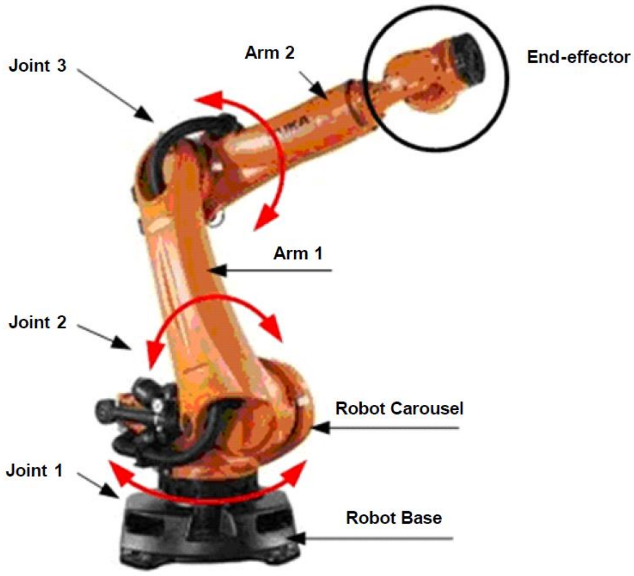


Fig. 13.19 Industrial robot with moving axes

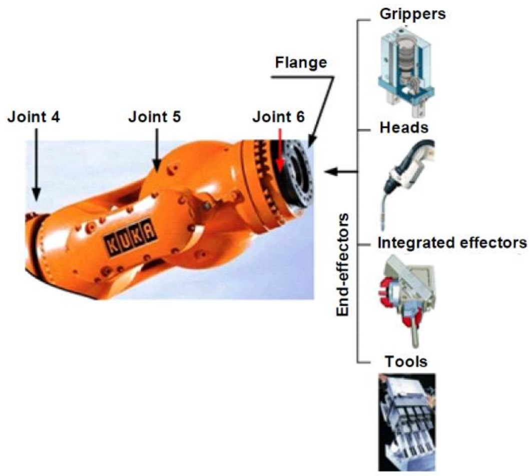


Fig. 13.20 The end of a robot arm

A robot control system

Its mission is based on information stored in the memory of a control computer and information obtained from sensors. This collected information is used for a robot action plan to decide on the actions to be performed. A block diagram of control of industrial robot is shown in Fig. 13.21.

It includes all functions of management of positioning. In addition it offers the possibility of the current management of peripheral devices. Management of the robot is a microprocessor system, which operates under multitasking methods. It is possible to handle simultaneously several sequential management processes.

Input and output level control is the choice of the design presented by either as a fieldbus technology or as discrete inputs and outputs. The serial interface is configurable and can be used to connect intelligent devices such as barcode scanners, image processing systems, etc.

Robot control systems are built on a PC base with a processor, equipped with a CD drive or floppy disk. As an external storage unit is used hard drive on which operating system is stored and real-time module to work in real time. The control system can be equipped with multifunctional card, which forms the interface between the programming unit, a PC security logic.

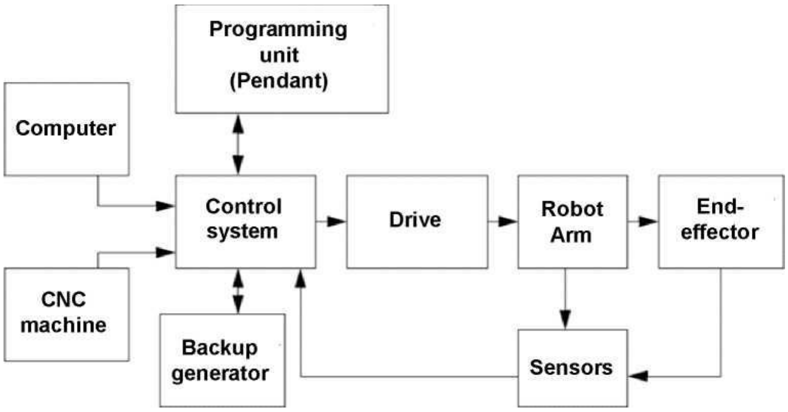


Fig. 13.21 Block diagram of industrial robot control

Control system allows:

- The establishment of programs, editing and saving programs
- Diagnosis of putting into operation
- Path planning
- Power control of servo module
- Communication with external modules

Programming unit

Programming a robot is done by so-called programming panel – pendant. This is a large, clear display, showing the progress of the program and its current status bar, switch between manual and automatic operation and selecting multiple imaging windows. By its sides it has function keys for various settings such as speed, choice of coordinate system, and more. Pendant also includes 6D mouse to control the robot in manual mode as well as buttons to control the robot separately in each axis, Fig. 13.22. As with any electrical device, pendant is equipped with a central stop for ensuring safety. To facilitate programming and diagnostics there is a software used with a number of additional features.

Fig. 13.22 Pendant functions description

13.4 Kinematics

The choice of the robot kinematics is mainly dependent on the mutual arrangement of a number of kinematic pairs (providing the individual movements). Kinetic properties of the robot are given by the number of rotational and translational axes and their arrangement.

13.5 Control of robots

Robot control is achieved through the robot control system using information obtained from sensors. The control system enables to guide the robot to the desired position by means of programming unit (pendant). Through Inputs/Outputs it is possible to attach various devices on the robot workstation operated by robot control, such as(conveyor, positioning tables,...). The control system is able to manage directly the unit as an axis. Modern control systems are equipped with connectivity to one of many technological hubs (DeviceNet, Profibus, Interbus, Ethernet, ...) and connect the robot to better structure the manufacturing system (CIM). Robots are equipped with industrial applications that simulate the programming unit of the PC. Controlling the robot and its programming can be implemented as follows directly from the PC. The governance structure of the robot is in Fig. 13.23.

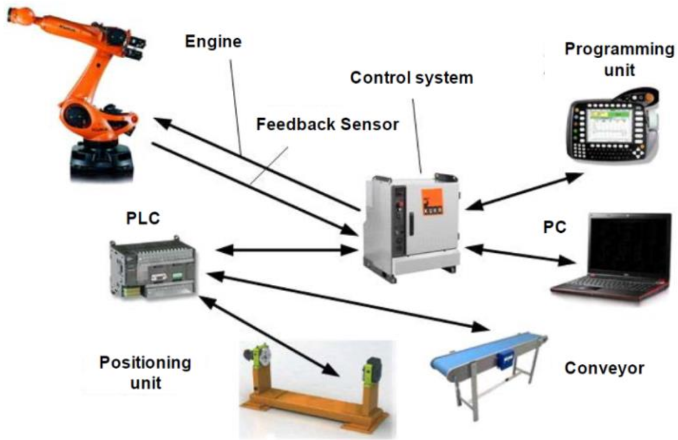


Fig. 13.23 Structure of robot control

The control system ensures the generation of signals for each axis of the robot. To ensure the required action of a robot, the control system must receive feedback information from the sensors (speed, torque, position ...). Block diagram of control power unit with electric drive is shown in Fig. 13.24.

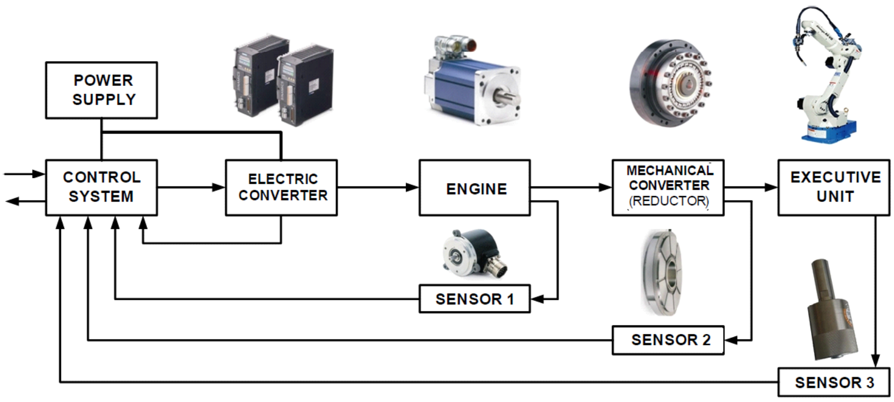


Fig. 13.24 Scheme of control of robot electric propulsion units

The main task of the robot control system is to generate signals for servo propulsion of a robot created by the program in automatic mode, or generate signals based on the functions of movement counterpart. Control system processes information from internal sensors of the state of the system, but also information from external sensors around. For this purpose,

control system is equipped with a number of digital inputs/outputs as well as several analog inputs/outputs. The internal structure of the robot control system is in Fig. 13.25.
The control system can be divided into a central control unit that processes and executes the robot program.

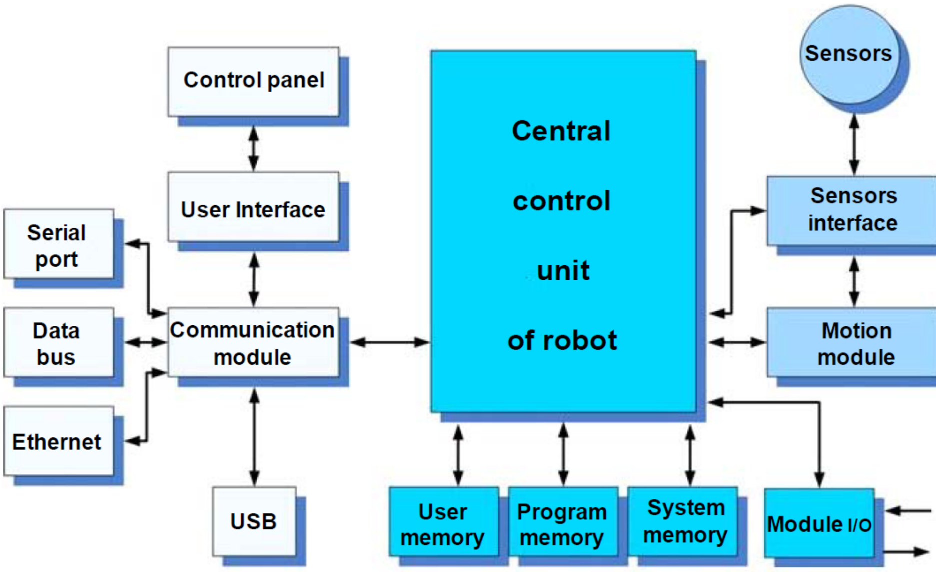


Fig. 13.25 The internal structure of the robot control system

Communication module for interfacing the robot with the environment through technology hub and a module for motion control and management of the powertrain.
Among the standard features of robot control systems are included:

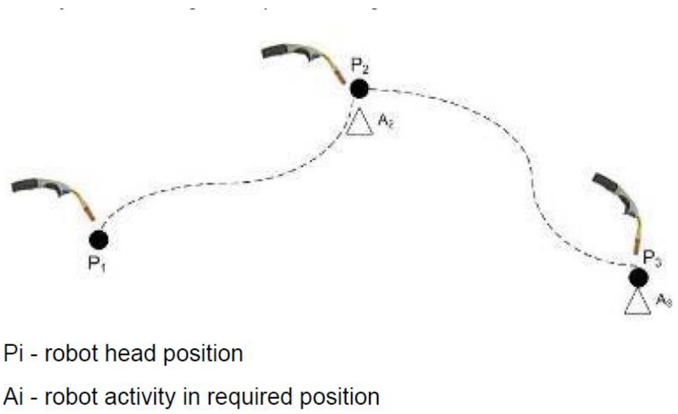
- Multitasking operation system
- Built-in mathematical functions
- Discontinuation treatment
- Axial, linear and circular interpolation
- Tracing the conveyor belt
- Connection of camera systems
- Possibility to create macros and subprograms
- Built-in palletizing
- Management of multiple axes
- The cooperation of several robots in a common workspace
- Digital inputs and outputs

13.6 Motion control of industrial robots

According to the nature of the motion control we distinguish management of industrial robots:

- Point control (PTP – Point to Point) this type of management is used when it is necessary to achieve certain points in robot workplace, between whom there is no link.
- Track control (CP – continuous path) this type of management is used if we need to drive the robot throughout the path of movement.

Point control is going through a sequence of discrete points in space. Fig. 13.26 shows a possible sequence of movements P1-P2-P3 at various points with example of a program formulated in natural language. Trajectory between the points is not defined, axis move without a functional context. Point control is used mainly for handling and spot welding [6].

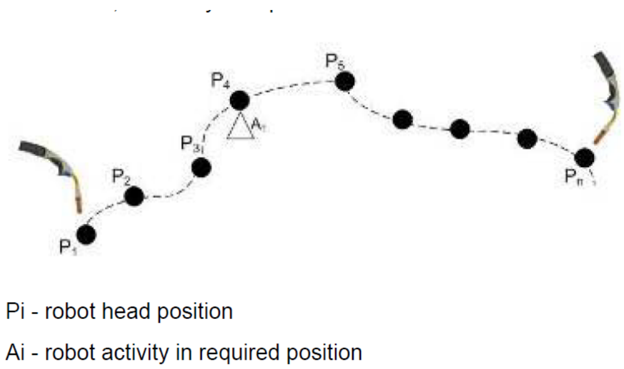


Step	Activity
1	Start in P1
2	Go to position P2
3	When P2 reached, execute activity A2
4	After finishing A2, go to position P3
5	When P3 reached, execute activity A3

Fig. 13.26 An example of PTP robot motion control [6]

The robot passes each point with a positioning accuracy. After reaching the exact location of the command it causes the execution of activity A2-A3. Sometimes the accuracy is not required (for example – in circumvention of obstacles), and time loss for positioning works rather distracting. After reaching a defined point of the surrounding of auxiliary point movement is not interrupted, but continues with given rate at the next point. For example, if Fig. 13.26 is such a point P2 (A2 may come off), shall enter an order no. 2 shapes: 2 Go through the auxiliary point P2.
Control with path behavior allows programming and navigating through defined motion paths with a functional relationship in movement in different axes. In doing so, there are two possibilities:

1. **Multipoint control (MP)** program includes motion in the form of a dense sequence of points in space that are entered in a fast time sequence (after 10 to 100 ms) from the axis position controller. These procedures are shown in Fig. 13.27. Activity A1 is performed when the value entered into the coordinates of a point P4 position controller, regardless of whether that position has been reached or not. A small loss of time, which in fact occurs, is usually acceptable.



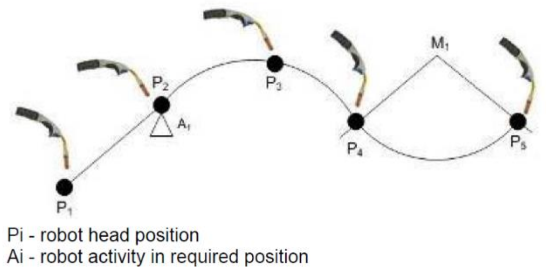
Step	Activity
1	Start in position P1
2	Go to position P2
3	Go to position P3
4	Go to position P4
5	Execute required activity A1 in position P4
6	Go to position P5

Fig. 13.27 An example of MP robot motion control [6]

Multipoint control is mainly used in paint spraying robots, but also for spot welding and surfaces processing such as grinding, polishing. Programming is done mainly by the method of direct teaching.

2. **Path control (CP)** allows you to scroll through mathematically defined pathways. An example is shown in Figure 13.28. Programming is performed directly by "teach-in" or through the text. Computer (interpolator) determines certain number of values while evaluating the same motion on a curve path and enter them into the position controller in

accordance with the program speed. A typical example of path control/management is arc welding.



Step	Activity
1	Start in position P1
2	Go to position P2 via line
3	Execute activity A1 in position P2
4	Go to poisition P4 via P3 via circle;
5	Continue via circle to position P5

Fig. 13.28 An example of path control robot motion

13.7 Interpolations

Interpolation is the process of defining the functions, which, according to certain values, passes given points. The robotic technology uses point, linear and circular interpolation.
When moving PTP (point-to-point) movement begins and ends for all axes simultaneously. PTP mode is used for preparatory movements, which requires the robot move as quickly as possible to the desired position. When you move the robot to the desired point, approximation can be applied, which allows us to move smoothly without unnecessary slowdown. In case of KUKA robot, approximation is indicated in welding as a percentage (0-100%). When 0% is a robot with an explanation to the selected point in the case of other values that point is bypassed by the set value of approximation. In the approximation of 100% and linear motion, the robot begins to circumvent the desired point in the middle of the path between the start and end point. Fig. 13.29 shows the type PTP movement without approximation. The track of robots arm movement cannot be determined in advance in this case. Endpoints are defined, in which the robot will come exactly or circumvent them according to the approximation parameters, Fig. 13.30.

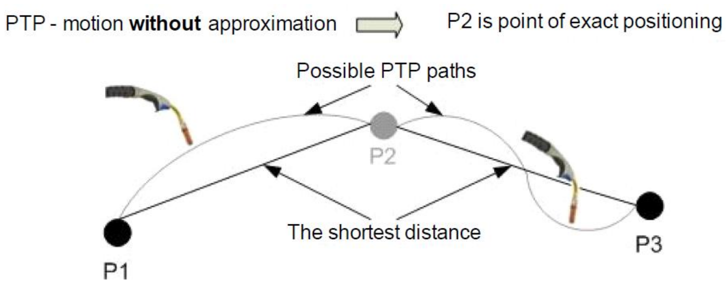


Fig. 13.29 PTP motion without approximation [6]

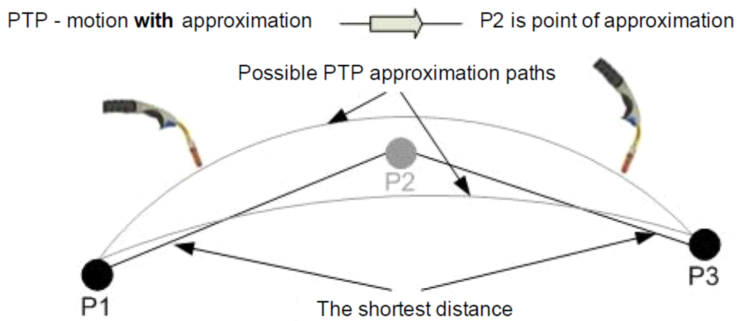


Fig. 13.30 PTP motion with approximation

If necessary, of keeping the track in the desired direction arm movements can be performed by LIN type (motion in a straight line) and CIR (Circular motion). When the movement type LIN, robot arm moves along in a straight line. This type of movement is slower than the PTP, since there is a need to calculate the points of robot path. Movement in a straight line is used when you need accurate placing of the end of effector to the desired position, for example during installation, or circumvention of obstacles in the robot workspace. Fig. 13.31 illustrates the movement in a straight line without setting a parameter of approximation and Fig. 13.32 using the approximation.

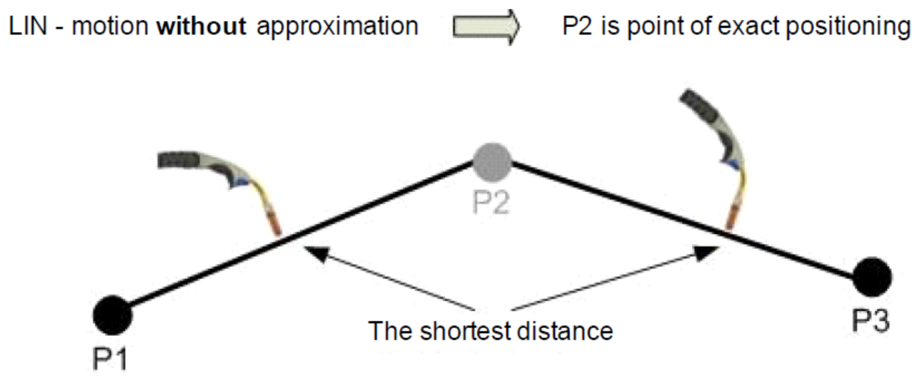


Fig. 13.31 LIN motion without approximation

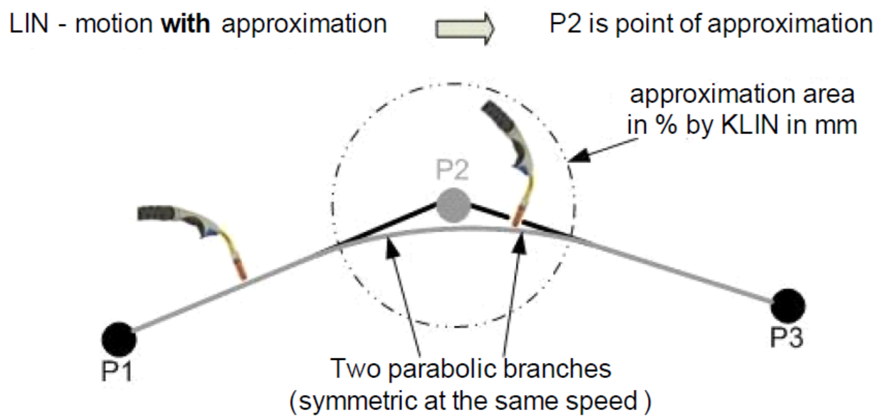


Fig. 13.32 LIN motion with approximation

The last type of motion is circular – CIR. It is needed to enter not only the start and the end point but an auxiliary point as well into the control system so that it can then realize the type of motion. Again, it is possible to use motion without approximation. Approximation does not apply to an auxiliary point, Fig.13.33 and Fig. 13.34.

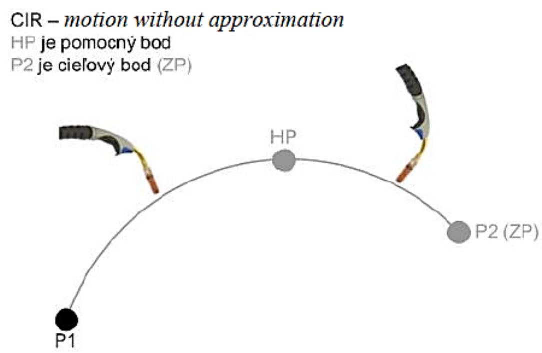


Fig. 13.33 CIR motion without approximation

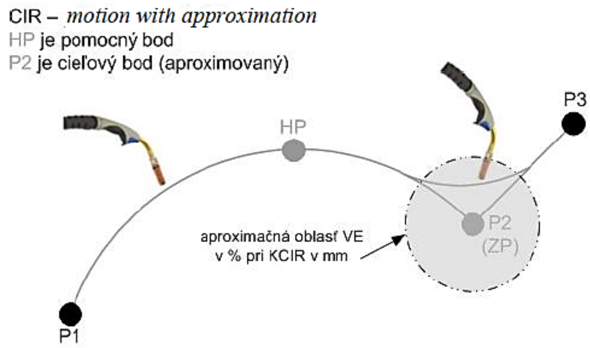


Fig. 13.34 CIR motion with approximation

13.8 Robots Programming

Robot is operating under a pre-established program. The program is defined as a sequence of instructions which lead to execution of required activity. Programming of robot is defined as the compilation and creation of program based on algorithm. According to the approach to the development of the program the programming is divided:

- on-line programming (programming by robot using pendant)
- off-line programming (programming outside robot using PC)

On-line programming

On-line programming is done directly by an operator through required handling points and robot is controlled manually to single points which allows him to put these points into memory and save the points. As step No2 logical part of controlling and periphery devices must be programmed. In this section, the speeds of the movement paths of the robot are set. The main advantage of this mode is programming and in the real environment which avoids accuracy problems as well as functional test can be done. See Fig. 13.35.

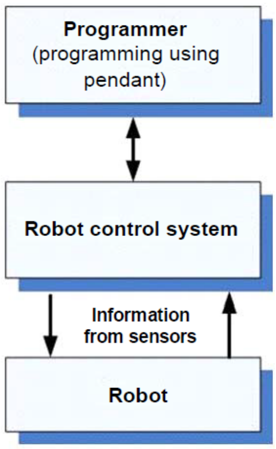


Fig. 13.35 The procedure for on-line programming of robots

Nowadays, modern programming units are already built on a PC platform. Operation is easy via function keys. The display has an option to view several windows for visual display features of a robot, or technology program and their parameters. Built-in color display allows the operator to program directly through the I/O of their activities. Some programming unit (Coma Robotics) are capable of transmitting data to the control system wireless, Fig. 13.36.

Fig. 13.36 Programming units of OTC Daihen and Comau Robotics

The disadvantage of on-line programming is a relatively long programming time, physical burden on programmer in handling complex movements and long cycles. Another disadvantage is that all workplace of production is out of order when programming the robot, only some of the devices may work in limited mode.

Play-back mode programming
Entire working process is controlled manually, and control unit remembers the movement every 20ms and records information about positions, points and orientation. Such programming can be used at simple, inaccurate robots like painting robots. When an automated mode is launched robot plays recorded path and operations. Repeating paths is not exact. Another disadvantage of this mode is the presence of operating staff. Program producing is fast and quick and is suitable for low production too.

Teach-in programming
Robot arm is guided by robot staff using the control panel keys. Coordinates of defined positions and tools orientation are saved in memory. Speed information is also set at this stage. Then in automated mode robot uses saved data. Single points of position and tools orientation are entered manually. Other functions are programmed using Path control (PC). Fig. 13.37.

Fig. 13.37 Teach-in programming

Description of selected features of the programming unit depicted in Fig. 13.38.

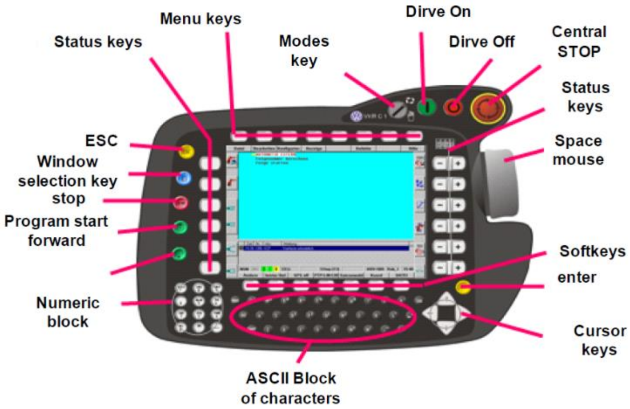


Fig. 13.38 Programming unit - pendant for robot programming (KUKA)

Some videos of programming with pendant are here:
https://www.youtube.com/watch?v=FZ3aX2SbZaQ
https://www.youtube.com/watch?v=dSKFwYMvyxU
https://www.youtube.com/watch?v=MueFQdrQD4M

Key	**Description**
ESC	Any activity can be stopped at any time, dates not saved, opened windows closed
Window selection key	Switching over opened windows if windows accessible. Selected window marked with color.
Stop	Stops active programme which is in automated mode.
Start forward	Starts selected programm
Starts backward	Robot moves opposite programmed direction, moves are completed backwards way to START position.
Number keys	Entering numbers. On second level the num. section is substituted by operating keys.
Enter	Enter or Return
6D mouse	For manual robot operating in all 6 axes. Measures the deviation and adjusts movement speed. +/- alternative keys
Emergency	The most important security key. Red color
Stop	In case of emergency, robot movements stop, before another operation the key must be unblocked.
Drive On	Starts movements
Drive Off	Stops movements
Operation mode	Manual mode- robot operates only when confirming key pressed. External mode- robot executes programme at programmed speed. Operating done using by operating key or by SPS (PLC)
Menu keys	Opening menu options, closing with Escape key
Status key	Left or right next to display, selecting the operation mode, values setting, functions
Softkeys	Selecting the options or functions in the soft-key. Functions modifying according to requests, soft-key changes visually

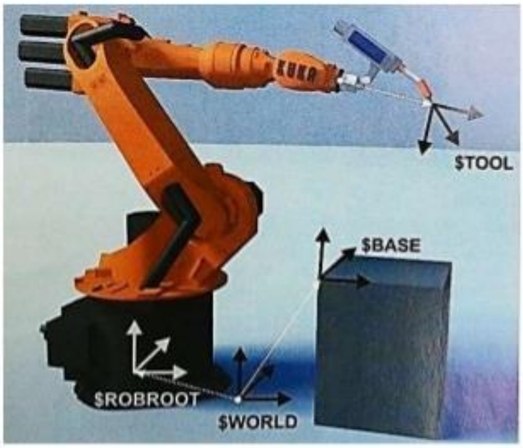
Programming unit allows monitoring of input/output and of system information, writing programs in the editor, allowing access to production data (average cycle time, number of production cycles, ...), set operating parameters, eg. Welding directly from the pendant. Modern units have analytical capabilities for optimizing robot work.

Coordinate system

A robot operates in a standard Cartesian coordinate system World. If necessary, it can be selected for a robot guidance a tool coordinate system or an external coordinate system, which is situated outside the robot, to get the robot to the desired position. If individual robot axes are necessary for a robot guidance an axially oriented system to be selected.

- Axial coordinate system
- TOOL/Tool Coordinate system TOOL
- BASE/External Coordinate system BASE
- WORLD/Basic Coordinate system WORLD

For manual robot operating to the required position in manual mode it is necessary to set the desired coordinate system. Each programming unit is equipped with buttons to manipulate individual axes of a robot arm, then we move the robot to the desired position. When adjusting the positions of the robot programmer can choose one of the options (Fig. 13.39). Choice of coordinate system is performed by setting appropriate positions according to the robot technology used.



1. **ROB ROOT COORDINATE SYSTEM**
   - The Cartesian coordinate system with the origin located at the base center of the robot.
2. **WORLD COORDINATE SYSTEM**
   - The Cartesian coordinate system with any point in the world which always remains stationary with respect to the robot as the origin.
3. **BASE COORDINATE SYSTEM**
   - The Cartesian coordinate system in which any point on the work piece is selected as the origin.

4. TOOL COORDINATE SYSTEM
- It is the Cartesian coordinate system in which the TCP (tool center point) is taken as the origin.
- It is different from other coordinate systems as the origin is dynamic in this case.

Fig. 13.39 Options of the robot coordinate system

Axial coordinate system

Every axis of robot is possible to move independently in the positive or negative direction of the axis. Movements are performed using the operating keys. Movement of the axis of robot in axial coordinate system is shown in ig. 13.40.

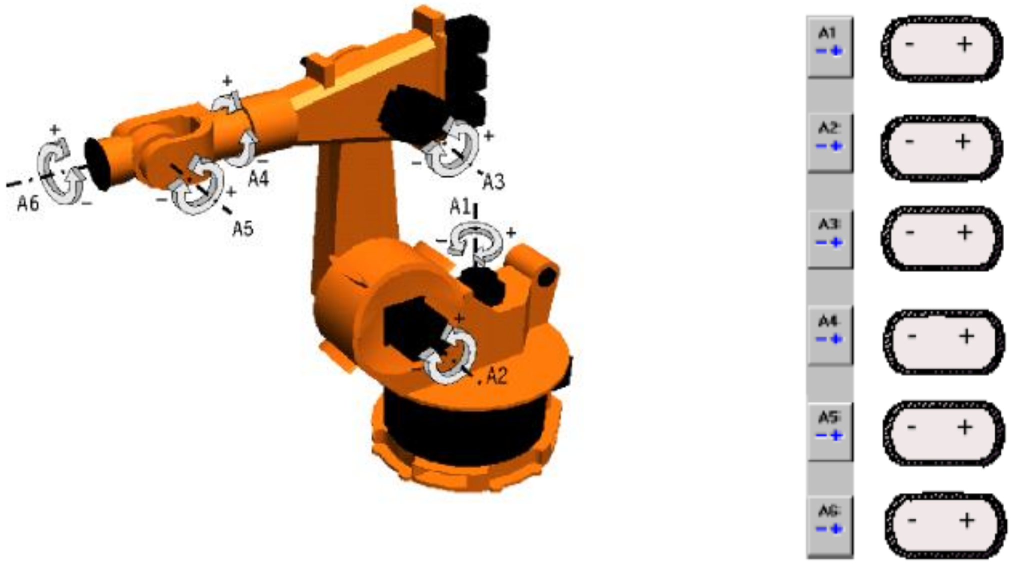


Fig. 13.40 Axial coordinate system and buttons on the pendant

Coordinate system WORLD

Relative coordinate system WORLD is absolute Cartesian rectangular coordinate system. Its beginning lies inside the cell center of robot, Fig. 13.41. The zero point of the system does not move.

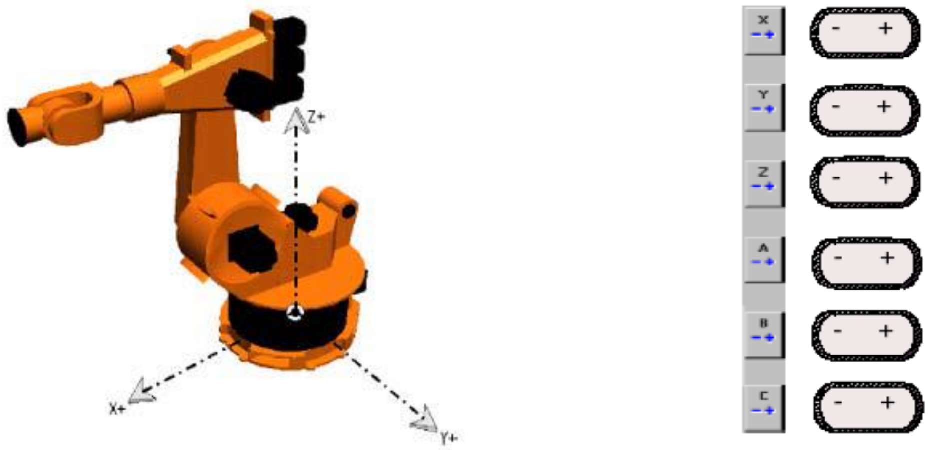


Fig. 13.41 Coordinate system WORLD and the buttons on the pendant

Coordinate system BASE

Relative coordinate system BASE is Cartesian rectangular (right-angled) coordinate system. Its center lies outside the cell, in external tool such as welding pliers, Fig. 13.42. External tool coordinate system is decisive while working in BASE. The workpiece is being moved around or alongside the axes [6].

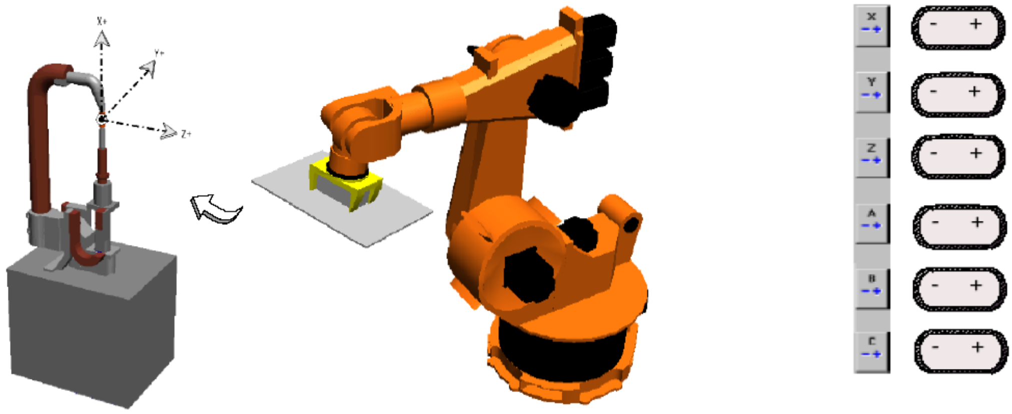


Fig. 13.42 Coordinate system BASE and the buttons on the pendant

Coordinate system TOOL

Relative coordinate system TOOL is Cartesian rectangular (right-angled) coordinate system, its center lies in the tool, Fig. 13.43. Orientation of this coordinates system is chosen so that its axis X is identical to working instrument direction. Coordinate system TOOL always follows the tool movements.

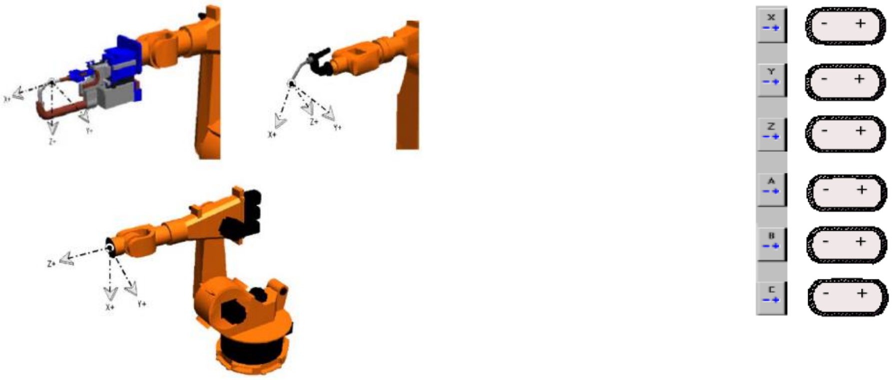


Fig. 13.43 Coordinate system TOOL

Basics of motion programming

The most important point in programming industrial robots is the actual movement of the robot along the programmed path. The operator or programmer enters, using the KRL programming language, the individual instructions for the movement of the robot, which are stored in the memory of the control system.

Motion instructions and commands allow the robot to move from point to point or move along the path. For both types of movements, programming takes place from the current position to the new position of the robot. Each movement has well-defined parameters that affect the progress of the programmed movement. [1,4]

The motion parameters are as follows:

- Type of movement (PTP, LIN, CIR)
- Endpoint Name
- Select the progress of the program
- Motion data

13.9 PTP movement

Moving from point to point in the abbreviation PTP is the fastest type of movement from the current TCP position to the programmed point position of the point. The control system recalculates the fastest and most efficient path between the current point P1 and the target point P2 (Fig. 13.44). As a rule, this track is not the shortest runway (it is not a straight line) and the exact course of movement cannot be determined in advance.

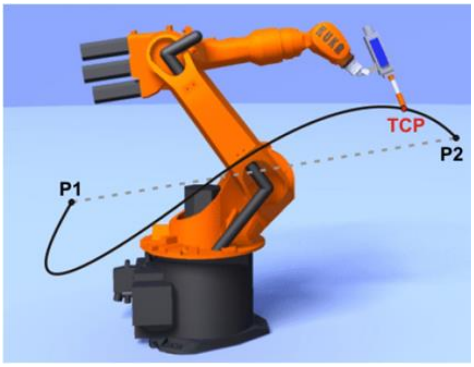


Fig. 13.44 PTP movement [7]

Approximation of PTP movement

Track approximation (rounding) is the most noticeable parameter that significantly affects the course of the track.

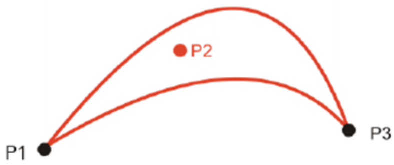


Fig. 13.45 PTP approximation

When the PTP movement is approximated, TCP does not start directly at the target point, but continues to move along a faster path. Point P1 is the starting point of the track, the P2 point being approximated by bypassing it and continuing the fastest path to the target point P3 (Fig. 13.45). The course cannot be determined in advance and by changing the parameters of approximation it is possible to bypass the approximate point P2 from different positions.

Programming the movement of PTP in an inline form

Inline forms are used for user programming to write a motion programming command. In the inline form, the movement parameters are entered in the appropriate fields (Fig. 13.46).

Fig. 13.46 Inline form PTP movement

The parameters entered in the inline form for PTP movement are as follows:

1. Type of movement
    - PTP (LIN, CIR)
2. Endpoint name (when you move the cursor to this box, the Frames status window is called up)

3. Choose how you want to move
    - CONT (target point is approximated)
    - Blank box (stop at destination)
4. Speed (1 to 100%)
5. Name of the motion data set (when moving the cursor to this field, the Motion parameter status window is called up).

Procedure for programming the movement of PTP (Fig. 13.47):

1. The program is selected
2. Selected manual operating mode T1
3. We set the cursor to the line in which the movement command is to be inserted using the *Sentence selection* option in the software bar.
4. With TCP, we run into the position of the point to be programmed as the target point
5. In the main menu bar, select *Commands* → *Motion* → *PTP*, or in the software menu bar, directly select the *Motion option*.
6. An inline form appears in which we input motion parameters
7. Movement in the individual fields of the inline form is possible by a short press of the cursor key
8. Use the Enter button to confirm the programmed point

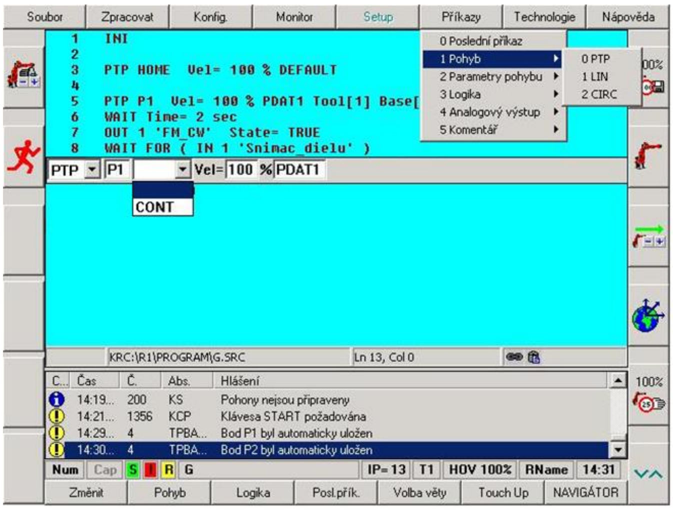


Fig. 13.47 PTP motion programming

13.10 LIN movement

LIN motion is a linear type of linear motion from the current TCP position to the programmed target position of a point. In this movement, the runway is bounded by the starting point P1 and the target point P2 (Fig.13.48). This path is the shortest in terms of distance between programmed points, but in this type of movement the most stressed engines are. The maximum speed is limited to 2 m/s.

LIN motion is a linear type of linear motion from the current TCP position to the programmed target position of a point. In this movement, the runway is bounded by the starting point P1 and the target point P2 (Fig. 13.48). This path is the shortest in terms of distance between programmed points, but in this type of movement the most stressed engines are. The maximum speed is limited to 2 m/s.

Fig. 13.48 Movement of LIN

Approximation of movement LIN

When the LIN movement is approximated, TCP does not start directly at the destination point, but continues to move along a shorter path. Point P1 is the starting point of the track where point P2 is approximated, thus bypassing it, and continuing the shortest path to the P3 destination point (Fig. 13.49). When programming LIN movement, the distance from the destination points at which the TCP can deviate from the original programmed path shall be entered. During the path in the region of approximation, the shape of the resulting arc is non-circular in shape.

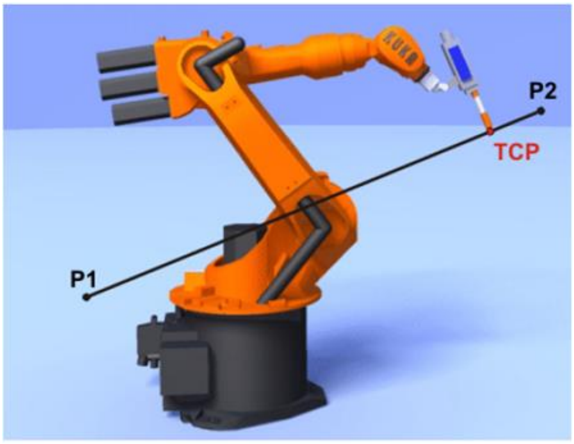


Fig. 13.49 LIN approximation

LIN motion programming in inline form

Fig. 13.50 Inline form LIN movement

The parameters entered in the LIN movement inline form are as follows:

1. Type of movement
   - LIN (PTP, CIR)
2. Endpoint name (when cursor enters this box, the *Frames* status window is invoked)
3. Choose how you want to move
   - CONT (target point is approximated)
   - Blank box (stop at destination)
4. Speed (0.001 to 2 m/s)
5. Name of the motion data set (when the cursor enters this field, the status window *Motion Parameter* is *invoked*)

Procedure for programming the movement of LIN:

1. The program is selected
2. Selected manual operating mode T1
3. We set the cursor to the line in which the movement command is to be inserted using the *Sentence selection* option in the software bar.
4. With TCP, we run into the position of the point to be programmed as the target point
5. In the main menu bar, select *Commands → Motion → LIN* or in the software menu bar, select the *Motion option* directly
6. An inline form appears in which we input motion parameters
7. You can move through individual inline form fields by pressing the arrow key briefly
8. Use the *Enter button* to confirm the programmed point

13.11 Movement of the CIR

CIR motion is a circular type of semicircle movement from the current TCP position along the circular path to the programmed target point position. The circular path is defined by the starting point, the HP auxiliary point and the ZP target point (Fig. 13.51).

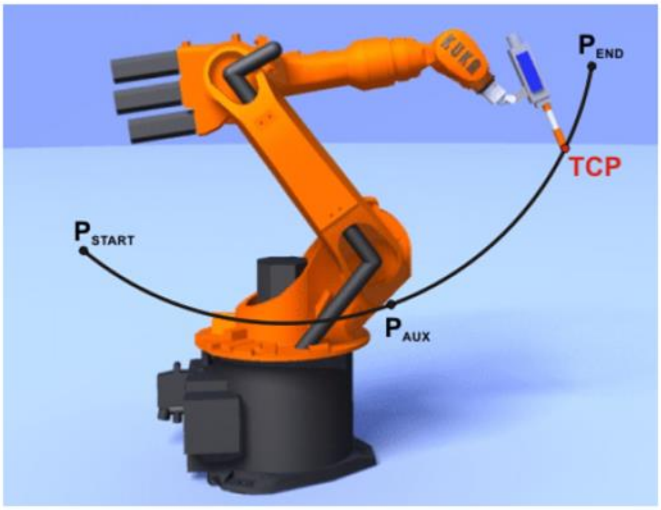


Fig. 13.51 Cir movement

Approximation of CIR movement

As the CIR approximates, TCP leaves the programmed path at which it would otherwise run to the destination point and continues along the shortest path (Fig. 13.52). The HP auxiliary point (P_{AUX}) is always approached accurately, and the approximation is performed only at the destination point ZP (P_{END} When programming a CIR motion, you enter the distance from the target point at which TCP can deviate from the original programmed path. During the path in the region of approximation, the shape of the resulting arc is non-circular in shape.

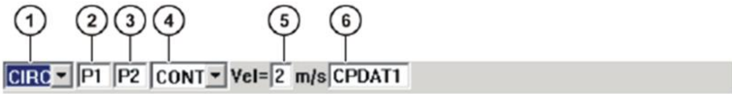


Fig. 13.52 CIR approximation

Programming the movement of the CIR in the inline form

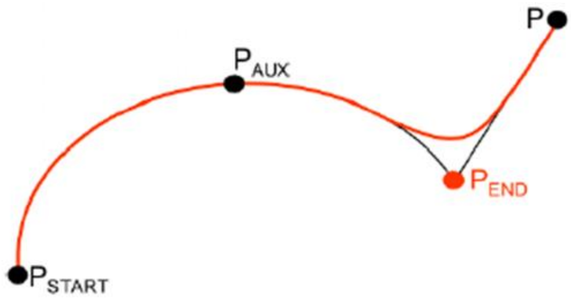


Fig. 13.53 Inline form CIR movement

The parameters entered in the inline form for CIR movement are as follows:

1. Type of movement
   - CIR (PTP, LIN)
2. Name of the auxiliary point
3. Endpoint name (when moving the cursor to this field, the Frames status window is called up)
4. Choose how you want to move
   - CONT (target point is approximated)
   - Blank box (stop at destination)
5. Speed (0.001 – up to 2 m/s)
6. Name of the motion data file (when moving the cursor to this field, the *Motion parameter* status window is called up).

CIR movement programming procedure (Fig. 13.54):

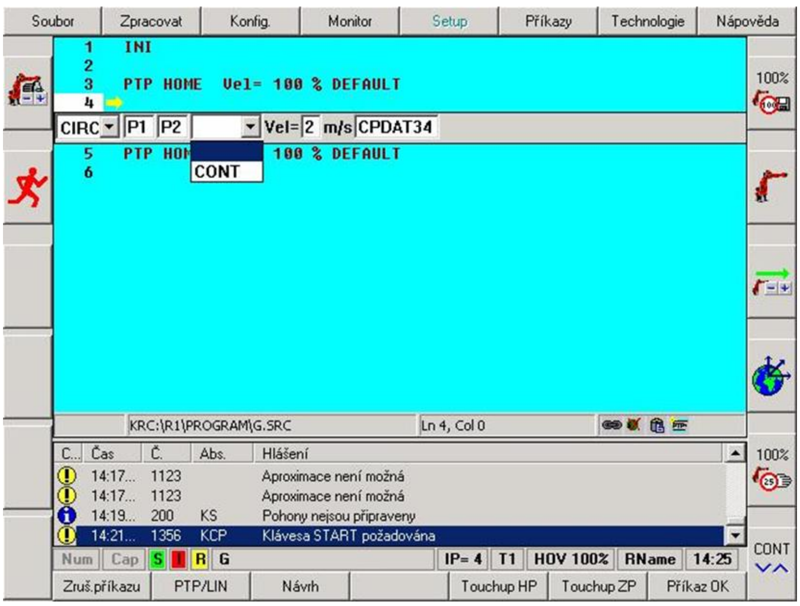


Fig. 13.54 CIR motion programming

1. The program is selected
2. Selected manual operating mode T1
3. Set the cursor to the line in which the movement command is to be inserted using the *Sentence selection* option in the software bar.

4. With TCP, we run into the position of the point to be programmed as an HP auxiliary point
5. In the main menu bar, select *Commands* → *Motion* → *CIRC* or in the software menu bar, select the *Motion* option directly
6. An inline form appears in which we input motion parameters
7. In the software bar, we choose to coordinate the HP *Touchup* help point
8. With TCP we run into the position of the point to be programmed as the target point of the ZP
9. In the software bar we choose to coordinate the target point Touchup *ZP*
10. You can move through individual inline form fields by pressing the arrow key briefly
11. Use the Enter button to confirm the programmed point

Conclusion

People, machines, devices, logistic systems and products can mutually directly communicate and cooperate – this is the principle Industry 4.0 operates. Everything advances towards total networking. The reason is represented by a huge amount of so far elusive information for fast and correct decision taking. Close connection of products, devices and people increases the effectiveness of manufacturing machines and devices, lowers costs and saves sources. Intelligent monitoring and transparent processes provide the companies with constant overview allowing thus for flexible and fast reaction to changes on the market. The companies move to so called intelligent (Smart) factories. The result of an intelligent factory is an intelligent product which is not overpriced and corresponds to customer´s individual needs. In addition, it is profitable for the manufacturer.

1. Vertical interconnection of intelligent manufacturing systems such as intelligent factories and intelligent products, and interconnection of e.g. intelligent logistics, production and marketing and intelligent services with strong orientation on the needs, and individual and specific possibilities of a customer.
2. Horizontal integration via a new generation of global networks forming thus an added values including the integration of business partners and customers, new models of entrepreneurship, and cooperation across countries and continents.
3. Application of technology throughout the whole value chain, not only within the manufacturing process but also with the ready product – i.e. within the whole life cycle of the product.
4. Acceleration via exponential technologies, which do not have to be really new from the point of view of their history development, however, they become capable of their mass implementation on the market now since their price rapidly decreases (e.g. with various sensors) and their productivity significantly improves.

Authors think that the textbook can help the employees in industry to obtain knowledge as well as it can help technical university students to complement their knowledge in various expert fields.

The publication is focused on the field of programming and production and manipulation technology. The textbook is primarily determined for foreign students studying at technical universities. The textbook comprises also a sample of programming procedure for production and manipulation technology together with related control processes. These mentioned programs have been practically tested. The textbook represents a thorough background for successive acquisition of knowledge in CNC manufacturing machines programming, especially manipulation technology as illustrated in the publication.

ACKNOWLEDGEMENT

The contribution is sponsored by the project VEGA 1/0488/23 „Research into the influence of new technologies of composite materials on the design parameters of technological equipment using experimental and computer methods".

LITERATURE

[1] Valentino, J.V., Goldenberg, J.: Introduction to Computer Numerical Control, 4th edition, 2008, Pearson, Prentice Hall, ISBN: 0-13-243690-6

[2] Gibbs, D., Crandell, T.M.: An Introduction to CNC machining and programming, Industrial Press Inc., 1991, ISBN 0-8311-3009-1

[3] Krar, S. F., Gill, A. R., Smid, P.: Technology of Machine Tools, 6th edition, Mc Graw Hill Higher Education, 2005, ISBN 978-0-07-830722-5

[4] Nagy, F. N, Siegler, A.: Engineering Foundations of Robotics, Prentice Hall, 1987, ISBN 0-13-278797-0

[5] Gourdeau, R.: Object oriented programming for robotic manipulator simulation. IEEE ROBOTICS & AUTOMATION MAGAZINE, Vol. 4, Iss. 3, pp. 21-29, DOI10.1109/100.618020

[6] https://www.emco-world.com/uploads/tx_commerce/Sinumerik810820Turn_en.pdf, 1.12.2017

[7] http://www.sjf.tuke.sk/vitralab/files/vystupy/handbook_en_final.pdf, 15.12.2017

[8] https://www.machiningcloud.com/wp-content/uploads/2016/05/MachiningCloud_TurningToolsAndTheirApplication.pdf , 15.12.2017

[9] HEIDENHAIN|TNC 640|Conversational Programming User's Manual|10/2017. Original document 892903-26 · Ver06 · SW08 · 10/2017 · H · Printed in Germany https://content.heidenhain.de/doku/tnc_guide/pdf_files/TNC640/34059x-08/bhb/892903-26.pdf (04.12.2018)

[10] Michalík, P., Duplák, J., Kormoš, M.: Číslicovo riadené obrábacie stroje a ich programovanie. Prešov 2016, 24 p. ISBN: 978-80-553-1837-0.

[11] Kaczmarek, W., Panasiuk, J.: Programming of robots of selected companies. Programowanie robotów przemysłowych, pp. 60-269, 2017.

[12] Michalík, P., Zajac, J.: Intelligently programming of holes machining. In: Výrobné inžinierstvo. Roč. 9, č. 4 (2010), pp. 63-65. ISSN 1335-7972

[13] Králik, M., Semjon, J., Božek, P.: Industrial robots and manipulators in the automotive industry. 1. Wyd. STE GROUP sp. z o. o. Warszawa 2023. 145 p. ISBN 978-83-952420-9-0.

[14] Králik, M., Panda, A.: Manufacturing Technologies. Theory and Experiments of Machining. RAM – Verlag, 2023. ISBN 978-3-96595-032-0. https://ram-verlag.org/volumes, https://ram-verlag.org/wp-content/uploads/vol-11-032-0_manufacturing-technologies.pdf

[15] Králik, M., Biały, W., Michalík, P., Božek, P., Morávek, I.: Programovanie CNC Systémov (Programowanie Systemów CNC), Gliwice 2023, pp. 1-164, ISBN 978-83-65593-35-1, https://doi.org/10.32056/KOMAG/Monograph2023.3 , https://www.komag.eu/wydawnictwa/monografie/cyfrowa-gospodarka

[16] Králik, M., Morávek, I.: Programovanie CNC výrobných strojov. Slovenská technická univerzita, vyd. FX s.r.o. Bratislava 2008, ISBN: 978-80-89313-36-5. 141 s.

[17] Softwarebeschreibung EMCO WinNC Sinumerik 810/820 T. EMCO MAIER GmbH., Hallein 1996

[18] Matejíčka, Ľ.: Sústruh EMCO PC Turn 50. Obsluha a programovanie. SPŠ Martin, Centrum nových technológií, 1995

[19] Zielinski, C.: Object-oriented robot programming. Volume 15, pp. 41-48, Part 1, DOI10.1017/S0263574797000064

[20] Rybín, P., Rafaj, J.: Obsluha a programování CNC strojů. Vydavatelství ČVUT, 1995.

[21] Marek, J.: Konstrukce CNC obráběcích strojů. MM Průmyslové sprektrum. Speciální vydání. Září 2006, CZK 99, ISSN 1212-2572

[22] Michalik, P.: Vytvorenie riadiaceho programu pre CNC sústruh v prostredí CATIA V5. 2015. In: IT strojár. Č. 1(2015), pp. 1-5. ISSN 1338-0761.

[23] Krsek, A., Bachratý, M. Základy kontroly kvality geometrických parametrov výrobkov. Bratislava: FX s. r. o., 2008. 155 p. ISBN 978-89313-37-2.

[24] Krsek, A. Ossana, P. H. Kuric, I. Prostredník, D. Strojárska metrológia a riadenie kvality. Bratislava: Vydavateľstvo STU, 1998. 290 p. ISBN: 80-227-1025-3.
[25] STN EN ISO 10360-1:2000: Geometrické špecifikácie výrobkov (GPS). Akceptačné a verifikačné skúšky súradnicových meracích strojov (CMM). Časť 1: Slovník.
[26] WILCOX ASSOCIATES, HEXAGON METROLOGY. Reference manual PC-DMIS 4.3. 2007.
[27] Buda, J., Kováč, M.: Priemyselné roboty Bratislava: Alfa, 1977, 254 p.
[28] Michalik, P., Zajac, J., Duplak, J., Pivovarník, A.: CAM Software Products for Creation of Program for CNC Machining. Future Communication, Computing, Control and Management: Volume 1: LNEE 141. Berlin Heidelberg: Springer-Verlag, 2012 pp. 421-425. ISBN 978-3-642-27310-0. ISSN 1876-1100.
[29] Yeung, M.K.: Intelligent process planning system for optimal CNC programming – a step towards complete automation of CNC programming. ICMA 2002 INTERNATIONAL CONFERENCE ON MANUFACTURING AUTOMATION, pp. 169-177, Published 2002.
[30] Tirpak, P., Michalik, P., Macej, J.: Computer-Aided Manufacturing of a Shaft for Wood Chipper using Autodesk Inventor 2020/2021. In: SAR Journal – Science and Research. Novi Pazar (Srbsko): Association for Information Communication Technology Education and Science Roč. 4, č. 2 (2021), pp. 47-51 [print, online]. ISSN 2619-9955.
[31] Widl, G.: Robots programmed for expansion. UMSCHAU DAS WISSENSCHAFTSMAGAZIN Volume 82, Iss. 3, pp. 84-88. Published 1982

2024