



Ahsanullah University of Science and Technology (AUST)
Department of Mechanical and Production Engineering

LABORATORY MANUAL

For the students of

Department of Mechanical and Production Engineering
3rd Year, 2nd Semester

Student Name :
Student ID :

**Department of Mechanical and Production Engineering
Ahsanullah University of Science and Technology (AUST)**

IPE 3212: Computer Aided Manufacturing and Robotics Sessional

Credit Hour: 0.75

List of Experiments:

1. Study of Industrial Robot Fundamentals, Forward Kinematics, and Motion Planning Using Denavit–Hartenberg (DH) Parameters
2. CNC Toolpath Generation Using SolidCam
3. Study of GT and Part Family Formation Using the OPITZ Coding System
4. Study of CNC Programming and Virtual Simulation of Machining Operations

Total Marks		
Assignment	Attendance	Quiz
40	10	50

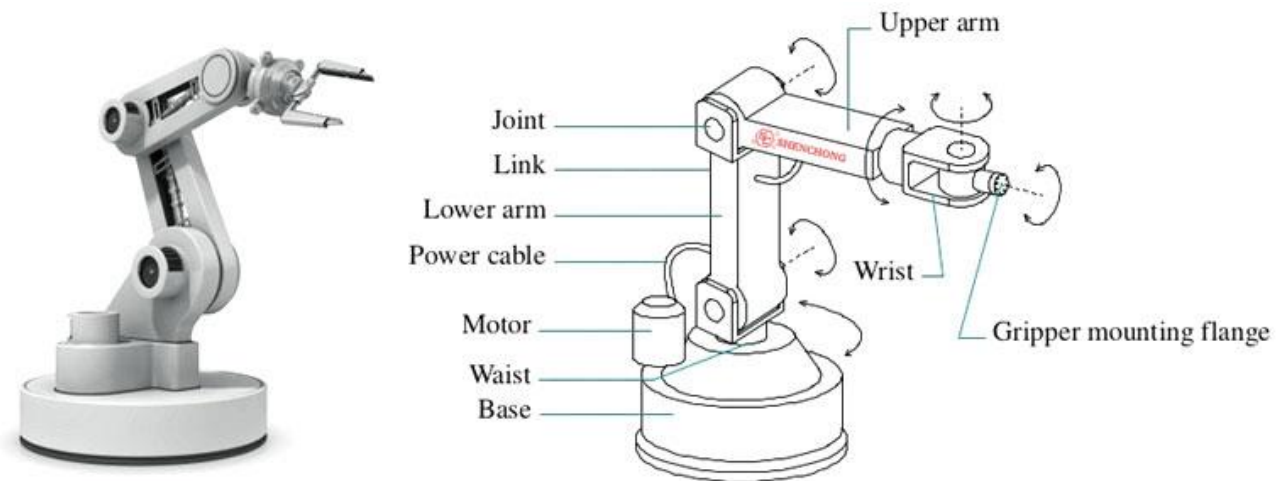
EXPERIMENT 1

Study of Industrial Robot Fundamentals, Forward Kinematics, and Motion Planning Using Denavit–Hartenberg (DH) Parameters

Objective: To understand robot coordinate systems and perform a basic pick-and-place operation.

Equipment

- Dobot Magician / Dobot Magician E6
- Gripper



Structure of a Robot Manipulator:

Figure 1.1: Components of a Robot Manipulator

A typical robot consists of:

- **Base:** Fixed part that supports the robot
- **Links:** Rigid bodies connecting joints
- **Joints:** Provide motion (rotational or prismatic)
- **End-effector:** Tool attached to the robot (e.g., gripper)

Degrees of Freedom (DOF): The number of independent motions a robot can perform is called its degrees of freedom (DOF).

- Each joint contributes one DOF

- A higher DOF allows greater flexibility in positioning and orientation

Example:

- 4-axis robot → limited orientation control
- 6-axis robot → full positioning + orientation capability (similar to industrial robots)

Coordinate Systems in Robotics: Robot motion is defined using coordinate systems

a) Joint Coordinate System: Motion is defined by joint variables (angles or displacements). Each axis moves independently and motion path is not necessarily straight

b) Cartesian Coordinate System: Motion defined in X, Y, Z space, end-effector follows a straight-line path and easier for tasks like pick-and-place

Robot Workspace: The workspace is the total volume that the robot's end-effector can reach. It depends on link lengths, joint limits and robot configuration. Workspace is important because objects must lie within reachable area and improper placement leads to failure of operation

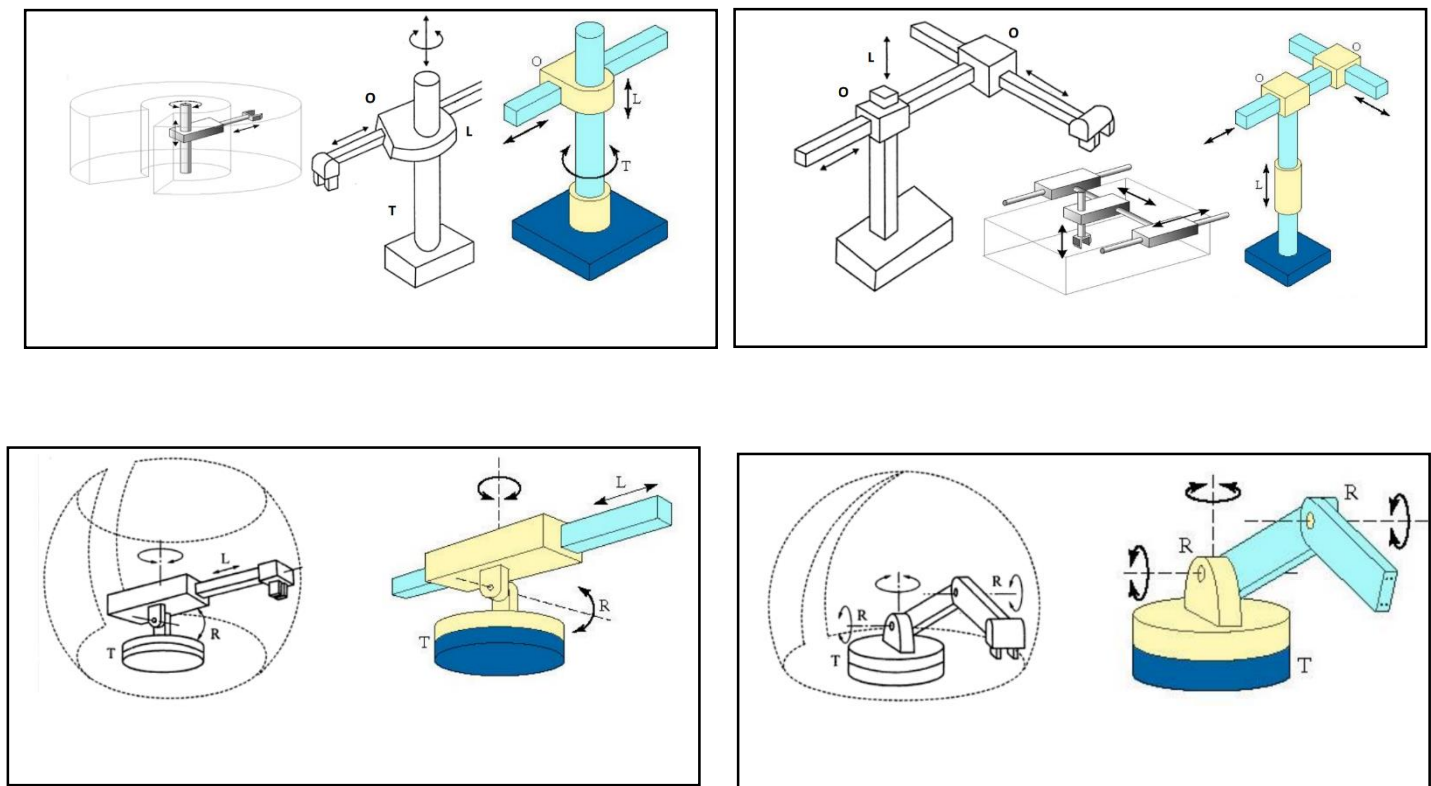


Figure 1.2: Workspace of different configurations of robots

Point Teaching, Pick-and-Place Operation and Programming: Industrial robots are often programmed using teach-and-repeat method. This method is widely used because it is simple to implement and requires minimal coding

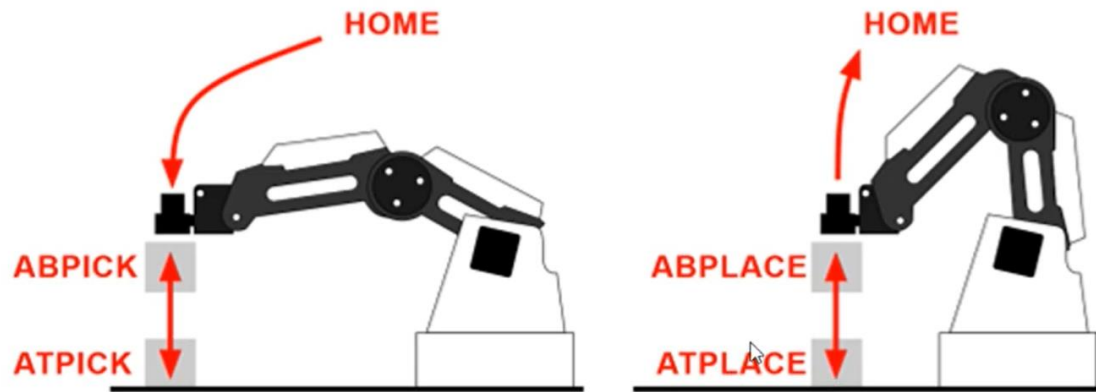


Figure 1.3: Pick and Place operation using a robotic arm

General Steps:

1. Move robot manually to a position
2. Save the position (teaching point)
3. Repeat for multiple points
4. Execute sequence automatically

Pick-and-place is a basic robotic task involving:

1. Moving to object location
2. Gripping the object
3. Lifting and transporting
4. Releasing at target location

It is widely used in assembly lines, packaging systems, material handling

Theory: Denavit–Hartenberg (DH) Parameters for Robot Kinematics

In robotics, **kinematics** deals with the relationship between joint variables and the position and orientation of the robot's end-effector. For multi-link manipulators, this relationship can become complex. The Denavit–Hartenberg (DH) convention provides a systematic method to model robot geometry and derive transformation equations.

Purpose of DH Parameters: The DH method simplifies kinematic analysis by

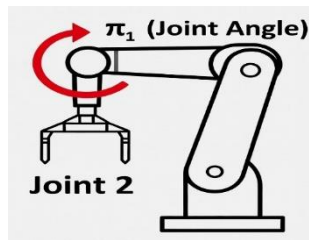
- standardizing coordinate frame assignment
- reducing complex geometry into a set of parameters
- enabling matrix-based computation of position and orientation

Using DH parameters, the pose of the end-effector can be calculated through a sequence of coordinate transformations.

DH Parameters: Each link of a robot is described using four parameters

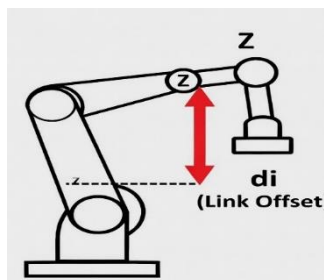
1. **Joint angle (θ_i)**

Rotation about the Z-axis
(variable for revolute joints)



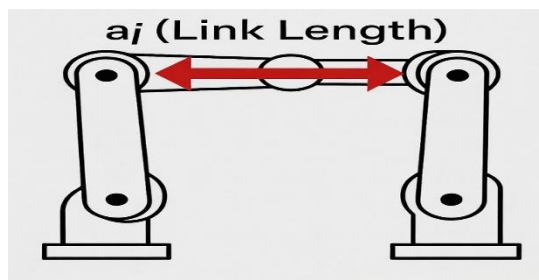
2. **Link offset (d_i)**

Distance along the Z-axis



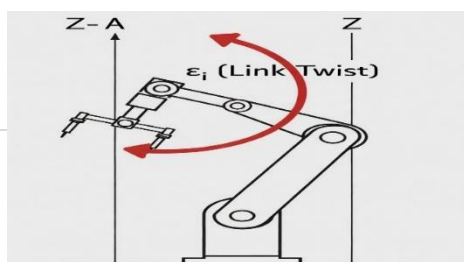
3. **Link length (a_i)**

Distance along the X-axis between two joints



4. **Link twist (α_i)**

Angle between two Z-axes about the X-axis



These parameters define the relative position between two consecutive links.

To apply the DH method, coordinate frames are assigned following specific rules:

- Z-axis is aligned with the axis of motion of each joint
- X-axis is perpendicular to both adjacent Z-axes
- Origins are placed at joint intersections

This consistent frame assignment ensures correct transformation between links. The relationship between two consecutive links is expressed using a homogeneous transformation matrix:

$$T_i^{i-1} = \begin{bmatrix} \cos \theta_i & -\sin \theta_i \cos \alpha_i & \sin \theta_i \sin \alpha_i & a_i \cos \theta_i \\ \sin \theta_i & \cos \theta_i \cos \alpha_i & -\cos \theta_i \sin \alpha_i & a_i \sin \theta_i \\ 0 & \sin \alpha_i & \cos \alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

This matrix includes both:

- rotation (orientation)
- translation (position)

Overall Transformation

The position and orientation of the end-effector with respect to the base frame is obtained by multiplying all individual transformations:

$$T = T_1 T_2 T_3 \dots T_N$$

The final matrix gives:

$$T = \begin{bmatrix} r_{11} & r_{12} & r_{13} & x \\ r_{21} & r_{22} & r_{23} & y \\ r_{31} & r_{32} & r_{33} & z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- position (x, y, z) from the last column
- orientation from the rotation sub-matrix

Cycle Time and Productivity

Cycle time = time required to complete one operation

$$\text{Production Rate} = \frac{3600}{\text{Cycle time (seconds)}}$$

Lower cycle time → higher productivity. Factors affecting cycle time: speed and acceleration, path planning, number of intermediate points, efficiency of program.

Experimental Procedure

1. Open the DobotStudio software and connect to the robot
2. Click on “Home” option
3. Move the robot arm manually to Above Pick point
4. Move the robot arm to At Pick point and turn on the End Effector
5. Values are recorded in the software every time the arm is moved
6. Move the arm to Above Pick point along with the cubic work pieces
7. Move or rotate arm to Above Place point and then to At Place point.
8. Turn off End Effector while arm is at “At Place” point
9. Move arm to Above Place point
10. Click on Home button
11. Press Start to initiate the program
12. Using stopwatch record the cycle time
13. Repeat for 3-5 trials.

Data Table:

Trial	Above Pick (X,Y,Z,R)	At Pick (X,Y,Z,R)	Above Place (X,Y,Z,R)	At Place (X,Y,Z,R)	Cycle Time (seconds)
1					
2					
3					
4					
5					

Simplified DH Model

Link	θ (degrees)	d (mm)	a (mm)	α (degrees)
1				
2				
3				
4				
5				
6				

Assignment

1. Explain the difference between Joint space and Cartesian space motion.
2. Draw the robot workspace used in this experiment showing pick and place positions.
3. Why is a safe point necessary in robot programming?
4. If one cycle takes 6 seconds, calculate output per hour.

EXPERIMENT 2

CNC Toolpath Generation Using SolidCam

Objective: The objective of this experiment is to generate CNC toolpaths from a given CAD geometry using CAM software, simulate the machining process, and analyze the relationship between CAM-generated toolpaths and CNC programs for both milling and turning operations.

Equipment:

- CAM software
- CNC simulator

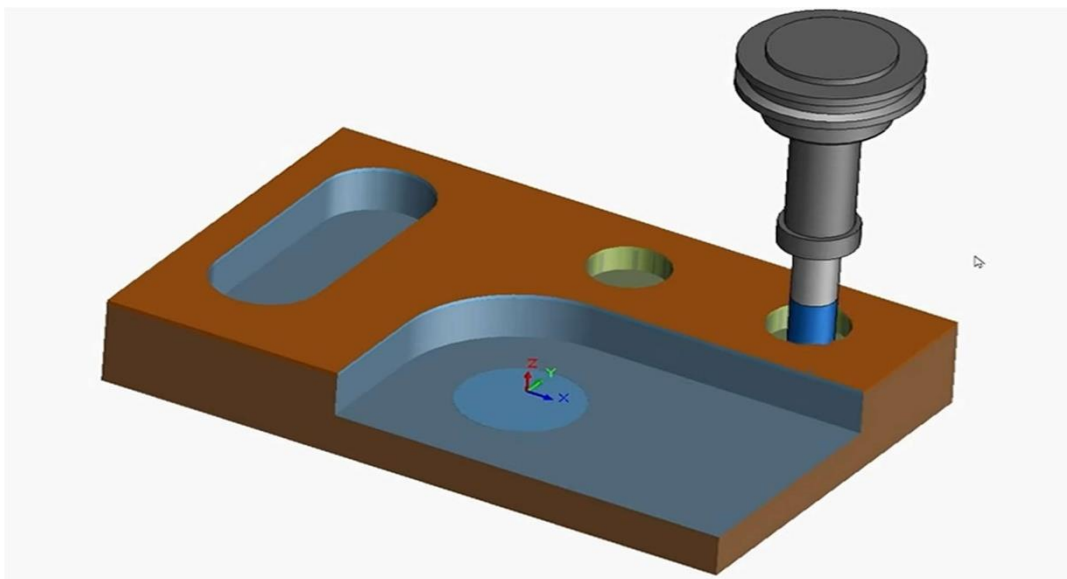
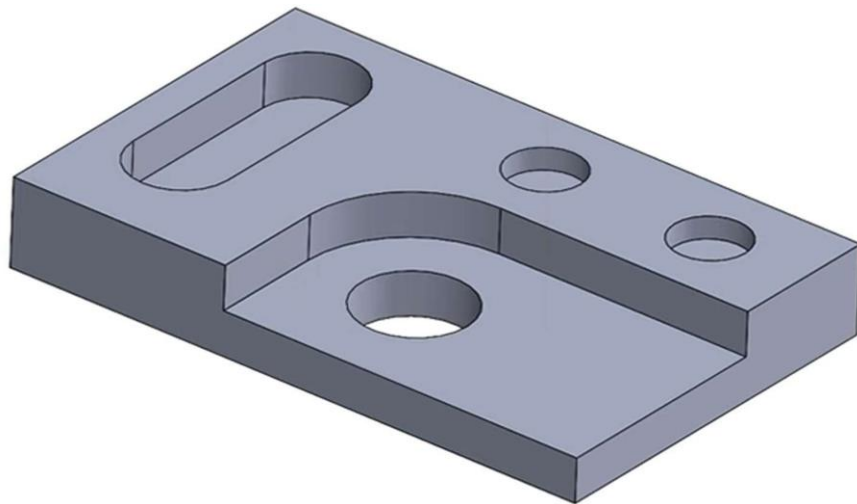
Theory:

Computer Aided Manufacturing (CAM) is an essential component of modern manufacturing systems that enables the automatic generation of machining instructions from digital part models. Unlike manual CNC programming, where each instruction is explicitly written by the programmer, CAM systems generate toolpaths based on the geometry of the part and selected machining parameters. In a CAM environment, the user begins by defining the machining setup, including the work coordinate system, stock geometry, and cutting tools. Based on the geometry of the part, appropriate machining operations such as contouring, pocketing, drilling, and surface machining are selected. The software then calculates the toolpath required to remove material efficiently while maintaining dimensional accuracy and desired surface finish.

The generated toolpath consists of a sequence of tool movements that follow the geometry of the part. These movements are internally represented and later converted into machine-specific G-code through a process known as post-processing. Understanding this relationship between the graphical toolpath and the underlying CNC code is critical for debugging, optimization, and effective machining. Simulation is a key feature of CAM systems. It allows visualization of tool motion, material removal, and potential collisions before actual machining. By analyzing the simulation, engineers can optimize toolpath strategies, adjust cutting parameters, and ensure that the machining process is safe and efficient. This experiment focuses on the generation, simulation, and analysis of CNC toolpaths using CAM software. Students are expected to evaluate different machining strategies, compare CAM-generated toolpaths with manually written programs, and understand how CAM contributes to productivity and accuracy in computer integrated manufacturing systems

Procedure

1. Study the given component drawing or CAD model and identify the machining features such as pockets, contours, holes, or profiles.
2. Import or create the part geometry in CAM software.
3. Define the stock material dimensions and set the appropriate work coordinate system.
4. Select suitable cutting tools based on the geometry and machining requirements.
5. Choose appropriate machining operations such as contouring, pocketing, drilling, or turning.
6. Define machining parameters including depth of cut, step-over, feed rate, and spindle speed.
7. Generate the toolpath using the selected machining strategy.
8. Simulate the machining process and observe tool movement, material removal, and possible collisions.
9. Post-process the toolpath to generate CNC code.
10. Analyze the generated CNC program and compare it with manually written code where applicable.



Observations

- Toolpath
- Machining sequence

Result

Successful CAM-based toolpath generation

Learning Outcome

Upon completion of this experiment, students will be able to generate toolpaths using CAM software, define machining parameters, simulate machining processes, and interpret the relationship between CAD models, toolpaths, and CNC programs. They will also develop the ability to evaluate and optimize machining strategies for improved performance.

Assignment:

1. Difference between manual and CAM programming
2. Factors affecting toolpath
3. Importance of simulation
4. Difference between contour and pocket

EXPERIMENT 3

Study of GT and Part Family Formation Using the OPITZ Coding System

Objective

The objective of this experiment is to classify mechanical components into part families using Group Technology principles and to develop a structured manufacturing process plan for selected components using Computer Aided Process Planning concepts.

Theory

Group Technology (GT) is a manufacturing philosophy that groups similar parts into families based on similarities in design, geometry, manufacturing processes, and production requirements. The main objective of GT is to improve production efficiency by identifying and classifying parts with comparable characteristics so that similar manufacturing methods, tooling, and machine setups can be used. Part classification and coding systems play a vital role in GT by assigning a structured code to each component according to its physical and functional features. One widely used coding method is the Opitz Coding System, which classifies parts based on geometry, dimensions, and manufacturing attributes. In this experiment, the selected component is analyzed and assigned an Opitz form code to understand the principles of GT classification and the grouping of similar manufactured parts.

OPITZ Classification System

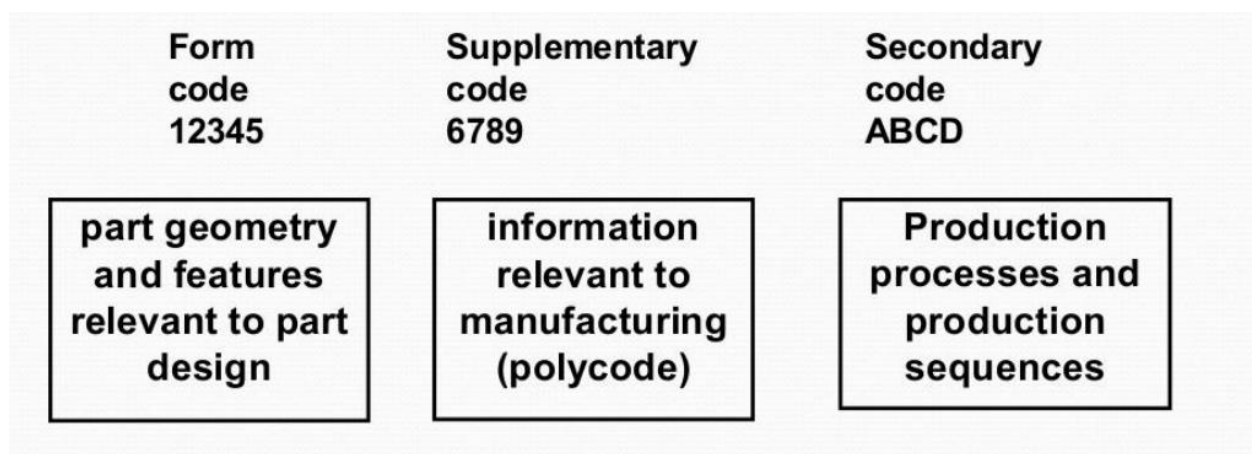


Figure 3.1: Different coding groups in OPTIZ classification system

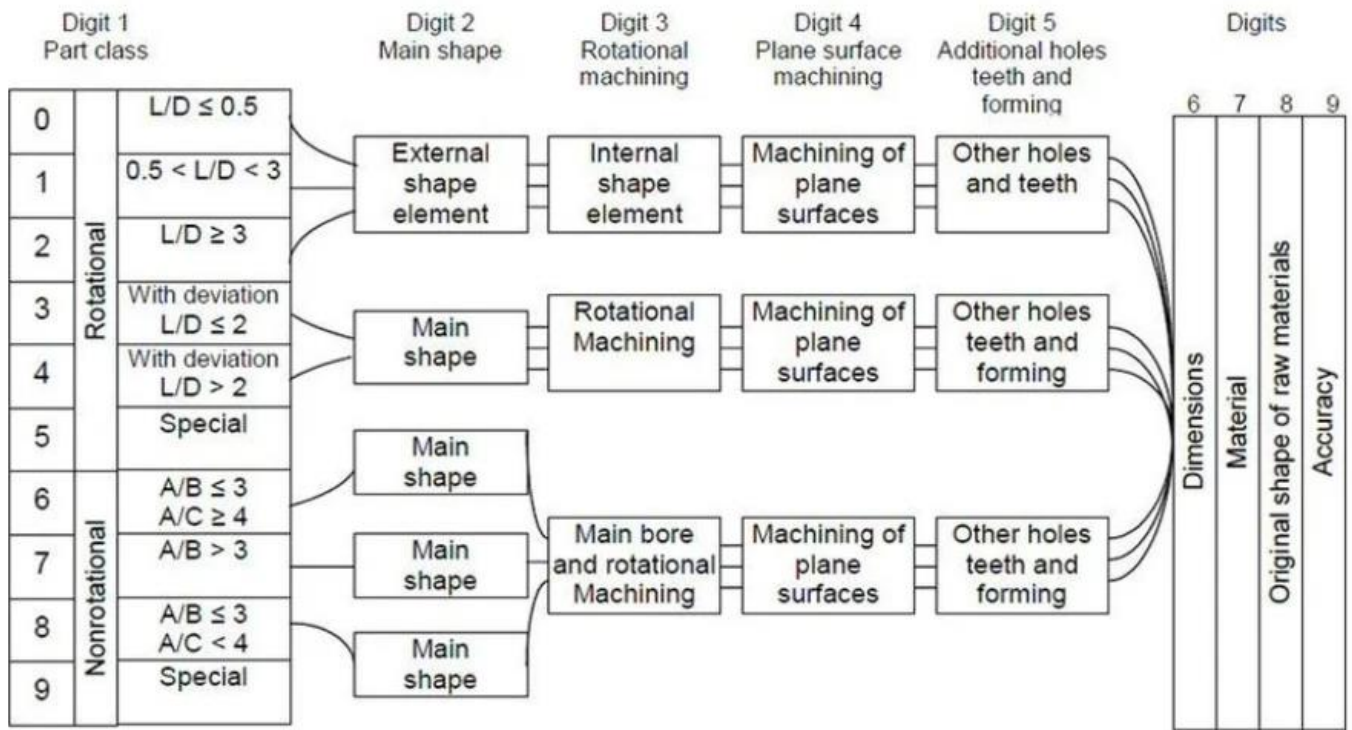


Figure 3.2: OPITZ form code system

Digit 1			Digit 2			Digit 3			Digit 4			Digit 5		
Part class			External shape, external shape elements			Internal shape, internal shape elements			Plane surface machining			Auxiliary holes and gear teeth		
0	Rotational parts	$L/D \leq 0.5$	0	Smooth, no shape elements		0	No hole, no breakthrough		0	No surface machining		0	No auxiliary hole	
1		$0.5 < L/D < 3$	1	Stepped to one end or smooth	No shape elements	1	Smooth or stepped to one end	No shape elements	1	Surface plane and/or curved in one direction, external		1	Axial, not on pitch circle diameter	
2		$L/D \geq 3$	2		Thread	2		Thread	2	External plane surface related by graduation around the circle		2	Axial on pitch circle diameter	
3			3		Functional groove	3		Functional groove	3	External groove and/or slot		3	Radial, not on pitch circle diameter	
4			4	Stepped to both ends	No shape elements	4	Stepped to both ends	No shape elements	4	External spline (polygon)		4	Axial and/or radial and/or other direction	
5		5	Thread		5	Thread		5	External plane surface and/or slot, external spline		5	Axial and/or radial on PCD and/or other directions		
6		6	Functional groove		6	Functional groove		6	Internal plane surface and/or slot		6	Spur gear teeth		
7	Nonrotational parts		7	Functional cone		7	Functional cone		7	Internal spline (polygon)		7	Bevel gear teeth	
8			8	Operating thread		8	Operating thread		8	Internal and external polygon, groove and/or slot		8	Other gear teeth	
9			9	All others		9	All others		9	All others		9	All others	

Figure 3.3: Form Code (digits 1-5) for rotational parts in Opitz coding system

Problem:

Assign a 5 digit form code to the following part using Opitz coding system

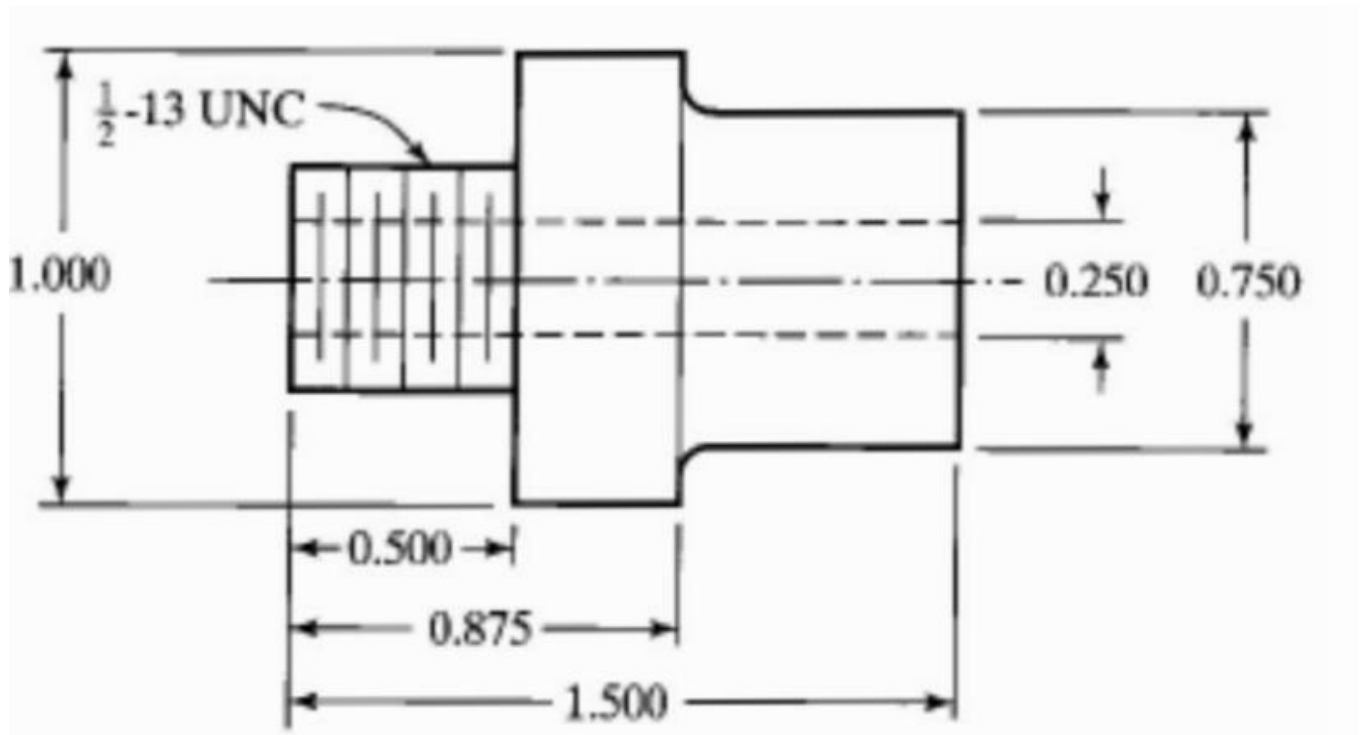


Figure 3.4: Design of a rotational part.

Working Procedure

- Identify the basic shape of the part (rotational, prismatic, sheet metal, etc.)
- Check the largest dimension ratio (length to diameter or width)
- Identify external features such as grooves, shoulders, or threads
- Identify internal features like holes, bores, or cavities
- Check and classify the flat machined surfaces on the part
- Identify and classify holes and other internal machining features of the part

Form Code	Code Interpretation

Report Assignment

Assign a 5 digit form code to the following part using Opitz coding system

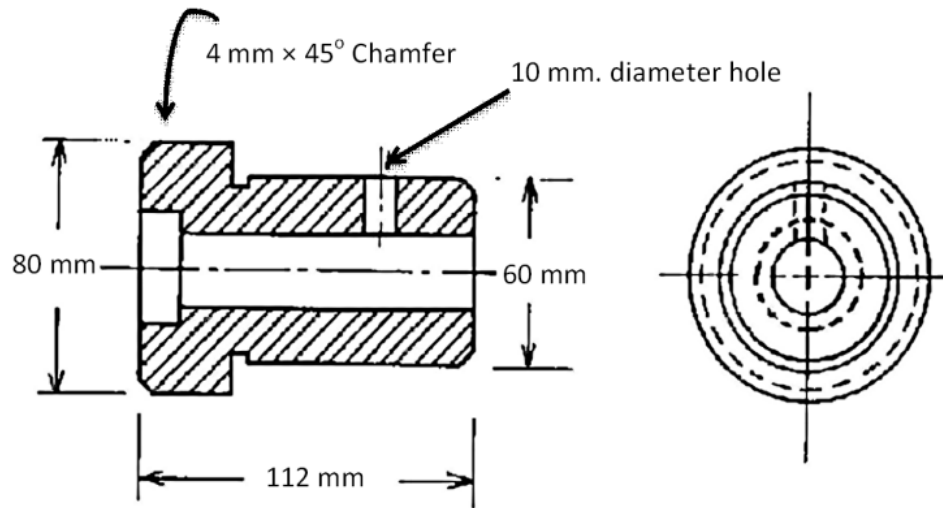


Figure 3.5: Rotational mechanical part

Assignment Questions

1. Explain how Group Technology contributes to improved efficiency in manufacturing systems.
2. What is the purpose of the Opitz coding system?
3. What is meant by a part family in manufacturing?

EXPERIMENT 4

Study of CNC Programming and Virtual Simulation of Machining Operations

Objective: The objective of this experiment is to develop a basic understanding of CNC programming through virtual simulation of real life machining operations.

Software:

- CNC Simulator (Swansoft)

Computer Numerical Control (CNC) machining is based on the precise control of tool motion through a sequence of programmed instructions, typically expressed in G-code and M-code format. In contrast to manual machining, CNC systems rely on a well-defined coordinate system, interpolation algorithms, and motion control strategies to execute complex geometries with high accuracy and repeatability. In manual CNC programming, the programmer directly defines tool motion using absolute or incremental positioning within a selected coordinate system. Linear interpolation (G01) and circular interpolation (G02/G03) form the foundation for generating most machining geometries. Advanced programming also involves the use of cutter radius compensation (G41/G42), canned cycles for drilling operations (e.g., G81–G89), tool length compensation (G43/G44), and coordinate system transformations. The correct selection of feed rate, spindle speed, and depth of cut is essential for achieving optimal machining performance.

The geometric interpretation of a CNC program is governed by the controller's interpolation mechanism, which calculates intermediate points along the tool path. Errors in programming, such as incorrect arc definitions, improper coordinate referencing, or omission of compensation codes, can result in dimensional inaccuracies or tool collisions. Therefore, verification and simulation are critical steps in CNC programming. Modern CNC systems support multi-axis machining and advanced features such as subprograms, parametric programming, and macro variables, which enable flexible and efficient production. However, a strong understanding of manual G-code programming remains fundamental, as it provides insight into toolpath generation, machine behavior, and process optimization.

CNC Programming Language

CNC programs are written in a block-by-block format, where each line contains instructions for the machine. Each block typically consists of:

- Preparatory functions (G-codes)
- Miscellaneous functions (M-codes)
- Coordinates (X, Y, Z)
- Feed rate (F)
- Spindle speed (S)

CNC Coordinate System

CNC machines use a Cartesian coordinate system:

- X-axis → Horizontal movement
- Y-axis → Side movement (mainly in milling machines)
- Z-axis → Vertical or lengthwise movement

In CNC lathe programming:

- U-axis → Incremental movement in the X direction (diameter control)
- W-axis → Incremental movement in the Z direction (length control)

These are used for relative positioning instead of absolute coordinates.

G-Codes (Geometric Codes)

G-codes control the movement of the machine tool. They define how the tool moves in space.

They are responsible for:

- Positioning the tool
- Defining cutting paths
- Specifying motion types (rapid, linear, circular)

Examples:

- G00 → Rapid positioning
- G01 → Linear cutting movement
- G02 / G03 → Circular interpolation
- G90 / G91 → Absolute / Incremental positioning

M-Codes (Miscellaneous Codes)

M-codes control auxiliary machine functions that are not related to motion.

They are used for:

- Spindle control
- Coolant control
- Program start/stop operations

Examples:

- M03 → Spindle ON (clockwise)
- M05 → Spindle STOP

- M08 → Coolant ON
- M09 → Coolant OFF
- M30 → End program and reset

CNC programming is important because it:

- Automates manufacturing processes
- Improves accuracy and repeatability
- Reduces human error
- Enables mass production
- Allows simulation before actual machining

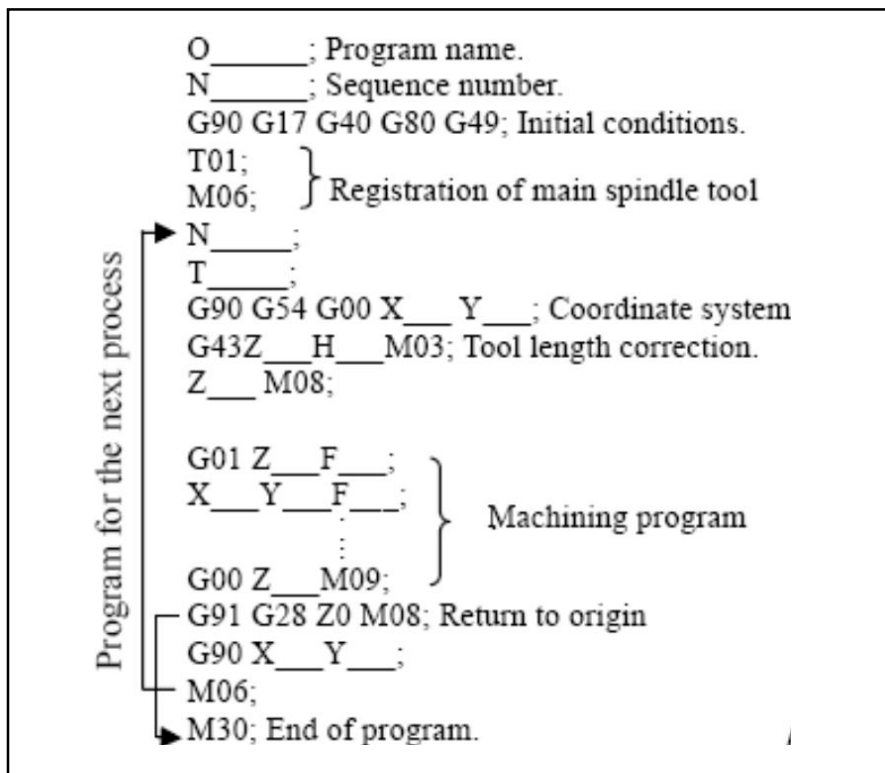


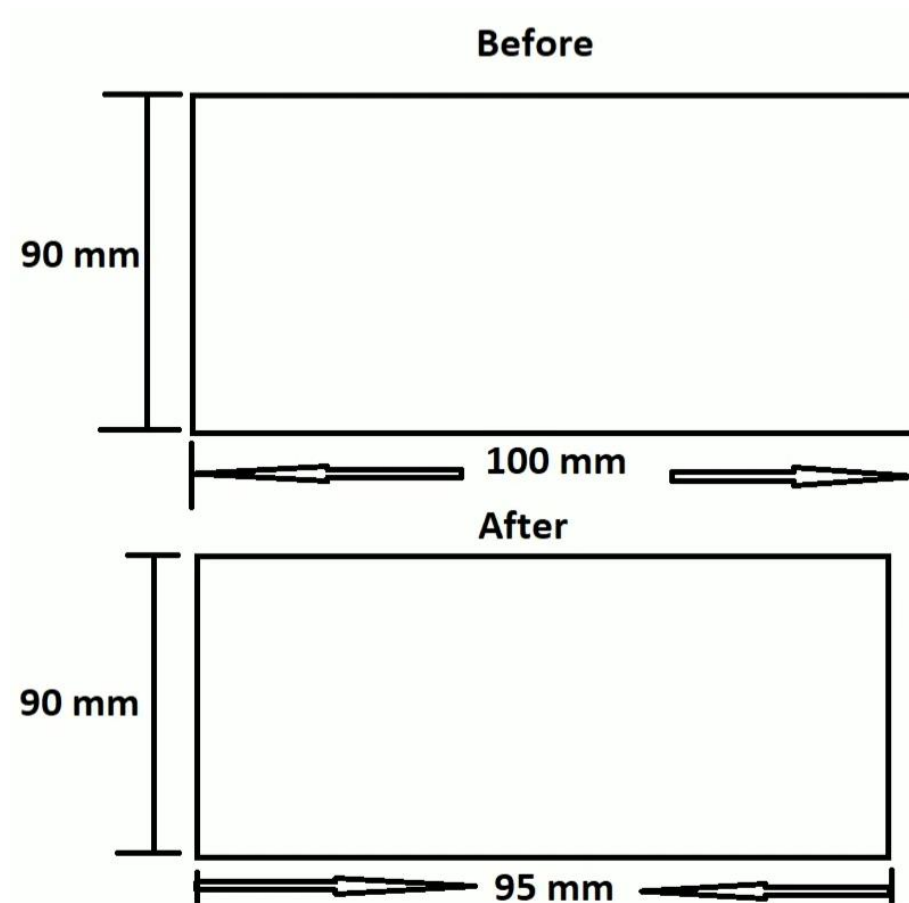
Figure 4.1: Basic structure of CNC coding

Common CNC Codes (Reference List)

Code	Type	Function / Description
G00	Motion	Rapid positioning (non-cutting move)
G01	Motion	Linear interpolation (cutting move)
G02	Motion	Circular interpolation (clockwise)
G03	Motion	Circular interpolation (counterclockwise)
G17	Plane	Select XY plane
G18	Plane	Select ZX plane
G19	Plane	Select YZ plane
G20	Unit	Programming in inches
G21	Unit	Programming in millimeters
G28	Reference	Return to machine home position
G30	Reference	Return to secondary home position
G40	Compensation	Cancel cutter compensation
G41	Compensation	Cutter compensation (left)
G42	Compensation	Cutter compensation (right)
G43	Compensation	Tool length compensation (+)
G44	Compensation	Tool length compensation (-)
G49	Compensation	Cancel tool length compensation
G54	Coordinate	Work coordinate system 1
G55	Coordinate	Work coordinate system 2
G56	Coordinate	Work coordinate system 3
G57	Coordinate	Work coordinate system 4
G58	Coordinate	Work coordinate system 5
G59	Coordinate	Work coordinate system 6
G80	Cycle	Cancel canned cycle
G81	Cycle	Drilling cycle
G82	Cycle	Drilling with dwell
G83	Cycle	Peck drilling
G84	Cycle	Tapping cycle

G90	Mode	Absolute programming
G91	Mode	Incremental programming
G96	Lathe	Constant surface speed
G97	Lathe	Constant spindle speed (RPM)
G98	Lathe	Feed per minute
G99	Lathe	Feed per revolution
G70	Lathe	Finishing cycle
G71	Lathe	Rough turning cycle
G72	Lathe	Facing cycle
G75	Lathe	Grooving cycle
G76	Lathe	Thread cutting cycle
M00	M-code	Program stop
M01	M-code	Optional stop
M03	M-code	Spindle ON (clockwise)
M04	M-code	Spindle ON (counterclockwise)
M05	M-code	Spindle OFF
M06	M-code	Tool change
M08	M-code	Coolant ON
M09	M-code	Coolant OFF
M30	M-code	End of program and reset
S	Spindle	Spindle speed command (RPM)
F	Feed Rate	Feed rate command
T	Tool	Tool Selection

Experiment:



CNC Code:

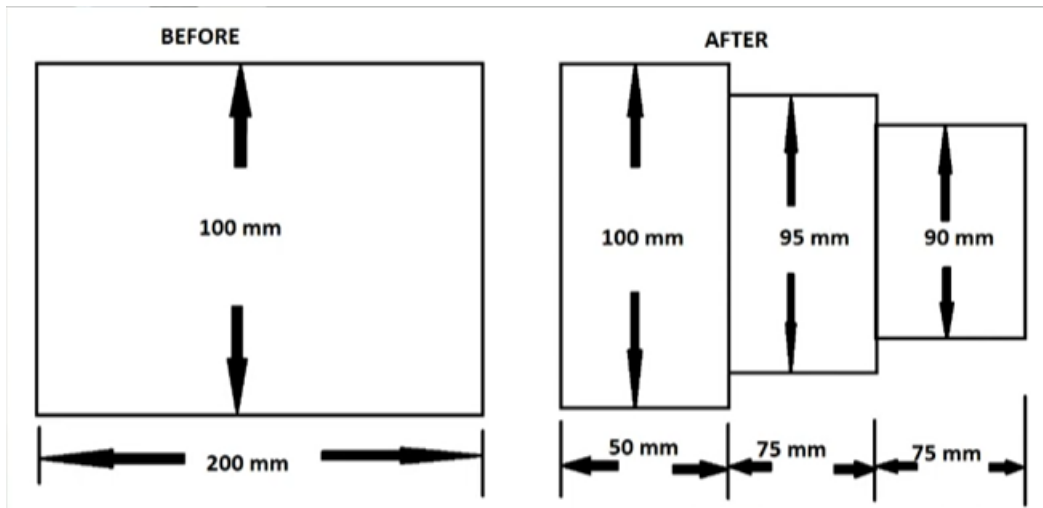
```
N1;  
G28U0.0W0.0;  
T0101;  
G97S1000M03;  
G21G99;  
G00X91.0Z10.0;  
G00Z0.0;  
G01X-2.0F0.1;  
G00W1.0;  
G00X91.0;
```

G00Z-1.0;
G01X-2.0;
G00W1.0;
G00X91.0;
G00Z-2.0;
G01X-2.0;
G00W1.0;
G00X91.0;
G00Z-3.0;
G01X-2.0;
G00W1.0;
G00X91.0;
G00Z-4.0;
G01X-2.0;
G00W1.0;
G00X91.0;
G00Z-5.0;
G01X-2.0;
G00W10.0;
G00X90.0;
G28U0.0W0.0;
M05;
M30;
;

Working Procedure

1. At first using the Reference position option, set the X and Z coordinates of the tool to 0.
2. Set the workpiece parameters to diameter =90 mm and length = 100 mm using the “Workpiece Setup” option.
3. Mount the cutting tool using the tool management option and set tool to External Cutter 1. Use edit option to select tool insert as CNMG 75⁰. Select Mount to turret option and add to tool station 1.
4. Select JOG mode and press Offset Settings. Press Offset, Geometry. Use the tool traverse options from the control panel to bring the tool towards the workpiece. Once the tool touches the workpiece outer surface, set X offset to 90.0. When the tool touches the workpiec face, set Z offset to 0.0. Once the offset is done, bring tool back to home position.
5. Select Program option and Edit option. Go to Directory and create a new program name and press insert. Press open file option and select the CNC Code notepad file (make sure your code is pasted and saved in a notepad file)
6. Once the code is loaded, select Auto option and Press cycle start. Use feed rate meter and reduce feed rate to 15 or 20.

Report Assignment



Cutting Tool: External Tool 001 (TNMG 60⁰ insert)

Sample Questions:

1. Write the difference between G-code and M-code.
2. Why is tool offsetting important?
3. Differentiate between absolute and incremental positioning system.