

Q2] Teach pendant programming (TPP):

→ A Manual robot programming method where an operator uses a teach pendant (hand-held controller) to:-

- 1) Jog the robot to desired positions using arrow keys/touch controls.
- 2) Save these positions and commands as part of a program.
- 3) Run the program, so the robot executes the taught task automatically.

→ Widely used in Industrial robots such as,

1) FANUC	3) KUKA
2) ABB	4) Yaskawa.

Structure of Teach pendant:-

- Display Screen - Shows robot coordinates, errors, program steps.
- Arrow keys - Move robot joints or in Cartesian space.
- Manual Selector - Manual / Auto.
- Function keys - Save position, Run, Edit, program, safety.
- Emergency stop Button - Ensure operator safety.

Process:-

1) Jogging the Robot:-

→ Operator uses the pendant to move the robot to specific points.

→ Movements can be,

* Joint Mode (Move each joint individually)

* Cartesian Mode (Move in x, y, z direction)

2) Recording Instructions:-

→ At each point, the operator records commands like:-

* MOVE → Move robot arm.

* GRIPPER CLOSE → pick object.

* GRIPPER OPEN → place object.

* WAIT → Pause until condition met.

→ These instructions are stored in robot memory as a program.

2) Program Execution:-

→ Once the sequence is ready, the operator switches to Auto Mode.

→ Robot repeats the sequence of instructions exactly as taught.

→ Example:- pick → Move → place → Return.

Program Code:-

program = [

```
{ "Command": "MOVE", "position": [0,0,0] },  
{ "Command": "GRIPPER", "action": "CLOSE" },  
{ "Command": "MOVE", "position": [10,5,0] },  
{ "Command": "MOVE", "position": [10,5,5] },  
{ "Command": "GRIPPER", "action": "OPEN" },  
{ "Command": "MOVE", "position": [0,0,0] }  
]
```

print("Executing TPP Code:")

for step in program:

if step["Command"] == "MOVE":

print(f"Moving to {step['position']}")

elif step["Command"] == "GRIPPER":

print(f"Gripper {step['action']}")

→ Simulated app:-

Executing TPP Code:

Moving to [0,0,0]

Gripper CLOSE

Moving to [10,5,0]

Moving to [10,5,5]

Gripper OPEN

Moving to [0,0,0]

Comparison Table:-

Aspect	Lead-Through programming	Teach-pendant programming	Robot programming Languages
How they taught	By Guiding robot directly through Motions	By Jogging robot & recording discrete steps/commands	By writing code with movement & control instructions
Type of path	Continuous path	Discrete path	Flexible
Suited for	Welding, painting & smooth motions	pick-place, assembly	Vision-based robotics, complex automation.
Ease of use	Very Easy, No coding needed.	user-friendly	Harder, Requires programming knowledge
Flexibility	Limited	Limited	Very flexible.