

2) Lead Through Programming:-

→ Programming a robot by physically guiding it through the desired motions while the robot records the poses/commands.

→ Mainly used for, * Industrial arm.

* Mobile robots.

* Cobots → Collaborative Robots.

⇒ To detect human presence.

→ Procedure:-

→ First, physically move the Robot using teach pendant/Joystick.

→ Controller records joint angles / Cartesian poses and timing / I/O metadata.

→ After demonstration, the conversion of that recording into a program with speeds, blends, dwell times...

→ Replay it on the robot.

→ Workflow:-

Set Robot to Safe, Reduced-Speed Mode

↓

Turn on gravity Compensation.

↓

Enter teach/Record Mode

↓
Demonstrate the task

↓
stop & save

↓
post-process (Filter & Compress raw data)

↓
simulate & validate

↓
Deploy

→ program Code:-

```
recorded_points = []  
def record_point(position):  
    recorded_points.append(position)  
record_point([0,0,0])  
record_point([10,5,0])  
record_point([10,5,5])  
record_point([0,0,0])  
print("Replaying Robot Motion:")  
for point in recorded_points:  
    print(f"Moving to {point}")
```

→ simulated opp:-

```
Replaying Robot Motion:      Moving to [0,0,5]  
Moving to [0,0,0]  
Moving to [10,5,0]  
Moving to [10,5,5]
```

Program explanation:-

→ recorded points → Record the point physically moving robot.

→ $[0,0,0]$ → start position.

→ $[10,5,0]$ → Move Right.

→ $[10,5,5]$ → Move up

→ $[0,0,0]$ → Return back.

2) 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.