



Robot Programming Methods

Pete Archer – Product Specialist

March 2019

Three common ways to program a robot are by a teaching pendant, teaching by demonstration, and offline programming. Here are the advantages and disadvantages of each of those methods.

Teaching Pendant

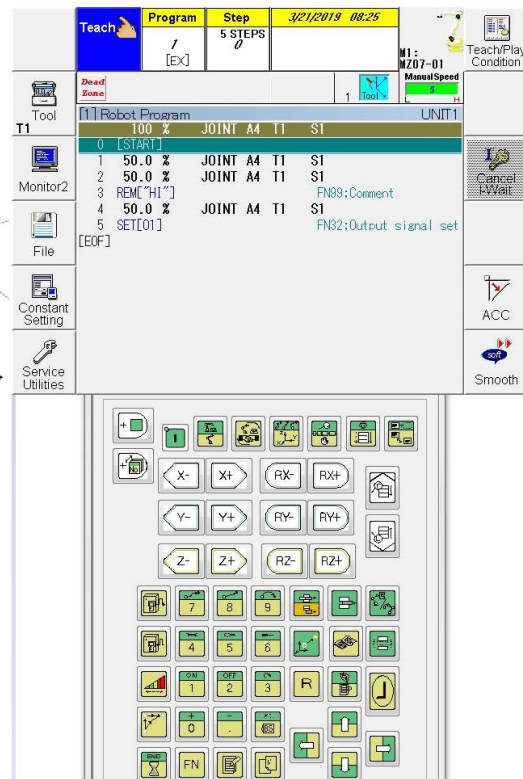
Advantages of a Teaching Pendant

- Most traditional industrial robots come with a teach pendant, which makes them familiar to technicians.
- They allow precise positioning, as the robot can be programmed using numerical coordinates in world coordinates, robot coordinates, or another coordinate system.
- Teach pendants are great for simple movements, such as painting in a straight line or over a large flat surface.

Disadvantages of a Teaching Pendant

- Disruptive to the whole system due to robot downtime. The robot must be put into "teach mode" and all operations using the robot halted until it has been programmed.
- Requires training to learn and program.
- Might be difficult for skilled craftspeople who are unfamiliar with programming.

Examples of Teaching Pendants:





Automation, Inc.

www.automationinc.com
(763) 571-3336 | (800) 706-4775
4830 Azelia Ave N, Minneapolis, MN 55429

Teaching by Demonstration

Advantages of Teaching by Demonstration

- Quicker than traditional teach pendants. It removes the need for multiple button pressing, allowing the operator to simply move the robot to the desired position.
- More intuitive than both traditional teach pendants and offline programming as the task is programmed in almost the same way a human operator would perform it. This makes it simple for operators to learn. Generally, this method requires no knowledge of programming concepts or being familiar with 3D CAD environments (as simulation does).
- Very good for detailed tasks which would require many lines of code to achieve the same effect, such as path following and error handling (failed picks for example).

Disadvantages of Teaching by Demonstration

- As with a traditional teach pendant, this method uses the physical robot for programming. This means that it does not reduce downtime, as much as offline programming.
- Moving the robot to precise coordinates is not as straightforward as with the other methods. This is especially true with some joystick-based systems, where there is no way of entering a numerical value. Some systems combine these features by allowing for the entering of exact numerical coordinates along with positioning-based coordinates.
- Not so good for tasks which are "algorithmic" in nature. For example, if a robot had to paint a flat surface by moving horizontally along the surface, then move down an inch, move horizontally in the opposite direction, etc. Moving the robot by hand would be arduous and inaccurate for such a task.

Offline Programming

Advantages of Offline Programming

- Reduces downtime required for robot programming. Programs are developed offline, so the robot only must be halted while the new program is being downloaded and tested.
- Offline programming with simulation can be available to visualize the final motion in a virtual work environment.
 - Can be quite intuitive, especially if the robot can be moved around in a 3D CAD environment with drag and drop techniques.
 - Easy to test many different approaches to the same problem, which would be inefficient for online programming methods.

Disadvantages of Offline Programming

- Virtual models will (probably) never be able to represent the real world with 100% accuracy. Programs may still need to be altered after they are applied to the real robot.
- Might take longer overall. Although offline programming reduces the downtime of the robot, it means that someone must spend extra time developing the simulation, as well as testing it on the robot.



Styles of Offline Programming

1) Structured Text Based Programming

```
75 Mov PLACE, 10
76 '
77 Mov PPicture
78 m1= m1 + 1
79 If m1 <= (Row%*Column%) Then *loop
80 If m1 > (Row%*Column%) Then *Unload
81 GoTo *loop
```

- Easier to follow the program flow
 - Mov = Move
 - Math displayed as an equation
 - Logic and program jumps are easy to read (If-Then, GoTo, ...)

2) Line by Line, Function-Based Programming

[1] Robot Program					UNIT 1
	100 %	JOINT A4	T1	S1	
0	[START]				
1	50.0 %	JOINT A4	T1	S1	
2	50.0 %	JOINT A4	T1	S1	
3	REM["HI"]				FN99;Comment
4	50.0 %	JOINT A4	T1	S1	
5	SET[01]				FN32;Output signal set
[EOF]					

- More difficult to follow the program flow.
 - “50.0% JOINT A4 T1 S1” is a move instruction
 - Math, Logic, and program jumps are cryptically displayed as Functions



3) PLC Add-on-Instruction Based Programming



- Easy to follow the program flow
 - Ladder is more understood by end user
- Good for simple pick and place operations
- Complex path following generally not supported