Robot Programming

By

Bhavik Soneji

Indus University

Robot Programming

Content:

- Introduction
- On-line programming
- Concepts
- Teach-in
- Playback
- Sensor function
- Off-line programming
- What is an off-line programming?
- Arguments for an off-line programming
- Programming levels
- Summary

Introductory review

From the definition:

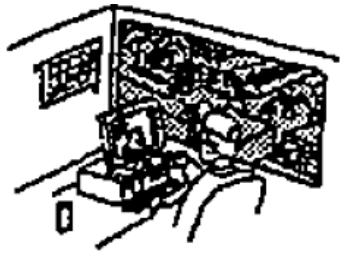
The industrial robot is an automatically controlled freely programmable, multifunctional manipulator with three or more programmable axes, for Industrial applications in either a fixed or mobile platform, can be used.

Freely programmable: programmable movements or auxiliary functions without physical amendment to be changed.

Robot programming

On-Line

Off-Line



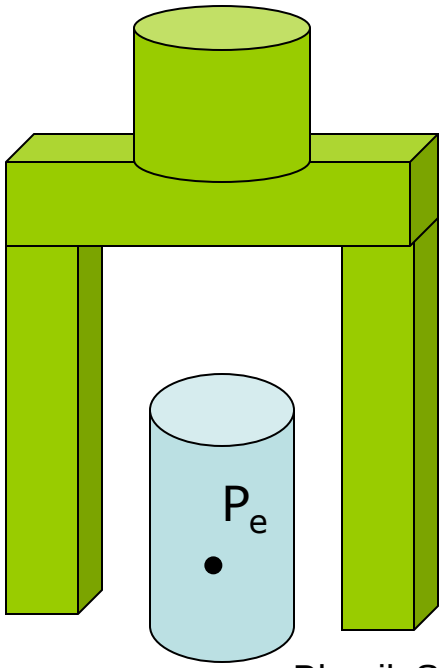
Robot Programming

- **Programming** is the *identification* and *specification* of a series of *basic actions* which, when executed in the specified order, achieve some specific task or realize some specific process.
- **Robot Programming** is the defining of desired motions so that the robot may perform them without human intervention.

Review: Some Definitions

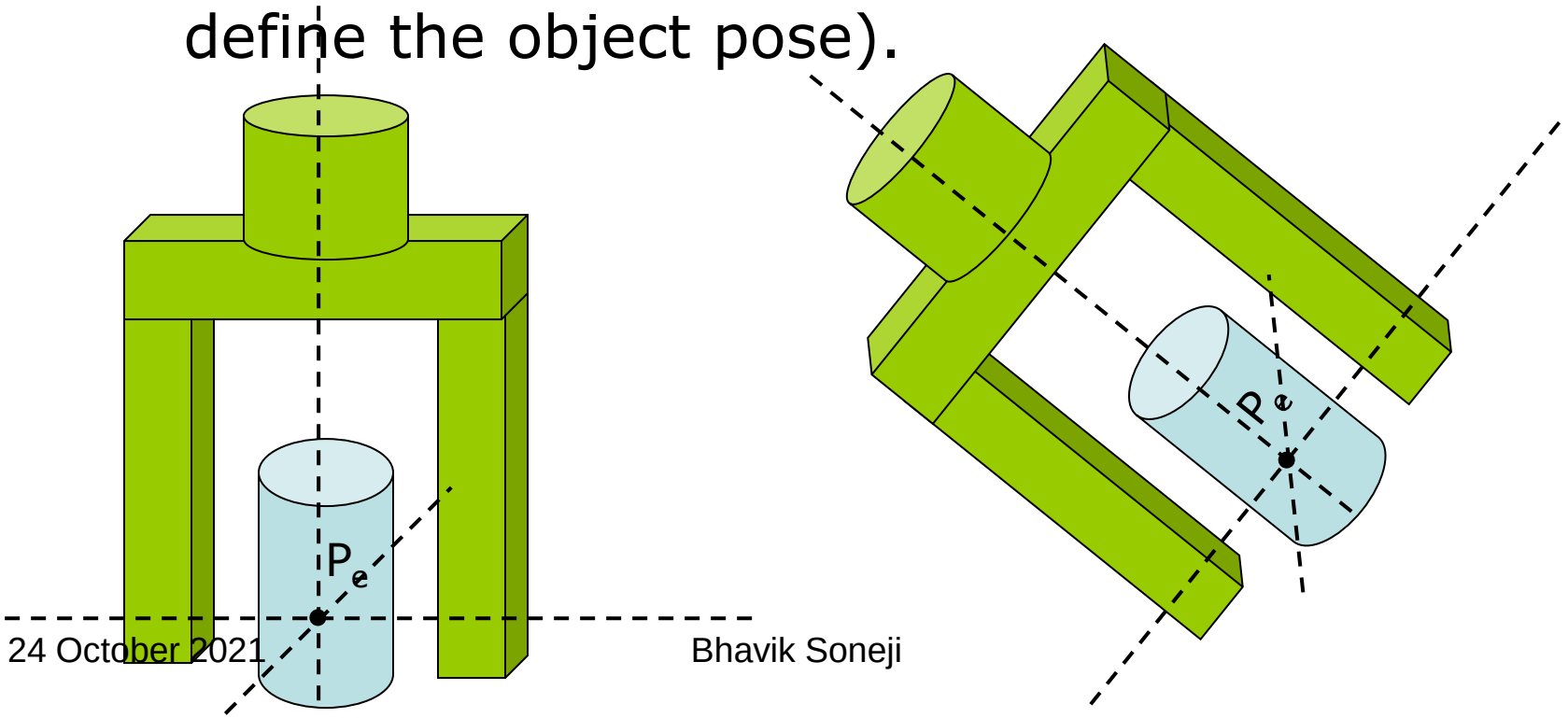
- **DoF:** The *degrees of freedom* [degrees of mobility] of the robot will be numbered as q_1, q_2, q_3 etc.
  - Usually industrial robot arms have between 4 and 6 degrees of freedom, one at each joint.
- **End-effector**: The end of the robot arm, where the gripper or other tool that the robots uses is located, we will define as the end-point (P_e) of the robot.

If, for example, the robot has a two finger gripper, to pick things up with, we usually define P_e to be a point between the two fingers, so that when this point is geometrically inside some object to be picked up, all the robot has to do is to close the fingers of its gripper to grasp the object. It can then move away with the object between its fingers.



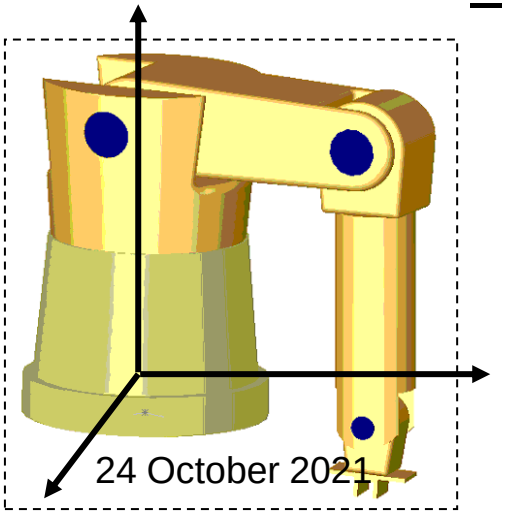
Pose: both the position of P_e in space, and its orientation

- It is not sufficient for P_e just to be defined as a point, we also need to attach or (conceptually) fix a coordinate system to it, so that we can define both the position of P_e in space, and its orientation (together they define the object pose).

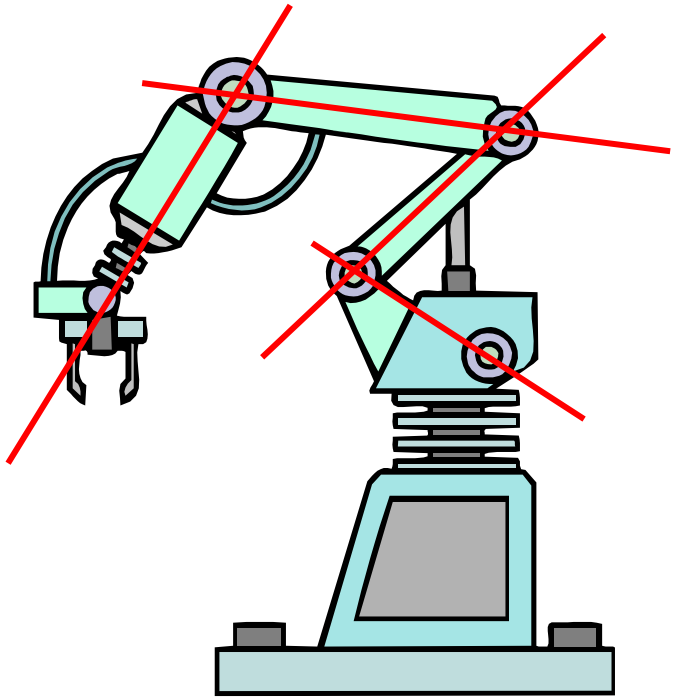


Base Frame: The position and orientation of P_e must be defined with respect to some *global frame of reference*, some global coordinate system. For this we usually use a frame of reference fixed to the base of the robot, which should not move.

- NOTE: The position and orientation of P_e in the work space of the robot are determined by the values of the joint positions of the arm—q_1, q_2, q_3,etc.



Configuration: Any particular position and orientation of P_e in space, and so any particular set of joint values, is called a *configuration* of the robot arm.



Robot Programming Revisited

- **Robot Programming** is the defining of desired motions so that the robot may perform them without human intervention.
  - identifying and specifying the robot configurations (i.e. the pose of the end-effector, P_e, with respect to the base-frame)

Introductory review

From the definition:

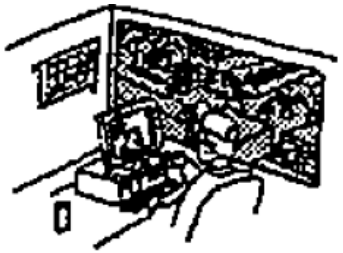
The industrial robot is an automatically controlled freely programmable, multifunctional manipulator with three or more programmable axes, for Industrial applications in either a fixed or mobile platform, can be used.

Freely programmable: programmable movements or auxiliary functions without physical amendment to be changed.

Robot programming

On-Line

Off-Line



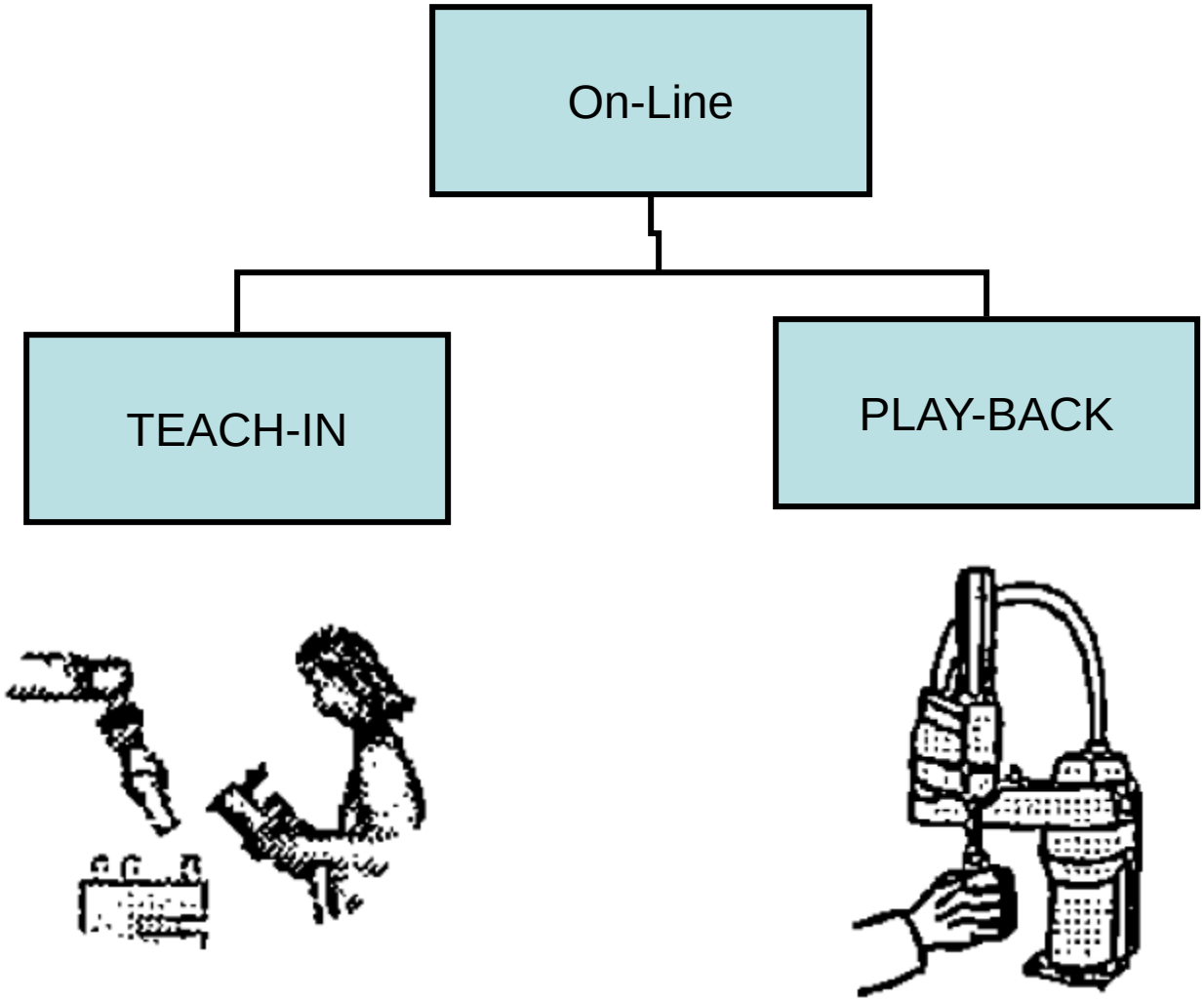
Robot Programming Methods

- Offline:
  - write a program using a text-based robot programming language
  - does not need access to the robot until its final testing and implementation
- On-line:
  - Use the robot to generate the program
    - Teaching/guiding the robot through a sequence of motions that can them be executed repeatedly
- Combination Programming:
  - Often programming is a combination of on-line and off-line
    - on-line to teach locations in space
    - off-line to define the task or “sequence of operations"

On-Line/Lead Through

- Advantage:
  - Easy
  - No special programming skills or training
- Disadvantages:
  - not practical for large or heavy robots
  - High accuracy and straight-line movements are difficult to achieve, as are any other kind of geometrically defined trajectory, such as circular arcs, etc.
  - difficult to *edit out* unwanted operator moves
  - difficult to incorporate external sensor data
  - Synchronization with other machines or equipment in the work cell is difficult
  - A large amount of memory is required

On-Line Programming

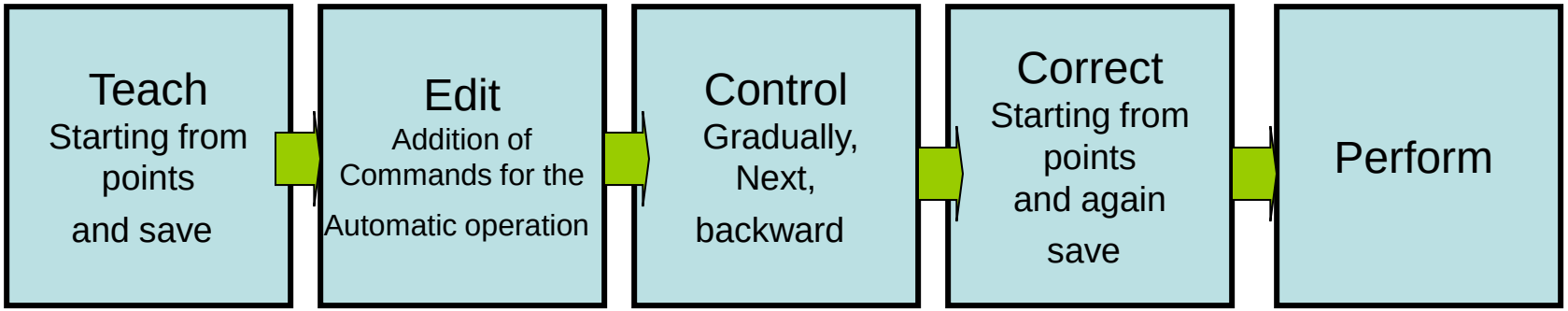


Teach-In

Task: transfer of appropriate programs in the robot controller

The robot is along certain trajectories through the two conducted manually with the both the robot and its movement and other devices in the robot cell monitored closely.

Teach-in Process:



Teach pendant

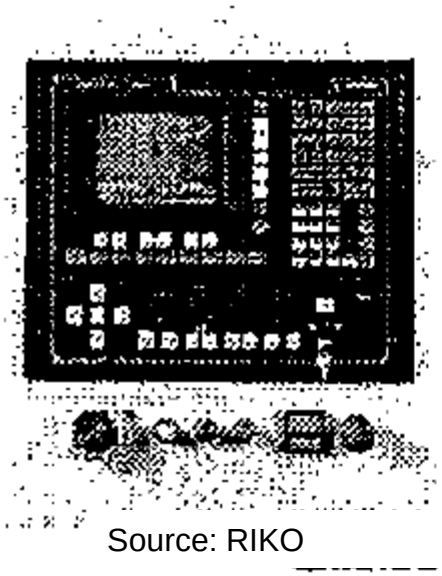
PHG-types to a Teach-In program:

Features:

- To enable long cable to a good view
- Emergency stop switch
- Enabling switch
- Selection of TCP (Tool Center Point) function
- Selection of the coordinate
- Range of motion mode
- Selection of the speed of movement
- Status Display

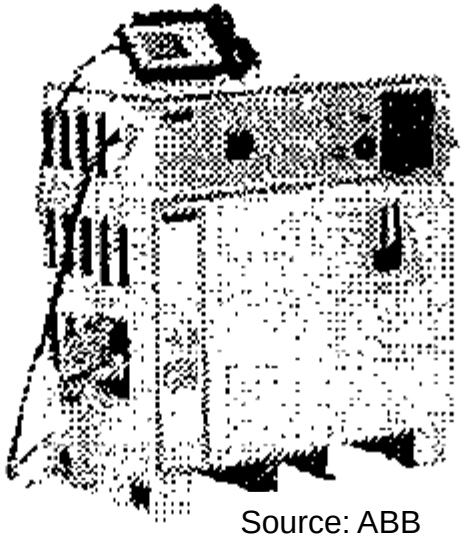
Control Equipment

Types of robot-Program:



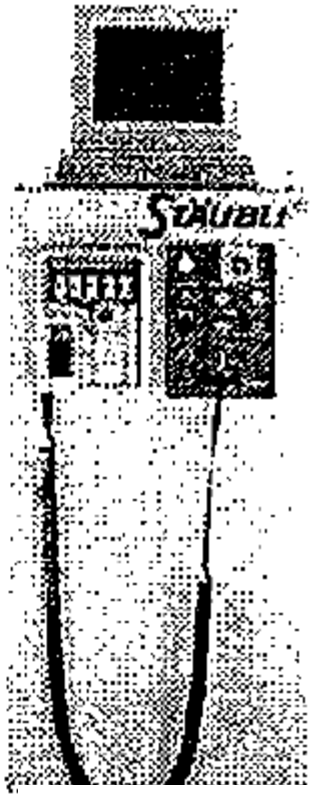
Source: RIKO

Editing on a CNC robot controlled by buttons and a built-in display



Source: ABB

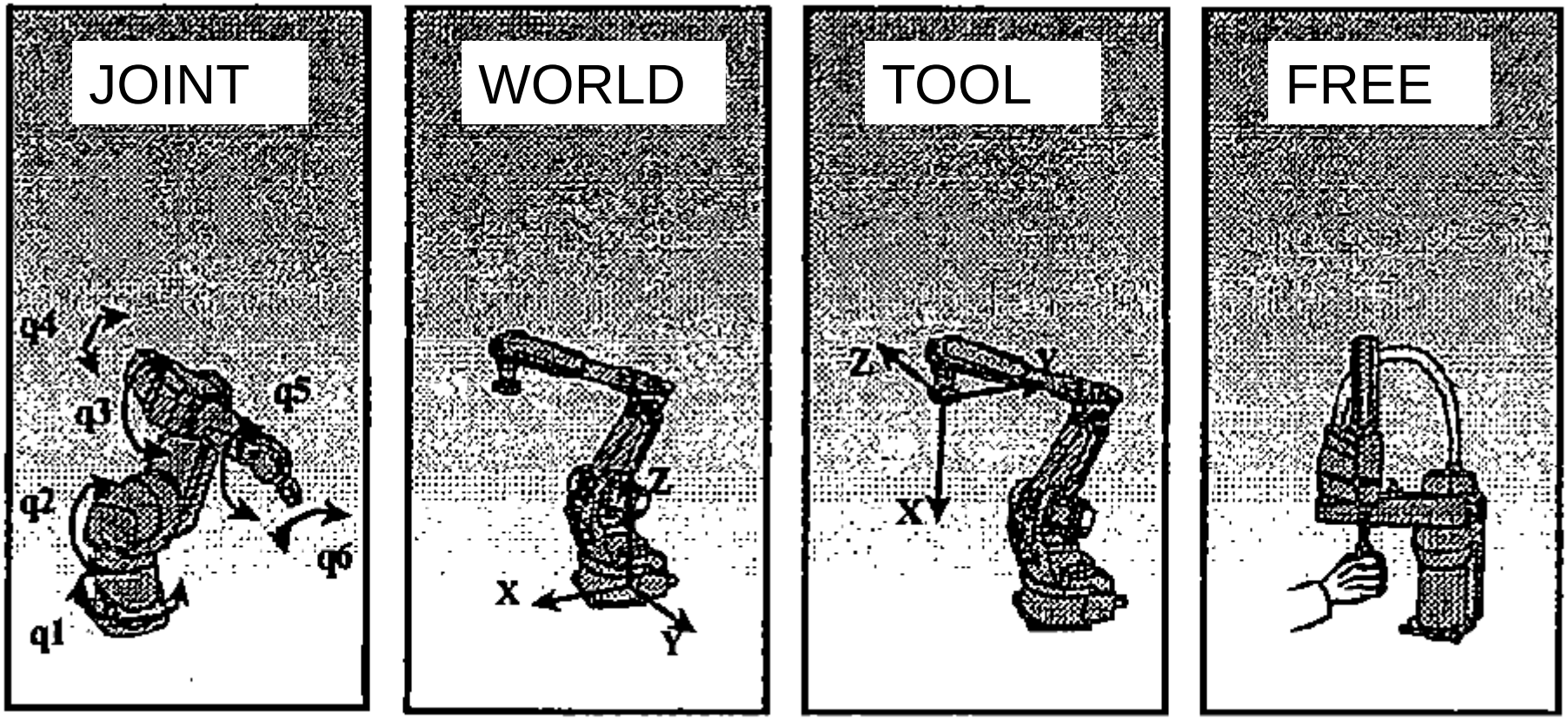
Editing by a keyboard and an LCD display on PHG



Source: Stäubli

Editing by a keyboard and LCD display

Teach-in coordinate systems



On-Line/Teach Box

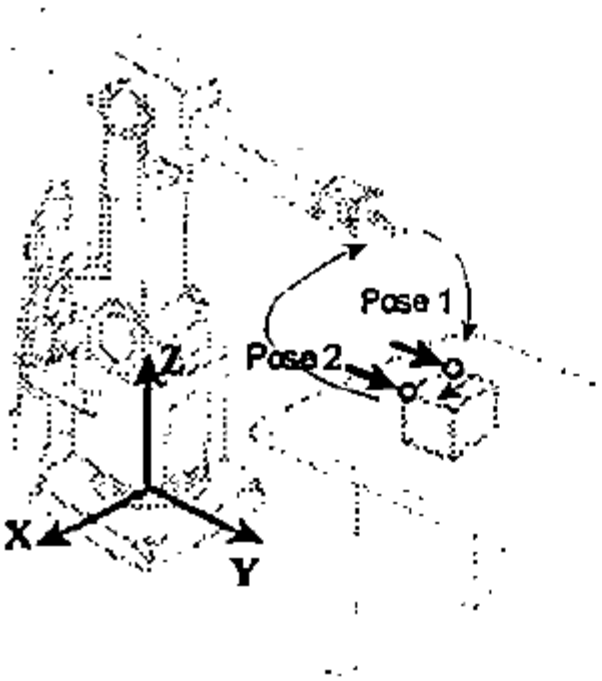
- Advantage:
  - Easy
  - No special programming skills or training
  - Can specify other conditions on robot movements (type of trajectory to use – line, arc)
- Disadvantages:
  - Potential dangerous (motors are on)

On-Line Programming

- Requires access to the robot
- Programs exist only in the memory of robot control system – often difficult to transfer, document, maintain, modify

Pose Programming

Abstract task:



Status:	X	Y	Z	R	P	Y
P1	30,21	956,34	987,22	0,00	1,25	16,45
P2	-35,62	957,28	987,98	0,00	1,25	16,45

- Enter the "TCP" data
- Selection of the "Joint" coordinates
- Rotation of axes 1, 2 and 3 to pose a
- Choosing the "World or tool" coordinates
- Align the tool with "Roll, Pitch, Yaw Movement
- Switch to "Step or Slow Motion
Pose, exact position and orient
- Pose 1 save
  - Poses are always stored in set-point (position and orientation)
- Motion in Cartesian coordinates to pose 2
- Pose, position them and orient
- Pose 2 save
- Movement in the "wait-and-home" position

Programming functions and signal

Example of a program on a very simple but defined robot!

Robot program	Explanation
MAXSP = 1000 60	-Determination of the total velocity in mm / s and degree / s
ACC = 2	- Determination of the acceleration factor
TCP1 = 0 155 0 0 0 0	- Determination of the TCPs in mm (X, Y, Z, R, P, Y)
SPEED = 100	- Determination of the rate in % overall speed
APPRO TO 1 FOR 0 -20 0	- Bringing the Pose 1 for X, Y, Z
SPEED = 20	-
SMOVE TO 1	Motion in Cartesian coordinates to Pose 1
SPEED = 7	-
WEAVE 2 5 8 0	Determination of the pendulum motion during welding
WELD START	- Welding start (signal to welding device)
LINE TO 2 CON	- A straight weld line to pull Pose 2
WELD STOP	- Welding End
SPEED = 20	-
DEPART FOR 0 30 0	Move away from Pose 2 for X, Y, Z
SPEED = 100	- Movement in joint coordinates to the home position
HOME	

Quelle: RKO und IJS

Programming Languages

Almost every robot has its own programming language!

ARLA	ABB
AML	IBM
BAPS	Bosch
DOROB	AEG
HELP	DEA
KAREL	GMF
ROBOTsar	Reis
ROLF	Cloos
SIGLA	Oliveti
SRCL	Siemens
VAL II	Unimation
V+	Stäubli
V+	ADEPT

Control independent language:

PasRo
SRL
C++

Play-Back Programming

- The operator uses the robot (or a device driver) directly.
- He leads the mechanism manually the task accordingly.
- During the movement the respective joint angle values can be read in a fixed time frame and stored.

Source: Gorenje

Requirements:

- Kinematic balance

- Low friction in the joints

- Very high flexibility of movement

Advantages:

- Very easy programming

- No prior knowledge of the operator necessary

Disadvantages:

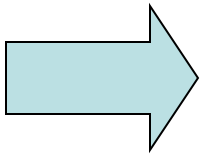
- Correction in difficult sections of path

- Difficult change of velocity

Source: ABB

Task of the sensors

Approach to the programming ease, flexibility and versatility

Important role  **SENSORS**

- Searching for objects or poses

  Parts on an assembly line, seam beginning, seam edge, plate boundary, installation help ...
- Tracking contours or litigation

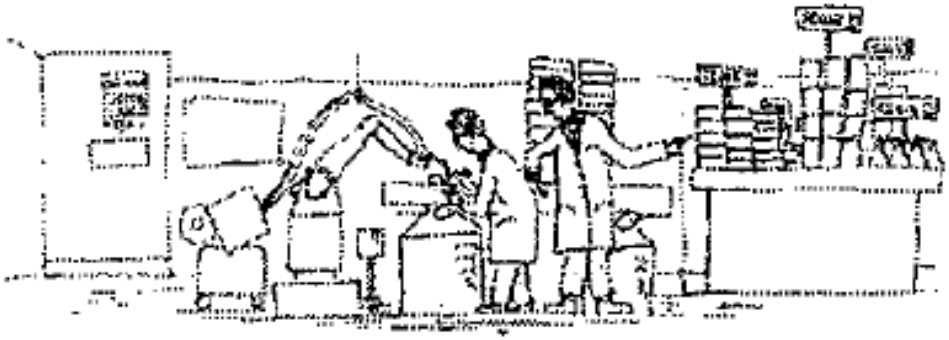
  Seam tracking, editing, with constant force (brushes, polishing, grinding, ...)

  Bending process pursue prosecution of Machine removal of parts ...
- Velocity fitting

  Teaching at a standstill, performing the move (coating on the assembly line, ...)

Off-Line Programming

What is Off-Line Programming?

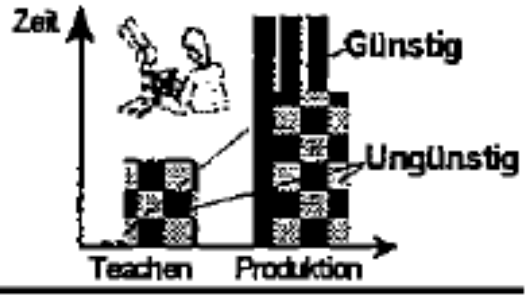


When?

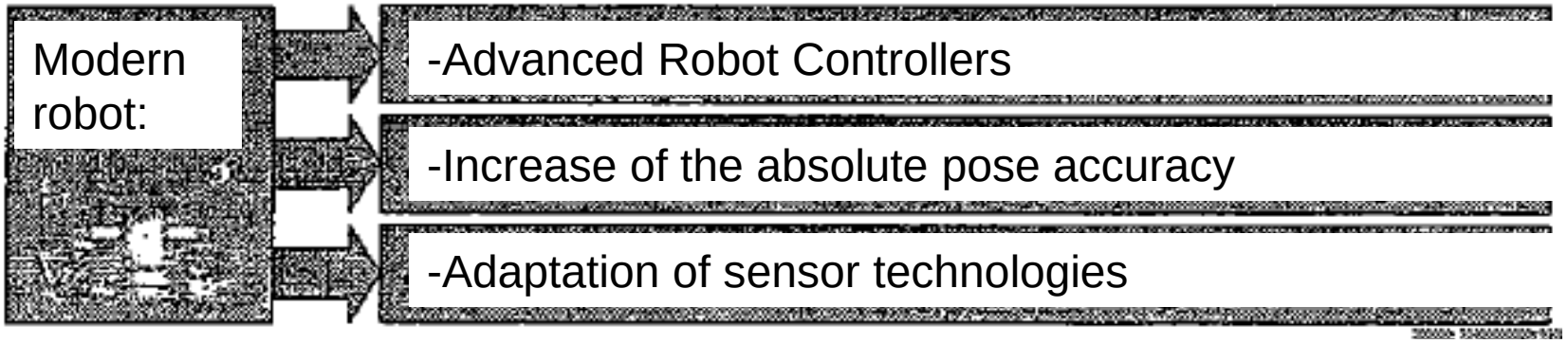
Why Off-Line Programming?

What is Off-Line Programming?

Teach-in methods:
Satisfactory only when the time for teaching in comparison to the production time is low.
The robot must be used instead for the programming for the production.



Off-Line Programming:
The robot can work, while a new program is created.
Robot programs are independent of the singly or in total produced in the production of industrial robots working.



These techniques are increasingly used in the industry.

Off-line Programming

- Programs can be developed without needing to use the robot
- The sequence of operations and robot movements can be optimized or easily improved
- Previously developed and tested procedures and subroutines can be used
- External sensor data can be incorporated, though this typically makes the programs more complicated, and so more difficult to modify and maintain
- Existing CAD data can be incorporated-the dimensions of parts and the geometric relationships between them, for example.
- Programs can be tested and evaluated using simulation techniques, though this can never remove the need to do final testing of the program using the real robot
- Programs can more easily be maintained and modified
- Programs can more be easily properly documented and commented.

Robot Program Development Process

- Analyze and decompose the task into a series of operations on the objects involved, and specify their order.
- Identify and specify all the situations needed to program all the movements and actions of the robot.
- Identify any types of repeated actions or operations and specify them as subroutines with parameters.
- Design and develop the complete robot program and its documentation.
- Test and debug the program using a simulator of the robot and its work space.
- Test the program on the real robot.

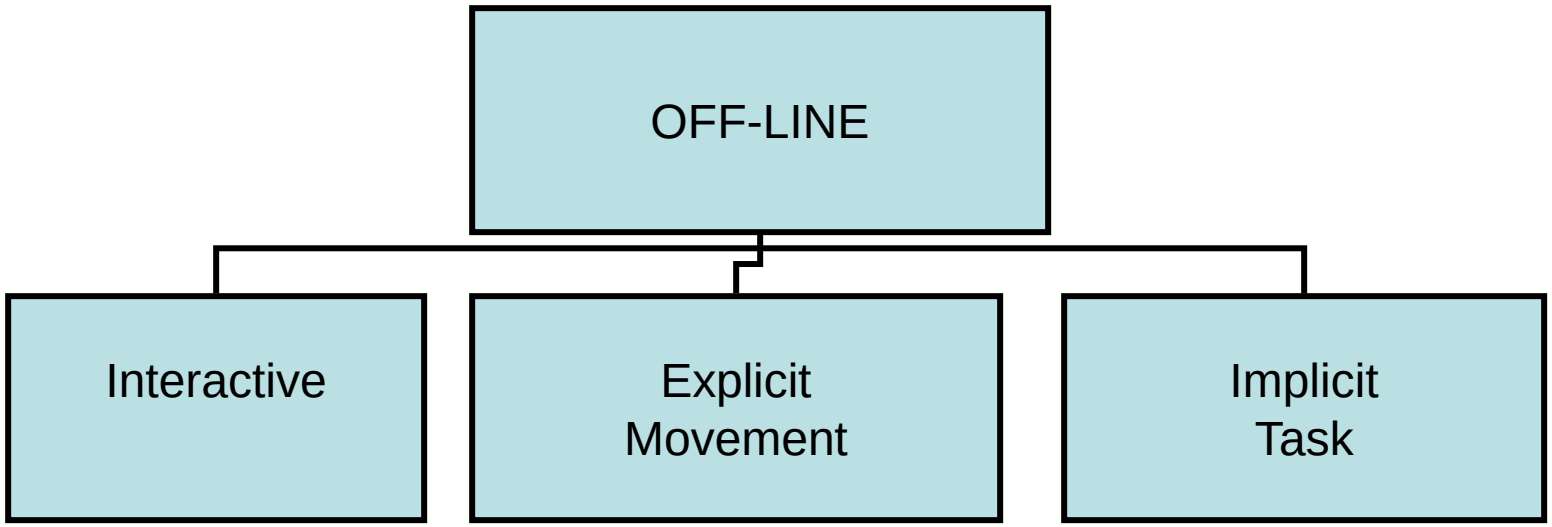
Why Off-Line Programming?

- Teach-In - Time-consuming activity
  - This increases with the complexity of the task
- Teaching reduces the cost
  - Robots can not produce during the Teachers
- Robot in mass production
  - Spot welding in automotive industry
  - Does teach in new constitute a minimal effort
- Robots in small and medium production
  - Programming time can be considerably-
  - Useful introduction of off-line programming
- Increasing complexity of robot tasks
  - Off-line programming can be very attractive

Benefits of Off-Line Programming

1. Reduction of the robot operating time
2. Dislocation of both ends of the danger area of the robot
3. Single robot programming system
4. Integration of CAD / CAM systems
5. Simplification of complex tasks
6. Optimization of robot programs
7. Access control
8. Cycle time analysis
9. Verification of robot programs

Off-Line Programming



Now we can say:

- Off-line programming is a significant increase in the productivity of the robot in the industry.
- The main arguments for Off-Line system is the reduction of the overall robot use time by the overlap of the robot-programming with the actual robot work.
- There are universal off-line programming systems, which allow the generation of robot program codes accessible for almost any industrial robot. In order to reduce their dependence start from a single robot.

Type of Robot Programming

- Joint level programming
  - basic actions are positions (and possibly movements) of the individual joints of the robot arm: joint angles in the case of rotational joints and linear positions in the case of linear or prismatic joints.
- Robot-level programming
  - the basic actions are positions and orientations (and perhaps trajectories) of P_e and the frame of reference attached to it.
- High-level programming
  - Object-level programming
  - Task-level programming

Object Level Programming

- basic actions are operations to be performed on the parts, or relationships that must be established between parts

pick-up part-A **by** side-A1 **and** side-A3

move part-A **to** location-2

pick-up part-B **by** side-B1 **and** side-B3

put part-B **on-top-off** part-A

with side-A5 **in-plane-with** side-B6 **and**

with side-A1 **in-plane-with** side-B1 **and**

with side-A2 **in-plane-with** side-B2

Task Level Programming

- basic actions specified by the program are complete tasks or subtasks

paint-the car-body *red*

assemble the gear-box

Literature:

http://www.societyofrobots.com/robottheory/Survey_of_Robot_Programming_Systems.pdf