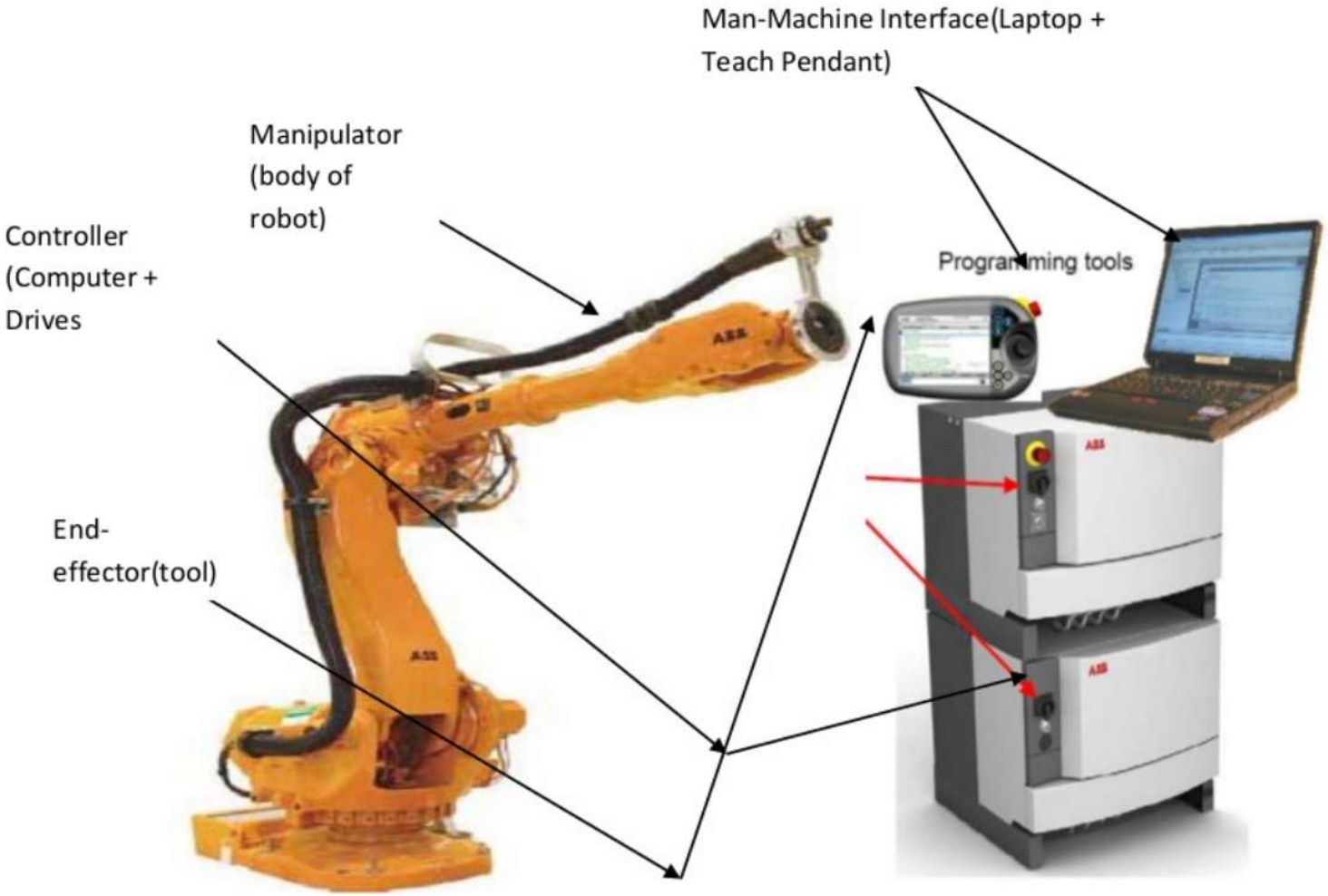
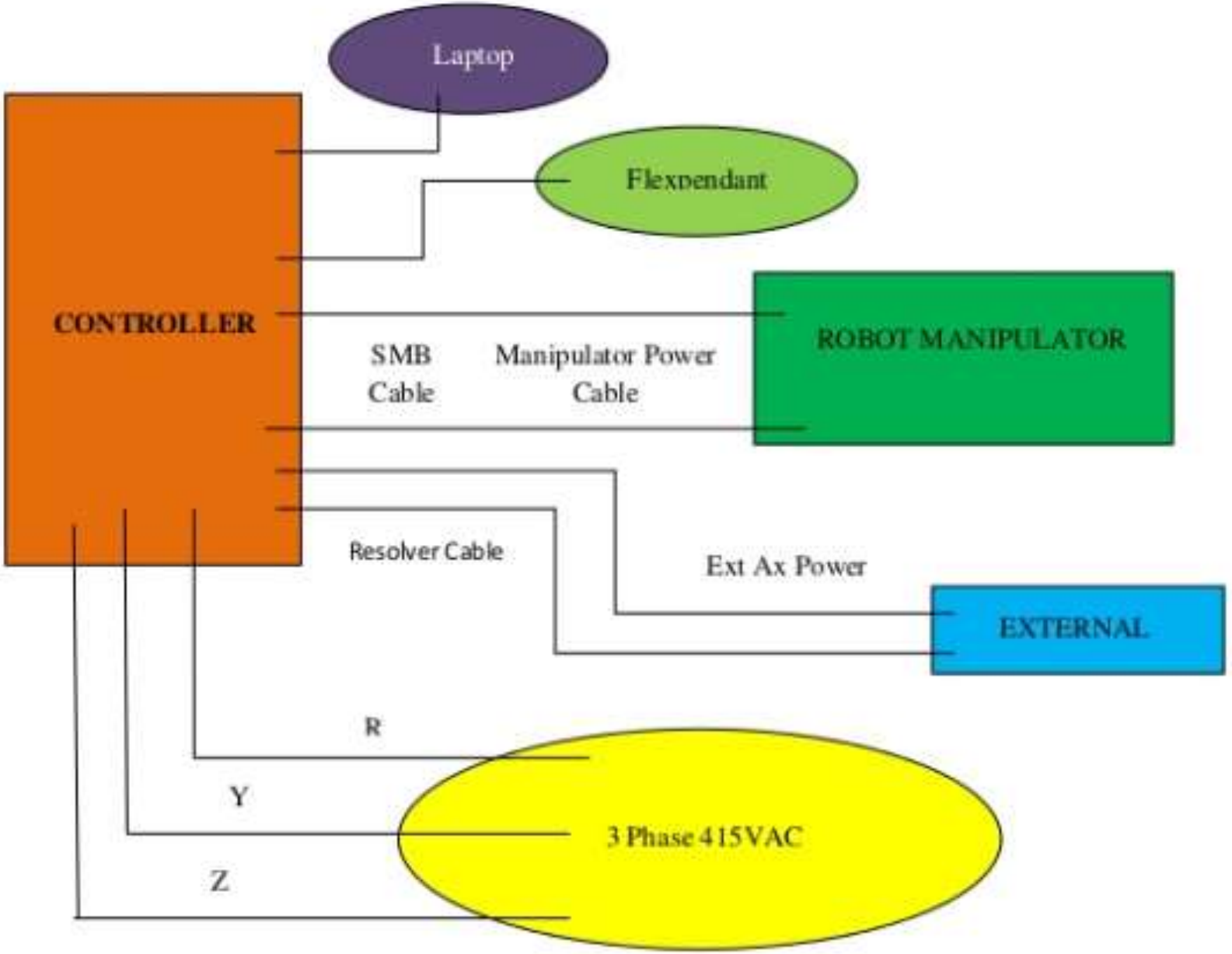
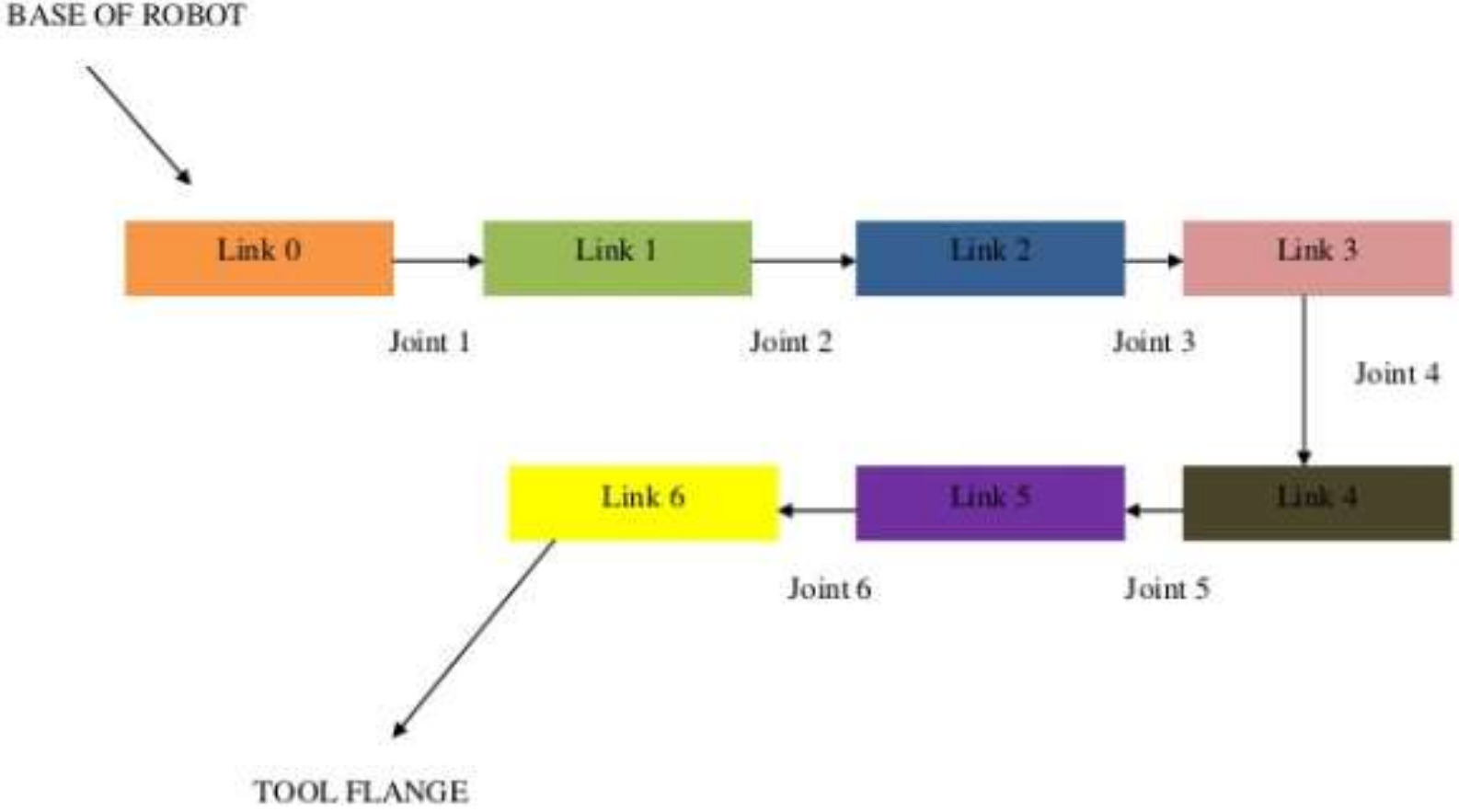
Robot Programming


Man-Machine Interface(Laptop + Teach Pendant)
Manipulator (body of robot)
Controller (Computer + Drives
Programming tools
End-effector(tool)

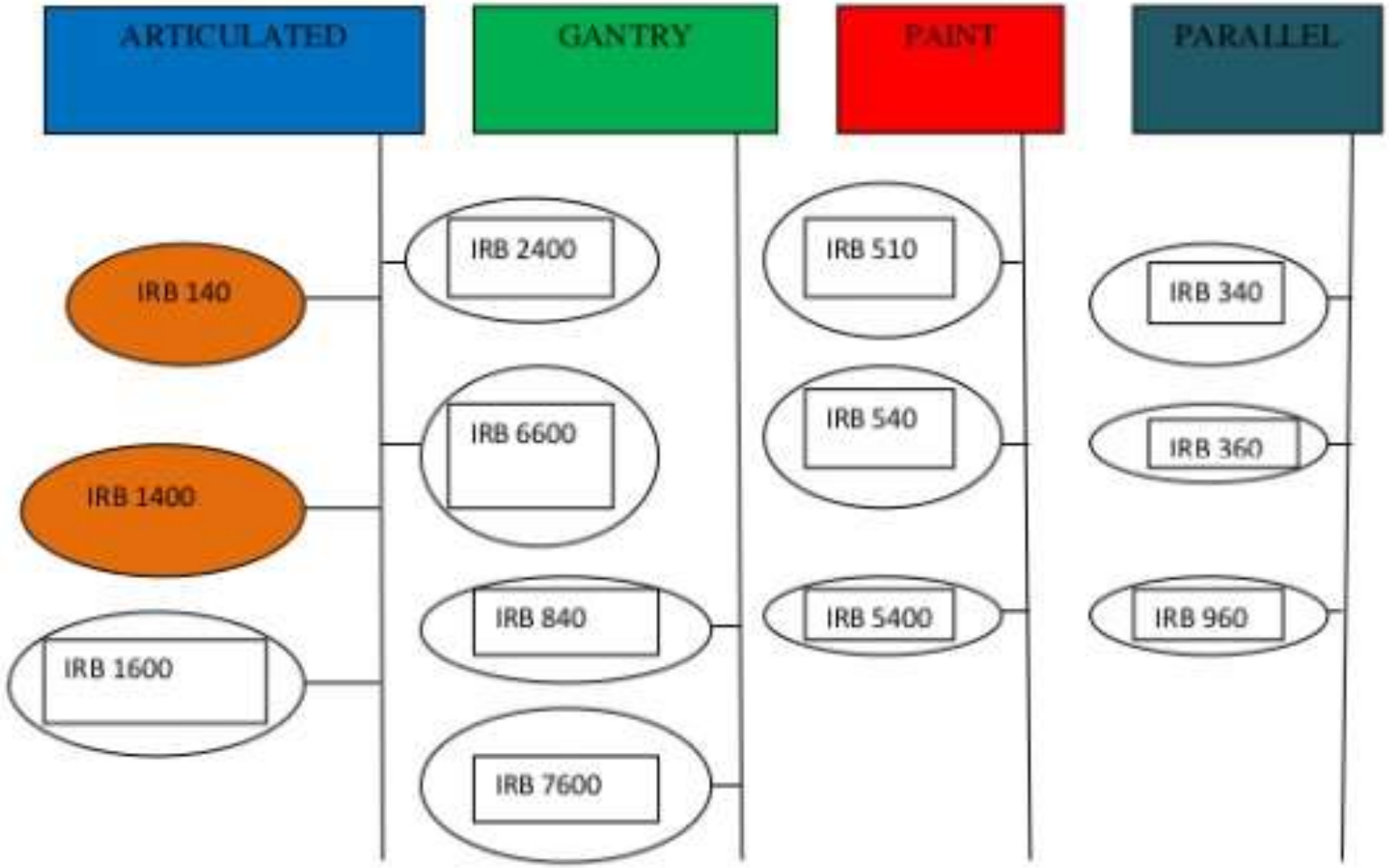

Laptop
Flexpendant
CONTROLLER
SMB Cable
Manipulator Power Cable
ROBOT MANIPULATOR
Resolver Cable
Ext Ax Power
EXTERNAL
R
Y
Z
3 Phase 415VAC



DESCRIPTION OF MANIPULATOR:

- A manipulator is an assemblage of rigid links connected by joints.
- Each Robot is driven by an actuator (A.C. Servo Motor for ABB Robots).
- Actuators are coupled to joints via geared transmission.
- An industrial manipulator has 4 or 6 Degree of Freedom.
- Brakes are installed in every joint motor to hold the manipulator in position against gravity in motors off state.

The ABB Robots are designated by IRB (Industrial Robot Body)



CONTROLLER:

1. The controller is the brain behind the functioning of a robot. The pictures below depicts the IRC5 Controller.

SINGLE CABINET

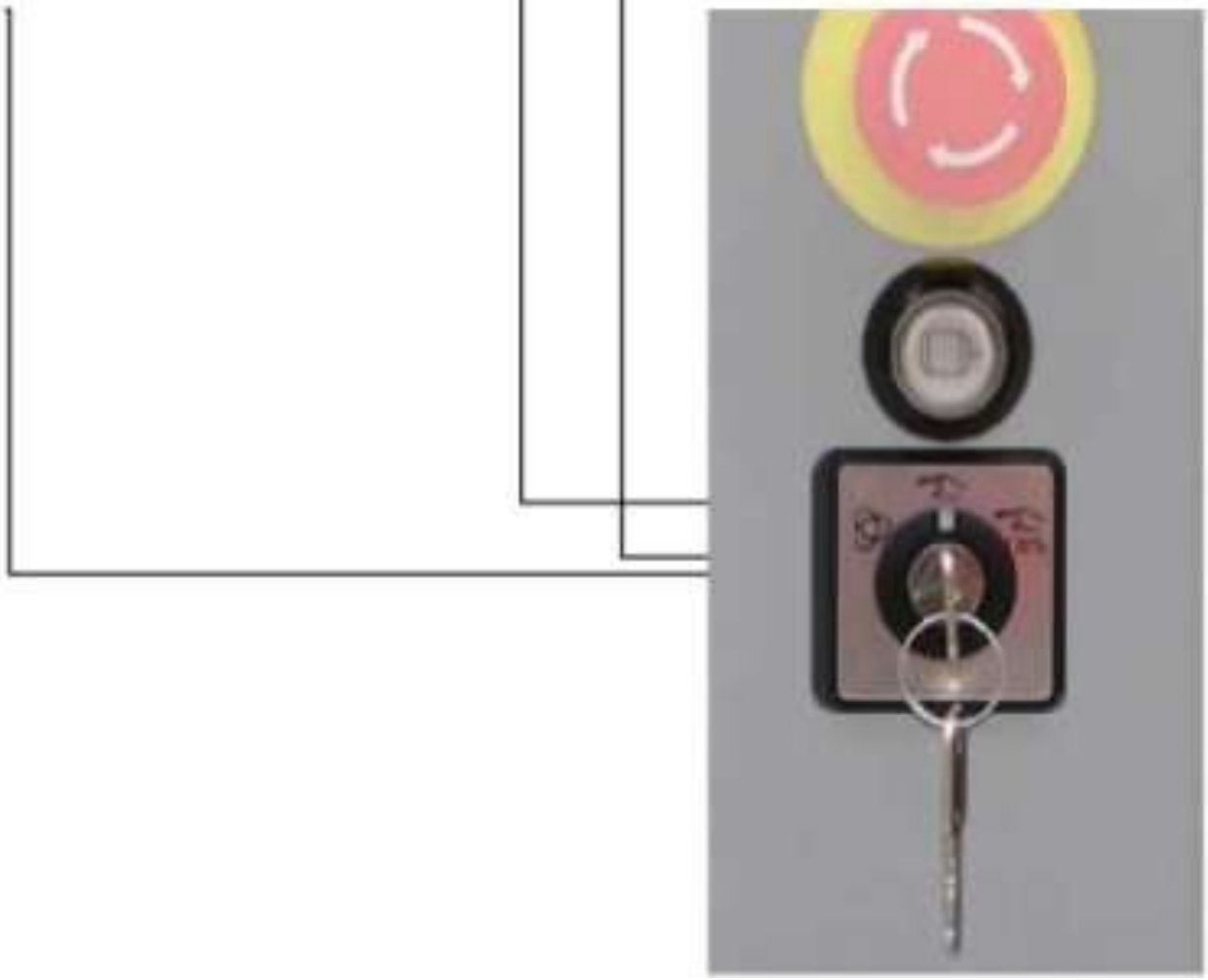
DUAL CABINET

THE MAN-MACHINE INTERFACE:

OPERATING MODES OF A ROBOT

A Robot can be operated in three different modes:

- Manual Mode
- Manual 100% Mode



1) **Manual Mode:**

- Robot can be jogged at less than 250 mm/sec.
- Enabling device needs to be pressed.
- Program speed is not followed.

2) **Manual 100% Mode:**

- Robot can be jogged at less than 250 mm/sec.
- Enabling device and Hold to Run button need to be pressed.
- Program speed is followed.

3) **Automatic Mode:**

- Robot cannot be jogged.
- No need of enabling device or hold to run button.
- Program speed is followed.

Co-ordinate systems

It is used to specify the position of point in space. The various types of Co-ordinate system used in a robot are:

- The Base Co-Ordiante System
- The World Co-Ordiante System
- The Tool Co-Ordiante System
- The Work Object Co-Ordiante System

JOGGING:

- Jogging means manually moving a robot using the joystick on the Flexpendant.
- Jogging cannot be done in auto mode.
- Jogging is used while teaching a robot points in space.
- Jogging can be done while programming.

MODES OF JOGGING:

Jogging can be done in three modes:

1. Axes mode (joint by joint).
2. Linear mode (along X / Y / Z).
3. Reorient mode (changing orientation of tool).

1. Axis Mode:

- We can jog axes 1-3 or axes 4-6 at one go.
- The position format shows the angular position of each joint in degrees or radians.

2. Linear Mode:

- In linear mode the TCP moves in a straight line.
- The TCP can move parallel to either the x-axis or the y-axis or the z-axis of the selected coordinate system of the robot which can be the base, world, tool or work object coordinate system.
- The position format shows the position of the TCP w.r.t the coordinate system selected in mm and orientation of tool in Quaternions or Euler Angles.
- During linear jogging orientation of tool remains same.

3. **Reorientation Mode:**

- In reorientation mode the TCP of the selected tool remains at a fixed position in space.
- However the orientation of the tool about that fixed point changes.

- The programming language used by ABB robots is the RAPID programming language.
- Programs can be accessed by going to the program editor window.
- To start writing a new program click on "Tasks and Programs" then on "File" and then on "New".
- Type in your new program name using the soft keyboard and you are ready to start.

A RAPID PROGRAM:

```
MoveJ Target_10, v1500, z100, tool10 \WObj:=
MoveJ Target_20, v1500, z100, tool10 \WObj:=
MoveJ Target_30, v1500, z100, tool10 \WObj:=
MoveJ Target_40, v1500, z100, tool10 \WObj:=
MoveJ Target_50, v1500, z100, tool10 \WObj:=

    ENDPROC

    PROC main()

    Path_10;

    Path 10;
```

INSTRUCTION SET:

The common instructions available can be classified under the following categories:

1. Motion instructions.
2. Program flow instructions.
3. Assignment.
4. Communication instructions.

1. **MOTION INSTRUCTION:**

   a. MoveJ *,v500,z50,tool0;
   b. MoveL *,v1000,z20,tool1;
   c. MoveC *,*,v250,z40,gripper;
   d. MoveC *,*,v250,z40,gripper;
   e. MoveAbsJ *,v500,z40,torch;

   a. MOVEJ:

   MoveJ *, v500, z80, gripper;

   * Represents the Robtarget where the TCP of the selected tool is to be moved.

   V500 means that the TCP moves at a speed of 500 mm/s.

   Z80 is the zone error i.e. 80 mm, if instead of z80 we select "fine" the zone error is zero.

   Gripper is the selected tool.

   TCP does not follow a straight line between initial position of robot and the robtarget.

b. MOVEL:

MoveL *, v500, z20, torch;

Rest is same as MoveJ only difference being that the TCP of the selected tool moves in a straight line from the initial position of the robot to the robtarget.

c. MOVEC:

MoveC *,*, v1000, z100, cutter;

The TCP of the selected tool moves in a circular arc joining the initial TCP position to the two robtargets respectively.

d. MOVABSJ:

MoveAbsJ *;

Here the * represents a joint-target that is the angular positions of the 6 joints.

2. PROGRAM FLOW INSTRUCTIONS:

a. IF ELSE
b. GOTO
c. FOR
d. COMPACT IF
e. TEST CASE

a. IF ELSE:

```
IF reg2=10 THEN
MoveJ *,v500,z80,tool0;
MoveL *,v1000,z50,tool0;
ELSE
MoveL *,v500,z20,tool0;
MoveC *,*, v500, z20, tool0;
ENDFOR
```

If a given condition is true it executes a set of instructions and if it is false then it executes another set of instructions.

b. GOTO:

```
GOTO start;
.....
.....
.....
start:
```

On seeing the instruction the program pointer goes to the line containing the label start.

c. FOR:

FOR x FROM 1 TO 10 STEP1 DO

.....

.....

.....

.....

ENDFOR

It is used to repeat a given set of instructions a fixed number of times.

d. COMPACT IF:

IF reg1=1 MoveJ *, v500, z20, tool0;

It executes a single instruction if a given condition is found to be true.

e. **TEST:**

TEST reg1
Case 1: .....
Case 2: .....
Case 3: .....
ENDTEST
Executes set of instructions based on the integer values of a variable e.g. reg1.

3. COMMUNICATION INSTRUCTIONS:

a. TPWrite "TIME FOR THE CYCLE IS",reg1;
b. TPErase;
c. TPReadNum reg2;