Drexel University
Goodwin College of Professional Studies
Engineering Technology
MET 205: Robotics and Mechatronics
Lab 3 Robot Controller - X series

Objective:

1. To learn the basics of a robot (YX Series)
2. To learn about the robot Controller. (RCX Series)
3. To learn to teach points.
4. To write a Simple program.

YK 250 X/ SCARA :

SCARA stands for Selective Compliance Assembly Robot Arm. As may be known from the name, the robot has compliance only in specific directions (X and Y directions) and has high rigidity in other direction (Z direction), and thus, has been designed mainly for automation of assembling works. At present, it is used in various production sites as a robot that is very effective not only in assembling works but also in component carrying works (pick & place works) because of its outstanding speed. Although SCARA robot is examined by comparing with Cartesian coordinate robot in many cases owing to its operating range, it can be said that SCARA robot is suitable for works requiring speeds on three axes or four axes motions because of its excellent cost-performance ratio. The features of this robot include its small installation area that provides higher degree of freedom in design of a system, and in addition, provides an advantage that, in case the system is disused in the future, the robot can be installed easily on other system.

Features of YK250 X

Cartesian coordinate robot
consists mainly of axis structure with Cartesian coordinate system. The robot scarcely requires conversion of coordinate by using control unit. Owing to its low cost, it is used in many cases for building into system.

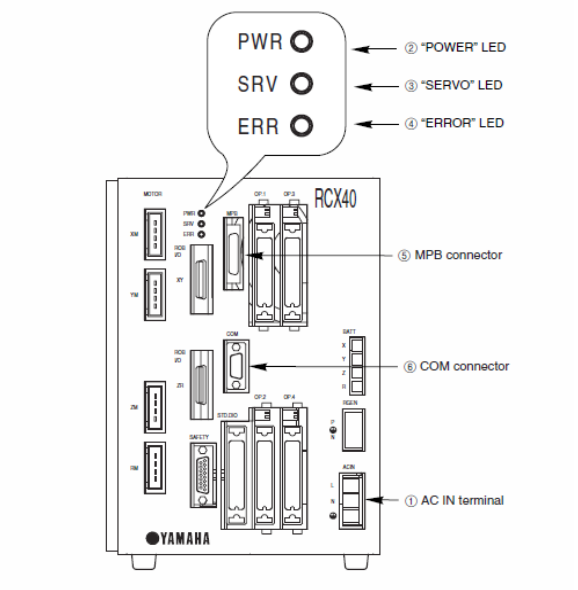
SCARA robot (horizontal multi-joint robot)

Has a joint structure mainly for operations in horizontal direction. This robot is given a high rigidity in vertical direction, and thus is adopted to move relatively heavy objects in horizontal direction. The features of this robot include very high speed in horizontal movement resulting in making the cycle time shorter.

Vertical multi-joint robot

It has a joint structure mainly for operations in vertical direction. Since the robot has the structure that is similar to the arm of men, it provides a flexible motion. However, the robot executes complex coordinate conversion processing, and thus requires a high performance control unit.

RCX Series



It uses a 4 axis controller.

Before Creating a Program

When writing a program, it's necessary to define the points first. There are two ways:
(1) By providing coordinate points in mm, and
(2) By moving the robot to a desired location.

(1) Teaching points by specifying the coordinates (mm)

☐Go to Manual
☐Press F1 (point)
☐Select point from the list
☐Press Edit
☐Enter x, y, z coordinates (the numbers should be in the format of ###.00)

(2) Teaching points by jogging the robot (Recommended)

☐Press F3 (Manual)
☐Press F1 (Point)
☐Use down arrow key, move cursor to the empty point
☐Jog the robot using x, y, z buttons to a desired location
☐Press F2 (Teach)
☐Press F4 (Yes) to overwrite the point

Creating a New Program:

1. Press ***Program [F2] → DIR [F3] → NEW [F1] → Enter Program Name → Press***  **.**
2. Press ***EDIT [F1]*** to enter command. To go to next line ***press***  .
3. After writing the program is complete press ***[ESC] → COMPILE [F5] → YES***
4. To run the program, go to initial screen by pressing ***[ESC] → AUTO [F1] → [START]***.

Some Common Commands:-

1. DELAY Statement
   Syntax: DELAY <expression>
   The value of <expression> must be from 10 to 655340. (Unit: ms)

The DELAY statement will cause the robot to delay movement for the period of time defined by the <expression>. The delay time is set in milliseconds, and the lowest allowable value is 10 milliseconds.

2. HALT Statement
   Syntax: HALT
   This statement will stop the execution of the program and reset the robot.
3. MOVE Statement
   Syntax: MOVE P, <point definition>
   This statement executes moving commands for the main robot.
4. SPEED Statement
   Syntax: SPEED<expression>
   The value of <expression> must be from 1 to 100. (Unit: %)
   This command changes the moving command speed of the main groups after this statement, to the value specified by the <expression>.
5. GOTO Statement
   Syntax: GOTO <label>
   This command will make the program to unconditionally jump to the line specified by the label.
6. DO(#) Statement
   DO(22) = 1 // turn the conveyor belt ON
   DO(22) = 0 // turn the conveyor belt OFF
   DO(24) = 1 // turn the suction cup ON
   DO(24) = 0 // turn the suction cup OFF

Sample Program:

To continuously move between 2 points (A and B) at different speeds and turn the suction cup and the conveyor belt ON and OFF.

Solution:

1. *TOP:
2. SPEED 20
3. MOVE P, P30
4. SPEED 20
5. MOVE P, P31
6. DELAY 1000
7. GOTO *TOP

Requirements:
1. Robotic calibration in XYZ coordinate: Gently but firmly tape a pencil or marker around the robot arm. Bring the robot to a position in which the pencil point touches a mark of a ruler. Calibrate if the robot moves to the positions (actual) as the positions programmed in teach pendant. Record the positions (mm) the robot moves.
2. Try 4 different speeds at 10%, 20%, 30%, and 40% of system speed for running the program and record the cycle time for each test. Plot a graph of cycle time vs. speed percent.
3. Calculate speeds in terns of mm/s and ipm (inch per minute). Plot graphs of speed vs. percent speed.
4. What is the effect of cycle time on productivity and quality in a production line?