HUMBER

ROBOTICS

LESSON 03
Robotic Programming – Part I
Motion & Input/Output Commands

IMRAN KHAN

School of Applied Technology Robotics and Automation

Motion Programming

A robot can be programmed for varieties of motion to reach the desired task. Common type of robotic movements are:

PTP : rough motion – not precise - saving few points between two positions

Linear : Precise motion – saving many points between two positions

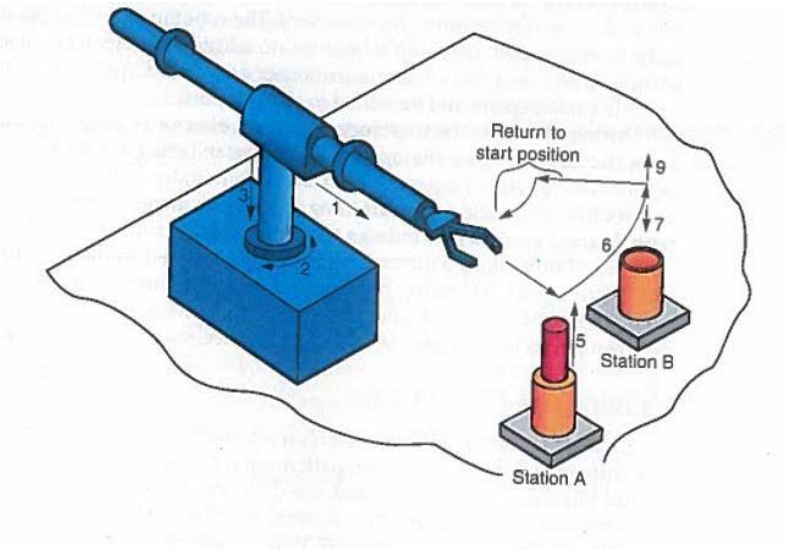
Reduce Accuracy: smooth continue motion between two points using cutting corner

Circular: produce motion in semi-circle or in circular pathe

Spline: continue smooth motion using spling block

Point to Point Motion

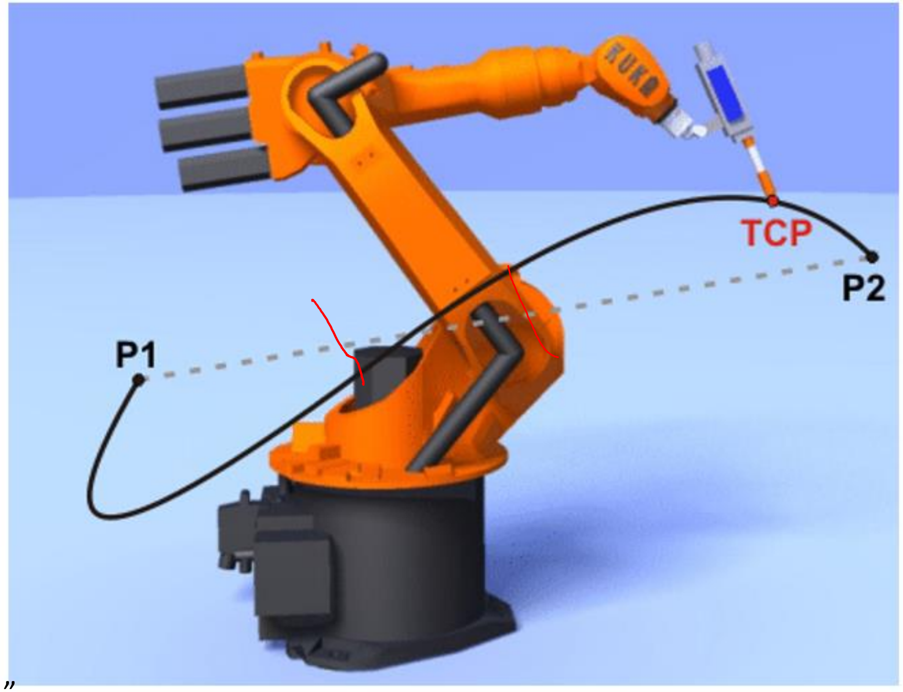
- Movement of a robot end effector through a number of points in space to desire location



Point to Point Motion

The robot guides the TCP along the fastest path to the end point. The fastest path is generally not the shortest path and is thus not a straight line. As the motions of the robot axes are rotational, curved paths can be executed faster than straight paths.

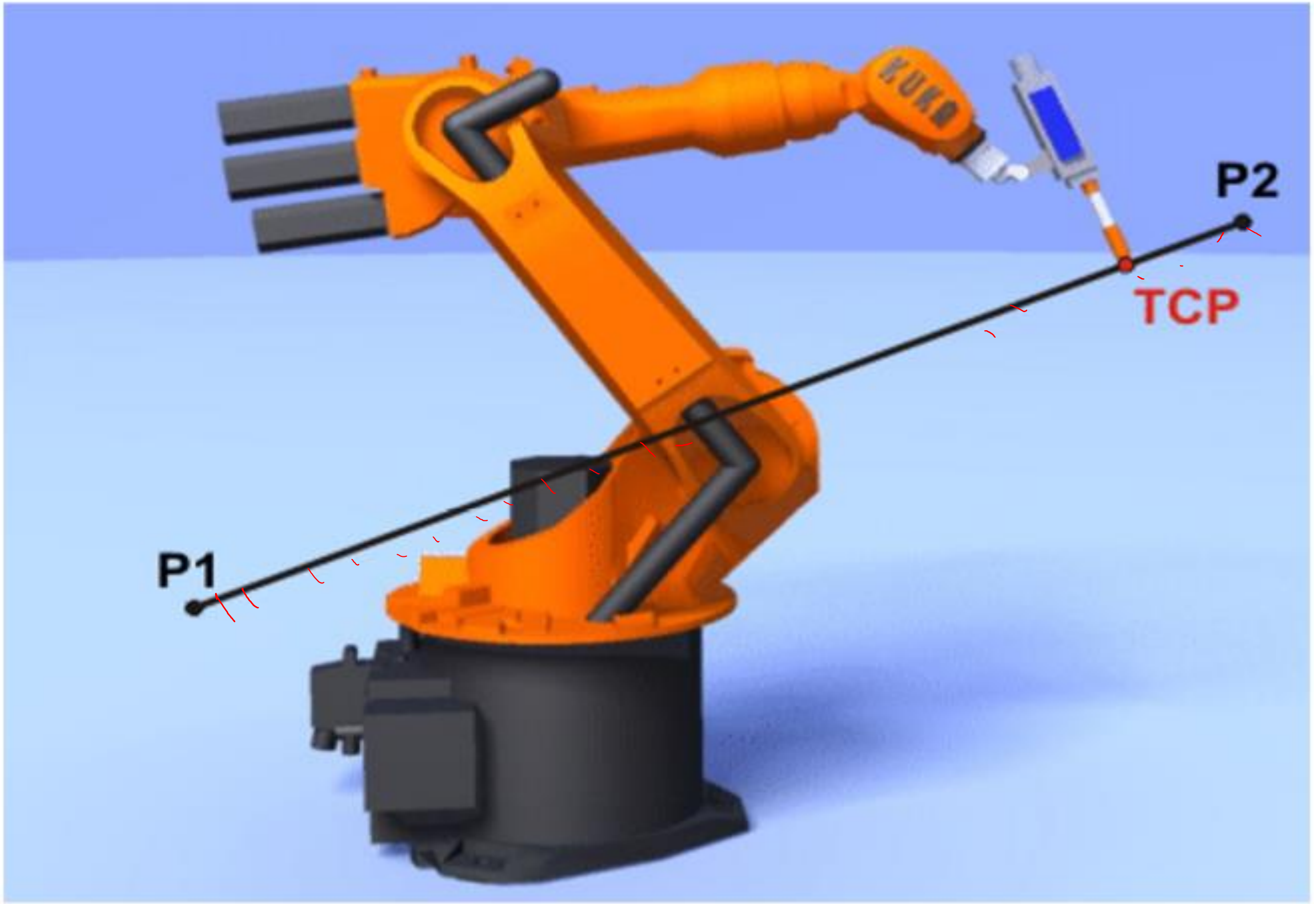
The exact path of the motion cannot be predicted.



Resource: KUKA KRC4 Manual -"KUKA system Software 8.3, section 9."

Linear Motion

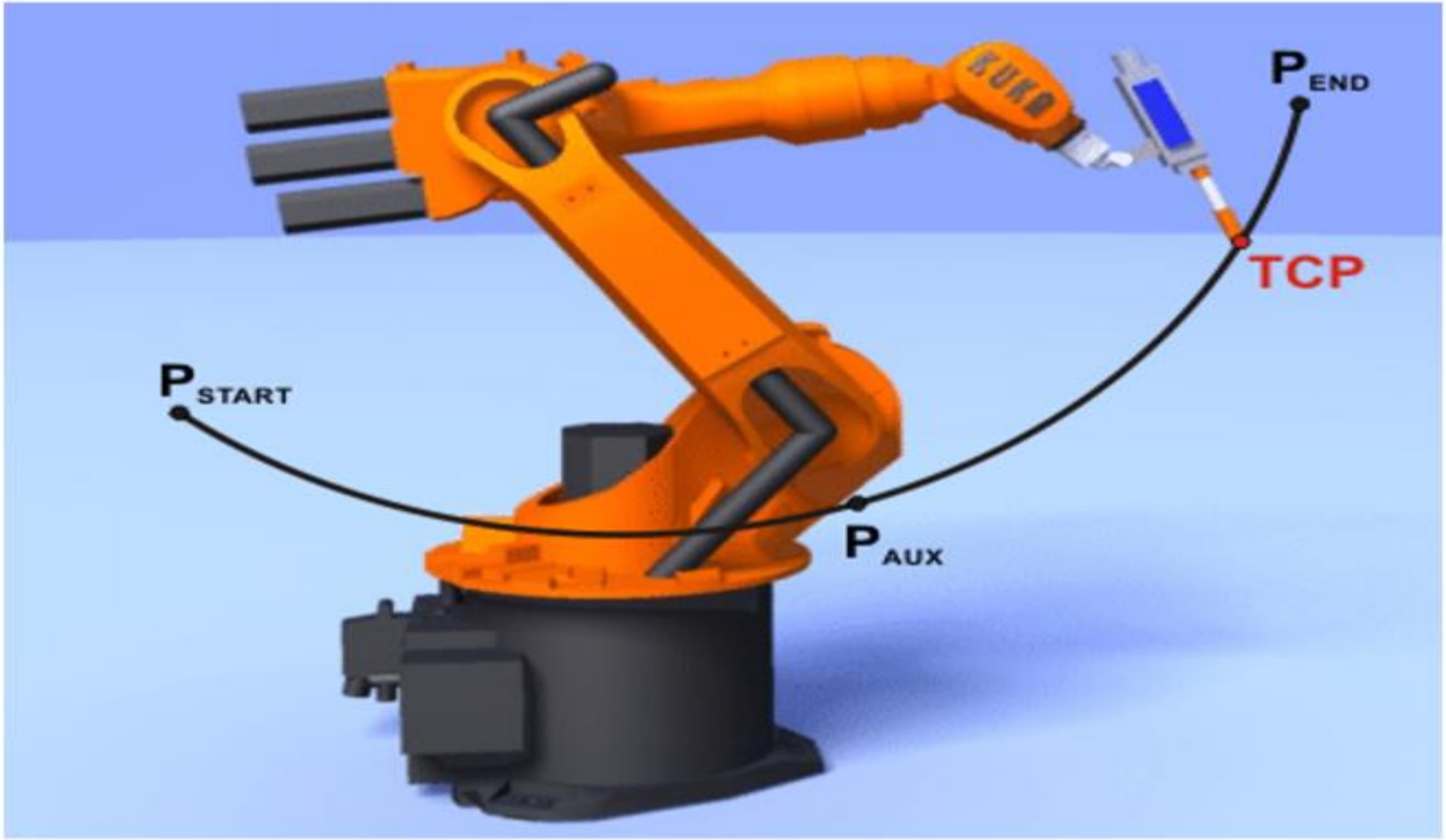
The robot guides the TCP at a defined velocity along a straight path to the end point.



Resource: KUKA KRC4 Manual -“KUKA system Software 8.3, section 9.”

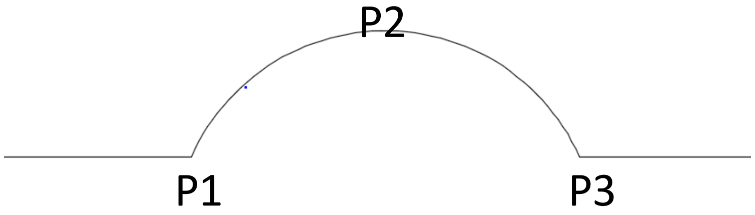
Circular Motion

The robot guides the TCP at a defined velocity along a circular path to the end point. The circular path is defined by a start point, auxiliary point and end point.

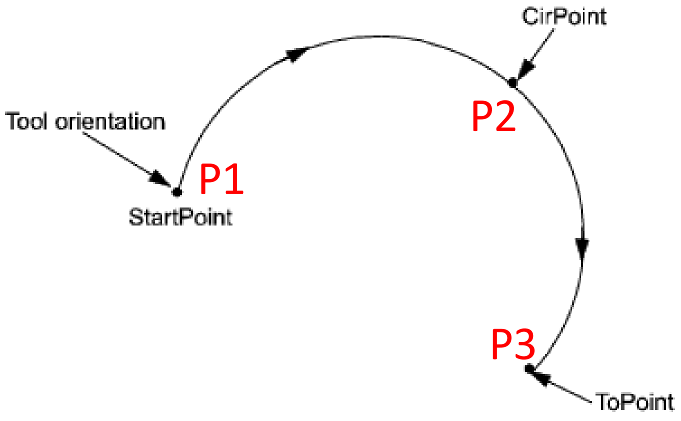


Resource: KUKA KRC4 Manual -“KUKA system Software 8.3, section 9.”

MOVE C

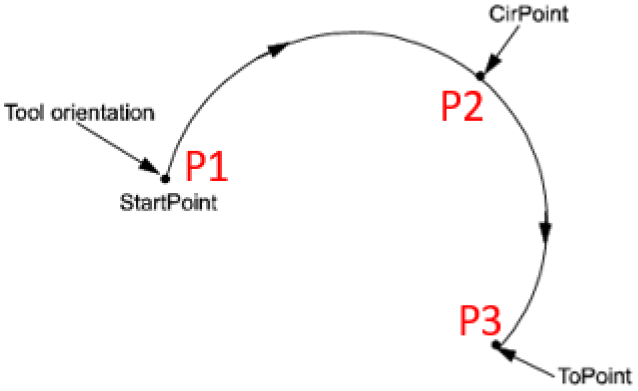


Move J P1
Move C P2, P3



Circular Motions in the Lab Robots

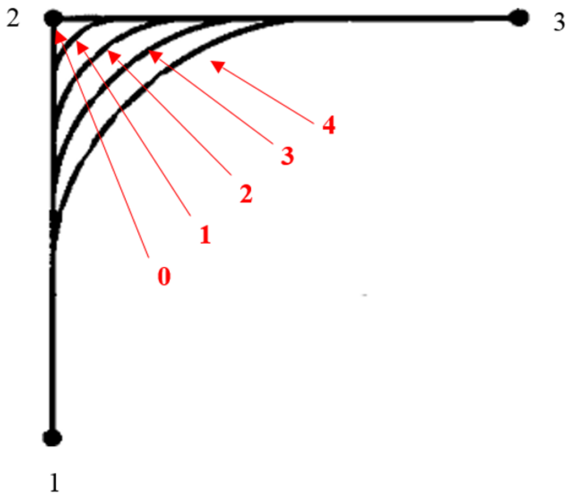
Type of Robot	CIRCULAR MOTION
KUKA	PTP P1 CIRC P2 P3
ABB	Move J P1 MOVEC P2 P3
MOTOMAN	MOVE C P1 MOVE C P2 MOVE C P3
DENSO	MOVE P P1 MOVE C P2,P3
PANASONIC	MOVE C P1 MOVE C P2 MOVE C P3
FANUC	J P1 C P2 P3



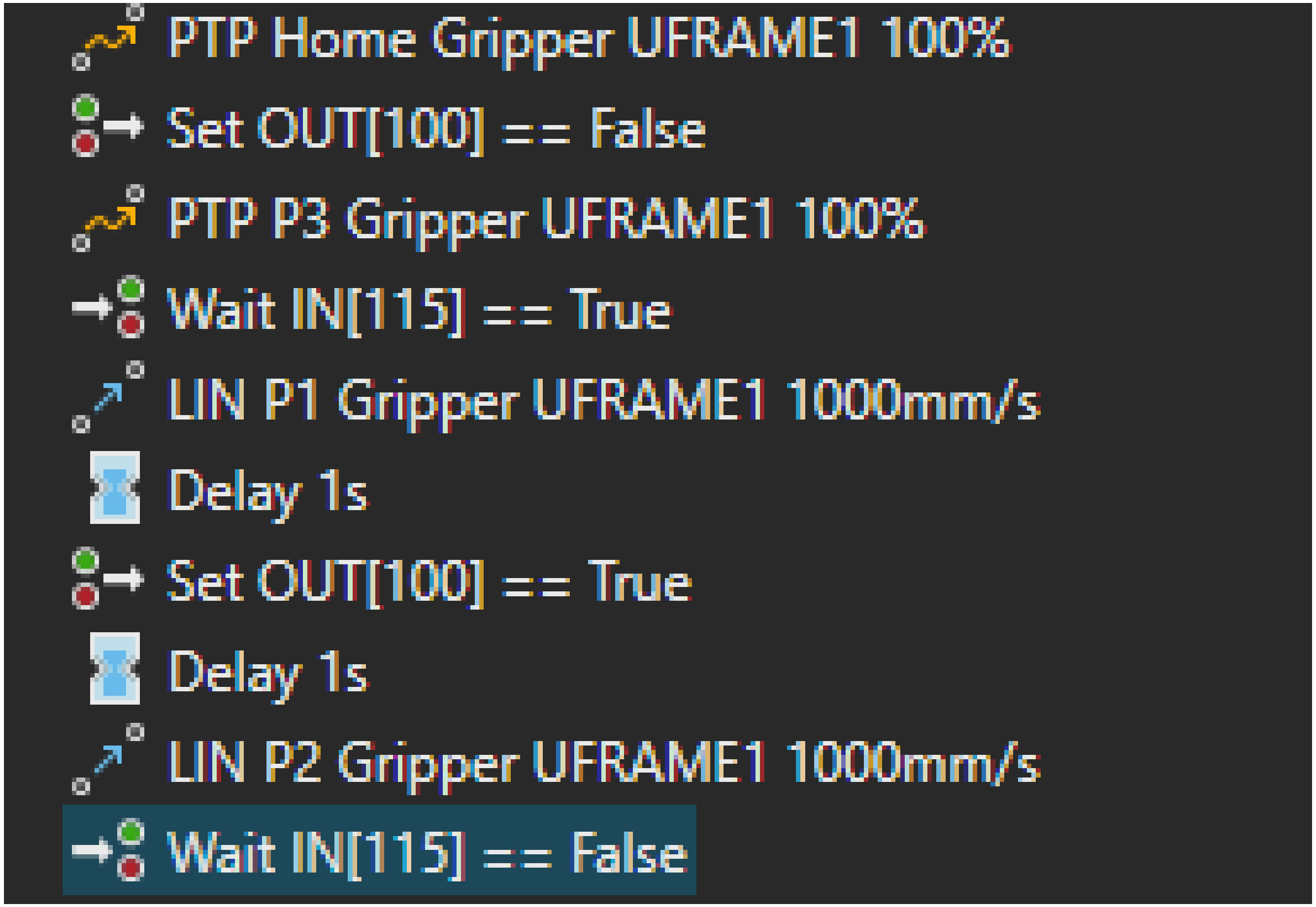
Reduced Accuracy in the Lab Robots

Type of Robot	Reduced Accuracy
KUKA	Cont. (Continuous) %
ABB	Z____ Z 250 (higher z value more reduced accuracy)
MOTOMAN	PL (0 - 8) Higher value more reduced accuracy
DENSO	@p
WITTMANN	Curve /Relative curve More visible in 3D movement instead of Linear movement
FANUC	Cont. (%)

Reduced Accuracy (Motoman)



Programming Example in Octopuz



```
PTP Home Gripper UFRAME1 100%
Set OUT[100] == False
PTP P3 Gripper UFRAME1 100%
Wait IN[115] == True
LIN P1 Gripper UFRAME1 1000mm/s
Delay 1s
Set OUT[100] == True
Delay 1s
LIN P2 Gripper UFRAME1 1000mm/s
Wait IN[115] == False
```

Point to Point & Circular Motion in KUKA

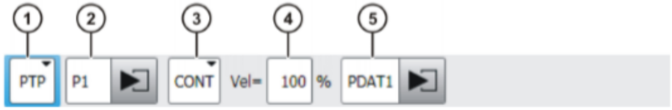
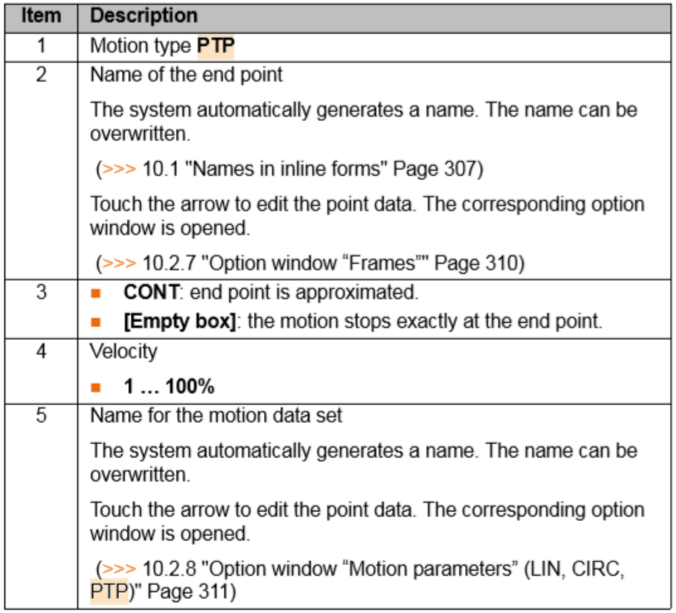
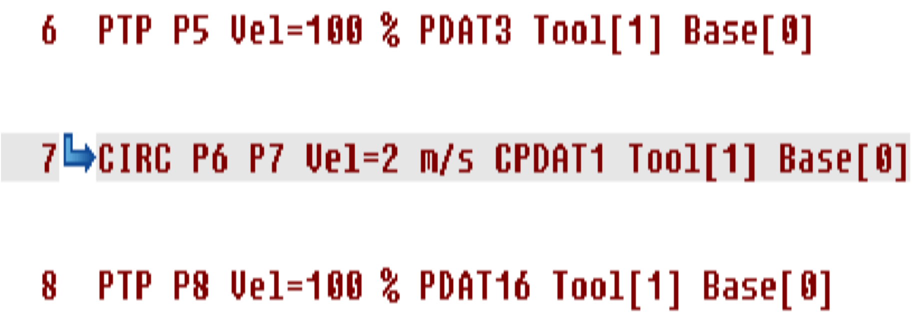


Fig. 10-1: Inline form for PTP motions



Item	Description
1	Motion type **PTP**
2	Name of the end point The system automatically generates a name. The name can be overwritten. (>>> 10.1 "Names in inline forms" Page 307) Touch the arrow to edit the point data. The corresponding option window is opened. (>>> 10.2.7 "Option window "Frames"" Page 310)
3	■ **CONT**: end point is approximated. ■ **[Empty box]**: the motion stops exactly at the end point.
4	Velocity ■ **1 ... 100%**
5	Name for the motion data set The system automatically generates a name. The name can be overwritten. Touch the arrow to edit the point data. The corresponding option window is opened. (>>> 10.2.8 "Option window "Motion parameters" (LIN, CIRC, PTP)" Page 311)

Example of PTP and Circular motion



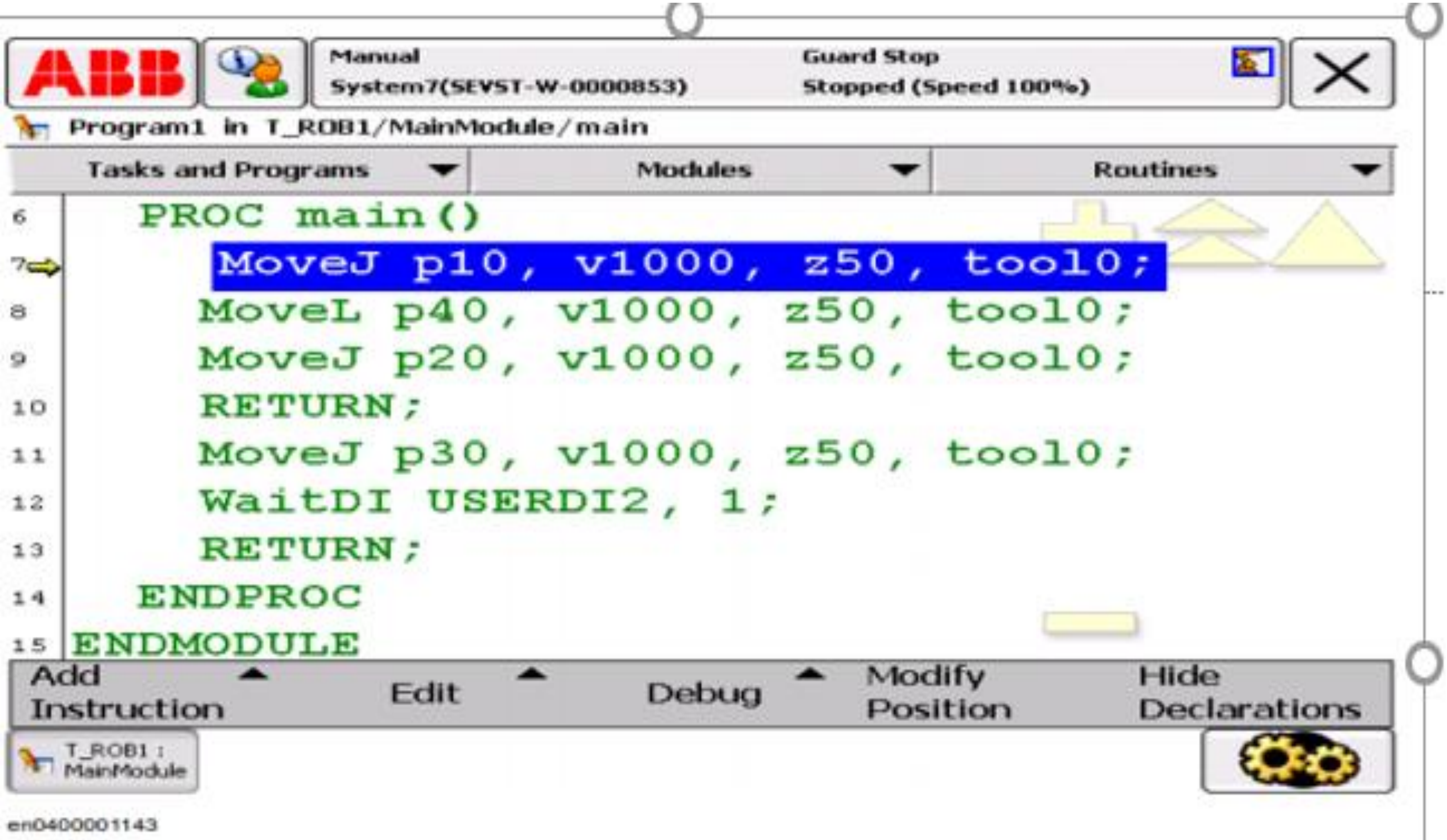
```
PTP P5 Vel=100 % PDAT3 Tool[1] Base[0]

CIRC P6 P7 Vel=2 m/s CPDAT1 Tool[1] Base[0]

PTP P8 Vel=100 % PDAT16 Tool[1] Base[0]
```

Resource: "KUKA system Software 8.3, section 8.2,8.4

Program Example ABB



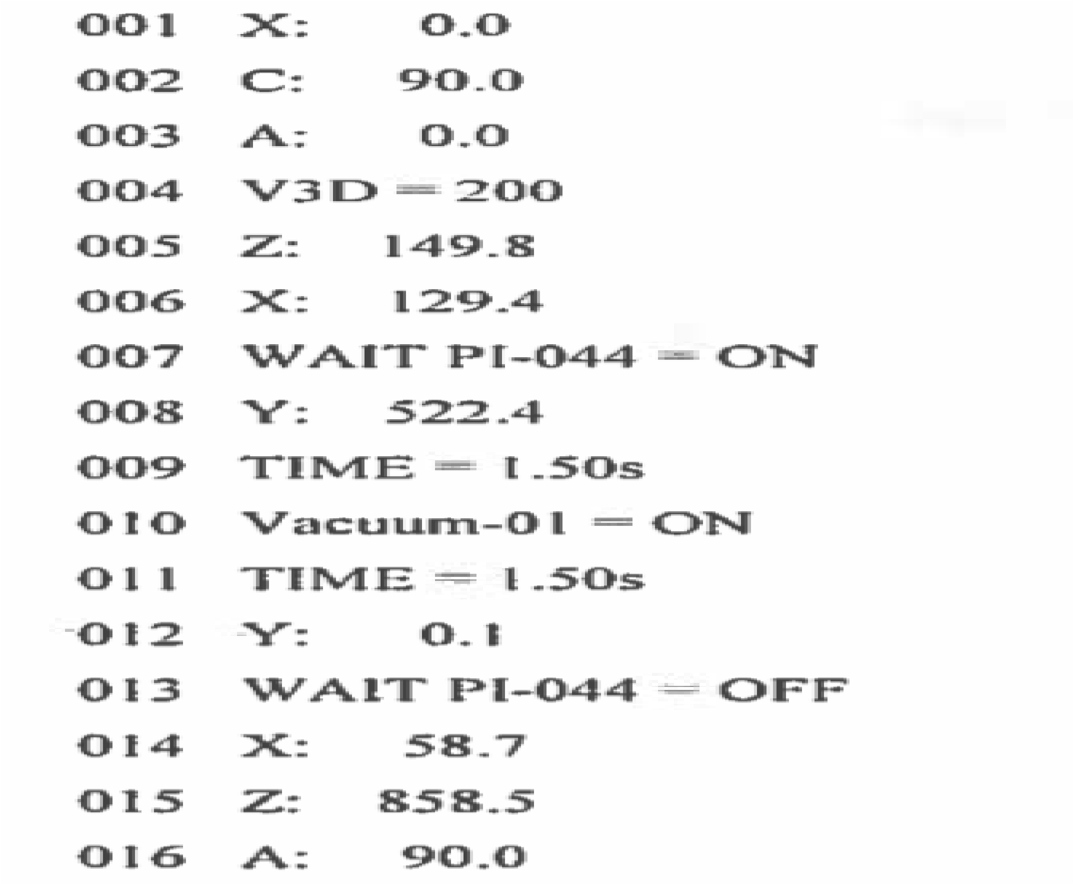
Manual
System7(SEVST-W-0000853)

Guard Stop
Stopped (Speed 100%)

Program1 in T_ROB1/MainModule/main

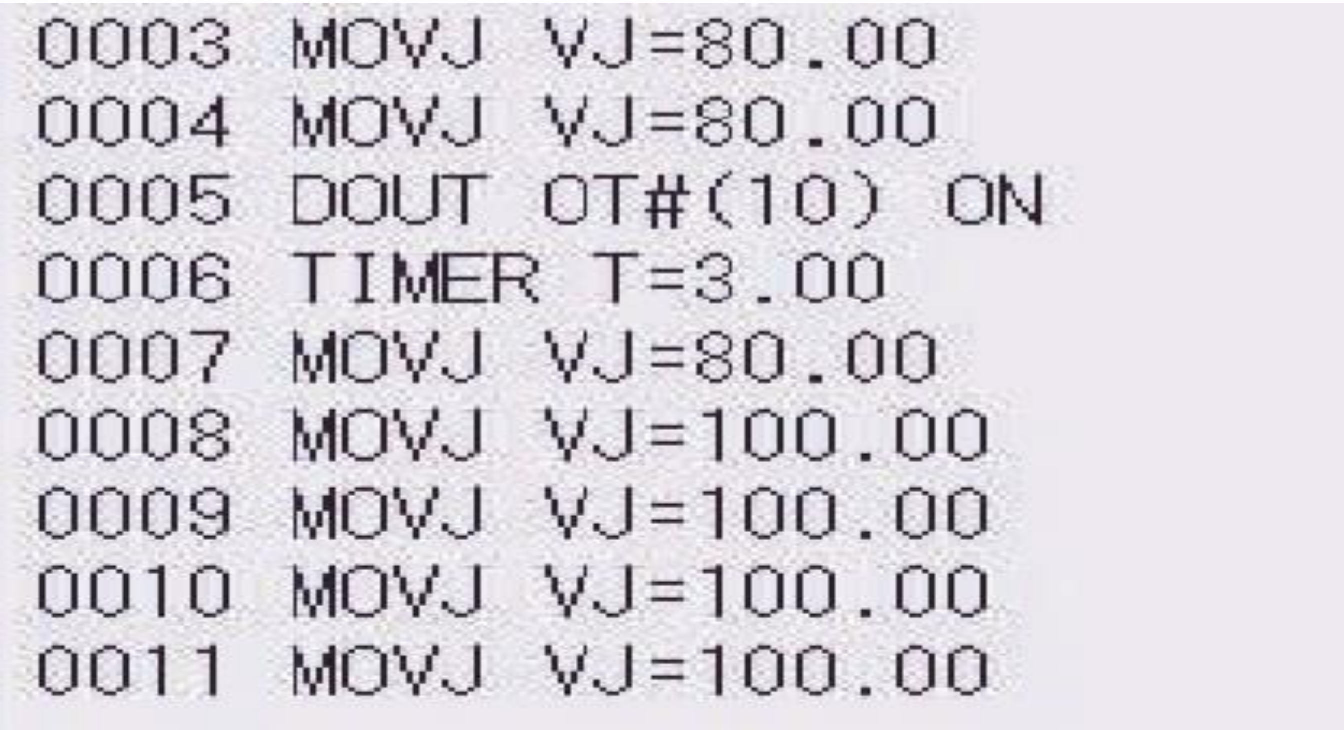
```
   PROC main()
      MoveJ p10, v1000, z50, tool0;
      MoveL p40, v1000, z50, tool0;
      MoveJ p20, v1000, z50, tool0;
      RETURN;
      MoveJ p30, v1000, z50, tool0;
      WaitDI USERDI2, 1;
      RETURN;
   ENDPROC
ENDMODULE
```

Program Example of Wittmann



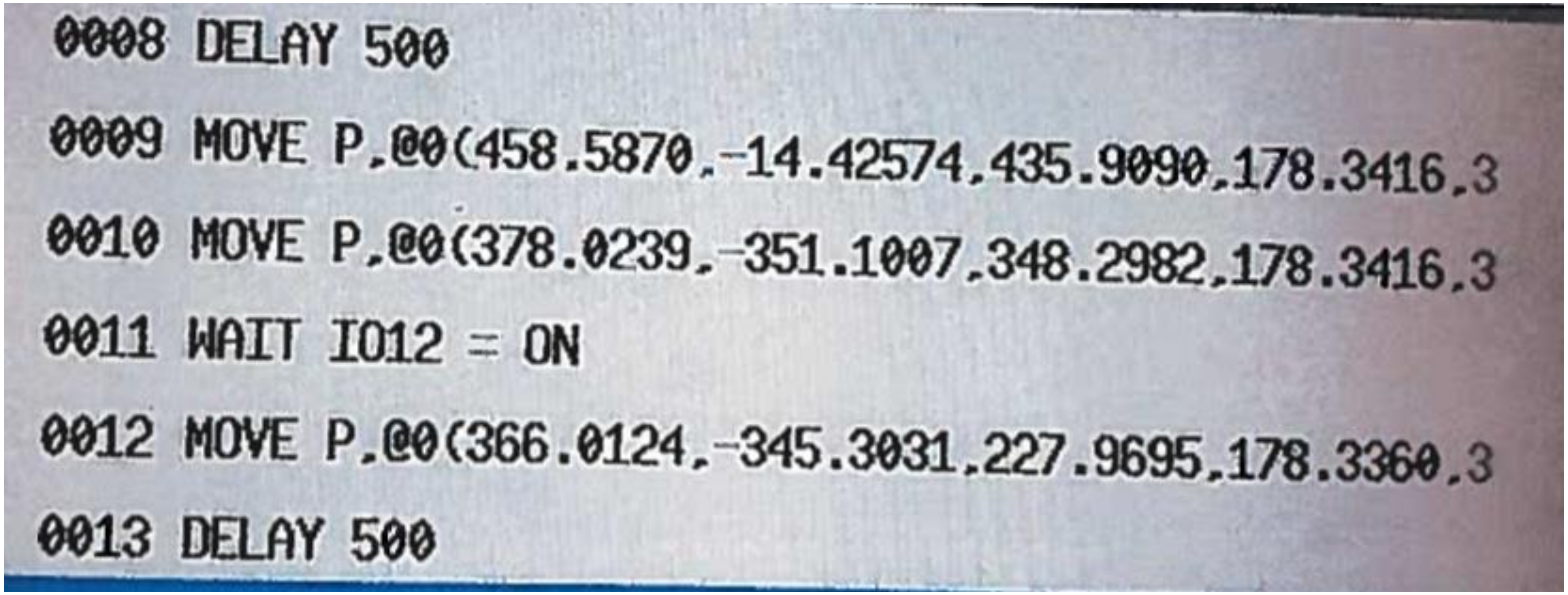
```
001   X:      0.0
002   C:     90.0
003   A:      0.0
004   V3D = 200
005   Z:    149.8
006   X:    129.4
007   WAIT PI-044 = ON
008   Y:    522.4
009   TIME = 1.50s
010   Vacuum-01 = ON
011   TIME = 1.50s
012   Y:      0.1
013   WAIT PI-044 = OFF
014   X:     58.7
015   Z:    858.5
016   A:     90.0
```

Program Example of MOTOMAN



```
0003 MOVJ VJ=80.00
0004 MOVJ VJ=80.00
0005 DOUT OT#(10) ON
0006 TIMER T=3.00
0007 MOVJ VJ=80.00
0008 MOVJ VJ=100.00
0009 MOVJ VJ=100.00
0010 MOVJ VJ=100.00
0011 MOVJ VJ=100.00
```

Program Example DENSO Point to Point



```
0008 DELAY 500
0009 MOVE P,@0(458.5870,-14.42574,435.9090,178.3416,3
0010 MOVE P,@0(378.0239,-351.1007,348.2982,178.3416,3
0011 WAIT IO12 = ON
0012 MOVE P,@0(366.0124,-345.3031,227.9695,178.3360,3
0013 DELAY 500
```

Input and Output commands

Robot uses input and output commands to interact with an environment.

Input signals are used to detect the environment.
Examples: sensor and switch signals

Output signal are used for giving signals and action to the environment. Examples: light bulb, actuators, grippers

Logic Commands

KUKA uses logic command for checking input signals and delaying the process

Logic Commands

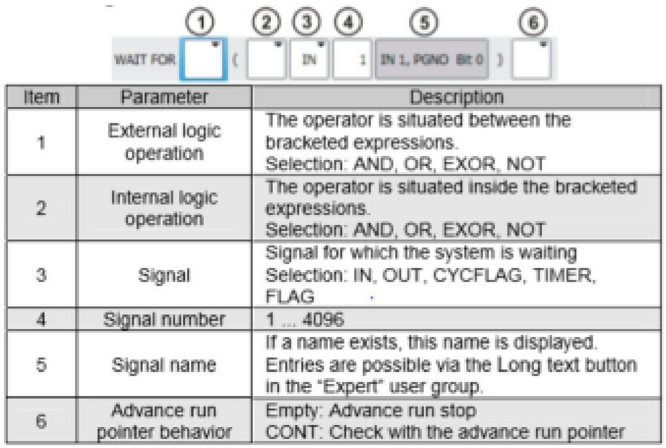
The logic command can be used from the command >logic menu option. Important logic commands are given below.

Wait

Wait is used to program a wait time. Robot motion is stopped for the given time.

Wait for Input

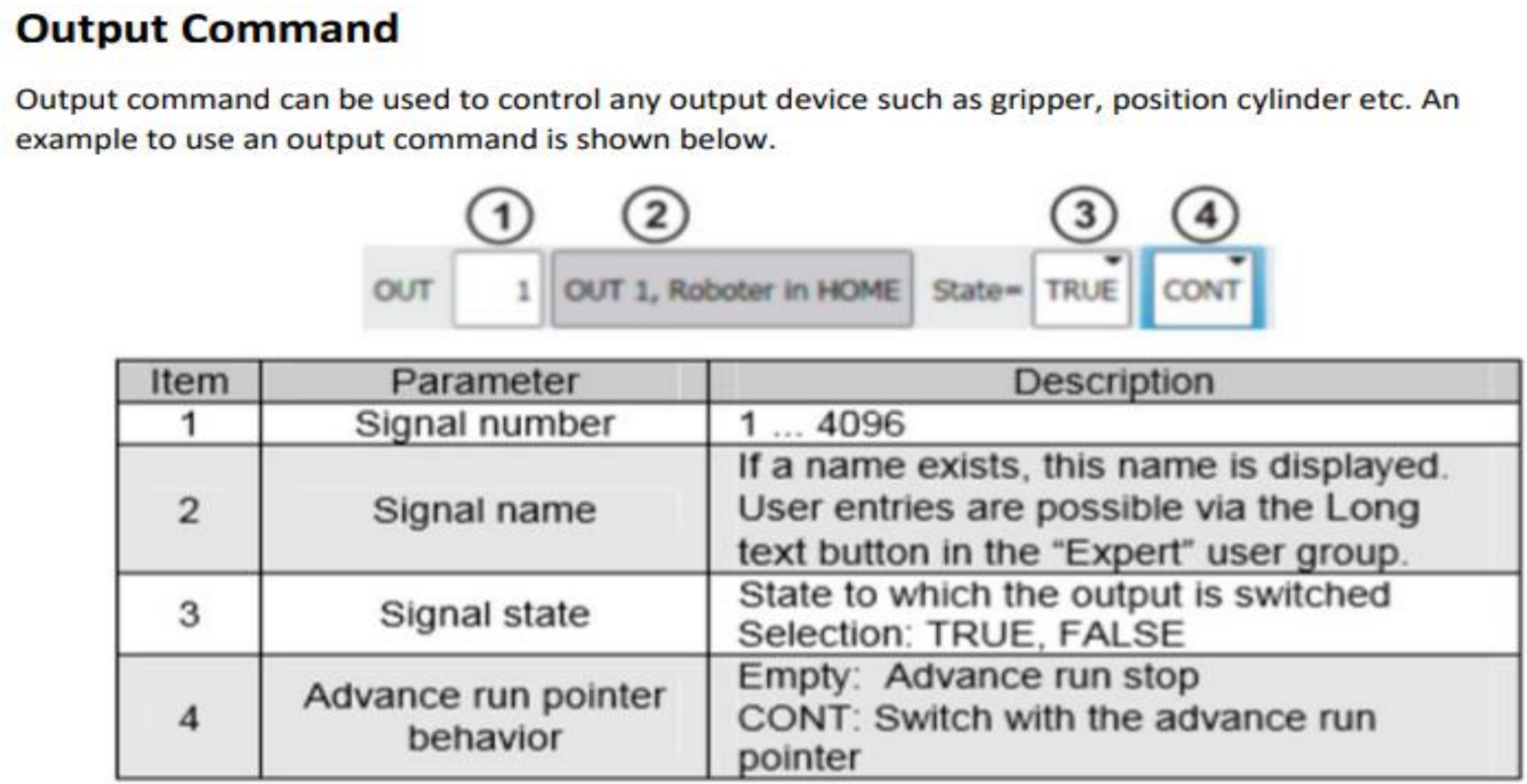
Wait for is signal dependent wait function. Robot will not execute for the next step until the signal is not received. Example of a wait for input function is given below



WAIT FOR (IN 1 IN 1, PGNO Bit 0)

Item	Parameter	Description
1	External logic operation	The operator is situated between the bracketed expressions. Selection: AND, OR, EXOR, NOT
2	Internal logic operation	The operator is situated inside the bracketed expressions. Selection: AND, OR, EXOR, NOT
3	Signal	Signal for which the system is waiting Selection: IN, OUT, CYCFLAG, TIMER, FLAG
4	Signal number	1 ... 4096
5	Signal name	If a name exists, this name is displayed. Entries are possible via the Long text button in the "Expert" user group.
6	Advance run pointer behavior	Empty: Advance run stop CONT: Check with the advance run pointer

Output Commands



Output Command

Output command can be used to control any output device such as gripper, position cylinder etc. An example to use an output command is shown below.

OUT 1 OUT 1, Roboter in HOME State= TRUE CONT

Item	Parameter	Description
1	Signal number	1 ... 4096
2	Signal name	If a name exists, this name is displayed. User entries are possible via the Long text button in the "Expert" user group.
3	Signal state	State to which the output is switched Selection: TRUE, FALSE
4	Advance run pointer behavior	Empty: Advance run stop CONT: Switch with the advance run pointer

Note: Parameter "4" should be blanked instead of continuous to execute the output command at correct position.