



## FANUC

- 1.1
- 1.2
- 1.3
- 1.4

1.1  
 FANUC (R--J3 i MODEL B )  
 . FANUC FANUC  
 R--J3 i MODEL B

5

	FANUC
	LED

	B.3
	A.1

” Warning ” ” Caution ” ” Note ”

” Warning ” ” Caution ” ” Note ”



” Note ”

” Caution ”

1.2

FANUC

1.3

--  
--  
--  
--  
--  
--  
--  
--  
--  
--  
--

FANUC

FANUC

--  
--  
--  
--

FANUC

FANUC

FANUC

--  
--

:

--  
--

FANUC



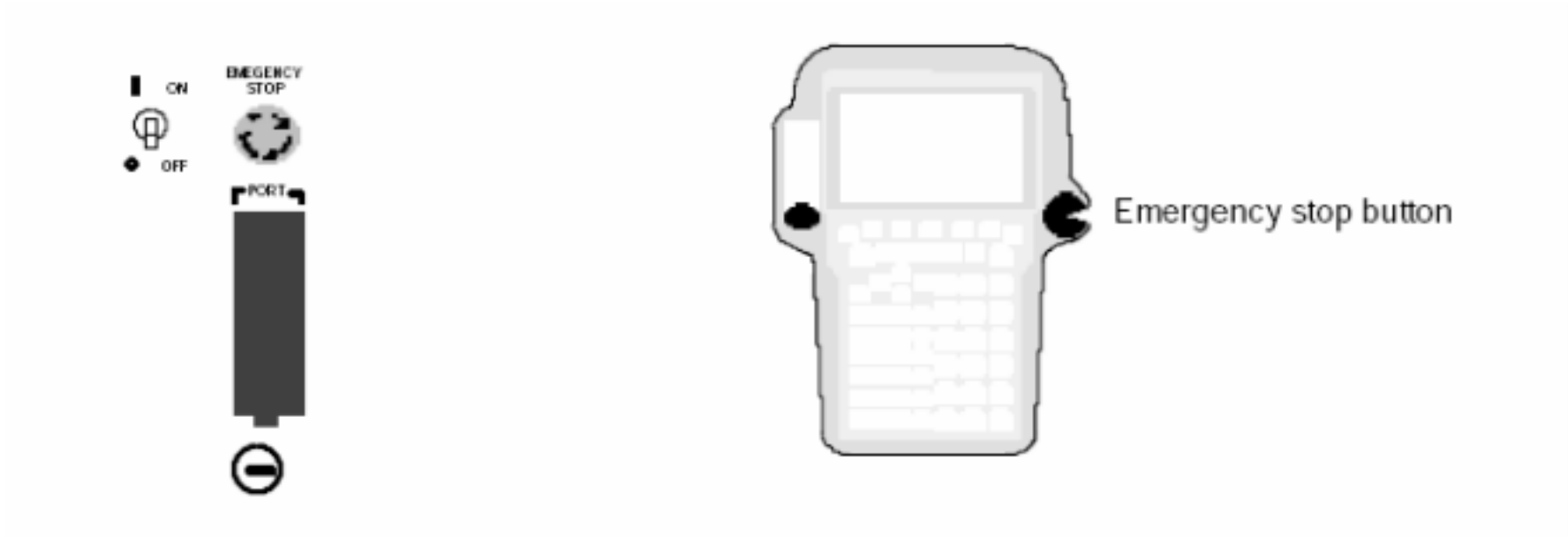
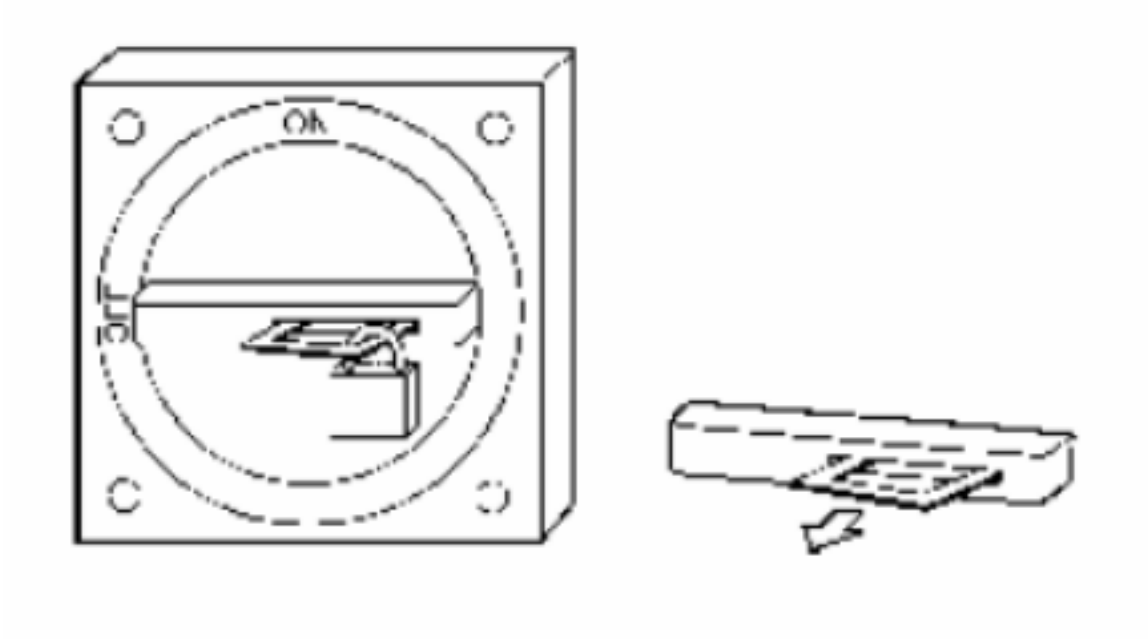
1.4

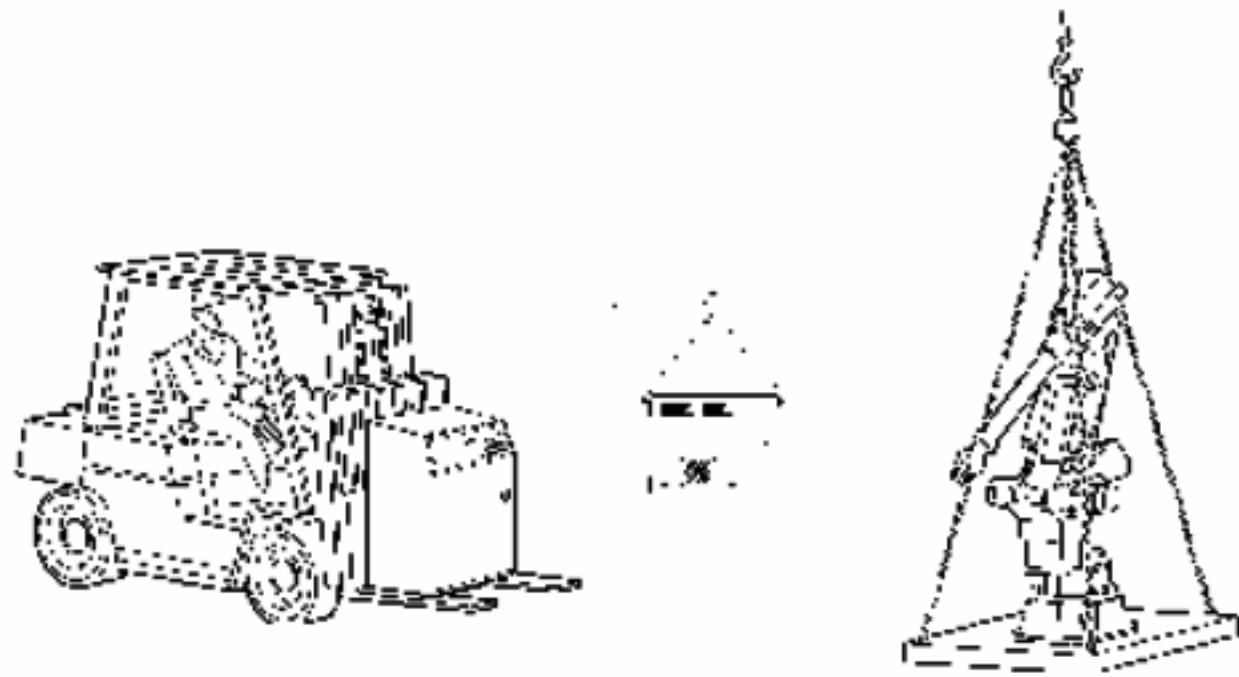
1--1.

**Teaching  
Do not enter**

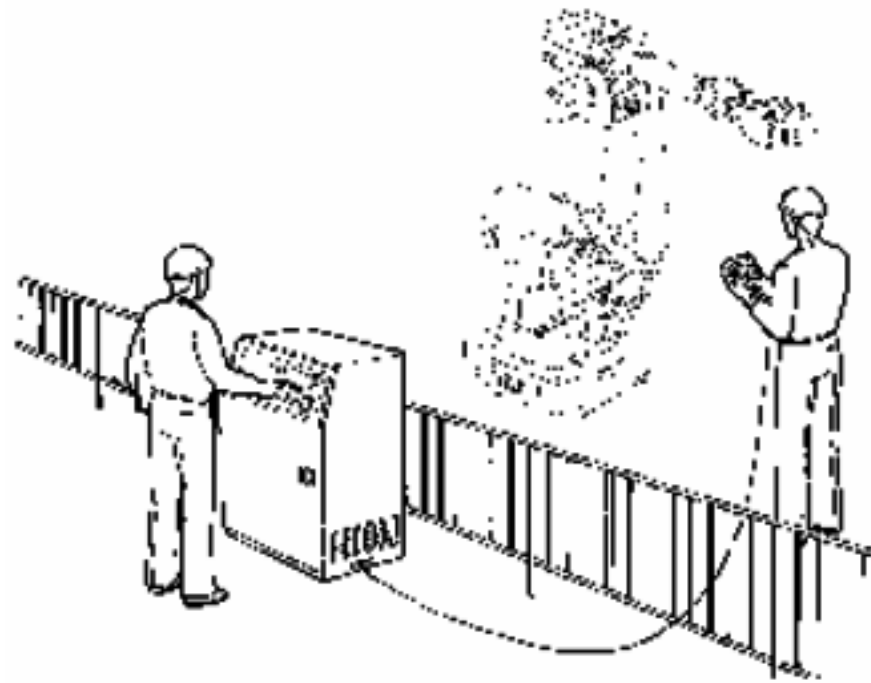


\*FENCE(     )  
\*SFSPD





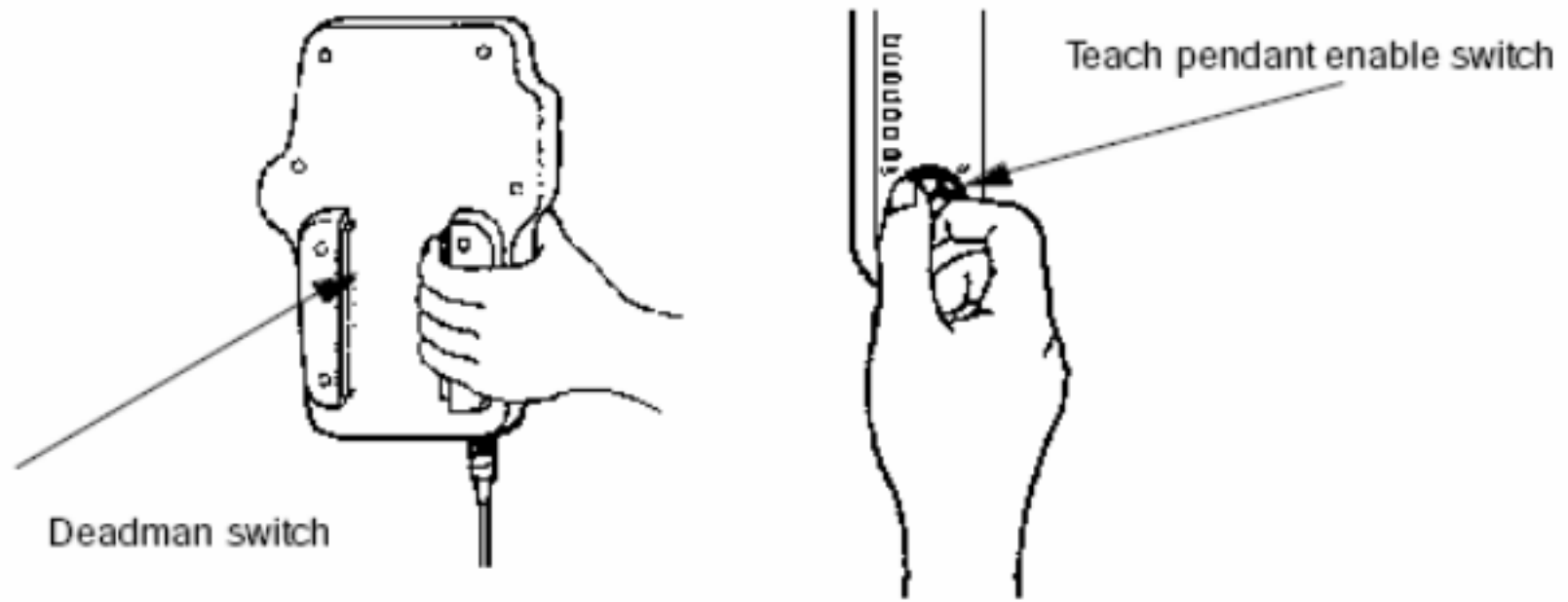
# FANUC



	FANUC	



Deadman



FANUC

FANUC  
FANUC

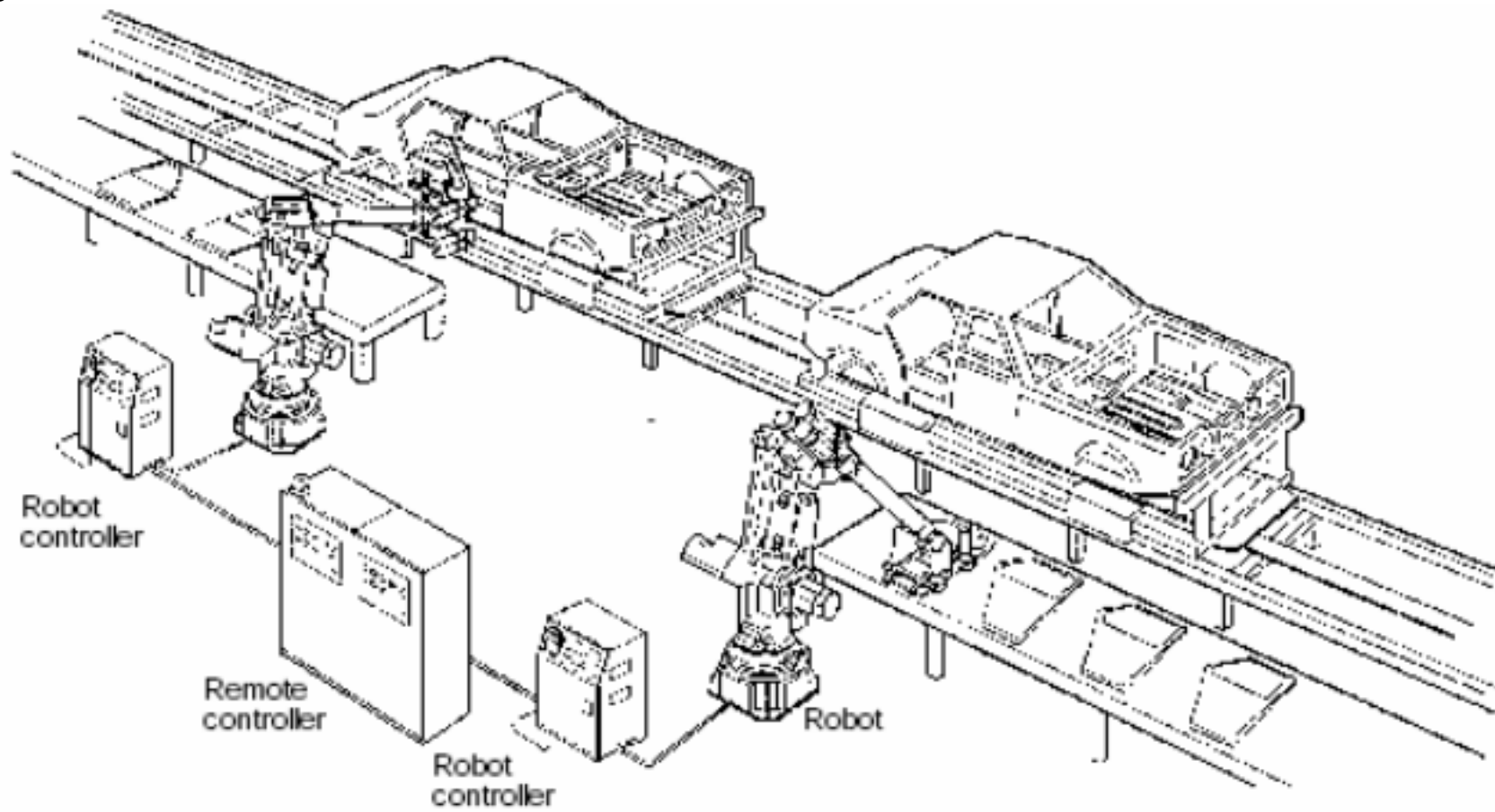
I/O

FANUC

(I/O)

. 2--1

Figure 2--1.



R--J2 Mate

FANUC SYSTEM R--J2 Mate

R--J2 Mate

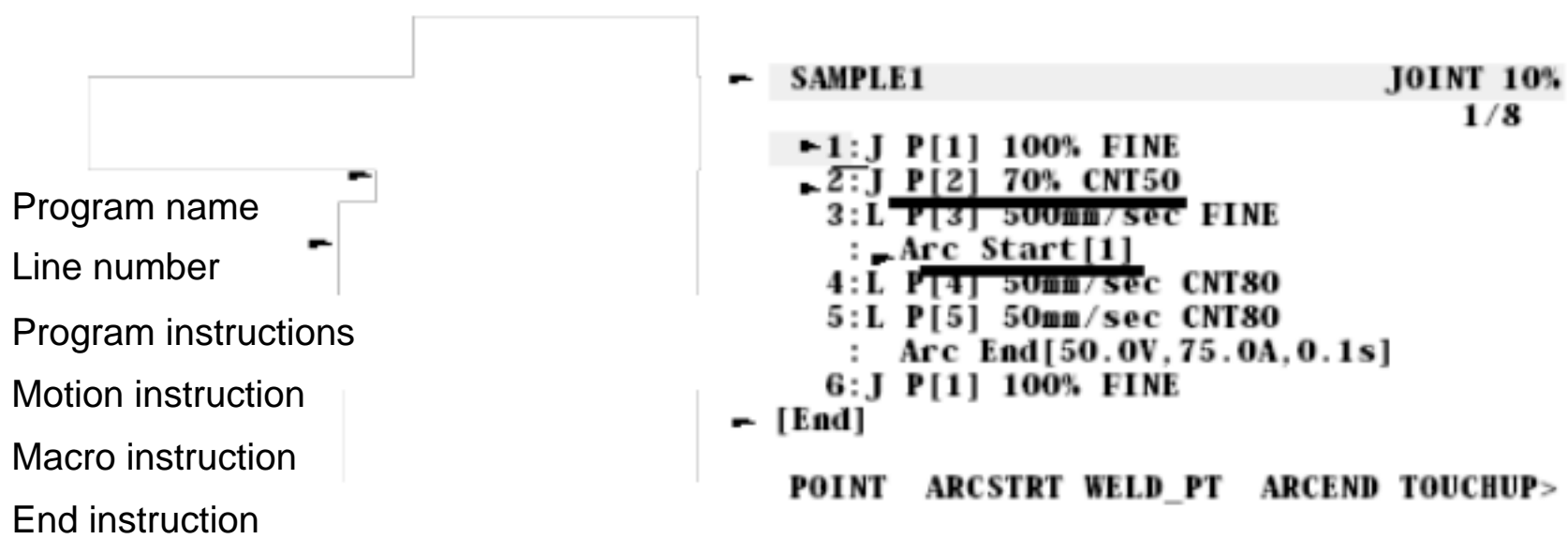
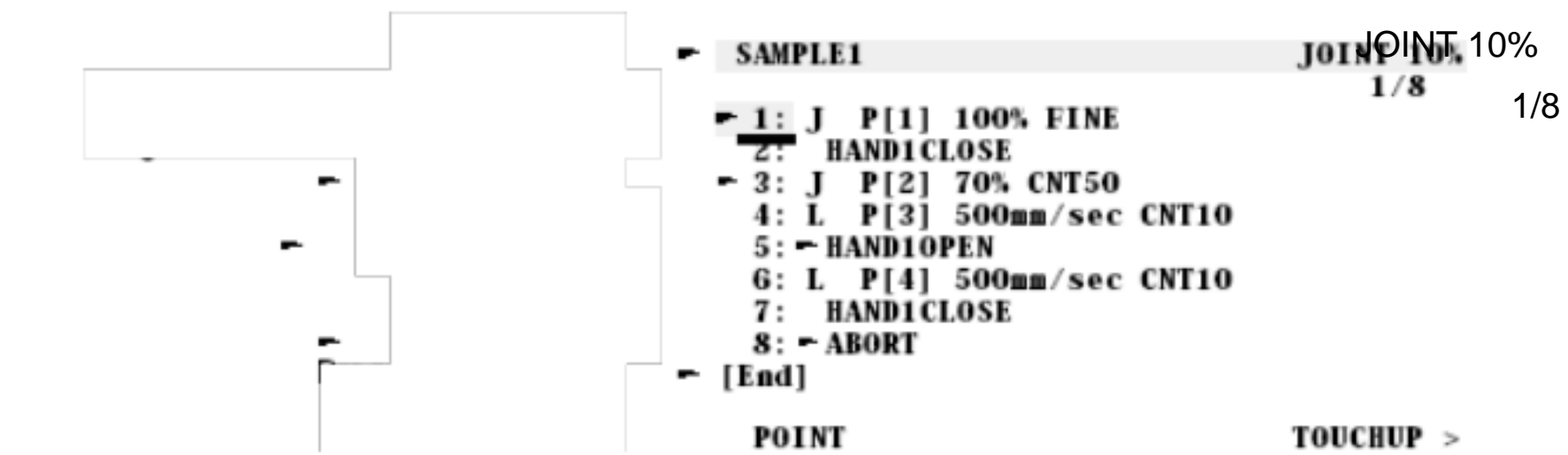


1.1

1.2

1.3

2—3



Program name  
 Line number  
 Program instructions  
 Motion instruction  
 Macro instruction  
 End instruction

1.4 ( )

6.2 6.3

1.5 ( )

( 3.11 6.8 )

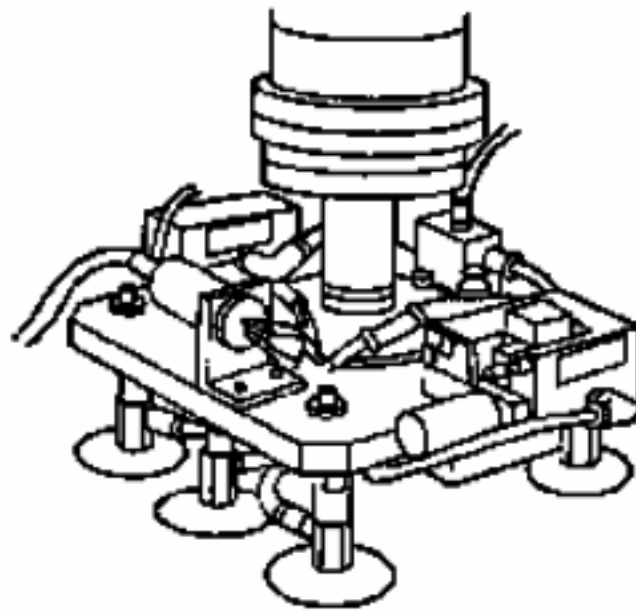
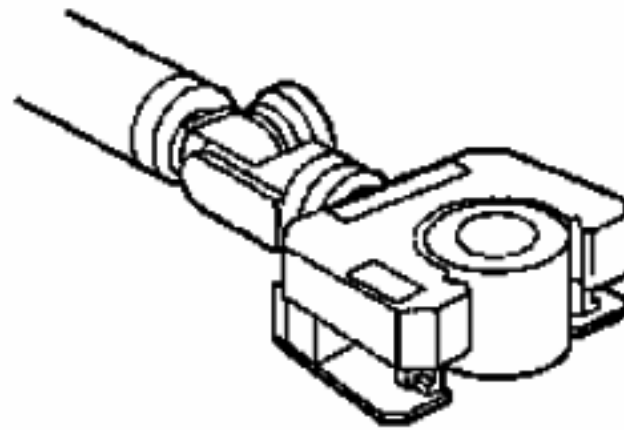
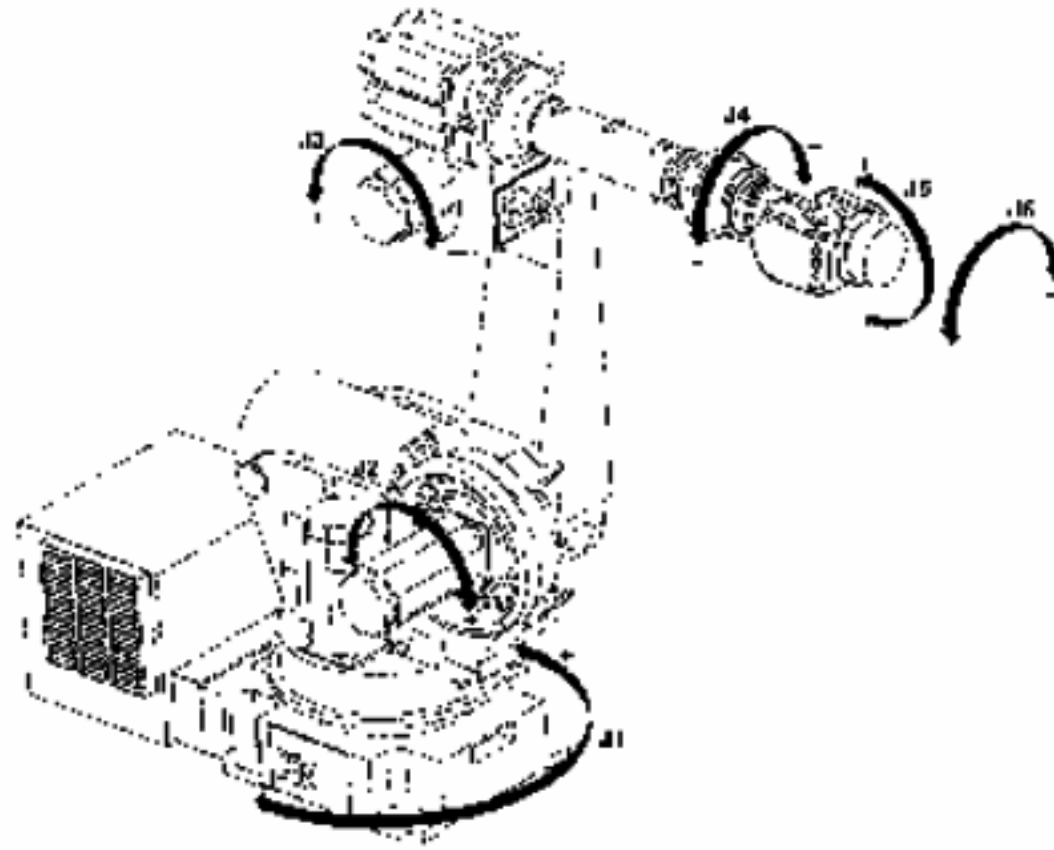
( 6.9 )

( \*\*\* )

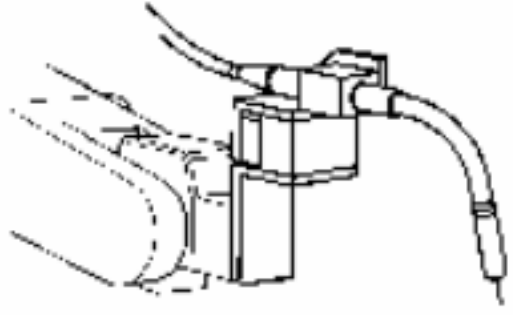
( 6.1 )

2

J1, J2, J3

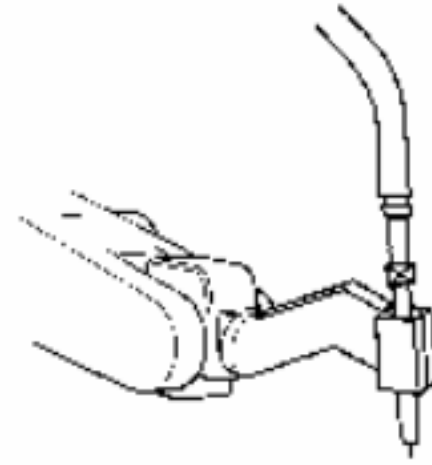


LR



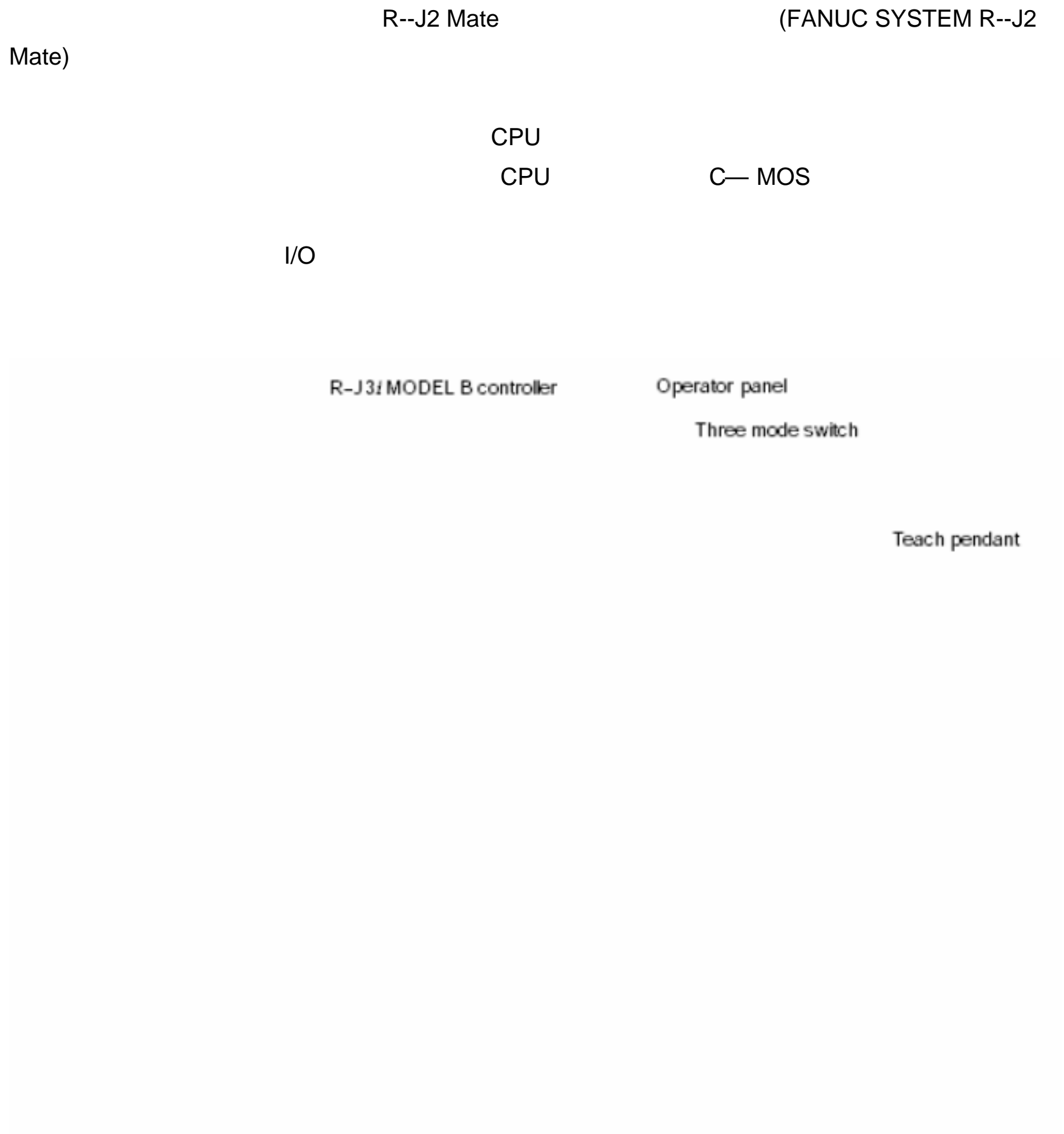
Curved torch for MAG welding

MAG



Straight torch for TIG welding

TIG



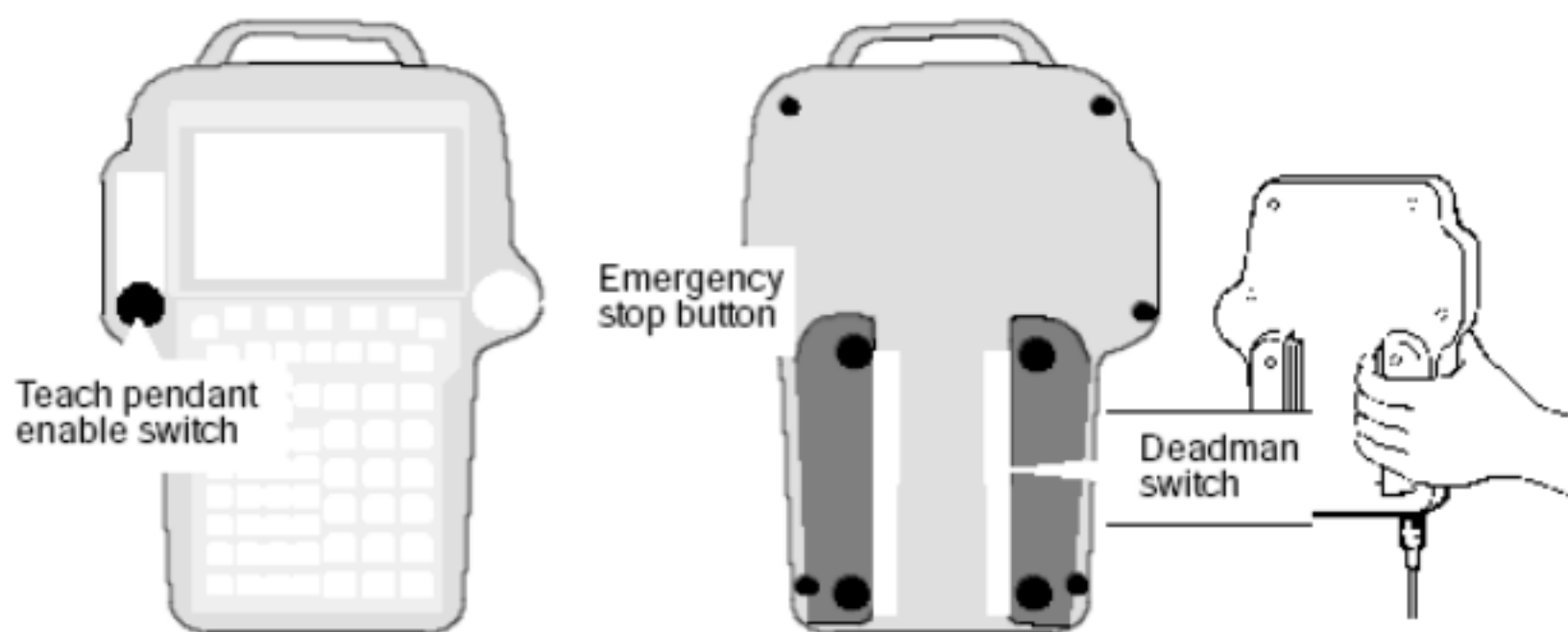
3.1

J

16            40  
11 LED  
11 LED            LR            LED)  
61  
61            LR            3



Teach pendant enable switch	
deadman switch	deadman
Emergency stop button	

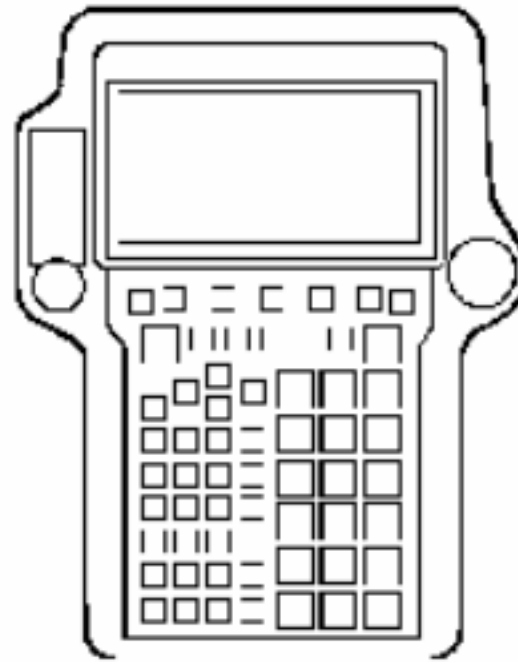
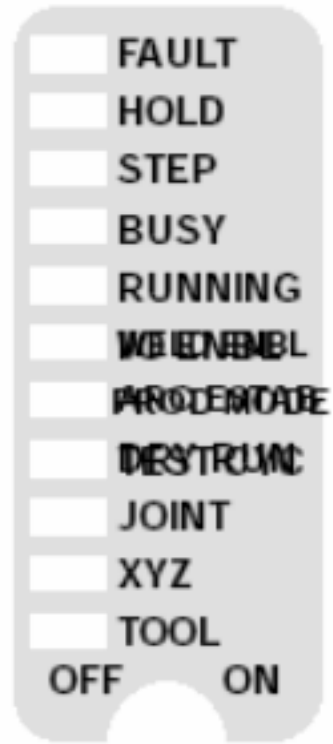


Teach pendant enable switch  
Emergency stop button  
Deadman switch

Deanman

LED

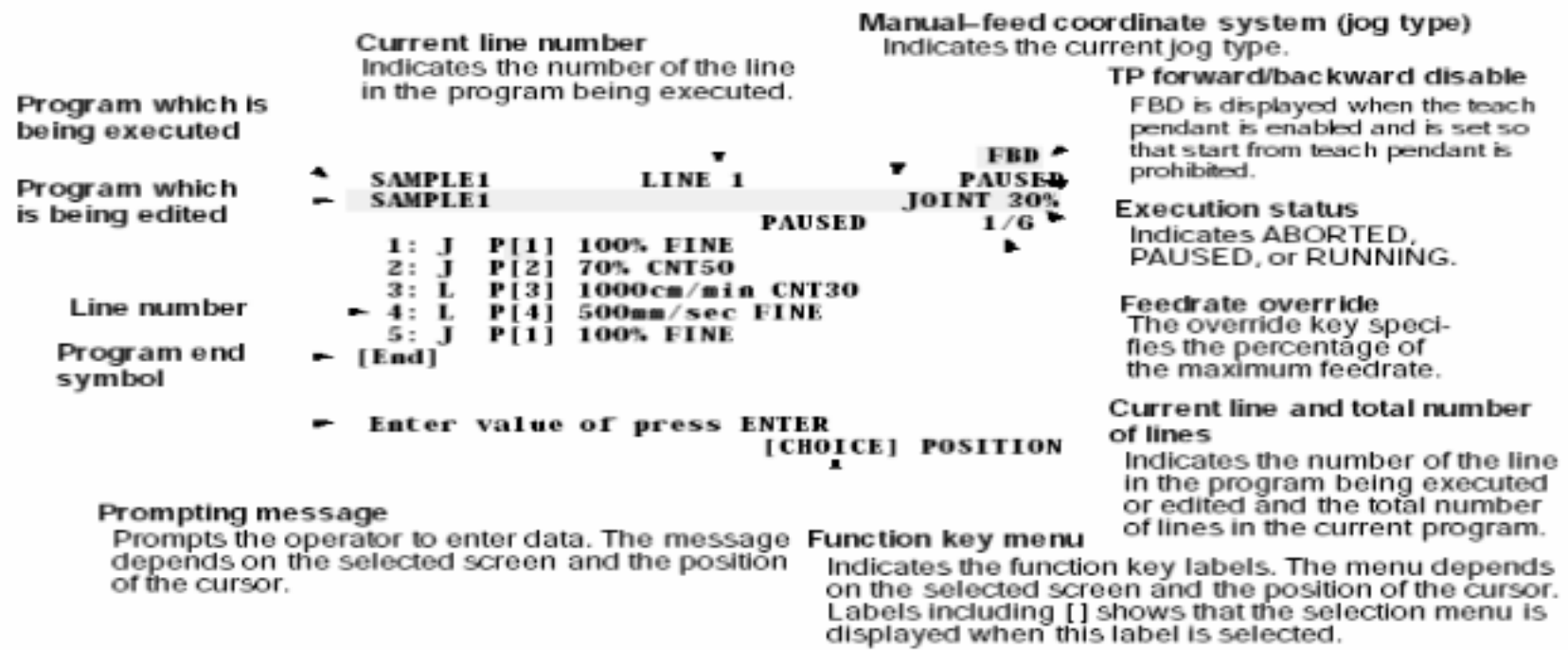
LED



LED

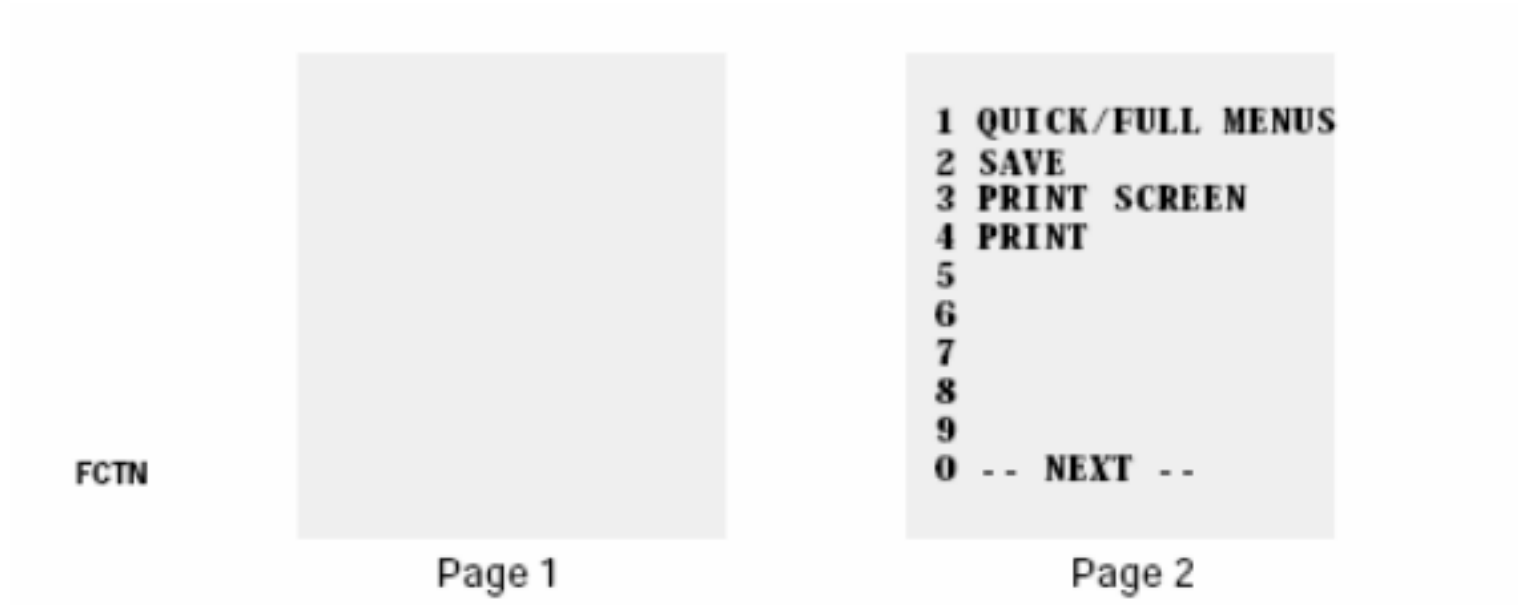
LED	
FAULT	
HOLD	HOLD HOLD
STEP	
BUSY	
RUNNING	
WELD ENBL	
ARC ESTAB	
DRY RUN	
JOINT	JOINT
XYZ	World frame      USER      JGFRM
TOOL	TOOL

3.1.1



LED	
UTILITIES	
TEST CYCLE	
MANUAL FCTNS	
ALARM	
I/O	
SETUP	
FILE	
USER	
SELECT	
EDIT	
DATA	
STATUS	
POSITION	
SYSTEM	

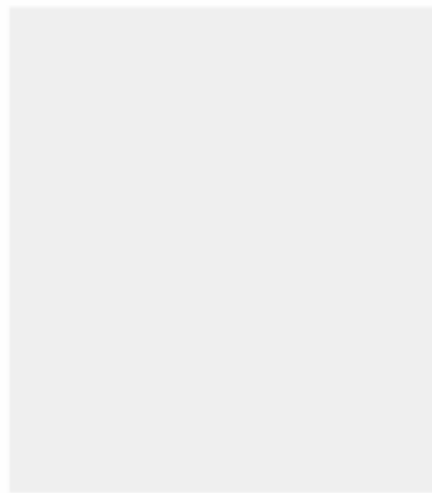




LED	
ABORT (ALL)	
Disable FWD/BWD	
CHANGE GROUP	
TOGGLE SUB GROUP	
TOGGLE WRIST JOG	
RELEASE WAIT	
QUICK/FULL MENUS	
SAVE	
PRINT SCREEN	

QUICK/FULL

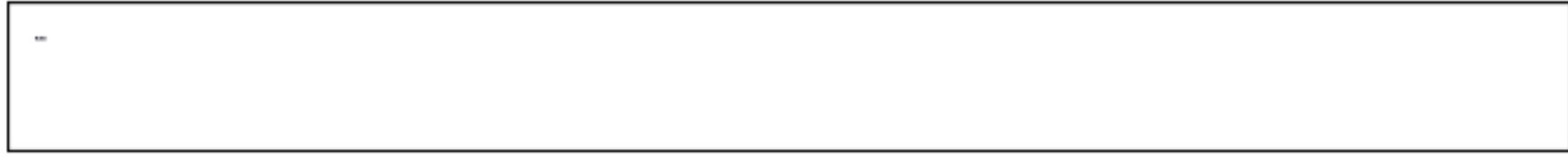
ALARM	/			
UTILITIES	/			
Setup	/			
DATA	/			
MANUAL FCTNS				
STATUS				
I/O	/	I/O	I/O	I/O
POSITION				



- 1 ALARM**
- 2 UTILITIES**
- 3 SET UP**
- 4 DATA**
- 5 MANUAL FCTNS**
- 6**
- 7 STATUS**
- 8**
- 9 POSITION**
- 0**

SELECT  
EDIT

3.2



RS--232C

Power--on/off button	
Emergency stop button	
Alarm release button	
User #1 and #2 buttons	
Three mode switch	

LED

LED	
Alarm	alarm release



3.3

CRT/KB

RS--232 —C

3.5

RS--232 —C

RS--232 —C

3.6

3.7

3.7

3.8

TCP

16

9

JOINT

Circular

Linear

Fine Cnt

3.9

3.10

6

16

3.1

I/O

I/O

I/O

I/O

[i]

I/O

[i]

I/O

I/O

I/O

0 = I/O  
1 16 = I/O A/B

SLOT I/O  
A/B I/O SLOT SLOT1 SLOT2

I/O

18 I/O

20 I/O

I/O

Process I/O PCB CA

24V

I/O

I/O

I/O

CRM2B

I/O

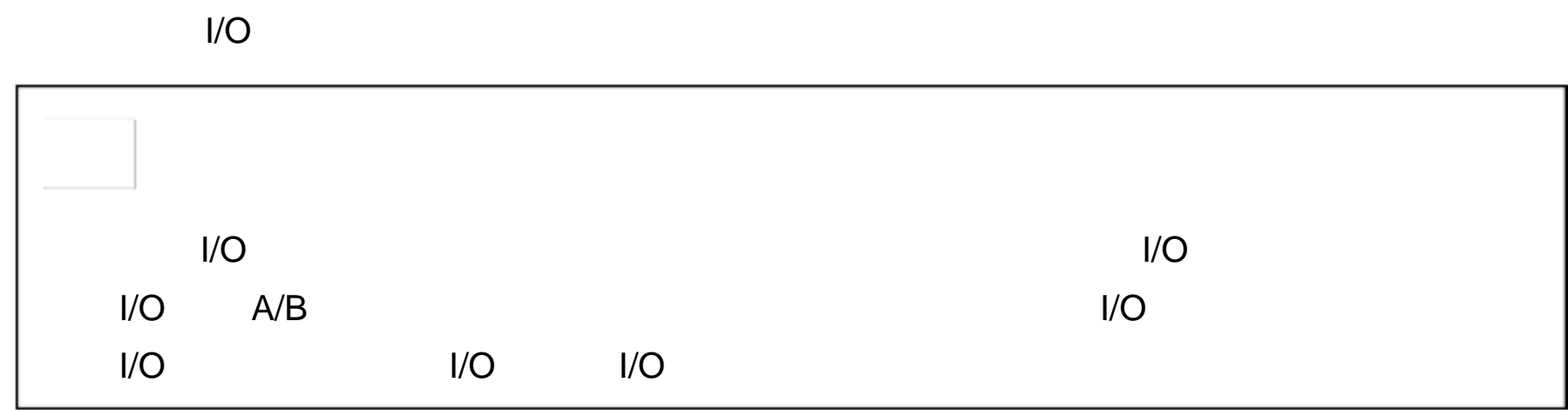
Printed circuit board for controlling the robot

Process I/O printed board CA / CB  
RACK 0 SLOT 1

	CRW2	CRW1	CRM2A	Peripheral equipment A1
JD1A		JD4A	CRM2B	Peripheral equipment A2
		JD4B	CRW1	
			CRW2	

Process I/O printed board DA

	CRM2A	Peripheral equipment A1
JD4A	CRM2B	Peripheral equipment A2
JD4B	CRM2C	Peripheral equipment A3
	CRM2D	Peripheral equipment A4
	CRM4A	Peripheral equipment B1
	CRM4B	Peripheral equipment B2



- 1 MENUS
  - 2 5 [I/O].
  - 3 F1 [TYPE].
  - 4 " Digital. "
- I/O

Digital I/O Selection Screen

- 4 ALARM
- 5 I/O
- 6 SETUP

MENUS

**Digital**

[TYPE]

I/O Digital Out			JOINT 30%	
#	SIM	STATUS		
DO[1]	U	OFF	[	]
DO[2]	U	OFF	[	]
DO[3]	U	OFF	[	]
DO[4]	U	OFF	[	]
DO[5]	U	OFF	[	]
DO[6]	U	OFF	[	]
DO[7]	U	OFF	[	]
DO[8]	U	OFF	[	]
DO[9]	U	OFF	[	]

[TYPE]    CONFIG    IN/OUT    ON    OFF

5 F3    IN/OUT



6    I/O,    F2,    CONFIG    ,    F2,    MONITOR.



I/O

Digital I/O Configuration Screen

[ TYPE ]	CONFIG	IN/OUT	I/O Digital Out		JOINT	10 %		
			#	RANGE	RACK	SLOT	START	STAT.
1	D0	[ 1- 20]	0	1	21	ACTIV		
2	D0	[ 21-512]	0	0	0	UNASG		

[ TYPE ] MONITOR IN/OUT DELETE HELP

[ TYPE ]	CONFIG	IN/OUT	I/O Digital Out		JOINT	30 %		
			#	RANGE	RACK	SLOT	START	PT
1	D0	[121-128]	0	1	5			
2	D0	[129-136]	0	1	13			
3	D0	[137-144]	0	1	21			
4	D0	[145-152]	0	1	29			
5	D0	[153-160]	0	1	37			
6	D0	[161-168]	0	1	0			
7	D0	[169-176]	0	1	29			
8	D0	[177-184]	0	1	37			
9	D0	[185-192]	0	0	0			

[ TYPE ] MONITOR IN/OUT DETAIL HELP >  
[ TYPE ] VERIFY

- 7 I/O
- 8 F2, MONITOR.
- 9 I/O NEXT F4, DETAIL
- 10
- 11
- 12 I/O F3, NEXT.
- 13 PREV
- 14

<input type="checkbox"/>
<input type="checkbox"/>
I/O

15 ON OFF

I/O  
(AI/AO) I/O

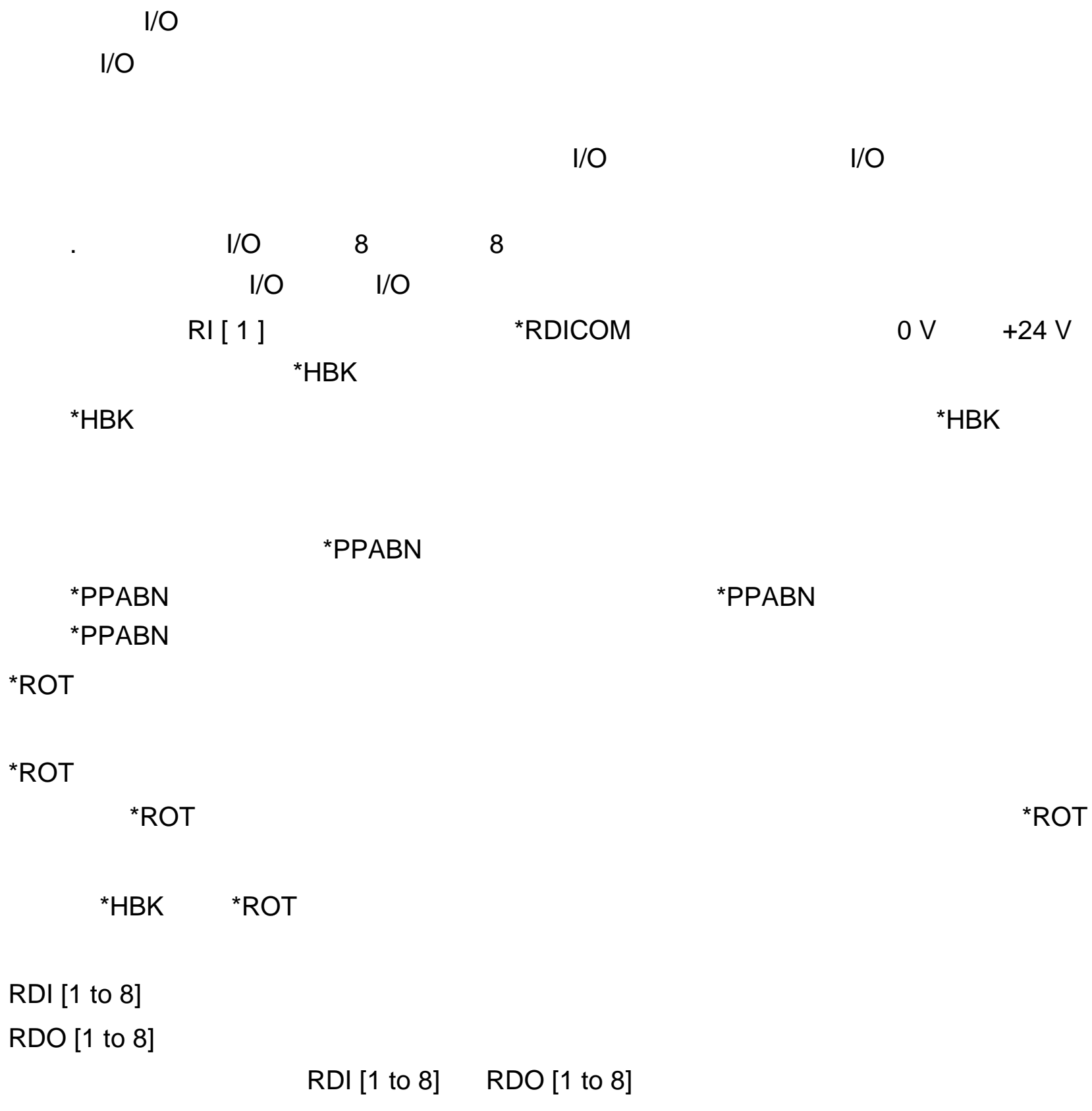
I/O



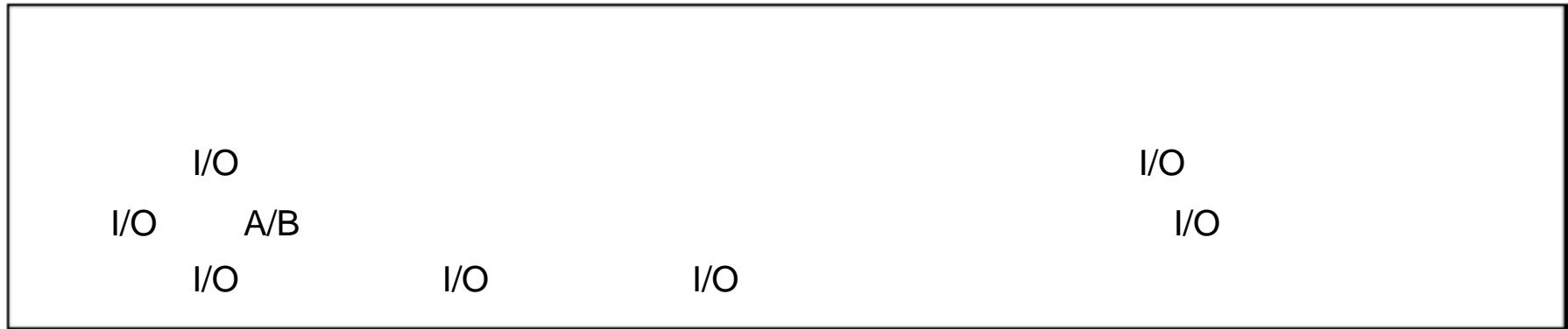
I/O  
0 = I/O  
1 16 = I/O A/B  
I/O A I/O B  
RACK1 2  
SLOT I/O A I/O  
SLOT  
I/O SLOT1 SLOT2  
A I/O  
B I/O SLOT DIP  
I/O I/O  
I/O I/O



I/O



I/O  
 I/O (UI/UO)  
 I/O  
 I/O  
 LR  
 CRM5  
 20  
 I/O  
 16  
 18  
 I/O  
 (SDI/SDO)  
 20



I/O  
 (\*HOLD,ENBL)

(SI[2])  
 ENBL  
 I/O\*SFSPD

CMDENBL

TRUE  
 I/O  
 I/O  
 " Enable UI signals "

I/O

RSR

RSR1 RSR8

(PNS)

RSR1

RSR8

PNSTROBF

START

PROD\_START input)

(CSTOPI )

-- ABORT CSTOPI FALSE

RSR

-- ABORT CSTOPI TURE

RSR

START

-- CONTINUE START FALSE

RSR

-- CONTINUE START TRUE

I/O

(SI[2])

I/O\*SFSPD

ENBL

\$RMT\_MASTER

CMDENBL

" START for CONTINUE only "

TRUE

START

(RSR)

8

(RSR1 RSR8)

1 RSR1 RSR8 RSR RSR  
 \$RSR1 \$RSR8 RSR1 RSR8 RSR  
 RSR UI UI

TRUE,

2 RSR 8 RSR RSR  
 RSR2

( ) = (RSR2 ) + ( )

RSR + ( )  
 RSR + ( ) RSR0121m 4  
 RSR121  
 \$SHELL\_CFG.\$JOB\_BASE RSR

3 RSR1 RSR8 RSR ACK1 ACK8  
 ACK1 ACK8 RSR

4

RSR

5 CSTOPI

PNS

PNS [6 (SETUP). RSR/PNS]. PNS

Program select mode	RSR PNS
Base number	PNS PNS
Acknowledge function	RSR ACK1 ASK8
Acknowledge pulse width	PNS (SNACK) ACK1 ASK8

PNS

1 MENU .

2 6 (SETUP). "

3 F1 TYPE.

4 RSR/PNS. RSR/PNS

5 " Program select mode F4 [CHOICE] RSR, F3

DETAIL.

PNS

PNS Setting Screen

5 I/O  
6 SETUP  
7 FILE

1/2

PNS  
1 Base number [ 100]  
2 Acknowledge pulse width (msec) [ 200]

RSR/PNS

[TYPE]

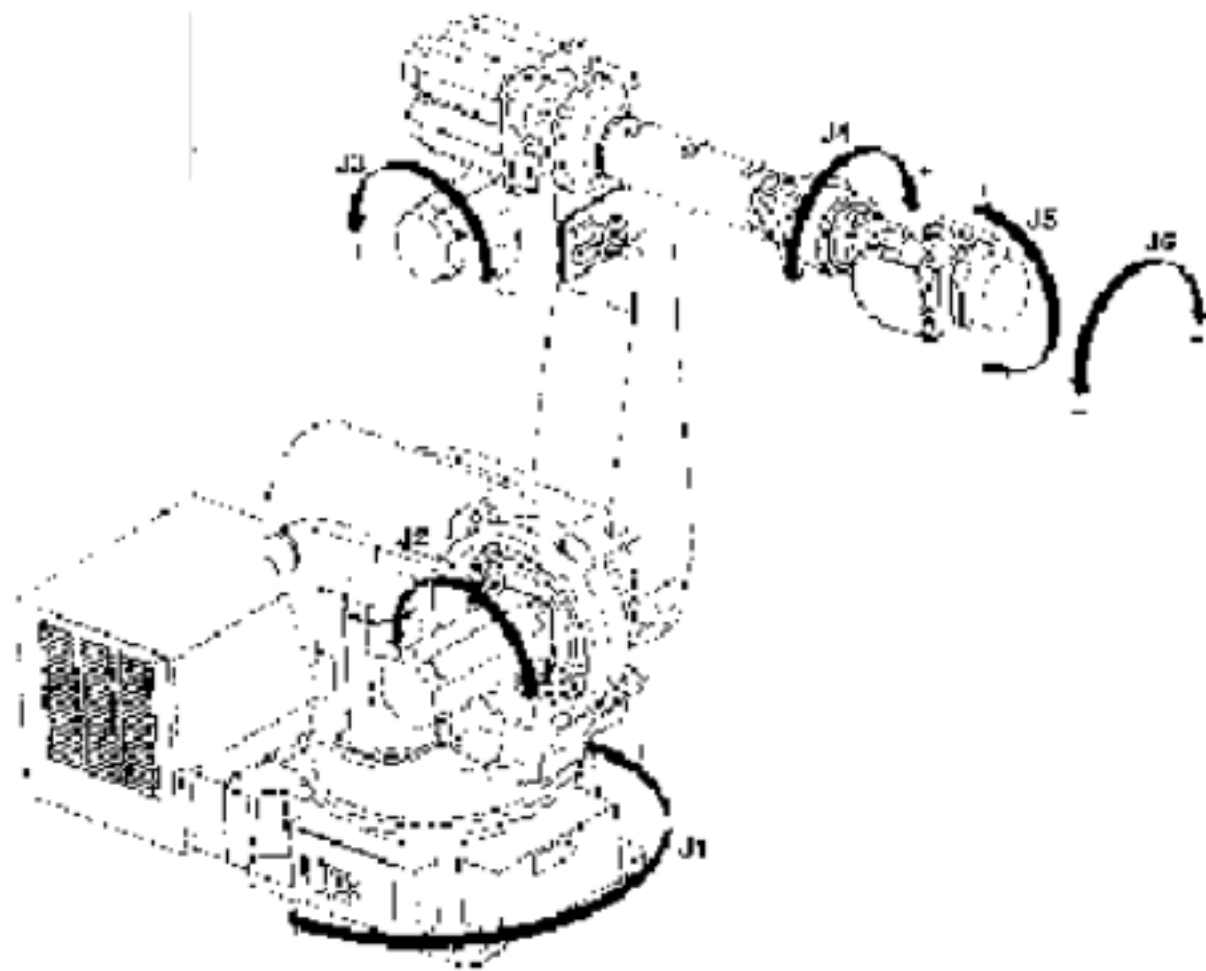
[TYPE]

PNS RSR

F4

6

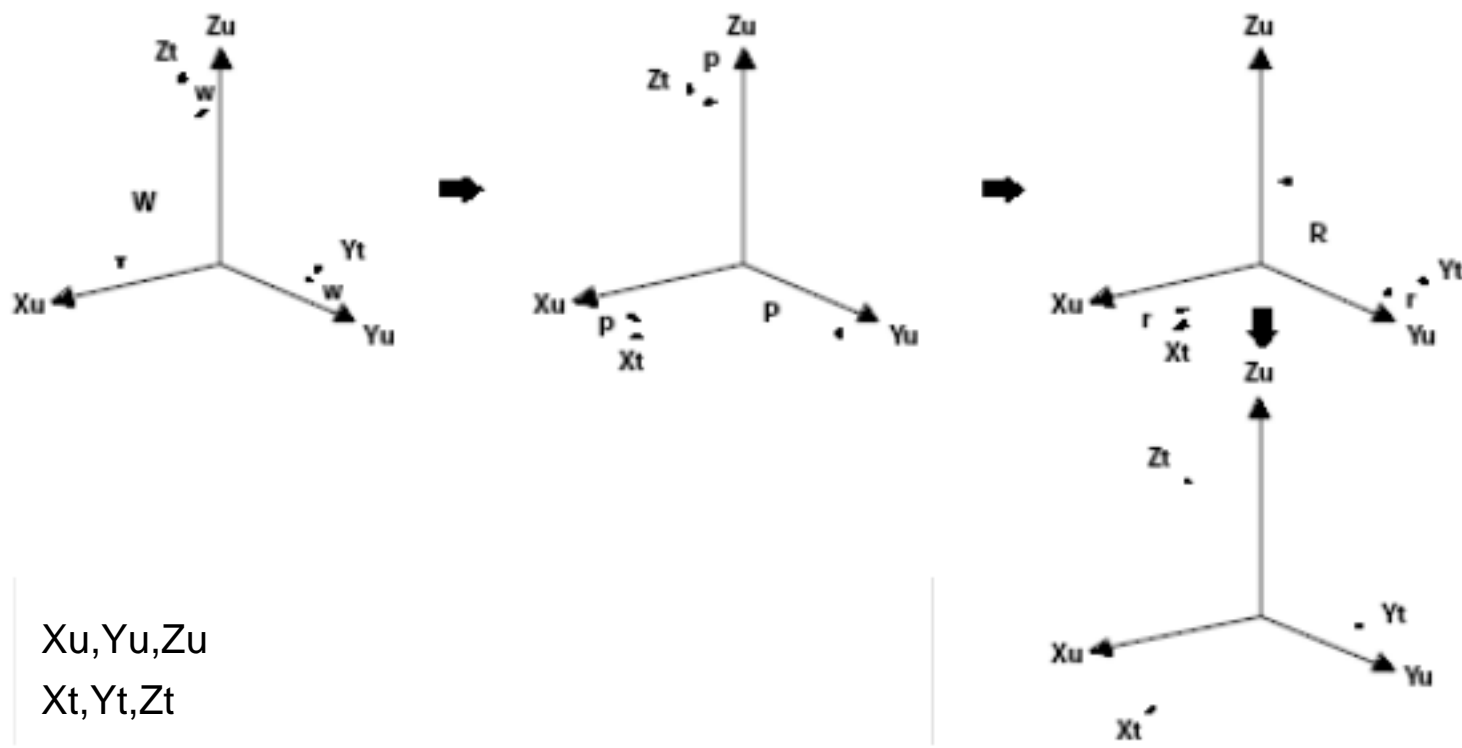
7 RSR PNS



$X_u, Y_u, Z_u$        $X, Y, Z$   
 $(w, p, r)$        $(w, p, r)$

3--28. (w, p, r)

Figure 3--28. Meaning of (w, p, r)

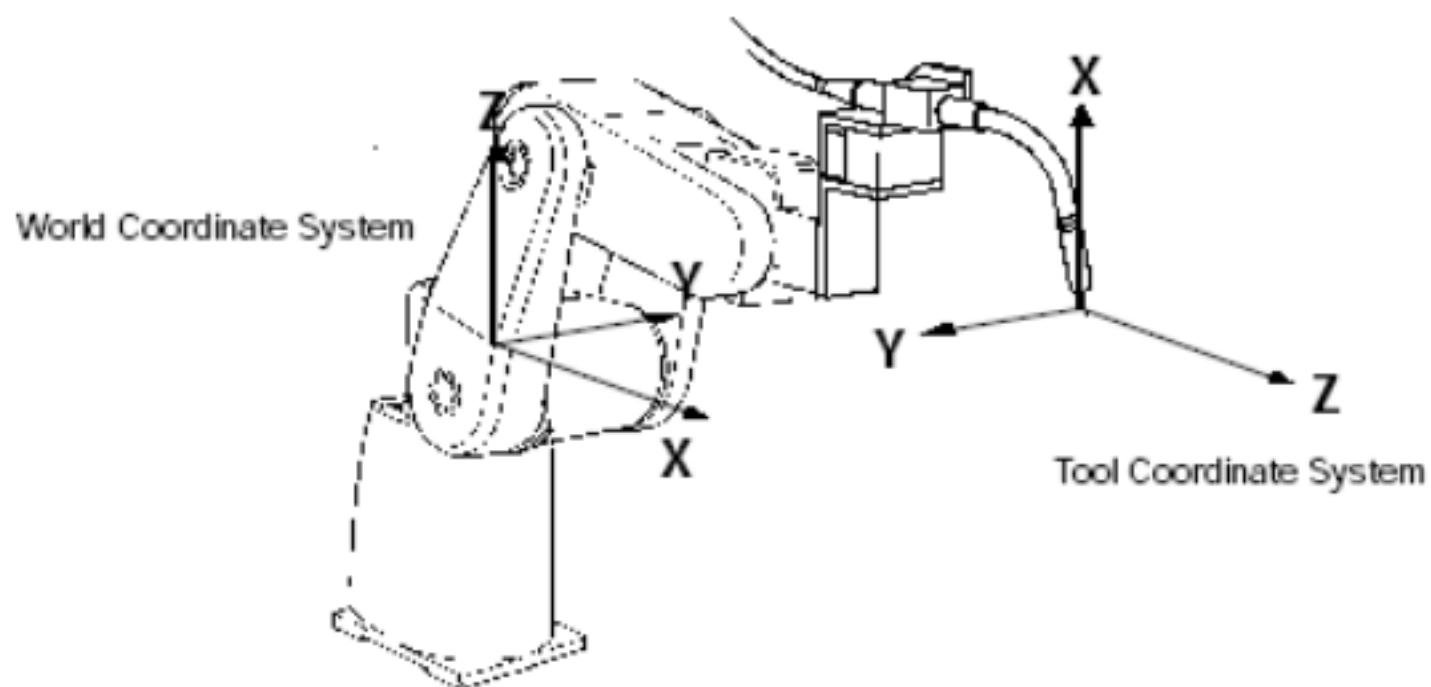
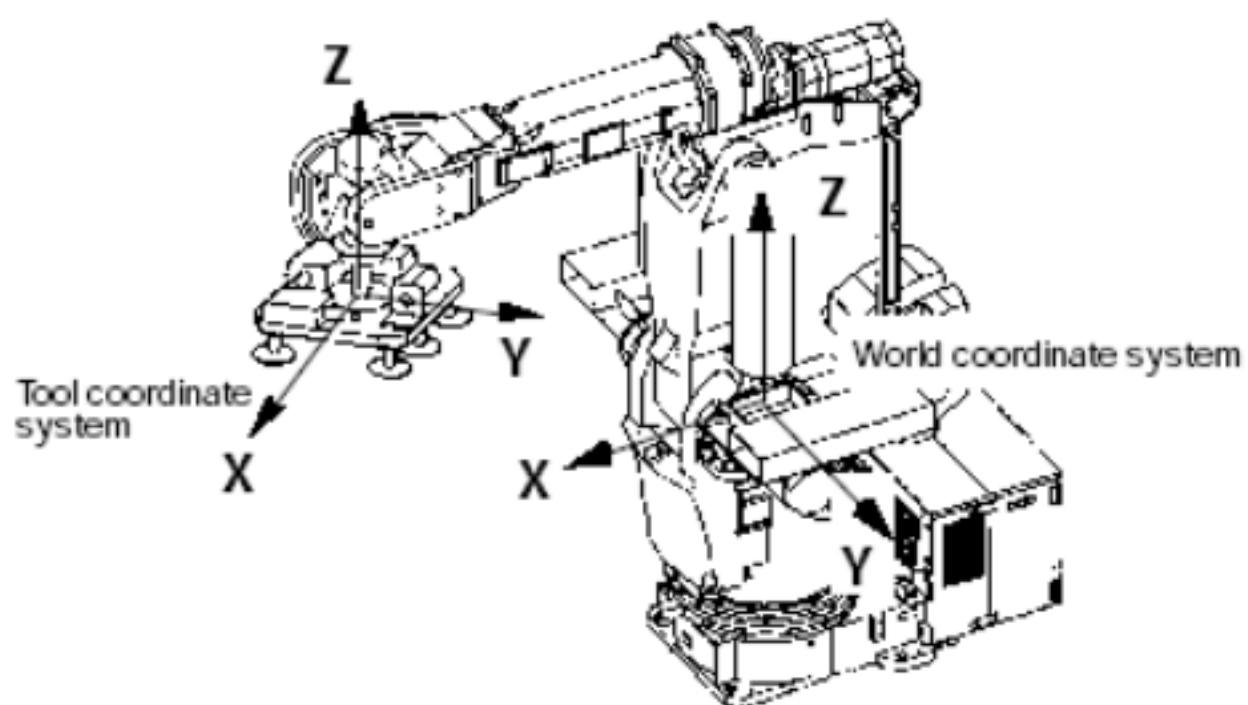


$X_u, Y_u, Z_u$   
 $X_t, Y_t, Z_t$



( )

(TCP)





TCP Z

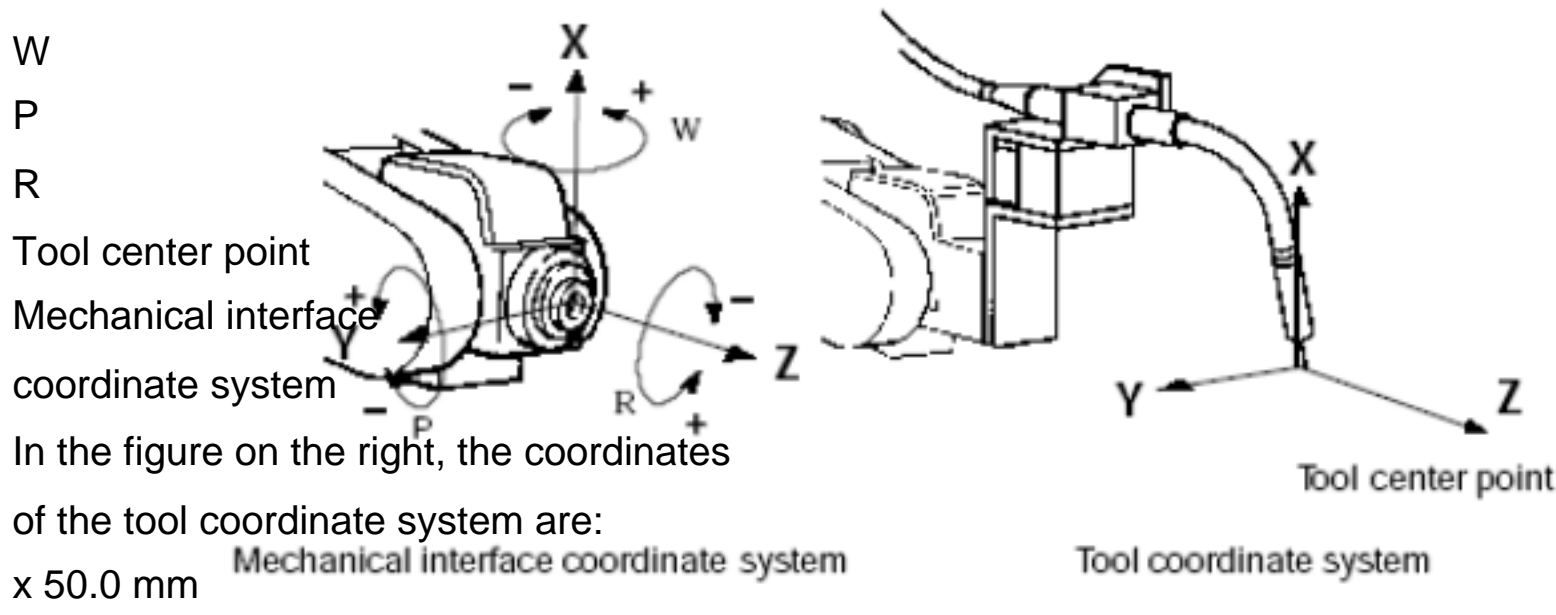
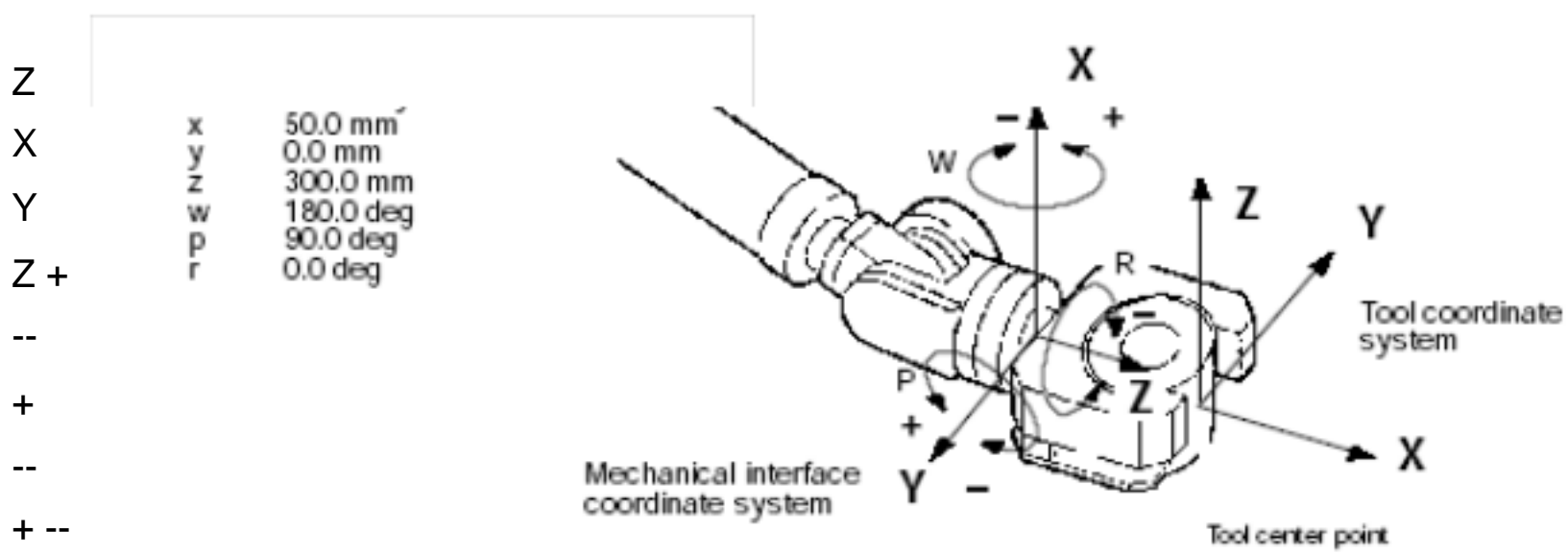
TCP

(x, y, z)

TCP (w, p, r)

X, Y Z TCP W, P R

X--, Y -- Z--

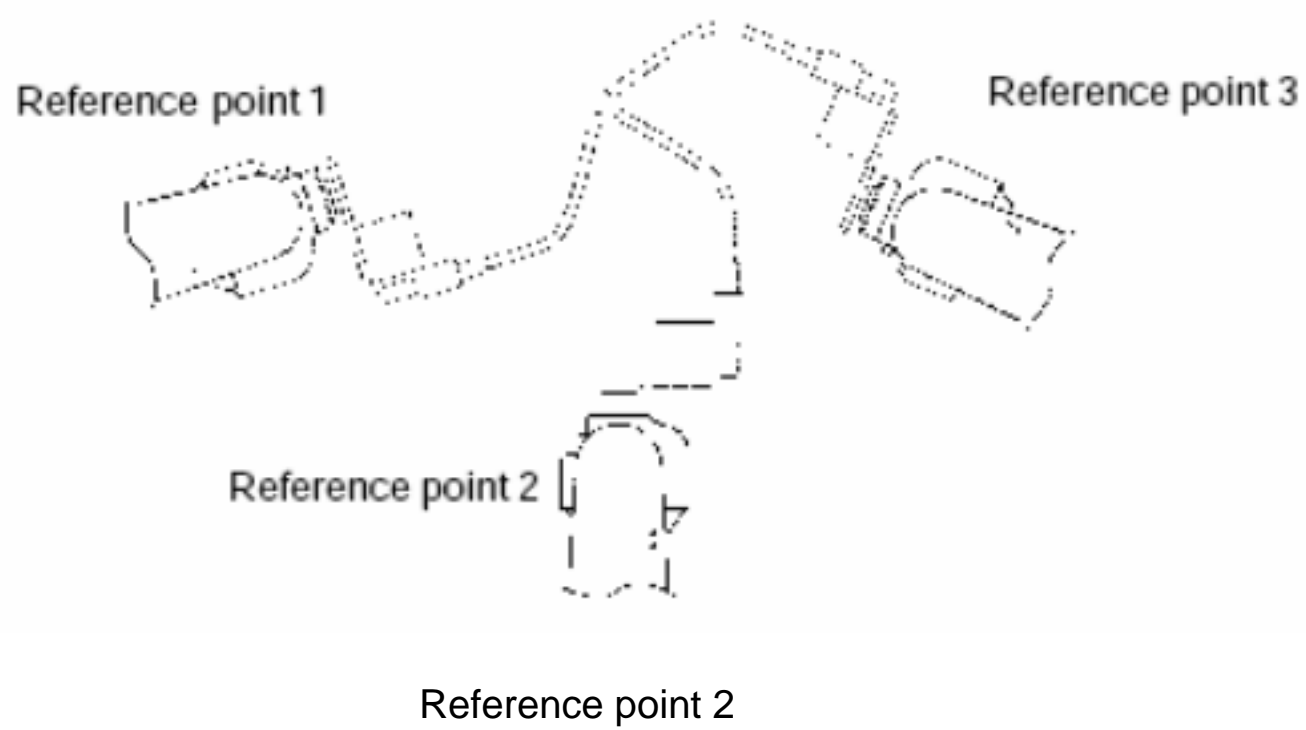
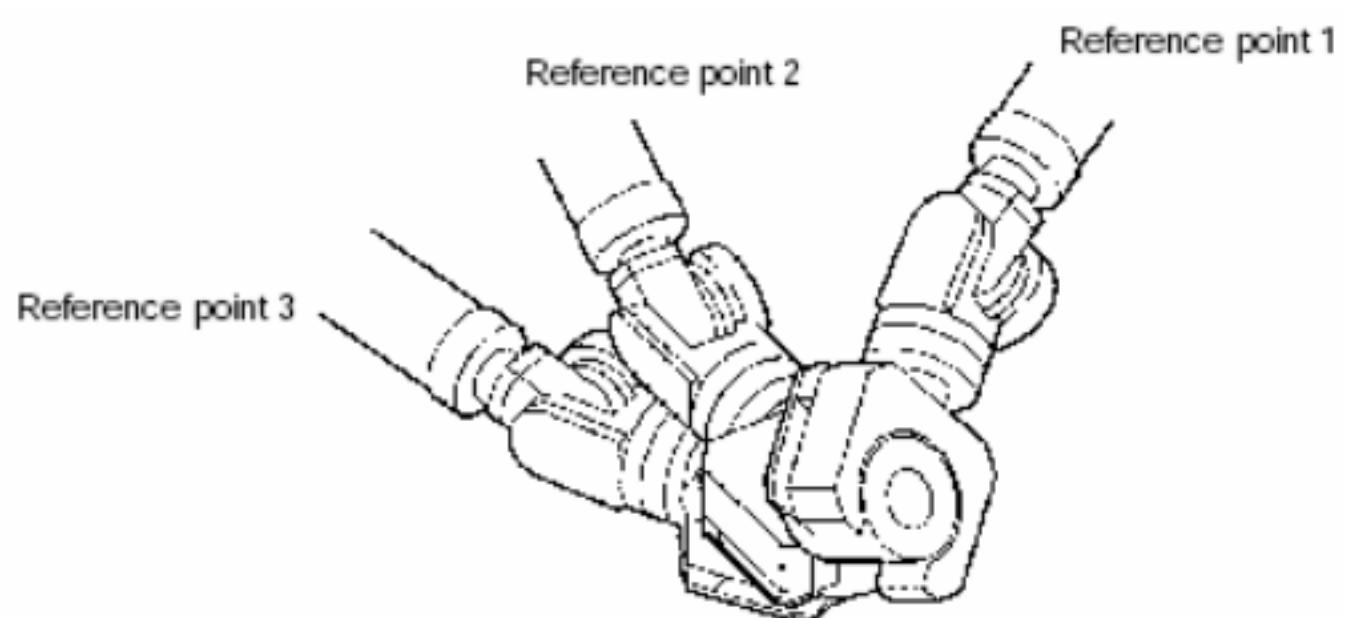


10

```

F $MNUTOOL [ 1, i ] (          i = 1  10)
F $MNUTOOLNUM [ group ]
      3
      (TCP          ) [Optional function          ]
      TCP)
      TCP
      TCP
      x y z          w p r
0 0 0
  
```

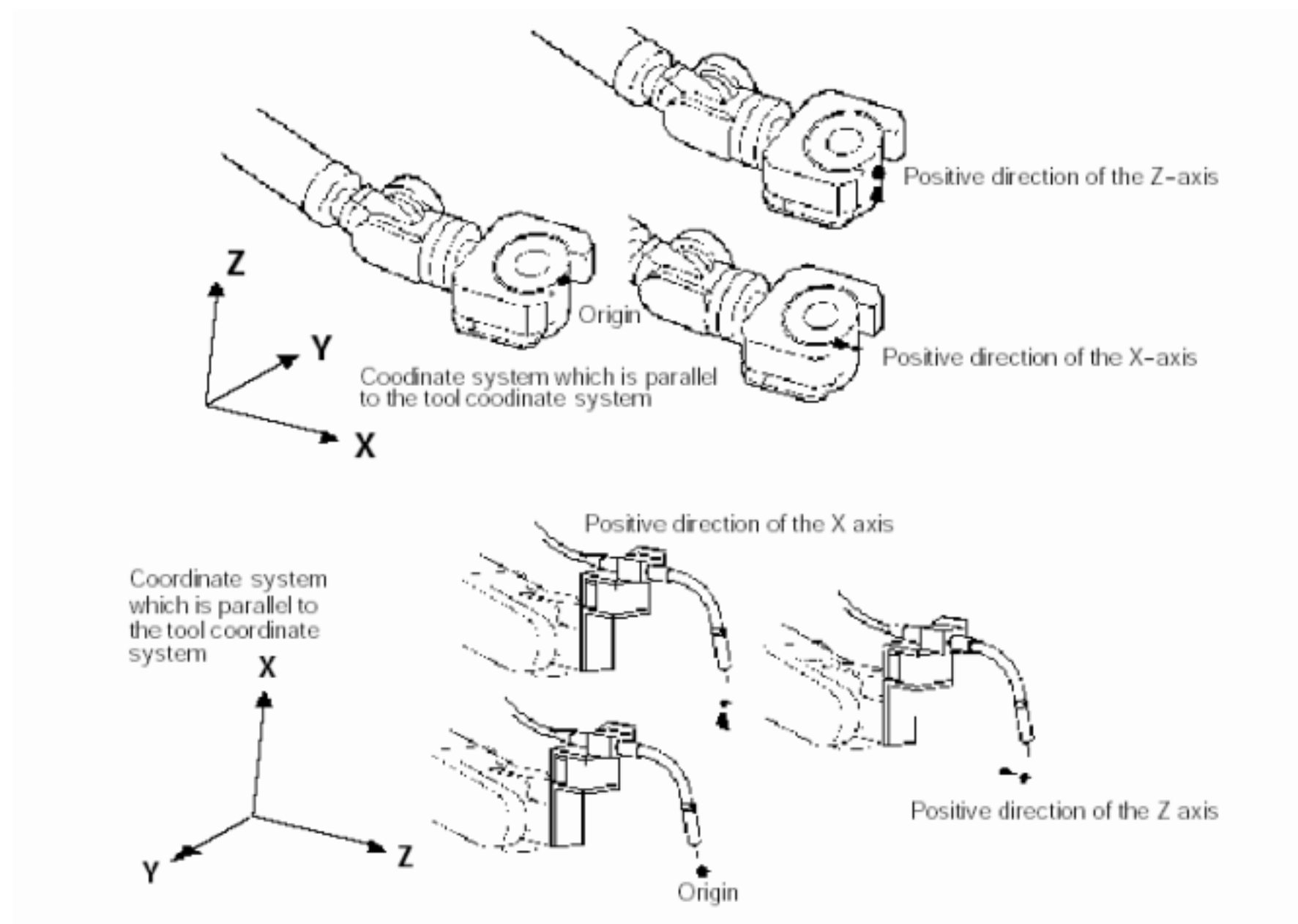
TCP



w p r  
w p r

x

XZ



TCP                      x,y,z  
 x--,y    —z—                      w,p,r    ,  
 TCP                      (                      )  
 1    MENU  
 2    t    “ 6 (SETUP).    ”  
 3    F1  
 4    Frames.  
 5    F3, OTHER                      Tool Frame.

Tool frame list screen

5 I/O  
6 SETUP  
7 FILE

SETUP Frames JOINT 30 %  
Tool Frame Setup/ Direct Entry 1/9

Frames  
[TYPE]

	X	Y	Z	Comment
1:	0.0	0.0	0.0	*****
2:	0.0	0.0	0.0	*****
3:	0.0	0.0	0.0	*****
4:	0.0	0.0	0.0	*****
5:	0.0	0.0	0.0	*****
6:	0.0	0.0	0.0	*****
7:	0.0	0.0	0.0	*****
8:	0.0	0.0	0.0	*****
9:	0.0	0.0	0.0	*****

Active TOOL \$MNUT00LNUM[1]=1  
[ TYPE ] DETAIL [OTHER ] CLEAR SETIND

1 Tool Frame  
2 Jog Frame  
3 User Frame

[ TYPE ] DETAIL OTHER

6  
7 F2,DETAIL.

[ TYPE ] DETAIL [OTHER ]

8 F2,METHOD Three Point.

Tool frame setup screen (Three Point Method)

1 Three Point  
2 Six Point  
3 Direct Entry

SETUP Frames JOINT 30 %  
Tool Frame Setup/ Three Point 1/4

[ TYPE ] METHOD FRAME

Frame Number: 1  
X: 0.0 Y: 0.0 Z: 0.0  
W: 0.0 P: 0.0 R: 0.0  
Comment: TOOL 1  
Approach point 1: UNINIT  
Approach point 2: UNINIT  
Approach point 3: UNINIT  
Active TOOL \$MNUT00LNUM[1]=1  
[ TYPE ][METHOD] FRAME

F2

9

a

ENTER

```

          _____
          1 [ _____ ] ENTER

```

b

c

d

ENTER

10

a

b

c

SHIFT F5,RECORD

RECORDED

```

Approach point 3:
          SETUP Frames          JOINT 30 %
FRAME  MOVE_TO RECORD          Approach point 1:  RECORDED
          Approach point 2:  RECORDED
          Approach point 3:  UNINIT
SHIFT          [ TYPE ][METHOD] FRAME  MOVE_TO RECORD

```

d

USED

```

SETUP Frames          JOINT 30 %
Tool Frame Setup/ Three Point 4/4
Frame Number: 1
  X: 100.0  Y: 0.0  Z: 120.0
  W: 0.0  P: 0.0  R: 0.0
Comment:          TOOL 1
  Approach point 1:  USED
  Approach point 2:  USED
  Approach point 3:  USED
[ TYPE ][METHOD] FRAME  MOVE_TO RECORD

```

11

SHIFT F4, MOVE\_TO.

```

SETUP Frames          JOINT 30 %
Tool Frame Setup/ Three Point 4/4
Frame Number: 1
  X: 100.0  Y: 0.0  Z: 120.0
  W: 0.0  P: 0.0  R: 0.0
Comment:          TOOL 1
  Approach point 1:  USED
  Approach point 2:  USED
  Approach point 3:  USED
[ TYPE ][METHOD] FRAME  MOVE_TO RECORD

```

12

ENTER

PREV

SETUP Frames			JOINT	30 %
Tool Frame Setup/ Direct Entry			1/9	
	X	Y	Z	Comment
1:	100.0	0.0	120.0	T00L1
2:	0.0	0.0	0.0	*****
3:	0.0	0.0	0.0	*****
4:	0.0	0.0	0.0	*****
5:	0.0	0.0	0.0	*****
6:	0.0	0.0	0.0	*****
7:	0.0	0.0	0.0	*****
8:	0.0	0.0	0.0	*****
9:	0.0	0.0	0.0	*****

[ TYPE ] DETAIL [OTHER ] CLEAR SETIND

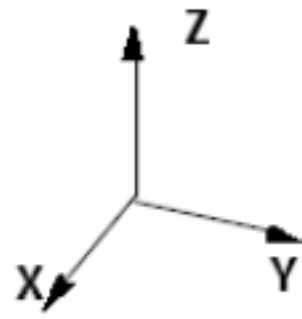


(x, y, z)

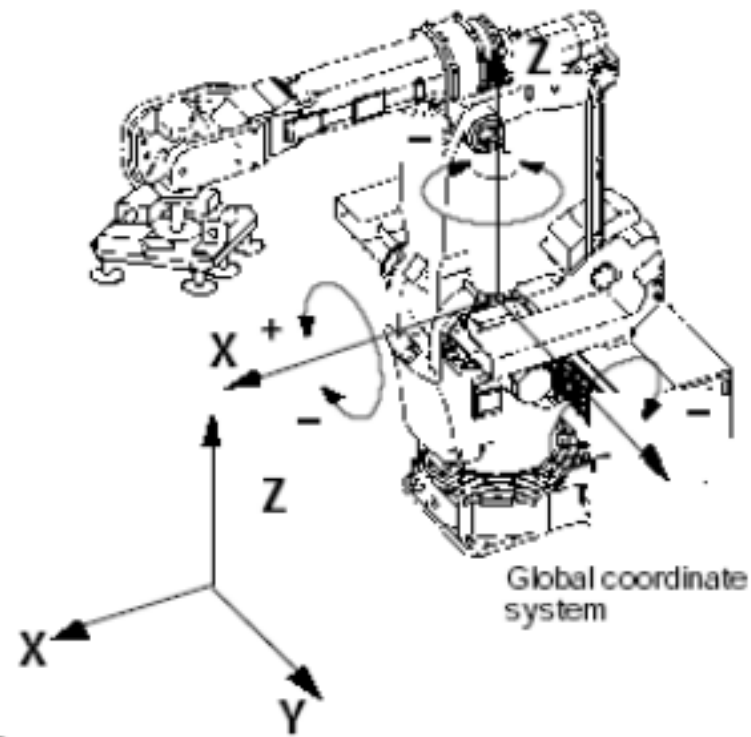
(w, p, r)

X-,Y- Z

User coordinate system 2  
 User coordinate system 1  
 Global coordinate system



User coordinate system 1



Global coordinate system

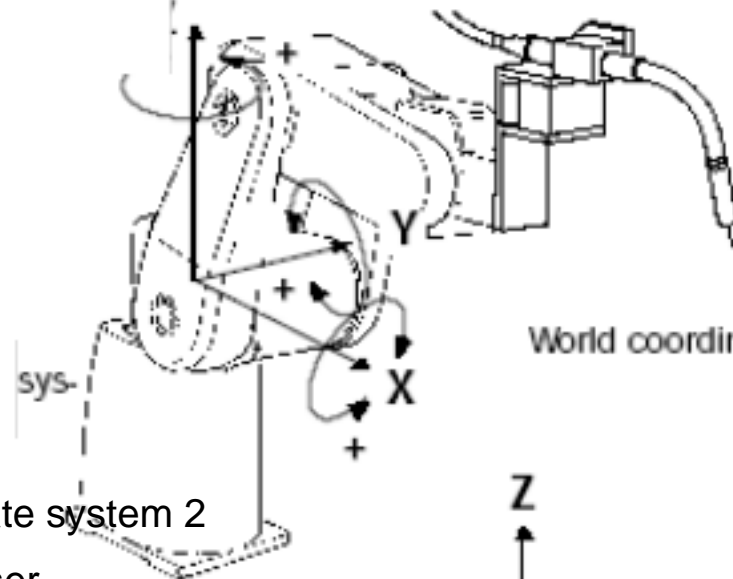
User coordinate system 2

X  
 Y  
 Z  
 X Y  
 Z  
 X  
 Y  
 Z

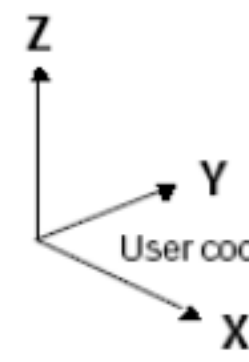
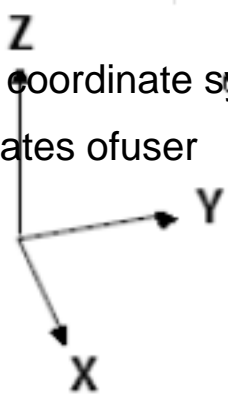
+  
 --  
 +  
 --  
 + --

1	
x :	70.0 mm
y :	-2500.0 mm
z :	200.0 mm
w :	0.0 deg
p :	0.0 deg
r :	-30.0 deg
2	

World coordinate system  
 User coordinate system 1  
 User coordinate system 2



World coordinate system

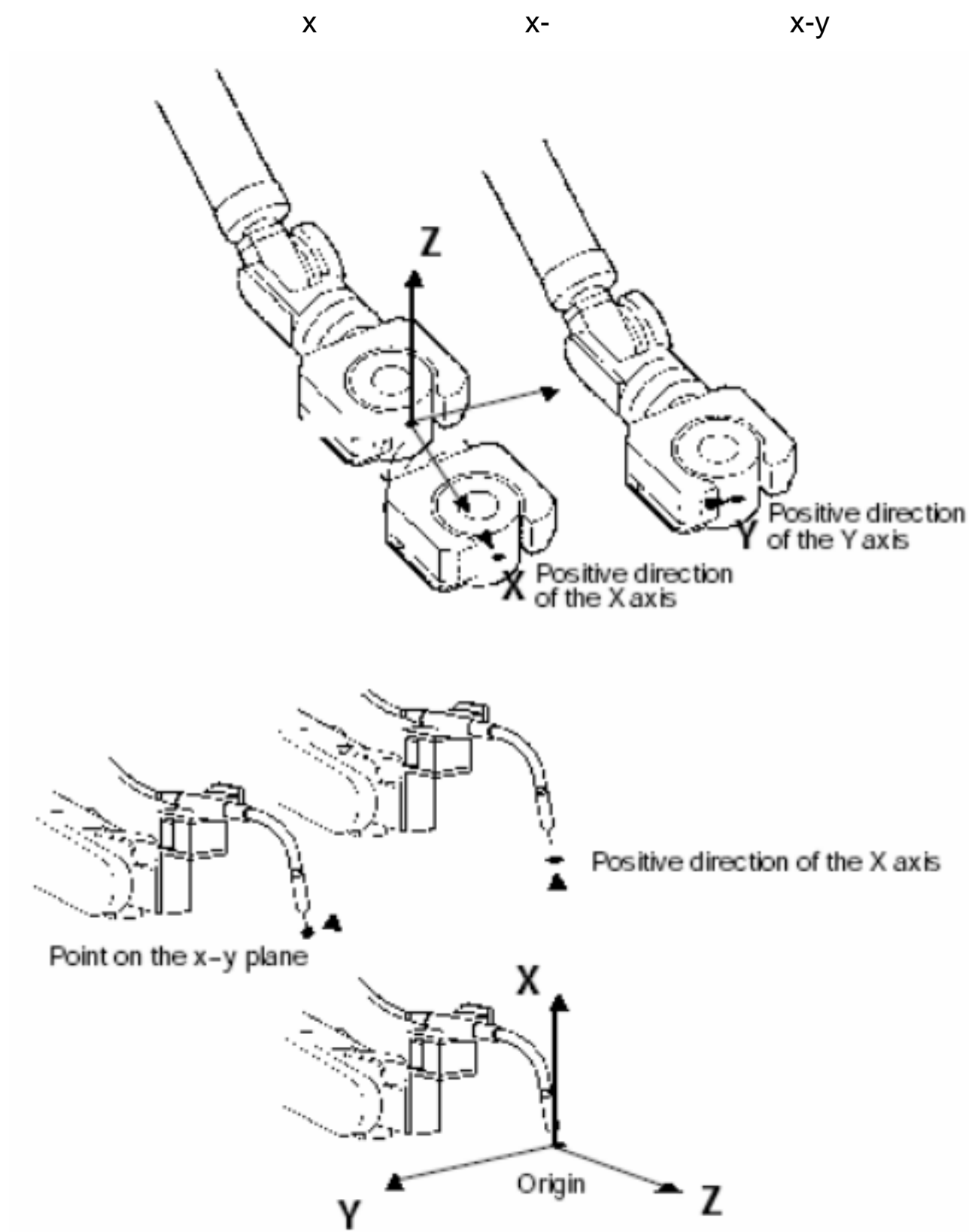


User coordinate system 2

In the right figure, the coordinates of user coordinate system 1 are

x : 70.0 mm  
 y : -2500.0 mm  
 z

```
F $MNUFRAME [ 1, i ] ( i = 1 9 ) .
F $MNUFRAMENUM [ 1 ]
```



- 1 MENU .
- 2 " 6 (SETUP). "
- 3 F1 TYPE.
- 4 Frames.
- 5 F3, OTHER User Frame.

User frame list screen

5 I/O  
 6 SETUP  
 7 FILE

SETUP Frames JOINT 30 %

User Frame Setup/ Direct Entry 1/5

Frames  
 [TYPE]

	X	Y	Z	Comment
1:	0.0	0.0	0.0	*****
2:	0.0	0.0	0.0	*****
3:	0.0	0.0	0.0	*****
4:	0.0	0.0	0.0	*****
5:	0.0	0.0	0.0	*****

Active UFRAME SMNUFRAMNUM[1]=0  
 [ TYPE ] DETAIL [OTHER ] CLEAR SETIND

- 1 Tool Frame
- 2 Jog Frame
- 3 User Frame

[ TYPE ] DETAIL OTHER

- 6
- 7 F2,DETAIL.



- 7 F2,METHOD Three Point.

User frame setup screen (Three Point Method)

- 1 Three Point
- 2 Four Point
- 3 Direct Entry

[ TYPE ] METHOD FRAME

F2

SETUP Frames JOINT 30 %

User Frame Setup/ Three Point 1/4

Frame Number: 1

X:	0.0	Y:	0.0	Z:	0.0
W:	0.0	P:	0.0	R:	0.0

Comment:\*\*\*\*\*

Orient Origin Point: UNINIT  
 X Direction Point: UNINIT  
 Y Direction Point: UNINIT

Active UFRAME SMNUFRAMNUM[1]=0  
 [ TYPE ][METHOD] FRAME

9

a

ENTER

**JOINT 30 %**

[ ]

ENTER

b

c

d

ENTER

**SETUP Frames JOINT 30 %**

**Comment: REFERENCE FRAME**

**[ TYPE ][METHOD] FRAME**

10

a

b

c

SHIFT

F5,RECORD

RECORDED

FRAME MOVE\_TO RECORD

**SETUP Frames JOINT 30 %**

**Orient Origin Point: RECORDED**

**X Direction Point: RECORDED**

**Y Direction Point: UNINIT**

**Active UFRAME SMNUFRAMNUM[1]=0**

**[ TYPE ][METHOD] FRAME MOVE\_TO RECORD**

**F5**

d

USED

**SETUP Frames JOINT 30 %**

**User Frame Setup/ Three Point 4/4**

**Frame Number: 1**

**X: 1243.6 Y: 0.0 Z: 10.0**

**W: 0.1 P: 2.3 R: 3.2**

**Comment: REFERENCE FRAME**

**Orient Origin Point: USED**

**X Direction Point: USED**

**Y Direction Point: USED**

**Active UFRAME SMNUFRAMNUM[1]=0**

**[ TYPE ][METHOD] FRAME MOVE\_TO RECORD**

11

SHIFT

F4,MOVE\_TO

FRAME MOVE\_TO RECORD

SHIFT

12

ENTER

PREV

13

PREV

PREV

SETUP Frames				JOINT 30 %
User Frame	Setup/	Three	Point	1/9
X	Y	Z	Comment	
1:	1243.6	0.0	43.8	REFERENCE FR>
2:	0.0	0.0	0.0	*****
3:	0.0	0.0	0.0	*****
4:	0.0	0.0	0.0	*****
5:	0.0	0.0	0.0	*****
6:	0.0	0.0	0.0	
7:	0.0	0.0	0.0	
8:	0.0	0.0	0.0	
9:	0.0	0.0	0.0	

Active UFRAME SMNUFRAMNUM[1]=0  
 [ TYPE ] DETAIL [OTHER ] CLEAR SETIND

14

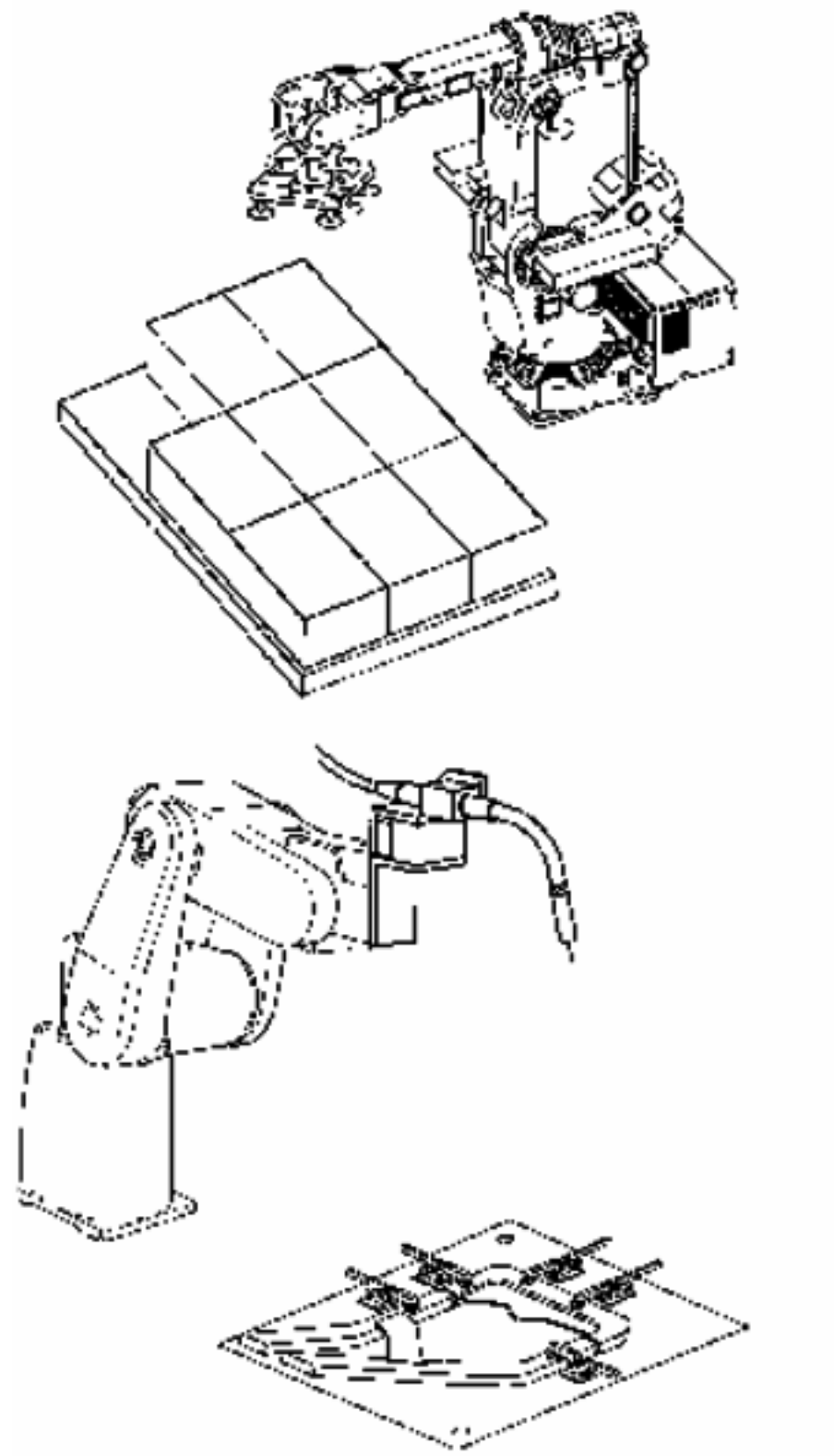
(SETIND),



15

F4 CLEAR.

[OTHER ] CLEAR SETIND



DO

SDO

1

I/O

(ATPERCH)

[6 (SETUP). Ref Position]


safe :	
dangerous	



SETUP Space



Space/DETAILED SCREEN( )  
 (Area Details Screen )

Enable/disable	
Comment	10
Output signal	
Input signal	
Drriority	<p>High Low High Low</p> <p>1</p>  <p>2 3 4</p>
inside/outside	

Rectangular Space/SPACE SETUP  
 (Area Setting Screen )

BASIS VERTEX	
SIDE LENGTH/SECOND VERTEX	<p>SIDE LENGTH              X, Y, Z</p> <p>SECOND VERTEX</p>

- Step 1 MENUS
- 2 SETUP.
- 3 F1 " TYPE. "
- 4 Space fnct.

**Area List Screen**

5 I/O  
 6 SETUP  
 7 FILE

LIST SCREEN				%
No.	End/Dsbl	Comment	Usage	
1	ENABLE	[	] Common Space	
2	DISABLE	[	] Common Space	
3	DISABLE	[	] Common Space	

Space fnct.

TYPE

[TYPE]                   DETAIL   ENABLE   DISABLE

- 5
- a. Enter
- b.
- c.
- d. Enter

**Rectangular Space**

LIST SCREEN	
No.	Enb/Dsbl
1	ENABLE [
2	DISABLE [
3	DISABLE [

ENTER

6                   Enb/Dsbl                   Comment                   F3 (DETAIL).

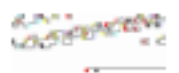
**Rectangular Space**

DETAILED SCREEN			
SPACE: 1	GROUP: 1		
USAGE:	Common Space		
1 Enable/Disable:	ENABLE		
2 Comment:	[*****]		
3 Output Signal:	D0[0]		
4 Input Signal:	DI[0]		
5 Priority:	High		
6 Inside/Outside:	Inside		
[TYPE]	SPACE	ENABLE	DISABLE

DETAIL ENABLE DISABLE

7

		SPACE SETUP				%	
		SPACE: 1		GROUP: 1		1/4	
		UFRAME: 0		UTOOL: 1			
[TYPE]	SPACE	1: BASIS VERTEX		[SIDE LENGTH]			
		2: X	0.0 mm		0.0 mm		
		3: Y	0.0 mm		0.0 mm		
		4: Z	0.0 mm		0.0 mm		
		[TYPE]	OTHER			RE10RD	
	9					T	
a.		X,Y	Z				
b.				SHIFT	+F5 RECORD		
				RE10RD			
				F5	#74		
		UF	UT,	b		UF	UT
				SHIFT	+F5 RECORD		
	10		PREV				PREV



[6 SETUP General]

Break on hold

Current language

Ignore Offset command

Ignore Tool--offset

<p>Break on hold</p>	<p>HOLD</p> <p>DISABLED HOLD</p> <p>( )</p> <p>ENABLE HOLD</p> <div style="border: 1px solid black; height: 100px; width: 100%; margin-top: 10px;"> <input type="checkbox"/> </div>
<p>Current language</p>	<p>DEFAULT</p>
<p>Ignore Offset command</p>	<p>DISABLED</p> <p>ENABLED</p>
<p>Ignore Tool--offset</p>	<p>DISABLED</p> <p>ENABLED</p>

```

Program detail                               JOINT 30 %
                                           1/6
Creation Date:                          10-MAR-1998
Modification Date:                      11-MAR-1998
Copy Source:                            [*****]
Positions: FALSE  Size:                312 Byte
1 Program name:                         [SAMPLE3 ]
2 Sub Type:                             [      None]
3 Comment:                              [SAMPLE PROGRAM 3]
4 Group Mask:                           [1, *, *, *, *]
5 Write protect:                        [      OFF]
6 Ignore pause:                         [      OFF]
END PREV NEXT

```

```

Memory available capacity ←
Select                               JOINT 30 %
                                           58740 bytes free  1/7
No. Program name Comment
1 SAMPLE1 [SAMPLE PROGRAM 1]
2 SAMPLE2 [SAMPLE PROGRAM 2]
3 SAMPLE3 [SAMPLE PROGRAM 3]
4 PROG001 [PROGRAM001 ]
5 PROG002 [PROGRAM001 ]
6 CLAMP1 [CLAMP OPEN ] → Attribute
7 CLAMP2 [CLAMP CLOSE ]
Program name ←
[ TYPE ] CREATE DELETE MONITOR [ATTR ]>
COPY  DETAIL  LOAD  SAVE  PRINT  >

```

```

Program name ← SAMPLE1                               JOINT 10%
                                           1/8
Line number
Program statement
  - Motion instruction  -1: J P[1] 100% FINE
                       2:  HAND1CLOSE
  - Macro instruction   -3: J P[2] 70% CNT50
                       4: L P[3] 500mm/sec CNT10
                       -5:  HAND1OPEN
                       6: L P[4] 500mm/sec CNT10
                       7:  HAND1CLOSE
  - Abort instruction  -8:  ABORT
Program end symbol    -[End]

POINT                               TOUCHUP >

```

Program name	➤ <b>SAMPLE1</b>	<b>JOINT 30 %</b>
Line number		<b>1/8</b>
Motion instruction	➤ <b>1:J P[1] 100% FINE</b>	
Program statement	➤ <b>2:J P[2] 70% CNT50</b>	
➤ Arc welding instruction	<b>3:L P[3] 500mm/sec FINE</b>	
	➤ <b>: Arc Start[1]</b>	
	<b>4:L P[4] 50mm/sec CNT80</b>	
	<b>5:L P[5] 50mm/sec CNT80</b>	
	<b>: Arc End[50.0V,75.0A,0.1s]</b>	
	<b>6:J P[1] 100% FINE</b>	
➤ Abort instruction	➤ <b>7: ABORT</b>	
Program end symbol	<b>⚡End]</b>	
	<b>[ INST ]</b>	<b>[EDCMD]&gt;</b>
	<b>POINT ARCSTRT WELD_PT</b>	<b>ARCEND TOUCHUP&gt;</b>

1 8

at @ 0 9 \* -

A HAND\_A

RSR PNS

RSR RSRnnnn nnnn 4  
RSR0001.

PNS PNSnnnn nnnn 4  
PNS0001.

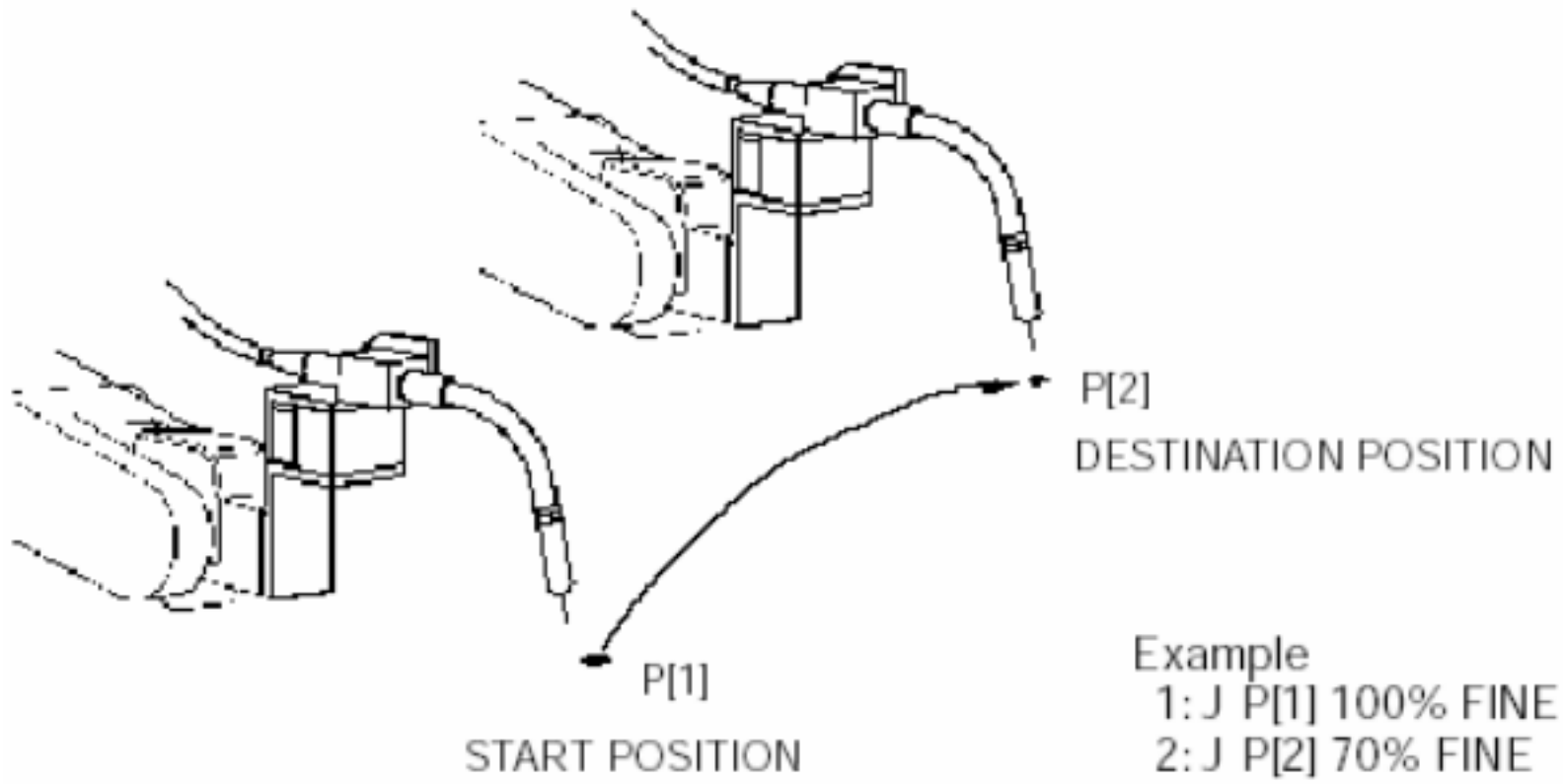
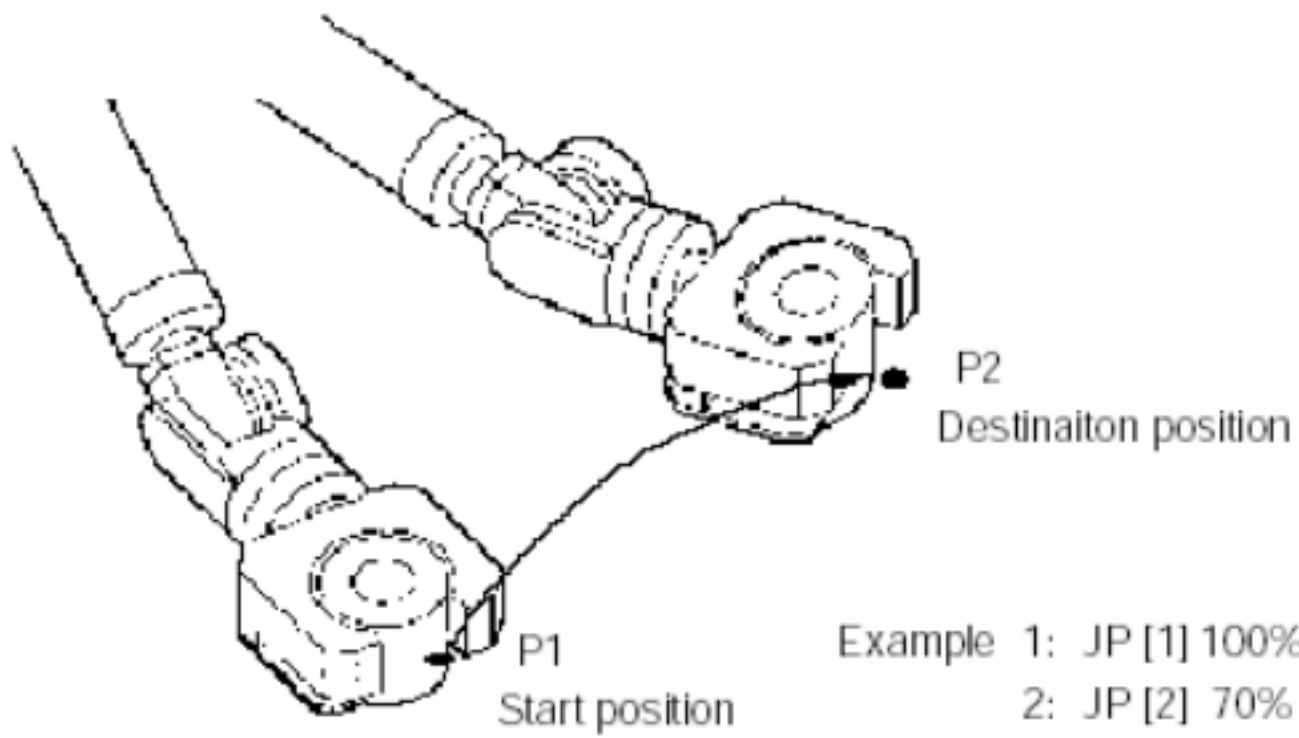
ON  
 ON  
 ON  
 OFF  
 OFF  
 robot tool  
 4-5

			Position data
			UF:0 UT:1
Position data format		➔	X: 1500.374 W: 10.000
P 1 to 1500 *			Y: -342.992 P: 20.000
PR 1 to 10			Z: 956.895 R: 40.000
			CONF: N, R, D, F, 0, 0, 0
J	P[i]	J%	CNTk
Motion format		Feedrate	Positioning path
J		1 to 100%	FINE
L		1 to 2000mm/sec	CNT 0 to 100
C		1 to 12000cm/min	
		0.1 to 4724.0inch/min	
		1 to 272deg/sec	
		1 to 3200deg/sec	
		1 to 32000msec	

\* A position number can be as large as the memory capacity allows.



J  
C  
L



start position:  
L

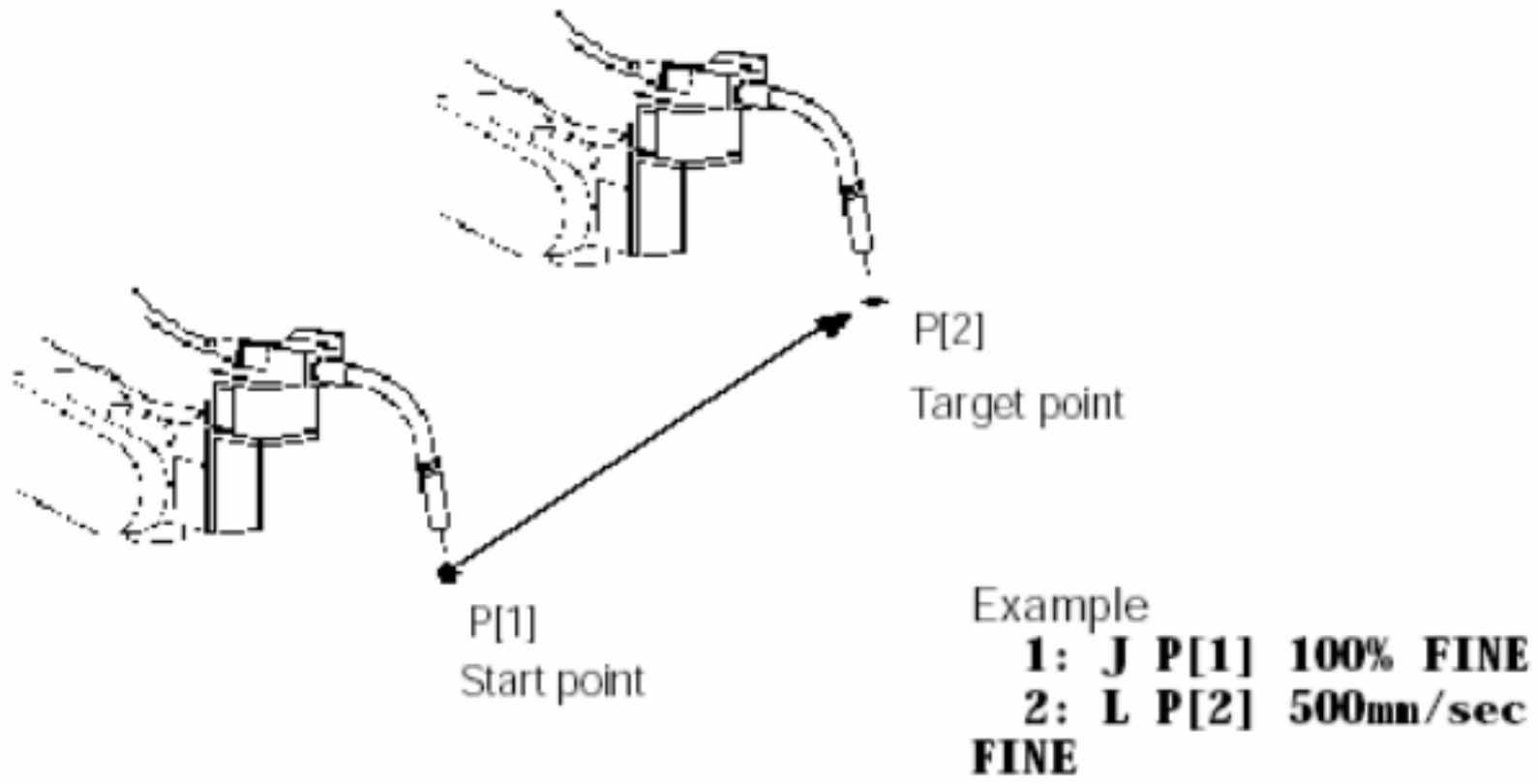
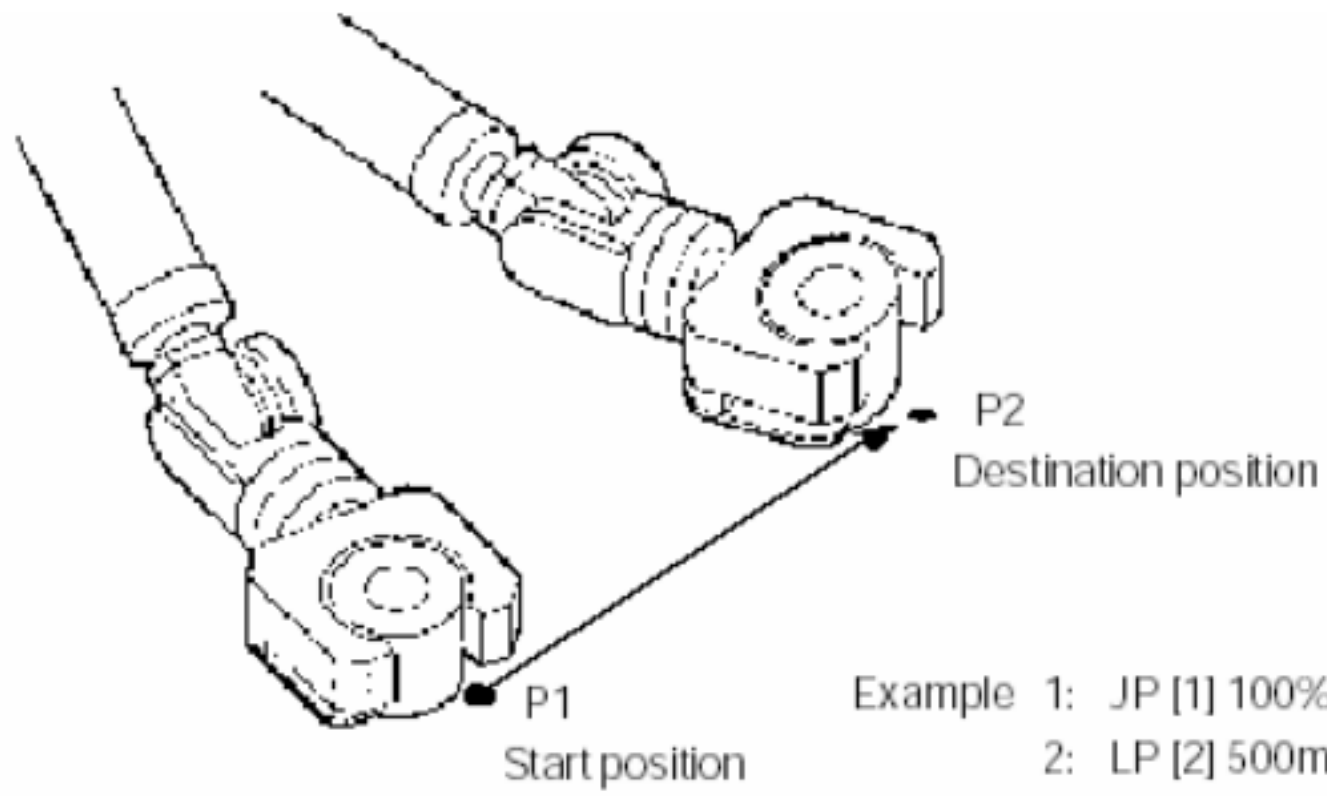
destination position:

example:

(TCP)

mm/sec / cm/min /

inch/mm( / )

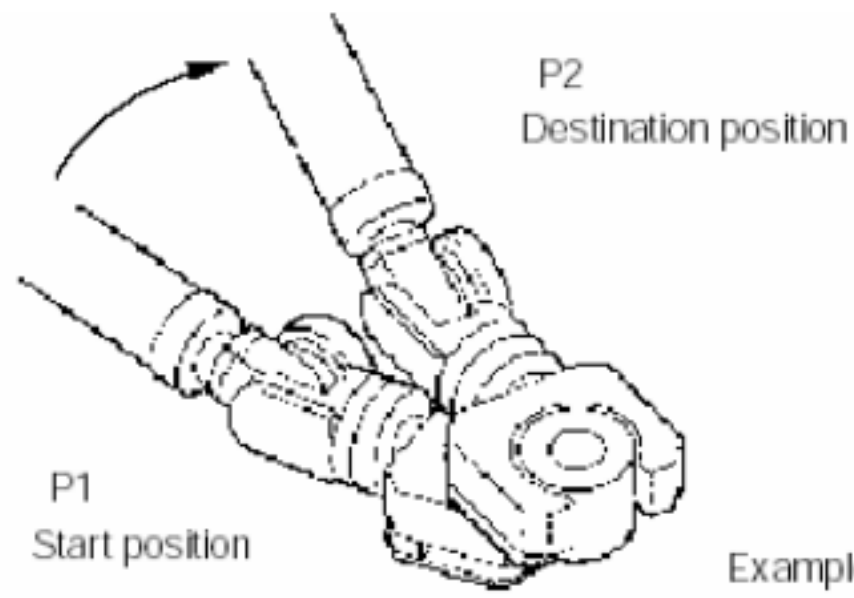


start point  
start position

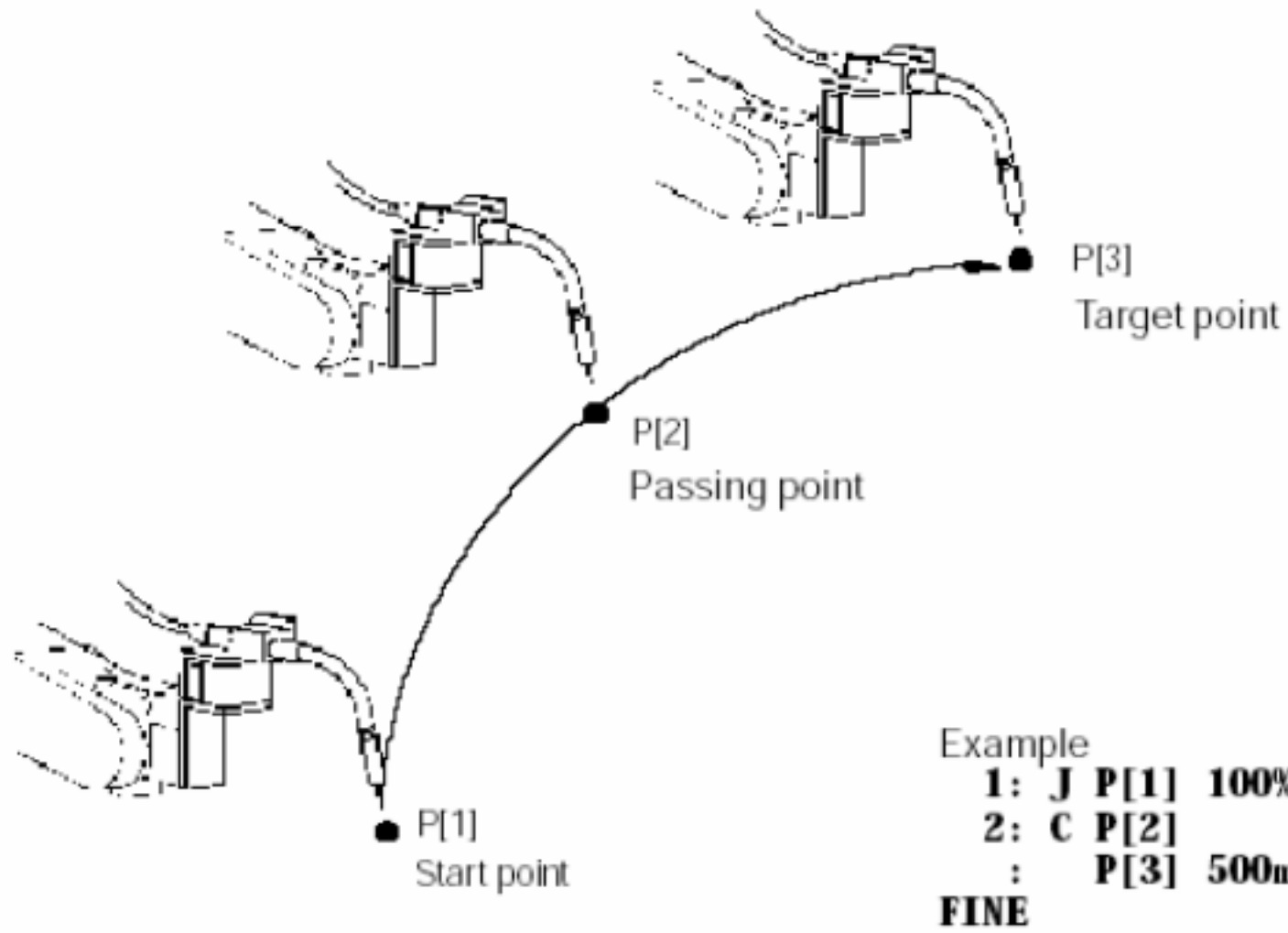
target point  
destination position

example

deg/sec /



Example 1: JP [1] 100% FINE  
2: LP [2] 30deg/sec FINE



Example  
1: J P[1] 100% FINE  
2: C P[2]  
: P[3] 500mm/sec  
FINE

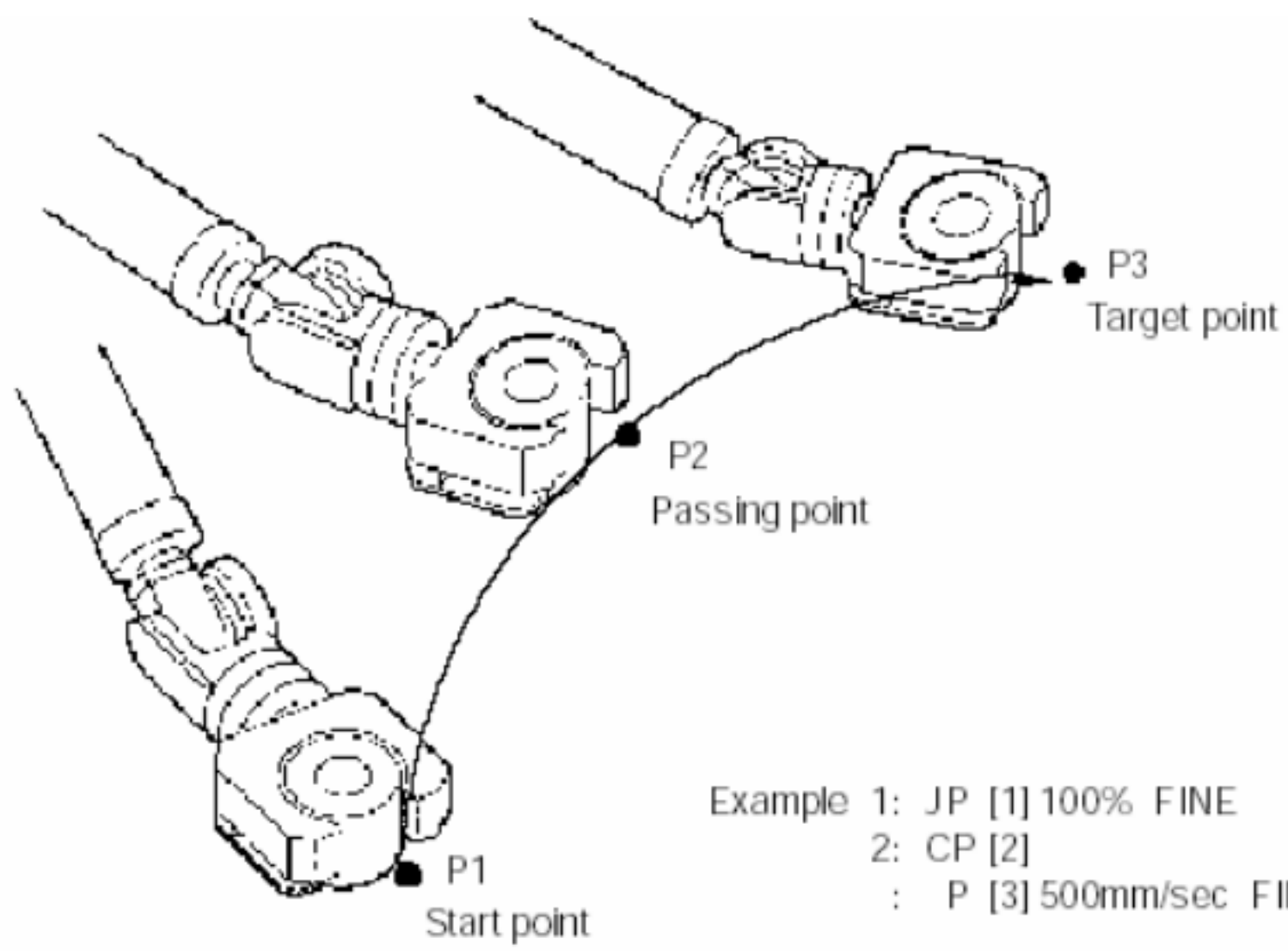
start point  
start position

passing point  
destination position

target point  
example

inch/mm( / )

mm/sec / cm/min /



start point  
passing point  
target point  
example

UF , UT , (X , Y , Z , W , P , R) , Configuration  
 User coordinate system number      Tool coordinate system number      Position      Attitude      Configuration

user coordinate system number  
 tool coordinate system number  
 position  
 attitude  
 configuration

x,y,z

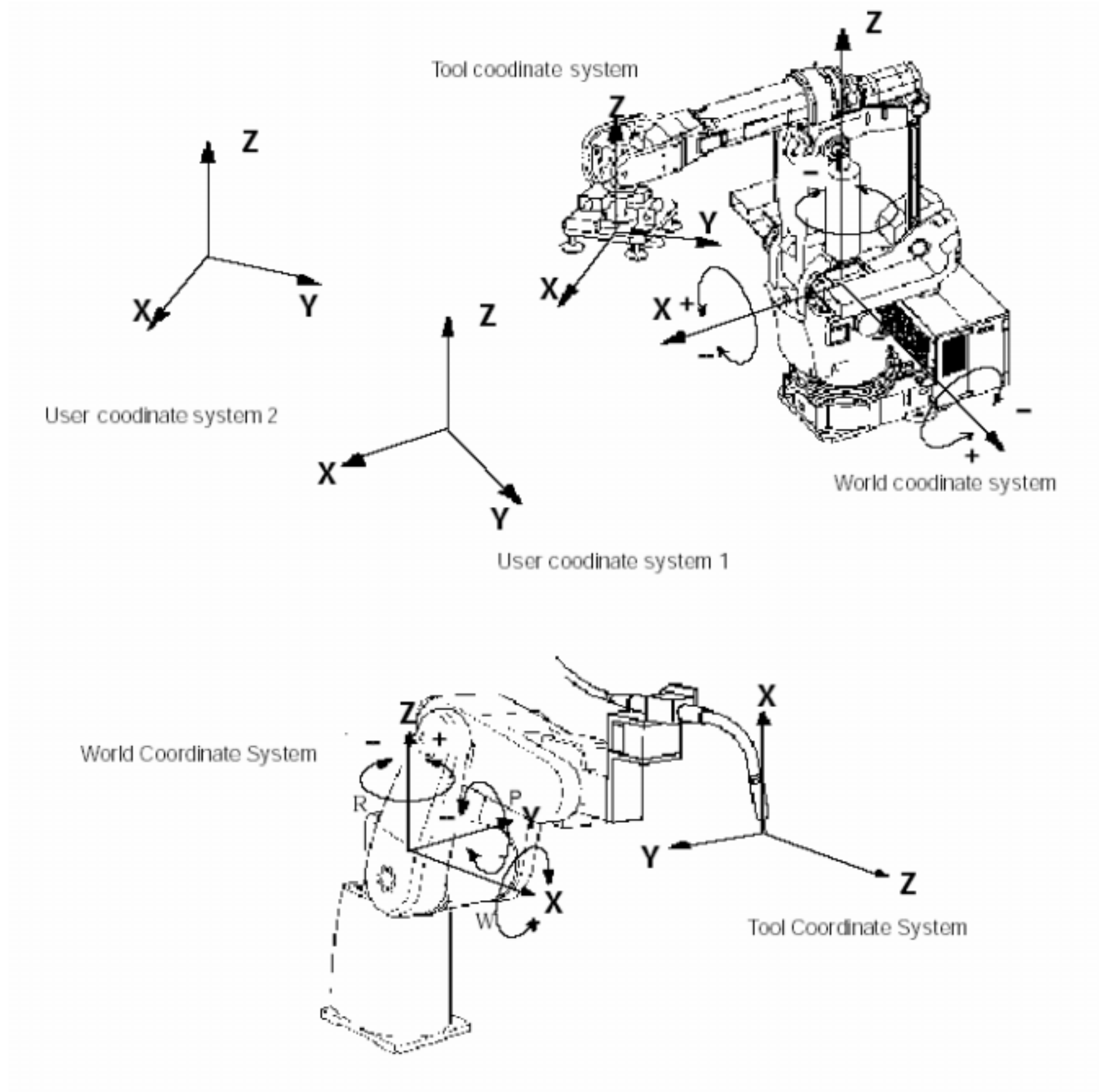
w,p,r

X

Y

Z

/



x,y,z,w,p,r

1% 100%

J P[1] 50% FINE

1% 100%

0.1 3200

1 32000

L P[1] 100mm/sec FINE

/ 1 2000 /  
/ 1 12000 /  
/ 0..1 4724.4 /

0.1 3200

1 32000

L P[1] 50 deg/sec FINE

/ 1 272 /

0.1 3200

1 32000

GI



J P[1] R[i]% FINE  
L P[1] R[i] / FINE  
C P[1]  
P[2] R[i] / FINE

J PAL\_1[A\_1] R[i]% FINE  
J PAL\_1[BTM] R[i]% FINE  
J PAL\_1[R\_1] R[i]% FINE



GP1 JP[1] R[i]% FINE  
GP2 JP[1] R[i]% FINE

/

/

	1	100			
	0.1	3200.0	*1	/	
	1	32000	*1		
/	1	2000	*1		
/	1	12000	*1		
/	0.1	4724.2	*2	/	
/	1	272	*3		

FALSE

---

10: R [1] = 100  
11: J P[5] R[1]% FINE  
12: R[1]=10  
14: J P[6] R[1]% FINE

10      100      12      10

14

---

FINE  
CNT

4.3.6.

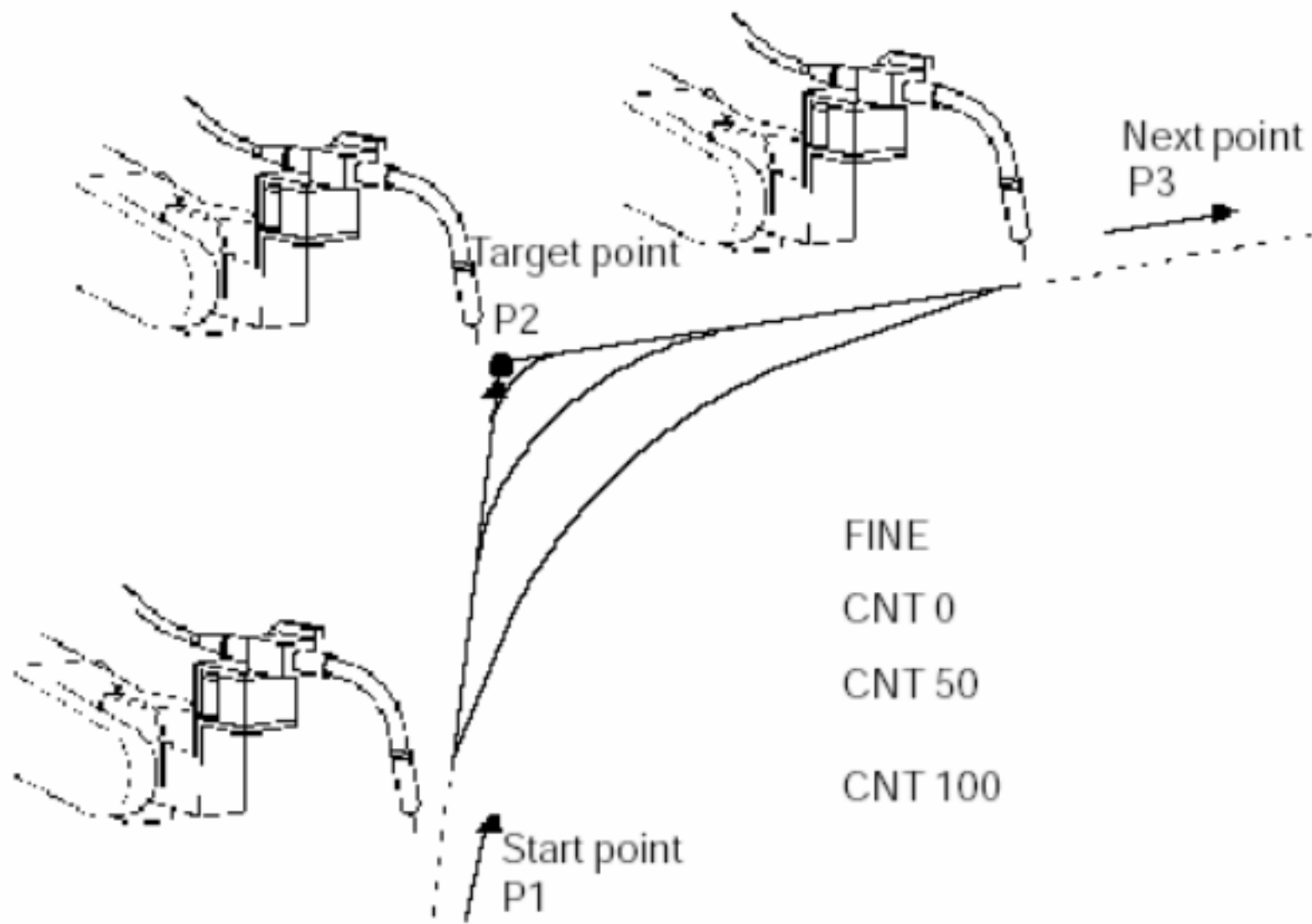
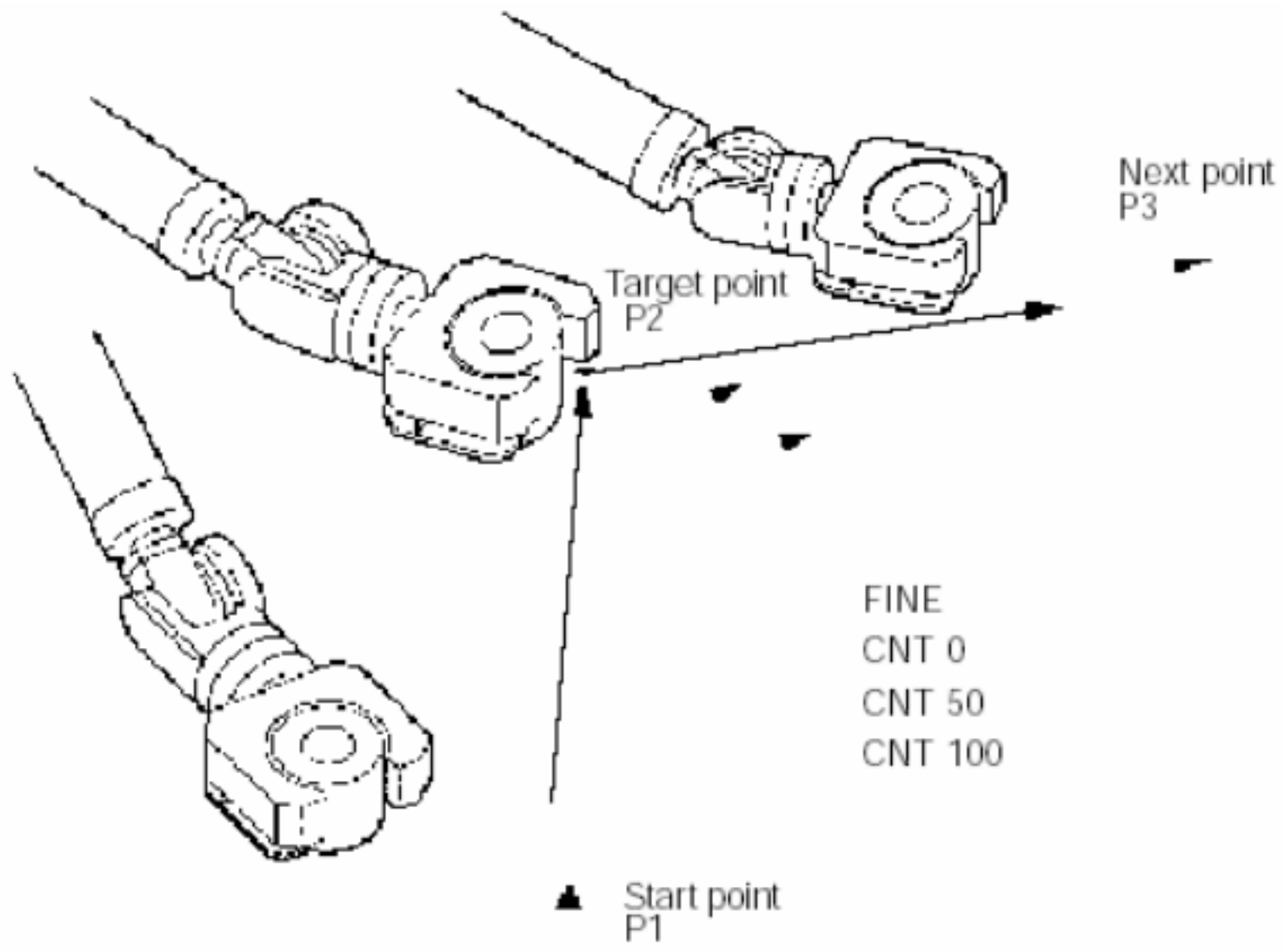
FINE J P[i] 50% FINE  
FINE  
CNT J P[i] 50% CNT 50  
CNT

100

CNT

CNT 100

continuous termination



Wjnt  
 Skip LBL[i]  
 Offset, PR[i]  
 Tool\_Offset, PR[i]  
 INC  
 EV EVi%  
 EV ind.EVi%  
 PTH  
 /

F4[CHOICE]

<b>JOINT 30%</b> 4/5 <b>500mm/sec CNT10</b> CHOICE	<b>Motion Modify</b> <b>1 No option</b> <b>2 Wrist Joint</b> <b>3 Offset/Frames</b> <b>4 Offset.PR[ ]</b> <b>PROGRAM1</b>	<b>JOINT 30%</b> <b>5 Incremental</b> <b>6 Skip,LBL[ ]</b> <b>7</b> <b>8</b>
---	--	--

LP[i] 50% FINE Wjnt

<b>Motion Modify</b>	
<b>1 No option</b>	<b>5 In</b>
<b>2 Wrist Joint</b>	<b>6 Sk</b>
<b>3 Offset</b>	<b>7</b>
<b>4 Offset,PR[ ]</b>	<b>8</b>

**Acceleration override**  
**J P[1] 50% FINE ACC80**

- Motion modify**
- 1 No option**
- 2 Wrist Joint**
- 3 ACC**
- 4 Skip, LBL[]**
- PROGRAM1**

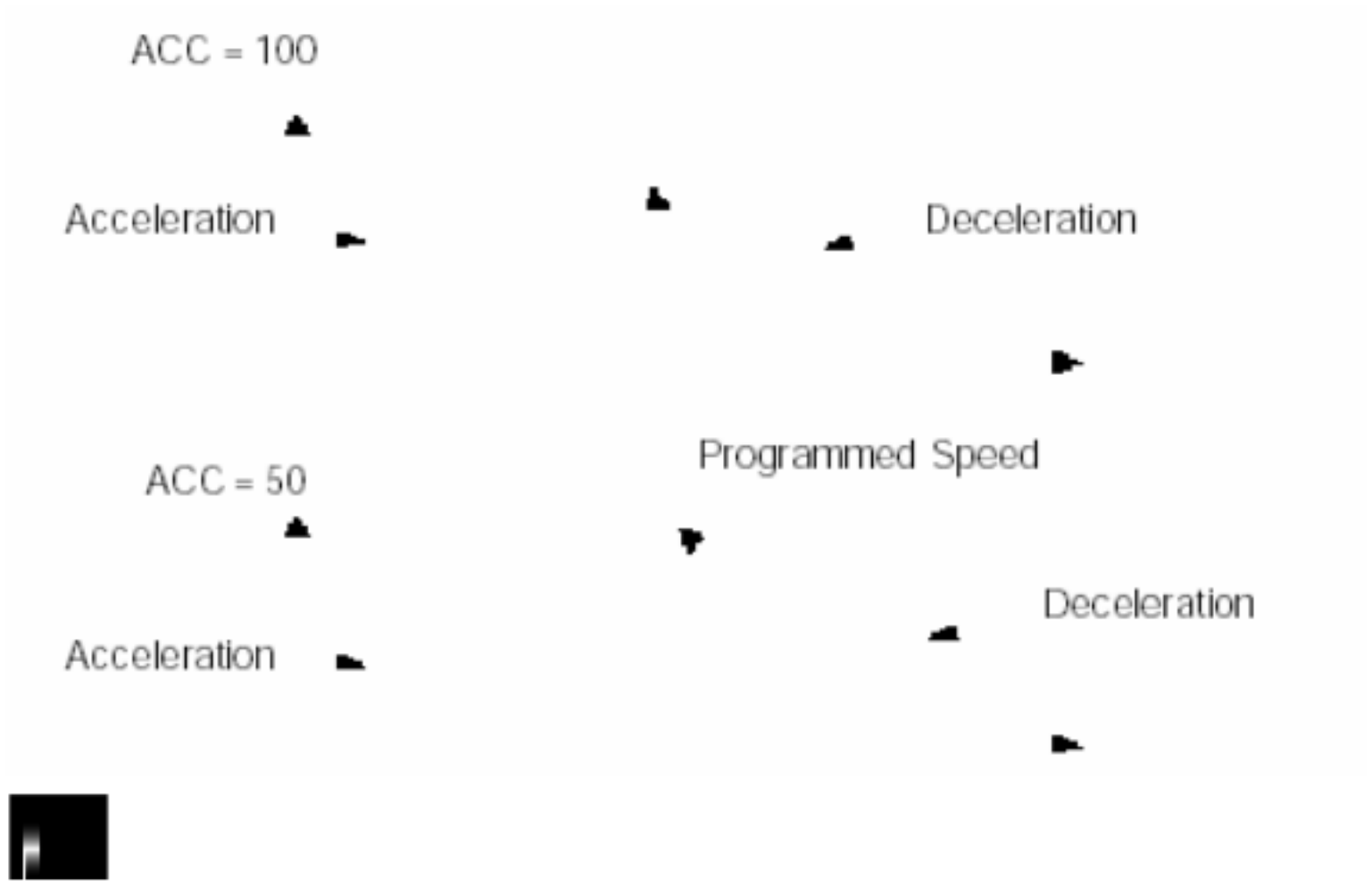
acceleration override

/

ACC 100%

ACC 100%

0 150



---

**Skip instruction**

**SKIP CONDITION [I/O] = [value] J P[1] 50 FINE Skip,LBL[3]**

```
JOINT 30%  
5 Incremental  
6 Skip,LBL[ ]  
7  
8
```

skip instruction

branch destination label

---

**Incremental instruction**  
**J P[1] 50% FINE INC**

**JOINT 30 %**  
**5 Offset**  
**6 Offset, PR[ ]**  
**7 Incremental**  
**8 ---next page---**

P[i]

4.3.2

INC

OFFSET



---

CNT

100

CNT

CNT100  
CNT

L P[1] 100 / CDy

F2 F4

**POINT ARCSTRT WELD\_PT ARCEND TOUCHUP>**

Instruction	JOINT 30 %
1 Registers	5 JMP/LBL
2 I/O	6 CALL
3 IF/SELECT	7 Arc
4 WAIT	8 ---next page---
SAMPLE1	

[i]  
[V,A]  
[i]  
[i]

[i]  
[V,A,s]  
[i]  
[i]

TIME 0

WDI

WO

R[i]=WDI[i]

R[i]=WDI[i]

On 1 Off 0

## PALLETIZING--[pattern]\_i

B, BX, E, EX

Palletizing number(1 to 16)

palletizing number 1 to 16

1 16

4

B	
BX	
E	
EX	



---

<b>Instruction</b>	<b>JOINT 30%</b>
<b>1 Registers</b>	<b>5 JMP/LBL</b>
<b>2 I/O</b>	<b>6 CALL</b>
<b>3 IF/SELECT</b>	<b>7 Palletizing</b>
<b>4 WAIT</b>	<b>8 ---next page---</b>
<b>PROGRAM</b>	

<b>Instruction</b>	<b>JOINT 30%</b>
<b>1 Registers</b>	<b>5 JMP/LBL</b>
<b>2 I/O</b>	<b>6 CALL</b>
<b>3 IF/SELECT</b>	<b>7 Arc</b>
<b>4 WAIT</b>	<b>8 ---next page---</b>
<b>PROGRAM</b>	

**Example** 1:  $R[2]=R[3]-R[4]+R[5]-R[6]$   
 2:  $R[10]=R[2]*[100/R[6]]$

5

**Example** 1:  $R[2]=R[3]+R[4]+R[5]+R[6]+R[7]+R[8]$

5

“ + ” “ ”

“ \* ” “ / ”

“ + ” “ ”

“ \* ” “ / ”

4.6.1

$R[i]=(value)$

$R[i]=(value)$

---

Instruction		JOINT 30%
1 Registers	5 JMP/LBL	
2 I/O	6 CALL	
3 IF/SELECT	7 Palletizing	
4 WAIT	8 ---next page---	
PROGRAM		

Instruction		JOINT 30%
1 Registers	5 JMP/LBL	
2 I/O	6 CALL	
3 IF/SELECT	7 Arc	
4 WAIT	8 ---next page---	
PROGRAM		

/

---

## SDO [ i ] = (value)

Digital output signal number

ON : Turns on the digital output signal.

OFF: Turns off the digital output signal.

Example    **3: SDO[1] = ON**  
              **4: SDO[ R[3] ] = OFF**

---

## SDO [ i ] = PULSE, (value)

Digital output signal number

Pulse width (sec) (0.1 to 25.5 sec)

Example    **5: SDO[1] = PULSE**  
            **6: SDO[2] = PULSE, 0.2sec**  
            **7: SDO[ R[3] ] = PULSE, 1.2sec**

SDO[i] = R[i]

SDO[i] = R[i]

0

0

SDO[i] = R[i]

## SDO [ i ] = R [ i ]

Digital output signal number

Register number (1 to 32)

Example    **7: SDO[1] = R[2]**  
            **8: SDO[ R[5] ] = R [ R[1] ]**

## JMP LBL [ i ]

Label (1 to 32767)

Example    **3: JMP LBL[2:hand open]**  
            **4: JMP LBL[ R[4] ]**

CALL

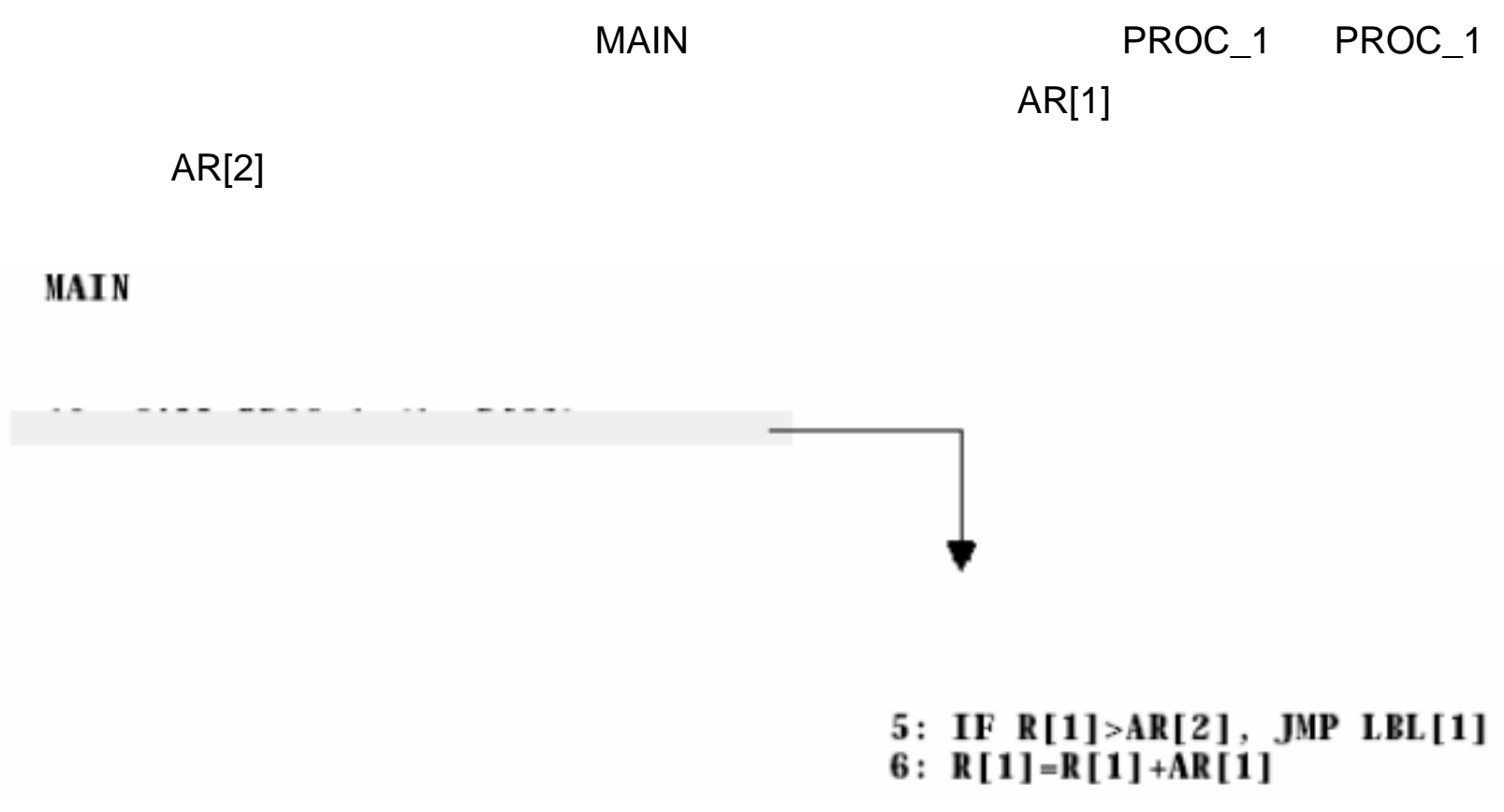
CALL

END

F5,STRINGS

**Instruction** **JOINT 30%**  
**1 Registers**            **5 JMP/LBL**  
**2 I/O**                   **6 CALL**  
**3 IF/SELECT**           **7 Palletizing**  
**4 WAIT**                 **8 ---next page---**  
**PROGRAM**

**Instruction** **JOINT 30%**  
**1 Registers**            **5 JMP/LBL**  
**2 I/O**                   **6 CALL**  
**3 IF/SELECT**           **7 Palletizing**  
**4 WAIT**                 **8 ---next page---**  
**PROGRAM**



	1,3.5
	' Perch '
	A R[3]
	R[6]



	CALL SUBPRG (1, R[3], AR[1])
	Vacuum hand open (2.5)

```

IF R[1]=3, CALL PROC_5      ! IF R[1]<>3, JMP LBL[1]
                           CALL PROC_5 (1, R[2])
                           LBL[1]

```

	R[1]=AR+R[2]+AR[4] IF R[1]=AR[1], JMP LBL[1]
	AO[1]=AR[2] GO[1]=AR[2]
	IF AO[1]=AR[1], JMP LBL[1] WAIT GI[1]<>AR[2], TIMEOUT, LBL[1]
	UTOOL_NUM=AR[4]
	R[AR[1]]=R[AR[2]] SDO[AR[1]]=ON
	CALL SUBPRG1 (AR[5])
	hand 3 open (AR[1])

10

1 16 0

```

if R[AR[1]]
  R[R[AR[1]]]

```

ENTER " ! " " # "

[CHOICE].

CALL PROC\_1  
[CHOICE]

1 R[ ]	5 <None>	%
2 Constant	6 <Insert>	#123
3 String	7	
4 AR[ ]	8	

[CHOICE]

1

```
Parameter select      1: CALL PROC_1 (      )
1 R[ ]
2 Constant
3 String              #124
4 AR[ ]
1: CALL PROC_1 ( 1 )
```

[CHOICE]

```
Parameter select      1 PARTS          5 DEV          %
1 R[ ]                2 TOOL          6 PALT          #126
2 Constant            3 WORK          7 GRIP
3 String              4 POS          8 --- next page ---
4 AR[ ]
MAIN
1: CALL PROC_1 ( 1 ) 1/2
[End]
```

[CHOICE] [STRING]

F4

```
String select        JOINT 10%
1 Parts ITEM1        5 Parts ITEM5
2 Parts ITEM2        6 Parts ITEM6   #127
3 Parts ITEM3        7 Parts ITEM7
4 Parts ITEM4        8 --- next page ---
```

```
1: CALL PROC_1 ( ' Parts ITEM2 ' ) Parts ITEM2
```

STRINGS

```
JOINT 10%
1 Words
2 Upper Case
3 Lower Case
4 Options
MAIN
1: CALL PROC_1 ( 'Too11 ' ) 1/2 #130
[End]
$ [ ] [ PART ] [ POS ] [ TOOL ]
F1 F3 F5
```

ENTER

```
1: CALL PROC_1 ( ' Tool 12 ' )
```

**JOINT 10%**

**CALL PROC\_1 (1, 'T00L12')** #132

**[CHOICE] CHANGE**

[CHANGE]

[CHOICE]

AR[]

**Parameter select**

**1 R[ ]**

**2 Constant**

**3 String**

**4 AR[ ]**

#133

1: CALL PROC\_1 (AR[ ... ])

Enter an index

1: CALL PROC\_1 (AR[1])

[INDIRECT]

AR[R[...]]! AR[AR[...]]! AR[R[...]]!

[CHOICE]

4R[]

**Parameter select**

**1 R[ ]**

**2 Constant**

**3 String**

**4 AR[ ]**

#134

1: CALL PROC\_1 (R[ ... ])

Enter an index

1: CALL PROC\_1 (R[1])

[INDIRECT]

R[R[...]]! R[AR[...]]! R[R[...]]!

1: CALL PROC\_1 (1

[CHOICE]

1: CALL PROC\_1 (1,Constant)

1: CALL PROC\_1 (1, Constant)

1: CALL PROC\_1 (1, 2)

2

**JOINT 10%**

5 <None>

6 <Insert>

#151

1: CALL PROC\_1 (1 [ ]  
[CHOICE]

< >

1: CALL PROC\_1 (1, [ ] , 2 )

1: CALL PROC\_1 (1, R[ ] , 2)

1: CALL PROC\_1 (1, R[3], [ ] )  
" ) "

" 3 "

**JOINT 10%**

5 <None>

6 <Insert>

#157

1: CALL PROC\_1 (1, [ ] , 3)  
[CHOICE]

<None>

1: CALL PROC\_1 (1, [3]  
" ) "

<None>

1: R[1]=AR[ ... ]  
 1: R[1]=AR[ 1 ]

AR[]

#167

**DIRECT INDIRECT [CHOICE]**

F3

1: WAIT R[R ... ]  
 1: WAIT R[AR ... ]

F3

F3

AR[1]

PROC\_1

5

**MAIN**

**10: CALL PROC\_1 ('ABCD')**



**5: R[1]=AR[1] X**

PROC\_1

PROC\_1 6

**MAIN**

**10: CALL PROC\_1 (1, 2)**

**30: CALL PROC\_1 (R[1])**



**5: R[1]=AR[1]**

**6: R[1]=R[1]+AR[2] X**

---

index  
 1: CALL PROC\_1 ( Constant )  
 2: CALL PROC\_1 (R[ ... ] )  
 ” INTP--201 Unspecified Statement( ) ”

	\$STRING_PRM=TRUE/FALSE Standard value=FALSE	
	\$ARG_STRING[i].\$TITLE (i = 1 to 10)	16
	\$ARG_STRING[i].\$ITEMJ (i = 1 to 10, j=1 to20)	16
	\$ARG_WORD [i] (i = 1 to 5)	7

KAREL

---

Instruction		JOINT 30%
1 Miscellaneous	5 MACRO	
2 Program control	6 SENSOR	
3 SKIP	7 Multiple control	
4 Offset	8 ---next page---	
PROGRAM		

RSR

RSR  
 RSR[i]=(value )  
 RSR RSR  
 RSR

## RSR [ i ] (value)

(1 to 4)

ENABLE: Enable RSR function  
 DISABLE: Disable RSR function

Example **RSR[2:Workproc.2.] = ENABLE**

RSR [ i ] (value) RSR [ i ] ( )  
 ENABLE: Enable RSR function RSR  
 DISABLE: Disable RSR function RSR

UALM[i]

3.15

\$UALM\_MSG

---

UALM [i]

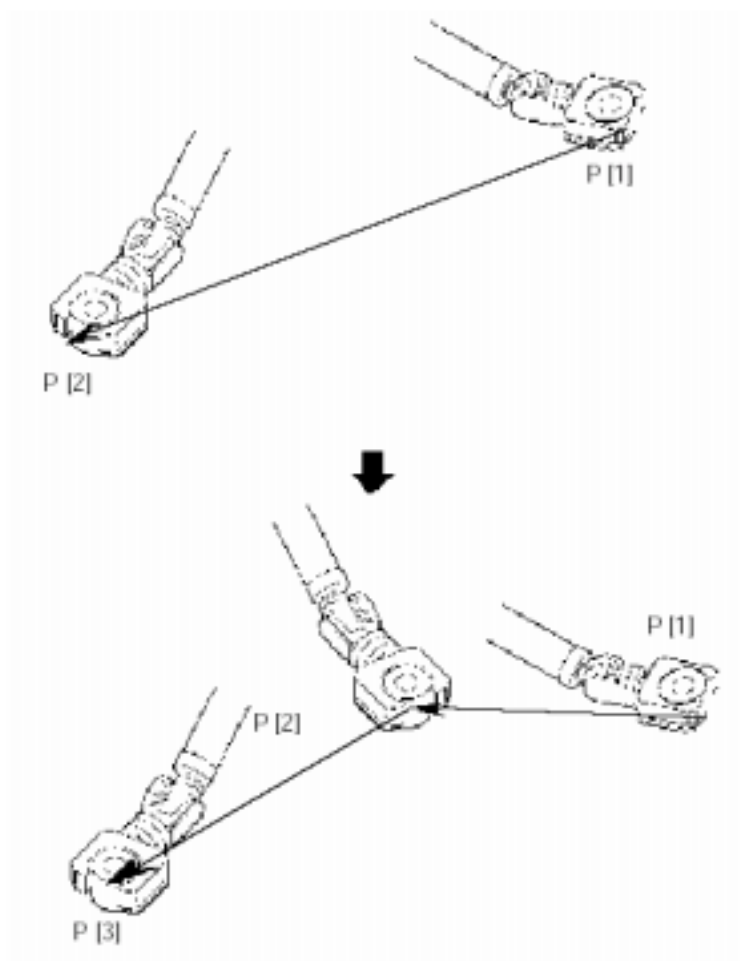
Alarm number

Example **1: UALM[1] (\$UALRM\_MSG[1] = WORK NOT FOUND**

Alarm number







1

2

3

4

5

6

7



---

1  
2  
3  
4  
5  
6  
7  
8

3

6

6 7

---





Pounce Position

Pounce position

HOME  
HOME

FINE  
CNT



	f	f	
	f	 [10%]	
	f	N( 3)	
	f	 [ ]	
	f( 1)	 [ ]	
TP	n 2	 [ ]	

F:

N:



1

SDO

[6 SYSTEM Config

]

2

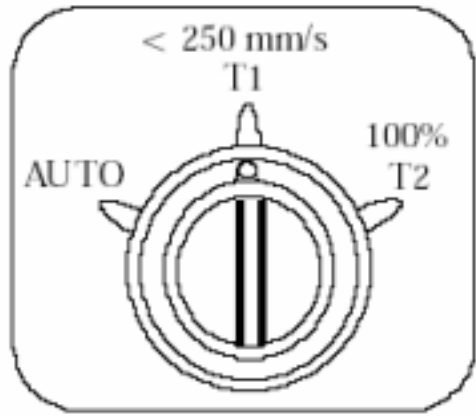
3



---

AUTO

T1 T2



AUTO mm/s /

CE RIA

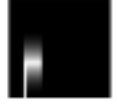
T2



---

deadman

deadman



T2

T2

CE RIA

T2

RESET

---

AUTO

RIA

AUTO

\$MNSING\_CHK

TRUE

SHIFT POINT

SHIFT

TOUCH UP

UF

0

TPIF--060 Can't record on cartesian (G:1)TPIF --060

(G:1)

MOTN--023 In singularity MOTN-- 023

i: Move group number at a singular point

YES NO

YES

NO

/



---

(end)

" ABORTED "

<b>SAMPLE1</b>	<b>LINE 7</b>	<b>ABORTED</b>
----------------	---------------	----------------

( )

" PAUSED "

<b>SAMPLE1</b>	<b>LINE 7</b>	<b>PAUSED</b>
<b>SAMPLE1</b>		<b>JOINT 30%</b>

paused

IMSTP

HOLD

HOLD

1 ABORT(ALL)

CSTOPI

( )

1

PAUSED

FAULT



**SRV0-002 Teach Pendant E-Stop**  
**SAMPLE1 LINE 2 ABORTED**  
**SAMPLE1 JOINT 30%**

emergency stop button

2

3



---

HOLD

HOLD

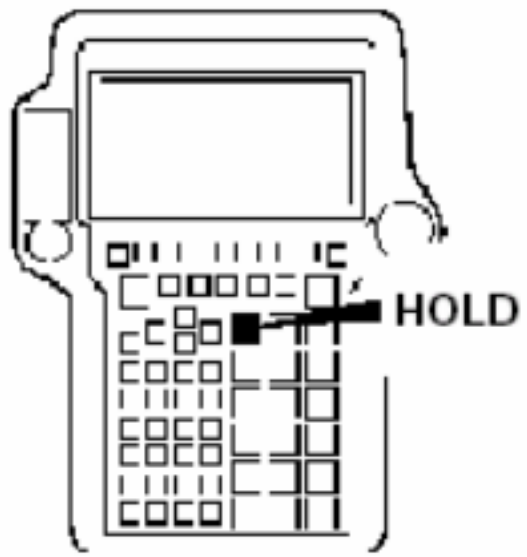
HOLD

SETUP General

1

HOLD

PAUSED

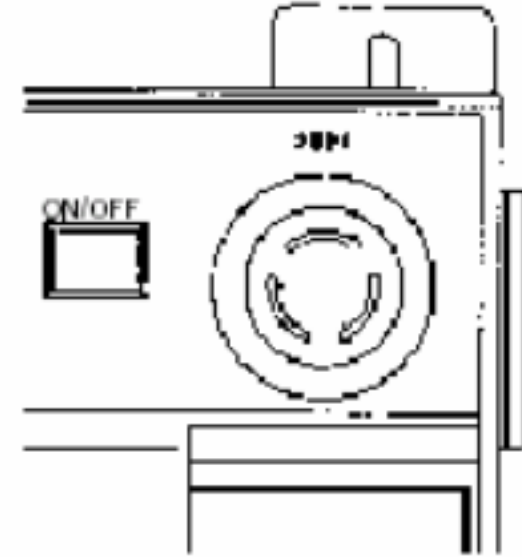
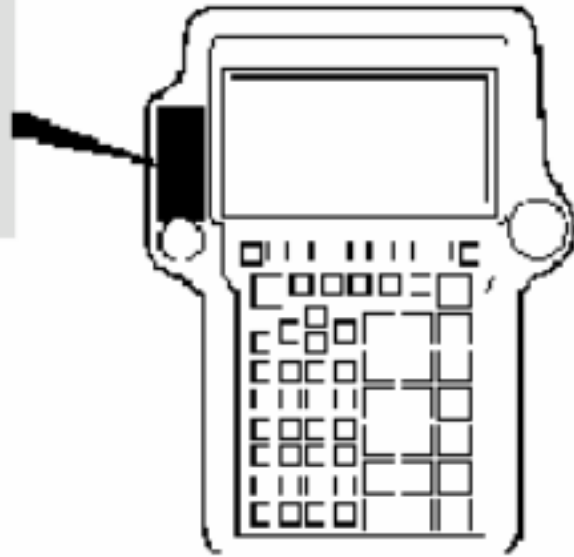


HOLD

2

FAULT

- FAULT
- HOLD
- STEP
- BUSY



Alarm code  
INTP--224  
ID No.  
Alarm detail code  
MEMO--027

**INTP-224 (SAMPLE1, 7) Jump label is fail**  
**MEMO-027 Spedified line does not exist**  
**Alarm JOINT 30 %**  
**1/7**  
**1 INTP-224 (SAMPLE1, 7) Jump label is**  
**2 SRV0-002 Teach pendant E-stop**

4 ALARM

3 MANUAL FCNTS  
4 ALARM  
5 I/O

MENUS

CLEAR HELP

INTP-224 (SAMPLE1, 7) Jump label is fail

INTP-224 (SAMPLE1, 7) Jump label is fail  
MEMO-027 Spedified line does not exist  
30-MAY-98 07:15  
STOP.L 00000110

Alarm history 1/7

1 INTP-224 (SAMPLE1, 7) Jump label is  
2 SRV0-002 Teach pendant E-stop  
3 R E S E T  
4 SRV0-027 Robot not mastered(Group:1)  
5 SYST-026 System normal power up

[ TYPE ] CLEAR HELP

4 ALARM

F5 HELP

CLEAR HELP

INTP-224 (SAMPLE1, 7) Jump label is fail  
MEMO-027 Spedified line does not exist  
30-MAY-98 07:15  
STOP.L 00000110

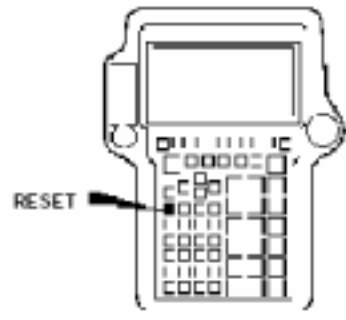
Alarm History

1 Alarm code  
2 Alarm detail code  
3 Generation date/time  
4 Alarm severity

/

RESET

RESET

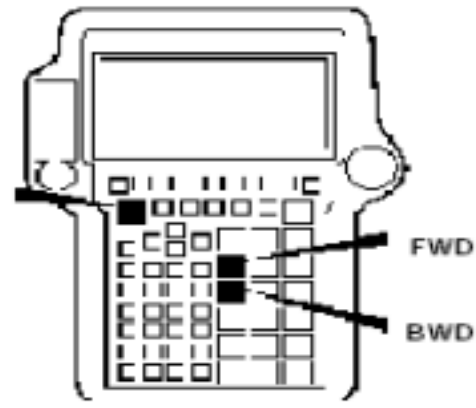


HOLD

6 SETUP General

NONE( )				
WARN				
PAUSED.L PAUSED.G				
STOP.L STOP.G				
SERVO				
ABORT.L ABORT.G				
SERVO2				
SYSTEM				

SHIFT      FWD      BWD  
START(      )



- Teach pendant enable switch
  - On
    - ▶ STEP key
    - ▶ On
      - ▶ Step operation
    - ▶ Continuous operation
  - Off
    - ▶ A program is started on the teach pendant.
    - ▶ Cycle operation
    - ▶ A program is started on the operator's panel.
    - ▶ Automatic operation
    - ▶ A program is started in a peripheral.
- Remote/Local switch
  - Off

Teach pendant enable switch	
STEP key	
Remote/Local switch	/
Step operation	
Continuous operation A program is started on the teach pendant	
Cycle operation A program is started on the operator's panel.	
Automatic operation A program is started in a peripheral	





0 mechanical interface

1 9

F

UF

UF

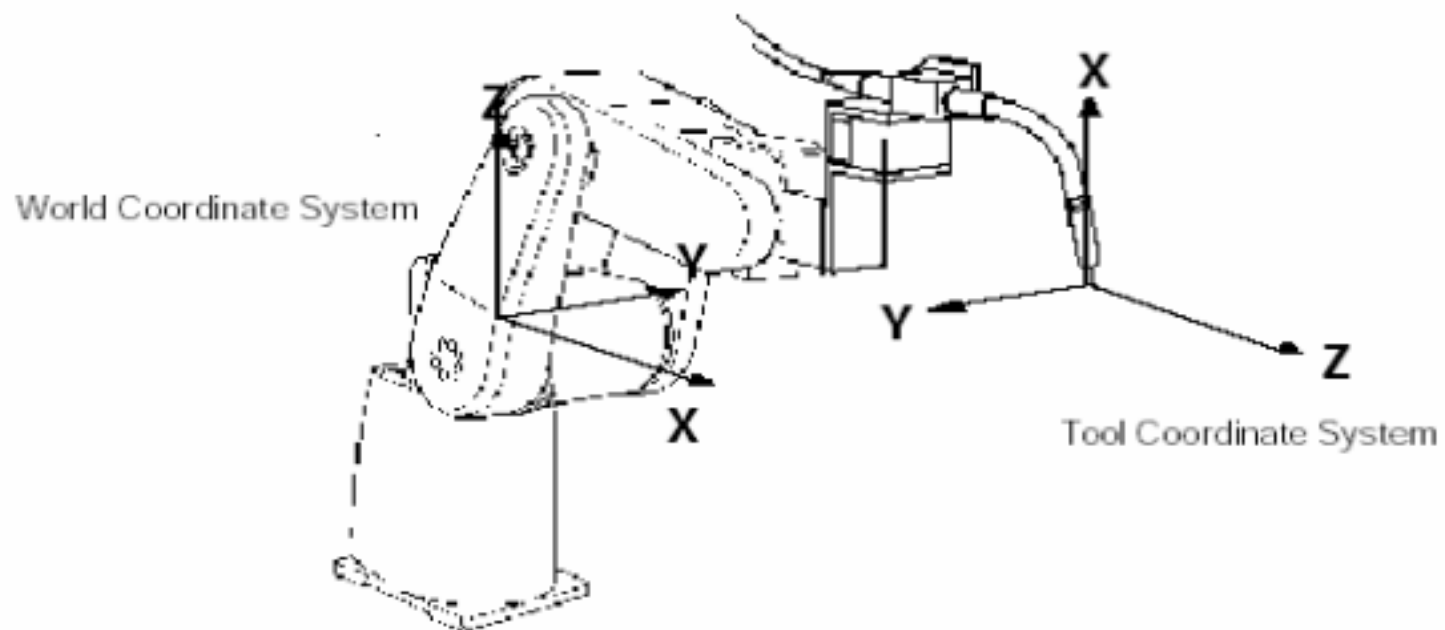
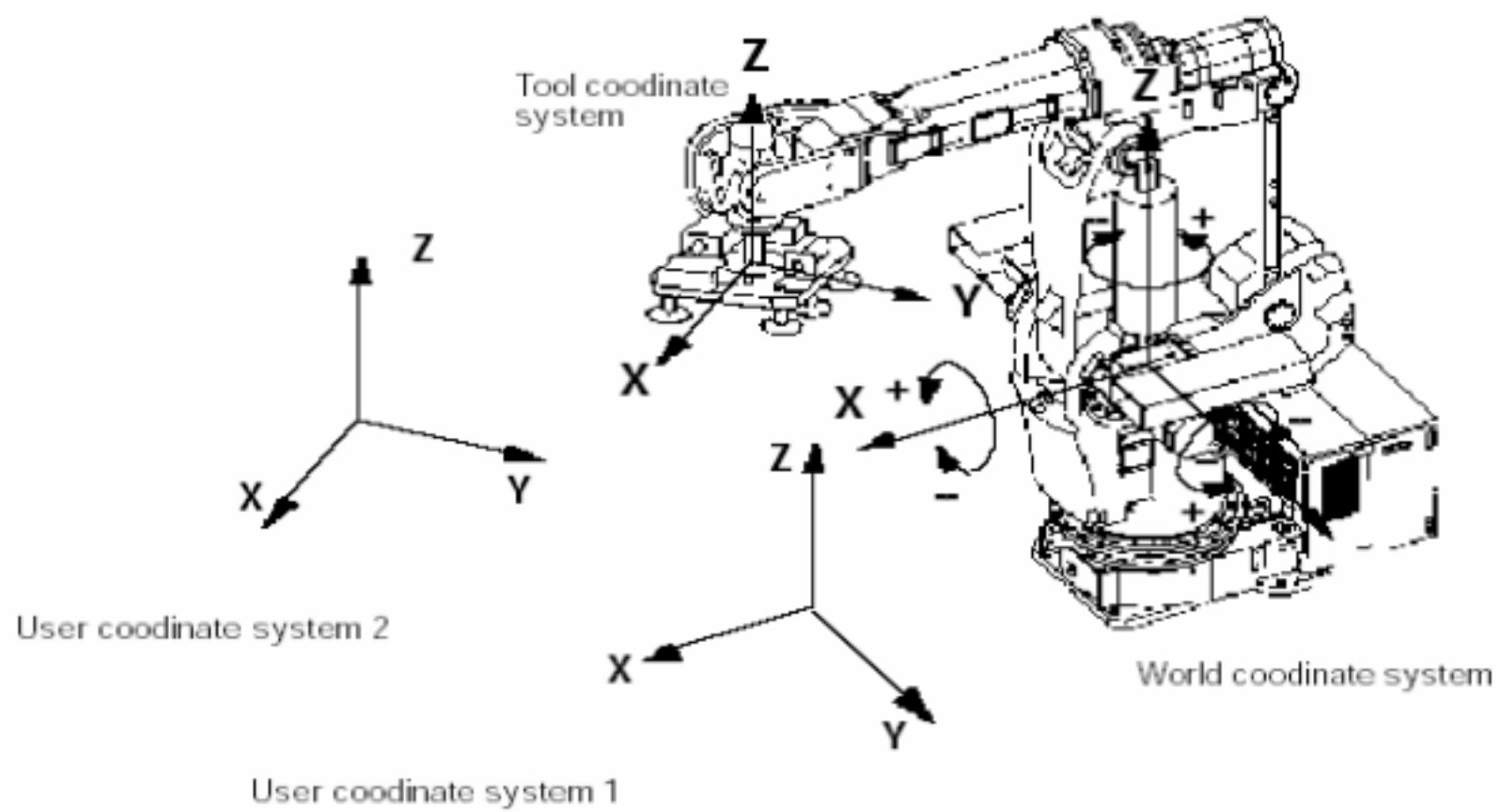
0

1 9

F

F5,[DETAIL]( )

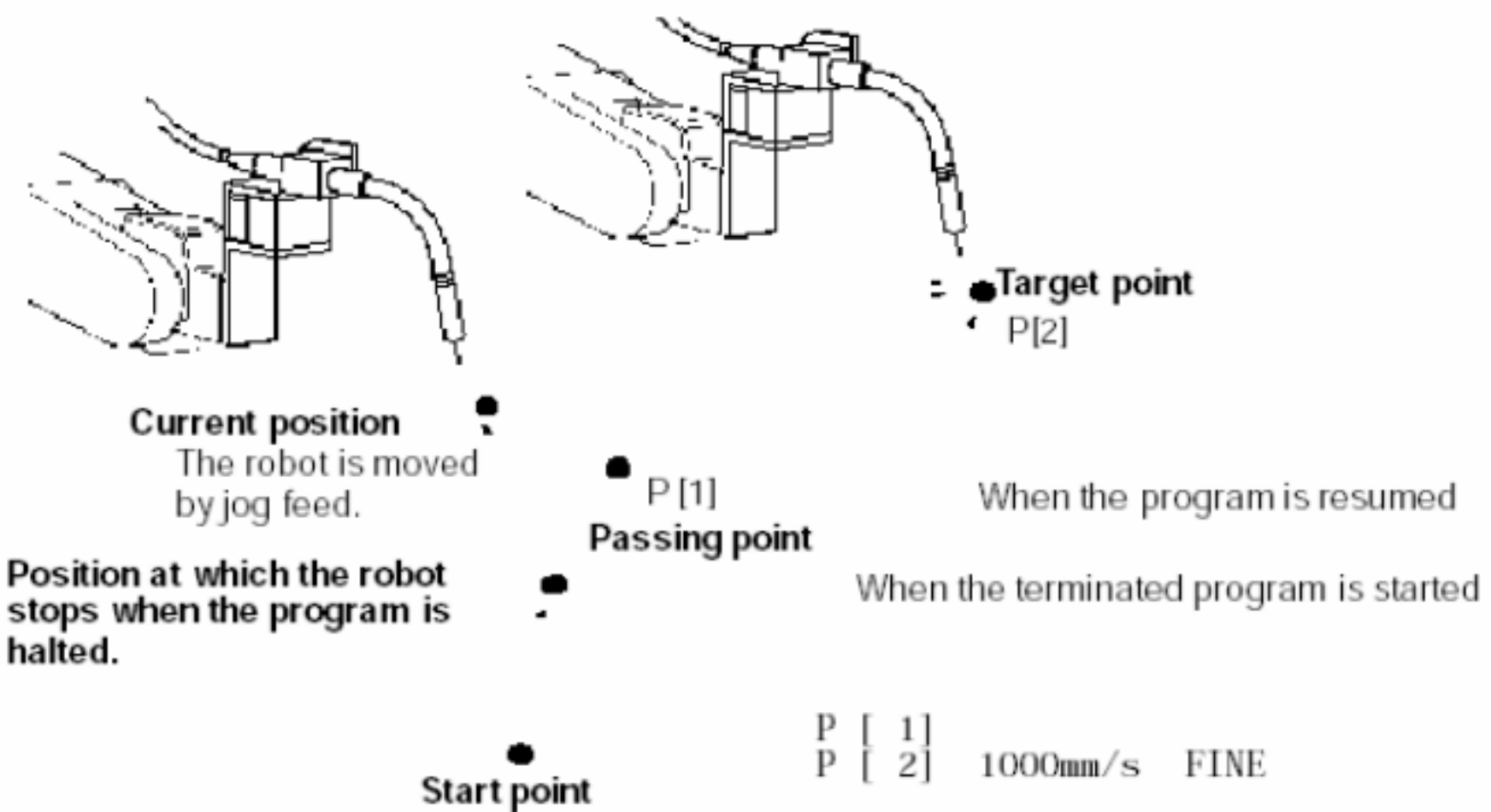
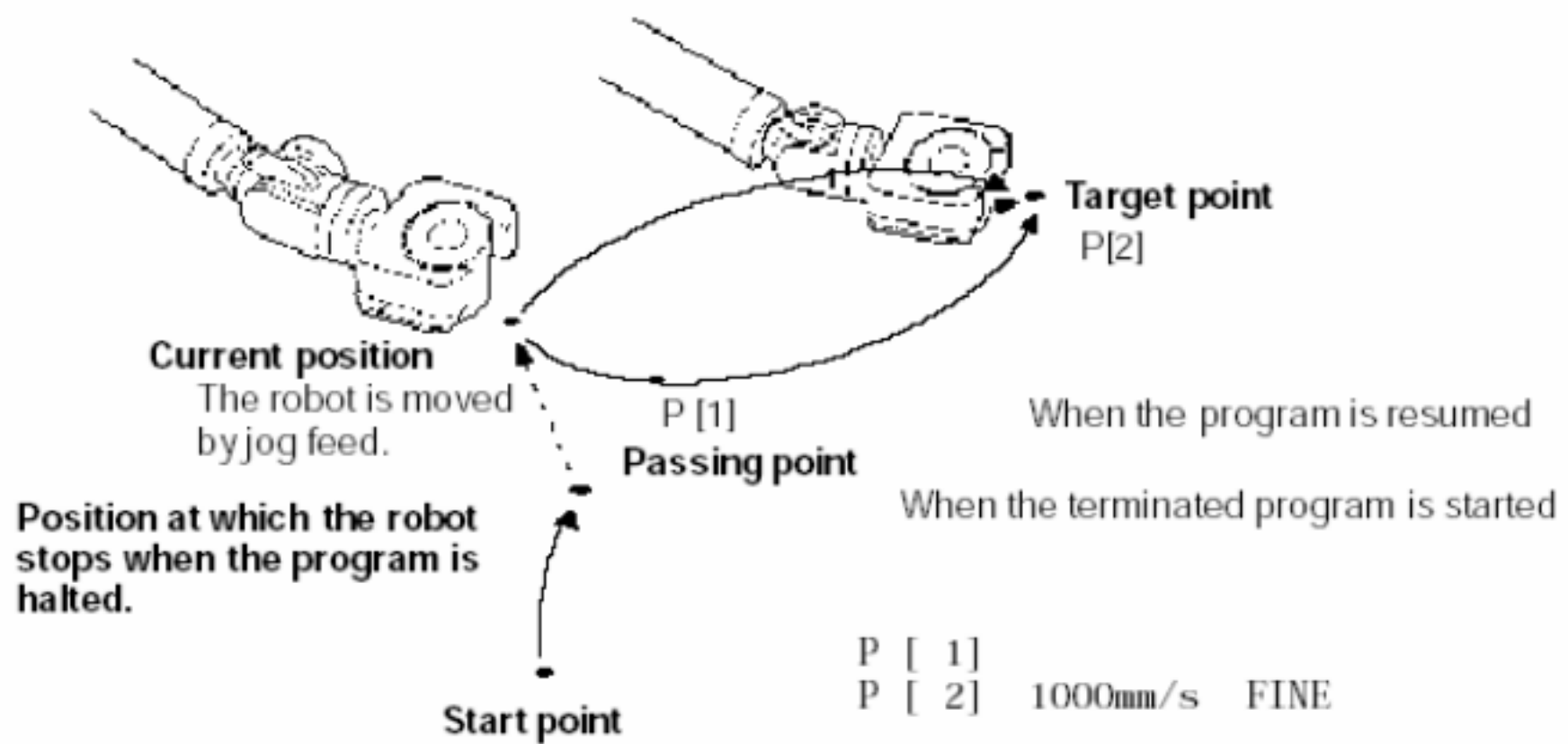
Position Detail		JOINT 30%	
P[1]	UF:0	UT:1	CONF:FT,0
X:	1500.374	mm	W: 40.000 deg
Y:	-242.992	mm	P: 10.000 deg
Z:	956.895	mm	R: 20.000 deg
EDCMD			



Tool coordinate system	
World coordinate system	
User coordinate system	







Current position The robot is moved by jog feed.	
Position at which the robot stops when the program is halted.	
Start point Passing point Target point	
When the program is resumed When the terminated program is started	

---

1 PROGRAM ABORT( )

YES( )

NO( )

YES( ) NO( )

---

H

PAUSED( )

**LINE 2**

**PAUSED**

1 FCTN

2 1 PROGRAM ABORT( )

ABORTED( )

**1 ABORT (ALL)**  
**2**

**LINE 2**

**ABORTED**  
**JOINT 30%**

FCTN

H

PAUSED( )

**LINE 2** **PAUSED**

1

2

3: L P[3] 1(  
4: L P[4] 5(  
5: J P[1] 1(  
[End]

The cursor is on a different  
line from where the program  
PAUSED [2].  
Are you sure you want to run  
from this line ?  
YES NO

3 YES( )

**NO**

**SAMPLE1** **LINE 4** **PAUSED**  
**SAMPLE1** **JOINT 30%**  
**4/6**

1: J P[1] 100% FINE  
2: J P[2] 70% CNT50  
3: L P[3] 1000cm/min CNT30  
4: L P[4] 500mm/sec FINE  
5: J P[1] 100% FINE  
[End]

NO( )

**YES**

AUTO( )

1

2

**SETUP RESUME TOL**

**6/6**

**1 Group : 1**  
**2 Enable Tolerance checking : YES**  
**3 Distance Tolerance (mm) [250.0]**  
**4 Orientation Tolerance (deg) [ 20.0]**

**Axes Tolerance**  
**5 Rotary Axes (deg) [ 20.0]**  
**6 Translational Axes (mm) [250.0]**

**[ TYPE ]**

1.

2.

/

YES( )

YES( )

3.

mm( )

4.

deg( )

5.

deg( )

6.

mm( )

**The robot position is out of  
stop tolerance.  
Please select action.  
Choosing CONTINUE will  
require cycle start.**

**STOP**

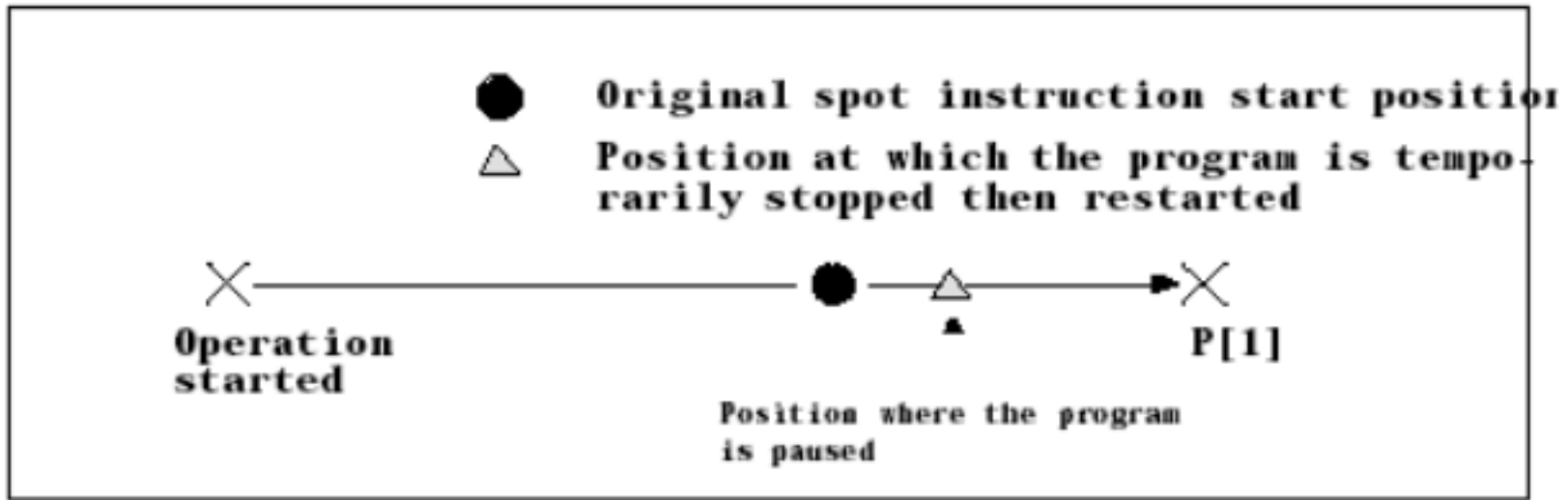
**CONTINUE**

1

STOP( )

STOP( )

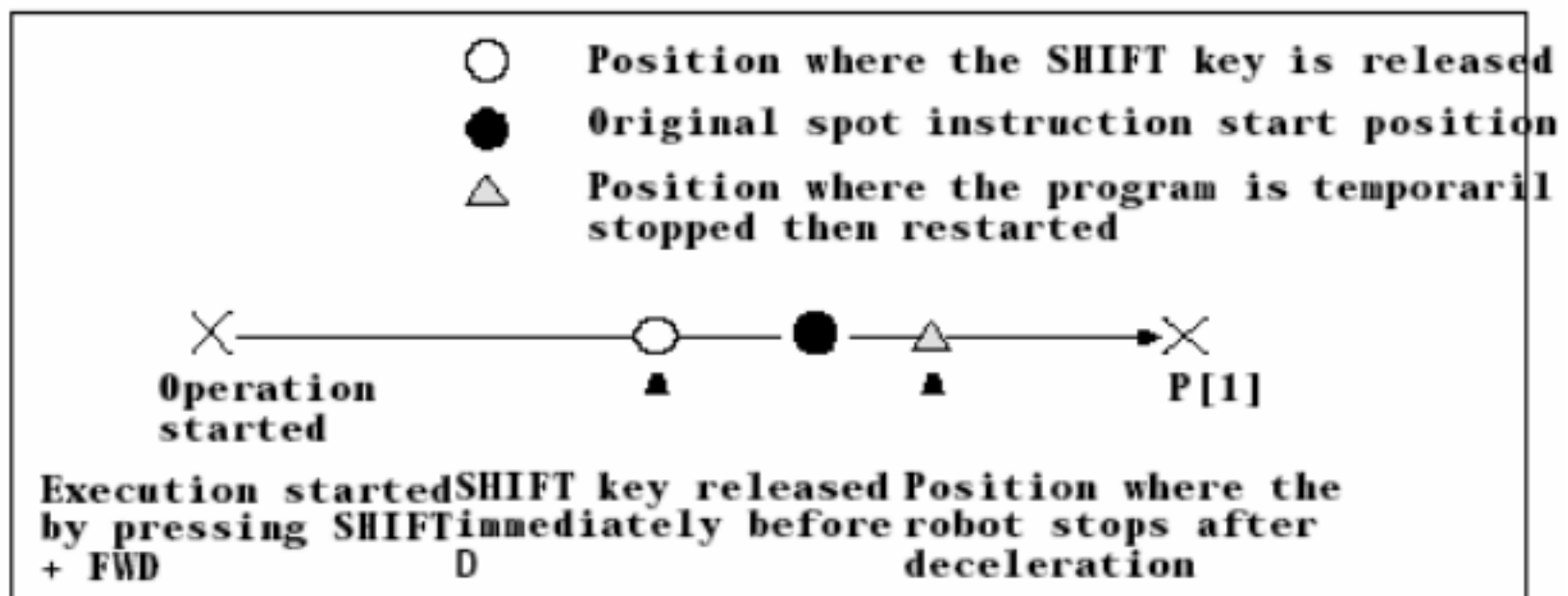
STOP( )



**1: J P[1] 100% FINE**  
**2: SPOT[1]**

SPOT[1]

\*  
\*



...

...



/

OFF

/

I/O( / ) " SFSPD "

I/O( / ) ENBL

1 I/O( / )

2 I/O(

3 / ),

4

**TEST CYCLE Setup** **JOINT 30 %**  
**1/7**

**GROUP:1**

**1 Robot lock: OFF**

**2 Dry run: OFF**

**3 Cart. dry run speed: 300.000 mm/s**

**4 Joint dry run speed: 25.000 %**

**5 Digital/Analog I/O: ENABLE**

**6 Step statement type: STATEMENT**

**7 Step path node: OFF**

**[ TYPE ] GROUP ON OFF**

	<p>ON( )</p> <p>OFF( )</p> <p>ON( )</p> <p>RESET(</p> <p>)</p> <p>ON( )</p>
	Cart dry run speed

	/
/ I/O( / )	/ I/O( / ) I/O( / ) I/O( / ) I/O( / ) I/O S
	STATEMENT( ) CALL TP& KAREL TP&
	ON KAREL MOVE ALONG"

- 1 MENUS( )
- 2 2 TEST CYCLE

- 1 UTILITIES
- 2 TEST CYCLE
- 3 MANUAL FCTNS

**TEST CYCLE Setup**                      **JOINT 30 %**  
1/7

**GROUP:1**

- 1 Robot lock:                      **OFF**
- 2 Dry run:                              **OFF**
- 3 Cart. dry run speed:              **300.000 mm/s**
- 4 Joint dry run speed:              **25.000 %**
- 5 Digital/Analog I/O:              **ENABLE**
- 6 Step statement type:              **STATEMENT**
- 7 Step path node:                    **OFF**

**[ TYPE ] GROUP                      ON              OFF**

3

4

F2 GROUP

1

ON( )

I/O( / )

2

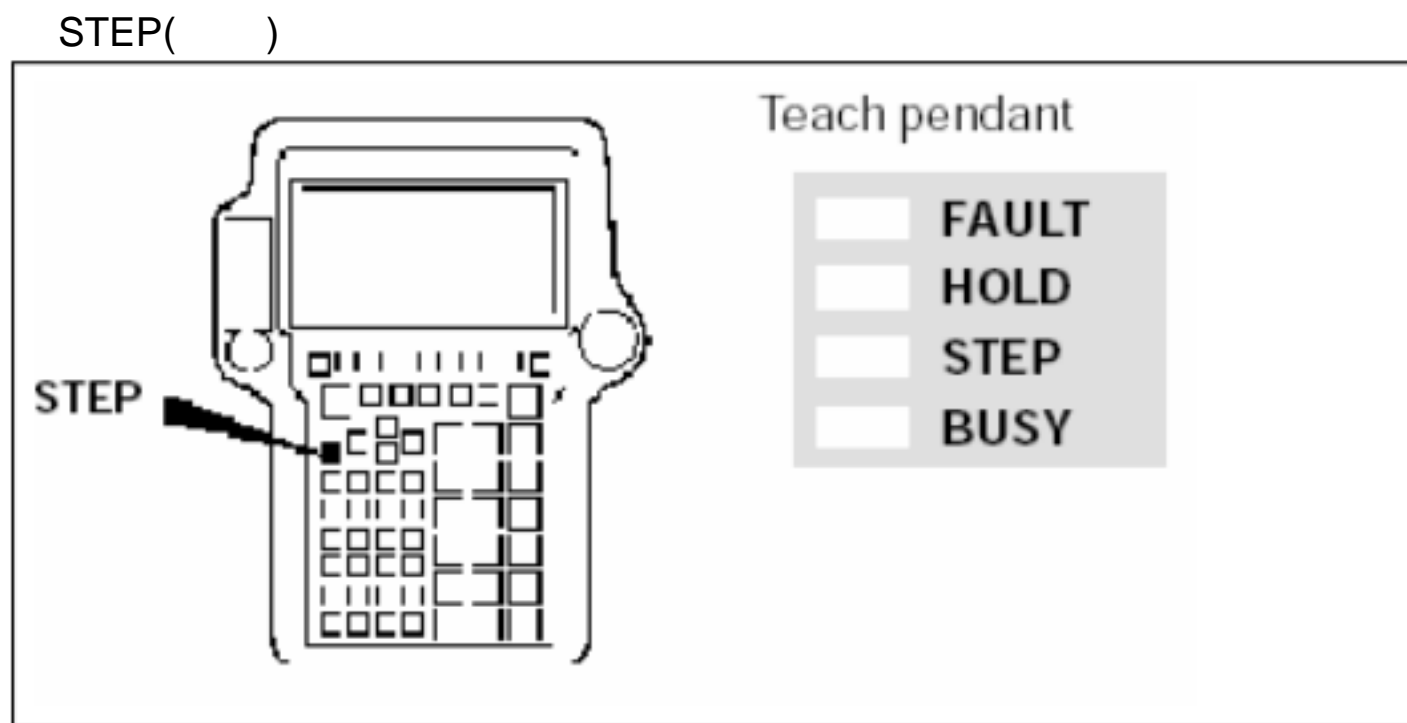
I/O( / )

3

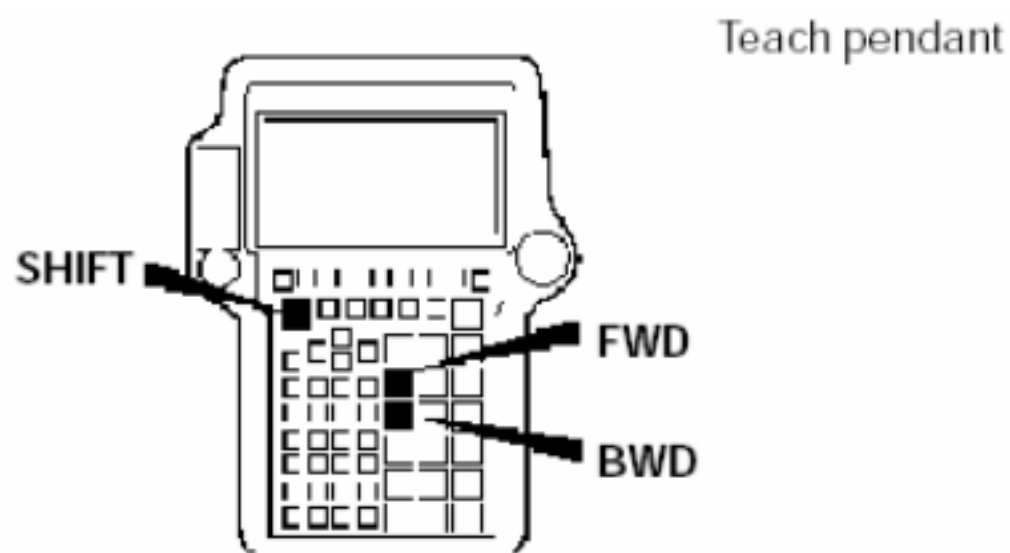
---

4  
5

LED( ) STEP LED( )



FAULT	HOLD	STEP	BUSY



FWD

SHIFT

---

SHIFT

BWD



PAUSE( )  
ABORT( )  
    END( )  
JMP LBL[]  
    UALM[]  
RUN( )  
  
INC

PAUSE

SHIFT+BWD

I/O( / )

I/O( / )

)

I/O( / )

1 MENUS( )

2 7 FILE 7

```

6 SETUP
7 FILE
8

```

MENUS

```

FILE                                JOINT 10%
MC: *.*                              1/17
1 * * (all files)
2 * KL (all KAREL source)
3 * CF (all command files)
4 * TX (all text files)
5 * LS (all KAREL listings)
6 * DT (all KAREL data files)
7 * PC (all KAREL p-code)
8 * TP (all TP programs)
9 * MN (all MN programs)
10 * VR (all variable files)
Press DIR to generate directory
[TYPE] [DIR] LOAD [BACKUP] [UTIL] >

```

3 F5 UTIL , Set Device

```

Set Device
LOAD [BACKUP] UTIL

```

F5

```

JOINT 10%
1 Floppy disk
2 Back up (FRA:)
3
4
FILE
1 * * (all files)
2 * KL (all KAREL source)
3 * CF (all command files)
4 * TX (all text files)
5 * LS (all KAREL listings)
6 * DT (all KAREL data files)
Press DIR to generate directory
[TYPE] [DIR] LOAD [BACKUP] [UTIL] >

```

4 I/O( / )

I/O( / )

Abbreviation	File I/O device
MC :	Memory card
FLPY :	Floppy disk
FRA :	Area used for automatic backup of the F-ROM in the controller

I/O( / )
MC FLPY
FRA F-ROM

FLPY

Floppy Cassette adapter (A16B--0150--B001)

Handy File (A16B--0159--B002)

3.5

	3.5	(2HD)	2DD
Floppy Cassette adapter	2HD, FANUC	71	
Handy File	2HD, FANUC 2HD, MS-DOS 2DD, MS-DOS	71	

RS-232-C

1 Port 1

Floppy adapter	Cassette	9600	baud	2 bit	ISO 0 sec
Handy File		9600	baud	2bit	ISO 0 sec
Handy FMS-DOS		9600	baud	1bit	ISO 0 sec

I/O( / )





---

H /  
H  
1 MENUS( )  
2 NEXT , 1 1 SELECT

**1 SELECT**  
**2 EDIT**

MENUS

Select		JOINT 30%	
56080 bytes free		5/5	
1	PROG1	PR	[PROGRAM001 ]
2	PROG2	PR	[PROGRAM002 ]
3	SAMPLE1	JB	[SAMPLE PROGRAM1 ]
4	SAMPLE2	JB	[SAMPLE PROGRAM2 ]
5	SAMPLE3	JB	[SAMPLE PROGRAM3 ]

[TYPE] CREATE DELETE MONITOR [ATTR] >  
COPY DETAIL LOAD SAVE PRINT >

3

NEXT ,

F4

F4 SAVE

LOAD SAVE PRINT >

F4

JOINT 30%

1 Words  
2 Upper Case  
3 Lower Case  
4 Options

---Insert---

Select

---Save Teach Pendant Program---

Program Name [SAMPLE3 ]

Enter program name

PRG MAIN SUB TEST

4

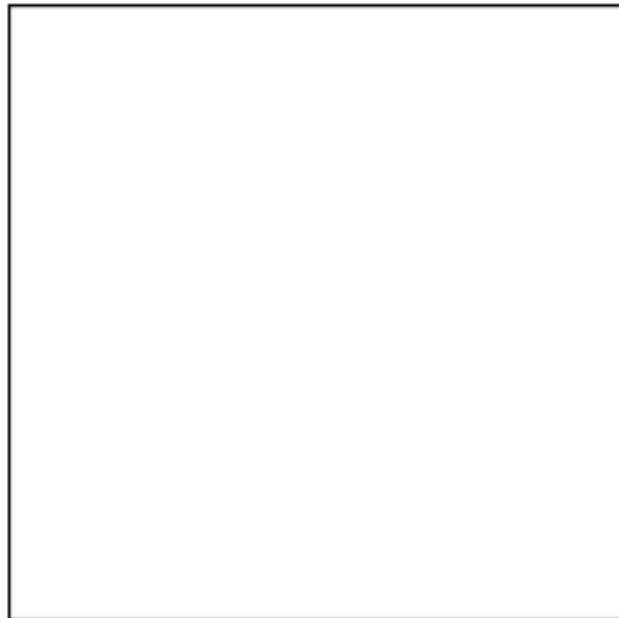
ENTER( )

Select

---Save Teach Pend

Program Name [SAMPLE3] ENTER

5



6

F4,

CONTINUE

No room to save file

CONTINUE CANCEL

```

RAM
F4    BACKUP
      (*.TP)
      (*.DF)
      (*.SV)
      (SYSVARS.SV)
--    (SYSSERVO.SV)
--    (SYSMAST.SV)
--    (SYSMACRO.SV)
--    (FRAMEV AR.SV)
/     (I/O)          (DIOCFGSV .IO)
      (NUMREG.VR)

```

```

PREV
      F4    RESTOR    BACKUP
RESTOR/BACKUP /      BACKUP

```

FCTN

**2 RESTORE/BACKUP**

```

H          /
H
1  MENUS(  )
2    7    7 FILE

```

**6 SETUP**  
**7 FILE**  
**8**

MENUS

```

FILE                                JOINT 30 %
P3: *.*
1 *      *      (all files)
2 *      KL     (all KAREL source)
3 *      CF     (all command files)
4 *      TX     (all text files)
5 *      LS     (all KAREL listings)
6 *      DT     (all KAREL data files)
Press DIR to generate directory
[ TYPE ] [ DIR ] LOAD [BACKUP][UTIL ]>

DELETE COPY DISPLAY >

```

3 F4 BACKUP" ( ) TPE programs TPE

1 System files  
2 TPE programs  
3 Application

LOAD BACKUP [UTIL] >

**F4**

FILE	JOINT 30%
7 *	PC (all KAREL p-code)
8 *	TP (all TP programs)
9 *	MN (all MN programs)
10 *	VR (all variable files)

Save FLPY:\SAMPLE1.TP ?  
EXIT ALL YES NO

-- F2, EXIT ( )

-- F3,ALL

-- F4 YES

-- F5,NO

4 \*.MN

EXIT ALL Saving FLPY:\SAMPLE1.TP, please wait...

**F3**

5

FLPY:\SAMPLE1.TP already exists  
OVERWRITE SKIP CANCEL

-- F3,OVERWRITE

-- F4,SKIP

-- F5,CANCEL

6 F4,SAVE( )

1 System files  
2 TPE programs  
3 Application

LOAD BACKUP [UTIL] >

**F4**

FILE Backup	JOINT 30 %
FLPY:\*.*	

Saving the following files to FLPY:\

DIOCFG.SV  
FRAMEVAR.SV  
NUMREG.VR  
SYSVARS.SV  
SYSSERV0.SV  
SYSMAS.T.SV  
SYSMACRO.SV

Backup to disk?  
YES NO

---

7

F4,YES( )

DIOCFGSV.IO FRAMEV AR.SV

NUMREG.VR SYSV ARS.SV,SYSSERVO.SV,SYSMAST.SV,SYSMACRO.SV

**YES**

**NO**

**Backing up to disk: FLPY:\SYSVARS.SV**

**F4**

8

**FLPY:\SYSVARS.SV already exists**  
**OVERWRITE SKIP CANCEL**

-- F3,OVERWRITE  
-- F4,SKIP  
-- F5,CANCEL

9

F4, CONTINUE

**Disk is full, change to empty disk**  
**CONTINUE CANCEL**

10 F4 BACKUP( ) ALL of above

**ALL of above**

**LOAD BACKUP [UTIL] >**

**F4**

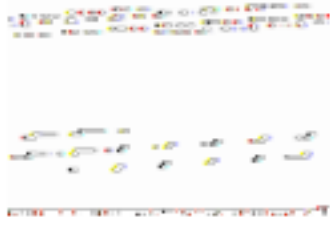
FILE	JOINT	10 %
FLPY: *.*		1/17
1 * *		(all files)
2 * KL		(all KAREL source)
3 * CF		(all command files)
4 * TX		(all text files)
5 * LS		(all KAREL listings)
6 * DT		(all KAREL data files)
7 * PC		(all KAREL p-code)
8 * TP		(all TP programs)
9 * MN		(all MN programs)
10 * VR		(all variable files)
Del Handy File, backup all files?		
	YES	NO

F4 BACKUP( )

2

11 F4,YES





SAVE

(\*TP)

(SYSVARS.SV)

( SYSMAST.SV )

( SYSMACRO.SV )

( FRAMEV AR.SV )

( NUMREG.VR )

( POSREG.VR )

(PALREG.VR)

/ (I/O) / (I/O) ( DIOCFGSV .IP)  
( \*.DF)

H /

H

1

Select			JOINT	30 %
		49828 bytes free	1/5	
No.	Program name	Comment		
1	PR0G001	PR	[PR0GRAM001	]
2	PR0G002	PR	[PR0GRAM002	]
3	SAMPLE1	JB	[SAMPLE PROGRAM 1]	
4	SAMPLE2	JB	[SAMPLE PROGRAM 2]	
5	SAMPLE3	JB	[SAMPLE PROGRAM 3]	

[ TYPE ] CREATE DELETE MONITOR [ATTR ]>

2 FCTN

3 2 SAVE " (2 )

- 1 QUICK/FULL MENUS
- 2 SAVE
- 3 PRINT SCREEN

FCTN

4



5

F4, CONTINUE

**Disk is full, change to empty disk  
CONTINUE CANCEL**



1

**DATA Registers** **JOINT 30 %**  
1/32

R[ 1: COUNTER1 ]=12  
R[ 2: ]=0  
R[ 3: ]=0  
R[ 4: ]=0  
R[ 5: ]=0  
R[ 6: ]=0

[ TYPE ]

2 FCTN

3 2 SAVE " (2 )

**1 QUICK/FULL MENUS**  
**2 SAVE**  
**3 PRINT SCREEN**

FCTN

4

5 F4, CONTINUE

**FLPY-005 Disk is full**

**DATA Registers** **JOINT 30 %**



5

DI

(MC:)

F-ROM

FRA

FRA.

1

99

2

FRA:

Maximum number of versions " ( )

FRA

F-ROM

1

Maximum

number of versions " ( )

F-ROM

200

Maximum

number of versions " ( )

Maximum number of versions " ( )

->8.7.6  
all backup "

(1) Loadable version " ( ) F4(CHoice) ( )

```
1 99/06/16 12:00      5 99/06/14 12:00
2 99/06/15 23:30      6 99/06/13 23:30
3 99/06/15 12:00      7 99/06/13 12:00
4 99/06/14 23:30      8 -- Next Page --
AUTO BACKUP                JOINT 100 %
Version Management-----
13 Maximum number of versions: 1
14 Loadable version: 99/06/16 12:00

[ TYPE ]INIT_DEV          [CHOICE]
```

(2) Loadable version " ( )

(3)

