YASKAWA

YRC1000micro OPTIONS INSTRUCTIONS

FOR RELATIVE JOB FUNCTION

Upon receipt of the product and prior to initial operation, read these instructions thoroughly, and retain for future reference.

MOTOMAN INSTRUCTIONS

MOTOMAN-□□□ INSTRUCTIONS
YRC1000micro INSTRUCTIONS
YRC1000micro OPERATOR'S MANUAL
YRC1000micro MAINTENANCE MANUAL
YRC1000micro ALARM CODES (MAJOR ALARMS) (MINOR ALARMS)

The YRC1000micro alarm codes above consists of "MAJOR ALARMS" and "MINOR ALARMS".

Please have the following information available when contacting Yaskawa Customer Support:
- System
- Primary Application
- Software Version *(Located on Programming Pendant by selecting: {Main Menu} - {System Info} - {Version})*
  Robot Serial Number *(Located on robot data plate)*
  Robot Sales Order Number *(Located on controller data plate)*

Part Number: 181288-1CD
Revision: 0

MANUAL NO.
HW1484476

DANGER

- This manual explains the relative job function of the YRC1000micro system. Read this manual carefully and be sure to understand its contents before handling the YRC1000micro. Any matter not described in this manual must be regarded as "prohibited" or "improper".
- General information related to safety are described in "Chapter 1. Safety" of the YRC1000micro INSTRUCTIONS. To ensure correct and safe operation, carefully read "Chapter 1. Safety" of the YRC1000micro INSTRUCTIONS.

CAUTION

- In some drawings in this manual, protective covers or shields are removed to show details. Make sure that all the covers or shields are installed in place before operating this product.
- YASKAWA is not responsible for incidents arising from unauthorized modification of its products. Unauthorized modification voids the product warranty.

NOTICE

- The drawings and photos in this manual are representative examples and differences may exist between them and the delivered product.
- YASKAWA may modify this model without notice when necessary due to product improvements, modifications, or changes in specifications. If such modification is made, the manual number will also be revised.
- If your copy of the manual is damaged or lost, contact a YASKAWA representative to order a new copy. The representatives are listed on the back cover. Be sure to tell the representative the manual number listed on the front cover.

Notes for Safe Operation

Read this manual carefully before installation, operation, maintenance, or inspection of the YRC1000micro.

In this manual, the Notes for Safe Operation are classified as “DANGER”, “WARNING”, “CAUTION”, or “NOTICE”.

Indicates an imminently hazardous situation which, if not avoided, will result in death or serious injury. Safety Signs identified by the signal word DANGER should be used sparingly and only for those situations presenting the most serious hazards.

Indicates a potentially hazardous situation which, if not avoided, will result in death or serious injury. Hazards identified by the signal word WARNING present a lesser degree of risk of injury or death than those identified by the signal word DANGER.

Indicates a hazardous situation, which if not avoided, could result in minor or moderate injury. It may also be used without the safety alert symbol as an alternative to “NOTICE”.

NOTICE is the preferred signal word to address practices not related to personal injury. The safety alert symbol should not be used with this signal word. As an alternative to “NOTICE”, the word “CAUTION” without the safety alert symbol may be used to indicate a message not related to personal injury.

Even items described as “CAUTION” may result in a serious accident in some situations.
At any rate, be sure to follow these important items.

To ensure safe and efficient operation at all times, be sure to follow all instructions, even if not designated as “DANGER”, “WARNING” and “CAUTION".

DANGER

- Before operating the manipulator, make sure the servo power is turned OFF by performing the following operations. When the servo power is turned OFF, the SERVO ON LED on the programming pendant is turned OFF.
  - Press the emergency stop button on the programming pendant or on the external control device, etc.
  - Disconnect the safety plug of the safety fence. (when in the play mode or in the remote mode)

If operation of the manipulator cannot be stopped in an emergency, personal injury and/or equipment damage may result.

Fig. : Emergency Stop Button

- Before releasing the emergency stop, make sure to remove the obstacle or error caused the emergency stop, if any, and then turn the servo power ON.

Failure to observe this instruction may cause unintended movement of the manipulator, which may result in personal injury.

Fig. : Release of Emergency Stop

- Observe the following precautions when performing a teaching operation within the manipulator's operating range:
  - Be sure to perform lockout by putting a lockout device on the safety fence when going into the area enclosed by the safety fence. In addition, the operator of the teaching operation must display the sign that the operation is being performed so that no other person closes the safety fence.
  - View the manipulator from the front whenever possible.
  - Always follow the predetermined operating procedure.
  - Always keep in mind emergency response measures against the manipulator’s unexpected movement toward a person.
  - Ensure a safe place to retreat in case of emergency.

Failure to observe this instruction may cause improper or unintended movement of the manipulator, which may result in personal injury.

- Confirm that no person is present in the manipulator's operating range and that the operator is in a safe location before:
  - Turning ON the YRC1000micro power
  - Moving the manipulator by using the programming pendant
  - Running the system in the check mode
  - Performing automatic operations

Personal injury may result if a person enters the manipulator's operating range during operation. Immediately press an emergency stop button whenever there is a problem. The emergency stop button is located on the right of the programming pendant.

- Read and understand the Explanation of the Warning Labels before operating the manipulator.

DANGER

- In the case of not using the programming pendant, be sure to supply the emergency stop button on the equipment. Then before operating the manipulator, check to be sure that the servo power is turned OFF by pressing the emergency stop button.
  Connect the external emergency stop button to the 4-14 pin and 5-15 pin of the Safety connector (Safety).
- Upon shipment of the YRC1000micro, this signal is connected by a jumper cable in the dummy connector. To use the signal, make sure to supply a new connector, and then input it.

If the signal is input with the jumper cable connected, it does not function, which may result in personal injury or equipment damage.

WARNING

- Perform the following inspection procedures prior to conducting manipulator teaching. If there is any problem, immediately take necessary steps to solve it, such as maintenance and repair.
  - Check for a problem in manipulator movement.
  - Check for damage to insulation and sheathing of external wires.
- Return the programming pendant to a safe place after use.

If the programming pendant is left unattended on the manipulator, on a fixture, or on the floor, etc., the Enable Switch may be activated due to surface irregularities of where it is left, and the servo power may be turned ON. In addition, in case the operation of the manipulator starts, the manipulator or the tool may hit the programming pendant left unattended, which may result in personal injury and/or equipment damage.

Definition of Terms Used Often in This Manual

The MOTOMAN is the YASKAWA industrial robot product.

The MOTOMAN usually consists of the manipulator, the YRC1000micro controller, manipulator cables, the YRC1000micro programming pendant (optional), and the YRC1000micro programming pendant dummy connector (optional).

In this manual, the equipment is designated as follows:

Equipment	Manual Designation
YRC1000micro controller	YRC1000micro
YRC1000micro programming pendant	Programming pendant (optional)
Cable between the manipulator and the controller	Manipulator cable
YRC1000micro programming pendant dummy connector	Programming pendant dummy connector (optional)

Descriptions of the programming pendant keys, buttons, and displays are shown as follows:

Equipment		Manual Designation
Programming Pendant	Character Keys /Symbol Keys	The keys which have characters or symbols printed on them are denoted with [ ]. ex. [ENTER]
	Axis Keys /Number Keys	[Axis Key] and [Numeric Key] are generic names for the keys for axis operation and number input.
	Keys pressed simultaneously	When two keys are to be pressed simultaneously, the keys are shown with a “+” sign between them, ex. [SHIFT]+[COORD]
	Mode Key	Three kinds of modes that can be selected by the mode key are denoted as follows: REMOTE, PLAY, or TEACH
	Button	Three buttons on the upper side of the programming pendant are denoted as follows: HOLD button START button EMERGENCY STOP button
	Displays	The menu displayed in the programming pendant is denoted with { }. e.g. {JOB}
PC Keyboard		The name of the key is denoted. e.g. Ctrl key on the keyboard

Description of the Operation Procedure

In the explanation of the operation procedure, the expression "Select • • • " means that the cursor is moved to the object item and [SELECT] is pressed, or that the item is directly selected by touching the screen.

Registered Trademark

In this manual, names of companies, corporations, or products are trademarks, registered trademarks, or brand names for each company or corporation. The indications of (R) and TM are omitted.

Contents

1 Relative Job 1-1
1.1 Coordinate Systems 1-1
1.2 Relative Job Shift Functions 1-1
2 Examples of Use of Relative Jobs 2-1
2.1 Shift Function to Offset Workpiece Position Error 2-1
2.2 A Single Manipulator to Work on the Same Type of Workpiece in Different Locations 2-4
2.2.1 By Modifying the Teaching Coordinate 2-4
2.2.2 By Converting the User Coordinate When Operating 2-5
2.3 Using One Job on Multiple Manipulators 2-6
3 Operations Related to Relative Jobs 3-1
3.1 Converting into a Related Job 3-1
3.2 Verification of Information Related to Relative Job 3-3
3.2.1 Verifying Coordinate System 3-3
3.2.2 Verifying Command Positions 3-4
3.3 Instructions Related to Relative Job 3-5
3.3.1 CALL/JUMP 3-5
3.3.2 MFRAME 3-6
3.3.3 Registering an Instruction 3-7
3.4 Editing Relative Jobs 3-9
3.4.1 Blink Indication of Move Instruction on Job Content Window 3-9
3.4.2 Addition and Modification of Steps 3-9
3.4.3 Cut & Paste Function 3-9
3.5 Relative Job Operation Method 3-10
3.5.1 Previous Step Regarded (Constant B-Axis Sign) 3-11
3.5.2 Previous Step Regarded (Minimum R-Axis Movement) 3-13
3.5.3 Type Regarded 3-14
4 Interface with an Easy Offline Teaching System 4-1
4.1 Job Data Format 4-1
4.1.1 JOB 4-3
4.1.2 NAME 4-3
4.1.3 POS 4-3
4.1.4 INST 4-6
4.2 Relative Job Data Examples 4-9

4.2.1 Job for User Coordinate System No. 3, Only for Robot Axis (6-Axis Robot)..... 4-9

4.2.2 Job for User Coordinate System No. 3, Only for Robot Axis (7-Axis Robot)..... 4-10

4.2.3 Job for Robot Axis (6-Axis) + Base Axis (Base Coordinate System) 4-11

4.2.4 Job for Robot Axis (6-Axis) + Base Axis + Station Axis (Base Coordinate System, Single Job) 4-12

4.2.5 Job for Robot Axis (6-Axis) + Base Axis + Station Axis (Base Coordinate System, Coordinate Job) 4-13

4.2.6 Job for Robot Axis (6-Axis) + Robot Axis (6-Axis) (Base Coordinate System, Coordinate Job) 4-14

4.2.7 Job for Robot Axis (7-Axis) + Robot Axis (6-Axis) (Base Coordinate System, Coordinate Job) 4-15

4.3 Configuration of Position Data 4-16

4.3.1 Position Data of Each Axis 4-16

4.3.1.1 Robot Axis 4-16

4.3.1.2 Station Axis 4-16

4.3.1.3 Base Axis 4-16

4.3.2 Position Data of Each Coordinate System 4-17

4.3.2.1 Base Coordinate System..... 4-17

4.3.2.2 Robot Coordinate System 4-17

4.3.2.3 User Coordinate System 4-18

4.4 Manipulator Type 4-19

4.4.1 Flip/No-flip 4-19

4.4.2 R-Axis Angle..... 4-20

4.4.3 T-Axis Angle 4-20

4.4.4 Front/Back 4-21

4.4.5 Upper/Lower Arm (Type Comprised of L-Axis and U-Axis 4-23

4.4.6 S-Axis Angle 4-23

5 Alarm and Error Message List 5-1

5.1 Alarm Messages 5-1

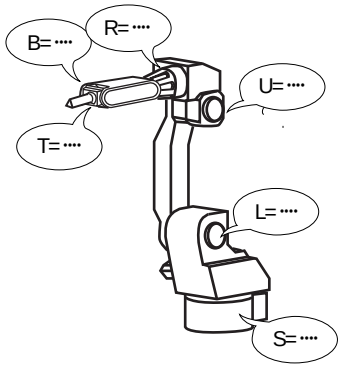
5.2 Error Messages 5-1

6 Instruction List..... 6-1

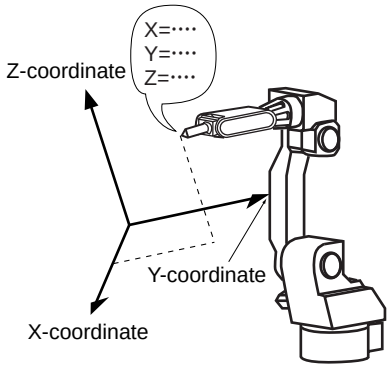
1 Relative Job

In a standard job, each position is defined by a set of pulse numbers, which represent the amount of revolutions of the S, L, U, R, B, and T axes.

In a relative job, however, each position is represented with a set of three values (X, Y, Z) in a specified coordinate system.



Pulse Type Position Data of Standard Job



XYZ-type Position Data of Relative Job

1.1 Coordinate Systems

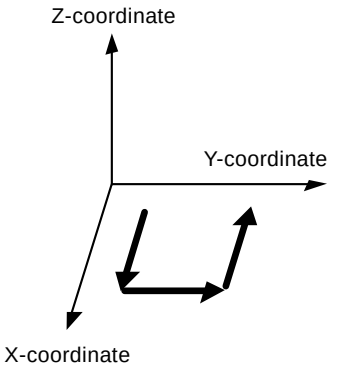
In a relative job, any of the following three types of coordinate systems can be used:

- Base coordinate system
- Robot coordinate system
- User coordinate system (63 systems available)

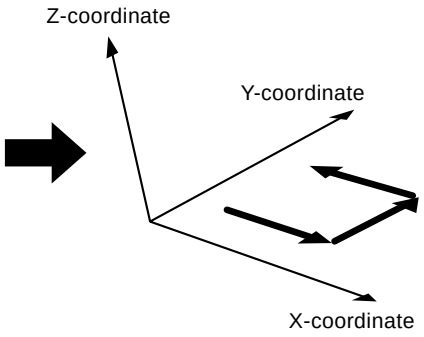
1.2 Relative Job Shift Functions

In a relative job that uses a user coordinate system, changing the definition points to re-determine the coordinate system also changes the coordinates used for the robot operations accordingly.

Also when the operating coordinate system number is changed, the coordinates used for operations are also changed accordingly.



Movement in User Coordinate System No. 1



Relative Job Shifting Operation when Definition Point is Changed

NOTE

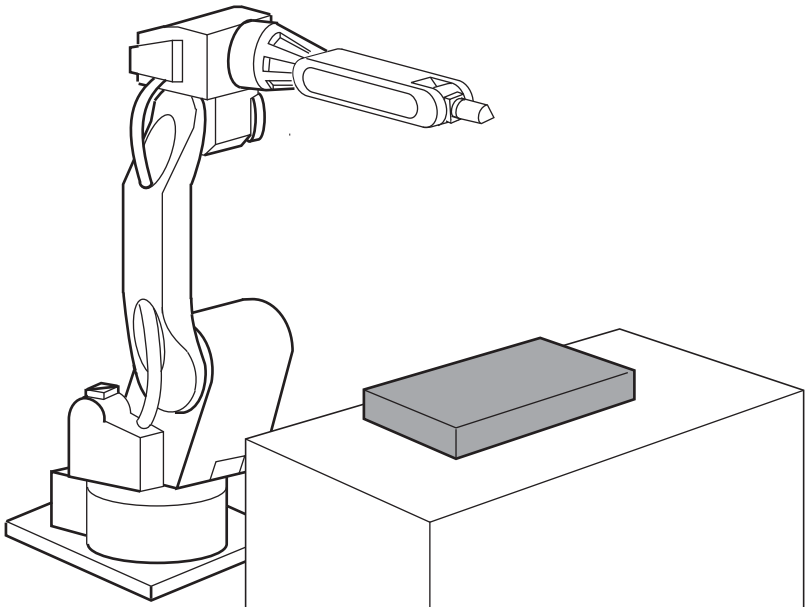
- Changing definition points of the user coordinate system or varying the coordinate system number without due consideration may cause the manipulator to move in an unexpected direction when the job is executed. Be careful when changing the coordinate system.
- Shifting the steps that have been taught by the MOVJ instruction may distort the path. Be careful and avoid interference with jigs and other machinery.

2 Examples of Use of Relative Jobs

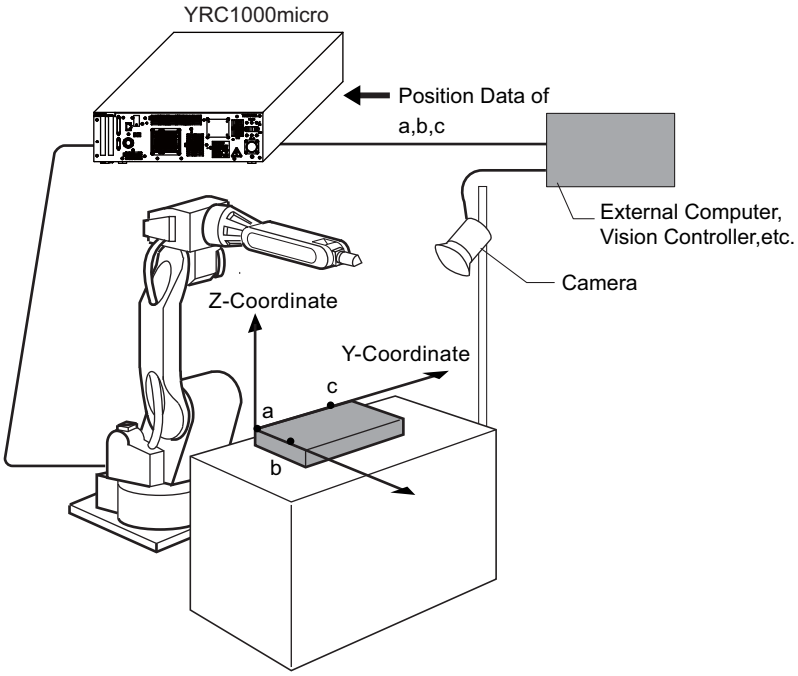
2.1 Shift Function to Offset Workpiece Position Error

After teaching a standard job for a workpiece placed at a reference point, the job is converted into a relative job in a user coordinate system. With a shift function and sensors, possible differences of workpiece positions between teaching and playback can be offset.

1. Place a workpiece at a reference point and teach as usual. Name the job as “STANDARD-1”.



2. Create a user coordinate system for the workpiece. Execute a job that creates a user coordinate system based on the position data of three definition points detected on the workpiece by sensors.




```
NOP

LOADV P000 ← a
LOADV P001 ← b
LOADV P002 ← c

MFRAME  UF#(1)  P000  P001  P002

END
```

Position data of points detected by external sensors is received and stored as position variables.

←User coordinate system created.

3. Create a relative job.
   – Convert “STANDARD-1” created in step 1 into a relative job called “RELATIVE-1” using user coordinate system No. 1 created in step 2.

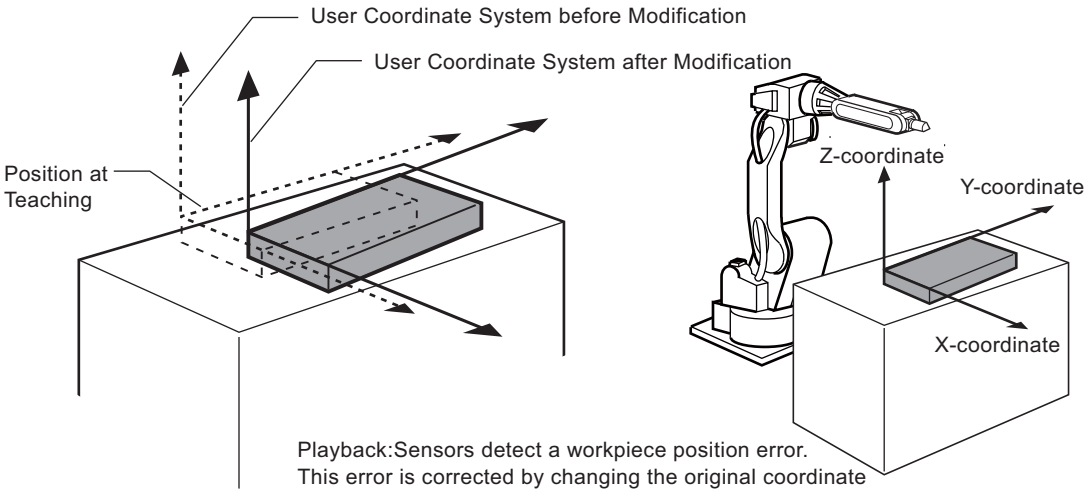
4. Play back the job.
   Use the following job to perform the relative job.

```
NOP
LOADV P000
LOADV P001
LOADV P002
```

Position data of points detected by external sensors is received and stored as position variables.

```
MFRAME UF#(1) P000 P001 P002   ←User coordinate system created
MOVJ VJ=50.0                   ←Moving to waiting position
CALL JOB:RELATIVE-1            ←"RELATIVE-1" of user coordinate system No.1 executed
END
```

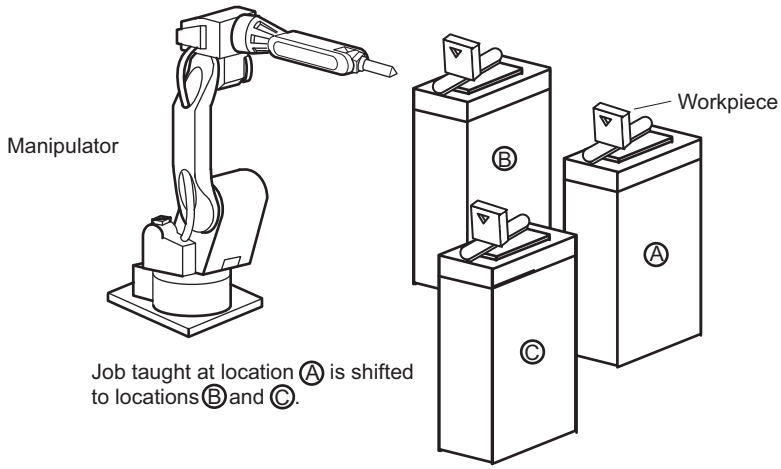
.



Playback:Sensors detect a workpiece position error.
This error is corrected by changing the original coordinate system to create a new user coordinate system.
The workpiece is processed using this new user coordinate system.

2.2 A Single Manipulator to Work on the Same Type of Workpiece in Different Locations

Fig. 2-1: A Single Manipulator to Work on the Same Workpiece in Different Locations



With the relative job function, a single manipulator can easily work on the same type of workpiece in different locations in the following two ways.

- By modifying the teaching coordinate
- By converting the user coordinate when operating

2.2.1 By Modifying the Teaching Coordinate

Create a job for a single workpiece, and then shift the job to another location to create another job.
In this way, create a job at each location, fine adjustment at each position is possible.

1. Create a job for a single workpiece.
2. Teach a user coordinate system (for instance, UF#1) for the workpiece in that location.
3. Convert the job created in step 1 into a relative job in the user coordinate system taught in step 2.
4. Move the workpiece to another location and teach another user coordinate system (for instance, UF#2) for the workpiece in that location.
5. Call up the job header display of the relative job created in step 3 and change the coordinate system to the user coordinate system (UF#2) taught in step 4.
6. Create a new standard job by converting the relative job of the modified coordinate system. This new standard job is to be executed at the location of step 4.
7. To create more jobs for other locations, repeat steps 4 to 6 using other user coordinate system numbers.

2.2.2 By Converting the User Coordinate When Operating

A single relative job can be executed at more than one position. In this way, the memory can be used effectively.

1. Create job “ABCDEF” for a single workpiece.
2. Teach a user coordinate system (for instance, UF#1) for the workpiece in that location.
3. Convert the job created in step 1 into a relative job using the user coordinate system taught in step 2.
4. Move the workpiece to another location and teach another user coordinate system (for instance, UF#2) for the workpiece in that location.
5. Specify the operating coordinate system when calling the work job from a control job.

```
NOP
  :
CALL JOB:ABCDEF UF#2
  :
END
```

   – When the CALL instruction is executed, relative job “ABCDEF” that was taught using UF#1 is performed using UF#2.

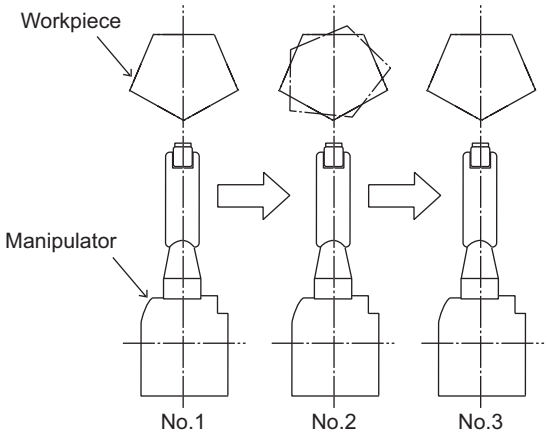
6. To create more jobs to execute at other locations, repeat steps 4 and 5 using other user coordinate system numbers.

The manipulator is not always capable of carrying out the operation depending on the operating position.

Do not force the manipulator to make excessive position changes.

2.3 Using One Job on Multiple Manipulators

Fig. 2-2: Using One Job on Multiple Manipulators



A job taught to one manipulator can be used for other manipulators on the line.

1. On manipulator No.1, create a job for a single workpiece.
2. Teach a user coordinate system (for instance, UF#1) for the workpiece in that location.
3. Convert the job created in step 1 into a relative job using the user coordinate system taught in step 2.
4. Save the created relative job in an external memory device.
5. Set workpieces to manipulators No. 2 and No. 3 to which the job will be shifted. Teach the user coordinate system (UF#1) to the manipulators.
6. Load the relative job saved in step 4 and convert it to standard jobs for manipulators No. 2 and No. 3.
   – This operation creates jobs operated at the posture specified in step 5.

NOTE

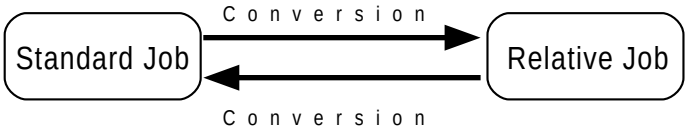
The manipulator is not always capable of carrying out the operation depending on the operating position.

Check the positions by FWD and BWD operations.

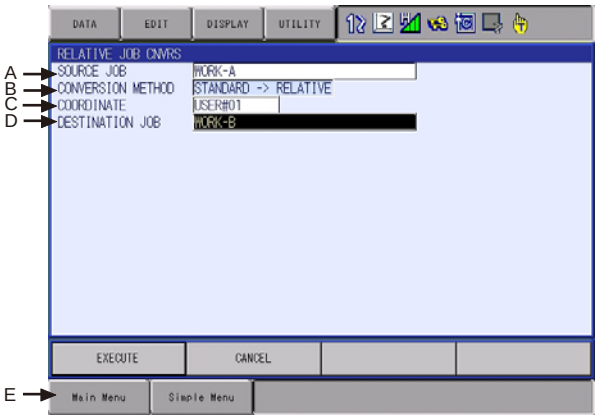
3 Operations Related to Relative Jobs

3.1 Converting into a Related Job

To create a relative job, convert a standard job into a relative job. The conversion of a relative job into a standard job is also possible.



1. Select {JOB} under the main menu.
2. Select {JOB CONTENT}.
3. Select {UTILITY} under the menu.
4. Select {RELATIVE JOB}.
   – Relative job conversion window appears.



A. SOURCE JOB

Selects the job to be converted.

(1) Select {SOURCE JOB} and the job list display is shown.

(2) Select the job to be converted.

B. CONVERSION METHOD

Displays the conversion method.

- STANDARD→RELATIVE: Converts a standard job into a relative job
- RELATIVE→STANDARD: Converts a relative job into a standard job.

C. COORDINATE

Selects a coordinate system where a standard job is converted into a relative job.

(1) Select {COORDINATE}, and the selection dialog is displayed.

(2) Select one either BASE, ROBOT or USER for the conversion destination.

– When selecting “USER”, press [ENTER] after inputting user coordinate numbers.

D. DESTINATION JOB

Sets a destination job.

(1) Select {DESTINATION JOB}.

(2) Enter a job name for the conversion destination.

– When a job name for the conversion destination has been set, a new job is created when converting.
When a job name has not been set, the job at the conversion source itself is used.

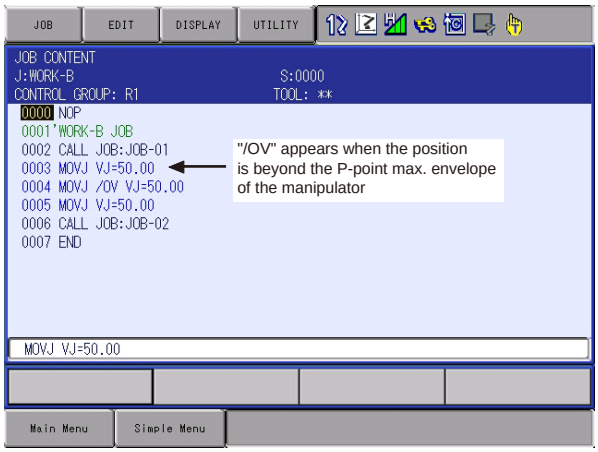
E. EXECUTE

Executes job conversion.

(1) Select {EXECUTE} to execute the conversion.

NOTE

- During conversion, all key operations are unavailable.
- Any alarm during the conversion interrupts the operation.
- When the conversion is completed, the job contents window appears.
- The following display shows that when a relative job has been converted into a standard job, “/OV” appears in the programming steps when the new position is beyond the P-point maximum envelope of the manipulator.
- When “/OV” appears to a step, the pulse of this step is same the pulse of immediately prior step.
- If “/OV” appears to the first step after executing the conversion, robot's present pulse of each axis is regarded as the pulse of the first step.
- The “/OV” indication disappears after modifying the position.

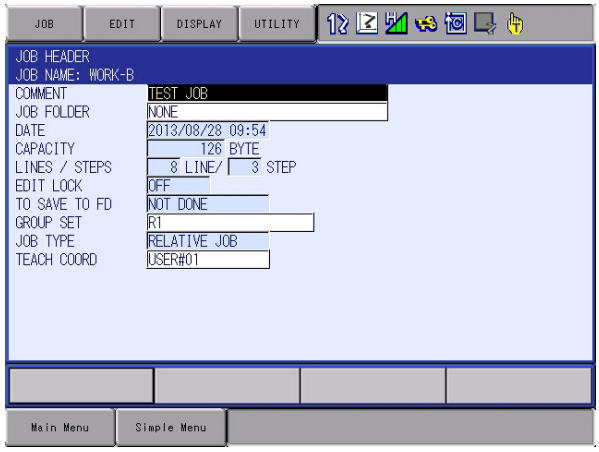


3.2 Verification of Information Related to Relative Job

3.2.1 Verifying Coordinate System

The coordinate system used for teaching can be verified in the job header window.

1. Select {JOB} under the main menu.
2. Select {JOB CONTENT}.
3. Select {DISPLAY} under the menu.
4. Select {JOB HEADER}.
   – The job header window appears.
   – The window is scrolled with the cursor.



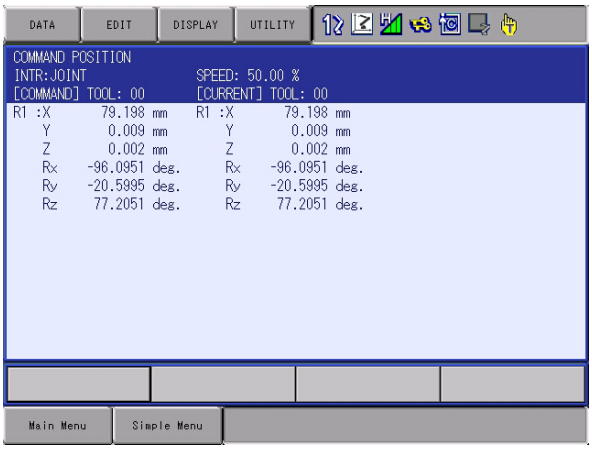
When the coordinate system used for teaching is a user coordinate system, the user coordinate numbers can be changed in this window.

1. Select {TEACH COORD}.
2. Enter the user coordinate number and press [ENTER].

3.2.2 Verifying Command Positions

The command position of XYZ-type can be verified by calling up the command position window for the relative job.

1. Select {ROBOT} under the main menu.
2. Select {COMMAND POSITION}.
   – The command position window appears.



3.3 Instructions Related to Relative Job

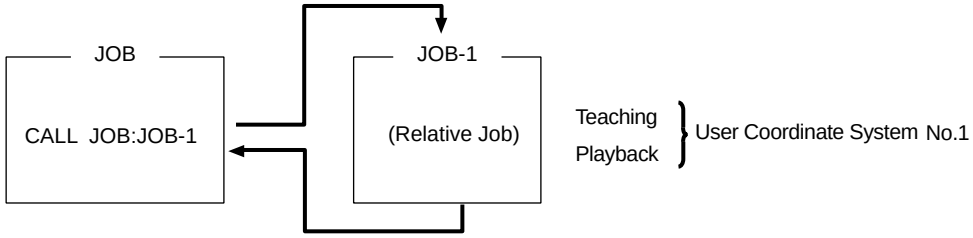
3.3.1 CALL/JUMP

CALL or JUMP are the instructions used to call and execute a relative job.

If no coordinate system number is specified for the job, the job is carried out with the coordinate system used for teaching.

CALL JOB: JOB-1

JUMP JOB: JOB-1 IF IN#(1)=OFF

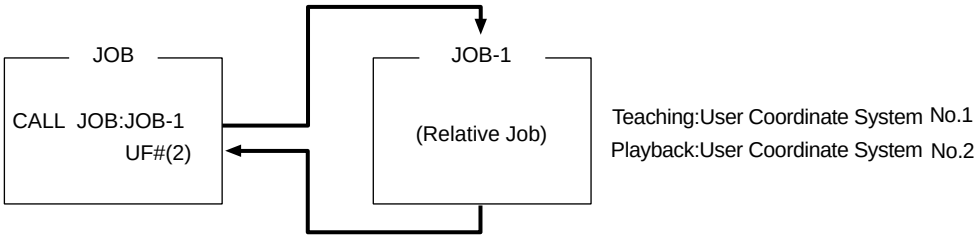


If the job was taught using a user coordinate system, the job can be carried out using another user coordinate system when called by CALL or JUMP.

<Example>

A relative job “JOB-1”, which was taught using the user coordinate system No. 1, is changed to the coordinate system to No. 2 when it is executed. The coordinates of the steps in JOB-1 are converted into the coordinates of coordinate system No. 2.

CALL JOB: JOB-1 UF#(2)

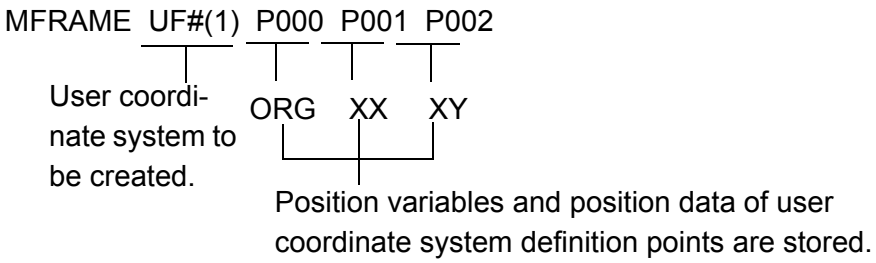


3.3.2 MFRAME

The MFRAME instruction creates a user coordinate system from position data detected by sensors.

<Example>

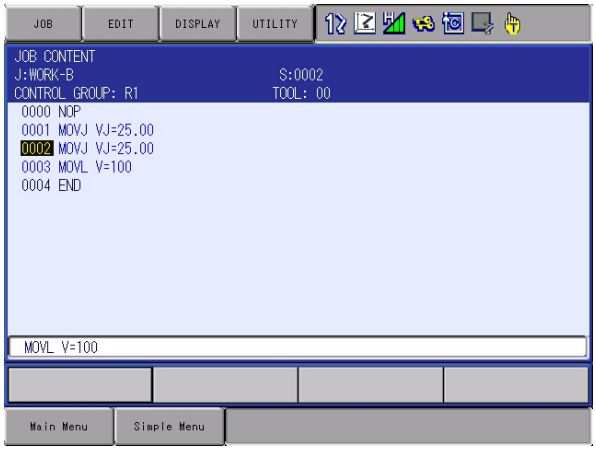
Position data of sensor-detected definition points of the user coordinate system is stored as position variables. A user coordinate system is created using the position variables.



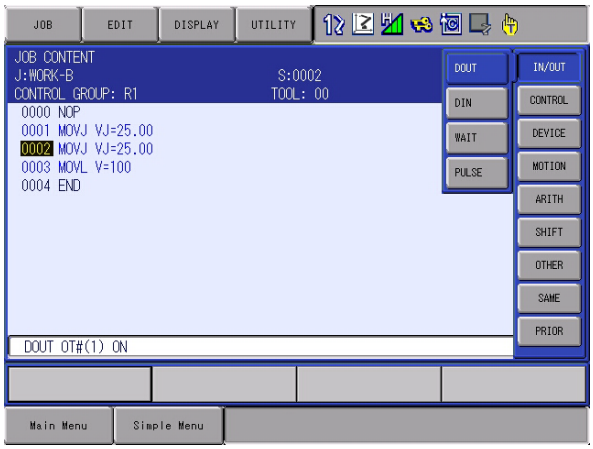
NOTE As for the position variable (Cartesian) to be specified at MFRAME instruction, set the position where the manipulator is capable of carrying out the operations.

3.3.3 Registering an Instruction

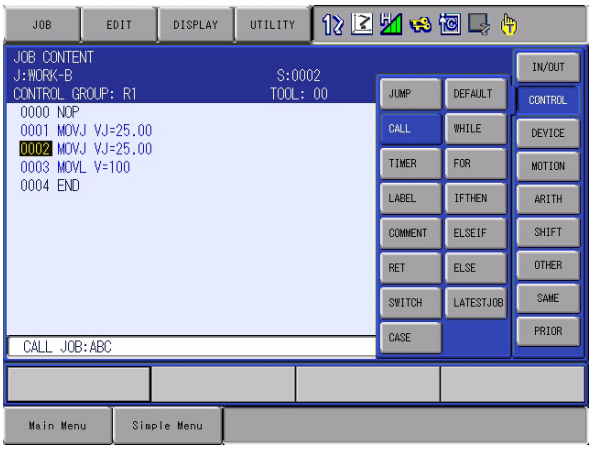
1. Move the cursor to the address area.
2. In the job content window, move the cursor to the line immediately above the place where an instruction is to be registered.
   – In the job content window in the teach mode, move the cursor to the line immediately above the place where an instruction is to be registered.



3. Press [INFORM LIST].
   – The instruction list dialog appears. The cursor moves to the instruction list dialog while the cursor in the address area changes to an underscore.



4. Select an instruction to be registered.
   – Synchronizing with the cursor, instructions appear in the input buffer line the same way as the additional items are registered last time.



5. Change the additional items and variable data.
   – <To register items as they appear in the input buffer>
   Perform operation 6.
   – <To edit any additional items>
   Move the cursor to the additional item to be changed, and then press [SELECT] to display an input line.

   – Move the cursor to the job to additionally register as a CALL instruction, then press [SELECT].

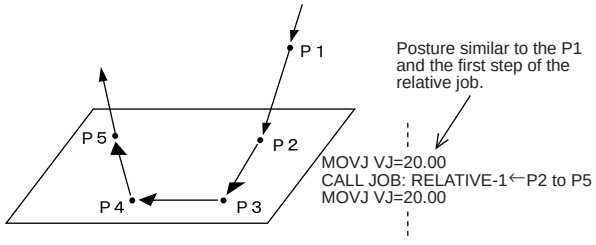
Notes on registering CALL and JUMP instructions

When a relative job is started, the manipulator moves from its current posture.

Therefore, teach the manipulator so that the posture right before calling the relative job is similar to the posture on the first step of the relative job.

If the manipulator starts a relative job in a posture which is extremely different from the posture in the first step, it may move in an unexpected way.

<Example>



6. Press [ADD] and [ENTER].

3.4 Editing Relative Jobs

Relative jobs, like standard jobs, can be edited with the programming pendant to add, modify, and delete positions.

The differences in editing relative jobs and editing standard jobs are explained in this section.

3.4.1 Blink Indication of Move Instruction on Job Content Window

XYZ type current positions are updated by the FWD operation, and are not updated by just calling the job. Therefore, the move instructions merely blink right after calling the job and do nothing else even if the current and the commanded manipulator positions match.

3.4.2 Addition and Modification of Steps

Coordinate system of teaching	Taught coordinate system of a relative job is used. If the taught coordinate system is a user coordinate system and another user coordinate system has been specified for the operating coordinate system, the operating coordinate system is used.
Teaching tool	Coordinates of operation positions are registered by the tool used when the axis is operated.

3.4.3 Cut & Paste Function

Cutting and pasting from a standard job to a relative job is impossible, and vice versa.

Cutting and pasting between relative jobs in different coordinate systems is also impossible.

3.5 Relative Job Operation Method

When a relative job is performed, there are several ways to move to the step position. The following four methods can be used to designate the motion.

Previous Step Regarded (Constant B-axis sign)	Effective for a job of which the B-axis does not pass the point 0°. (i.e., in case of an operation performed with the B-axis pointed downward.)
Previous Step Regarded (Minimum R-axis movement)	Effective for a job of which the B-axis passes the point 0°.
Type Regarded	Effective for a job created in offline teaching is to be performed.
Previous Step Regarded (Minimum R-axis movement + Pulse limit avoidance)	Effective for a job of which the B-axis passes the point 0°, and for the target job may be out of the pulse limit.

The operation method can be specified by the following parameter.

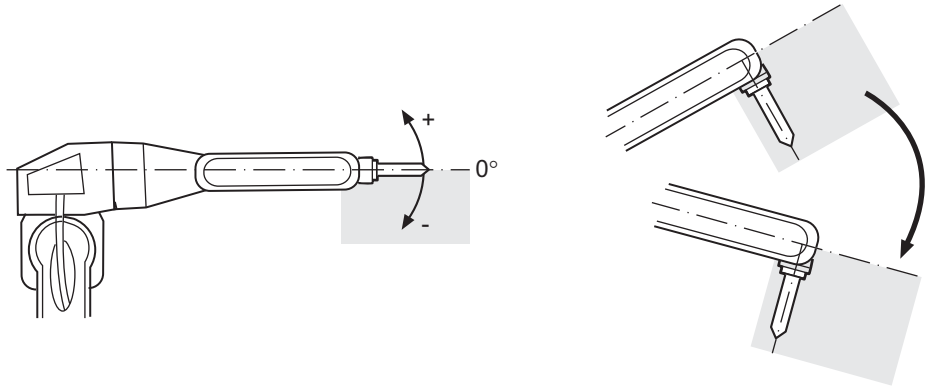
When converting a relative job into a standard job, the method specified by this parameter is also used.

Parameter	Settings	Initial value
S2C430	0: Previous step regarded (constant B-axis sign) 1: Type regarded 2: Previous step regarded (Minimum R-axis movement) 4: Previous Step Regarded (Minimum R-axis movement + Pulse limit avoidance)	4

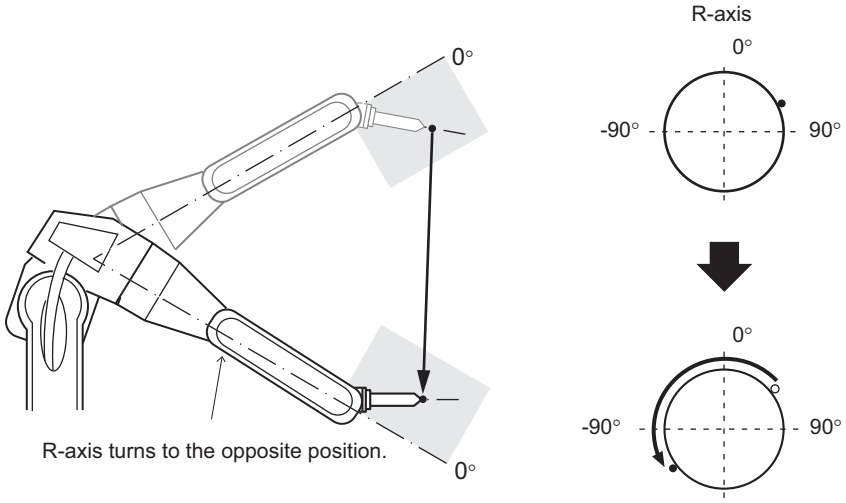
3.5.1 Previous Step Regarded (Constant B-Axis Sign)

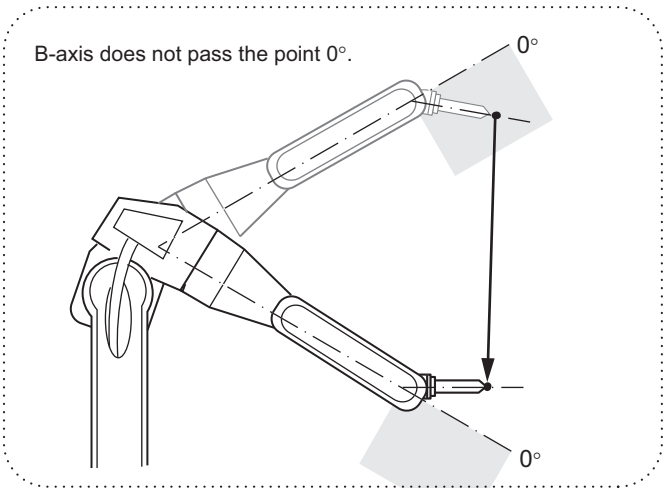
This method allows movement to the specified step in relation to the B-axis angle of the previous step. During operation, movement is made keeping the B-axis angle sign (+/-) constant so that it does not change.

Therefore, it is used for a job of which the B-axis does not pass the point 0°.



If a job of which the B-axis passes the point 0° is executed, the angle of the B-axis stays unchanged even when it should be changed, resulting in the R-axis turning to the position opposite by 180°.





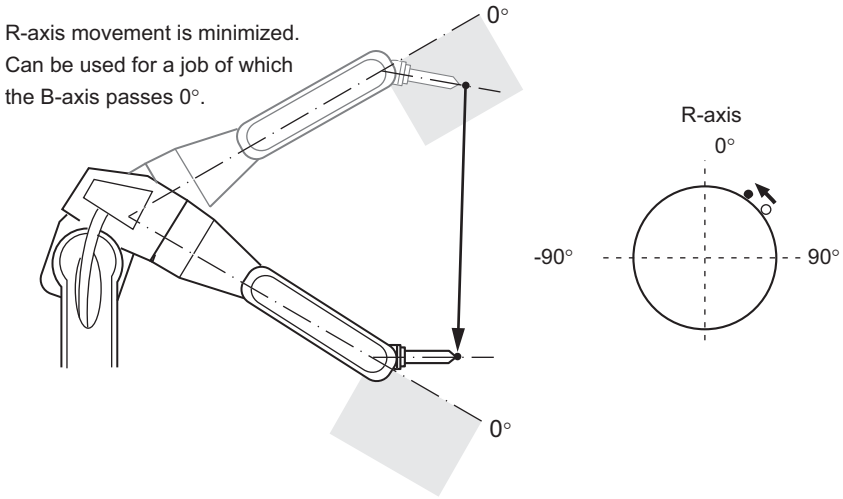
NOTE

- When converting a standard job into a relative job, teach a standard job where the B-axis does not pass 0°.
- When converting a relative job into a standard job, the manipulator's current posture is referred. To perform a conversion, posture the manipulator in a similar posture in the first step of the relative job to be converted.

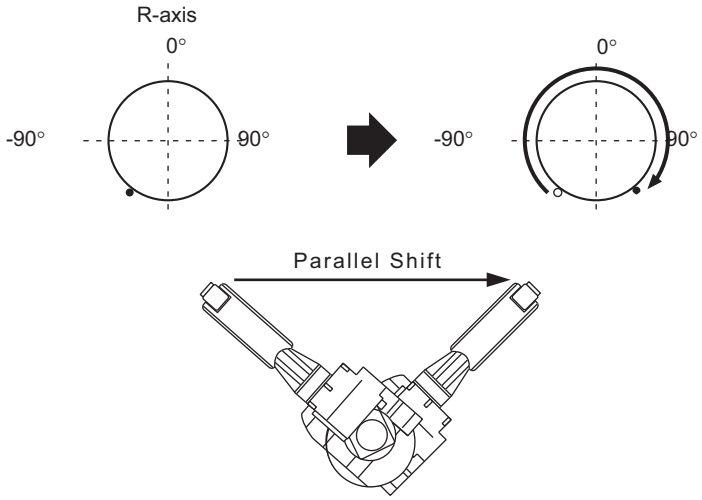
3.5.2 Previous Step Regarded (Minimum R-Axis Movement)

This method keeps the R-axis movement to a minimum when the manipulator moves from the previous step to the next step. Therefore, it can be used for jobs of which the B-axis passes the point 0°.

Since this method minimizes R-axis movement, add some steps to the prescribed path if it is desired to move the R-axis in a wider range.



If a parallel shift is executed in a relative job, the R-axis may have to rotate greatly when the shift amount is large. This method can not be applied to such cases: use the method *chapter 3.5.1 "Previous Step Regarded (Constant B-Axis Sign)"* .



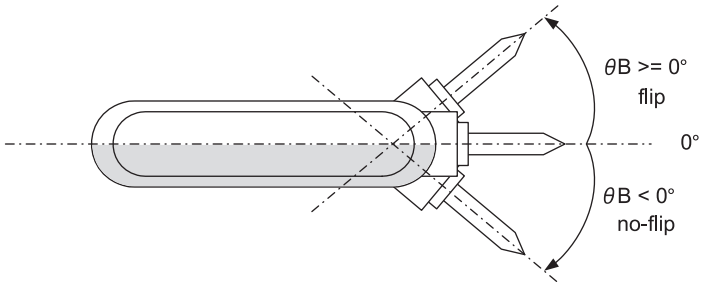
NOTE

- When converting a standard job into a relative job, a standard job should be taught so that the R-axis movement between steps does not exceed 90°. To move the R-axis in an angle measuring more than 90°, add some steps to divide the large angle into smaller ones.
- When converting a relative job into a standard job, the manipulator's current posture is referred. To perform a conversion, posture the manipulator in a similar posture in the first step of the relative job to be converted.

3.5.3 Type Regarded

When a relative job is converted from a standard job, position data of each step is classified into XYZ type position data and type data.
In the type operation method, a movement is operated so that the type is added to the position data.

Since a movement is made for the specified type in any case, this method is effective when performing a job taught offline.

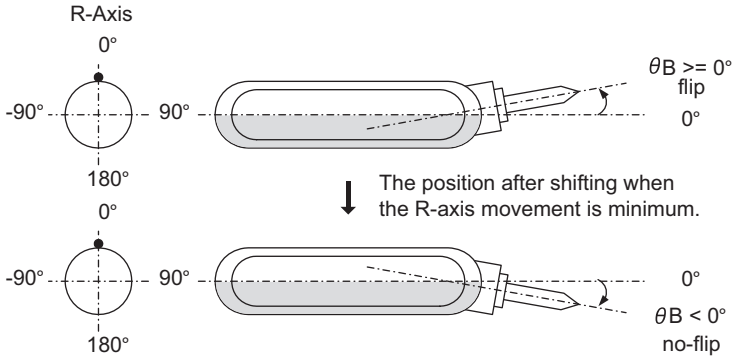


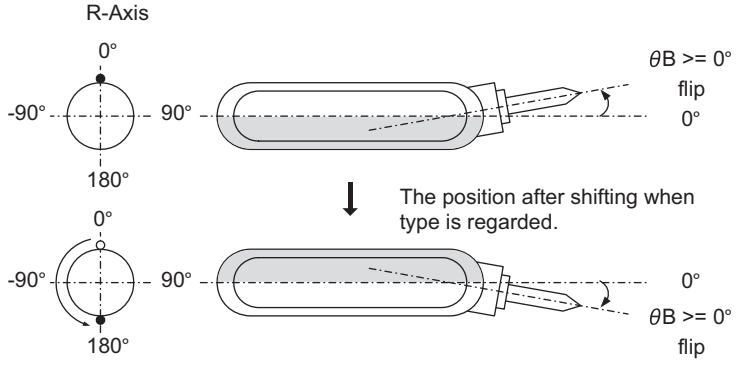
However, in case where a relative job is used as the shifting function for workpiece dislocation, special attention must be paid. If the teaching position is near the pole changing point, the movement may not be made for the specified type when the position is shifted according to the workpiece dislocation.

For example, if a standard job that was taught when the angle of B-axis is close to but more than 0° (θB>=0°) and this position is shifted, the angle of the B-axis may change to exceed 0° (θB<0°).

The movement is made in “flip” before shifting and “no-flip” after shifting.

Since a movement is made for a specified type even when using the type regarded method, the angle of the R-axis is turned 180° in the opposite direction.
Therefore, moving 180° in the opposite direction to the position taught by the R-axis may cause interference with the workpiece. The final tool position and stance are not changed before and after shifting.





4 Interface with an Easy Offline Teaching System

A relative job can be used as an interface with an easy offline teaching system. This chapter describes the necessary information to create a relative job on the easy offline teaching system.

4.1 Job Data Format

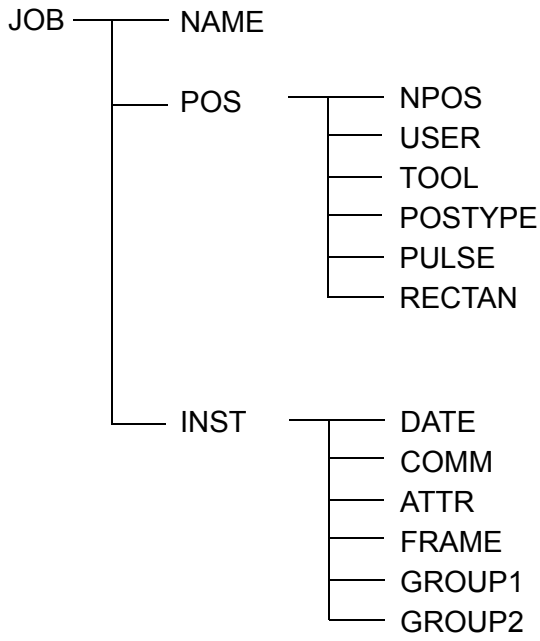
When a relative job is saved in FD/PC card or output by data transmission, the output file contents are as follows.

```
FILE NAME .JBI


/JOB
//NAME <JOB NAME>
//POS
///NPOS  <C>,<BC>,<EC>,<P>,<BP>,<EX>
///USER <N>
///TOOL <N>
///POSTYPE <T>
///RECTAN///RCONF <l>,<m>,<n>,<o>,<p>,<q>,<r1>,<r2>,
                   <r3>,<r4>,<r5>,<r6>,<r7>,<r8>,<r9>,
                   <r10>,<r11>,<r12>,<r13>,<r14>,<r15>,
                   <r16>,<r17>,<r18>
Cxxxxx =X,Y,Z,Rx,Ry,Rz,Re
BCxxxxx=X0,Y0,Z0
ECxxxxx=1,2
          :
          :
//INST
///DATE <YYYY>/<MM>/<DD> <HH>:<TT>
///COMM  <COMMENT CHARACTER LINE>
///ATTR  <ATTRIBUTE 1>,<ATTRIBUTE 2>,···,<ATTRIBUTE 16>
///FRAME <C>
///GROUP1 <m1>,<m2>,<m3>
///GROUP2 <m1>,<m2>,<m3>
NOP
MOVJ  Cxxxxx BCxxxxx ECxxxxx VJ=xxx.x
          :
          :
END
```

A pseudo instruction is distinguished by a single slash (/) at its beginning. Each level of individual instructions are marked with a double slash (//), a triple slash (///), and a fourfold slash (////).

A pseudo instruction related to the job is made as follows.



4.1.1 JOB

Function: Shows that it is a job.

Format: /JOB

4.1.2 NAME

Function: Represents the job name.

Format: //NAME <Name>

<Name>: up to 32 characters

4.1.3 POS

Function: Represents the position data.

Format: //POS

- **NPOS**
  Function: Represents the number of position data items.
  Format: ///NPOS <C>,<BC>,<EC>,<P>,<BP>,<EX>
  <C>: Number of robot axis teaching positions
  <BC>: Number of base axis teaching positions
  <EC>: Number of external (station) axis teaching positions
  <P>: Number of robot axis position variables
  <BP>: Number of base axis position variables
  <EX>: Number of external (station) axis position variables

- **USER**
  Function: Represents the currently selected user coordinate system No.
  Format: ///USER <N>
  <N>: User coordinate system No.(1 to 63)

- **TOOL**
  Function: Represents the currently selected tool No.
  Format: ///TOOL <N>
  <N>: Tool No.(0 to 63)

- **POSTYPE**
  Function: Represents the position data type.
  Format: ///POSTYPE<T>
  <T>: |PULSE||BASE||ROBOT||TOOL||USER||MTOOL|
  <PULSE>: Pulse data
  <BASE>: Cartesian data, base coordinate system
  <ROBOT>: Cartesian data, robot coordinate system
  <TOOL>: Cartesian data, tool coordinate system
  <USER>: Cartesian data, user coordinate system
  <MTOOL>: Cartesian data, master tool coordinate system

■ **PULSE**

Function: Represents that pulse data is defined during and after this pseudo instruction.

Format: ///PULSE

<Pulse data>:<C>|<BC>|<EC>|<P>|<BP>|<EX>

<C>:<Cxxxxx>=<S>,<L>,<U>,<R>,<B>,<T>,<E>

<BC>:<BCxxxxx>=<1>,<2>,<3>

<EC>:<ECxxxxx>=<1>,<2>,<3>,<4>,<5>,<6>

<P>:<Pyyyy>=<S>,<L>,<U>,<R>,<B>,<T>,<E>

<BP>:<BPyyyy>=<1>,<2>,<3>

<EX>:<EXyyyy>=<1>,<2>,<3>,<4>,<5>,<6>

<Cxxxxx>: Robot axis teaching position

<BCxxxxx>: Base axis teaching position

<ECxxxxx>: External (station) axis teaching position

<Pyyyy>: Robot axis position variables

<BPyyyy>: Base axis position variables

<EXyyyy>: External (station) axis position variables

<S>: S-axis pulse data

<L>: L-axis pulse data

<U>: U-axis pulse data

<R>: R-axis pulse data

<B>: B-axis pulse data

<T>: T-axis pulse data

<E>: E-axis pulse data

<1>: 1-axis pulse data

<2>: 2-axis pulse data

<3>: 3-axis pulse data

<4>: 4-axis pulse data

<5>: 5-axis pulse data

<6>: 6-axis pulse data

xxxxx:= A number from 00000 to 09997

yyyy:= A number from 0000 to 9999

■ **RECTAN**

Function: Represents that Cartesian data is defined during and after this pseudo instruction.

Format: ///RECTAN

<Cartesian data>: <C>|<BC>|<P>|<BP>

<C>: <Cxxxxx> = <X>,<Y>,<Z>,<Rx>,<Ry>,<Rz>,<Re>

<BC>: <BCxxxxx> = <1>,<2>,<3>

<P>: <Pyyyy> = <X>,<Y>,<Z>,<Rx>,<Ry>,<Rz>,<Re>

<BP>: <BPyyyy> = <1>,<2>,<3>

<Cxxxxx>: Robot axis teaching position

<BCxxxxx>: Base axis teaching position

<Pyyyy>: Robot axis position variables

<BPyyyy: Base axis position variables

<X>: X-axis Cartesian data

<Y>: Y-axis Cartesian data

<Z>: Z-axis Cartesian data

<Rx>: Rx-axis Cartesian data

<Ry>: Ry-axis Cartesian data

<Rz>: Rz-axis Cartesian data

<Re>: Re-axis Cartesian data

<1>: 1-axis Cartesian data

<2>: 2-axis Cartesian data

<3>: 3-axis Cartesian data

■ **RCONF**

Function: Represents the manipulator type of the Cartesian data defined during and after this pseudo instruction.

Format: ///RCONF <l>,<m>,<n>,<o>,<p>,<q>,<r1>,<r2>,<r3>,
<r4>,<r5>,<r6>,<r7>,<r8>,<r9>,<r10>,<r11>,
<r12>,<r13>,<r14>,<r15>,<r16>,<r17>,<r18>

<l>: 0: Flip, 1: No-flip

<m>: 0: Upper arm, 1: Lower arm

<n>: 0: Front, 1: Rear

<o>: 0: R < 180, 1: R >= 180

<p>: 0: T < 180, 1: T >= 180

<q>: 0: S < 180, 1: S >= 180

<r1> to <r18>: unused

For manipulator type, refer to *chapter 4.3 "Configuration of Position Data"* .

4.1.4 INST

Function : Represents that it is an instruction.

Format : //INST

- **DATE**

Function: Represents the date.

Format: ///DATE <YYYY>/<MM>/<DD> <HH>:<TT>

<YYYY>: Year

<MM>: Month

<DD>: Day

<HH>: Hour

<TT>: Minute

- **COMM**

Function: Represents that it is a job comment.

Format: ///COMM <Comment character line>

<Comment character line>: Up to 32 characters

- **ATTR**

Function: Represents the job attribute.

Format: ///ATTR <Attribute 1>,<Attribute 2>,···,<Attribute16>

<Attribute>: JD|DD|SC|{RO|WO|RW}|RJ

<JD>: Job Destroy

<DD>: Directory Destroy

<SC>: Save Complete

{RO|WO|RW}

<RO>: Writing disabled “Edit-lock” (Read Only)

<WO>: Reading disabled (Write Only)

<RW>: Reading/Writing capable (Read/Write)

<RJ>: Relative job

- **FRAME**

Function: Represents relative job teaching coordinate system.

Format: ///FRAME <C>

<C>: BASE|ROBOT|USER<N>|

<BASE>: Base coordinate system (Cartesian)

<ROBOT>: Robot coordinate system (Cartesian)

<USER>: User coordinate system (Cartesian)

<N>: User coordinate system No. (1 to 63)

■ **GROUP1**
Function: Represents 1st MOVE control group.

Format: ///GROUP1 <m1>,<m2>,<m3>

Add any of the following to<m1>,<m2>, and<m3>.

RB1(robot 1)	ST1(station 1)
RB2(robot 2)	ST2(station 2)
BS1(base 1)	ST3(station 3)
BS2(base 2)	

■ **GROUP2**
Function: Represents 2nd MOVE control group.

Format: ///GROUP2 <m1>,<m2>,<m3>

Add any of the following to<m1>,<m2>, and<m3>.

RB1(robot 1)	ST1(station 1)
RB2(robot 2)	ST2(station 2)
BS1(base 1)	ST3(station 3)
BS2(base 2)	

■ **GROUP3**
Function: Represents 3rd MOVE control group.

Format: ///GROUP3 <m1>,<m2>,<m3>

Add any of the following to<m1>,<m2>, and<m3>.

RB1(robot 1)	ST1(station 1)
RB2(robot 2)	ST2(station 2)
BS1(base 1)	ST3(station 3)
BS2(base 2)	

■ **GROUP4**
Function: Represents 4th MOVE control group.

Format: ///GROUP4 <m1>,<m2>,<m3>

Add any of the following to<m1>,<m2>, and<m3>.

RB1(robot 1)	ST1(station 1)
RB2(robot 2)	ST2(station 2)
BS1(base 1)	ST3(station 3)
BS2(base 2)	

■ **LVARS**

Function: Represents the number of local variables.

Format: ///LVARS <LB>,<LI>,<LD>,<LR>,<LS>,<LP>,<LBP>,<LEX>

<LB>: Number of byte type local variables

<LI>: Number of integer type local variables

<LD>: Number of double-precision type local variables

<LR>: Number of real number type local variables

<LS>: Number of character type local variables

<LP>: Number of robot axis position type local variables

<LBP>: Number of base axis position type local variables

<LEX>: Number of external (station) axis position type local variables

4.2 Relative Job Data Examples

4.2.1 Job for User Coordinate System No. 3, Only for Robot Axis (6-Axis Robot)

File Name : SAMPLE1.JBI

```
/JOB
//NAME SAMPLE1
//POS
///NPOS 5,0,0,0,0,0
///USER 3
///TOOL 0
///POSTYPE USER
///RECTAN
///RCONF  1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
C00000=-2.605,-3.745,-2.487,-67.4787,39.3194,30.6741
C00001=263.527,101.601,209.722,-67.0846,39.4051,31.2718
C00002=-91.821,100.624,96.994,-67.0848,39.4011,31.2722
C00003=-202.120,-147.587,177.161,-67.0849,39.4007,31.2719
C00004=-2.605,-3.745,-2.487,-67.4787,39.3194,30.6741
//INST
///DATE 2008/10/23 12:00
///ATTR SC,RW,RJ
////FRAME USER 3
///GROUP1 RB1
NOP
MOVJ C00000 VJ=50.00
MOVL C00001 V=46.0
MOVL C00002 V=46.0
MOVL C00003 V=46.0
MOVJ C00004 VJ=50.00
END
```

4.2.2 Job for User Coordinate System No. 3, Only for Robot Axis (7-Axis Robot)

File Name : SAMPLE2.JBI

```
/JOB
//NAME SAMPLE2
//POS
///NPOS 5,0,0,0,0,0
///USER 3
///TOOL 0
///POSTYPE USER
///RECTAN
///RCONF 1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
C00000=1.516,-0.379,0.800,168.5608,2.1509,-55.2154,0.1459
C00001=103.781,7.954,128.240,166.1249,-1.1481,-56.7026,0.2007
C00002=91.337,25.556,125.652,166.1254,-1.1503,-56.7031,25.0345
C00003=-11.495,121.527,-16.154,166.1257,-1.1490,-56.7009,14.3574
C00004=7.040,121.978,-8.973,167.9672,1.8172,-55.9887,28.7314
//INST
///DATE 2008/10/23 12:00
///ATTR SC,RW,RJ
////FRAME USER 3
///GROUP1 RB1
NOP
MOVJ C00000 VJ=50.00
MOVL C00001 V=46.0
MOVL C00002 V=46.0
MOVL C00003 V=46.0
MOVJ C00004 VJ=50.00
END
```

4.2.3 Job for Robot Axis (6-Axis) + Base Axis (Base Coordinate System)

File Name : SAMPLE3.JBI

```
/JOB
//NAME SAMPLE3
//POS
///NPOS 3,3,0,0,0,0
///TOOL 0
///POSTYPE BASE
///RECTAN
///RCONF 1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
C00000=820.000,0.000,614.000,180.0000,0.0000,0.0000
C00001=938.810,115.601,612.394,180.0000,0.1362,0.8904
C00002=1030.204,-460.007,454.071,179.9986,0.1269,0.8976
///RCONF 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
BC00000=0.000,0.000
BC00001=114.807,102.795
BC00002=114.807,102.795
//INST
///DATE 2008/10/23 12:00
///ATTR SC,RW,RJ
////FRAME BASE
///GROUP1 RB1,BS1
NOP
MOVJ C00000 BC00000 VJ=25.00
MOVJ C00001 BC00001 VJ=25.00
MOVJ C00002 BC00002 VJ=25.00
END
```

4.2.4 Job for Robot Axis (6-Axis) + Base Axis + Station Axis (Base Coordinate System, Single Job)

File Name:SAMPLE4.JBI

```
/JOB
//NAME SAMPLE3
//POS
///NPOS 2,2,2,0,0,0
///TOOL 0
///POSTYPE BASE
///RECTAN
///RCONF 1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
C00000=820.000,0.000,614.000,180.0000,0.0000,0.0000
C00001=1120.945,125.866,693.276,174.9881,-5.2855,-4.3206
///RCONF 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
BC00000=0.000,0.000
BC00001=168.151,-13.261
///POSTYPE PULSE
///PULSE
EC00000=0
EC00001=15505
//INST
///DATE 2008/10/23 12:00
///ATTR SC,RW,RJ
///FRAME BASE
///GROUP1 RB1,BS1,ST1
NOP
MOVJ C000 BC000 VJ=25.00 +MOVJ EC000 VJ=25.00
MOVJ C001 BC001 VJ=25.00 +MOVJ EC001 VJ=25.00
END
```

4.2.5 Job for Robot Axis (6-Axis) + Base Axis + Station Axis (Base Coordinate System, Coordinate Job)

File Name:SAMPLE5.JBI

```
/JOB
//NAME SAMPLE5
//POS
///NPOS 3,3,3,0,0,0
///TOOL 0
///POSTYPE BASE
///RECTAN
///RCONF 1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
C00000=820.000,0.000,614.000,180.0000,0.0000,0.0000
C00001=310.699,-189.229,878.797,129.5429,3.6290,6.7380
C00002=645.072,351.344,537.037,129.5434,3.6303,6.7375
///RCONF 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
BC00000=0.000,0.000
BC00001=12.215,-11.629
BC00002=8.329,-11.629
///POSTYPE PULSE
///PULSE
EC00000=0
EC00001=-26647
EC00002=-49430
//INST
///DATE 2008/10/23 12:00
///ATTR SC,RW,RJ
////FRAME BASE
///GROUP1 RB1,BS1
///GROUP2 ST1
NOP
MOVJ C00000 BC00000 VJ=25.00  +MOVJ EC00000 VJ=25.00
MOVJ C00001 BC00001 VJ=25.00  +MOVJ EC00001 VJ=25.00
MOVJ C00002 BC00002 VJ=25.00  +MOVJ EC00002 VJ=25.00
END
```

4.2.6 Job for Robot Axis (6-Axis) + Robot Axis (6-Axis) (Base Coordinate System, Coordinate Job)

File Name:SAMPLE6.JBI

```
/JOB
//NAME SAMPLE6
//POS
///NPOS 10,0,0,0,0,0
///TOOL 0
///POSTYPE BASE
///RECTAN
///RCONF 1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
C00000=820.000,0.000,614.000,180.0000,0.0000,0.0000
///TOOL 1
C00001=820.000,0.000,614.000,180.0000,0.0000,0.0000
///TOOL 0
C00002=902.507,80.143,654.194,173.2600,-5.5468,-5.0081
///TOOL 1
C00003=819.450,2.043,613.932,-179.7161,0.2478,-0.2509
///TOOL 0
C00004=896.573,78.209,654.051,172.4974,-6.8732,-5.6059
///TOOL 1
C00005=859.924,20.939,626.880,-177.7488,2.8874,3.4094
///TOOL 0
C00006=892.011,74.458,633.981,171.6846,-7.8341,-8.7050
///TOOL 1
C00007=845.605,0.378,618.082,178.4225,-1.4757,1.3121
///TOOL 0
C00008=889.037,71.428,633.680,171.2305,-8.1815,-9.3640
///TOOL 1
C00009=838.406,-7.953,611.442,178.0552,-1.7633,0.2547
//INST
///DATE 2008/10/23 12:00
///ATTR SC,RW,RJ
////FRAME BASE
///GROUP1 RB1
///GROUP2 RB2
NOP
MOVJ C00000 VJ=50.00  +MOVJ C00001 VJ=50.00
SMOVL C00002 V=46.0  +MOVL C00003
SMOVL C00004 V=46.0  +MOVL C00005
MOVL C00006 V=46.0  +MOVL C00007 V=11.0
MOVL C00008 V=46.0  +MOVL C00009 V=22.0
END
```

4.2.7 Job for Robot Axis (7-Axis) + Robot Axis (6-Axis) (Base Coordinate System, Coordinate Job)

File Name:SAMPLE7.JBI

```
/JOB
//NAME SAMPLE7
//POS
///NPOS 10,0,0,0,0,0
///TOOL 0
///POSTYPE BASE
///RECTAN
///RCONF 1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
C00000=824.237,246.386,520.293,-179.5792,0.4824,0.7807,0.4685
///TOOL 1
C00001=820.000,0.000,614.001,-72.5917,0.7106,89.1770,0.0000
///TOOL 0
C00002=793.954,422.714,861.754,171.6558,-30.9222,-1.2583,4.5765
///TOOL 1
C00003=714.876,195.197,968.938,-104.5394,7.2542,82.4590,0.0000
///TOOL 0
C00004=778.995,426.244,865.671,171.3396,-31.1937,-1.4285,3.4157
///TOOL 1
C00005=442.699,4.409,408.710,-111.1509,7.0959,38.4715,0.0000
///TOOL 0
C00006=734.683,393.846,851.343,169.7611,-34.8869,-2.5952,1.5340
///TOOL 1
C00007=509.009,35.286,412.152,-99.7284,9.2743,49.9918,0.0000
///TOOL 0
C00008=944.084,48.552,706.077,158.4685,-15.9402,-6.9278,0.0000
///TOOL 1
C00009=635.334,-161.233,614.005,-74.8003,1.5534,89.1139,0.0000
//INST
///DATE 2008/10/23 12:00
///ATTR SC,RW,RJ
////FRAME BASE
///GROUP1 RB1
///GROUP2 RB2
NOP
MOVJ C00000 VJ=50.00  +MOVJ C00001 VJ=50.00
SMOVL C00002 V=46.0  +MOVL C00003
SMOVL C00004 V=46.0  +MOVL C00005
MOVL C00006 V=46.0  +MOVL C00007 V=11.0
MOVL C00008 V=46.0  +MOVL C00009 V=22.0
END
```

4.3 Configuration of Position Data

The configuration of the position data for each axis in each coordinate system is as follows.

4.3.1 Position Data of Each Axis

4.3.1.1 Robot Axis

R1 = X, Y, Z, Rx, Ry, Rz + type

X-, Y- and Z-axes represent the coordinate position of the control point based on a specified coordinate system.

Rx-, Ry- and Rz-axes represent the coordinate posture of the control point based on a specified coordinate system with roll, pitch or yaw.

By rotating the specified coordinate system in the order of Rz, Ry and Rx, it takes the coordinate system posture of the control point.

4.3.1.2 Station Axis

S1 = W1, W2

The position of a station axis is represented by a pulse number.

4.3.1.3 Base Axis

B1 = X0, Y0, Z0

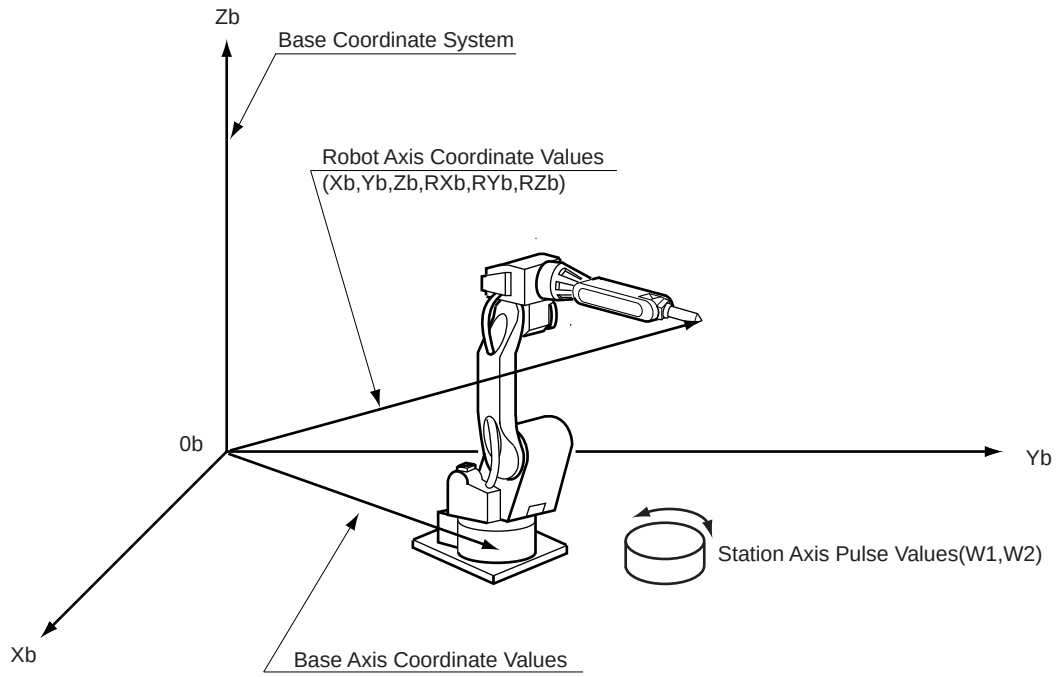
The position of a base axis is represented as the distance of the axis from the origin of the coordinate system of teaching. (For the used axes only.)

- For the base coordinate system, the distance from the origin of the base coordinate system.
- For the robot coordinate system, the distance from the origin of the base coordinate system.
- For the user coordinate system, the distance from the origin of the user coordinate system.

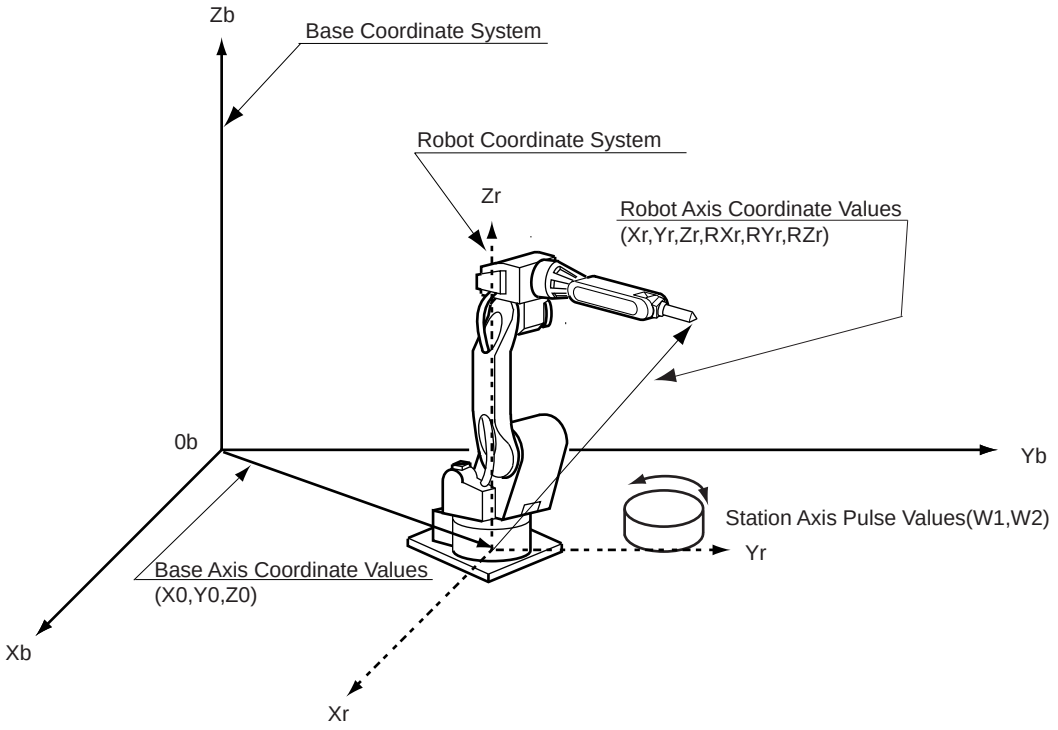
4.3.2 Position Data of Each Coordinate System

The position data of a robot axis, base axis, and station axis in each coordinate system is as follows.

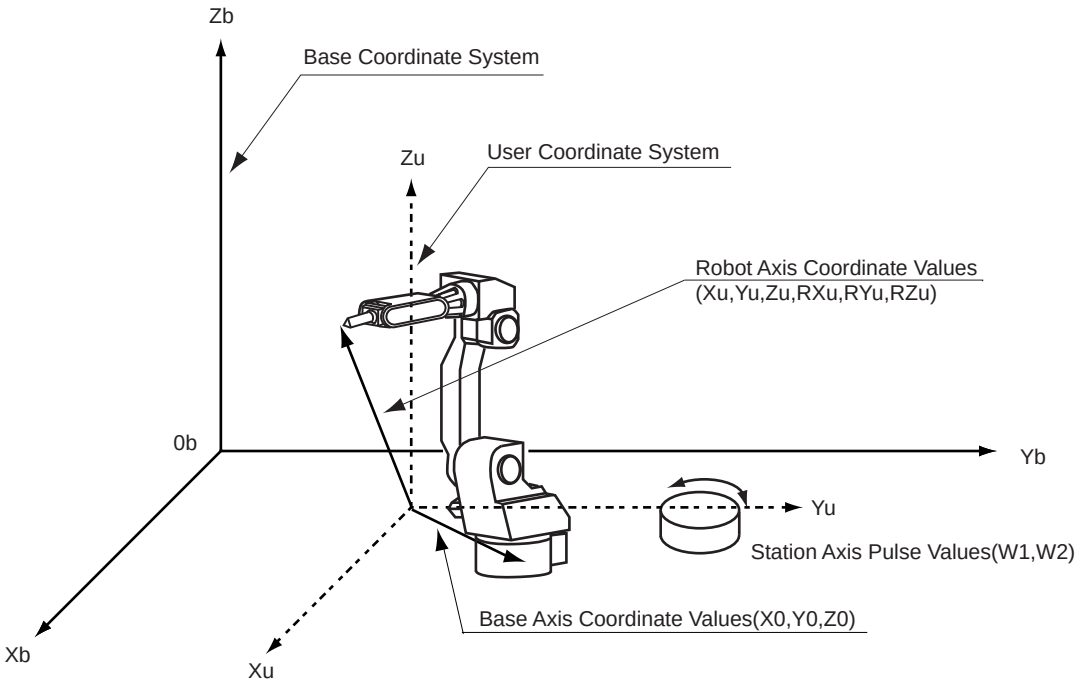
4.3.2.1 Base Coordinate System



4.3.2.2 Robot Coordinate System



4.3.2.3 User Coordinate System

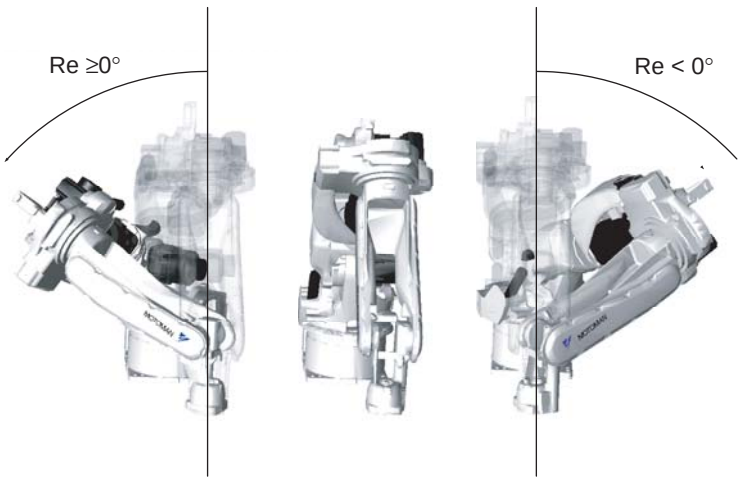


NOTE

The user coordinate system of 7-axis manipulator takes X, Y, Z, Rx, Ry, Rz, Re+ type.

Re is an element which shows the posture of 7-axis manipulator and it would not change by the specified coordinate system.

The definition of Re is as follows.



4.4 Manipulator Type

To describe robot axis position data in the XYZ type, several postures can be taken due to the manipulator mechanism when the manipulator is moved to the described position.

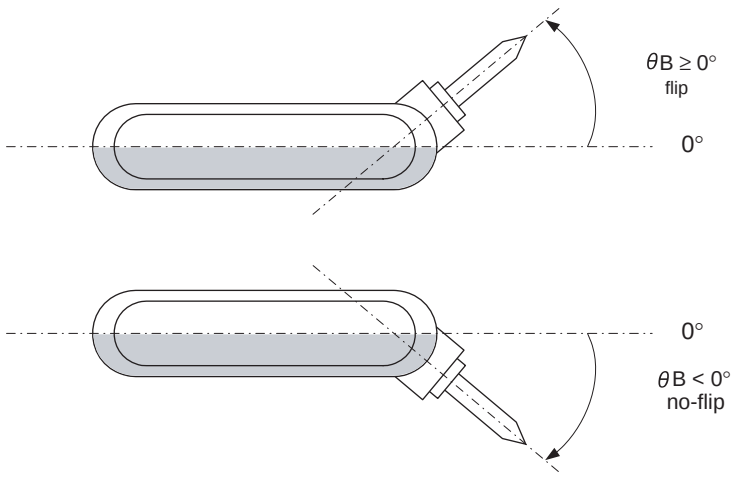
These postures have different pulses at each axis while sharing the same coordinate system at the control point.

In consequence, apart from the coordinate value, the data to specify the manipulator's poseur should also be specified since the posture of the manipulator cannot be determined merely with the coordinate value. This is called "TYPE".

The TYPE varies depending on the type of the manipulator.

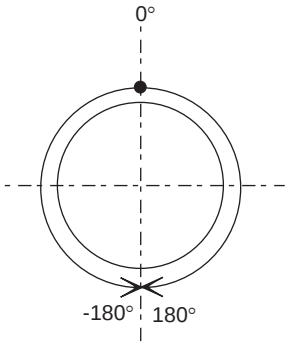
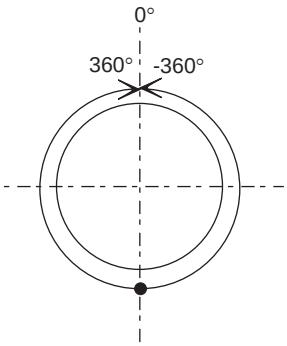
4.4.1 Flip/No-flip

As shown in the following figure, when B is in (+) direction (θB>=0°), it is defined as "flip" and it is "no-flip" when B is in (-) direction (θB<0°).



4.4.2 R-Axis Angle

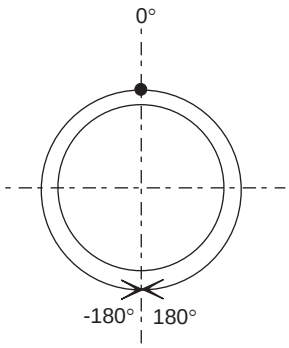
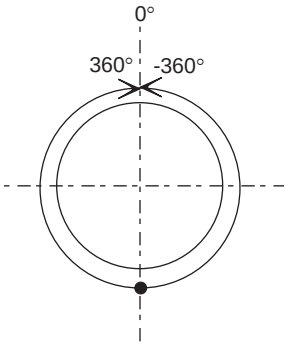
Specify whether the R-axis angle is within ±180°or exceeds ±180°.

R<180°	R>=180°
	
-180°<θR<=180°	180°<θR, θR<=-180°

Note: θR is the angle when the R-axis zero-point position is assumed to be at 0°.

4.4.3 T-Axis Angle

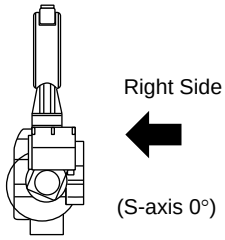
Specify whether the T-axis angle is within ±180°or exceeds ±180°.

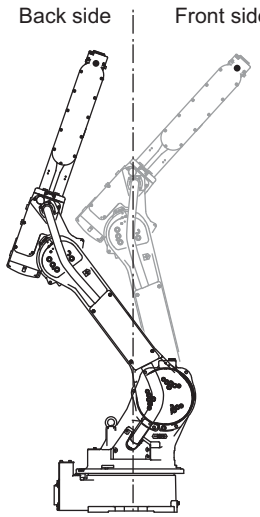
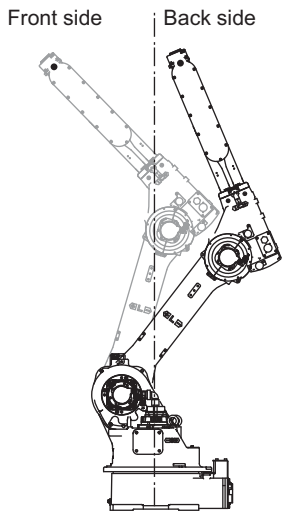
T<180°	T>=180°
	
-180°<θT<=180°	180°<θT, θT<=-180°

Note: θT is the angle when the T-axis zero-point position is assumed to be at 0°.

4.4.4 Front/Back

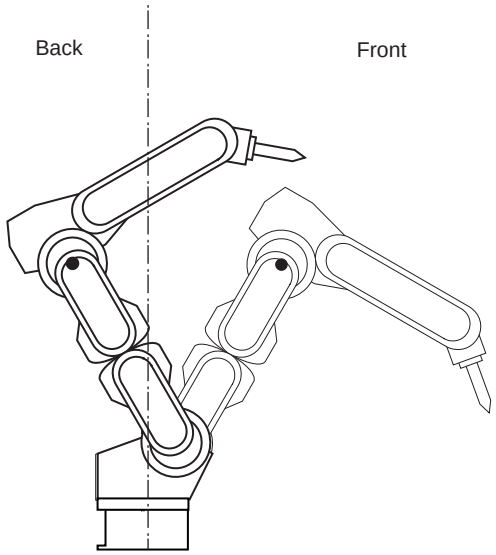
Viewing the L-axis and U-axis from the right side, specify the location of the **B**-axis rotation center either right or left side of the S-axis rotation center.
If its center is at the right side of the S-axis rotation center, it is defined as “front” and left side is “back”.



S-axis 0°	S-axis 180°
	

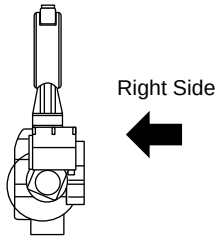
The above figures show the S-axis angle at 0°and 180°. Specification must be performed by always viewing the L-axis and U-axis from the right side.

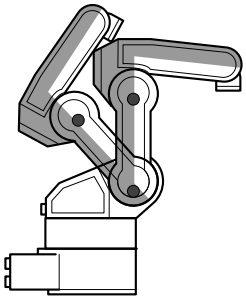
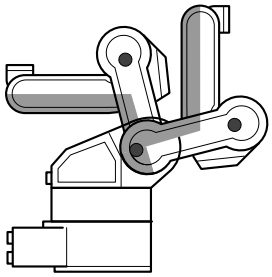
When it is 7-axis manipulator, viewing the L-axis and U-axis from the right side, specify the side of the U-axis rotation center either right or left side of the S-axis rotation.
If its center is at the right side of the S-axis rotation center, it is defined as “front” and left side is “back”.



4.4.5 Upper/Lower Arm (Type Comprised of L-Axis and U-Axis

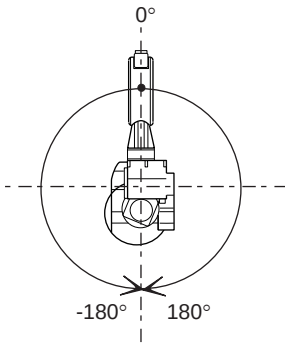
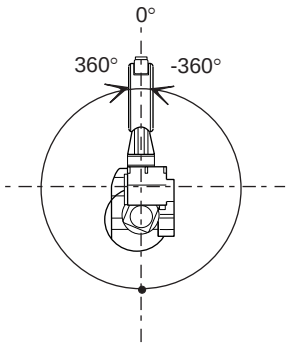
Specify the type comprised of the L-axis and the U-axis when the L-axis and the U-axis are viewed from the right side.



Upper arm	Lower arm
	

4.4.6 S-Axis Angle

Specify whether the S-axis angle is within ±180°or exceeds ±180°.

S<180°	S>=180°
	
-180°<θs<=180°	180°<θS, θS<=-180°

Note: θs is the angle when the S-axis zero-point position is assumed to be at 0°.

This specification is required for the manipulators of which S-axis operating range is greater than ±180°.

5 Alarm and Error Message List

5.1 Alarm Messages

Alarm Number	Message	Cause	Remedy
4499	UNDEFINED POSITION VARIABLE [Decimal Data]	Undefined position-type variable were used.	Resister the position file (VARIABLE)
4500	UNDEFINED USER FRAME [Decimal Data]	Undefined user coordinates were used.	Define the user coordinate.
4512	THREE STEPS SAME LINE [No data display]	The three points for creating the user coordinates lie on the same line.	Teach again so that the three points do not lie on the same line.

5.2 Error Messages

Error Number	Message	Contents
300	Undefined user frame	The user coordinate system to be used at conversion is not registered.
2470	Wrong JOB type	Setting of coordinate system for a standard job is not possible.
2480	Wrong JOB coordinates setting	Coordinate systems other than the user coordinate system can not be changed.
2500	Cannot convert the JOB.	A job with only a station axis and without group axis can not be converted into a relative job.

6 Instruction List

< >shows number or character data. When there are more than one additional items in one section, choose one.

MFRAME	Function	Creates a user coordinate system using the position data of the given three points as the definition point. Format: MFRAME UF#(xx) <Data 1><Data 2><Data 3>		
	Additional items	UF#(<User coordinate system No.>)		1 to 63
		Data 1	Definition point ORG position data	
		Data 2	Definition point XX position data	
		Data 3	Definition point XY position data	
		IF statement		
	Example	MFRAME UF#(1) P001 P002 P003		
CALL	Function	Calls for a specified job and executes it. If a user coordinate system number is specified, the job is executed using the coordinate system indicated by that number when calling a relative job.		
	Additional items	JOB<Job name> IG#(<Input group No.>) B<Variable No.>		
		UF#(<User coordinate system No.>)		1 to 63
		IF statement		
	Example	CALL JOB:TEST-1 CALL JOB:TEST-1 UF#(2) CALL IG#(02) (Job call according to input signal pattern. In this case, job 0 can not be called.)		
JUMP	Function	Jumps to the specified job or label. If a user coordinate system number is specified, the job is executed using the coordinate system indicated by that number when jumping to a relative job.		
	Additional items	JOB:<Job name> IG#(<Input group No.>) B<Variable No.> <Label No.>		
		UF#(<User coordinate system No.>)		1 to 63
		IF statement		
	Example	JUMP JOB:TEST1 IF IN#(14)=OFF		

YRC1000micro OPTIONS
INSTRUCTIONS

FOR RELATIVE JOB FUNCTION

HEAD OFFICE
2-1 Kurosakishiroishi, Yahatanishi-ku, Kitakyushu 806-0004, Japan
Phone +81-93-645-7703 Fax +81-93-645-7802

YASKAWA America Inc. (Motoman Robotics Division)
100 Automation Way, Miamisburg, OH 45342, U.S.A.
Phone +1-937-847-6200 Fax +1-937-847-6277

YASKAWA Europe GmbH（Robotics Divsion ）
Yaskawastrasse 1, 85391 Allershausen, Germany
Phone +49-8166-90-100 Fax +49-8166-90-103

YASKAWA Nordic AB
Verkstadsgatan 2, Box 504 ,SE-385 25 Torsas, Sweden
Phone +46-480-417-800 Fax +46-486-414-10

YASKAWA Electric (China) Co., Ltd.
22F, One Corporate Avenue, No.222, Hubin Road, Huangpu District, Shanghai 200021, China
Phone +86-21-5385-2200 Fax ＋86-21-5385-3299

YASKAWA SHOUGANG ROBOT Co. Ltd.
No7 Yongchang North Road, Beijing E&T Development Area, China 100176
Phone +86-10-6788-2858 Fax +86-10-6788-2878

YASKAWA India Private Ltd. (Robotics Division)
#426, Udyog Vihar, Phase- IV, Gurgaon, Haryana, India
Phone +91-124-475-8500 Fax +91-124-475-8542

YASKAWA Electric Korea Corporation
35F, Three IFC, 10 Gukjegeumyung-ro, Yeongdeungpo-gu, Seoul, Korea 07326
Phone +82-2-784-7844 Fax +82-2-784-8495

YASKAWA Electric Taiwan Corporation
12F, No.207, Sec. 3, Beishin Rd., Shindian District, New Taipei City 23143, Taiwan
Phone +886-2-8913-1333 Fax +886-2-8913-1513

YASKAWA Electric (Singapore) PTE Ltd.
151 Lorong Chuan, #04-02A, New Tech Park, Singapore 556741
Phone +65-6282-3003 Fax +65-6289-3003

YASKAWA Electric (Thailand) Co., Ltd.
59,1st-5th Floor, Flourish Building, Soi Ratchadapisek 18,Ratchadapisek Road,
Huaykwang, Bangkok 10310, THAILAND
Phone +66-2-017-0099 Fax +66-2-017-0199

PT. YASKAWA Electric Indonesia
Secure Building-Gedung B Lantai Dasar & Lantai 1 JI. Raya Protokol Halim Perdanakusuma,
Jakarta 13610, Indonesia
Phone +62-21-2982-6470 Fax +62-21-2982-6741

YASKAWA

YASKAWA ELECTRIC CORPORATION

© Printed in Japan July 2017 17-07

MANUAL NO.
HW1484476