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

MOTOMAN INSTRUCTIONS

MOTOMAN-DX100 INSTRUCTIONS
DX100 INSTRUCTIONS
DX100 OPERATOR’S MANUAL
DX100 MAINTENANCE MANUAL

The DX100 operator’s manuals above correspond to specific usage.
Be sure to use the appropriate manual.
MANDATORY

• This manual explains the relative job function of the DX100. Read this manual carefully and be sure to understand its contents before handling the DX100.

• General items related to safety are listed in Section 1: Safety of the DX100 Instructions. To ensure correct and safe operation, carefully read the DX100 Instructions before reading this manual.

CAUTION

• Some drawings in this manual are shown with the protective covers or shields removed for clarity. Be sure all covers and shields are replaced before operating this product.

• 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.

• YASKAWA is not responsible for incidents arising from unauthorized modification of its products. Unauthorized modification voids your product’s warranty.
NOTES FOR SAFE OPERATION

Read this manual carefully before installation, operation, maintenance, or inspection of the DX100.

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

⚠️ WARNING
Indicates a potentially hazardous situation which, if not avoided, could result in death or serious injury to personnel.

⚠️ CAUTION
Indicates a potentially hazardous situation which, if not avoided, could result in minor or moderate injury to personnel and damage to equipment. It may also be used to alert against unsafe practices.

⚠️ MANDATORY
Always be sure to follow explicitly the items listed under this heading.

.subscription

🚫 PROHIBITED
Must never be performed.

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

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

• Before operating the manipulator, check that servo power is turned off when the emergency stop buttons on the front door of the DX 100 and programing pendant are pressed. When the servo power is turned off, the SERVO ON LED on the programing pendant is turned off.

Injury or damage to machinery may result if the emergency stop circuit cannot stop the manipulator during an emergency. The manipulator should not be used if the emergency stop buttons do not function.

**Fig. : Emergency Stop Button**

• Once the emergency stop button is released, clear the cell of all items which could interfere with the operation of the manipulator. Then turn the servo power ON.

Injury may result from unintentional or unexpected manipulator motion.

**Fig. : Release of EM**

• Observe the following precautions when performing teaching operations within the P-point maximum envelope of the manipulator:
  – View the manipulator from the front whenever possible.
  – Always follow the predetermined operating procedure.
  – Ensure that you have a safe place to retreat in case of emergency.

Improper or unintended manipulator operation may result in injury.

• Confirm that no person is present in the P-point maximum envelope of the manipulator and that you are in a safe location before:
  – Turning on the power for the DX100.
  – Moving the manipulator with the programming pendant.
  – Running the system in the check mode.
  – Performing automatic operations.

Injury may result if anyone enters the working envelope of the manipulator during operation. Always press an emergency stop button immediately if there are problems.

The emergency stop button is located on the right of the front door of the DX 100 and programing pendant.
CAUTION

- Perform the following inspection procedures prior to conducting manipulator teaching. If problems are found, repair them immediately, and be sure that all other necessary processing has been performed.
  - Check for problems in manipulator movement.
  - Check for damage to insulation and sheathing of external wires.
- Always return the programming pendant to the hook on the DX100 cabinet after use.
- The programming pendant can be damaged if it is left in the manipulator’s work area, on the floor, or near fixtures.
- Read and understand the Explanation of the Warning Labels in the DX100 Instructions before operating the manipulator.

Definition of Terms Used Often in This Manual

The MOTOMAN manipulator is the YASKAWA industrial robot product. The MOTOMAN usually consists of the controller, the programming pendant, and supply cables.

In this manual, the equipment is designated as follows.

<table>
<thead>
<tr>
<th>Equipment</th>
<th>Manual Designation</th>
</tr>
</thead>
<tbody>
<tr>
<td>DX100 Controller</td>
<td>DX100</td>
</tr>
<tr>
<td>DX100 Programming Pendant</td>
<td>Programming Pendant</td>
</tr>
<tr>
<td>Cable between the manipulator and the controller</td>
<td>Manipulator cable</td>
</tr>
</tbody>
</table>
Descriptions of the programming pendant and playback panel keys, buttons, and displays are shown as follows:

<table>
<thead>
<tr>
<th>Equipment</th>
<th>Manual Designation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Programming Pendant</td>
<td></td>
</tr>
<tr>
<td>Character Keys</td>
<td>The keys which have characters printed on them are denoted with [ ]. ex. [ENTER]</td>
</tr>
<tr>
<td>Symbol Keys</td>
<td>The keys which have a symbol printed on them are not denoted with [ ] but depicted with a small picture. ex. page key</td>
</tr>
<tr>
<td>Axis Keys</td>
<td>“Axis Keys” and “Numeric Keys” are generic names for the keys for axis operation and number input.</td>
</tr>
<tr>
<td>Number Keys</td>
<td></td>
</tr>
<tr>
<td>Keys pressed simultaneously</td>
<td>When two keys are to be pressed simultaneously, the keys are shown with a “+” sign between them, ex. [SHIFT]+[COORD]</td>
</tr>
<tr>
<td>Displays</td>
<td>The menu displayed in the programming pendant is denoted with { }. ex. {JOB}</td>
</tr>
</tbody>
</table>

**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 the SELECT key is pressed, or that the item is directly selected by touching the screen.
# Table of 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 .......... 3-4
     3.2.1 Verifying Coordinate System .............................. 3-4
     3.2.2 Verifying Command Positions ............................ 3-5
   3.3 Instructions Related to Relative Job .......................... 3-6
     3.3.1 CALL/JUMP .................................................. 3-6
     3.3.2 MFRAME ....................................................... 3-7
     3.3.3 Registering an Instruction .................................. 3-8
   3.4 Editing Relative Jobs ............................................... 3-10
     3.4.1 Blink Indication of Move Instruction on Job Content Window .............. 3-10
     3.4.2 Addition and Modification of Steps ......................... 3-10
     3.4.3 Cut & Paste Function ....................................... 3-10
   3.5 Relative Job Operation Method ................................. 3-11
     3.5.1 Previous Step Regarded (Constant B-axis Sign) .......... 3-12
     3.5.2 Previous Step Regarded (Minimum R-axis Movement) ............... 3-14
     3.5.3 Type Regarded ............................................. 3-15

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

4.2.1 Job for User Coordinate System No. 3, Only for Robot Axis (6-Axis Robot) ... 4-12
4.2.2 Job for User Coordinate System No. 3, Only for Robot Axis (7-Axis Robot) ... 4-13
4.2.3 Job for Robot Axis (6-axis) + Base Axis (Base Coordinate System) ............ 4-14
4.2.4 Job for Robot Axis (6-Axis) + Base Axis + Station Axis (Base Coordinate System, Single Job) ... 4-15
4.2.5 Job for Robot Axis (6-Axis) + Base Axis + Station Axis (Base Coordinate System, Coordinate Job) ... 4-16
4.2.6 Job for Robot Axis (6-Axis) + Robot Axis (6-Axis) (Base Coordinate System, Coordinate Job) ... 4-17
4.2.7 Job for Robot Axis (7-Axis) + Robot Axis (6-Axis) (Base Coordinate System, Coordinate Job) ... 4-19

4.3 Configuration of Position Data

4.3.1 Position Data of Each Axis
4.3.1.1 Robot Axis
4.3.1.2 Station Axis
4.3.1.3 Base Axis

4.3.2 Position Data of Each Coordinate System
4.3.2.1 Base Coordinate System
4.3.2.2 Robot Coordinate System
4.3.2.3 User Coordinate System

4.4 Manipulator Type

4.4.1 Flip/No-flip
4.4.2 R-axis Angle
4.4.3 T-axis Angle
4.4.4 Front/Back
4.4.5 Upper/Lower Arm (Type Comprised of an L-axis and U-axis)
4.4.6 S-axis Angle

5 Alarm and Error Message List

5.1 Alarm Messages
5.2 Error Messages

6 Instruction List
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.

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.
### 1. Relative Job

#### 1.2 Relative Job Shift Functions

**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.

![Diagram showing position data of a, b, c, with external computer, vision controller, etc., and user coordinate system creation process.]

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

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.
2 Examples of Use of Relative Jobs
2.1 Shift Function to Offset Workpiece Position Error

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

   ```
   NOP
   LOADV P000
   LOADV P001
   LOADV P002
   MFRAME UF#(1) P000 P001 P002
   MOVJ VJ=50.0
   CALL JOB:RELATIVE-1
   END
   ```

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

   User Coordinate System before Modification
   User Coordinate System after Modification

   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 Examples of Use of Relative Jobs

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

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 step 4 and step 5 using other user coordinate system numbers.

NOTE

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.
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.

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

Excutes job conversion.

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

- 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.

![Command Position Window](image)

> "OV" appears when the position is beyond the P-point max. envelope of the manipulator
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.

MFRAME UF#(1) P000 P001 P002

User coordinate system to be created.

Position variables and position data of user coordinate system definition points are stored.

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 underbar.
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.

   ![Input Buffer Screenshot]

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.

     ![Input Line Screenshot]

     – 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>
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

<table>
<thead>
<tr>
<th>Coordinate system of teaching</th>
<th>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.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Teaching tool</td>
<td>Coordinates of operation positions are registered by the tool used when the axis is operated.</td>
</tr>
</tbody>
</table>

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 three methods can be used to designate the motion.

<table>
<thead>
<tr>
<th>Method Description</th>
<th>Effective for</th>
</tr>
</thead>
<tbody>
<tr>
<td>Previous Step Regarded (Constant B-axis sign)</td>
<td>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.)</td>
</tr>
<tr>
<td>Previous Step Regarded (Minimum R-axis movement)</td>
<td>A job of which the B-axis passes the point 0°.</td>
</tr>
<tr>
<td>Type Regarded</td>
<td>A job created in offline teaching is to be performed.</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Settings</th>
<th>Initial value</th>
</tr>
</thead>
<tbody>
<tr>
<td>S2C430</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>0 : Previous step regarded (constant B-axis sign)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 : Type regarded</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2 : Previous step regarded (Minimum R-axis movement)</td>
<td></td>
</tr>
</tbody>
</table>
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 cannot be applied to such cases: use the method "3.5.1 Previous Step Regarded (Constant B-axis Sign)."

- 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.

The position after shifting when the R-axis movement is minimum.

The position after shifting when type is regarded.
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.

```
JOB
  NAME
  POS
  NPOS
  USER
  TOOL
  POSTYPE
  PULSE
  RECTAN
  INST
  DATE
  COMM
  ATTR
  FRAME
  GROUP1
  GROUP2
```
4.1 Interface with an Easy Offline Teaching System

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.(0 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>:<Pyyyyy>=<S>,<L>,<U>,<R>,<B>,<T>,<E>
<BP>:<BPyyyyy>=<1>,<2>,<3>
<EX>:<EXyyyyy>=<1>,<2>,<3>,<4>,<5>,<6>
<Cxxxxx>: Robot axis teaching position
<BCxxxxx>: Base axis teaching position
<ECxxxxx>: External (station) axis teaching position
<Pyyyyy>: Robot axis position variables
<BPyyyyy>: Base axis position variables
<EXyyyyy>: 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
yyyyy:= A number from 0000 to 9999
4 Interface with an Easy Offline Teaching System
4.1 Job Data Format

- **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” at page 4-21.
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)
RB3(robot 3) ST3(station 3)
RB4(robot 4) ST4(station 4)
RB5(robot 5) ST5(station 5)
RB6(robot 6) ST6(station 6)
RB7(robot 7) ST7(station 7)
RB8(robot 8) ST8(station 8)
BS1(base 1) ST9(station 9)
BS2(base 2) ST10(station 10)
BS3(base 3) ST11(station 11)
BS4(base 4) ST12(station 12)
BS5(base 5) ST13(station 13)
BS6(base 6) ST14(station 14)
BS7(base 7) ST15(station 15)
BS8(base 8) ST16(station 16)

ST17(station 17)
ST18(station 18)
ST19(station 19)
ST20(station 20)
ST21(station 21)
ST22(station 22)
ST23(station 23)
ST24(station 24)
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)
- RB3(robot 3)  ST3(station 3)
- RB4(robot 4)  ST4(station 4)
- RB5(robot 5)  ST5(station 5)
- RB6(robot 6)  ST6(station 6)
- RB7(robot 7)  ST7(station 7)
- RB8(robot 8)  ST8(station 8)
- BS1(base 1)   ST9(station 9)
- BS2(base 2)   ST10(station 10)
- BS3(base 3)   ST11(station 11)
- BS4(base 4)   ST12(station 12)
- BS5(base 5)   ST13(station 13)
- BS6(base 6)   ST14(station 14)
- BS7(base 7)   ST15(station 15)
- BS8(base 8)   ST16(station 16)
- ST17(station 17)
- ST18(station 18)
- ST19(station 19)
- ST20(station 20)
- ST21(station 21)
- ST22(station 22)
- ST23(station 23)
- ST24(station 24)
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) |
| RB3(robot 3)   | ST3(station 3) |
| RB4(robot 4)   | ST4(station 4) |
| RB5(robot 5)   | ST5(station 5) |
| RB6(robot 6)   | ST6(station 6) |
| RB7(robot 7)   | ST7(station 7) |
| RB8(robot 8)   | ST8(station 8) |
| BS1(base 1)    | ST9(station 9) |
| BS2(base 2)    | ST10(station 10) |
| BS3(base 3)    | ST11(station 11) |
| BS4(base 4)    | ST12(station 12) |
| BS5(base 5)    | ST13(station 13) |
| BS6(base 6)    | ST14(station 14) |
| BS7(base 7)    | ST15(station 15) |
| BS8(base 8)    | ST16(station 16) |
|               | ST17(station 17) |
|               | ST18(station 18) |
|               | ST19(station 19) |
|               | ST20(station 20) |
|               | ST21(station 21) |
|               | ST22(station 22) |
|               | ST23(station 23) |
|               | ST24(station 24) |
**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)
- RB3(robot 3)  ST3(station 3)
- RB4(robot 4)  ST4(station 4)
- RB5(robot 5)  ST5(station 5)
- RB6(robot 6)  ST6(station 6)
- RB7(robot 7)  ST7(station 7)
- RB8(robot 8)  ST8(station 8)
- BS1(base 1)  ST9(station 9)
- BS2(base 2)  ST10(station 10)
- BS3(base 3)  ST11(station 11)
- BS4(base 4)  ST12(station 12)
- BS5(base 5)  ST13(station 13)
- BS6(base 6)  ST14(station 14)
- BS7(base 7)  ST15(station 15)
- BS8(base 8)  ST16(station 16)
  - ST17(station 17)
  - ST18(station 18)
  - ST19(station 19)
  - ST20(station 20)
  - ST21(station 21)
  - ST22(station 22)
  - ST23(station 23)
  - ST24(station 24)
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
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 5,0,0,0,0,0,0
///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=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 1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
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
C00000=820.000,0.000,614.000,180.000,0.000,0.000
C00001=1120.945,125.866,693.276,174.988,5.285,5,-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
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=820.000,0.000,614.000,180.000,0.000,0.000
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 1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
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
4 Interface with an Easy Offline Teaching System

4.2 Relative Job Data Examples

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 10,0,0,0,0,0,0
///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
4 Interface with an Easy Offline Teaching System
4.2 Relative Job Data Examples

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.

Re $\geq 0^\circ$

Re $< 0^\circ$
4 Interface with an Easy Offline Teaching System
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 $\pm 180^\circ$ or exceeds $\pm 180^\circ$.

<table>
<thead>
<tr>
<th>$\theta_R$</th>
<th>$\pm 180^\circ$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R &lt; 180^\circ$</td>
<td>$R \geq 180^\circ$</td>
</tr>
</tbody>
</table>

Note: $\theta_R$ is the angle when the R-axis zero-point position is assumed to be at $0^\circ$.

4.4.3 T-axis Angle

Specify whether the T-axis angle is within $\pm 180^\circ$ or exceeds $\pm 180^\circ$.

<table>
<thead>
<tr>
<th>$\theta_T$</th>
<th>$\pm 180^\circ$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T &lt; 180^\circ$</td>
<td>$T \geq 180^\circ$</td>
</tr>
</tbody>
</table>

Note: $\theta_T$ is the angle when the T-axis zero-point position is assumed to be at $0^\circ$. 
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”.

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”.

Back

Front
4.4.5 Upper/Lower Arm (Type Comprised of an 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.

<table>
<thead>
<tr>
<th>Upper arm</th>
<th>Lower arm</th>
</tr>
</thead>
<tbody>
<tr>
<td>S&lt;180°</td>
<td>S&gt;=180°</td>
</tr>
</tbody>
</table>

4.4.6 S-axis Angle

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

<table>
<thead>
<tr>
<th>S&lt;180°</th>
<th>S&gt;=180°</th>
</tr>
</thead>
<tbody>
<tr>
<td>0°</td>
<td>0°</td>
</tr>
<tr>
<td>-180°</td>
<td>180°</td>
</tr>
<tr>
<td>-180°&lt;θs&lt;=180°</td>
<td>180°&lt;θS, θS&lt;=-180°</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Alarm Number</th>
<th>Message Description</th>
<th>Cause</th>
<th>Remedy</th>
</tr>
</thead>
<tbody>
<tr>
<td>4500</td>
<td>UNDEFINED USER FRAME [Decimal Data]</td>
<td>Undefined user coordinates were used.</td>
<td>Define the user coordinate.</td>
</tr>
<tr>
<td>4509</td>
<td>MFRAME ERROR [Decimal Data]</td>
<td>Impossible to create user coordinates.</td>
<td>Register the position file (variables).</td>
</tr>
<tr>
<td>4512</td>
<td>THREE STEPS SAME LINE [No data display]</td>
<td>The three points for creating the user coordinates lie on the same line.</td>
<td>Teach again so that the three points do not lie on the same line.</td>
</tr>
</tbody>
</table>

#### 5.2 Error Messages

<table>
<thead>
<tr>
<th>Error Number</th>
<th>Message Description</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>300</td>
<td>Undefined user frame</td>
<td>The user coordinate system to be used at conversion is not registered.</td>
</tr>
<tr>
<td>2470</td>
<td>Wrong JOB type</td>
<td>Setting of coordinate system for a standard job is not possible.</td>
</tr>
<tr>
<td>2480</td>
<td>Wrong JOB coordinates setting</td>
<td>Coordinate systems other than the user coordinate system can not be changed.</td>
</tr>
<tr>
<td>2500</td>
<td>Cannot convert the JOB.</td>
<td>A job with only a station axis and without group axis can not be converted into a relative job.</td>
</tr>
</tbody>
</table>
< > shows number or character data. When there are more than one additional items in one section, choose one.

<table>
<thead>
<tr>
<th>Instruction</th>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MFRAME</td>
<td>Function</td>
<td>Creates a user coordinate system using the position data of the given three points as the definition point. Format: MFRAME UF#(xx) &lt;Data 1&gt;&lt;Data 2&gt;&lt;Data 3&gt;</td>
</tr>
<tr>
<td>Additional items</td>
<td>UF#(&lt;User coordinate system No.&gt;)</td>
<td>1 to 63</td>
</tr>
<tr>
<td></td>
<td>Data 1 Definition point ORG position data</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Data 2 Definition point XX position data</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Data 3 Definition point XY position data</td>
<td></td>
</tr>
<tr>
<td></td>
<td>IF statement</td>
<td></td>
</tr>
<tr>
<td>Example</td>
<td>MFRAME UF#(1) P001 P002 P003</td>
<td></td>
</tr>
<tr>
<td>CALL</td>
<td>Function</td>
<td>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.</td>
</tr>
<tr>
<td>Additional items</td>
<td>JOB:&lt;Job name&gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>IG#(&lt;Input group No.&gt;)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>B&lt;Variable No.&gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>UF#(&lt;User coordinate system No.&gt;)</td>
<td>1 to 63</td>
</tr>
<tr>
<td></td>
<td>IF statement</td>
<td></td>
</tr>
<tr>
<td>Example</td>
<td>CALL JOB:TEST-1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>CALL JOB:TEST-1 UF#(2)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>CALL IG#(02)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(Job call according to input signal pattern. In this case, job 0 cannot be called.)</td>
<td></td>
</tr>
<tr>
<td>JUMP</td>
<td>Function</td>
<td>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.</td>
</tr>
<tr>
<td>Additional items</td>
<td>JOB:&lt;Job name&gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>IG#(&lt;Input group No.&gt;)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>B&lt;Variable No.&gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&lt;Label No.&gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>UF#(&lt;User coordinate system No.&gt;)</td>
<td>1 to 63</td>
</tr>
<tr>
<td></td>
<td>IF statement</td>
<td></td>
</tr>
<tr>
<td>Example</td>
<td>JUMP JOB:TEST1 IF IN#(14)=OFF</td>
<td></td>
</tr>
</tbody>
</table>
Specifications are subject to change without notice for ongoing product modifications and improvements.