

DENSO ROBOT

VM-6083D/VM-60B1D/VS-E/HM-E/HS-E SERIES

SUPPLEMENT

Extended-Joints Support

Copyright © DENSO WAVE INCORPORATED, 2002

All rights reserved. No part of this publication may be reproduced in any form or by any means without permission in writing from the publisher.

Specifications are subject to change without prior notice.

All products and company names mentioned are trademarks or registered trademarks of their respective holders

Preface

Thank you for purchasing the DENSO high-speed, high-accuracy assembly robot.

This manual is a supplement to those manuals listed on the next page. It describes "Extended-joint support system" that enables you to control up to 2 extended joints with the robot controller.

This new extended-joint support system requires no extended-joint control box that is essential in the conventional extended-joint support system.

This manual covers:

Robot systems configured with the following extended-joint support robot controllers listed below.

| Extended-joint support robot controller | Applicable robot series |
|---|------------------------------|
| RC5-EBVME6BA | VM-6083D and VM-60B1D series |
| RC5-EBVSE6BA | VS-E series |
| RC5-EBHME4BA | HM-E series |
| RC5-EBHSE4BA | HS-E series |

Important

To ensure operator safety, be sure to read the precautions and instructions in "SAFETY PRECAUTIONS" on pages 1 through 9.

How the documentation set is organized

The documentation set consists of the following books. If you are unfamiliar with this robot and option(s), please read all books and understand them fully before operating your robot and option(s).

SUPPLEMENT -this book-

Describes the extended-joint support system. Use this book in conjunction with other manuals.

GENERAL INFORMATION ABOUT ROBOT

Provides the packing list of the robot and outlines of the robot system, robot unit, and robot controller.

INSTALLATION & MAINTENANCE GUIDE

Provides instructions for installing the robot components and customizing your robot, and maintenance & inspection procedures.

BEGINNER'S GUIDE

Introduces you to the DENSO robot. Taking an equipment setup example, this book guides you through running your robot with the teach pendant, making a program in WINCAPSII, and running your robot automatically.

SETTING-UP MANUAL

Describes how to set-up or teach your robot with the teach pendant, operating panel, or mini-pendant.

WINCAPSII GUIDE

Provides instructions on how to use the teaching system WINCAPSII which runs on the PC connected to the robot controller for developing and managing programs.

PROGRAMMER'S MANUAL (I), (II)

Describes the PAC programming language, program development, and command specifications in PAC.

RC5 CONTROLLER INTERFACE MANUAL

Describes the RC5 controller, interfacing with external devices, system- and user-input/output signals, and I/O circuits.

ERROR CODE TABLES

List error codes that will appear on the teach pendant, operating panel, or PC screen if an error occurs in the robot series or WINCAPSII. These tables provide detailed description and recovery ways.

OPTIONS MANUAL

Describes the specifications, installation, and use of optional devices.

SAFETY PRECAUTIONS

Be sure to observe all of the following safety precautions.

Strict observance of these warning and caution indications are a **MUST** for preventing accidents, which could result in bodily injury and substantial property damage. Make sure you fully understand all definitions of these terms and related symbols given below, before you proceed to the text itself.

| | |
|--|--|
|  WARNING | Alerts you to those conditions, which could result in serious bodily injury or death if the instructions are not followed correctly. |
|  CAUTION | Alerts you to those conditions, which could result in minor bodily injury or substantial property damage if the instructions are not followed correctly. |

Terminology and Definitions

Maximum space: Refers to the volume of space encompassing the maximum designed movements of all robot parts including the end-effector, workpiece and attachments. (Quoted from the RIA* Committee Draft.)

Restricted space: Refers to the portion of the maximum space to which a robot is restricted by limiting devices (i.e., mechanical stops). The maximum distance that the robot, end-effector, and workpiece can travel after the limiting device is actuated defines the boundaries of the restricted space of the robot. (Quoted from the RIA Committee Draft.)

Motion space: Refers to the portion of the restricted space to which a robot is restricted by software motion limits. The maximum distance that the robot, end-effector, and workpiece can travel after the software motion limits are set defines the boundaries of the motion space of the robot. (The "motion space" is DENSO WAVE-proprietary terminology.)

Operating space: Refers to the portion of the restricted space (or motion space in Denso robot) that is actually used by the robot while performing its task program. (Quoted from the RIA Committee Draft.)

Task program: Refers to a set of instructions for motion and auxiliary functions that define the specific intended task of the robot system. (Quoted from the RIA Committee Draft.)

(*RIA: Robotic Industries Association)

1. Introduction

This section provides safety precautions to be observed during installation, teaching, inspection, adjustment, and maintenance of the robot.

2. Installation Precautions

2.1 Insuring the proper installation environment

2.1.1 For standard type

The standard type has not been designed to withstand explosions, dust-proof, nor is it splash-proof. Therefore, it should not be installed in any environment where:

- (1) there are flammable gases or liquids,
- (2) there are any shavings from metal processing or other conductive material flying about,
- (3) there are any acidic, alkaline or other corrosive gases,
- (4) there is cutting or grinding oil mist,
- (5) it may likely be submerged in fluid,
- (6) there is sulfuric cutting or grinding oil mist, or
- (7) there are any large-sized inverters, high output/high frequency transmitters, large contactors, welders, or other sources of electrical noise.

2.1.2 For dust-proof, splash-proof type

The dust-proof, splash-proof type is an IP54-equivalent structure, but it has not been designed to withstand explosions. (The HM/HS-E-W and the wrist of the VM-D-W/VS-E-W are an IP65-equivalent dust-proof and splash-proof structure.)

Note that the robot controller is not a dust- or splash-proof structure. Therefore, when using the robot controller in an environment exposed to mist, put it in an optional protective box.

The dust-proof, splash-proof type should not be installed in any environment where:

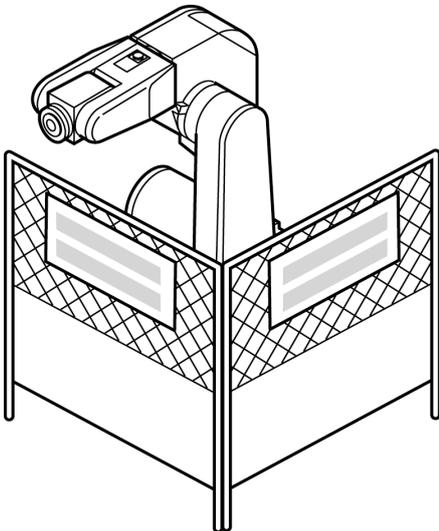
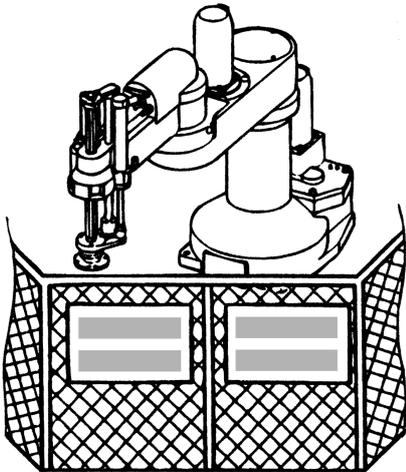
- (1) there are any flammable gases or liquids,
- (2) there are any acidic, alkaline or other corrosive gases,
- (3) there are any large-sized inverters, high output/high frequency transmitters, large contactors, welders, or other sources of electrical noise,
- (4) it may likely be submerged in fluid,
- (5) there are any grinding or machining chips or shavings,
- (6) any machining oil not specified in this manual is in use, or
Note: Yushiron Oil No. 4C (non-soluble) is specified.
- (7) there is sulfuric cutting or grinding oil mist.

2.2 Service space

The robot and peripheral equipment should be installed so that sufficient service space is maintained for safe teaching, maintenance, and inspection.

- 2.3 Control devices outside the robot's restricted space** The robot controller, teach pendant, and operating panel should be installed outside the robot's restricted space and in a place where you can observe all of the robot's movements when operating the robot controller, teach pendant, or operating panel.
- 2.4 Positioning of gauges** Pressure gauges, oil pressure gauges and other gauges should be installed in an easy-to-check location.
- 2.5 Protection of electrical wiring and hydraulic/pneumatic piping** If there is any possibility of the electrical wiring or hydraulic/pneumatic piping being damaged, protect them with a cover or similar item.
- 2.6 Positioning of emergency stop switches** Emergency stop switches should be provided in a position where they can be reached easily should it be necessary to stop the robot immediately.
- (1) The emergency stop switches should be red.
 - (2) Emergency stop switches should be designed so that they will not be released after pressed, automatically or mistakenly by any other person.
 - (3) Emergency stop switches should be separate from the power switch.
- 2.7 Positioning of operating status indicators** Operating status indicators should be positioned in such a way where workers can easily see whether the robot is on temporary halt or on an emergency or abnormal stop.

2.8 Setting-up the safety fence or enclosure



A safety fence or enclosure should be set up so that no one can easily enter the robot's restricted space. If it is impossible, utilize other protectors as described in Section 2.9.

- (1) The fence or enclosure should be constructed so that it cannot be easily moved or removed.
- (2) The fence or enclosure should be constructed so that it cannot be easily damaged or deformed through external force.
- (3) Establish the exit/entrance to the fence or enclosure. Construct the fence or enclosure so that no one can easily get past it by climbing over the fence or enclosure.
- (4) The fence or enclosure should be constructed to ensure that it is not possible for hands or any other parts of the body to get through it.
- (5) Take any one of the following protections for the entrance/exit of the fence or enclosure:
 - 1) Place a door, rope or chain across the entrance/exit of the fence or enclosure, and fit it with an interlock that ensures the emergency stop device operates automatically if it is opened or removed.
 - 2) Post a warning notice at the entrance/exit of the fence or enclosure stating "In operation--Entry forbidden" or "Work in progress--Do not operate" and ensure that workers follow these instructions at all times.

When making a test run, before setting up the fence or enclosure, place an overseer in a position outside the robot's restricted space and one in which he/she can see all of the robot's movements. The overseer should prevent workers from entering the robot's restricted space and be devoted solely to that task.

2.9 Positioning of rope or chain

If it is not possible to set up the safety fence or enclosure described in Section 2.8, hang a rope or chain around the perimeter of the robot's restricted space to ensure that no one can enter the restricted space.

- (1) Ensure the support posts cannot be moved easily.
- (2) Ensure that the rope or chain's color or material can easily be discerned from the surrounds.
- (3) Post a warning notice in a position where it is easy to see stating "In operation--Entry forbidden" or "Work in progress --Do not operate" and ensure that workers follow these instructions at all times.
- (4) Set the exit/entrance, and follow the instructions given in Section 2.8, (3) through (5).

2.10 Setting the robot's motion space

The area required for the robot to work is called the robot's operating space.

If the robot's motion space is greater than the operating space, it is recommended that you set a smaller motion space to prevent the robot from interfering or disrupting other equipment.

Refer to the "INSTALLATION & MAINTENANCE GUIDE."

2.11 No robot modification allowed

Never modify the robot unit, robot controller, teach pendant or other devices.

2.12 Cleaning of tools

If your robot uses welding guns, paint spray nozzles, or other end-effectors requiring cleaning, it is recommended that the cleaning process be carried out automatically.

2.13 Lighting

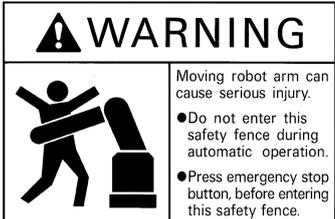
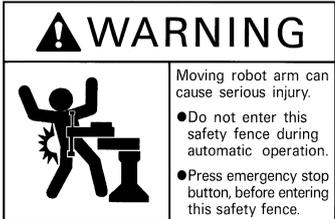
Sufficient illumination should be assured for safe robot operation.

2.14 Protection from objects thrown by the end-effector

If there is any risk of workers being injured in the event that the object being held by the end-effector is dropped or thrown by the end-effector, consider the size, weight, temperature and chemical nature of the object and take appropriate safeguards to ensure safety.

2.15 Affixing the warning label

Place the warning label packaged with the robot on the exit/entrance of the safety fence or in a position where it is easy to see.

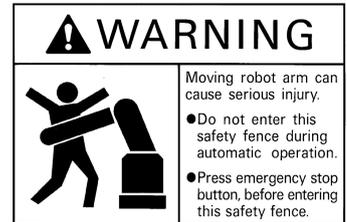
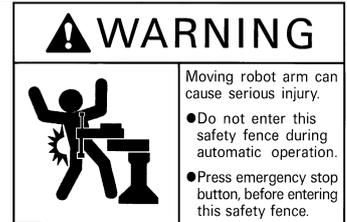


3. Precautions while robot is running



Warning

Touching the robot while it is in operation can lead to serious injury. Please ensure the following conditions are maintained and that the cautions listed from Section 3.1 onwards are followed when any work is being performed.



- 1) Do not enter the robot's restricted space when the robot is in operation or when the motor power is on.
- 2) As a precaution against malfunction, ensure that an emergency stop device is activated to cut the power to the robot motor upon entry into the robot's restricted space.
- 3) When it is necessary to enter the robot's restricted space to perform teaching or maintenance work while the robot is running, ensure that the steps described in Section 3.3 "Ensuring safety of workers performing jobs within the robot's restricted space" are taken.

3.1 Creation of working regulations and assuring worker adherence

When entering the robot's restricted space to perform teaching or maintenance inspections, set "working regulations" for the following items and ensure workers adhere to them.

- (1) Operating procedures required to run the robot.
- (2) Robot speed when performing teaching.
- (3) Signaling methods to be used when more than one worker is to perform work.
- (4) Steps that must be taken by the worker in the event of a malfunction, according to the contents of the malfunction.
- (5) The necessary steps for checking release and safety of the malfunction status, in order to restart the robot after robot movement has been stopped due to activation of the emergency stop device
- (6) Apart from the above, any steps below necessary to prevent danger from unexpected robot movement or malfunction of the robot.
 - 1) Display of the control panel (See Section 3.2 on the following page)
 - 2) Assuring the safety of workers performing jobs within the robot's restricted space (See Section 3.3 on the following page)

3) Maintaining worker position and stance

Position and stance that enables the worker to confirm normal robot operation and to take immediate refuge if a malfunction occurs.

4) Implementation of measures for noise prevention

5) Signaling methods for workers of related equipment

6) Types of malfunctions and how to distinguish them

Please ensure "working regulations" are appropriate to the robot type, the place of installation and to the content of the work.

Be sure to consult the opinions of related workers, engineers at the equipment manufacturer and that of a labor safety consultant when creating these "working regulations".

3.2 Display of operation panel

To prevent anyone other than the worker from accessing the start switch or the changeover switch by accident during operation, display something to indicate it is in operation on the operating panel or teach pendant. Take any other steps as appropriate, such as locking the cover.

3.3 Ensuring safety of workers performing jobs within the robot's restricted space

When performing jobs within the robot's restricted space, take any of the following steps to ensure that robot operation can be stopped immediately upon a malfunction.

- (1) Ensure an overseer is placed in a position outside the robot's restricted space and one in which he/she can see all robot movements, and that he/she is devoted solely to that task.
 - ① An emergency stop device should be activated immediately upon a malfunction.
 - ② Do not permit anyone other than the worker engaged for that job to enter the robot's restricted space.
- (2) Ensure a worker within the robot's restricted space carries the portable emergency stop switch so he/she can press it (the robot stop button on the teach pendant) immediately if it should be necessary to do so.

3.4 Inspections before commencing work such as teaching

Before starting work such as teaching, inspect the following items, carry out any repairs immediately upon detection of a malfunction and perform any other necessary measures.

- (1) Check for any damage to the sheath or cover of the external wiring or to the external devices.
- (2) Check that the robot is functioning normally or not (any unusual noise or vibration during operation).
- (3) Check the functioning of the emergency stop device.
- (4) Check there is no leakage of air or oil from any pipes.
- (5) Check there are no obstructive objects in or near the robot's restricted space.

3.5 Release of residual air pressure

Before disassembling or replacing pneumatic parts, first release any residual air pressure in the drive cylinder.

3.6 Precautions for test runs

Whenever possible, have the worker stay outside of the robot's restricted space when performing test runs.

3.7 Precautions for automatic operation

(1) At start-up

Before the robot is to be started up, first check the following items as well as setting the signals to be used and perform signaling practice with all related workers.

- 1) Check that there is no one inside the robot's restricted space.
 - 2) Check that the teach pendant and tools are in their designated places.
 - 3) Check that no lamps indicating a malfunction on the robot or related equipment are lit.
- (2) Check that the display lamp indicating automatic operation is lit during automatic operation.

(3) Steps to be taken when a malfunction occurs

Should a malfunction occur with the robot or related equipment and it is necessary to enter the robot's restricted space to perform emergency maintenance, stop the robot's operation by activating the emergency stop device. Take any necessary steps such as placing a display on the starter switch to indicate work is in progress to prevent anyone from accessing the robot.

3.8 Precautions in repairs

- (1) Do not perform repairs outside of the designated range.
- (2) Under no circumstances should the interlock mechanism be removed.
- (3) When opening the robot controller's cover for battery replacement or any other reasons, always turn the robot controller power off and disconnect the power cable.
- (4) Use only spare tools specified in this manual.

4. Daily and periodical inspections

- (1) Be sure to perform daily and periodical inspections. Before starting jobs, always check that there is no problem with the robot and related equipment. If any problems are found, take any necessary measures to correct them.
- (2) When carrying out periodical inspections or any repairs, maintain records and keep them for at least 3 years.

5. Management of floppy disks

- (1) Carefully handle and store the "Initial settings" floppy disks packaged with the robot, which store special data exclusively prepared for your robot.
- (2) After finishing teaching or making any changes, always save the programs and data onto floppy disks.
Making back-ups will help you recover if data stored in the robot controller is lost due to the expired life of the back-up battery.
- (3) Write the names of each of the floppy disks used for storing task programs to prevent incorrect disks from loading into the robot controller.
- (4) Store the floppy disks where they will not be exposed to dust, humidity and magnetic field, which could corrupt the disks or data stored on them.



CONTENTS

| | |
|--|-----------|
| Preface | i |
| How the documentation set is organized | ii |
| SAFETY PRECAUTIONS..... | 1 |
| Commands Listed in Alphabetical Order (Follows the Contents.) | |
| | |
| Chapter 1 Overview | |
| ■ Configuration of extended-joint support system..... | 1-1 |
| ■ Extended-joint related functions newly added..... | 1-2 |
| | |
| Chapter 2 Robot Components of Extended-Joint Support System | |
| 2.1 Configuration of Extended-Joint Support System..... | 2-1 |
| 2.2 Extended-Joint Support Robot Controller..... | 2-2 |
| 2.2.1 Robot Controller Models | 2-2 |
| 2.2.2 IPM Boards that Enable Robot Controllers to Support Extended Joints..... | 2-2 |
| 2.2.3 Encoder backup battery for extended-joint support robot controller | 2-3 |
| 2.2.4 Exhaust air filter for extended-joint support robot controller | 2-4 |
| 2.2.5 Total Output Capacity for Extended Joints | 2-5 |
| 2.2.6 Connecting Cables to the Robot Controller | 2-7 |
| 2.3 Parts Codes of Optional Components for the Extended-Joint Support System..... | 2-8 |
| 2.4 Detailed Specifications of Extended-Joint Related Options | 2-10 |
| 2.4.1 Extended-Joint Motors..... | 2-10 |
| 2.4.2 External Dimensions of Extended-Joint Motors | 2-11 |
| 2.4.3 Specifications of Extended-Joint Motors..... | 2-16 |
| 2.4.4 Safety Rules for Using Extended-Joint Motors | 2-21 |
| | |
| Chapter 3 Extended-Joint Related Function | |
| 3.1 Extended-Joint Function | 3-1 |
| 3.1.1 Operating Procedures of the Extended-Joint Function | 3-1 |
| 3.1.2 Commands Supporting Extended-Joints..... | 3-20 |
| 3.2 Setting the Extended-Joint Parameters..... | 3-52 |
| 3.2.1 Calling up "Path Parameters for Ex-Joint" and "Servo Parameters for Ex-Joint" Windows..... | 3-52 |
| 3.2.2 Detailed Description of Extended-Joint Parameter Setting..... | 3-59 |
| 3.2.3 Gain Tuning of Extended-Joints | 3-65 |
| 3.2.4 Extended-Joints Exclusive Operations..... | 3-78 |
| 3.2.5 Extended-Joint Parameter Setting Commands..... | 3-85 |

Commands Listed in Alphabetical Order

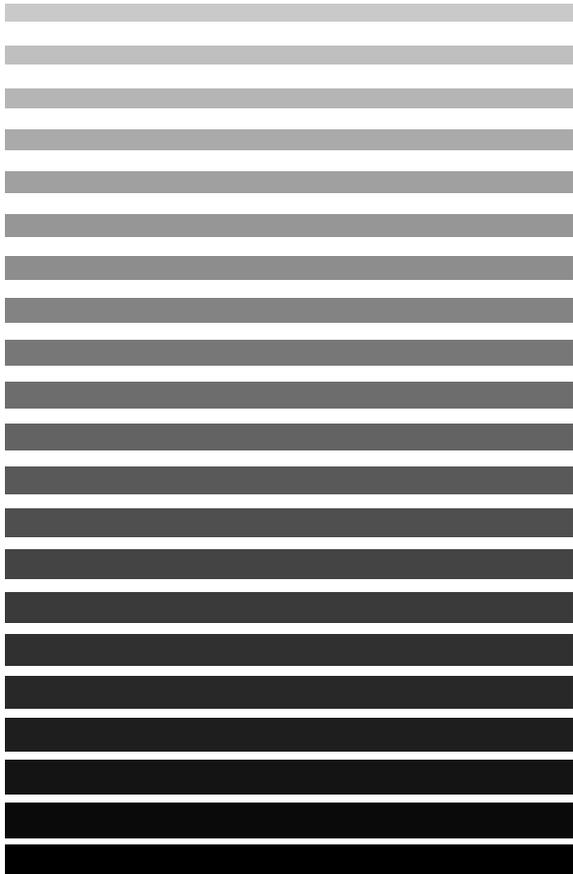
| Commands | Functions | Refer to: |
|------------------|---|-----------|
| A | | |
| ACCEL | Specifies the internal composite acceleration/deceleration of joints included in a currently held arm group. | 3-33 |
| ARRIVE | Defines the motion ratio relative to the programmed full travel distance to the target point in order to make the current program stand by to execute the next step until the robot reaches the defined motion ratio. | 3-30 |
| C | | |
| ClearSrvMonitor | Initializes the pointer of data obtained by the single-joint servo data monitor function. | 3-88 |
| CURACC | Gets the current internal composite acceleration of joints included in a currently held arm group. | 3-39 |
| CURDEC | Gets the current internal composite deceleration of joints included in a currently held arm group. | 3-41 |
| CUREXJ | Gets the current angle of an extended-joint into a floating-point variable. | 3-28 |
| CURJACC | Gets the current internal acceleration of individual joints included in a currently held arm group. | 3-40 |
| CURJDEC | Gets the current internal deceleration of individual joints included in a currently held arm group. | 3-42 |
| CURJSPD | Gets the current internal speed of individual joints included in a currently held arm group. | 3-38 |
| CURSPD | Gets the current internal composite speed of joints included in a currently held arm group. | 3-37 |
| D | | |
| DECEL | Specifies the internal composite deceleration of joints involved in a currently held arm group. | 3-35 |
| DESTEXJ | Gets the target position of an extended-joint invoked by the current motion command into a floating-point variable. If the robot is on halt, this command will get the current position (commanded value). | 3-29 |
| DRIVE | Carries out the relative motion of each joint. | 3-24 |
| DRIVEA | Carries out the absolute motion of each joint. | 3-25 |
| G | | |
| GetJntData | Gets the internal servo data of a specified joint. | 3-92 |
| GetSrvData | Gets the internal servo data of robot joints. | 3-91 |
| GIVEARM | Releases the currently held arm group. | 3-23 |
| I | | |
| INTERRUPT ON/OFF | Interrupts the current robot motion. | 3-43 |

| Commands | Functions | Refer to: |
|----------------------|--|-----------|
| J | | |
| JACCEL | Specifies the internal acceleration and deceleration of individual joints included in a currently held arm group. | 3-34 |
| JDECEL | Specifies the internal deceleration ratio of individual joints included in a currently held arm group. | 3-36 |
| JSPEED | Specifies the internal speed of individual joints included in a currently held arm group. | 3-32 |
| M | | |
| MotionComp | Judges whether execution of running motion commands is complete. | 3-90 |
| MotionSkip | Aborts running motion commands. | 3-89 |
| MOVE | Moves the robot flange to the specified coordinates. If specified with an EX option (relative motion of extended-joints) or EXA option (absolute motion of extended-joints), the MOVE can move both the robot flange and the extended-joints synchronously. | 3-26 |
| mvResetPulseWidthJnt | Resets the encoder pulse count for an allowable positioning error for a specified extended-joint to the default. | 3-47 |
| mvSetPulseWidthJnt | Sets the encoder pulse count for an allowable positioning error for a specified extended-joint. | 3-46 |
| P | | |
| POSCLR | Forcibly restores the current position of a joint to 0 mm or 0 degree. | 3-45 |
| R | | |
| ResetCurLmt | Releases the drive current limit set for a specified joint motor. | 3-50 |
| ResetCycloidJnt | Cancel the cycloid mode set for a specified extended-joint and restores the normal mode. | 3-49 |
| ResetEralw | Resets the positioning error allowance of a specified joint to the initial value. | 3-51 |
| S | | |
| SetCycloidJnt | Enters a specified extended-joint into the cycloid mode where the controller suppresses the peak of overshoot and residual oscillation that would occur in an end motion. | 3-48 |
| SetMonitorCond | Sets the monitoring conditions for single-joint servo data monitor. | 3-85 |
| SPEED | Specifies the internal composite speed of joints included in a currently held arm group. | 3-31 |
| StartSrvMonitor | Starts monitoring single-joint servo data. | 3-86 |
| StopSrvMonitor | Stops monitoring single-joint servo data. | 3-87 |
| T | | |
| TAKEARM | Gets an arm group. Upon execution of this statement, the programmed speed, acceleration and deceleration will be set to 100. If the gotten arm group includes any robot joint, this statement restores the tool coordinates and work coordinates to the origin. | 3-20 |

Chapter 1



Overview



This chapter outlines the extended-joint support system.

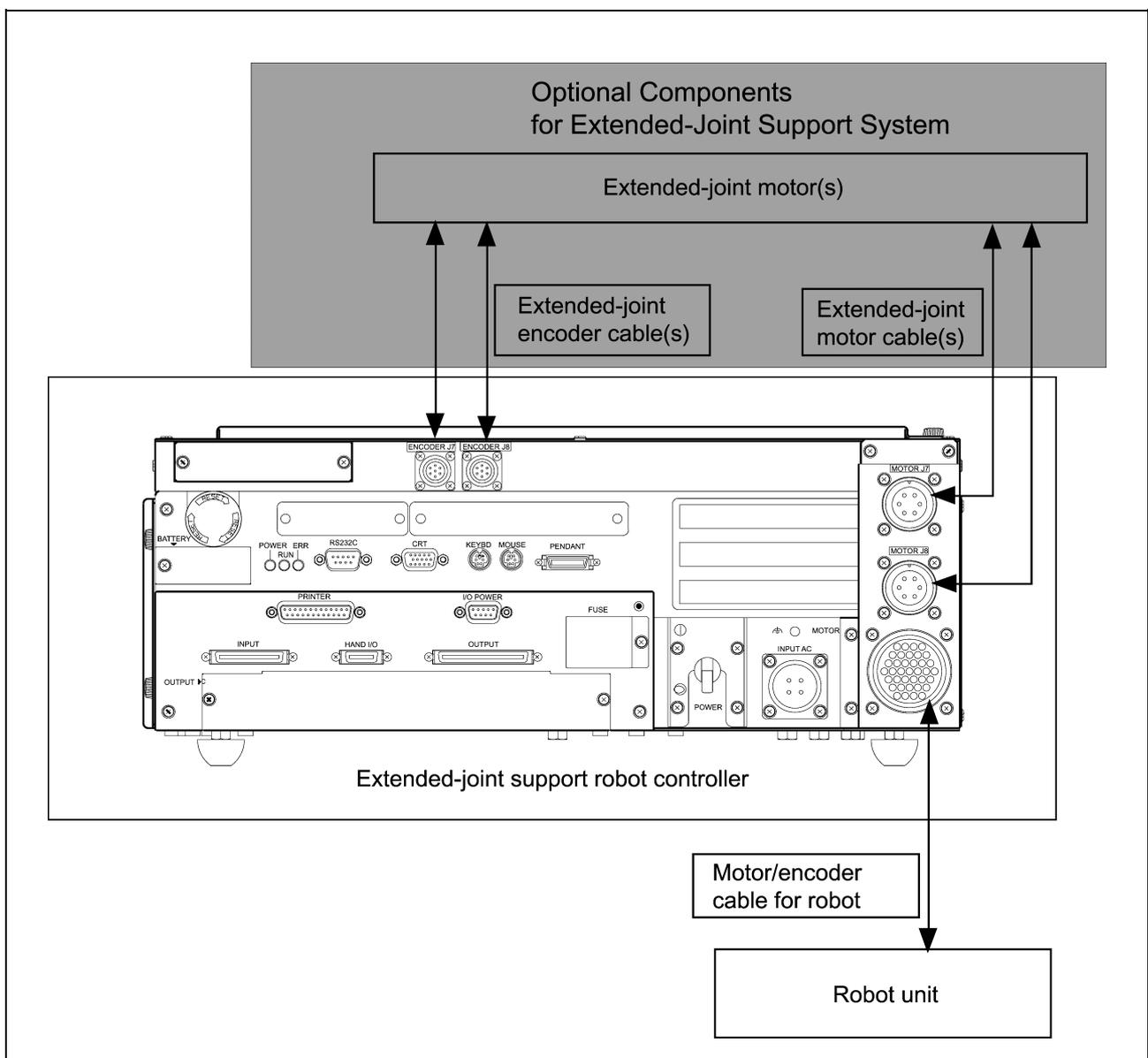
Chapter 1 Overview

The extended-joint support system, which can be optionally configured for the VM-6083D/VM-60B1D/VS-E/HM-E/HS-E series, enables the extended-joint support robot controller to control up to two extended-joints in addition to the conventional robot joints.

■ Configuration of extended-joint support system

The extended-joint support system is different from the conventional robot system in the following points:

- (1) Robot controller which is equipped with an IPM board (factory option) suited to extended-joint motors to be connected
- (2) Extended-joint motor cables and encoder cables (both are options) between the robot controller and extended-joint motors
- (3) Extended-joint motors (option) which are equipped with bus-line encoders (Up to 2 motors, motor type selectable)



■ Extended-joint related functions newly added

(1) Extended-joint function

The extended-joint function allows you to control extra joints independently of robot joints through the standard interface of the robot controller (NetwoRC).

(2) Setting extended-joint parameters

To use extended-joints, you need to set extended-joint path parameters and extended-joint servo parameters by using the teach pendant. The former is for motion definitions (including speed, acceleration, and range of motion); the latter is for setting the gain and others of the extended-joint servo system.

(3) Gain tuning for extended-joints

After setting the motion conditions of extended-joints and checking the motion of the optional mechanism connected to the extended-joint motors, you need to do gain tuning for the servo system. There are the following two types of tuning methods:

- Auto gain tuning

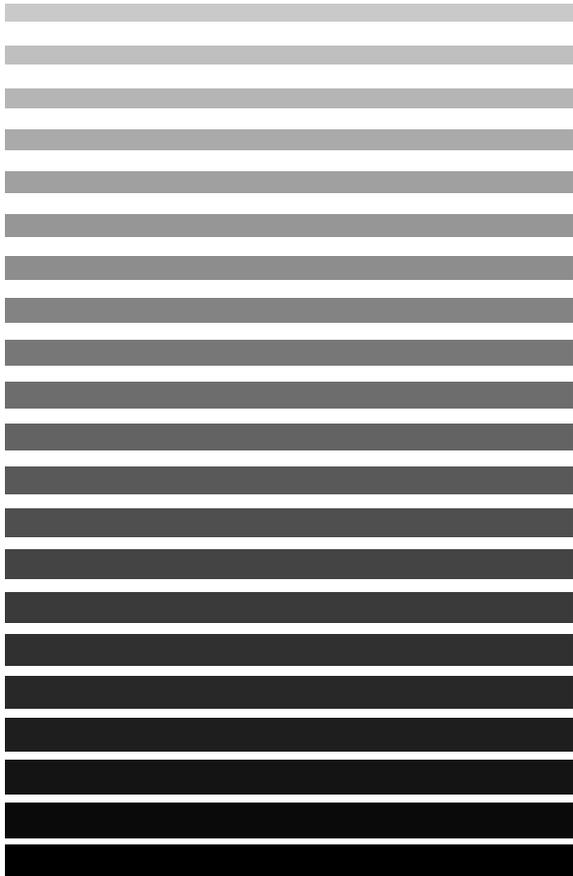
The robot controller performs acceleration/deceleration operation of the extended-joints according to the default pattern preset in the controller. Based on the motion of the extended-joints in that operation, the controller will estimate the inertia of payload and set the appropriate gain automatically. With this method, you may easily do gain tuning of the extended-joint motors.

- Manual gain tuning

The monitor function of the single-joint servo data monitors the motor speed control value, current motor speed, motor angle deviation, and torque control value. According to the monitored results, you may adjust the gain and torque control filter parameters for optimizing the motion of the extended-joints.

Chapter 2

Robot Components of Extended-Joint Support System



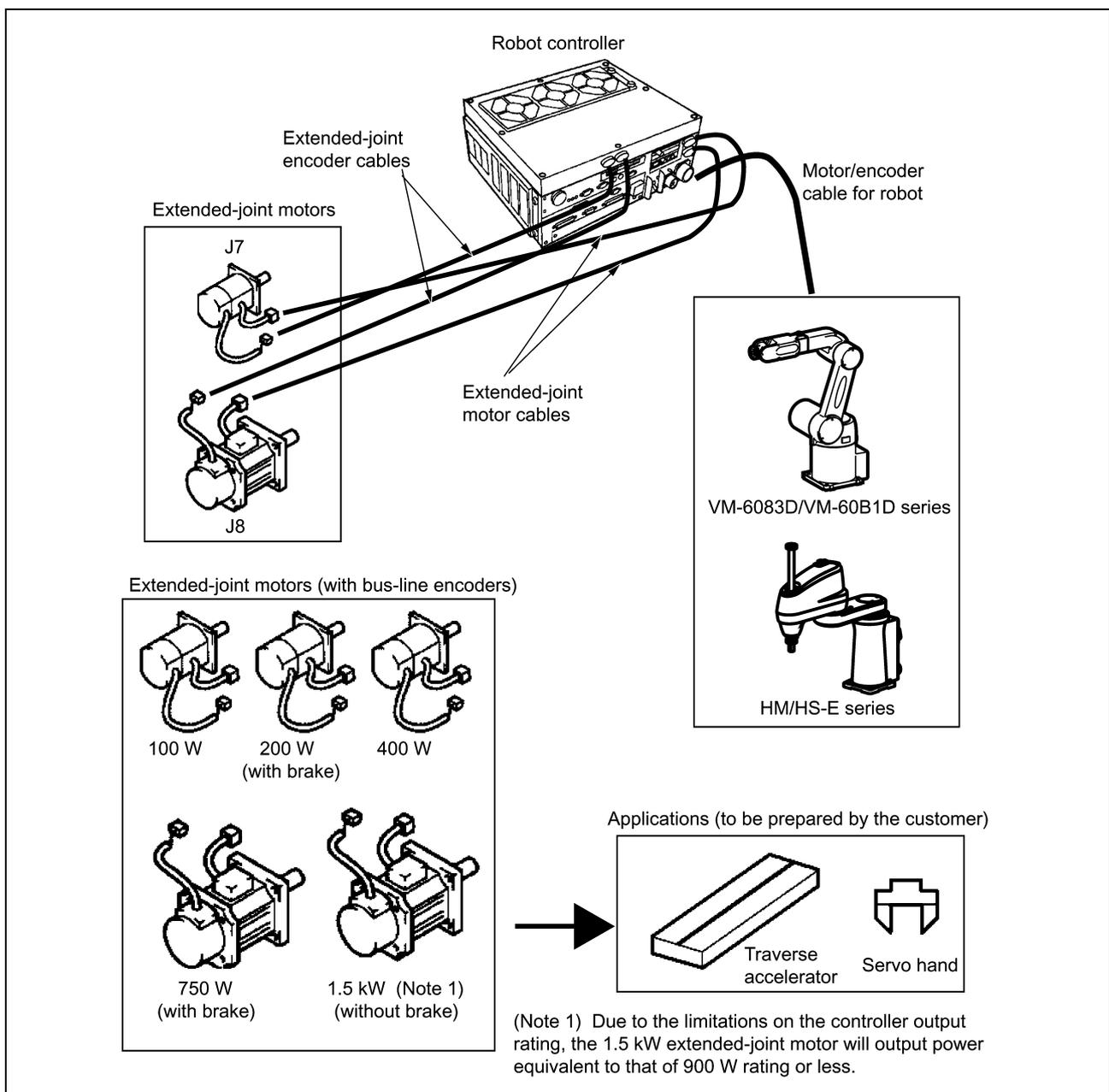
This chapter describes the components of the extended-joint support system and their specifications.

2.1 Configuration of Extended-Joint Support System

You may configure an extended-joint support system (see the illustration given below) by using a robot controller designed for extended-joint motors to be connected and connecting the options (DENSO genuine components) listed below.

- Extended-joint motors equipped with bus-line encoders (Up to 2 motors, motor type selectable)
- Extended-joint encoder cables (between the robot controller and extended-joint motors)
- Extended-joint motor cables (between the robot controller and extended-joint motors)

Extended-Joint Support System



2.2 Extended-Joint Support Robot Controller

2.2.1 Robot Controller Models

RC5-EBVME6BA: Extended-joint support controller for VM-6083D/VM-60B1D series

RC5-EBVSE6BA: Extended-joint support controller for VS-E series

RC5-EBHME4BA: Extended-joint support controller for HM-E series

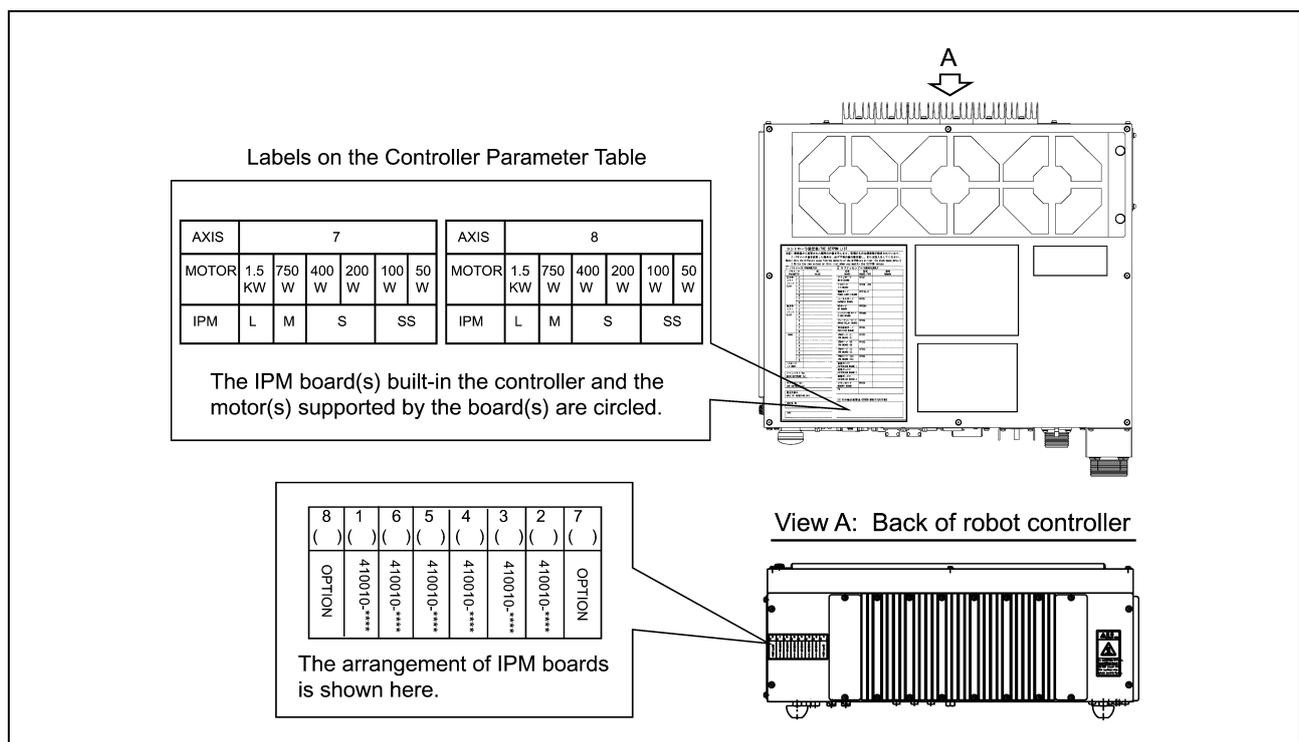
RC5-EBHSE4BA: Extended-joint support controller for HS-E series

2.2.2 IPM Boards that Enable Robot Controllers to Support Extended Joints

When shipped from the factory, an IPM board suited to extended-joint motors to be connected will be built into a robot controller.

Note: On the top of the robot controller is a "Controller Parameter Table" which is labeled "Extended-joint motors and IPM boards" as shown below. Check the label before connecting extended-joint motors.

On the back of the robot controller is an arrangement label of the IPM boards as shown below.



2.2.3 Encoder backup battery for extended-joint support robot controller

The encoder backup battery for the extended-joint support robot controller is mounted in the controller.

| Part Name | Part No. | Remarks |
|--|-------------|-------------|
| Encoder backup battery for extended-joint support robot controller | 410076-0270 | One battery |

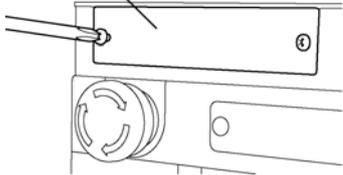
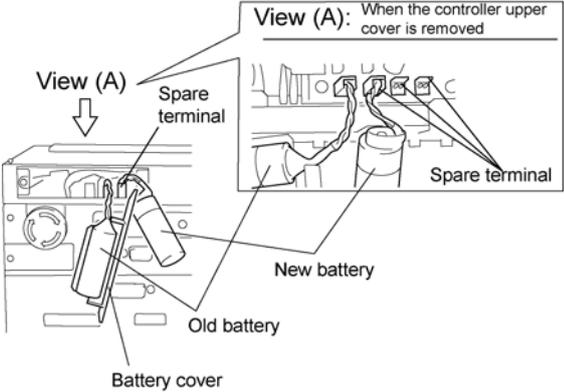
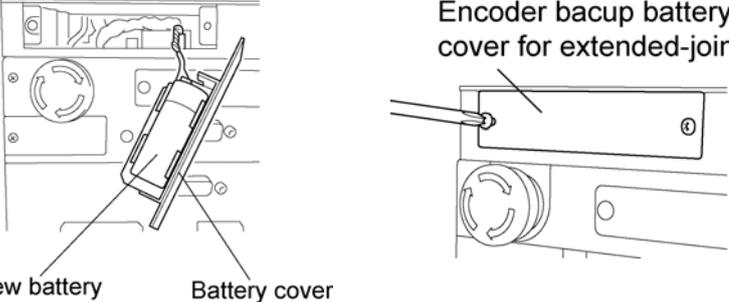
Replace the encoder backup battery for the extended-joint support robot controller, according to the procedure below.

Note : Precautions when replacing the battery

(1) Do not disconnect the extended-joint encoder cable from the controller when replacing the battery.

(2) Do not disconnect old backup battery before connecting a new one to the spare terminal.

If you do so, the encoder positional data may be lost.

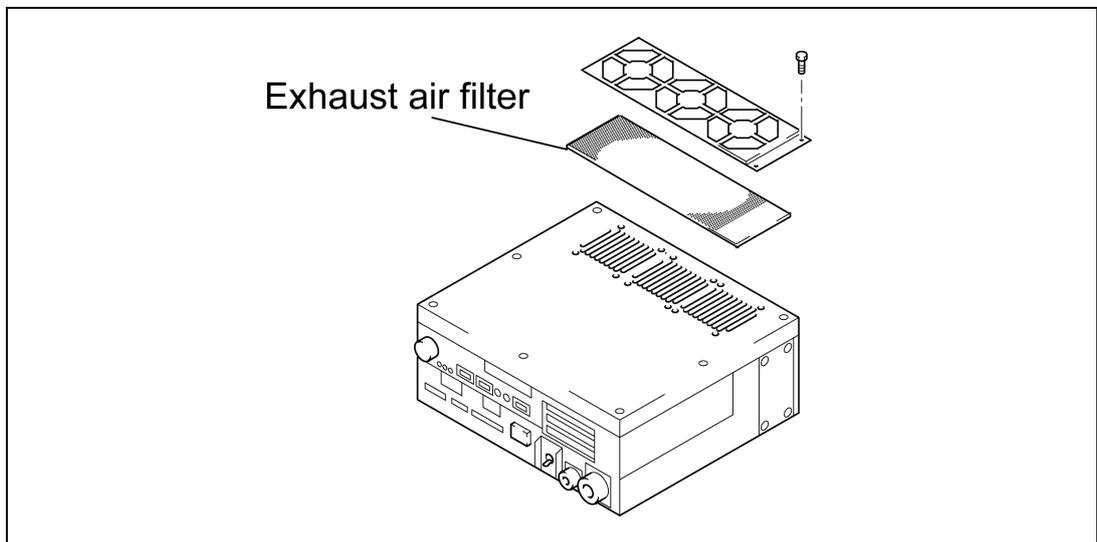
| Procedure | Illustration |
|--|--|
| <p>(1) Prepare a new backup battery for replacement.</p> <p>(2) Turn the controller power OFF.</p> |  |
| <p>(3) Loosen two screws and pull out the battery cover with battery.</p> | <p style="text-align: center;">Encoder bacup battery cover for extended-joint</p>  |
| <p>(4) Connect the new backup battery to the spare terminal.</p> <p>Note: Three spare terminals can be used.</p> | <p style="text-align: center;">View (A): When the controller upper cover is removed</p>  |
| <p>(5) Disconnect the old battery from the controller and fix the new battery to the battery cover folder.</p> <p>(6) Reinstall the battery cover.</p> | <p style="text-align: center;">Encoder bacup battery cover for extended-joint</p>  |

2.2.4 Exhaust air filter for extended-joint support robot controller

The exhaust air filter for extended-joint support robot controller is shown in the figure below.

| Part Name | Part No. | Remarks |
|--|-------------|---|
| Exhaust air filter for extended-joint support robot controller | 410041-1670 | Refer to “ VM-D series INSTALLATION AND MAINTENANCE GUIDE, Section 3.6. Supplies and Tools for Maintenance” |

The replacement procedure of this filter is same as standard VM-series controller.



Exhaust air filter for extended-joint support robot controller

2.2.5 Total Output Capacity for Extended Joints

Up to two extended-joint motors may be connected to an extended-joint support robot controller. Note that there is the restriction on the total output capacity of the robot controller for extended joints as listed below.

To connect extended-joint motors to the robot controller, you need to use dedicate cables listed in Section 2.3, "Parts Codes of Optional Components for the Extended-Joint Support System."

(1) If supplied with 3-phase, 200V power: Max. 3 kW

* Values in parentheses () in the table below denote motor capacities when conventional motors (for VM or VS) are used.

| Applicable robot and controller | | | Total output capacity for extended joints, when both the robot unit and extended joints operate simultaneously: (Note 1) | Total output capacity for extended joints, when robot unit is on halt: (Note 2) | |
|---------------------------------|---------------------|-------------------------|--|--|---|
| Robot series | Motor capacity | Robot controller models | | If extended-joints will be subjected to reaction force (e.g., when used as traverse accelerator): (Note 3) | If extended-joints will not be subjected to reaction force: |
| VM-6083D VM-60B1D | 3150 W | RC5-EBVME6BA | Not allowed. | Max. 525 W | Max. 2170 W |
| VS-E | 1580 W | RC5-EBVSE6BA | Max. 1420 W | Max. 1760 W | Max. 2580 W |
| HM-E | 2200 W (3100 W)* | RC5-EBHME4BA | Not allowed. | Max. 300 W | Max. 2600 W |
| HS-E | 1300 W (2200 W)* | RC5-EBHSE4BA | Max. 800 W | Max. 1060 W | Max. 2760 W |

(Note 1) When both the robot unit and extended-joint motors operate simultaneously, the total output capacity for extended joints may apply as listed above.

Connection example: If the output capacity for extended joints is 800 W, the robot controller may connect up to two joints of 400 W each.

(Note 2) If extended-joint motors are to be driven only when the robot is on halt, these output capacities may apply.

(Note 3) The total output capacity for extended-joint motors is further restricted if extended-joints are used as traverse accelerators of the robot so that they receive reaction force from the robot unit.

(2) If supplied with single-phase, 230V power: Max. 2 kW

* Values in parentheses () in the table denote motor capacities when conventional motors (for VM or VS) are used. The VM series may not be operated on single-phase power.

| Applicable robot and controller | | | Total output capacity for extended joints, when both the robot unit and extended joints operate simultaneously: (Note 1) | Total output capacity for extended joints, when robot unit is on halt: (Note 2) | |
|---------------------------------|---------------------|-------------------------|--|--|---|
| Robot series | Motor capacity | Robot controller models | | If extended-joints will be subjected to reaction force (e.g., when used as traverse accelerator): (Note 3) | If extended-joints will not be subjected to reaction force: |
| VM-6083D* VM-60B1D* | — | — | — | — | — |
| VS-E | 1580 W | RC5-EBVSE6BA | Max. 420 W | Max. 760 W | Max. 1580 W |
| HM-E | 2200 W (3100 W)* | RC5-EBHME4BA | Not allowed. | Not allowed. | Max. 1600 W |
| HS-E | 1300 W (2200 W)* | RC5-EBHSE4BA | Not allowed. | Max. 60 W | Max. 1760 W |

(Note 1) When both the robot unit and extended-joint motors operate simultaneously, the total output capacity for extended joints may apply as listed above.

Connection example: If the output capacity for extended-joints is 800 W, the robot controller may connect up to two joints of 400W each.

(Note 2) If extended-joint motors are to be driven only when the robot is on halt, these output capacities may apply.

(Note 3) The total output capacity for extended-joint motors is further restricted if extended-joints are used as traverse accelerators of the robot so that they receive reaction force from the robot unit.

⚠ CAUTION

(1) Use the extended-joint support robot controller with specified motors whose total capacity is less than the allowable limit.

If the total capacity of connected motors exceeds the limit, the robot controller will cause an error. In the worst case, the internal circuits in the robot controller may be damaged.

(2) Even with the specified capacity of motors, the robot controller may cause a power error, overcurrent, overvoltage, overload and other errors, depending upon the use conditions of the extended-joint motors.

You may solve those problems by adjusting acceleration, deceleration, maximum speed, and gain of each extended-joint servo loop.

2.2.6 Connecting Cables to the Robot Controller

A hook-up example of the extended-joint support system is illustrated below, where the VS-E series is connected.

Refer to this example and the connector names given in Section 2.2.3.

⚠ CAUTION

- (1) When using only one extended joint, be sure to cover the other encoder connector not in use with an encoder dummy cap that comes with the extended-joint support robot controller.

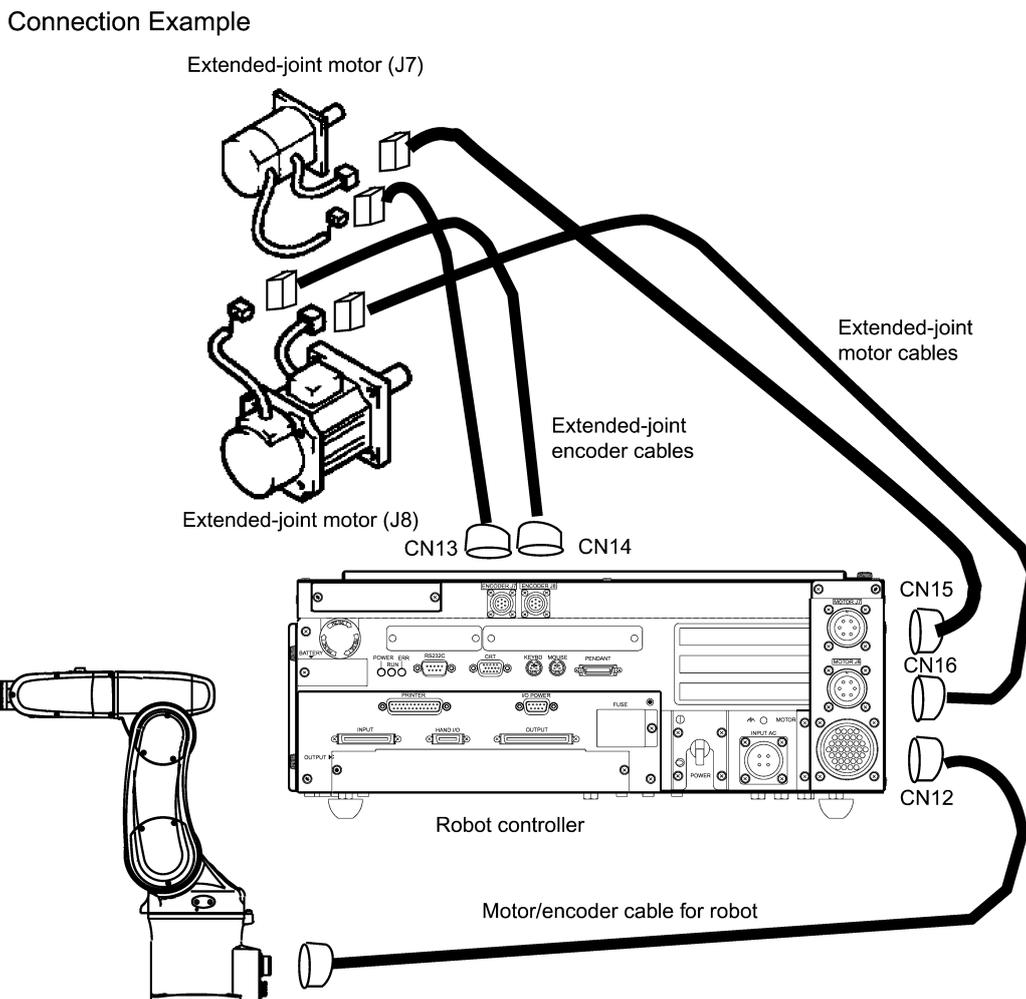
(If an extended-joint motor is connected to the J7 encoder connector, cover the J8 with the dummy cap; and vice versa.)

Without an encoder dummy cap connected, the extended-joint support system will not operate normally.

- (2) Use DENSO WAVE-authorized cables given in Section 2.3 "Parts Codes of Optional Components for the Extended-Joint Support System."

Never modify those cables or use cables other than DENSO WAVE-authorized ones to configure an extended-joint support system.

- (3) Always ground the shield conductors of extended-joint cables; otherwise, malfunctions may result due to noises.



2.3 Parts Codes of Optional Components for the Extended-Joint Support System_[p12]

For extended-joint support system, the following components are available as options. Purchase them according to your needs.

For components other than extended-joint related ones, refer to the "INSTALLATION & MAINTENANCE GUIDE."

| No. | Components | Parts No. | Remarks |
|------|--|-------------|---------------------------------------|
| 1 | Extended-joint motor, 100 W (With brake) | 410622-1510 | |
| 2 | Extended-joint motor, 200 W (With brake) | 410622-1520 | |
| 3 | Extended-joint motor, 400 W (With brake) | 410622-1530 | |
| 4 | Extended-joint motor, 750 W (With brake) | 410622-1540 | |
| 5 | Extended-joint motor, 1.5 kW (Without brake) (Note 1) | 410622-1550 | |
| 6 | Extended-joint cable set | 410149-0660 | Including Nos. 6-1 and 6-2. |
| 6-1 | Extended-joint encoder cable (Standard, 3 m) | 410141-2210 | |
| 6-2 | Extended-joint motor cable (Standard, 3 m) | 410141-2130 | For 100W, 200W or 400W motor |
| 7 | Extended-joint cable set | 410149-0670 | Including Nos. 7-1 and 7-2. |
| 7-1 | Extended-joint encoder cable (Standard, 3 m) | 410141-2210 | |
| 7-2 | Extended-joint motor cable (Standard, 3 m) | 410141-2150 | For 750 W or 1.5 kW motor (Note 1) |
| 8 | Extended-joint cable set | 410149-0680 | Including Nos. 8-1 and 8-2. |
| 8-1 | Extended-joint encoder cable (Standard, 6 m) | 410141-2220 | |
| 8-2 | Extended-joint motor cable (Standard, 6 m) | 410141-2140 | For 100W, 200W or 400W motor |
| 9 | Extended-joint cable set | 410149-0690 | Including Nos. 9-1 and 9-2. |
| 9-1 | Extended-joint encoder cable (Standard, 6 m) | 410141-2220 | |
| 9-2 | Extended-joint motor cable (Standard, 6 m) | 410141-2160 | For 750 W or 1.5 kW motor (Note 1) |
| 10 | Extended-joint cable set | 410149-0840 | Including Nos. 10-1 and 10-2. |
| 10-1 | Extended-joint encoder cable (Standard, 12m) | 410141-2460 | |
| 10-2 | Extended-joint motor cable (Standard, 12 m) | 410141-2480 | For 100W, 200W or 400W motor |
| 11 | Extended-joint cable set | 410149-0850 | Including Nos. 11-1 and 11-2. |
| 11-1 | Extended-joint encoder cable (Standard, 12m) | 410141-2460 | |
| 11-2 | Extended-joint motor cable (Standard, 12 m) | 410141-2500 | For 750 W or 1.5 kW motor (Note 1) |

Chapter 2 Robot Components of Extended-Joint Support System

| No. | Components | Parts No. | Remarks |
|------|---|-------------|------------------------------------|
| 12 | Extended-joint cable set | 410149-0700 | Including Nos. 12-1 and 12-2. |
| 12-1 | Extended-joint encoder cable (Enforced, 3 m) | 410141-2230 | |
| 12-2 | Extended-joint motor cable (Enforced, 3 m) | 410141-2170 | For 100W, 200W or 400W motor |
| 13 | Extended-joint cable set | 410149-0710 | Including Nos. 13-1 and 13-2. |
| 13-1 | Extended-joint encoder cable (Enforced, 6 m) | 410141-2230 | |
| 13-2 | Extended-joint motor cable (Enforced, 6 m) | 410141-2190 | For 750 W or 1.5 kW motor (Note 1) |
| 14 | Extended-joint cable set | 410149-0720 | Including Nos. 14-1 and 14-2. |
| 14-1 | Extended-joint encoder cable (Enforced, 6 m) | 410141-2240 | |
| 14-2 | Extended-joint motor cable (Enforced, 6 m) | 410141-2180 | For 100W, 200W or 400W motor |
| 15 | Extended-joint cable set | 410149-0730 | Including Nos. 15-1 and 15-2. |
| 15-1 | Extended-joint encoder cable (Enforced, 6 m) | 410141-2240 | |
| 15-2 | Extended-joint motor cable (Enforced, 6 m) | 410141-2200 | For 750 W or 1.5 kW motor (Note 1) |
| 16 | Extended-joint cable set | 410149-0860 | Including Nos. 16-1 and 16-2. |
| 16-1 | Extended-joint encoder cable (Enforced, 12 m) | 410141-2470 | |
| 16-2 | Extended-joint motor cable (Enforced, 12 m) | 410141-2490 | For 100W, 200W or 400W motor |
| 17 | Extended-joint cable set | 410149-0870 | Including Nos. 17-1 and 17-2. |
| 17-1 | Extended-joint encoder cable (Enforced, 12 m) | 410141-2470 | |
| 17-2 | Extended-joint motor cable (Enforced, 12 m) | 410141-2510 | For 750 W or 1.5 kW motor (Note 1) |

(Note 1) Due to the restrictions on the controller output rating, the robot controller uses 1.5 kW extended-joint motors as 900W-equivalent motors.

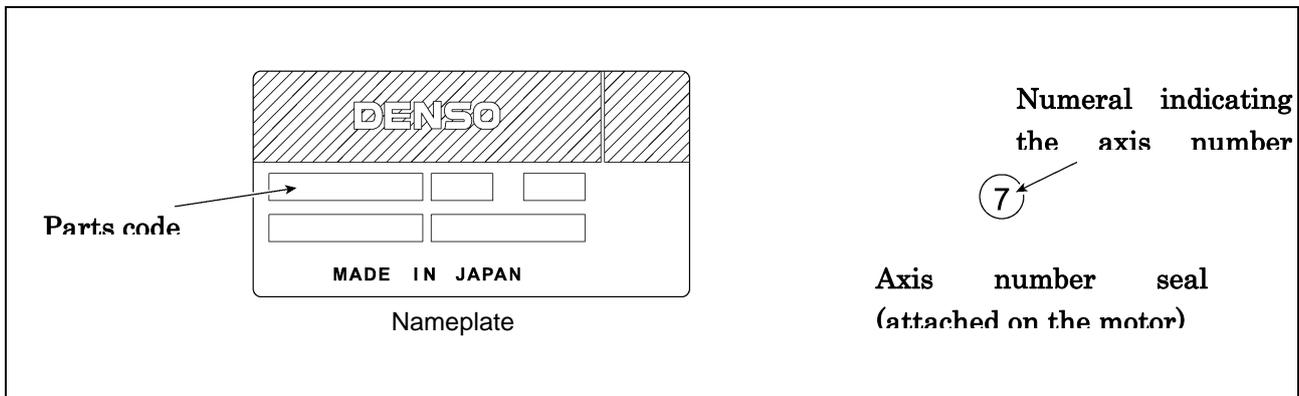
(Note 2) To connect the robot unit, use the motor/encoder cable that comes with the robot unit.

2.4 Detailed Specifications of Extended-Joint Related Options

2.4.1 Extended-Joint Motors

The following five types of servomotors are available as options for the extended-joint use.

On those servomotors are nameplates and axis number seals as illustrated below. Before using those motors, be sure to check the parts codes and axis numbers and connect motors to correct axes specified on axis number seals.



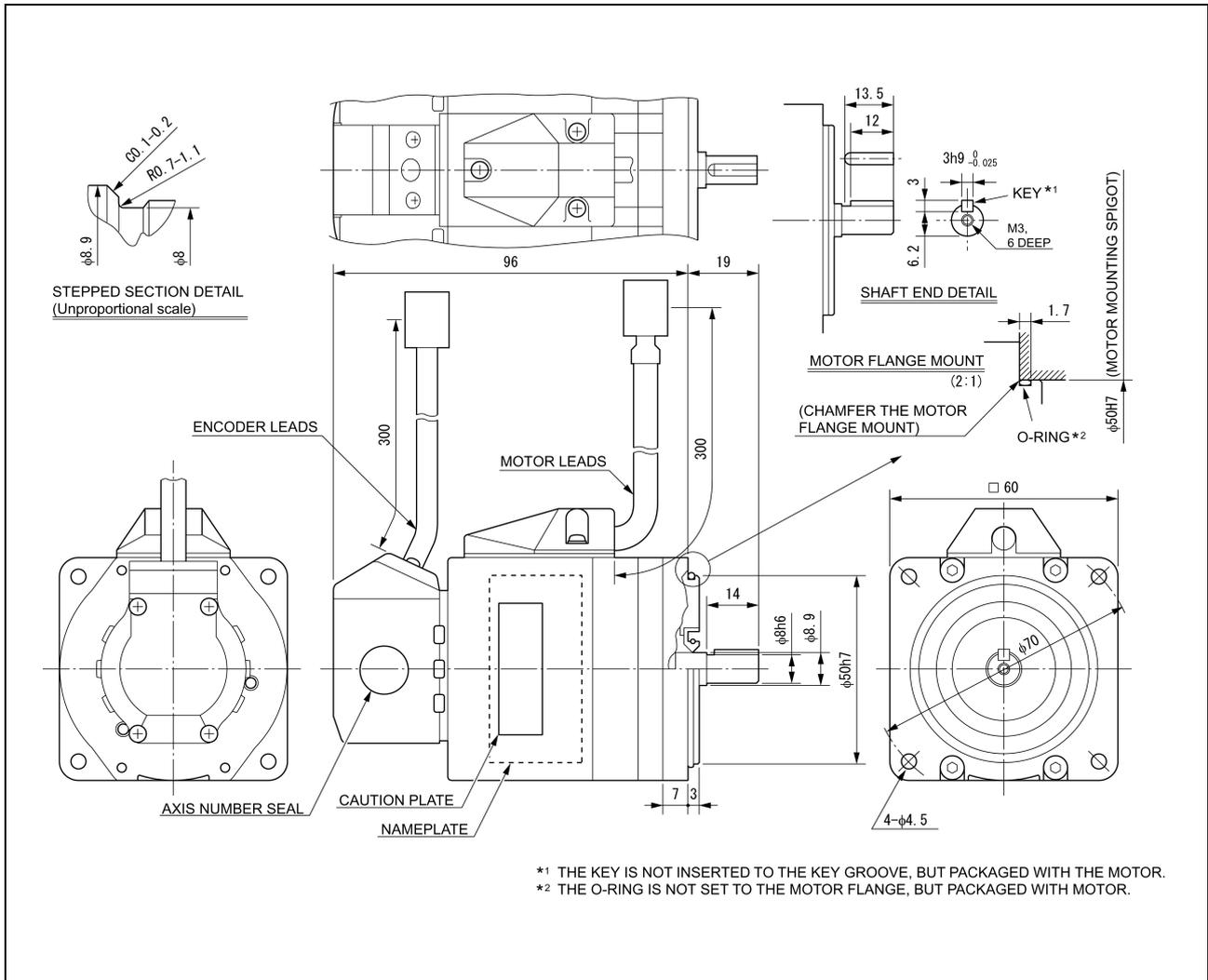
| Parts code | Rated output power (W) | Rated revolutions (rpm) | With/without brake | Model |
|-------------|------------------------|-------------------------|--------------------|-------------|
| 410622-1510 | 100 | 3,000 | With | MQMA012T2V2 |
| 410622-1520 | 200 | 3,000 | With | MQMA022T3V2 |
| 410622-1530 | 400 | 3,000 | With | MQMA042T3V2 |
| 410622-1540 | 750 | 3,000 | With | MQM082T2V2 |
| 410622-1550 | 1500 (Note 1) | 3,000 | Without | MQM152T2U2 |

(Note 1) Due to the restrictions on the controller output rating, the robot controller uses 1.5 kW extended-joint motors as 900W-equivalent motors.

2.4.2 External Dimensions of Extended-Joint Motors

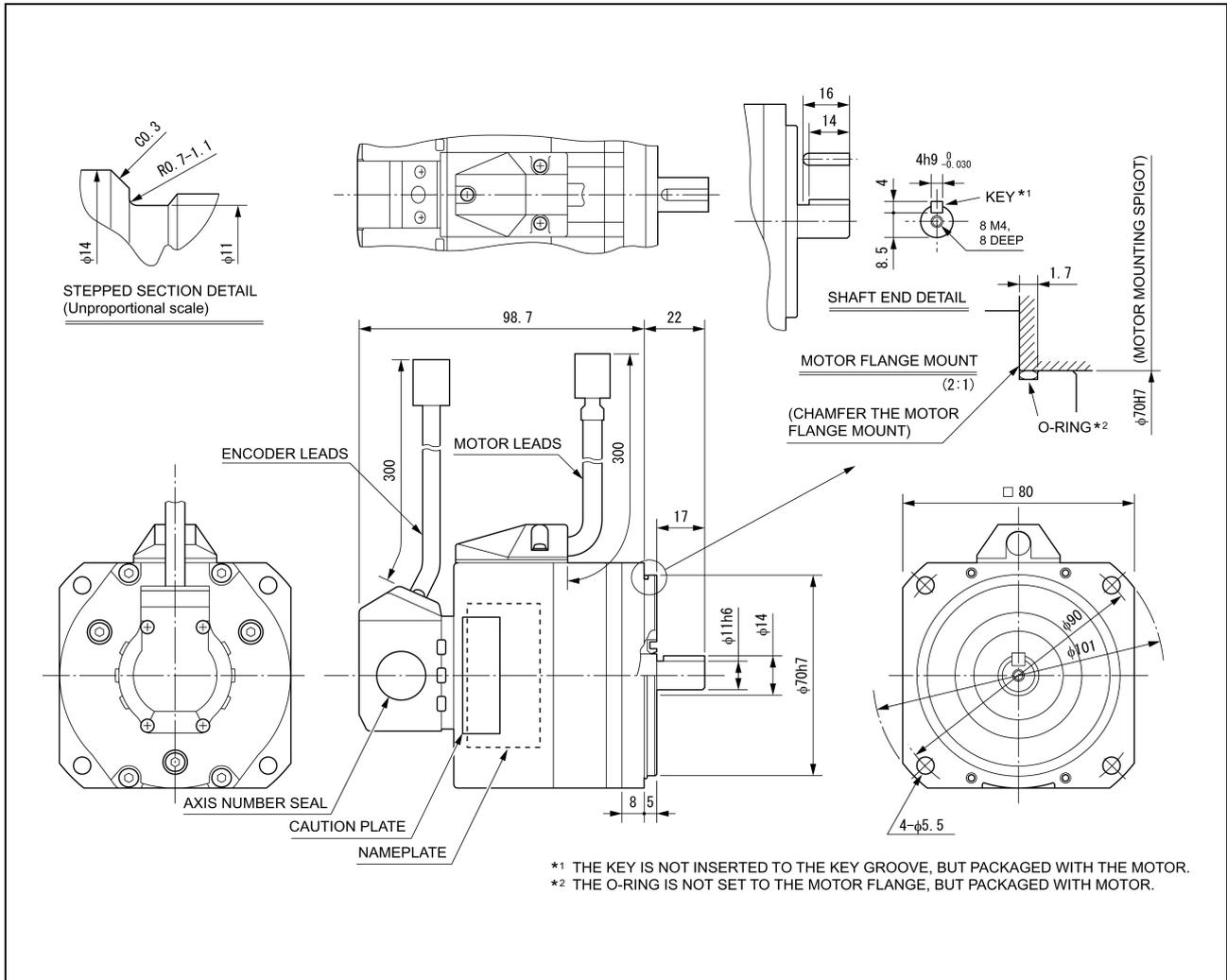
■ 410622-1510 (100W, with brake)

Model: MQMA012T2V2



■ 410622-1520 (200W, with brake)

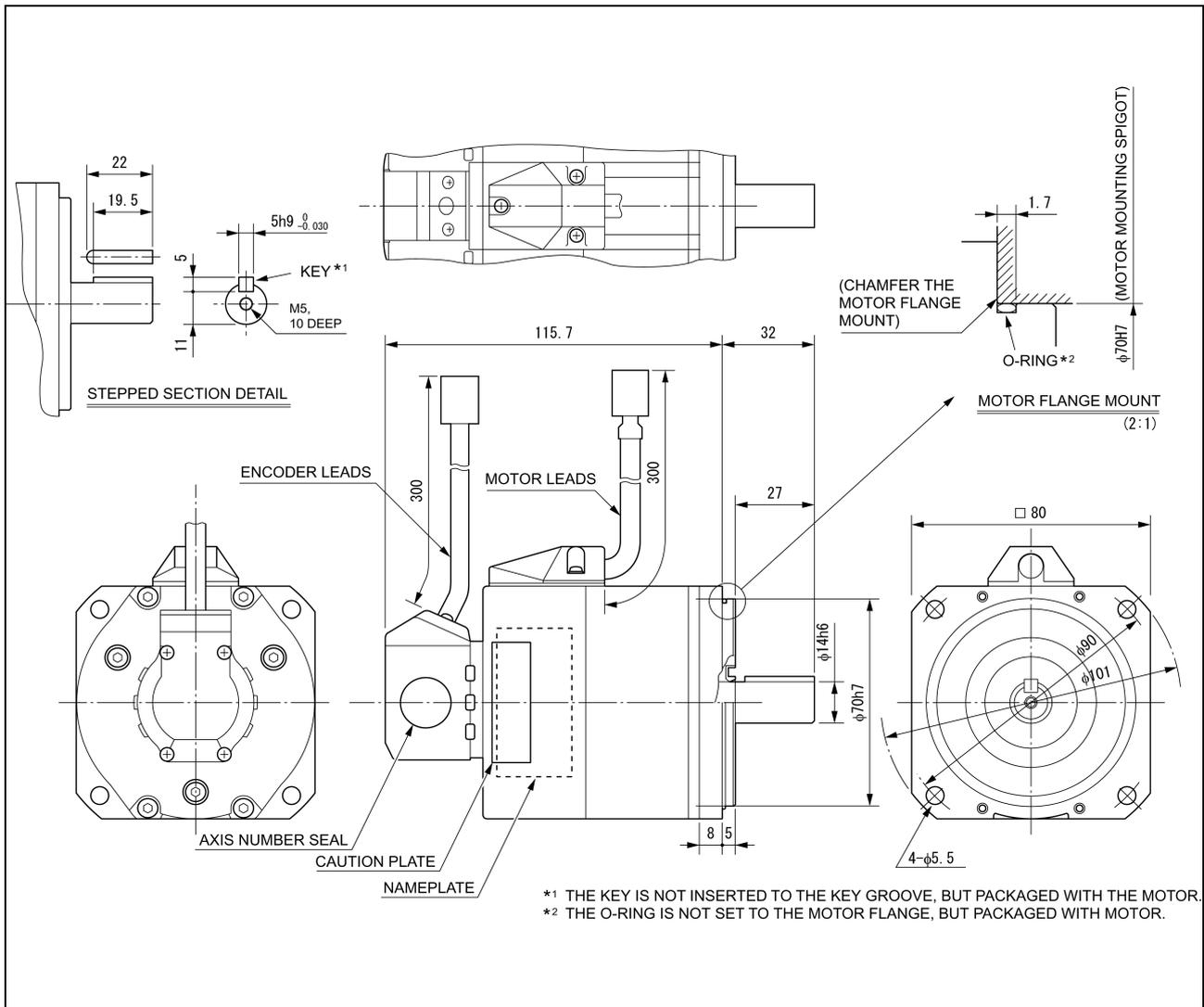
Model: MQMA022T3V2



Chapter 2 Robot Components of Extended-Joint Support System

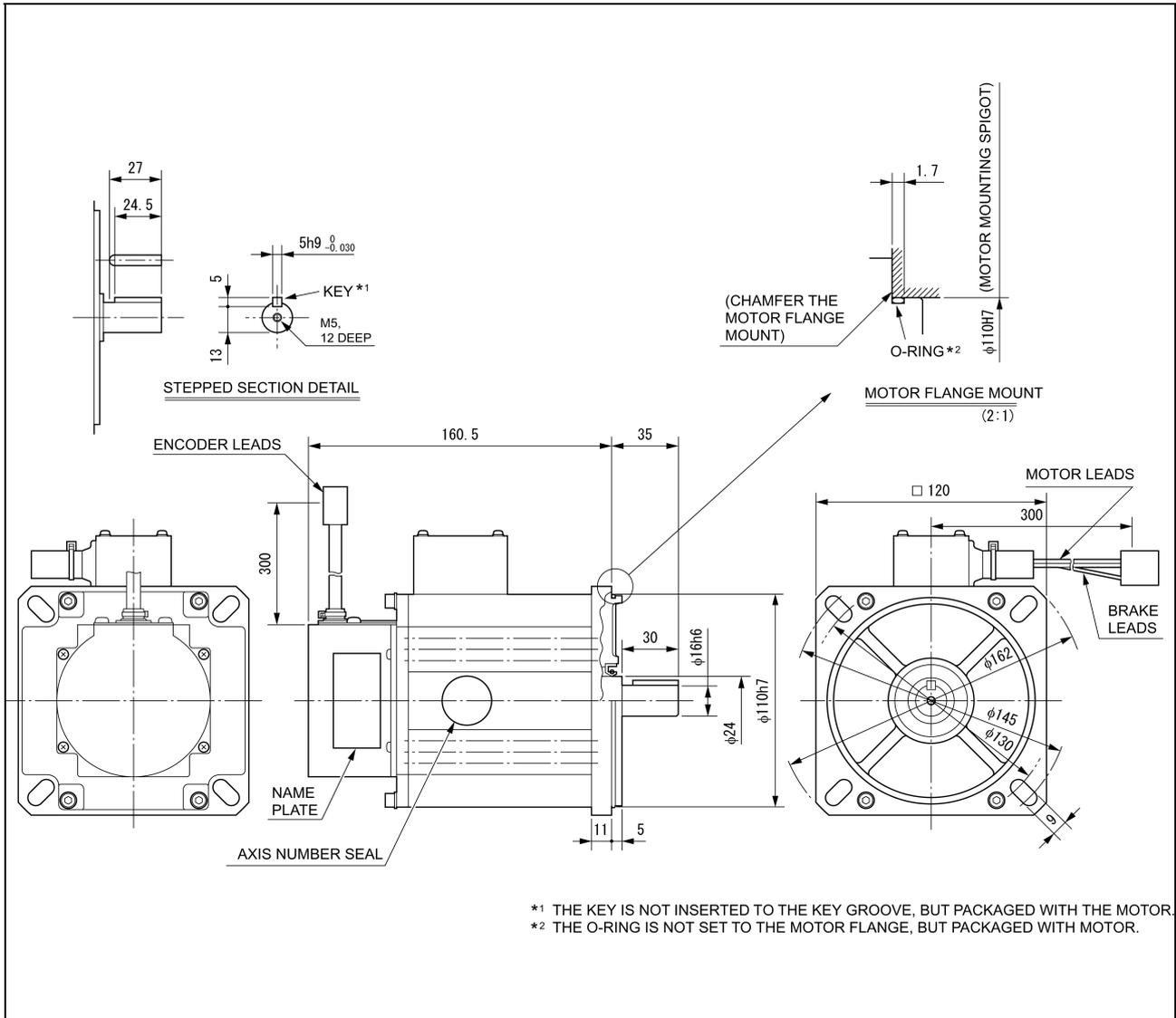
■ 410622-1530 (400W, with brake)

Model: MQMA042T3V2



■ 410622-1540 (750 W, with brake)

Model: MQM082T2V2

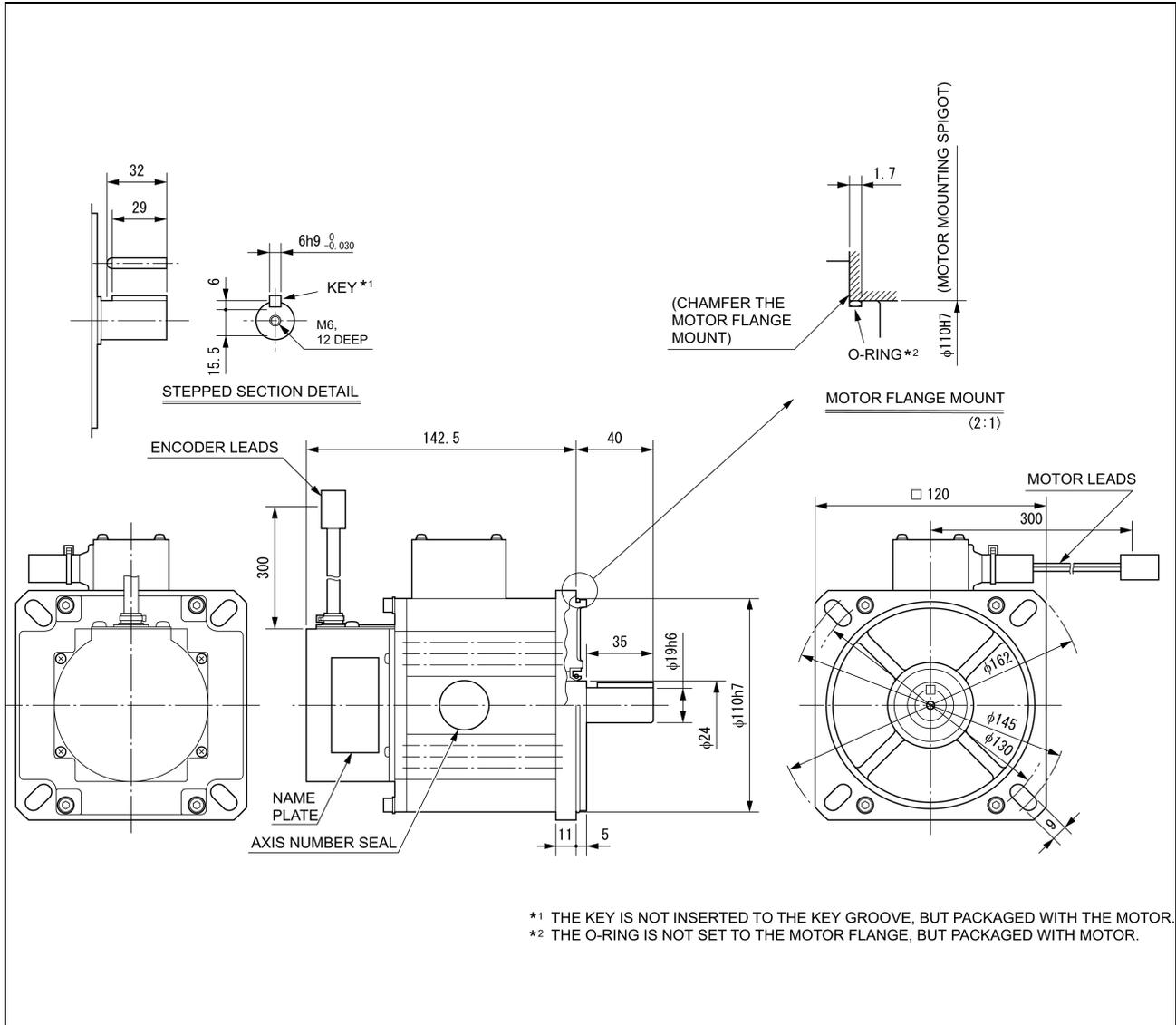


Chapter 2 Robot Components of Extended-Joint Support System

■ 410622-1550 (1.5 kW, without brake)

Model: MQM152T2U2

Note: Due to the restrictions on the controller output rating, the robot controller uses 1.5 kW extended-joint motors as 900W-equivalent motors.



2.4.3 Specifications of Extended-Joint Motors

(1) General specifications

| Items (Unit) | | 410622-1510 (100W, with brake) | 410622-1520 (200W, with brake) | 410622-1530 (400W, with brake) | 410622-1540 (750W, with brake) | 410622-1550 (1.5 kW, without brake) (Note 1) | | |
|--|--|---|--|--------------------------------------|--------------------------------------|---|-----|-----|
| Rated output (W) | | 100 | 200 | 400 | 750 | 900 (Note 1) | | |
| Rated rotation speed (rpm) | | 3000 | 3000 | 3000 | 3000 | 3000 | | |
| Maximum rotation speed (rpm) | | 5000 | 5000 | 5000 | 4500 | 4500 | | |
| Rated output torque (N*m) | | 0.32 | 0.64 | 1.3 | 2.4 | 2.9 (Note 1) | | |
| Maximum peak output torque (N*m) | | 0.95 | 1.91 | 3.82 | 6.9 | 9.5 (Note 1) | | |
| Rated input current (A) rms, typical | | 1.0 | 1.6 | 2.5 | 4.3 | 5.6 (Note 1) | | |
| Rotor Inertia ($\times 10^{-4}$ kg*m ²) | | 0.12 | 0.42 | 0.72 | 2.85 | 2.82 | | |
| Maximum rush current (A) (0 to peak) typical | | 4.3 | 6.8 | 10.5 | 18.9 | 43.4 | | |
| Allowable stress or load (N) | In assembly or installation (Note 2) | Radial stress | 147 | 392 | 392 | 686 | 686 | |
| | | Thrust stress | A-direction | 88 | 147 | 147 | 294 | 294 |
| | | | B-direction | 117.6 | 196 | 196 | 392 | 392 |
| | In Operation | Radial load | 98 | 284.2 | 303.8 | 441 | 392 | |
| | | Thrust load | 58.8 | 98 | 98 | 147 | 147 | |
| Mass (kg) | | 0.90 | 1.6 | 2.2 | 5.4 | 4.8 | | |
| Recommended load-inertia moment ratio | | 30 times or less of the servomotor inertia moment (When the system requires higher response, decrease the load inertia moment) | | | | | | |
| Environmental conditions | Heat resistance (Allowable temperature) | | Operation: 0 to +40°C Storage: -20°C to +80°C | | | | | |
| | Humidity resistance (Allowable relative humidity) | | 90% or less RH (Without condensation) | | | | | |
| | Vibration resistance | | 49m/s ² max. (8 hours, 20 Hz to 3 kHz sweep in each of X, Y, and Z axes. 24.5 m/s ² max. when the motor is stopped.) 24.5 m/s ² max. at the resonance frequency (10,000,000 times in each of X, Y, and Z axes) | | | | | |
| | Installation orientation | | Motor can be installed in either the horizontal or vertical direction. | | | | | |

Chapter 2 Robot Components of Extended-Joint Support System

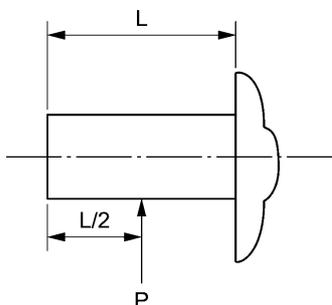
| Items (Unit) | | 410622-1510 (100W, with brake) | 410622-1520 (200W, with brake) | 410622-1530 (400W, with brake) | 410622-1540 (750W, with brake) | 410622-1550 (1.5 kW, without brake) (Note 1) |
|---------------------------------------|---|--------------------------------------|--------------------------------------|--------------------------------------|--------------------------------------|---|
| Specification of brake (Note 3) | Static friction torque (N*m) | 0.29 min. | 0.64 min. | 1.27 min. | 2.45 min. | Without brake |
| | Armature pull time (Typical) (ms) | 50 max. | 50 max. | 50 max. | 50 max. | |
| | Armature release time (Typical) (ms) (Note 4) | 15 max. | 20 max. | 20 max. | 20 max. | |
| | Release voltage (VDC) | 1 min. | 1 min. | 1 min. | 1 min. | |
| | Rated exciting voltage (VDC) | 24 ±2.4 | 24 ±2.4 | 24 ±2.4 | 24 ±2.4 | |
| | Rated exciting current (Amp. DC) (Cold start at 24 VDC typical) | 0.29 max. | 0.23 max. | 0.27 max. | 0.86 max. | |
| | Allowable work energy for every braking (J) (Typical value) | 137 | 196 | 196 | 392 | |
| | Allowable gross work load (J) (Typical) | 44.1 × 10 ³ | 147 × 10 ³ | 147 × 10 ³ | 490 × 10 ³ | |

(Note 1) Due to the restrictions on the controller output rating, the robot controller uses 1.5 kW extended-joint motors as 900W-equivalent motors.

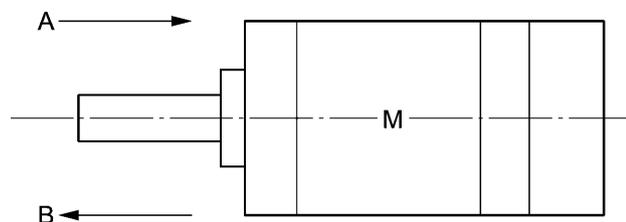
(The values of each item in the above table are equivalent to those of the 900W servomotor.)

(Note 2) The radial stress (P) position and thrust stress direction in assembling or installing of the motor are shown below.

Radial stress (P) position



Thrust stress direction



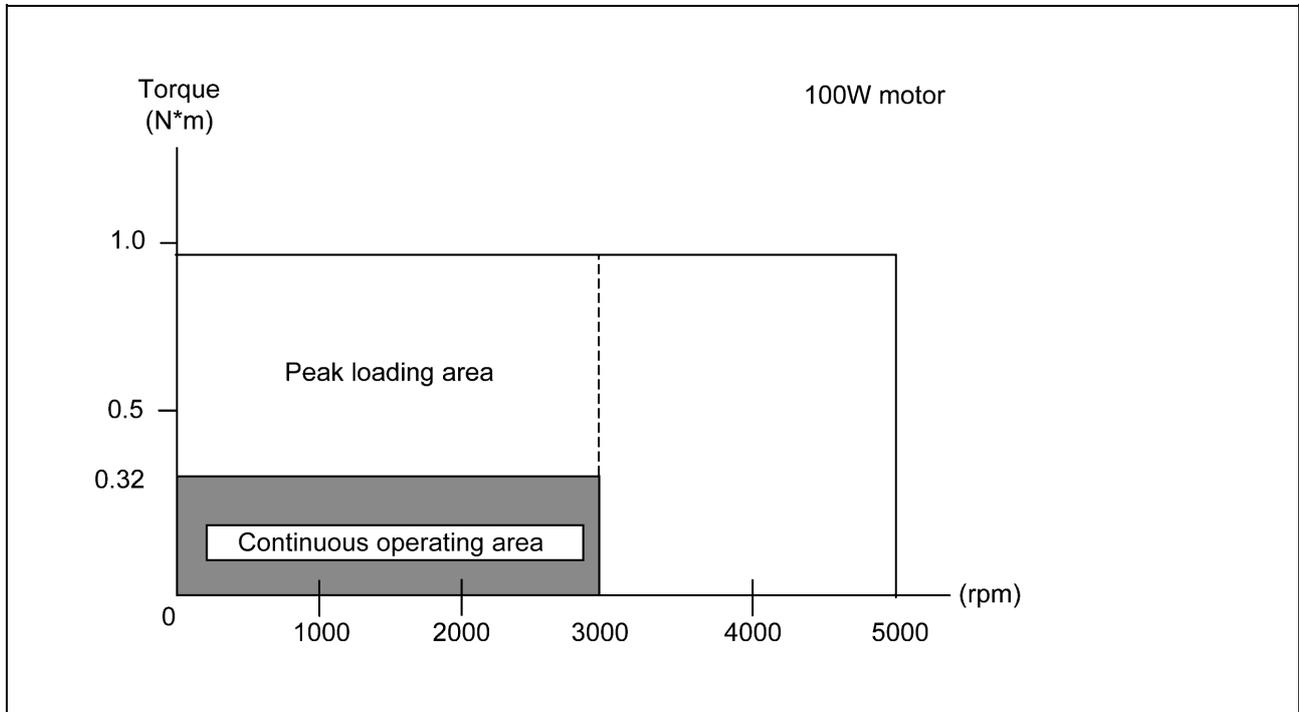
(Note 3) The brake backlash angle at the factory shipment is within $\pm 1^\circ$.

(Note 4) The armature release time refers to the time length from when the braking current is cut off until no DC current flows across the surge absorber coil used as dummy load.

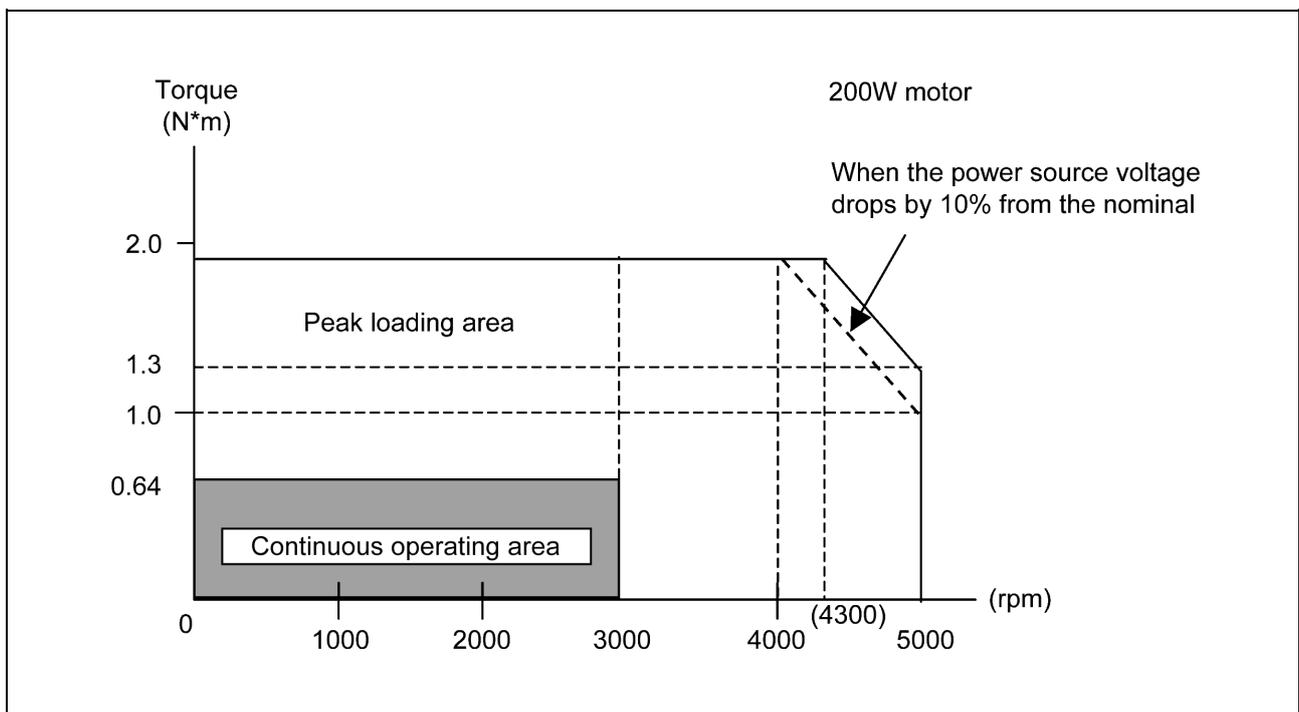
**(2) S vs. T characteristic curves of extended-joint motors
(Revolution speed vs. Torque characteristic)**

The torque characteristics of extended-joint motors are shown below in typical values.

■ 410622-1510 (100W, with brake)

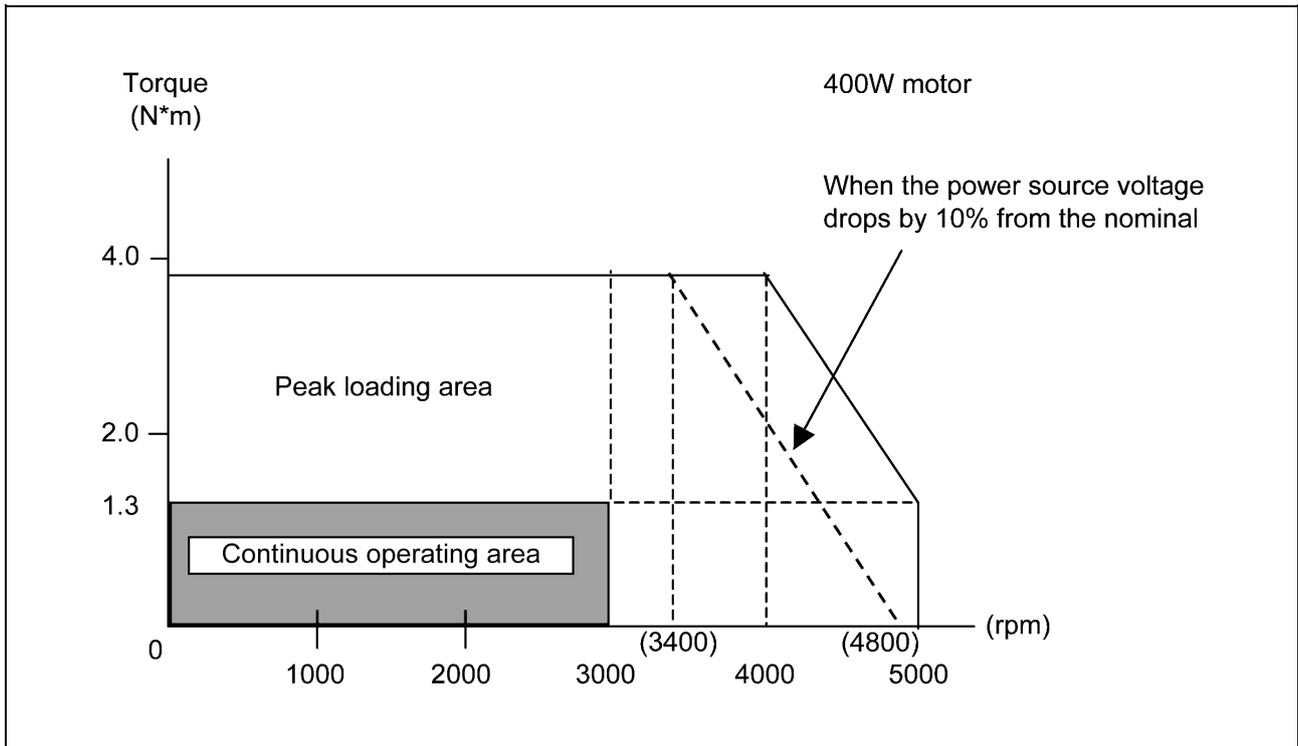


■ 410622-1520 (200W, with brake)

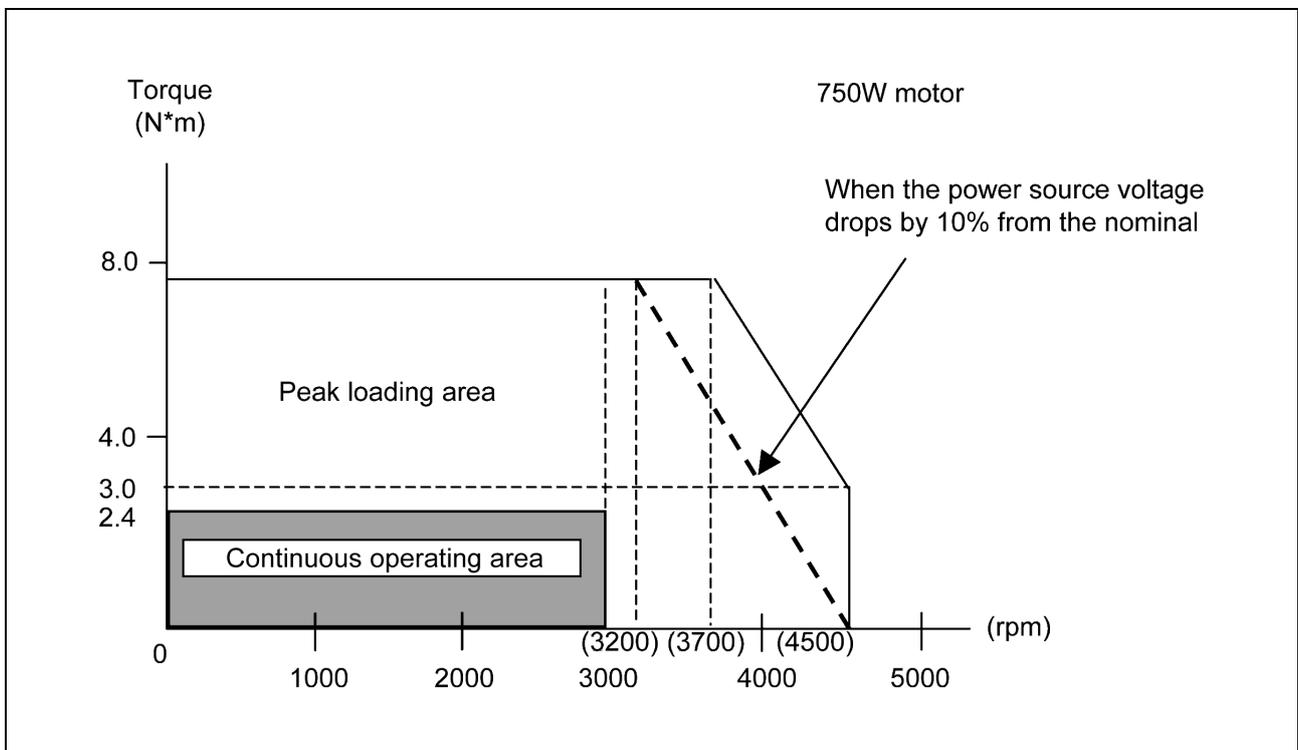


Chapter 2 Robot Components of Extended-Joint Support System

■ 410622-1530 (400W, with brake)

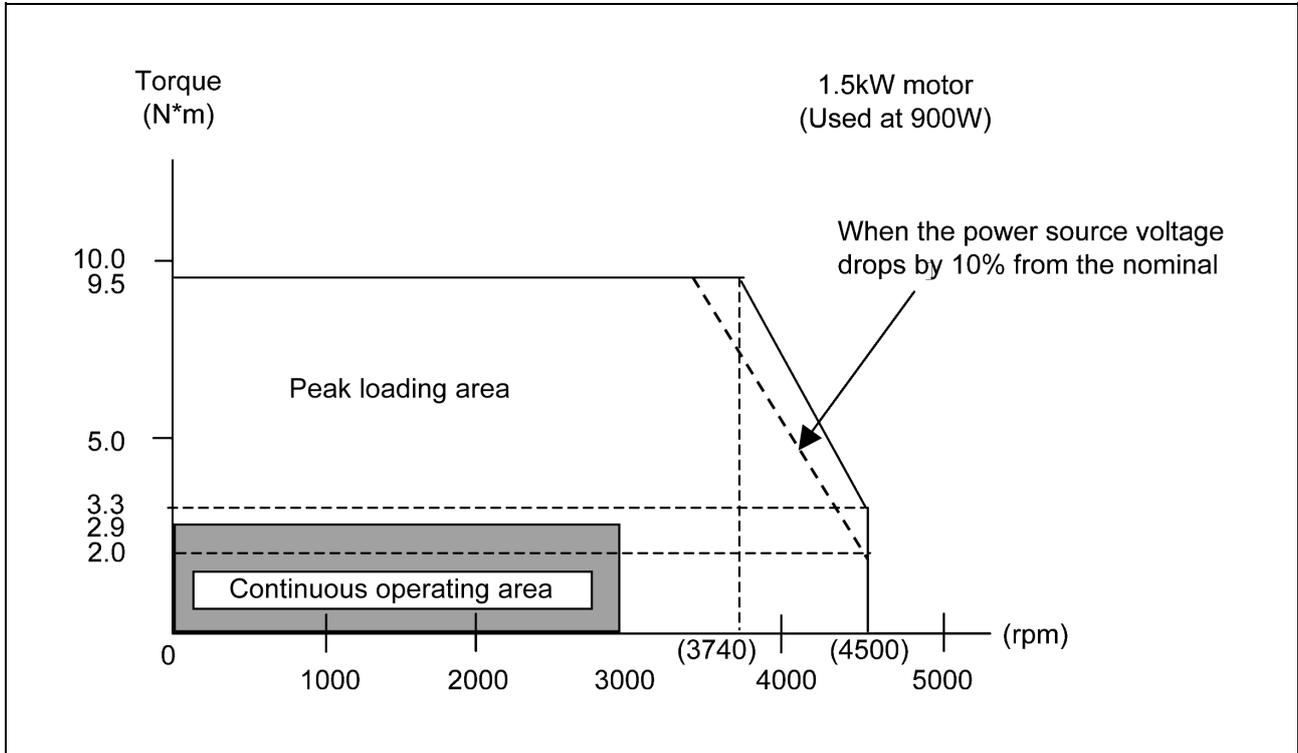


■ 410622-1540 (750W, with brake)



■ 410622-1550 (1.5 kW, without brake)

Note: Due to the restrictions on the controller output rating, the robot controller uses 1.5 kW extended-joint motors as 900W-equivalent motors.



2.4.4 Safety Rules for Using Extended-Joint Motors

Note: This section gives important points only for handling extended-joint motors. For the general precautions for handling robots, refer to manuals prepared for the robot series.

[1] Motor handling notes

- (1) When moving, wiring, maintaining motors, confirm that the controller power has been shut down.
- (2) Take care not to damage or crush motor cables or let them undergo much stress or heavy load.
- (3) When a motor is running, never reach out towards the rotating part.
- (4) Do not touch a motor and its peripheral devices when the motor power is on and for a while after it is cut off, since they are HOT.
- (5) When carrying a motor, hold the motor body not the cables, shafts or encoders.
- (6) Do not put heavy materials on motors or hang them onto motors.
- (7) Avoid letting motors undergo strong shock or impact.
- (8) If an earthquake or any other disaster happens, confirm the installation conditions of motors and machines for safety before starting motors.
- (9) Do not disassemble or modify motors.

[2] Precautions for installing motors

- (1) Install specified extended-joint motors. Do not install motors other than specified ones.
- (2) The robot controller and extended-joint motors will operate in combination. According to the specified capacity, select the proper motors.
- (3) When hooking up motors to an extended-joint support robot controller, be sure to check the motor models and joint numbers which are labeled on motors.
- (4) When routing motor cables, observe the following to prevent them from undergoing stress:
 - Protect the cable clamps and connector pins from excessive stress due to bending or cable weight.
 - When installing extended-joint motors on moving mounts, be sure to fasten their cables to those mounts at the connector ends and route the relay cables through cable bearers to minimize excessive bending stress.
 - Make the radius of each cable loop as large as possible.

-
- (5) Observe the following notes relating to the allowable loads of motor output shafts:
- Design mechanisms so that the radial or thrust load that will apply to motor shafts at installation or in operation will be kept within the allowable range specified for each motor model.
 - To keep radial load that would be caused by small misalignment between shafts within the allowable range, use flexible couplings exclusively designed for servomotors and as rigid as possible.
 - If you want to use rigid couplings, install them with extra care. If applied in installation, excessive bending load will break motor shafts or shorten the service life of bearings.
- (6) When installing or removing a coupling to/from the end of a motor shaft, do not apply any direct impact onto the motor shaft with a hammer. An encoder fitted on the other end of the shaft will be damaged.
- Align two shafts with each other sufficiently. Misalignment will produce vibration, resulting in damaged bearings.
- (7) Never use motors in an environment where they may be subjected to water splash, corrosive gases, or inflammable gases. Do not use them in the vicinity of the inflammable. (Extended-joint motors have not been designed to withstand explosions, dust-proof, nor are they splash-proof.)
- (8) Be sure to install and set up motors so that they will never cause a fire or bodily injury even if an earthquake or any other disaster occurs.
- (9) The shaft ends of extended-joint motors are lubricated with grease (SHELL Albania No. 2). Take into account that grease would influence plastics.

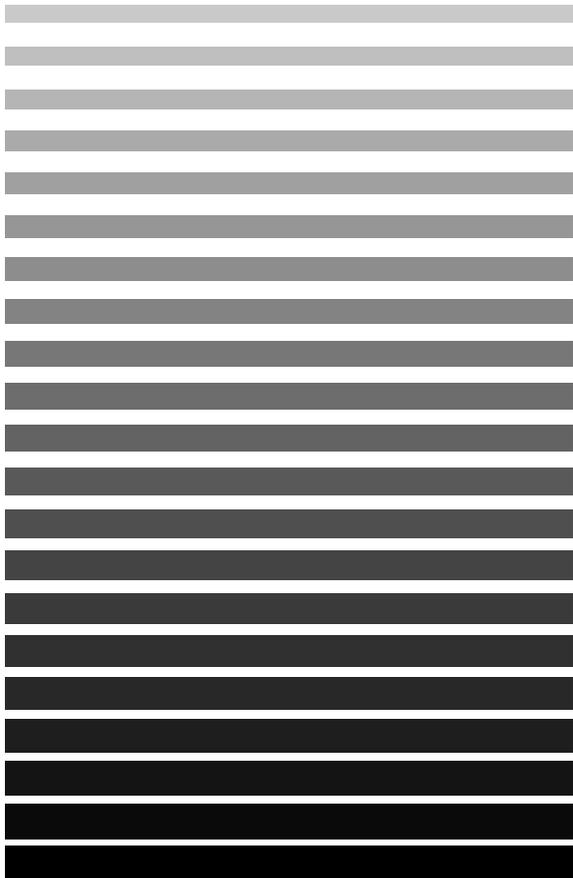
[3] Motor storage notes

- (1) Do not store motors in an environment where they may be subjected to rain, water splash, harmful gases or liquids.
- (2) Shield motors from direct sunlight and store them inside the specified temperature/humidity range (-20°C to 80°C, RH 90% or less, without dew condensation).

Chapter 3



Extended-Joint Related Function



This chapter covers new features relating to the extended-joints.

Chapter 3 Extended-Joint Related Function

3.1 Extended-Joint Function

The extended-joint function allows you to control extra joints independently of robot joints through the standard interface of the robot controller (NetwoRC).

To support the extended-joint function, Main Software version 1.5 or later enhances the following functions:

| Added functions | Extended functions |
|--|--|
| <ul style="list-style-type: none">• Operating extended-joints manually• Getting extended-joint positions• Operating extended-joints by specifying a desired variable• Operating extended-joints by program (Several commands are extended and added.)• Setting the boundless rotation of extended-joints | <ul style="list-style-type: none">• Command execution from the specified line• Continue function• Safe start (SS) function |

NOTE: Before using these functions, you need to set various parameters of extended-joints properly. Refer to Section 3.2 "Setting the Extended-Joint Parameters."

3.1.1 Operating Procedures of the Extended-Joint Function

This section describes the operating procedures of the extended-joint function:

- [1] Operating extended-joints manually
- [2] Getting extended-joint positions
- [3] Operating extended-joints by specifying a desired variable
- [4] Operating extended-joints by program
- [5] Setting the boundless rotation of extended-joints
- [6] Command execution from the specified line
- [7] Continue function (for extended-joints)
- [8] Safe start (SS) Function (for extended-joints)

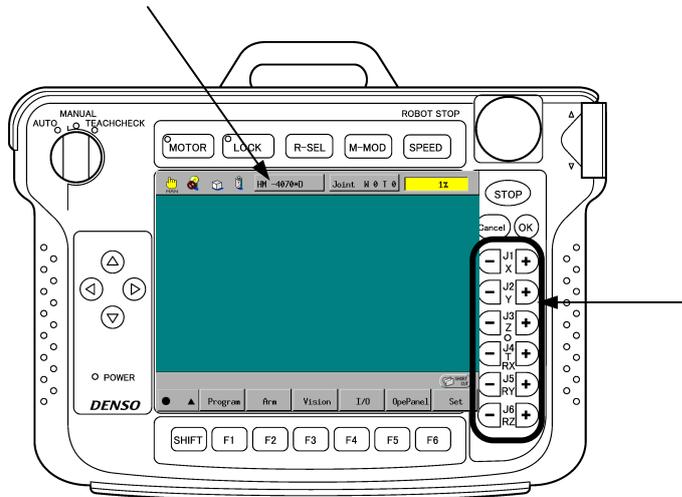
[1] Operating extended-joints manually

This subsection describes items relating to the extended-joints. For other details, refer to the SETTING-UP MANUAL, Section 3.2 "Manual Mode."

Extended-joints may operate only in Joint mode, independently of each other.

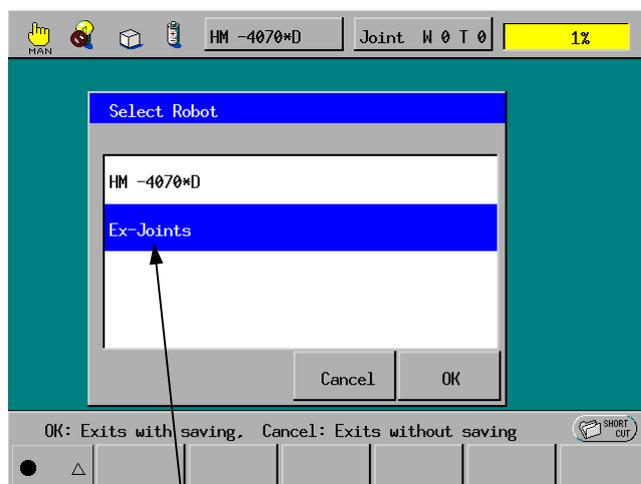
■ From the teach pendant

- Step 1** Set the mode switch to the MANUAL position and press the MOTOR key to turn the motor on.
- Step 2** On the top screen, press the robot select button (or R-SEL key).



The Select Robot window will display.

- Step 3** Choose the "Ex-Joints" and then press the OK button.



Extended-joints

If you change the robot selection as shown at left, assignment of joints to the arm traverse keys in manual mode operation will be switched.
(Choosing the "Ex-Joints" will switch the joint assignment made for the Arm traverse keys from robot joints to extended-joints.)

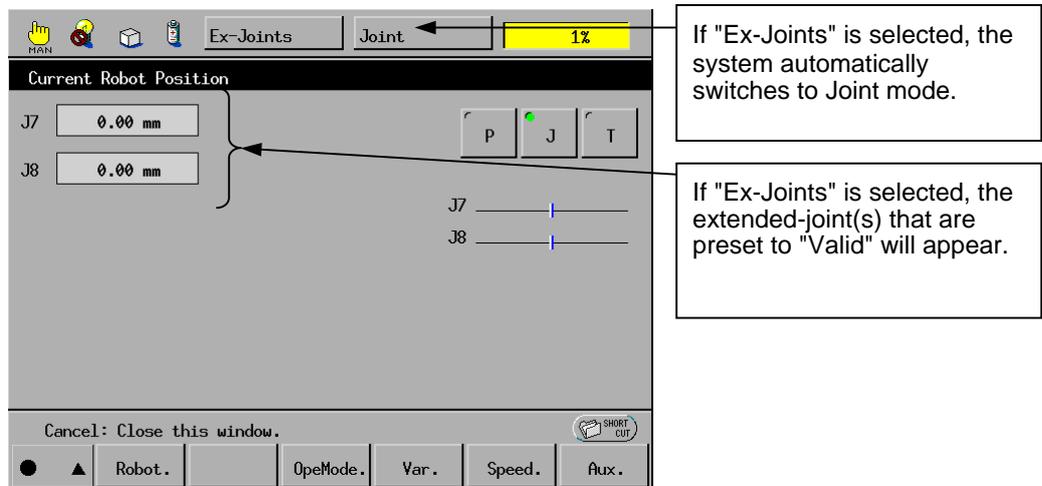
Note: Extended-joints may operate only in Joint mode. Choosing the Ex-Joints in the above window will automatically switch to Joint mode.

NOTE: In the Select Robot window, the "Ex-Joints" will appear only when any extended-joint is set to "Valid" in the [Servo Parameters for Ex-Joint] window. If no "Ex-Joints" appears, access [F2 Arm]—[F12 Maint.]—[F7 ExJoints]—[F8 ExServo] and then enable the desired extended-joint.

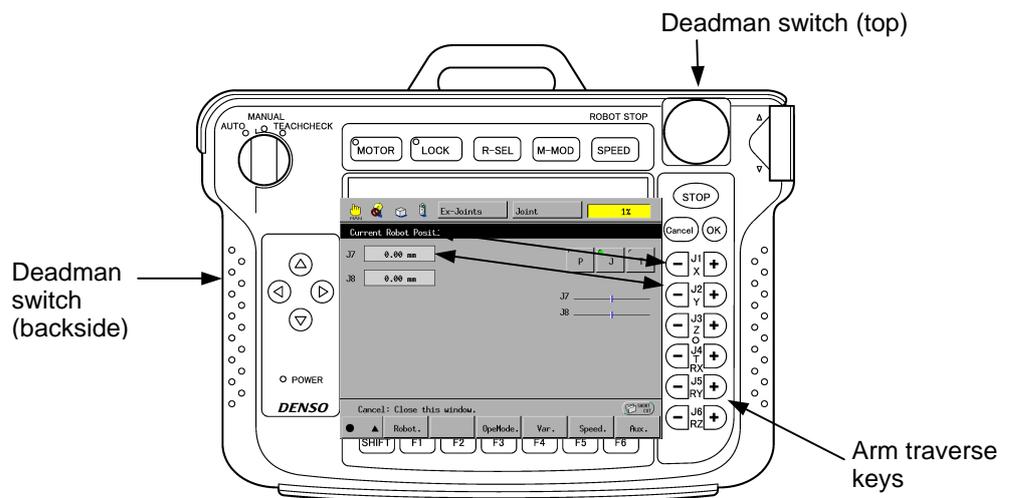
For more details, refer to Section 3.2 "Setting the Extended-Joint Parameters."

Step 4 On the top screen, press [F2 Arm] to display the Current Robot Position window as shown below.

Note: Without displaying this window, you may operate the extended-joints manually.



Step 5 While holding down the deadman switch, press either of the arm traverse keys that are assigned the extended-joint you want to move.

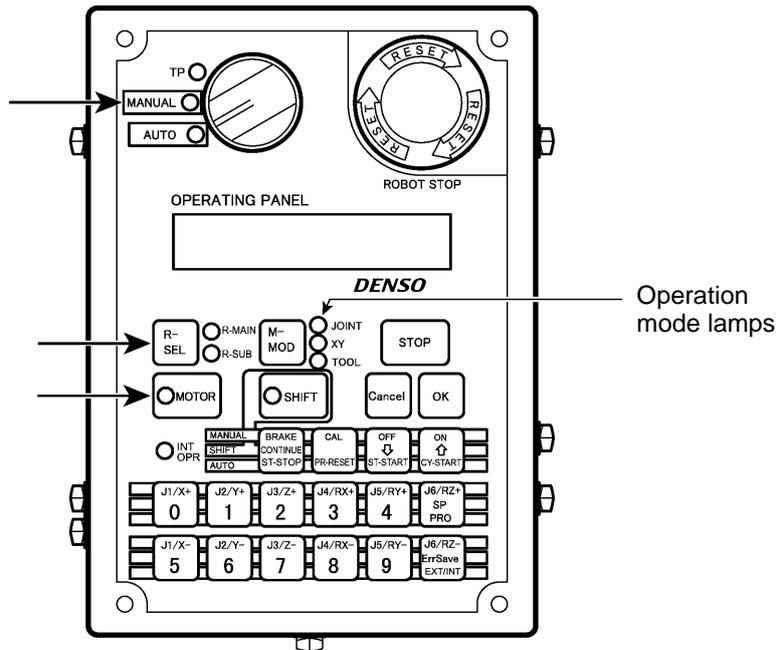


Note: Assignment of joints to the arm traverse keys is arranged from the top line first, as shown above. In this figure, you may move the 7th joint by using the [+ J1 X] and [- J1 X] keys.

■ **From the operating panel**

Step 1 Set the mode switch to the MANUAL position and press the MOTOR key. Make sure that the MOTOR lamp comes on.

Step 2 Press the R-SEL key to select the R-SUB. Only when the R-SUB lamp is on, the extended-joints may be driven from the operating panel.



NOTE: The extended-joints work only in Joint mode. Choosing the R-SUB in the above window will automatically switch to Joint mode.

Step 3 While holding down the deadman switch, press either of the arm traverse keys that are assigned the extended-joint you want to move.

NOTE: Assignment of joints to the arm traverse keys on the operating panel is the same as that on the teach pendant.

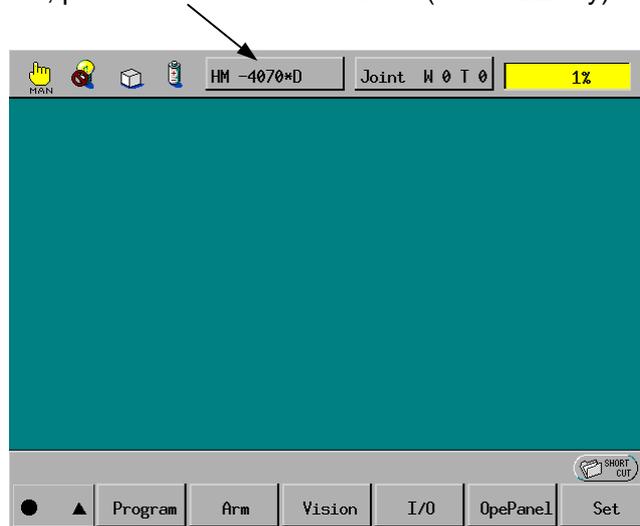
(Example) If the 7th and 8th joints are set to "Valid" in the [Servo Parameters for Ex-Joint] window (and other joints are not) and the R-SUB is selected with the R-SEL key, then you may move the 7th and 8th joints by using the [+ J1 X][-J1 X] and [+ J2 X][- J2 X], respectively.

[2] Getting extended-joint positions

Get the positions of extended-joints into floating-point variables, joint by joint.

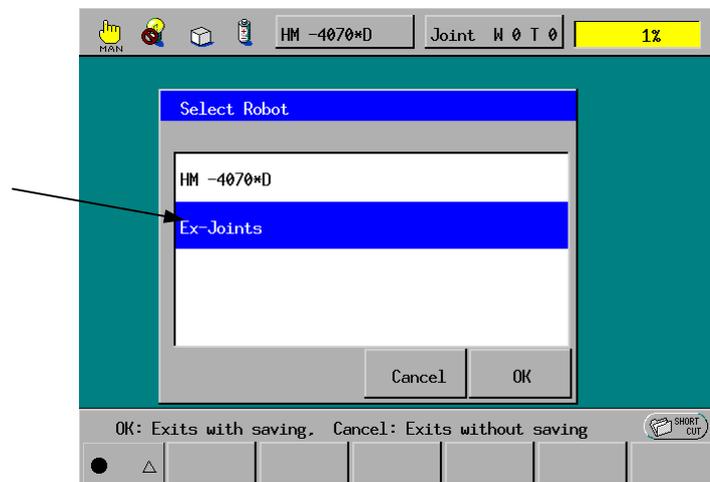
■ **From the teach pendant**

Step 1 On the top screen, press the robot select button (or R-SEL key).



The Select Robot window will display.

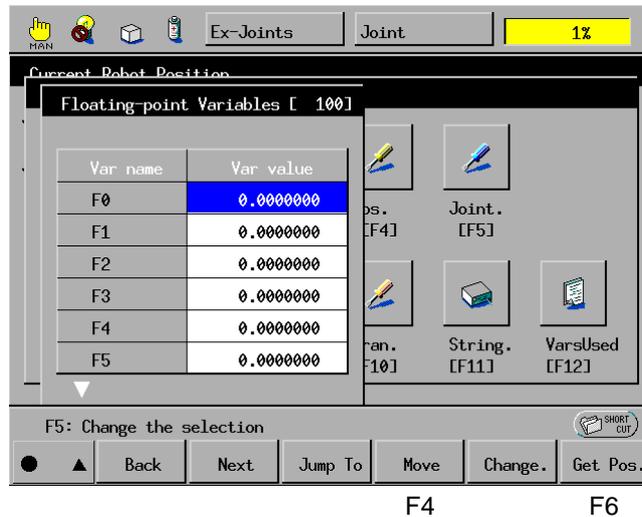
Step 2 Choose the "Ex-Joints" and then press the OK button.



NOTE: In the Select Robot window, the "Ex-Joints" will appear only when any extended-joint is set to "Valid" in the [Servo Parameters for Ex-Joint] window. If no "Ex-Joints" appears, access [F2 Arm]—[F12 Maint.]—[F7 ExJoints]—[F8 ExServo] and then enable the desired extended-joint.

For more details, refer to Section 3.2 "Setting the Extended-Joint Parameters."

- Step 3** From the top screen of the teach pendant, access [F2 Arm]—[F4 Var.]—[F2 Float.] to call up the Floating-point Variables window.
- Select a variable name to which you want to get a position value.

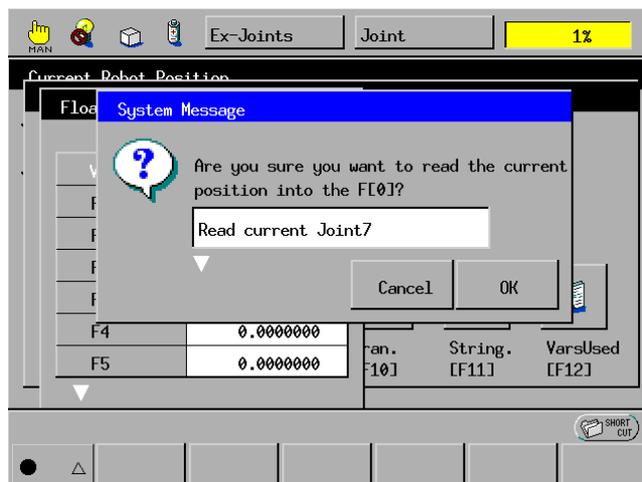


This example will get the extended-joint position value into variable F0.

NOTE: The [Move] and [Get Pos.] function buttons for the floating-point variables will display only when the "Ex-Joints" is selected in the Select Robot window.

- Step 4** Press [F6 Get Pos.].

The window below will appear. Select a joint whose position should be read in, and then press [OK].



This example will read in the position of the 7th joint.

NOTE: Only extended-joints that are set to "Valid" in the [Servo Parameters for Ex-Join] window may be read into floating-point variables. No robot joints can be read.

The above procedure reads the current position of the 7th joint into floating-point variable F0.

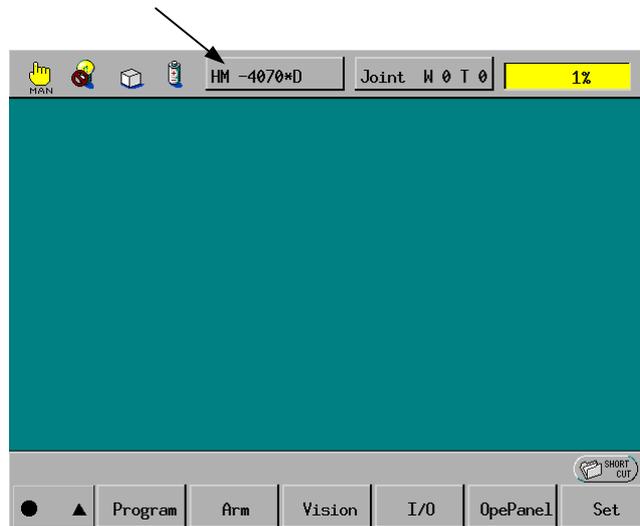
■ From the operating panel

- Step 1** Set the mode switch to the MANUAL position.
Confirm that the MANUAL lamp comes on.
- Step 2** Press the SHIFT key. Make sure that the SHIFT lamp comes on. Then press the BRAKE/CONTINUE/ST-STOP key.
- Step 3** Use the [↑] and [↓] keys to choose [Set CurPos F] on the LCD screen, then press the OK key.
- Step 4** Selecting a variable name to which you want to get a position value.
If "F?" appears on the LCD, enter the name of a floating-point variable by using the numeric keys and then press the OK key.
- Step 5** Selecting a joint whose position should be read in.
If "JointNo" appears on the LCD, enter the number of a floating-point variable by using the numeric keys and then press the [OK] key.
- NOTE:** Only extended-joints that are set to "Valid" in the [Servo Parameters for Ex-Join] window may be read into floating-point variables. No robot joints can be read.
- Step 6** Reading in the joint position.
If "SetCurPos" appears on the LCD, press the OK key. The "SetVarVal OK" appears on the LCD, showing that the position has been successfully read into the specified floating-point variable.

[3] Operating extended-joints by selecting a desired variable

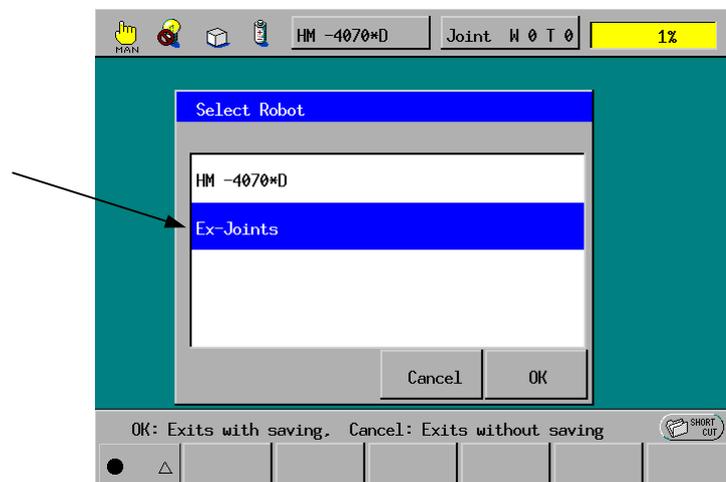
You can operate the extended-joints manually by specifying a desired floating-point variable for each joint.

Step 1 On the top screen, press the robot select button (or R-SEL key).



The Select Robot window will display.

Step 2 Choose the "Ex-Joints" and then press the OK button.

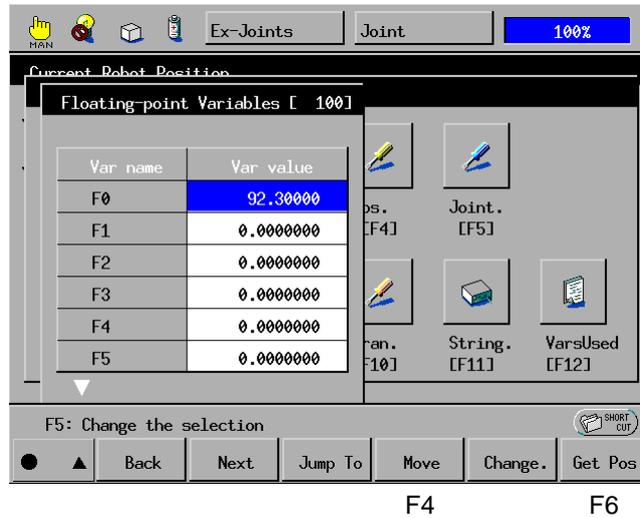


NOTE: In the Select Robot window, the "Ex-Joints" will appear only when any extended-joint is set to "Valid" in the [Servo Parameters for Ex-Joint] window.

If no "Ex-Joints" appears, access [F2 Arm]—[F12 Maint.]—[F7 ExJoints]—[F8 ExServo] and then enable the desired extended-joint.

For more details, refer to Section 3.2 "Setting the Extended-Joint Parameters."

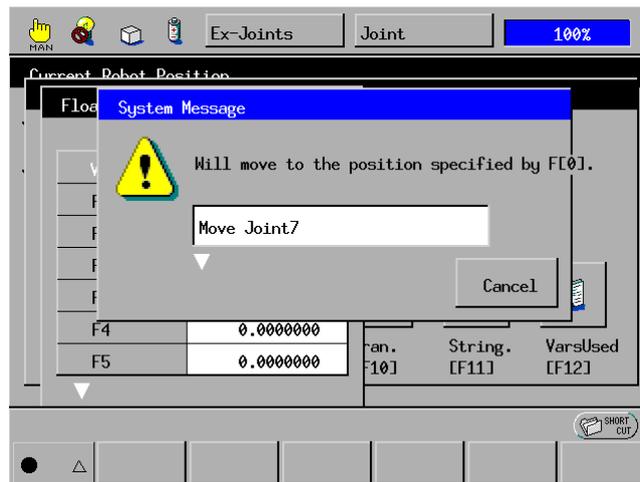
- Step 3** From the top screen of the teach pendant, access [F2 Arm]—[F4 Var.]—[F2 Float.] to call up the Floating-point Variables window. Select a variable having the desired movement value.



This example is to move an extended-joint to the position 92.30 assigned to floating-point variable F0.

NOTE: The [Move] and [Get Pos.] function buttons for the floating-point variables will display only when the "Ex-Joints" is selected in the Select Robot window.

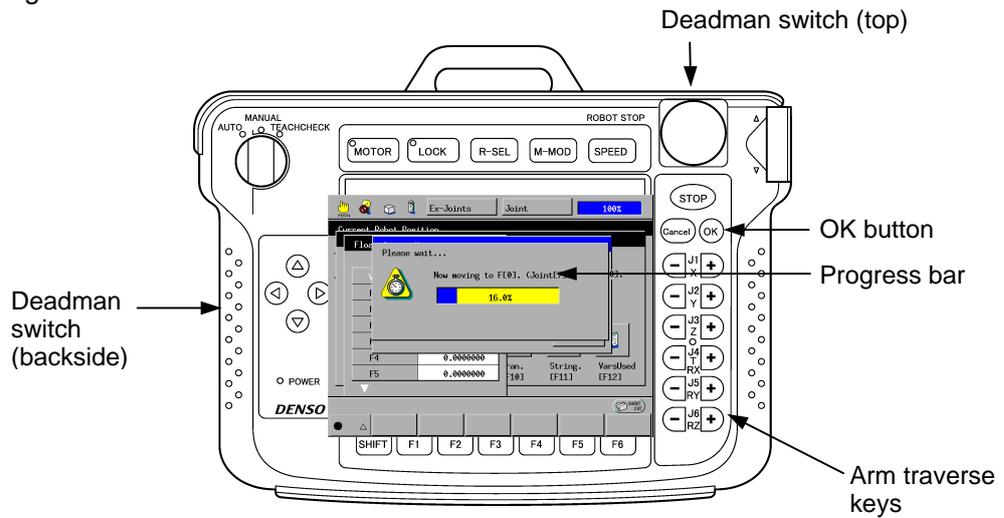
- Step 4** Press [F4 Move]. The window below will appear. Select an extended-joint you want to move by using the floating-point variable specified in Step 3.



This example is to move the 7th joint to the position 92.30 assigned to floating-point variable F0.

NOTE: Only extended-joints that are set to "Valid" in the [Servo Parameters for Ex-Join] window may be read into floating-point variables. No robot joints can be read.

Step 5 While holding down the deadman switch, keep on pressing the OK button until the progress bar shows 100%.



The above procedure moves the 7th joint to the position 92.30 assigned to floating-point variable F0.

Note: If an extended-joint(s) you want to move by using variables is contained in an arm group and any task holding its semaphore is active, then the extended-joint cannot be moved by using variables. For more details on the arm group semaphore, refer to the next subsection "[4] Operating extended-joints by program."

[4] Operating extended-joints by program

To operate extended-joints by program, you need to set up an arm group.

■ *Concept of an arm group*

"Arm Group" is a semaphore to control joints in a group as a single unit. Only when a task holds an "arm group" semaphore (herein called "arm group" and expressed as "Group n"), it can control robot joints and extended-joints. An arm group prevents more than one task from executing motion commands to a same joint at the same time.

Example of Arm Group Setting (For 4-axis robots)

| ArmGroup Settings | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|
| | J1 | J2 | J3 | J4 | J5 | J6 | J7 | J8 |
| Group 0 | ○ | ○ | ○ | ○ | × | × | × | × |
| Group 1 | × | × | × | × | × | × | ○ | ○ |
| Group 2 | ○ | ○ | ○ | ○ | × | × | ○ | ○ |
| Group 3 | × | × | × | × | × | × | × | × |
| Group 4 | × | × | × | × | × | × | × | × |

Robot joints enclosed in this box

In the above example:

Group 0: Only robot joints involved.

Group 1: Only extended-joints (7th and 8th) involved.

Group 2: Robot joints and extended-joints (7th and 8th) involved.

By activating a task holding the "arm group" semaphore with extended-joints involved, you may operate the extended-joints.

In the above setting example, a task holding Group 1 can control the 7th and 8th extended-joints only.

■ *Getting an arm group*

To make tasks get an arm group, give a TAKEARM command an arm group number as an argument as shown below.

```
PROGRAM PRO1
TAKEARM 1
.
.
END
```

Program PRO1 gets Arm Group 1 by TAKEARM command with an argument set to 1.

■ **Restrictions on the application of arm groups**

If more than one task involves joints shared in their arm groups, those tasks cannot be *simultaneously* executed.

Tasks involving arm groups without shared joints can be *simultaneously* executed.

Example: Getting an Arm Group in Programs

| | J1 | J2 | J3 | J4 | J5 | J6 | J7 | J8 |
|---------|----|----|----|----|----|----|----|----|
| Group 0 | ○ | ○ | ○ | ○ | × | × | × | × |
| Group 1 | × | × | × | × | × | × | ○ | ○ |
| Group 2 | ○ | ○ | ○ | ○ | × | × | ○ | ○ |
| Group 3 | × | × | × | × | × | × | × | × |
| Group 4 | × | × | × | × | × | × | × | × |

```

PROGRAM PRO0      PROGRAM PRO1      PROGRAM PRO2
TAKEARM 0         TAKEARM 1         TAKEARM 2
  ⋮
  ⋮
END               END               END
  
```

Arm Groups 0 and 1 involve no shared joint, so PRO0 and PRO1 can be executed simultaneously. (Robot joints and extended-joints can be simultaneously controlled by different programs.)

Arm Groups 0 and 2 involve shared joints (J1 to J4), so PRO0 and PRO2 cannot be executed simultaneously. When the later TAKEARM command is executed, an error will occur.

■ **Releasing the currently held arm group**

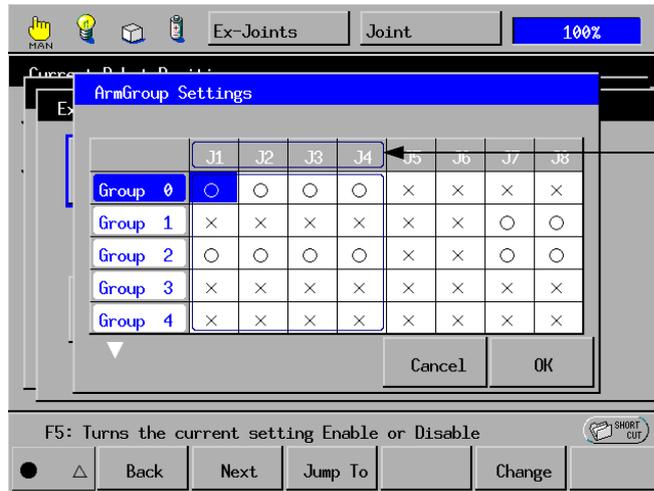
Executing a GIVEARM command releases the currently held arm group.

The currently held arm group will be automatically released also if a program stops due to occurrence of an error or because of normal program end.

NOTE: No arm group will be released by a Halt or Step stop signal.

■ **Setting up an arm group**

- Step 1** Call up the Arm Group Settings window by accessing:
[F2 Arm]—[F12 Maint.]—[F7 ExJoints]—[F1 ArmGroup]



Robot joints enclosed in this box

F5

- Step 2** Move the cursor to a joint to be grouped and then press [F5 Change].
- Step 3** After marking joints to be grouped or not with "O" or "X", respectively, press [OK].

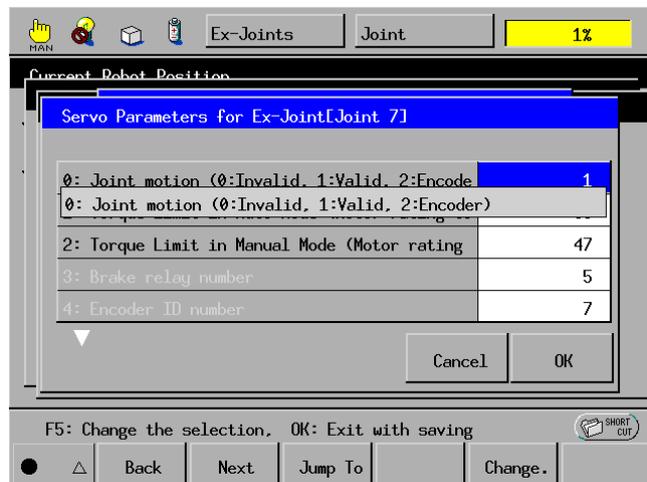
Notes for setting up an arm group

- (1) New settings will go into effect when the controller is turned off and then on after the settings is changed.
- (2) Arm Group 0 cannot be accessed.
- (3) Robot joints may be set all to either "Enabled" or "Disabled." For example, in a 4-joint robot, all four joints are either "Enabled" or "Disabled."
- (4) Extended-joints that you want to enable in an arm group should be set to "Valid" in the [Servo Parameters for Ex-Joint] window beforehand. If not, an error will occur when you are attempting to set up an arm group.

Access to the [Servo Parameters for Ex-Joint] window is:

[F2 Arm]—[F12 Maint.]—[F7 ExJoints]—[F8 ExServo]

For more details, refer to Section 3.2 "Setting the Extended-Joint Parameters."



[5] Boundless rotation of extended-joints

The boundless rotation function suppresses errors that could occur if an extended-joint keeps on rotating in the same direction by a relative motion command (DRIVE or MOVE with EX option).

You need to set the boundless rotation parameter to [1: Boundless] in the [Path parameter for Ex-Joint] window.

Notes for allowing boundless rotation on extended-joints

- (1) An absolute motion command (DRIVEA or MOVE with EXA option) cannot drive extended-joints whose boundless rotation is allowed.
- (2) Extended-joints whose boundless rotation is allowed cannot be moved by using variables.
- (3) When an extended-joint keeps on rotating in the same direction, the current value might suddenly jump to a large minus value.
- (4) In a DRIVEA or MOVE with EXA option, up to 7 digits of value may be specified for boundless rotation. If a value exceeding 7 digits is specified, the actual rotation amount will differ from the specified one.

For example

If DRIVE (5, 11111115555) is specified, 11111115555 will be internally interpreted as 1.111111*E+10 and 5555 will be ignored due to the definition of a single precision floating point number.

- (5) If a large value is specified as the amount of movement at one time in boundless rotation, then the "Out of range" error will occur. The quantum of movement depends on the gear ratio.

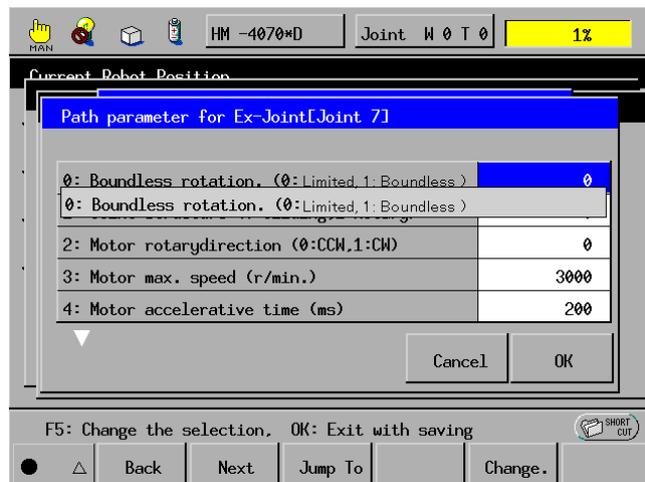
■ Allowing boundless rotation

From the top screen of the teach pendant, access:

[F2 Arm]—[F12 Maint.]—[F7 ExJoints]—[F7 ExPath]

In the [Path parameter for Ex-Joint] window, set the boundless rotation parameter to [1: Boundless].

For more details, refer to Section 3.2 "Setting the Extended-Joint Parameters."



[6] Command execution from the specified line for extended-joints

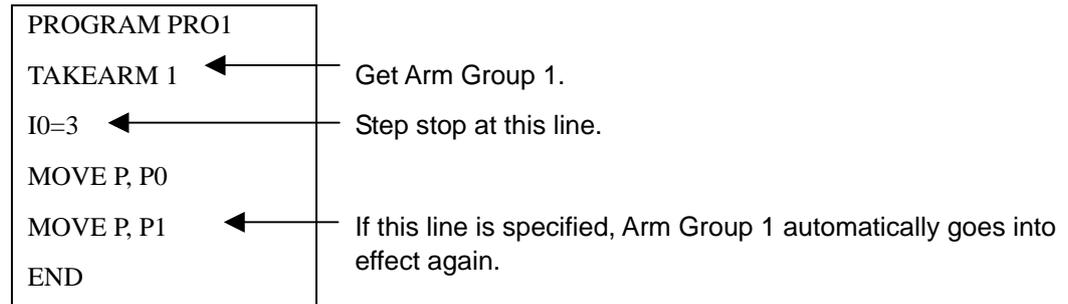
The specifications about command execution from the specified line are described in the SETTING-UP MANUAL, Section 3.3 "Teach Check Mode (TP), [4]."

This subsection describes only what is different from those specifications.

■ *Making effective the arm group previously obtained by TAKEARM*

If a command on the specified program line is executed, an arm group obtained by a TAKEARM command on the earlier line will automatically take effect.

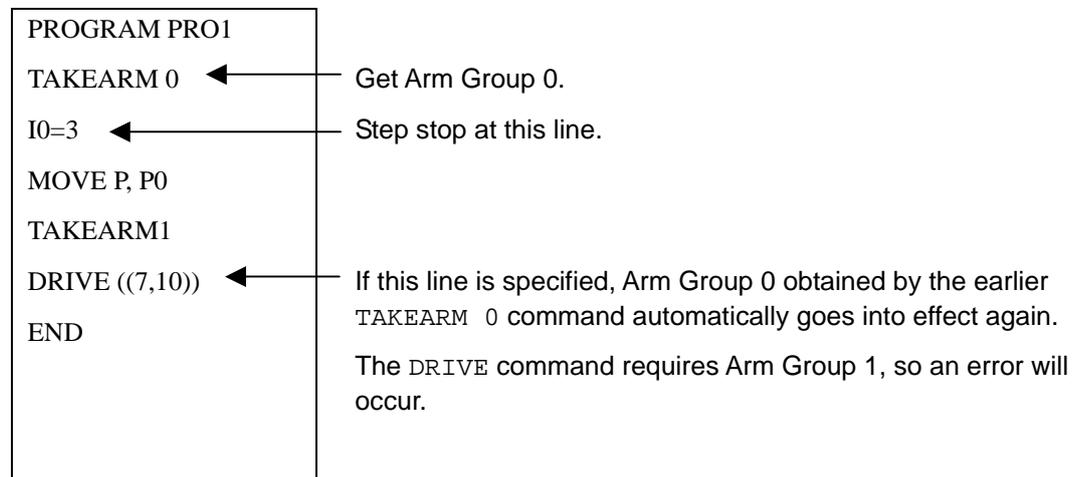
Example



Notes for executing a command from the specified line

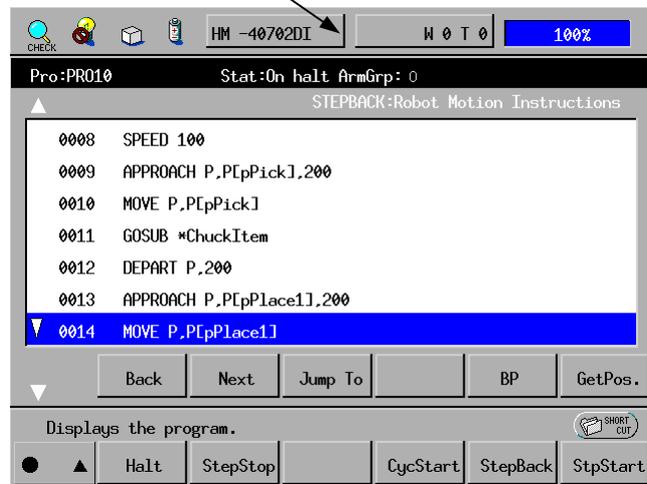
When using more than one arm group in a single task, be careful with the case given in the following sample:

Example



Chapter 3 Extended-Joint Related Function

You may check an active arm group currently held by a task by displaying the task. This program has got Arm Group 0.



[7] Continue function for extended-joints

The specifications of the continue function are described in the SETTING-UP MANUAL, Subsection 3.4.5.

This subsection describes what is different from those specifications.

■ *Auto position correction for extended-joints*

You may choose extended-joints you want to correct in position when Continue Start is executed.

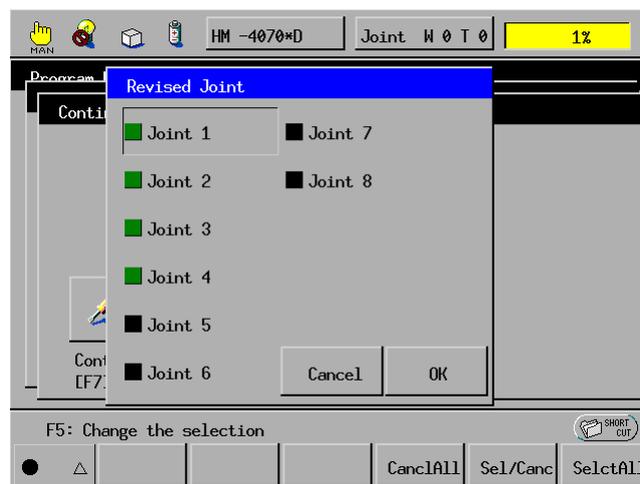
This function is useful for those extended-joints incapable of rotating in the reverse direction. For details, refer to the SETTING-UP MANUAL, Subsection 3.4.5.

■ *Choosing extended-joints to be corrected in position*

To call up the Revised Joint window, access:

**Top Screen—[F1 Program]—[F6 Aux.]—[F7 Continue]—
[F8 ReviseJnt]**

Window for Specifying Extended-Joints to be Corrected in Position



Choose joints to be corrected in position and then press [OK].

In the above example, robot joints (1 to 4) are selected so that those joints only will be corrected in position at Continue Start.

Notes

All robot joints (1 to 4) should be selected as a single unit. You cannot select any out of them.

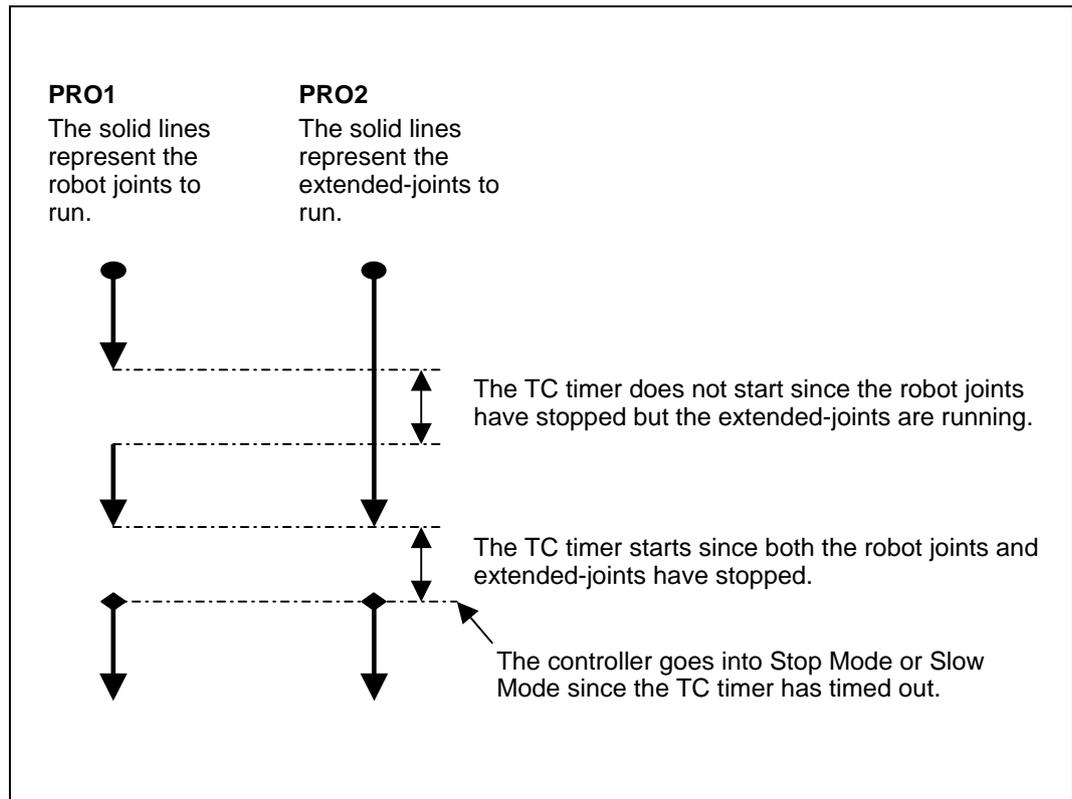
[8] SS (Safe Start) function for extended-joints

The specifications of the SS (Safe start) function are described in the SETTING-UP MANUAL, Subsection 3.4.6.

This subsection describes what is different from those specifications.

■ *Detecting TC time*

This function detects TC time if both the robot joints and extended-joints are not in motion for the specified period in the currently running program.



■ *Slow Mode*

If the TC timer times out, the speed of the subsequent operation of robot joints and extended-joints will become low.

■ *Stop Mode*

If the TC timer times out, the program will execute "Continue stop," stopping all the robot joints and extended-joints.

3.1.2 Commands Supporting Extended-Joints

NOTE: Commands described in this subsection may be used for general robots, not exclusively used for extended-joints.

TAKEARM (Statement)

Function

Gets an arm group. Upon execution of this statement, the programmed speed, acceleration and deceleration will be set to 100. If the gotten arm group includes any robot joint, this statement restores the tool coordinates and work coordinates to the origin.

Syntax

```
TAKEARM[ <ArmGroupName> ] [ <KEEP=DefaultValue> ]
```

Description

This section explains the functions relevant to extended-joints. For details, refer to the PROGRAMMER'S MANUAL, Section 14.3 "TAKEARM."

If <ArmGroupName> is omitted, the TAKEARM gets the semaphore of Arm Group 0 in which only robot joints are enabled.

If a task holding no arm group attempts to execute any of the following motion commands, an error will occur. Before executing those commands, get the arm semaphore (Arm Group) by using the TAKEARM command.

Commands Requiring Arm Group

| Commands | Requires: |
|---|--|
| HOME, TOOL, WORK, APPROACH, DEPART, DRAW, GOHOME, MOVE, ROTATEH, ROTATE, CHANGETOOL, CHANGEWORK, DRIVE, DRIVEA, SPEED, JSPEED, ACCEL, JACCEL, DECEL, JDECEL, INTERRUPT, LETENV, Motion optimization library, and Arm motion library | Arm group involving robot joints |
| DRIVE, DRIVEA, SPEED, JSPEED, ACCEL, JACCEL, DECEL, JDECEL, INTERRUPT, LETENV, and POSCLR | Arm group that may or may not involve robot joints |

Notes

- (1) The DRIVE and DRIVEA commands require an arm group involving a joint(s) to move.

Example: DRIVE (7,10)

To move the 7th joint, the task should hold the arm group involving the 7th joint.

- (2) The MOVE command with EX (EXA) option requires the arm group involving robot joints and extended-joints which are invoked by the EX (EXA) option.

Example: MOVE P, P0 EX ((7,10))

The arm group should involve robot joints as well as the 7th joint.

- (3) The speed setting commands will only change the speed of joints involved in an active arm group currently held in the task.
- (4) The LETENV command requires an arm group involving joints associated with parameters to be changed.

Related commands

GIVEARM

Examples

Shown below are program samples for the TAKEARM command when arm groups are arranged as listed below (for 4-axis robots).

| | J1 | J2 | J3 | J4 | J5 | J6 | J7 | J8 |
|---------|----|----|----|----|----|----|----|----|
| Group 0 | ○ | ○ | ○ | ○ | × | × | × | × |
| Group 1 | × | × | × | × | × | × | ○ | ○ |
| Group 2 | ○ | ○ | ○ | ○ | × | × | ○ | ○ |
| Group 3 | × | × | × | × | × | × | × | × |
| Group 4 | × | × | × | × | × | × | × | × |

```
PROGRAM PRO1
TAKEARM 1           'Get Arm Group 1 (7th extended-joint involved).
DRIVEA (7,100)     'Move the 7th extended-joint to an angle of 100
                   'degrees.
END
```

```
PROGRAM PRO2
TAKEARM 2           'Get Arm Group 2 (all robot joints and 7th
                   'extended-joint involved).
MOVE P,P0 EX ((7,10)) 'Simultaneously move robot joints and 7th
                   'extended-joint.
DRIVEA (7,100)     'Move the 7th extended-joint to an angle of
                   '100 degrees.
END
```

Notes

- (1) One program cannot hold more than one different arm group. However, it can get the same arm group again in one program.

```
Example: TAKEARM 0
        MOVE P, P0
        TAKEARM 0   'Can get Arm Group 0 again, even if this
                   'program has got it.
        TAKEARM 1   'Error occurs since TAKEARM attempts to
                   'get Arm Group 1 while Arm Group 0 has been
                   'held.
```

- (2) If TAKEARM with KEEP option being set to "1" gets an arm group, the arm speed will not be initialized and remain as that of the old arm group.

Example: If arm groups are arranged as listed below and the three programs exist:

| | J1 | J2 | J3 | J4 | J5 | J6 | J7 | J8 |
|---------|----|----|----|----|----|----|----|----|
| Group 0 | ○ | ○ | ○ | ○ | × | × | × | × |
| Group 1 | × | × | × | × | × | × | ○ | ○ |
| Group 2 | ○ | ○ | ○ | ○ | × | × | ○ | ○ |
| Group 3 | × | × | × | × | × | × | × | × |
| Group 4 | × | × | × | × | × | × | × | × |

```

PROGRAM PRO1          PROGRAM PRO2          PROGRAM PRO3
TAKEARM 1             TAKEARM 2             TAKEARM 1 keep=1
SPEED 10              SPEED 20              DRIVE(7,10)
DRIVE(7,10)          DRIVE(7,10)          END
END                   END

```

First run PRO1. The 7th-joint will move at speed 10.

Next run PRO2. The 7th-joint will move at speed 20.

Then run PRO3. Since the KEEP option is "1", the 7th-joint will move at speed 10 keeping the speed defined by SPEED command for Arm Group 1 in PRO1.

GIVEARM (Statement)

Function

Releases the currently held arm group.

Syntax

GIVEARM

Description

This subsection explains the functions relevant to extended-joints. For others, refer to the PROGRAMMER'S MANUAL, Section 14.2 "GIVEARM."

GIVEARM releases the currently held arm group, allowing other tasks to get the arm group semaphore.

If a task that holds no arm group attempts to execute the GIVEARM command, an error will result.

NOTE:

In any of the following cases, the program or task automatically releases the arm semaphore (arm group), so you may omit the GIVEARM command.

- **When an END command executes and normally terminates a program. (Excluding an END written at the end of a called program.)**
- **When a program or task stops due to any error or by execution of a KILL command.**

(Note that an arm group will not be released by Halt or Step stop.)

Related commands

TAKEARM

Examples

```
PROGRAM PRO1
TAKEARM 1      'Get Arm Group 1.
GIVEARM       'Release Arm Group 1 currently held.
END
```

```
PROGRAM PRO2
TAKEARM 2      'Get Arm Group 2.
END           'Stop the program normally and release Arm
              'Group 2 currently held.
```

Notes

If a GIVEARM command is issued, the controller will execute it after completion of all arm motions invoked by the current arm group. Even if you specify a pass motion preceding a GIVEARM, therefore, the controller will ignore it so that no pass motion will execute.

DRIVE (Statement)

Function

Carries out the relative motion of each joint.

Syntax

```
DRIVE[@PassStartOffset]( <JntNumber> , <RelativeDistance> )  
[ , ( <JntNumber> , <RelativeDistance> ) ... ] [ , <MotionOption> ]  
[ , NEXT ]
```

Description

DRIVE moves a joint specified by <JntNumber> towards the angle (in degrees) specified by <RelativeDistance>.

If <RelativeDistance> is positive, the joint moves in the positive direction; if negative, the joint moves in the negative direction.

To execute this command, you need to get an arm group involving a joint(s) to be moved.

If a same joint is specified repeatedly, the last specification takes effect.

For details about <@PassStartOffset>, <MotionOption>, and NEXT, refer to the PROGRAMMER'S GUIDE, Section 12.1 "DRIVE."

Related commands

DRIVEA, MOVE with EX or EXA option

(Commands applicable to the extended-joints include MOVE with EX or EXA option, DRIVE and DRIVEA only.)

Example

| | J1 | J2 | J3 | J4 | J5 | J6 | J7 | J8 |
|-------------------|----|----|----|----|----|----|----|----|
| Group \emptyset | ○ | ○ | ○ | ○ | × | × | × | × |
| Group 1 | × | × | × | × | × | × | ○ | ○ |
| Group 2 | ○ | ○ | ○ | ○ | × | × | ○ | ○ |
| Group 3 | × | × | × | × | × | × | × | × |
| Group 4 | × | × | × | × | × | × | × | × |

```
PROGRAM PRO1  
TAKEARM1 'Arm Group 1 involves 7th and 8th joints.  
DRIVE (7,30),(8,50) 'Operate 7th and 8th joints  
END
```

In the above example, to move the 7th and/or 8th extended-joint, the program needs to get Arm Group 1 or Arm Group 2.

Notes

If a numerical value is set to <@PassStartOffset> for the specified extended-joint, the joint will move in pass motion regardless of the entered value.

DRIVEA (Statement)

Function

Carries out the absolute motion of each joint.

Syntax

```
DRIVE[ <@PassStartOffset> ] ( <JntNumber> , <JntCoordinates> )
[ , ( <JntNumber> , <JntCoordinates> ) ... ] [ , <MotionOption> ]
[ , NEXT ]
```

Description

DRIVE moves a joint specified by <JntNumber> towards the angle (in degrees) specified by <JntCoordinates>.

To execute this command, it is necessary to get an arm group including a joint(s) to be moved.

If a same joint is specified repeatedly, the last specification takes effect.

For details about <@PassStartOffset>, <MotionOption>, and NEXT, refer to the PROGRAMMER'S GUIDE, Section 12.1 "DRIVEA."

Related commands

DRIVE, MOVE with EX or EXA option

(Commands applicable to the extended-joints include MOVE with EX or EXA option, DRIVE and DRIVEA only.)

Example

| | J1 | J2 | J3 | J4 | J5 | J6 | J7 | J8 |
|---------|----|----|----|----|----|----|----|----|
| Group 0 | ○ | ○ | ○ | ○ | × | × | × | × |
| Group 1 | × | × | × | × | × | × | ○ | ○ |
| Group 2 | ○ | ○ | ○ | ○ | × | × | ○ | ○ |
| Group 3 | × | × | × | × | × | × | × | × |
| Group 4 | × | × | × | × | × | × | × | × |

```
PROGRAM PR01
TAKEARM1           'Arm Group 1 involves 7th and 8th joints.
DRIVEA (7,30),(8,50) 'Operate 7th and 8th joints
END
```

In the above example, to move the 7th and/or 8th extended-joint, the program needs to get Arm Group 1 or Arm Group 2.

Notes

- (1) If a numerical value is set to <@PassStartOffset> for the specified extended-joint, the joint will move in pass motion regardless of the entered value.
- (2) This command is not applicable to any joints specified for boundless rotation.

MOVE (Statement)

Function

Moves the robot flange to the specified coordinates.

If specified with an EX option (relative motion of extended-joints) or EXA option (absolute motion of extended-joints), the MOVE can move both the robot flange and the extended-joints synchronously. (i.e. Synchronized start and stop from/at the specified positions is possible.)

Syntax

```
MOVE<InterpolationMethod>, [@<PassStartOffset>]<Pose>  
[<EX/EXAoption>][, [@<PathStartOffset>]<Pose>[<EX/EXAoption  
>]...][, <MotionOption>][, NEXT]
```

Description

This subsection explains the functions relevant to extended-joints. For others, refer to the PROGRAMMER'S MANUAL, Section 12.1 "MOVE."

The MOVE moves the joints from the current positions to the coordinates specified by <Pose>.

The syntax of an EX or EXA option is shown below.

<EX/EXAoption> syntax

```
EX((<JntNumber>, <RelativeDistance>)[, (<JntNumber>, <RelativeDistance>)...])
```

```
EXA((<JntNumber>, <AxisCoordinates>)[, (<JntNumber>, <AxisCoordinates>)...])
```

To <JntNumber>, you can specify only an extended-joint, never specify any robot joint.

For details about other options, refer to the PROGRAMMER'S MANUAL, Section 12.1 "MOVE."

Related commands

DRIVE, DRIVEA

(Commands applicable to the extended-joints include MOVE with EX or EXA option, DRIVE and DRIVEA only.)

Example

| | J1 | J2 | J3 | J4 | J5 | J6 | J7 | J8 |
|---------|----|----|----|----|----|----|----|----|
| Group 0 | ○ | ○ | ○ | ○ | × | × | × | × |
| Group 1 | × | × | × | × | × | × | ○ | ○ |
| Group 2 | ○ | ○ | ○ | ○ | × | × | ○ | ○ |
| Group 3 | × | × | × | × | × | × | × | × |
| Group 4 | × | × | × | × | × | × | × | × |

```

PROGRAM PRO1
TAKEARM 2                                'Get Arm Group 2 involving both robot
                                           'joints and extended-joints.
MOVE P, P0 EX ((7,30),(8,10))           'Move robot flange to P0 as well as
                                           'moving 7th and 8th extended-joints to
                                           'relative coordinates synchronously.
MOVE P, P1 EXA ((7,30))                 'Move robot flange to P1 as well as
                                           'moving 7th extended-joint to absolute
                                           'coordinates synchronously.
END
    
```

Notes

- (1) The MOVE with EXA option (absolute motion) is not applicable to any extended-joints specified for boundless rotation.
- (2) Given below is an example handling a POSE array.

Example: MOVE P,P[3 TO 5]EX((7,30))

The above sample will produce the same result as the following; that is, the motion of the 7th joint will synchronize with that of the robot flange to P5.

```

MOVE P,P3,NEXT
MOVE P,P4,NEXT
MOVE P,P5 EX((7,30))
    
```

CUREXJ (Statement)

Function

Gets the current angle of an extended-joint into a floating-point variable.

Syntax

```
CUREXJ (<JntNumber>)
```

Description

CUREXJ reads an angle detected by the encoder of an extended-joint specified by <JntNumber> into a floating-point variable. If the specified joint is in motion, this command will get the current angle of the joint detected when this command executes.

To get the target angle of an extended-joint, use a DESTEXJ command.

Related commands

CURJNT, CURPOS, CURTRN, and DESTEXJ

Example

```
PROGRAM PRO1
DIM lf1 AS SINGLE
TAKEARM 1
DRIVEA (7,100)           'Move the 7th joint to an angle of 100 degrees.
lf1 = CUREXJ(7)         'Assign current position of 7th joint to lf1.
END
```

Notes

When the machine is locked (Machine Lock state), the CUREXJ command will get not an angle detected by an encoder but a virtual angle (commanded angle).

DESTEXJ (Statement)

Function

Gets the target position of an extended-joint invoked by the current motion command into a floating-point variable. If the robot is on halt, this command will get the current position (commanded value).

Syntax

```
DESTEXJ(<JntNumber>)
```

Description

DESTEXJ reads the target position of an extended-joint invoked by the current motion command and specified by <JntNumber> into a floating-point variable.

To get a position detected by the encoder of an individual joint, use a CUREXJ command.

Related commands

CUREXJ, DESTJNT, DESTPOS, and DESTTRN

Example

```
PROGRAM PRO1
DIM lf1 AS SINGLE
TAKEARM 1
DRIVEA (7,100),NEXT      'Move 7th joint to an angle of 100 degrees.
                          'NEXT advances the process to the next command
                          'before the motion is completed.
lf1 = DESTEXJ(7)        'Assign target angle 100 of previously
                          'commanded motion into lf1.
END
```

Notes

If a robot motion is stopped by entering Stop motion command (Refer to INTERRUPT ON/OFF stated later), the DESTEXJ will get the joint angle where the motion is stopped.

Example

```
INTERRUPT ON
DRIVEA(7,100)           'Interrupt signal turns ON during motion.
                          'The 7th joint stops and the process advances
                          'to the next command.

INTERRUPT OFF
F1=DESTEXJ(7)          'The value of F1 will not be 100 but the value
                          'of the angle where the joint stopped.
```

ARRIVE (Statement)

Function

Defines the motion ratio relative to the programmed full travel distance to the target point in order to make the current program stand by to execute the next step until the robot reaches the defined motion ratio.

Syntax

ARRIVE<MotionRatio>

Description

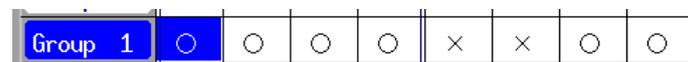
This subsection explains the functions relevant to extended-joints. For others, refer to the PROGRAMMER'S MANUAL, Section 12.1 "ARRIVE."

ARRIVE takes effect only for robot joints. Therefore, when operating only extended-joints by DRIVE or DRIVEA, this ARRIVE command does not work.

Related commands

Example

If an arm group is set in 4-joint robots as shown below, the program sample below works as follows.



```
PROGRAM PRO1
```

```
TAKEARM 1
```

```
DRIVE(1,30)(7,30),NEXT
```

```
ARRIVE 50
```

```
MOVE P,P0 EX((7,30)),NEXT
```

```
ARRIVE 50
```

```
DRIVE(7,30),NEXT
```

```
ARRIVE 50
```

```
END
```

Since the previous command involves robot joints also, the ARRIVE works.

Synchronized motion with extended-joints. So the ARRIVE will work.

Since the previous DRIVE command involves an extended-joint only, the ARRIVE does not work and the process advances to the next line.

SPEED (Statement)

Function

Specifies the internal composite speed of joints included in a currently held arm group.

Syntax

SPEED<SpeedRatio>

Description

The <SpeedRatio> should be the target ratio (%) of the maximum internal composite speed of joints in a currently held arm group. The entry range is from 0.1 to 100 of a real number.

Note: If a value 0.1 or less but greater than 0 is specified to <SpeedRatio>, no error will result, but the actual speed may be different from the specified speed.

The actual speed is equal to (external speed x internal speed ÷ 100).

If the SPEED is specified, the JSPEED, ACCEL, DECEL, JACCEL and JDECEL will be set automatically.

Example: If you write SPEED 50, the following will be set automatically:

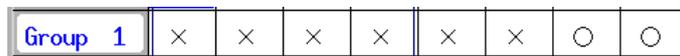
```
JSPEED 50 (same value as SPEED)
ACCEL 25 (SPEED*SPEED÷100)
JACCEL 25 (SPEED*SPEED÷100)
DECEL 25 (SPEED*SPEED÷100)
JDECEL 25 (SPEED*SPEED÷100)
```

If you write SPEED without getting any arm group beforehand, an error will result.

Related commands

ACCEL, DECEL, and JSPEED

Example



```
PROGRAM PRO1
TAKEARM 1      'Get Arm Group 1 (including 7th and 8th joints).
SPEED 100     'Specify composite speed of joints (7th and 8th)
               'involved in Arm Group 1.
END
```

JSPEED (Statement)

Function

Specifies the internal speed of individual joints included in a currently held arm group.

Syntax

JSPEED<JointSpeedRatio>

Description

The <JointSpeedRatio> should be the target ratio (%) of the maximum internal speed of individual joints in a currently held arm group. The entry range is from 0.1 to 100 of a real number.

Note: If a value 0.1 or less but greater than 0 is specified to <JointSpeedRatio>, no error will result, but the actual speed may be different from the specified speed.

The actual speed is equal to (external speed x internal speed ÷ 100).

If the JSPEED is specified, the JACCEL and JDECEL will be set automatically.

Example: If you write JSPEED 50, the following will be set automatically:

JACCEL 25 (JSPEED*JSPEED÷100)

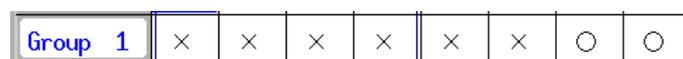
JDECEL 25 (JSPEED*JSPEED÷100)

If you write JSPEED without getting any arm group beforehand, an error will result.

Related commands

JACCEL, JDECEL and SPEED

Example



```
PROGRAM PRO1
TAKEARM 1           'Get Arm Group 1 (involving 7th and 8th joints).
JSPEED 100         'Specify speed of joints (7th and 8th) involved
                   'in Arm Group 1.
END
```

ACCEL (Statement)

Function

Specifies the internal composite acceleration/deceleration of joints included in a currently held arm group.

Syntax

ACCEL<AccelerationRatio>[, <DecelerationRatio>]

Description

The <AccelerationRatio> or <DecelerationRatio> should be the target ratio (%) of the maximum internal composite acceleration or deceleration of joints in a currently held arm group, respectively. The entry range is from 0.0001 to 100 of a real number.

Note: If a value 0.0001 or less but greater than 0 is specified, no error will result, but the actual acceleration may be different from the specified one.

The actual acceleration/deceleration is equal to (external x internal acceleration or deceleration ÷ 100).

If the ACCEL is specified, the JACCEL will be set automatically.

Example: If you write ACCEL 50, the JACCEL 50 (same value as ACCEL) will be set automatically:

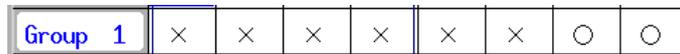
```
JACCEL 25 (JSPEED*JSPEED÷100)
JDECEL 25 (JSPEED*JSPEED÷100)
```

If you write ACCEL without getting any arm group beforehand, an error will result.

Related commands

DECEL and SPEED

Example



```
PROGRAM PR01
TAKEARM 1          'Get Arm Group 1 (involving 7th and 8th joints).
ACCEL 100          'Specify composite acceleration ratio of joints
                   '(7th and 8th) involved in Arm Group 1.
END
```

JACCEL (Statement)

Function

Specifies the internal acceleration and deceleration of individual joints included in a currently held arm group.

Syntax

JACCEL<JointAccelerationRatio>[,<JointDecelerationRatio>]

Description

The <JointAccelerationRatio> or <JointDecelerationRatio> should be the target ratio (%) of the maximum internal acceleration or deceleration of individual joints in a currently held arm group, respectively. The entry range is from 0.0001 to 100 of a real number.

Note: If a value 0.0001 or less but greater than 0 is specified, no error will result, but the actual acceleration may be different from the specified one.

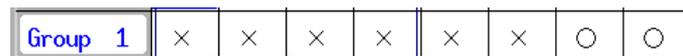
The actual acceleration/deceleration is equal to (external x internal acceleration or deceleration ÷ 100).

If you write JACCEL without getting any arm group beforehand, an error will result.

Related commands

JDECEL, JSPEED, and SPEED

Example



```
PROGRAM PRO1
TAKEARM 1      'Get Arm Group 1 (involving 7th and 8th joints).
JACCEL 100     'Specify acceleration ratio of joints (7th and 8th)
               'involved in Arm Group 1.
END
```

DECEL (Statement)

Function

Specifies the internal composite deceleration of joints involved in a currently held arm group.

Syntax

DECEL<DecelerationRatio>

Description

The <DecelerationRatio> should be the target ratio (%) of the maximum internal composite deceleration of joints in a currently held arm group. The entry range is from 0.0001 to 100 of a real number.

Note: If a value 0.0001 or less but greater than 0 is specified, no error will result, but the actual deceleration may be different from the specified one.

The actual deceleration is equal to (external x internal deceleration ÷ 100).

If the DECEL is specified, the JDECEL will be set automatically.

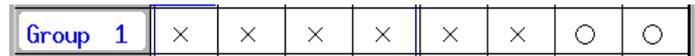
Example: If you write DECEL 50, the JDECEL 50 (same value as DECEL) will be set automatically:

If you write DECEL without getting any arm group beforehand, an error will result.

Related commands

ACCEL and SPEED

Example



```
PROGRAM PRO1
TAKEARM 1      'Get Arm Group 1 involving 7th and 8th joints.
DECEL 100      'Specify deceleration ratio of joints (7th and 8th)
                'in Arm Group 1.
END
```

JDECEL (Statement)

Function

Specifies the internal deceleration ratio of individual joints included in a currently held arm group.

Syntax

JDECEL<DecelerationRatio>

Description

The <DecelerationRatio> should be the target ratio (%) of the maximum internal deceleration of individual joints in a currently held arm group. The entry range is from 0.0001 to 100 of a real number.

Note: If a value 0.0001 or less but greater than 0 is specified, no error will result, but the actual deceleration may be different from the specified one.

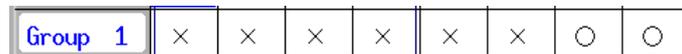
The actual deceleration is equal to (external x internal deceleration ÷ 100).

If you write JDECEL without getting any arm group beforehand, an error will result.

Related commands

JACCEL, JSPEED, and SPEED

Example



```
PROGRAM PRO1
TAKEARM 1      'Get Arm Group 1 involving 7th and 8th joints.
JDECEL 100     'Specify deceleration ratio of joints (7th and 8th)
               'involved in Arm Group 1.
END
```

CURSPD (Statement)

Function

Gets the current internal composite speed of joints included in a currently held arm group.

Syntax

CURSPD

Description

Executing this command in a task holding no arm group will return a value of 100.

Related commands

CURACC, CURDEC, CURJACC, CURJDEC, and CURJSPD

Example

```
PROGRAM PRO1
TAKEARM 1           'Get Arm Group 1.
SPEED 70            'Specify composite speed of joints included in
                    'Arm Group 1 at 70.
I0=CURSPD           'Return current speed 70 of Arm Group 1 to I0.
END
```

CURJSPD (Statement)

Function

Gets the current internal speed of individual joints included in a currently held arm group.

Syntax

CURJSPD

Description

Executing this command in a task holding no arm group will return a value of 100.

Related commands

CURACC, CURDEC, CURJACC, CURJDEC, and CURSPD

Example

```
PROGRAM PRO1
TAKEARM 1           'Get Arm Group 1.
JSPEED 70          'Specify speed of individual joints in Arm Group 1
                   'at 70.
I0=CURJSPD         'Return current speed 70 of Arm Group 1 to I0.
END
```

CURACC (Statement)

Function

Gets the current internal composite acceleration of joints included in a currently held arm group.

Syntax

CURACC

Description

Executing this command in a task holding no arm group will return a value of 100.

Related commands

CURDEC, CURJACC, CURJDEC, CURJSPD, and CURSPD

Example

```
PROGRAM PRO1
TAKEARM 1      'Get Arm Group 1.
ACCEL 70      'Specify composite acceleration of joints included in
              'Arm Group 1 at 70.
I0=CURACC     'Return current acceleration 70 of Arm Group 1 to I0.
END
```

CURJACC (Statement)

Function

Gets the current internal acceleration of individual joints included in a currently held arm group.

Syntax

CURJACC

Description

Executing this command in a task holding no arm group will return a value of 100.

Related commands

CURACC, CURDEC, CURJDEC, CURJSPD, and CURSPD

Example

```
PROGRAM PRO1
TAKEARM 1      'Get Arm Group 1.
JACCEL 70     'Specify acceleration of individual joints included
              'in Arm Group 1 at 70.
I0=CURJACC    'Return current acceleration 70 of Arm Group 1 to I0.
END
```

CURDEC (Statement)

Function

Gets the current internal composite deceleration of joints included in a currently held arm group.

Syntax

```
CURDEC
```

Description

Executing this command in a task holding no arm group will return a value of 100.

Related commands

CURACC, CURJACC, CURJDEC, CURJSPD, and CURSPD

Example

```
PROGRAM PRO1
TAKEARM 1      'Get Arm Group 1.
DECEL 70      'Specify composite deceleration of joints included
              'in Arm Group 1 at 70.
I0=CURDEC     'Return current deceleration 70 of Arm Group 1 to I0.
END
```

CURJDEC (Statement)

Function

Gets the current internal deceleration of individual joints included in a currently held arm group.

Syntax

CURJDEC

Description

Executing this command in a task holding no arm group will return a value of 100.

Related commands

CURACC, CURDEC, CURJACC, CURJSPD, and CURSPD

Example

```
PROGRAM PRO1
TAKEARM 1      'Get Arm Group 1.
JDECEL 70      'Specify current deceleration of individual joints
                'included in Arm Group at 70.
I0=CURJDEC     'Return current deceleration 70 of Arm Group 1 to I0.
END
```

INTERRUPT ON/OFF (Statement)

Function

Interrupts the current robot motion.

Syntax

INTERRUPT {ON/OFF }

Description

INTERRUPT ON and INTERRUPT OFF are used as a pair. In program lines sandwiched by the pair, if an interrupt skip signal turns ON during execution of motion commands, then the robot controller will interrupt execution of the current motion command and proceeds to the next program step.

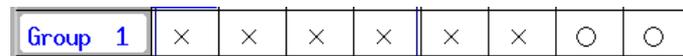
To execute the INTERRUPT command, the task must have gotten an arm group semaphore.

Without INTERRUPT ON written earlier, even if an interrupt skip signal turns ON, the controller will not interrupt execution of any motion commands.

If the program comes to a stop or GIVEARM command is executed, then INTERRUPT will be set to OFF automatically.

Related commands

Example



```

PROGRAM PRO1
TAKEARM 1           'Get Arm Group 1 involving 7th and 8th
                    'extended-joints.

INTERRUPT ON
DRIVE (7,100), (8,30) 'If interrupt skip signal turns ON during
                    'execution of this motion command between
                    'INTERRUPT ON/OFF program lines, then the
                    'controller interrupts the command and
                    'proceeds to the next program step.

INTERRUPT OFF
END
    
```

Notes

- (1) If the controller executes any relative motion command immediately following an interrupt skip, then the subsequent relative motion will start from the position where the robot stops. In program lines sandwiched by the INTERRUPT pair, therefore, use an absolute motion command.
- (2) Turning an interrupt skip signal ON interrupts all motion commands sandwiched by the INTERRUPT pair and stops all robot motions. Therefore carefully design your robot system for multitasking if you use the INTERRUPT.

Program Example

| | J1 | J2 | J3 | J4 | J5 | J6 | J7 | J8 |
|---------|----|----|----|----|----|----|----|----|
| Group 0 | ○ | ○ | ○ | ○ | × | × | × | × |
| Group 1 | × | × | × | × | × | × | ○ | ○ |
| Group 2 | ○ | ○ | ○ | ○ | × | × | ○ | ○ |
| Group 3 | × | × | × | × | × | × | × | × |
| Group 4 | × | × | × | × | × | × | × | × |

| | |
|---|---|
| PRO1 TAKEARM 0 INTERRUPT ON MOVE P,P0 'Motion command ⋮ INTERRUPT OFF END | PRO2 TAKEARM 1 INTERRUPT ON DRIVE(7,30) 'Motion command ⋮ INTERRUPT OFF END |
|---|---|

When the controller is running PRO1 and PRO2 programs concurrently in multitasking mode, if an interrupt skip signal turns ON, then the controller will interrupt all commands (MOVE in PRO1 and DRIVE in PRO2 in the above example) running in both programs and make the programs simultaneously proceed to the next step.

POSCLR (Statement)

Function

Forcibly restores the current position of a joint to 0 mm or 0 degree.

Syntax

```
POSCLR<JntNumber>
```

Description

POSCLR forcibly restores the angle of a joint specified by <JntNumber> to 0 mm or 0 degree.

This command is applicable only to joints specified for boundless rotation.

Use this command if a joint keeps on rotating in the same direction so that any of the following happens:

- The current position value becomes too large to handle.
- The current position value jumps to a large negative value (due to overflow or wrap-around of a variable value).

To execute this command, you need to get an arm group including a joint whose position is to be restored to its origin.

Related commands

Example

```
PROGRAM PRO1
TAKEARM 1
DRIVEA (7,100) 'Move 7th joint to an angle of 100 degrees.
POSCLR 7      'Force restore the current angle of the 7th
              'joint to 0 degree.
END
```

Notes

- (1) This command is not applicable to robot joints (Only to extended-joints).
- (2) The controller runs this command after the robot has completely stopped. Therefore, any pass motion command written preceding POSCLR will cause no pass motion.
- (3) The Step Back function cannot return the program control back to any command written preceding the POSCLR command.

mvSetPulseWidthJnt (Library)

Function

Sets the encoder pulse count for an allowable positioning error for a specified extended-joint.

Syntax

```
mvSetPulseWidthJnt (<JntNumber> , <AllowablePulseCount> )
```

Description

This library sets a value specified by <AllowablePulseCount> for an extended-joint specified by <JntNumber>.

If the actual count of positioning error pulses issued by a joint motor encoder is within <AllowablePulseCount> at execution of a motion command with @E option, the controller will recognize that the joint is on halt.

Macro definition

File <pacman.h> is required.

Related commands

mvResetPulseWidthJnt and mvSetPulseWidth

Notes

This mvSetPulseWidthJnt library is applicable to 7th and 8th extended-joints only. For robot joints, use the mvSetPulseWidth library.

Example

```
CALL mvSetPulseWidthJnt(7,10)      'Set allowable error pulse count  
                                   'for 7th extended-joint at 10.
```

mvResetPulseWidthJnt (Library)

Function

Resets the encoder pulse count for an allowable positioning error for a specified extended-joint to the default.

Syntax

```
mvResetPulseWidthJnt (<JntNumber> )
```

Description

This library resets the current encoder pulse count specified for an allowable positioning error for an extended-joint specified by <JntNumber> to the default value of 20.

For details about the encoder pulse count for an allowable positioning error,

Macro definition

File <pacman.h> is required.

Related commands

`mvResetPulseWidth` and `mvSetPulseWidthJnt`

Notes

This `mvResetPulseWidthJnt` library is applicable to 7th and 8th extended-joints only. For robot joints, use the `mvResetPulseWidth` library.

Example

```
CALL mvResetPulseWidthJnt(7)    'Reset allowable error pulse count  
                                'for 7th extended-joint to the default.
```

SetCycloidJnt (Library)

Function

Enters a specified extended-joint into the cycloid mode where the controller suppresses the peak of overshoot and residual oscillation that would occur in an end motion.

Syntax

```
SetCycloidJnt (<JntNumber> )
```

Description

- (1) In the cycloid mode, the controller will automatically reduce the deceleration ratio currently set for an extended-joint specified by <JntNumber> so as to suppress the peak of overshoot and residual oscillation. The extended-joint will smoothly arrive at the target position.
- (2) In the cycloid mode, the controller will take longer operation time than the normal mode. Use this library only when the robot operation can allow timing margins for all related operations.
- (3) Once specified, the cycloid mode will remain in effect for the specified extended-joint even driven by any other programs. To reset the cycloid mode, use the `ResetCycloidJnt` library.
- (4) When the specified extended-joint is synchronized with the robot operation, this library will not be called, so the extended-joint will operate according to the robot settings.
- (5) If more than one extended-joint runs synchronously, all of those joints will enter the cycloid mode once any one of them has entered the cycloid mode by `SetCycloidJnt`. This case cannot select joints to be involved.

Macro definition

File <pacman.h> is required.

Related commands

`ResetCycloidJnt`, `ResetCycloid`, and `SetCycloid`

Notes

Before calling this library, you need to hold an arm group semaphore. The `TAKEARM` should execute beforehand.

This `SetCycloidJnt` library is applicable to extended-joints only. For robot joints, use the `SetCycloid` library.

Example

```
Call SetCycloidJnt(8) 'Make 8th extended-joint enter cycloid mode.
```

ResetCycloidJnt (Library)

Function

Cancels the cycloid mode set for a specified extended-joint and restores the normal mode.

Syntax

```
ResetCycloidJnt (<JntNumber>)
```

Description

`ResetCycloidJnt` cancels the cycloid mode set for a joint specified by `<JntNumber>` and restores the normal mode. To set the cycloid mode again, the `SetCycloidJnt` should execute again.

Macro definition

File `<pacman.h>` is required.

Related commands

`ResetCycloid`, `SetCycloidJnt`, and `SetCycloid`

Notes

Before calling this library, you need to hold an arm group semaphore. The `TAKEARM` should execute beforehand.

This `ResetCycloidJnt` library is applicable to extended-joints only. For robot joints, use the `ResetCycloid` library.

Example

```
CALL ResetCycloidJnt (8) 'Release cycloid mode set for 8th  
    'extended-joint.
```

ResetCurLmt (Library for Ver.1.2 or later)

Function Releases the drive current limit set for a specified joint motor.

Syntax ResetCurLmt (<JntNumber>)

Description

ResetCurLmt releases the drive current limit set for the motor of a joint specified by <JntNumber>. The motor drive current limit and positioning error allowances will revert to the defaults.

[For Ver. 1.4 or earlier] If you set "0" to <JntNumber>, the drive current limit set for all joints will revert to the default.

[For Ver. 1.5 or later] If you set "0" to <JntNumber>, the drive current limit set for all joints involved in an arm group semaphore held by the current task running ResetCurLmt, will revert to the default.

Macro definition

File <pacman.h> is required.

Related commands

ResetEralw and SetCurLmt

Notes

- (1) When running this library to release the current limit, the controller will remove preset error allowances concurrently. If any external reaction causes any angle error in joint positioning, then the controller's processing time for the removal will vary depending upon the speed and acceleration/deceleration ratio previously set. If you need to reduce the processing time, set the ratio as high as possible.
- (2) This library is executable even if the motor power is off. To do so, run the program as written below after terminating a current task holding an arm semaphore.

```
PRO999
TAKEARM
CALL ResetCurLmt(0)
END
```

- (3) [For Ver. 1.4 or earlier] Write this library in a TAKEARMed task that has got robot arm semaphore. If you specify any joints not in the arm semaphore to <JntNumber>, then error [21F7 Cannot take arm semaphore] will result.

[For Ver. 1.5 or later] Write this library in a TAKEARMed task that has got an arm group. If you specify any joints not included in the arm group to <JntNumber>, then error [27D* Cannot take J* semaphore] will result.

Example

```
CALL ResetCurLmt(0) 'Release the drive current limit set for all
                    'joints.
CALL ResetGrvOffset 'Reset compensation for gravity torque applied
                    'to all joints.
```

ResetEralw (Library for Ver. 1.2 or later)

Function

Resets the positioning error allowance of a specified joint to the initial value.

Syntax

```
ResetEralw(<JntNumber>)
```

Description

`ResetEralw` resets the positioning error allowance set for a joint specified by `<JntNumber>` to the initial value.

[For Ver. 1.4 or earlier] If you set "0" to `<JntNumber>`, the positioning error allowance set for all joints will revert to the default.

[For Ver.1.5 or later] If you set "0" to `<JntNumber>`, the positioning error allowance set for all joints involved in an arm group semaphore held by the current task running `ResetEralw`, will revert to the default.

Macro definition

File `<pacman.h>` is required.

Related commands

`ResetCurLmt` and `SetEralw`

Notes

- (1) [For Ver. 1.4 or earlier] Write this library in a TAKEARMed task that has got robot arm semaphore. If you specify any joints not in the arm semaphore to `<JntNumber>`, then error [21F7 Cannot take arm semaphore] will result.
[For Ver. 1.5 or later] Write this library in a TAKEARMed task that has got an arm group. If you specify any joints not included in the arm group to `<JntNumber>`, then error [27D* Cannot take J* semaphore] will result.
- (2) Running the current limit release library `ResetCurLmt` also resets the positioning error allowance to the initial value.

Example

```
CALL ResetEralw(0)    'Reset positioning error allowance of all  
                      'joints to initial value.
```

3.2 Setting the Extended-Joint Parameters

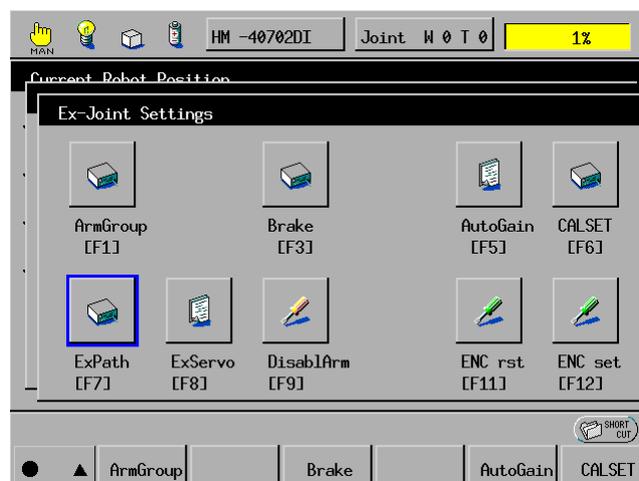
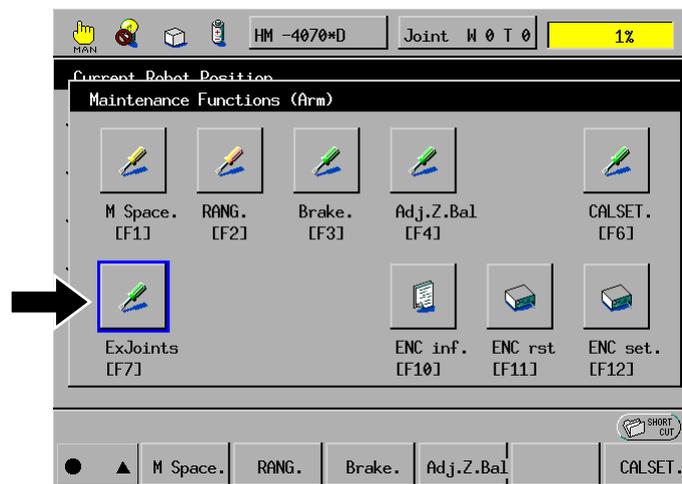
To use extended-joints, you need to set extended-joint parameters beforehand. There are two types of extended-joint parameters as described below, which can be set by using the teach pendant.

- (1) Extended-joint path parameters, which are provided for motion definitions (including speed, acceleration, and range of motion) of extended-joints.
- (2) Extended-joint servo parameters, which are provided for setting the gain and others of extended-joint servo system.

3.2.1 Calling up "Path Parameters for Ex-Joint" and "Servo Parameters for Ex-Joint" Windows

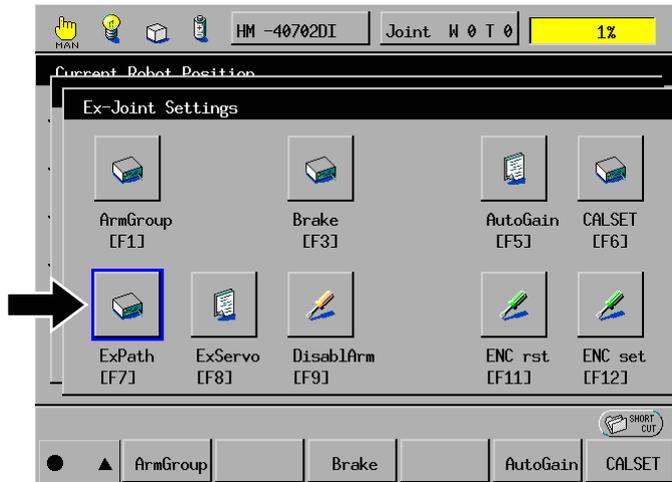
- (1) Calling up the "Ex-Joint Settings" window

Top Screen—[F2 Arm]—[F12 Maint.]—[F7 ExJoints]



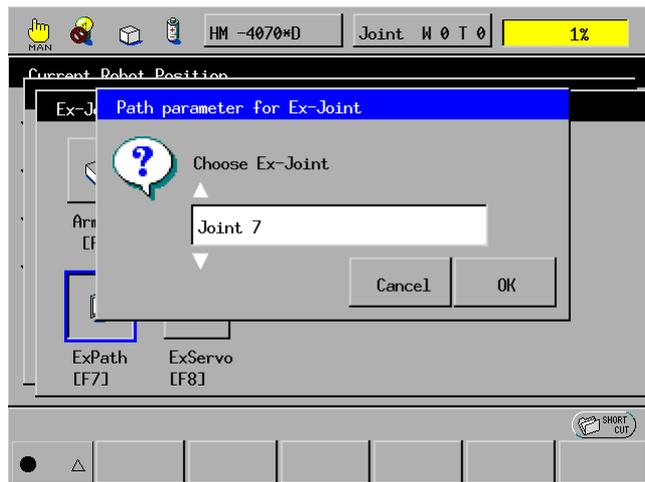
"Ex-Joint Settings" Window

(2) Changing the extended-joint path parameters

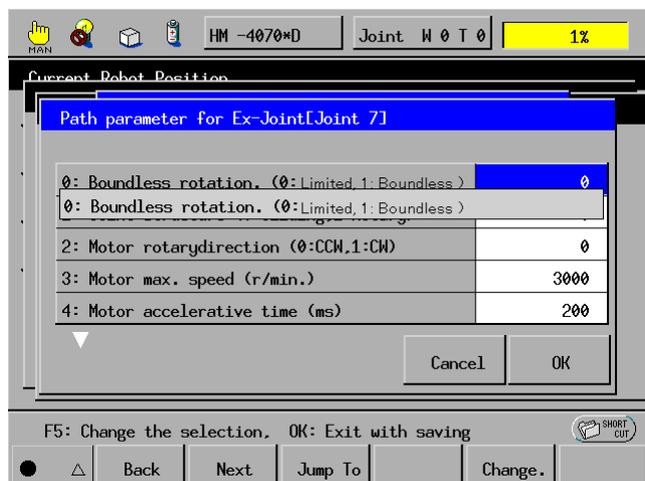


On the "Ex-Joint Settings" window shown above, press [F7 ExPath].

The "Path Parameters for Ex-Joint" window will appear as shown below. Choose the target joint (J7 in this example) by using the cursor keys or jog dial.



Press [OK]. The "Path Parameters for Ex-Joint[Joint 7]" window will appear as shown below. Change the path parameters and press [OK]. (For detailed setting procedure, refer to Subsection 3.2.2.)



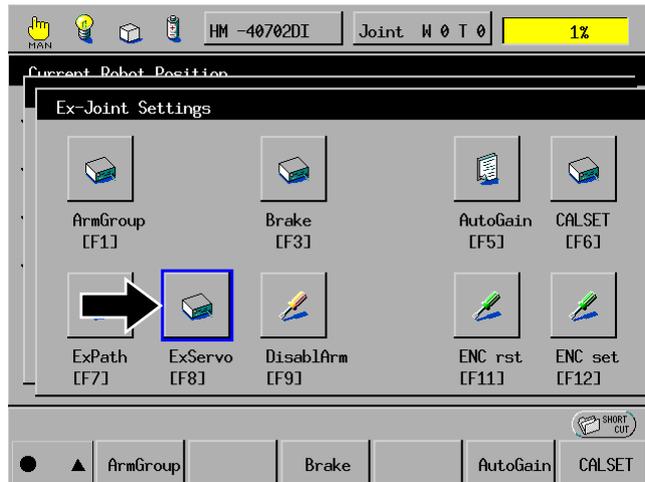
The path parameters are listed in Table 3.1 below.

NOTE: Some parameters will take effect after the controller power is turned off and then on.

Table 3.1 Extended-Joint Path Parameters

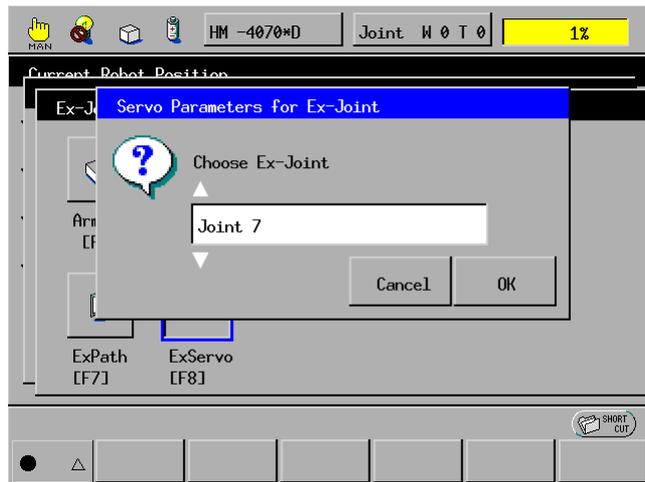
| Parameter name | Entry range | Factory default | Unit | Description | Remarks | Controller restart |
|--|--------------|-----------------|---|---|---|--------------------|
| Boundless rotation (0: Limited, 1: Boundless) | 0 or 1 | 0 | | If you want to rotate the motor 16384 times or more in the same direction, set this parameter to 1. | If 1 is set, the encoder joint number setting switch must be set to Unlatch. (Refer Sub-section 3.2.4.4.) | Needed |
| Joint structure (0: Sliding, 1: Rotary) | 0 or 1 | 1 | | If your optional mechanism to be connected to the specified motor has a linear joint, then set 0; if a rotary joint, set 1. | | Needed |
| Motor rotation direction (0: CCW, 1: CW) | 0 or 1 | 0 | | To convert the CCW rotation of the specified motor (when viewed from the load side) to the positive direction movement of the connected mechanism, set 0; to convert it to the negative one, set 1. | | Needed |
| Motor max. speed (rpm) | 1 to 5000 | 3000 | rpm | Set the maximum speed of the specified motor. | | Needed |
| Motor acceleration time (ms) | 1 min. | 200 | ms | Set the motor acceleration time required for the specified motor to reach the maximum speed. | | Needed |
| Gear ratio or lead (mm/r) | 0.00001 min. | 100 | For lead: mm/r | For rotary joints, set the deceleration ratio (motor rotation/joint rotation). For linear joints, set the lead (movement) per motor rotation. | Up to 100,000 may be set. But if a large value is set, the entered value may be different from the displayed one due to overflow. | Needed |
| Motion limit detection (0: Invalid, 1: Valid) | 0 or 1 | 1 | | To make the controller check the motion limit and issue an error if the specified joint is out of the range, set 1. | | Needed |
| Positive motion limit (deg.) (mm) | | 360 | For rotary joints: degrees For linear joints: mm | Set the positive motion limit. | | Not needed |
| Negative motion limit (deg.) (mm) | | -360 | For rotary joints: degrees For linear joints: mm | Set the negative motion limit. | | Not needed |
| CALSET position | | 0 | For rotary joints: degrees For linear joints: mm | Set the CALSET reference position. | | Not needed |

(3) Changing the extended-joint servo parameters

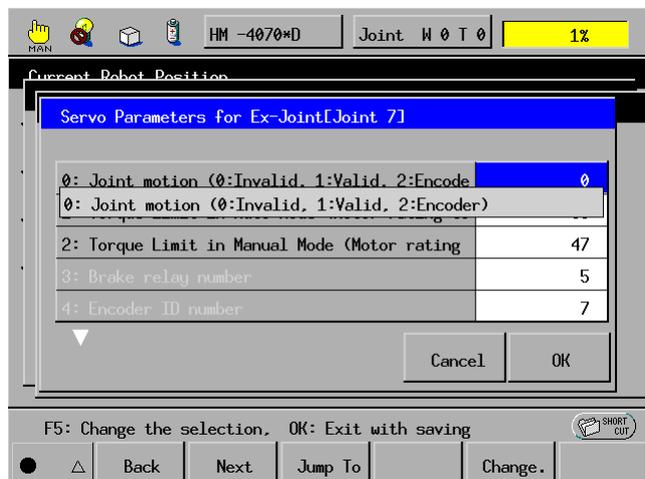


On the "Ex-Joint Settings" window shown above, press [F8 ExServo].

The "Servo Parameters for Ex-Joint" window will appear as shown below. Choose the target joint (J7 in this example) by using the cursor keys or jog dial.



Press [OK]. The "Servo Parameters for Ex-Joint[Joint 7]" window will appear as shown below. Change the servo parameters and press [OK]. (For detailed setting procedure, refer to Subsection 3.2.2.)



Chapter 3 Extended-Joint Related Function

The servo parameters are listed in Table 3.2 below.

NOTE: Some parameters will take effect after the controller power is turned off and then on.

Table 3.2 Extended-Joint Servo Parameters

| Parameter name | Entry range | Factory default | Unit | Description | Remarks | Controller restart |
|--|-------------|-----------------|------|--|--|--------------------|
| Joint motion (0: Invalid, 1: Valid, 2: Encoder) | 0 to 2 | | | To connect and drive a specified motor, set 1; to use the encoder only, set 2. | If "2: Encoder" is selected, turning the motor on will release the brake. CAUTION: If any unbalanced load is applied, the joint will move towards the load. | Needed |
| Torque limit in auto mode (Motor rating torque %) | 0 to 400 | 300 | % | Set the torque limit value to be applied in auto mode. | | Not needed |
| Torque limit in manual mode (Motor rating ratio %) | 0 to 400 | 150 | % | Set the torque limit value to be applied in manual mode. | | Not needed |
| Brake relay number | 0 to 8 | | | Displays the motor brake relay number. | No change allowed. | |
| Encoder ID number | 1 to 8 | | | Displays the encoder ID number. | No change allowed. | |
| Power module slot number | 1 to 8 | | | Displays the power module slot number. | No change allowed. | |
| Positional loop gain | 1 min. | 64 | | Set the response of the position control system. Increasing the value will decrease the positioning time. | The positioning loop gain can be converted in unit by Formula 3.2.3-1 given in Subsection 3.2.3. | Not needed |
| Positional loop feed forward gain (%) | 0 to 100 | 0 | % | Set the loop forward gain of the position control system. Increasing the value will decrease a positioning error and increase the response, but overshoot will easily occur. | | Not needed |
| Positioning error allowance (pulse) | 1 min. | 30000 | | Set the allowable value of positioning error. If a positioning error exceeding this allowable value occurs, an error will result. | Set the value that meets Formula 3.2.3-2 given in Subsection 3.2.3. | Not needed |

| Parameter name | Entry range | Factory default | Unit | Description | Remarks | Controller restart |
|--|-------------|---|------|---|---|--------------------|
| Speed proportional gain | 0 min. | 200 (for 200W or less) 400 (for 400W or greater) | | Set the response of the speed control system. Increasing the value will enable you to set a higher value of the positional loop gain. | The speed loop proportional gain can be converted to the speed response frequency in Hz by Formula 3.2.3-3 given in Subsection 3.2.3. | Not needed |
| Speed integral gain | 0 min. | 10 | | Set the integral compensation gain of the speed control system. Increasing the value will converge the speed deviation at the time of stop faster. | The speed loop integral gain can be converted to the time constant by Formula 3.2.3-4 given in Subsection 3.2.3. | Not needed |
| Filter parameter | 0 to 15 | 8 | | Set the primary delay filter band in the torque instruction section. Increasing the value will decrease the time constant of the low-pass filter. | | Not needed |
| Torque offset setting (Motor rating ratio %) | 0 to 100 | 0 | % | Set the torque offset value of the torque instruction value. If the motor undergoes any unbalanced load (movement towards the load), this offset will compensate it. | If you enable the gravity offset in auto gain tuning, the torque offset value will be automatically set. | Not needed |
| Motor capacity (1: 50W, 2: 100W, 3: 200W, 4: 400W, 5: 750W, 6: 1.5kW) | 1 to 6 | | | Display the connected motor capacity | No change allowed. | |

3.2.2 Detailed Description of Extended-Joint Parameter Setting

The extended-joint path parameters and extended-joint servo parameters should be set with extended-joint motors being connected.

(1) Resetting the encoder of an extended-joint motor

The encoder is not connected with a backup battery at the time of shipment, so the error message "J* encoder system down" or "J* encoder speed exceeded" will appear.

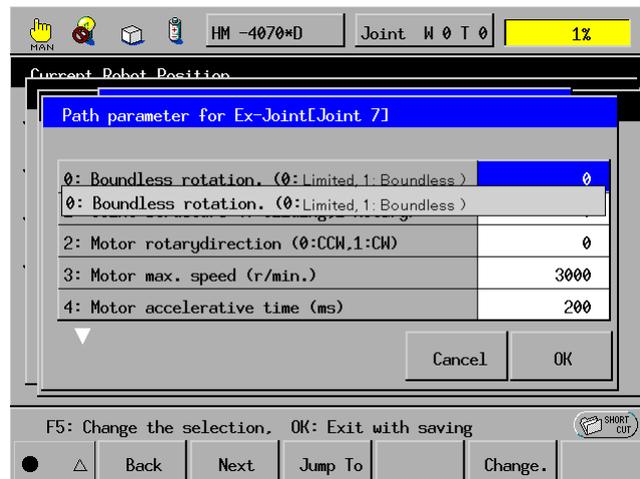
If this happens, reset the encoder (refer to Subsection 3.2.4.3) and restart the robot controller.

NOTE: When resetting the encoder, connect only the motor cable to the extended-joint support robot controller and be sure to disconnect the robot cable from it. This is to prevent the robot motor from getting reset.

(2) Setting the path parameters

For the calling-up procedure of the "Path Parameters for Ex-Joint" window, refer to Subsection 3.2.1.

(2-1) Boundless rotation

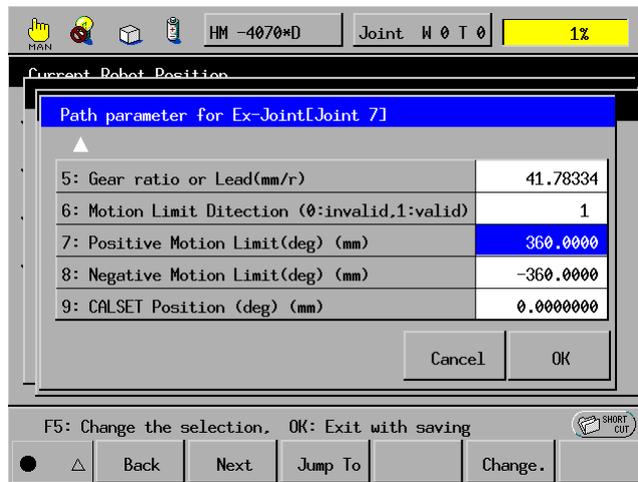
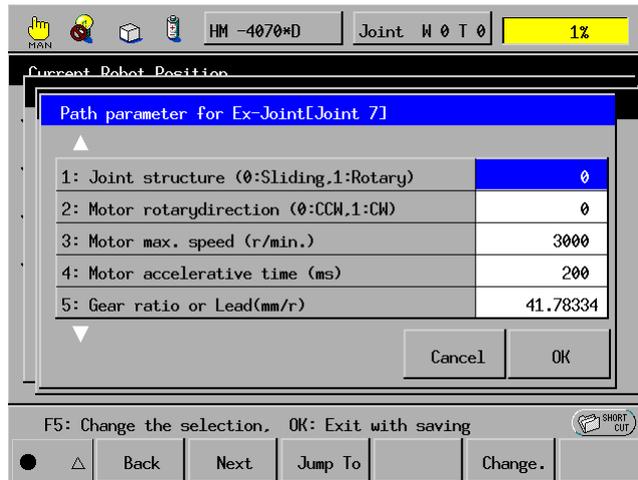


To rotate the motor 16,384 times or more in the same direction, you need to set the boundless rotation to "1: Boundless."

Specifying the boundless rotation requires the overflow error of the encoder to be set to "Unlatch." See the "Joint Number Setting Switch" window in Subsection 3.2.4.4 "Setting a joint ID to an extended-joint encoder."

NOTE: When setting a joint ID to a particular encoder, be sure to connect only the motor whose encoder joint ID should be set, to the extended-joint support robot controller. Disconnect the robot cable and other motor cables from the robot controller. The encoder ID setting operation will set all encoders being connected to the same ID.

(2-2) Setting the motion conditions



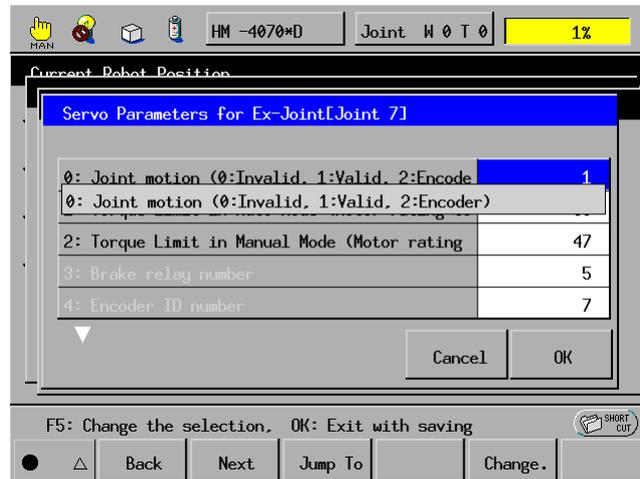
Set the motion-relation parameters--joint structure, motor rotation direction, motor maximum speed, motor acceleration time, gear ratio or lead, motion limit detection, positive motion limit, negative motion limit, and CALSET reference position.

(3) Setting the extended-joints servo parameters

For the calling-up procedure of the "Servo Parameters for Ex-Joint" window, refer to Subsection 3.2.1.

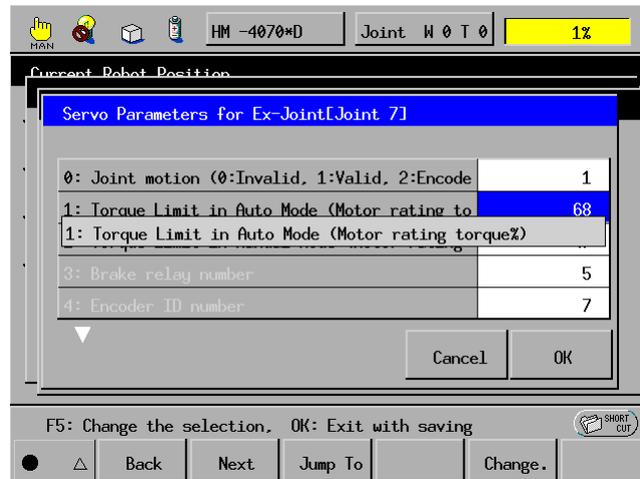
(3-1) Setting the joint motion

Set the joint motion to "1: Valid."



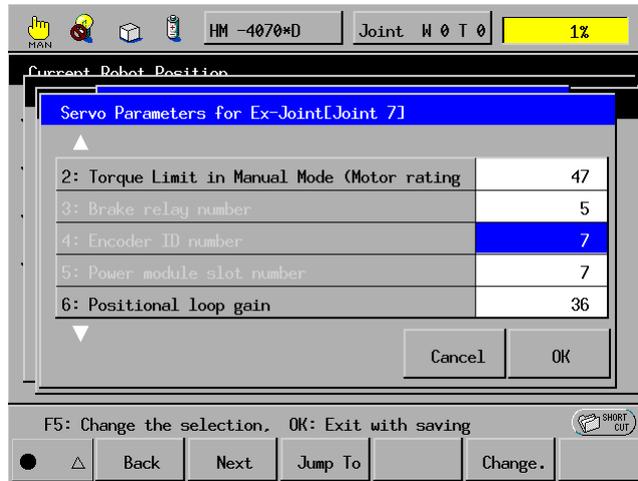
(3-2) Setting the torque limits

Set the torque limits in auto mode and in manual mode.



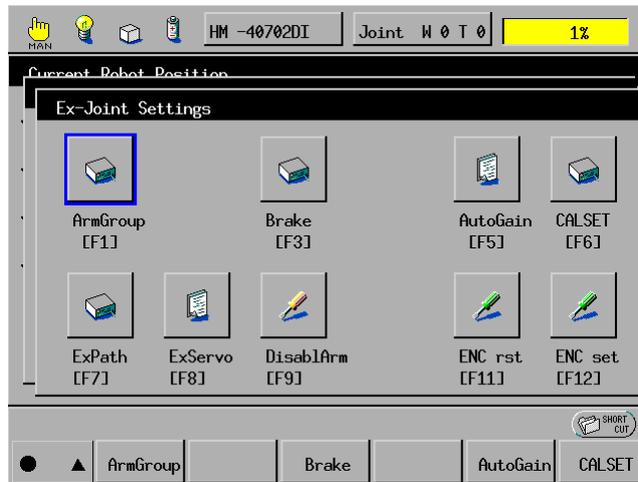
(3-3) Checking the encoder ID number, power module slot number, and motor capacity

Checks that the encoder ID number and power module slot number match the joint number. Also, check that the motor capacity is selected correctly.



(4) Registering extended-joints in an arm group

Register extended-joints into an arm group. Refer to Subsection 3.1.1 "Operating Procedures of the Extended-Joint Function, [4] Operating extended-joints by program."



After completion of steps (1) to (4), restart the robot controller.

(5) Checking the wiring

(5-1) Checking the brake wiring

If the extended-joint motor has a brake, release the brake and check that the brake of the specified joint will be released. For the brake releasing procedure, refer to Subsection 3.2.4.2 "Releasing and locking an extended-joint brake."

NOTE: Releasing the brake of a robot joint may drop the arm. Take care not to release the brakes of robot joints.

(5-2) Checking the encoder wiring

After releasing the brake of the extended-joint motor, apply external force to the motor and check that the data of the joint corresponding to the motor will change in the Current Robot Position window of the teach pendant.

(5-3) Checking the motor wiring

Turn the extended-joint motor on, set the motor speed at SP10, and check that you may drive the extended-joint manually in Joint mode.

If the motor vibrates abnormally or stops due to any error, check the wiring of the motor. If the wiring is correct, gradually decrease the positional loop gain and speed proportional gain of the extended-joint servo parameters.

(6) Executing CALSET

Release the brake of the extended-joint motor and move the optional mechanism connected to the motor to the CALSET reference position. Then execute CALSET in the CALSET reference position, referring to Subsection 3.2.4.1 "Performing CALSET operation on extended-joints."

NOTE: Performing the CALSET operation on a robot joint will change the reference angle of the robot. Take care not to execute CALSET on any robot joint.

After completion of CALSET operation, execute CAL operation.

(7) Checking the motion of the optional mechanism connected to the extended-joint motor

Run the optional mechanism connected to the motor manually in Joint mode and check that an error will be detected if the mechanism exceeds the positive or negative motion limit.

Also check that the actual movement amount matches the values displayed in the Current Robot Position window of the teach pendant. If not, check the gear ratio and lead.

3.2.3 Gain Tuning of Extended-Joints

In Subsection 3.2.2, you have set the motion conditions of extended-joints and checked the motion of the optional mechanism connected to the extended-joint motors manually in Joint mode. After that, proceed to the gain tuning for the servo system.

Tune the servo system according to the following two types of tuning methods:

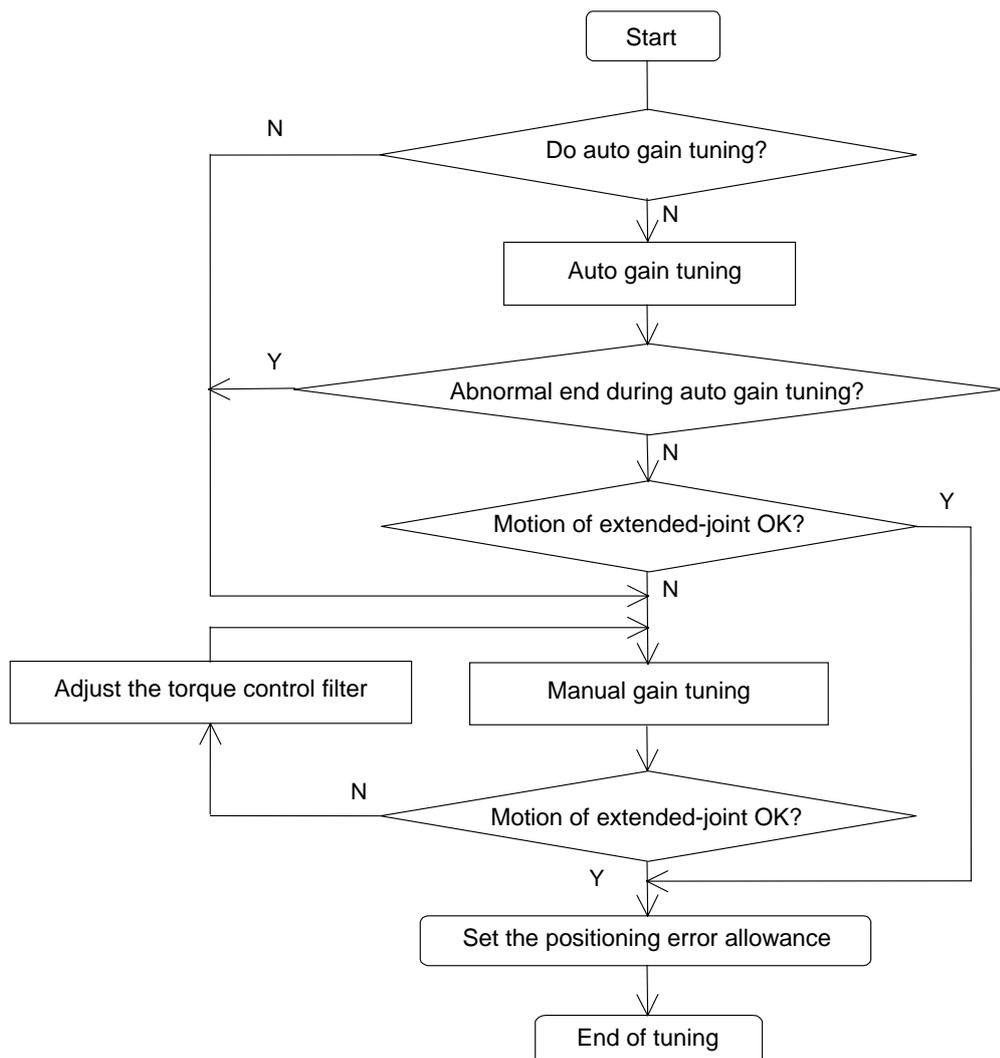
(1) Auto gain tuning

The controller performs acceleration/deceleration operation of the extended-joints according to the default pattern preset in the controller. Based on the motion of the extended-joints in that operation, the controller will estimate the inertia of payload and set the appropriate gain automatically.

(2) Manual gain tuning

The monitor function of the single-joint servo data monitors the motor speed control value, current motor speed, motor angle deviation, and torque control value. According to the monitored results, you may adjust the gain and torque control filter parameters for optimizing the motion of the extended-joints.

Follow the next flowchart to tune the servo system.



3.2.3.1 Auto gain tuning^[p65]

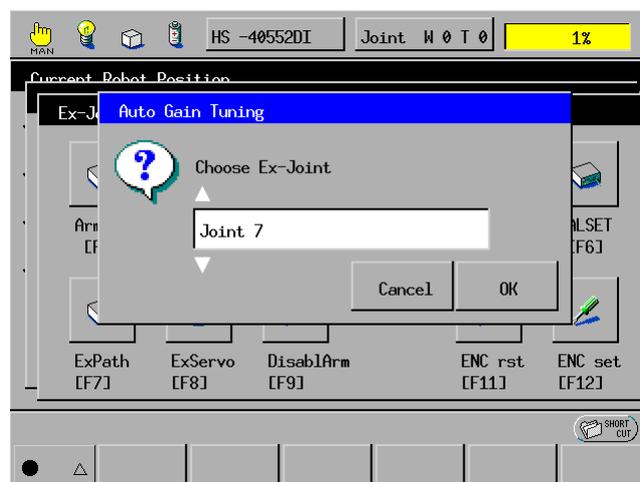
To implement auto gain tuning, your optional mechanism to be connected to the extended-joint motor should satisfy the requirements given in [1] below. Otherwise, some errors may occur and the auto gain tuning process may be interrupted. If such happens, implement manual gain tuning.

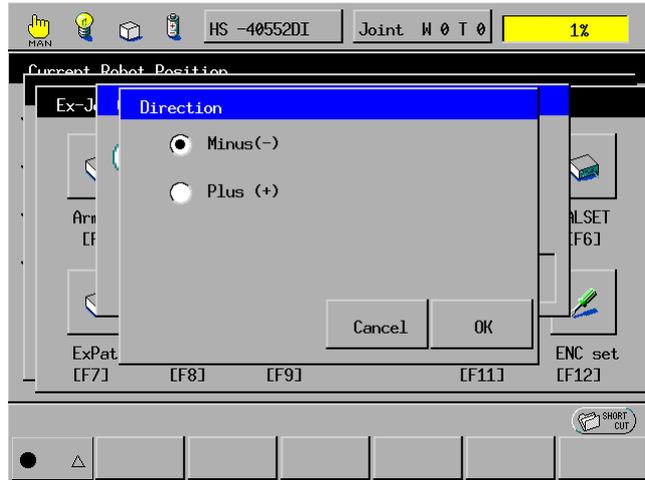
[1] Requirements for implementing auto gain tuning

- (1) The inertia of payload should be within 15 times that of the motor and should not deviate greatly.
- (2) The rigidity of the torque transmission mechanism (including motor and coupling) to be connected to the extended-joint motor should be high.
- (3) The backlash in the torque transmission mechanism should be minimized.
- (4) Rotating the motor in CCW and CW directions alternately two times each direction should result in no problem.

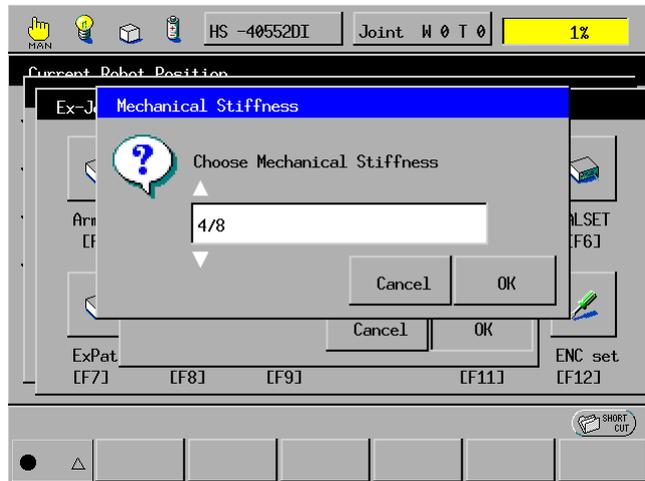
[2] Auto gain tuning procedure

- (1) Turn the motor power on and perform CAL
NOTE: If in Auto mode or Teach check mode, switch to Manual mode.
- (2) Get out of the motion range so that there will be no problem even if the motor rotates in CCW and CW directions alternately two times each direction.
- (3) On the teach pendant, call up the Ex-Joint Settings window.
Access: Top Screen—[F2 Arm]—[F12 Maint.]—[F7 ExJoints]
- (4) Press [F5 Auto Gain] to call up the Auto Gain Tuning window as shown below.
Choose the joint number that should undergo auto gain tuning and the motor rotation direction.



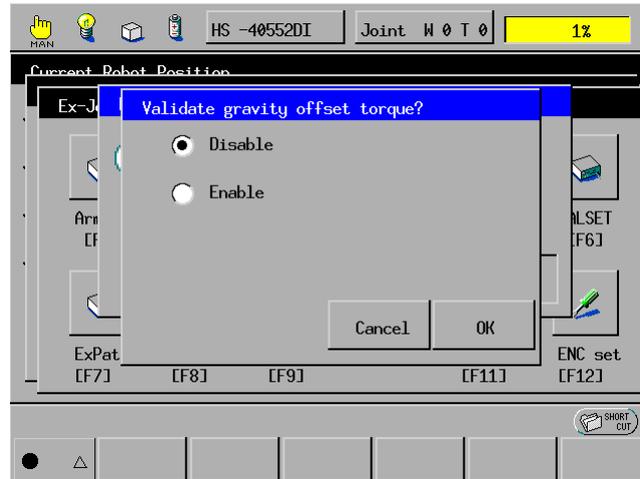


- (5) Select the mechanical rigidity, referring to the rigidity reference values listed below.



| Types of Torque Transmission Mechanisms | Mechanical Rigidity |
|---|---------------------|
| Ball screw direct connection | 4 to 8 |
| Ball screw with transmission mechanism | 3 to 7 |
| Timing belt | 3 to 6 |
| Gear or rack & pinion | 2 to 6 |
| Other mechanism with low rigidity | 1 to 3 |

- (6) Select whether the gravity offset torque should be enabled or disabled.



If an unbalanced load applies to the motor, be sure to enable the gravity offset torque.

NOTE: If you disable gravity offset torque when the motor undergoes any unbalanced load, then the extended-joint will drop in the gravity direction, causing an error. To implement auto gain tuning when an unbalanced load applies to the motor, be sure to enable the gravity offset torque.

If you enable the gravity offset torque for auto gain tuning, the controller will automatically calculate the torque offset included in extended-joint servo parameters. On the Ex-Joint Settings window, press [F8 ExServo] to call up the "Servo Parameters for Ex-Joint" window and then press [OK] to save the calculated torque offset value.

NOTE: If you turn the controller power off without saving the calculated torque offset value, then the value will be lost and the previous value will resume.

- (7) Hold down the deadman switch through all processes of auto gain tuning. Releasing it will interrupt auto gain tuning.

NOTE: During auto gain tuning, do not press any key on the teach pendant except for the deadman switch. Doing so will interrupt auto gain tuning.

NOTE: If the joint motion has been set to [2:Encoder] on the "Servo Parameters for Ex-Joint" window, then the error message "Not executable" will appear during auto gain tuning.

Chapter 3 Extended-Joint Related Function

- (8) In the dialog box shown below, confirm the conditions and press [OK]. In the confirmation window, press [OK]. Then auto gain tuning will start. The motor rotates in CCW and CW directions alternately two times each direction in two sequences to calculate a temporary servo loop gain.

After that, the motor will repeat the sequence up to 8 times to fine-tune the gain. If the gain is fixed within the eight sequences, auto gain tuning will complete.



- (9) After eight sequences of the above fine tuning operation, any of the following messages may display:

"Auto gain tuning warning 1": Overshoot found at the end of motion.

"Auto gain tuning warning 2": Slow settlement found at the end of motion.

"Auto gain tuning warning 3": Low-level oscillation found during motion.

If any of the above messages displays but there is no problem with the joint motion, then finish the gain tuning. If any abnormal noise or vibration is noted and there are some problems with the motion, then change the mechanical rigidity. After that, retry auto gain tuning or proceed to manual gain tuning.

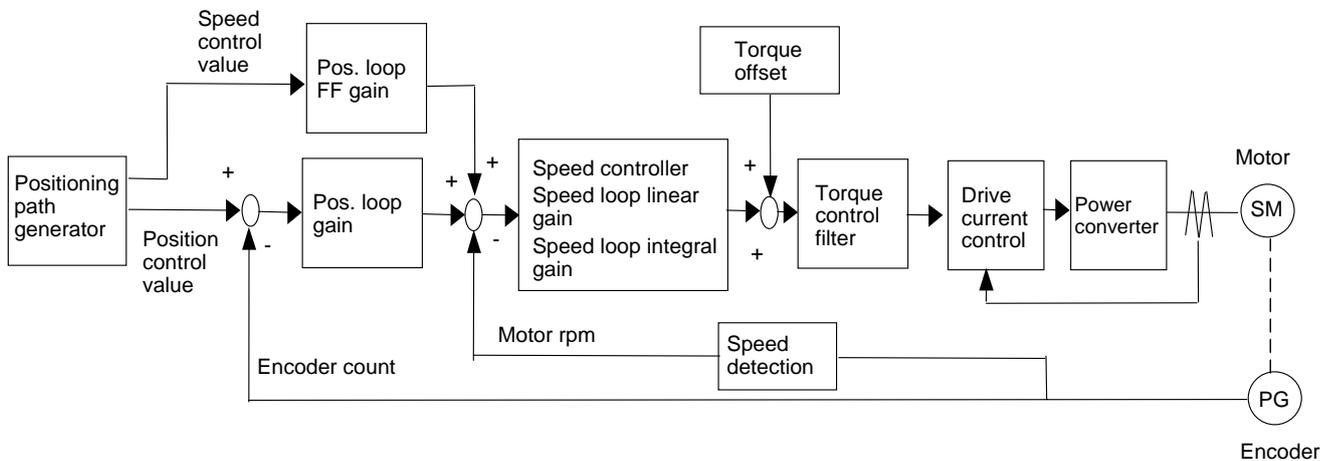
- (10) If you set higher mechanical rigidity for transmission mechanism having lower rigidity and vice versa, then an error may occur during auto gain tuning. Change the mechanical rigidity setting and retry auto gain tuning.

3.2.3.2 Manual gain tuning

You can manage the following parameters for manual gain tuning:

- (1) Positioning loop gain
- (2) Positioning loop feed forward gain
- (3) Positioning error allowance
- (4) Speed linear gain
- (5) Speed loop integral gain
- (6) Torque control filter
- (7) Torque offset

The block diagram for the servo system is shown below.



An electric servo loop system consists of the three feed back systems--positioning loop, speed control loop, and drive current loop. The inner the loop is, the quicker response required. If the response of an inner loop is not sufficiently high for an outer loop, then the overall system response degrades and vibrations or oscillations may occur in the extended-joint support system.

In this system, the innermost loop is the drive current loop and the outermost loop, the positioning loop.

You need to do gain tuning for the positioning loop and speed control loop. The drive current loop is designed to have sufficiently high response for all applications allowable to the extended-joint support system.

[1] Parameter details

(1) Positioning loop gain

Set the response of the positioning loop. The positioning loop gain is a dimensionless number, so it may be converted to the (1/s) unit according to the following formula:

$$\text{Positioning loop gain} \times 125/256 \text{ (1/s)} \quad \text{(Formula 3.2.3-1)}$$

For example, positioning loop gain 32 is equivalent to 15.625 (1/s).

Increasing the positioning loop gain will reduce the positioning time. However, increasing the gain exceeding the natural oscillation frequency of the connected mechanism will easily bring vibration or overshoot. If the natural oscillation frequency is 20 Hz, for instance, set the positioning loop gain to 20 (1/s), that is, approx. 41.

(2) Positioning loop feed forward gain

Set the speed feeding forward value of the positioning loop. Increasing the value will reduce the positioning error and increase the system response. Setting 100 may reduce the positioning error to almost 0 in constant speed operation. However, setting an excessively high value may easily cause vibration or overshoot in the system.

(3) Positioning error allowance

Set the positioning error allowance. If the actual positioning error exceeds the specified allowance, an error will occur. The positioning error allowance should satisfy the following formula:

$$\text{[Positioning error allowance]} > \frac{\text{[Maximum motor speed (rpm)]} \times (1.0 - \text{[Positioning loop forward gain (\%)]} \times 0.01)}{\text{[Positioning loop gain]} \times 524288/1875} \quad \text{(Formula 3.2.3-2)}$$

(4) Speed control linear gain

Set the response of the speed control system. Increasing the value will make it possible to set a high value to the positioning loop gain, thereby increasing the system response. The speed control linear gain to be set may be converted to the speed response frequency (in Hz) according to the following formula:

$$\text{Speed response frequency (Hz)} = \frac{\text{[Speed control linear gain]}}{\text{[Motor rotor inertia (kgm}^2\text{)]} + \text{Load inertia converted at motor joint (kgm}^2\text{)}} \times \frac{\text{[Drive current loop gain]}}{2\pi} \quad \text{(Formula 3.2.3-3)}$$

The drive current system gain is as listed below.

| Motor Model | Motor Rating | Drive Current System Gain |
|-------------|--------------|---------------------------|
| MQM012T2V | 100 W | 3.942E-05 |
| MQM022T2V | 200 W | 9.381E-05 |
| MQM042T2V | 400 W | 1.264E-04 |
| MQM082T2V | 750 W | 3.038E-04 |
| MQM152T2U | 1.5 kW | 2.785E-04 |

(5) Speed control integral gain

Set the integral compensation gain of the speed control system. You may convert the integral gain of the speed control loop into integral speed loop gain time constant (ms) according to the following formula:

$$\text{Integral speed loop gain time constant (ms)} = 0.5 \times \frac{[\text{Speed control linear gain}]}{[\text{Speed loop integral gain}]} \quad (\text{Formula 3.2.3-4})$$

Increasing the value will decrease the integral time constant, making the speed error converge faster at the end of joint motion. However, increasing the value for the connected transmission mechanism having lower rigidity will decrease the convergence of residual oscillation at the end of joint motion.

(6) Torque control filter

This value sets the band of the linear delay component for the torque control filter. The table below lists the relationship between the set and the band.

| Filter Set Value | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|----|----|----|----|
| Band (Hz) | 775 | 540 | 422 | 341 | 280 | 230 | 188 | 152 | 121 | 92 | 66 | 43 | 21 |

(7) Torque offset

This value gives an offset to the torque control value of the extended-joint support system. If the motor undergoes any unbalanced load due to the force of gravity, setting this value will compensate the torque caused by the unbalanced load. The maximum offset value you can set is equal to the rated output torque of the motor.

If you set a large torque offset at once, the connected mechanism may move in the preset direction immediately after the motor power is turned on. Gradually change the torque offset while confirming the current torque value and positioning error waveform in the next item "[2] Monitor of single-joint servo data."

As described in Subsection 3.2.3.1 "Auto gain tuning," the torque offset value will be automatically set if you enable the gravity offset torque in auto gain tuning.

[2] Monitor of single-joint servo data

This function allows you to monitor a specified joint servo data currently set in the robot controller with graphs in real-time.

(1) Monitoring capability

This function is capable of handling up to 1,250 samples of data at once. If the sampling interval is set to 1 ms, then you may monitor the servo data for 1.25 seconds; if 8 ms, you may monitor it for 10 seconds.

The following five types of data may be monitored, two types at a time:

1) Motor speed control value (rpm)

Shows the sampled control value of motor speed.

2) Current motor speed (rpm)

Shows the sampled current motor speed.

3) Motor angle deviation (pulse)

Shows the deviation between the actual motor angle and motor control angle.

4) Torque control value (%)

Shows the substantial torque control value; that is, (Torque control value - the Torque offset value). The unit is a ratio to the rated motor torque (%).

5) Motor current (%)

Shows the currently maximal motor drive current between the 3-phase driving lines. The unit is a ratio to the motor rated current.

(2) Defining the monitoring terms

To define the monitoring terms, call the single-joint servo data monitor definition library `SetMonitorCond` in your program. For details, refer to Subsection 3.2.5 "SetMonitorCond."

Once monitoring starts, the monitoring terms already defined can not be changed until the monitoring cycle has completed. Define all necessary monitoring terms before starting a monitoring cycle.

(3) Starting and stopping the monitoring cycle

To start monitoring, run the library `StartSrvMonitor`. To end it, run the library `StopSrvMonitor`. To clear the data collected in the monitoring cycle, run the library `ClearSrvMonitor`. For details, refer to Subsection 3.2.5, "StartSrvMonitor," "StopSrvMonitor," and "ClearSrvMonitor."

If the total number of data samples monitored in a monitoring cycle is 1250 or less, all data may be monitored. If it exceeds 1250 samples, the last 1250 data samples before the end of the cycle may be monitored and other data will be discarded.

If any error occurs and the motor being monitored is turned OFF during monitoring, then a maximum of 850 samples before the OFF and 400 samples after that may be monitored.

(4) Graphing the monitored data

Log Manager in WINCAPSII allows you to graph the monitored data on a PC screen. For details, refer to Subsection 4.3.8, "New Features in Log Manager."

(4.1) Read monitored data

To read the monitored data (as Servo Joint Log) into Log Manager, first establish connection between the PC and robot controller in Communications Setting Manager and choose "Import" from the File menu in Log Manager to call up the "Receive Table" dialog box. From the table, select the <Servo Joint Log> and receive it.

(4.2) Plot the graph of monitored data

In Log Manager, choose "Servo Joint Graph" from the Tools menu. Adjust the scale and offset of the graph and check the graphed data.

(5) Saving the monitored data into a file

Log Manager in WINCAPSII allows you to save the monitored data into a CSV file. For details, refer to Subsection 4.3.8, "New Features in Log Manager."

To save the data into a CSV file, choose "Export"—"Servo Joint Data" from the File menu.

[3] Operating procedure for manual gain tuning

(1) Initializing positioning loop gain

First set the positioning loop gain to almost the same value of that calculated from natural frequency of the connected mechanism.

If the natural frequency is 20 Hz, set the positioning loop gain to 41 which equals to $20 \times 256/125$ as calculated by Formula 3.2.3-1. If the natural frequency is unknown, use the default value 64.

(2) Tuning torque

If almost constant, unbalanced load (e.g., force of gravity) applies to the motor, then set the torque offset calculated from the load.

(3) Obtaining the limit of speed linear loop gain

While increasing the speed linear loop gain gradually, find the upper limit of the loop gain at which the connected mechanism will start producing abnormal noises or oscillations.

(4) Checking the effect of the torque control filter setting

First set the torque control filter parameter to "0" and find the upper limit of speed linear loop gain again. If the speed linear loop gain obtained here is lower than the previous one obtained in step (3), reset it to 8 (default).

(5) Determining the appropriate speed linear loop gain

Apply 80% of the limit obtained in steps (3) and (4) to the speed linear loop on the connected mechanism.

(6) Tuning the speed Integral loop gain

Gradually increase the speed integral loop gain so that the positioning time and peaks of overshoot and undershoot will be minimized to optimize the connected mechanism.

(7) Tuning the positioning loop gain

If the connected mechanism is still oscillatory after carrying out the procedure in step (6), then decrease the positioning loop gain.

If you decrease the positioning time further after tuning in steps (3) through (6), then gradually increase the positioning loop gain to the extent that no noise or oscillation will be produced.

(8) Tuning the positioning loop feed forward gain

If you further decrease the positioning time of the connected mechanism, gradually increase the positioning loop feed forward gain to the extent that no oscillation will be produced.

-
- (9) Checking operations of the connected mechanism in full motion range and in full speed range

Run the connected mechanism in the full motion range while changing the speed gradually. If any abnormal noises or vibrations occur at some particular points, then check whether the mechanism slides evenly.

If any abnormal noise occurs in some particular speed, tune the torque control filter parameter again and check whether the abnormal noise decreases.

If tuning-up of the mechanism and torque control filter parameter cannot suppress abnormal noises, then decrease the speed linear loop gain and speed integral loop gain in the same proportion. (It is convenient to use the quick tuning function for the speed control system gain described in the next item [4].)

If any vibration occurs in some particular speed, decrease the speed integral loop gain, positioning loop gain, and/or positioning loop feed forward gain.

NOTE: Servomotors recommended earlier in this manual will issue torque ripple 4 times per rotation. The torque transmission mechanism may also issue torque ripple specific times per rotation at its output shaft.

Therefore, the frequency of the torque ripple may vary according to the speed so as to become equal to the natural frequency of the connected mechanism.

If vibrations are large in some specific speed, decrease the speed integral loop gain, positioning loop gain and/or positioning loop feed forward gain as well as stated above.

[4] Quick tuning function for speed control system gain

As expressed in Formula 3.2.3-4, the ratio of the speed linear loop gain to speed integral loop gain makes the integral speed loop gain time constant. For fine tuning of the speed control system gain, therefore, change the speed linear loop gain and speed integral loop gain in the same proportion. This simultaneous and proportional adjustment of those gains is "Quick tuning function for speed control system gain." You may use this function with the teach pendant.

- (1) Calling up the Quick Loop Gain Tuning screen using the teach pendant

Access: Top Screen—[F2 Arm]—[F6 Aux.]—[F7 Config.]—[F3 Jump To]

Select #59.

- (2) Setting a value to "Gain Decreasing Ratio (J*)" for the joint (J*) to be tuned

The gain decreasing ratio is called Tuning Ratio. The robot controller automatically modifies the current speed linear loop gain and speed integral loop gain by the number of "Tuning Ratio" times.

The relationship between the value to be set and the tuning ratio is listed below.

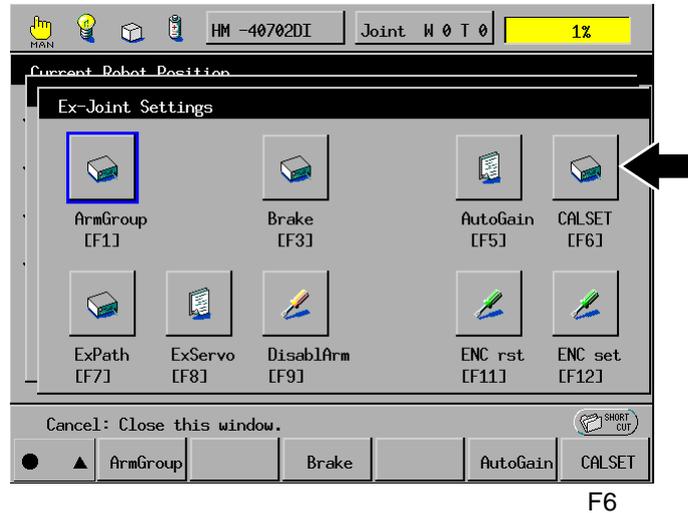
| | | | | | | | | | | | |
|--------------|-----|-----|-----|-----|-----|---|-----|-----|-----|-----|-----|
| Set Value | -5 | -4 | -3 | -2 | -1 | 0 | 1 | 2 | 3 | 4 | 5 |
| Tuning Ratio | 1.5 | 1.4 | 1.3 | 1.2 | 1.1 | 1 | 0.9 | 0.8 | 0.7 | 0.6 | 0.5 |

3.2.4 Extended-Joints Exclusive Operations_[p73]

3.2.4.1 Performing CALSET operation on an extended-joint_[p74]

(1) Calling up the Ex-Joint Settings window

Access: Top Screen—[F2 Arm]—[F12 Maint.]—[F7 ExJoints]

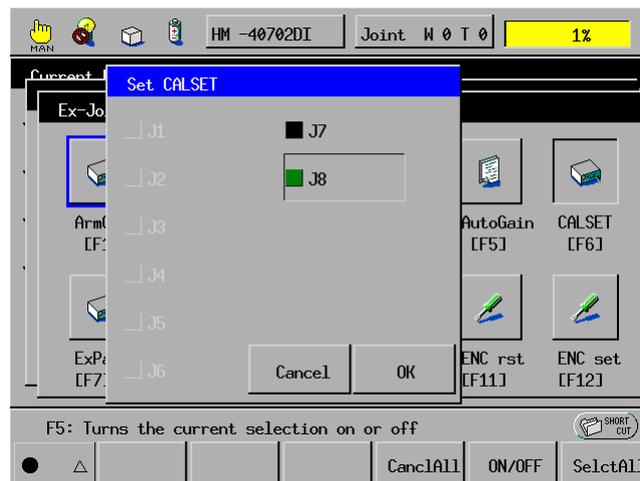


(2) Calling up the Set CALSET window

In the Ex-Joint Settings window, press [F6 CALSET] to call up the Set CALSET window.

(3) CALSETing a specified extended-joint

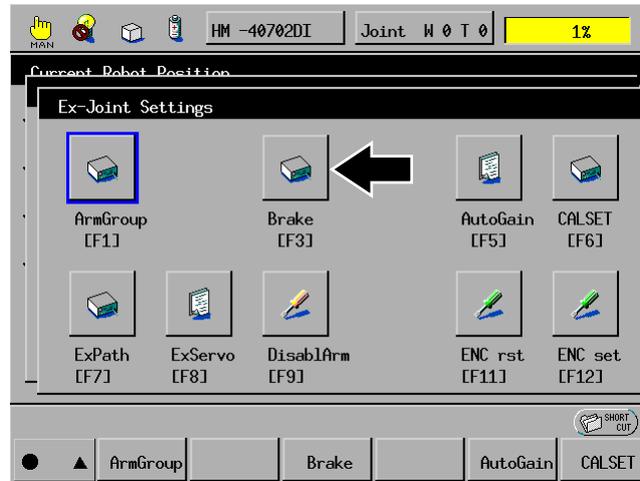
Select a joint to be CALSET and press [OK]. CALSET on the selected joint will start.



3.2.4.2 Releasing and locking an extended-joint brake_[p75]

(1) Calling up the Ex-Joint Settings window

Access: Top Screen—[F2 Arm]—[F12 Maint.]—[F7 ExJoints]



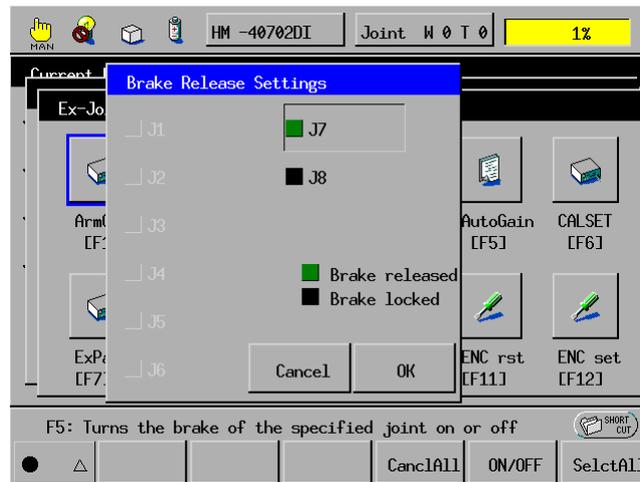
F3

(2) Calling up the Brake Release Settings window

In the Ex-Joint Settings window, press [F3 Brake] to call up the Brake Release Settings window. When a brake is released, the joint displays in green; when locked, it displays in black.

(3) Selecting a target joint to be released or locked

Select a target joint and press [F5 ON/OFF] to change the indicator color. To release the brake, turn the indicator color green; to lock it, turn it black.

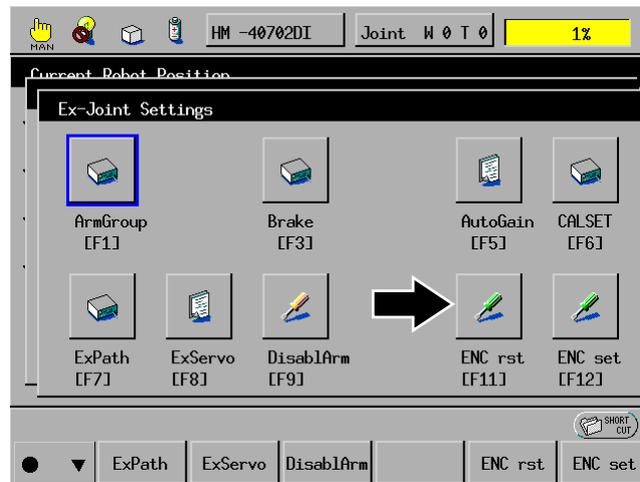


F5

3.2.4.3 Resetting an extended-joint encoder

(1) Calling up the Ex-Joint Settings window

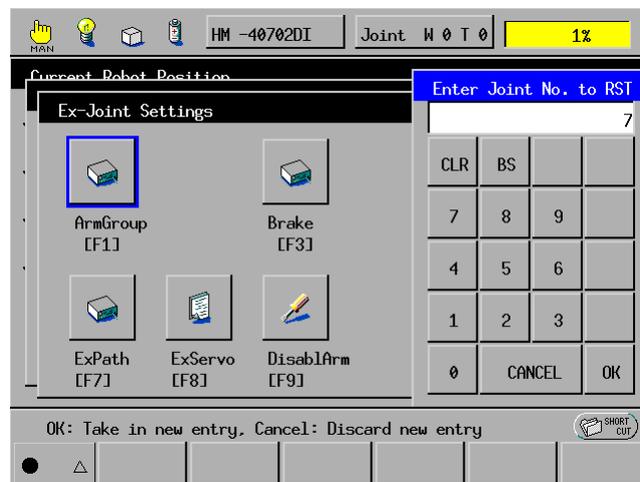
Access: Top Screen—[F2 Arm]—[F12 Maint.]—[F7 ExJoints]



(2) Entering the joint number of an encoder to be reset

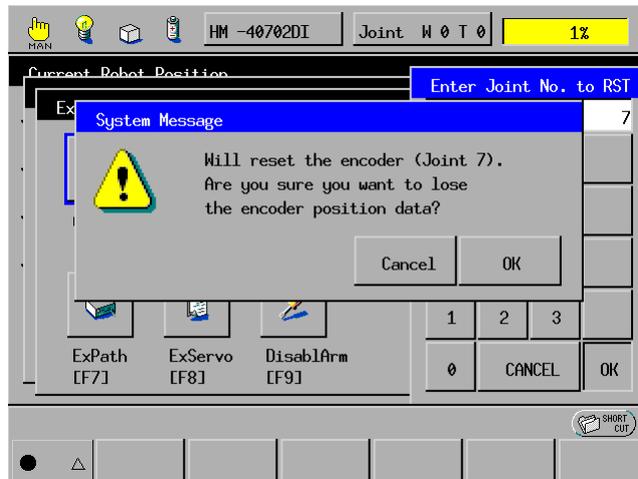
In the Ex-Joint Settings window, press [F11 ENC rst], and the numeric keypad for entering the joint number will appear as shown below.

Enter the joint number of an encoder to be reset and then press [OK]. Entering a robot joint number will cause an error.



(3) Confirming the specified joint number to be reset

The following confirmation message will appear. Check the specified joint number and press [OK].



3.2.4.4 Setting a joint ID to an extended-joint encoder^[p77]

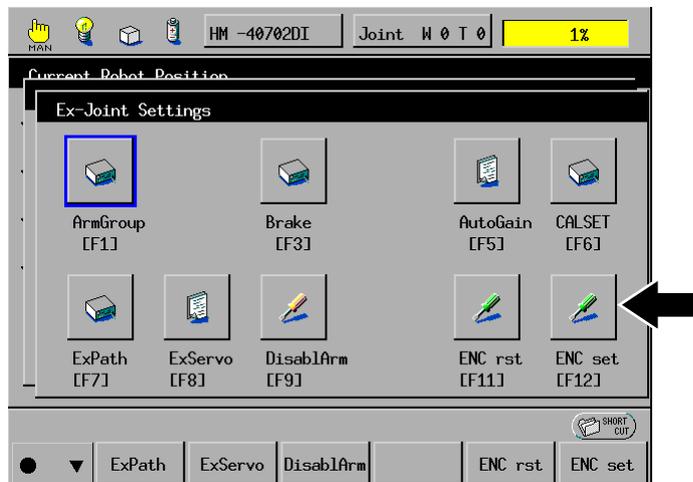
NOTE: When setting a joint ID, be sure to connect only the motor whose encoder joint ID should be set, to the extended-joint support robot controller. Disconnect the robot cable and other motor cables from the robot controller. The encoder ID setting operation will set the same ID to all encoders being connected.

NOTE: Disconnecting other encoders will cause the robot controller to issue an encoder error. Press the [LOCK] button on the teach pendant to place the controller in machine lock state and then clear the error.

NOTE: When setting a joint ID to the J8 encoder, connect an encoder dummy cap that comes with the extended-joint support robot controller, to the J7 encoder connector of the robot controller.

(1) Calling up the Ex-Joint Settings window

Access: Top Screen—[F2 Arm]—[F12 Maint.]—[F7 ExJoints]

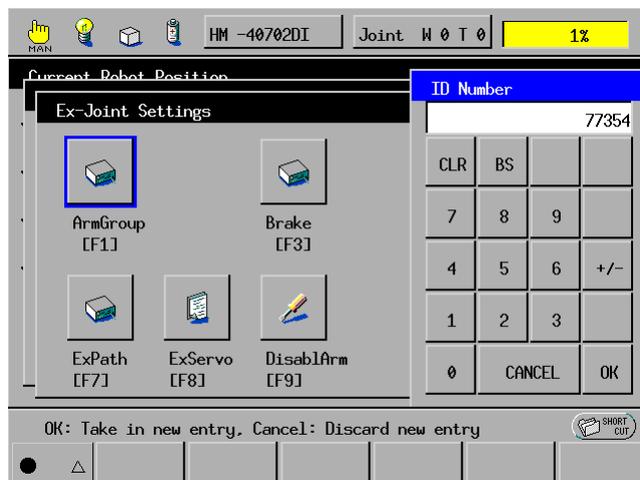


F12

(2) Entering a password

In the Ex-Joint Settings window, press [F12 ENC set], and the numeric keypad for entering a password will appear as shown below.

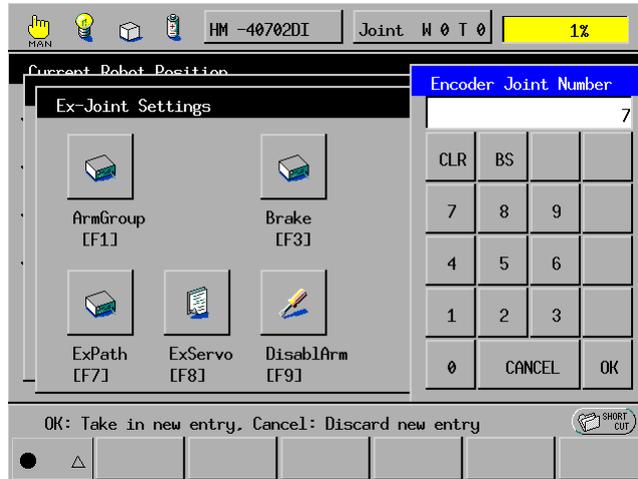
Enter the password "77354" and then press [OK].



(3) Entering an extended-joint ID number

The numeric keypad for entering an extended-joint ID number will appear as shown below.

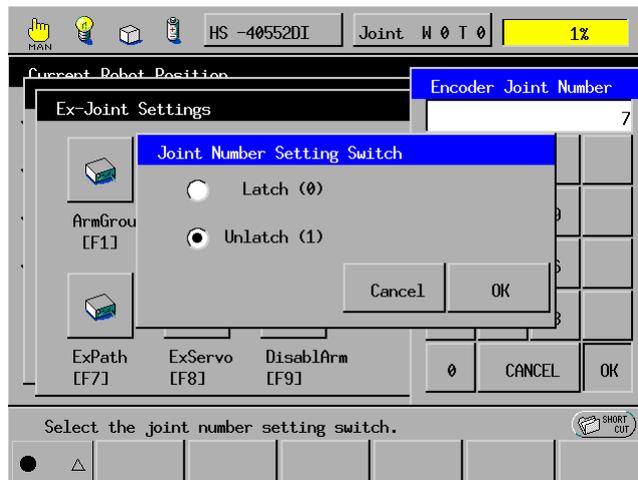
Enter an extended-joint ID number to be set and then press [OK].



(4) Latching/Unlatching the joint number setting switch

NOTE: In the window shown below, usually select "Latch" and press [OK].

Only when you want to rotate the motor 16,384 times or more in the same direction, select "Unlatch" and press [OK]. Also you need to set the boundless rotation to "1: Boundless" in the "Path Parameters for Ex-Joint" window.



(5) Confirming the specified joint ID number to be set

The confirmation message will appear. Check the specified joint ID number and press [OK].

If more than one encoder is connected, specifying a robot joint number will result in an error.

3.2.4.5 Enabling/disabling the robot arms

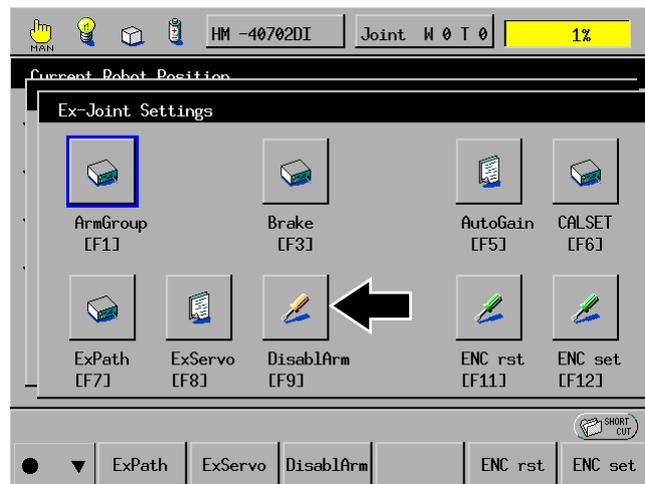
In adjustment of extended-joints, if you want to turn only extended-joint motors on for operational check without turning robot arm motors on, then disable the robot arm.

NOTE: Once you change this robot arm setting, be sure to restart the robot controller. Otherwise, an encoder-related error may occur.

NOTE: If the robot arm is disabled, turning the motor power on will not start the robot arm motors.

(1) Calling up the Ex-Joint Settings window

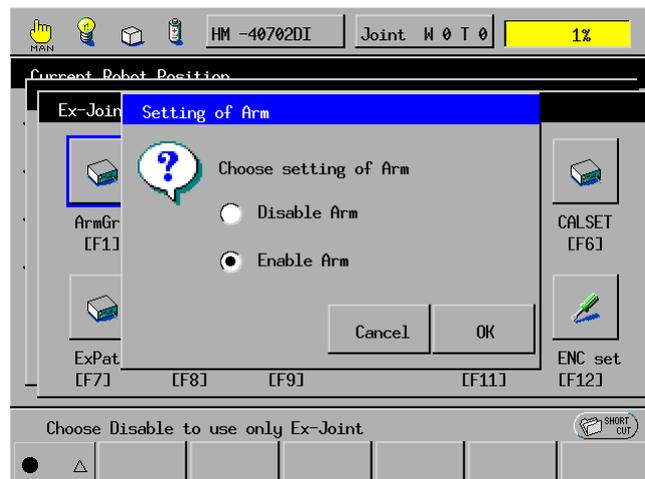
Access: Top Screen—[F2 Arm]—[F12 Maint.]—[F7 ExJoints]



F9

(2) Enabling/disabling the robot arm

In the Ex-Joint Settings window, press [F9 DisableArm], and the current arm setting will appear as shown below. Select the desired setting and press [OK].



3.2.5 Extended-Joint Parameter Setting Commands_[p79]

NOTE: Commands described in this section are used also for general robots.

[1] Single-Joint Servo Data Monitor Commands (Library)

SetMonitorCond

Function

Sets the monitoring conditions for single-joint servo data monitor. (For Ver. 1.5 or later)

Syntax

```
SetMonitorCond(<JntNumber>,<MonitorData1>,<MonitorData2>,<SampInterval>)
```

Description

SetMonitorCond sets the joint number to be monitored, monitor data (up to 2 types allowed per command), and sampling interval in ms as monitoring conditions.

The following five types of data may be monitored, two types at a time, by specifying <MonitorData1> and <MonitorData2>:

| <MonitorData1> and <MonitorData2> | Data to be monitored |
|-----------------------------------|--|
| 0 | Motor speed control value in rpm |
| 1 | Current motor speed (Actual speed) in rpm |
| 2 | Motor torque control value (excluding torque offset) in ratio (%) to the rated value |
| 3 | Motor rotation angle error (Motor angle control value - Actual motor angle value) in pulses |
| 4 | Motor current absolute value (Maximum value out of three absolute values detected from all 3 phases of the motor.) in ratio (%) to the rated value |

<SampInterval> must be set in ms as an integer between 1 and 8.

Macro definition

Not needed.

Related commands

ClearSrvMonitor, StartSrvMonitor, and StopSrvMonitor

Notes

- (1) If this library executes following the monitor start library StartSrvMonitor, the error "6001: Not executable" will result. Be sure to set the monitoring conditions before starting monitor.
- (2) If any of the joint number, data types, and sampling interval entered is wrong, the error message "The entered value is out of the range." will result. Correct those monitoring conditions you entered.

Example

```
CALL SetMonitorCond(7,0,3,4) 'For getting speed control value and
                             'motor angle error of J7 every 4 ms.
CALL StartSrvMonitor        'Start monitoring data.
```

StartSrvMonitor

Function

Starts monitoring single-joint servo data. (For Ver. 1.5 or later)

Syntax

```
StartSrvMonitor
```

Description

`StartSrvMonitor` fetches a maximum of 1250 samples of single-joint servo data until `StopSrvMonitor` executes.

Macro definition

Not needed.

Related commands

`ClearSrvMonitor`, `SetMonitorCond`, and `StopSrvMonitor`

Notes

- (1) If the total number of data samples monitored in a monitoring cycle is 1250 or less, all data may be monitored. If it exceeds 1250 samples, the last 1250 data samples before the end of the cycle may be monitored and other data will be discarded.
- (2) If any error occurs and the motor being monitored is turned OFF during monitoring, then a maximum of 850 samples before the OFF and 400 samples after that may be monitored.
- (3) No data may be monitored when the target motor is off. Execute this command with the motor power on.

Example

```
CALL SetMonitorCond(7,0,3,4) 'For getting speed control value and  
                             'motor angle error of J7 every 4 ms.  
CALL StartSrvMonitor        'Start monitoring data.
```

StopSrvMonitor

Function

Stops monitoring single-joint servo data. (For Ver. 1.5 or later)

Syntax

```
StopSrvMonitor
```

Description

In duration from execution of `StartSrvMonitor` to that of `StopSrvMonitor`, a maximum of 1250 samples of data may be obtained.

Macro definition

Not needed.

Related commands

`ClearSrvMonitor`, `SetMonitorCond`, and `StartSrvMonitor`

Notes

- (1) If the total number of data samples monitored in a monitoring cycle is 1250 or less, all data may be monitored. If it exceeds 1250, the last 1250 data samples before the end of the cycle may be monitored and other data will be discarded.
- (2) If any error occurs and the motor being monitored is turned OFF during monitoring, then a maximum of 850 samples before the OFF and 400 samples after that may be monitored.

Example

```
CALL StopSrvMonitor      'End monitoring data.
```

ClearSrvMonitor

Function

Initializes the pointer of data obtained by the single-joint servo data monitor function. (For Ver. 1.5 or later)

Syntax

```
ClearSrvMonitor
```

Description

`ClearSrvMonitor` initializes the pointer of data already obtained and starts monitoring new data up to 1250 samples.

Macro definitions

Not needed.

Related commands

`ClearSrvMonitor`, `SetMonitorCond`, and `StartSrvMonitor`

Notes

In duration from execution of `ClearSrvMonitor` to that of `StopSrvMonitor`, if the total number of data samples monitored is 1250 or less, all data may be monitored. If it exceeds 1250, the last 1250 samples before the end of the monitoring cycle may be monitored and other data will be discarded.

Example

```
CALL StartSrvMonitor      'Start monitoring data.
      :
      :
CALL ClearSrvMonitor      'Clear monitored data after
      :                    'execution of StartSrvMonitor.
      :
CALL StopSrvMonitor       'Stop monitoring data.
      :                    '(Data between ClearSrvMonitor and
      :                    'StopSrvMonitor processes are monitored.)
```

[2] Operation Termination Commands

MotionSkip

Function

Aborts running motion commands. (For Ver. 1.5 or later)

Syntax

```
MotionSkip
```

Description

`MotionSkip` aborts motion commands running in the task in which the `MotionSkip` executes.

Macro definition

Not needed.

Related commands

`GetJntData` and `GetSrvData`

Notes

- (1) Execute this command in a TAKEARMed task that holds an arm semaphore. If not in a TAKEARMed task, the error "Not executable" will result.
- (2) Executing `MotionSkip` in a robot motion task will abort robot joint motion commands. Executing it in an extended-joint motion task will abort extended-joint motion commands.

If `MotionSkip` executes in a motion task holding an arm group involving both robot joints and extended-joints, then both the robot and extended-joint motions will be aborted.

Example

```
defjnt lj1
defsng lf1
move p,P1,next
lj1=GetSrvState(2)      'Get errors of each joint rotation angle.
lf1=ABS(JOINT(2,lj1))   'Select rotation error of J2.
if lf1 > 10000 then
    CALL MotionSkip     'If the rotation error of J2 exceeds 10000
                        '(in pulses), then abort motion commands.
endif
```

MotionComp

Function

Judges whether execution of running motion commands is complete. (For Ver. 1.5 or later)

Syntax

```
MotionComp(<MotionCommandComplete>)
```

Description

If `MotionComp` judges that execution of running motion commands is complete, then it returns "1" in `<MotionCommandComplete>`.

This command checks motion commands running in the task in which the `MotionComp` executes. It is not applicable to motion commands in any other tasks.

If a motion command has an encoder value check option, then `MotionComp` will interpret the moment when the encoder count is converged within the positioning error allowance as completion of the motion command. For other operations, if motion control to the servo loop disappears, then `MotionComp` will judge that the command is complete.

Macro definition

Not needed.

Related commands

`GetJntData`, `GetSrvData`, and `MotionSkip`

Notes

- (1) Execute this command in a TAKEARMed task that holds an arm semaphore. If not in a TAKEARMed task, the error "Not executable" will result.
- (2) Executing `MotionComp` in a robot motion task will judge whether robot motion commands are complete. Executing it in an extended-joint motion task will judge whether extended-joint motion commands are complete.

If `MotionComp` executes in a motion task holding an arm group involving both robot joints and extended-joints, then completion of both the robot and extended-joint motions will be judged.
- (3) When the motion is on Halt, `MotionComp` will interpret it as operation being in progress.
- (4) If you use a local variable for `<MotionCommandComplete>`, the local variable must be reset to "0" beforehand.

Example

```
defint comp=0           'Initialize motion command completion status.
defjnt lj1
defsnrg lf1
move p,P1,next
DO
  lj1=GetSrvState(2)    'Get error of each joint rotation.
  lf1=ABS(JOINT(2,lj1)) 'Select the rotation error of J2.
  if lf1 > 10000 then
    CALL MotionSkip     'If the rotation error of J2 exceeds 10000 (in pulses),
                       'then abort motion commands and end the loop.
  EXIT DO
endif
CALL MotionComp(comp)
LOOP UNTIL comp=1      'Loop until the end of motion commands.
```

[3] Internal Servo Data Get Commands

GetSrvData

Function

Gets the internal servo data of robot joints. (For Ver. 1.5 or later)

Syntax

<InternalServoData> = GetSrvData(<DataNumber>)

Description

GetSrvData gets the internal servo data specified by <DataNumber> into <InternalServoData>.

<InternalServoData> is a joint type data of robot. <DataNumber> should be any of the following:

| <DataNumber> | <InternalServoData> |
|--------------|--|
| 1 | Current motor speed (Actual speed) in rpm |
| 2 | Motor rotation angle error in pulses |
| 4 | Motor current absolute value in ratio (%) to the rated value |
| 5 | Motor torque control value (excluding torque offset) in ratio (%) to the rated value |
| 8 | Joint position or angle control value in mm or degrees |
| 17 | Tool-end speed (3 position elements only in the work coordinates) in mm/s |
| 18 | Tool-end positioning speed (3 position elements only in the work coordinates) in mm |
| 19 | Tool-end speed (3 position elements only in the tool coordinates) in mm/s |
| 20 | Tool-end positioning speed (3 position elements only in the tool coordinates) in mm |

Macro definition

Not needed.

Related commands

GetJntData

Notes

- (1) Data numbers other than those given above are reserved. Do not use any other number other than the above, although no error will result if you specify any number up to 30.
- (2) If you attempt to fetch the servo data when the single-joint servo data monitor is running, the fetching process may become very slow. Take care when using the single-joint servo data monitor.
- (3) If you change <DataNumber>, the modification may take time. Do not change it so frequently.
- (4) Execute this command in a TAKEARMed task that holds an arm semaphore. If not in a TAKEARMed task, the error "21F7: Cannot take arm semaphore" will result.

Example

```
defjnt vel
defsnq absv,xvel,yvel,zvel
vel=GetSrvData(17)           'Get tool-end speed.
xvel=JOINT(1,vel)           'Select X component in work coordinates.
yvel=JOINT(2,vel)           'Select Y component in work coordinates.
zvel=JOINT(3,vel)           'Select Z component in work coordinates.
absv = SQR(xvel*xvel+yvel*yvel+zvel*zvel) 'Calculate total tool-end speed.
```

GetJntData

Function

Gets the internal servo data of a specified joint. (For Ver. 1.5 or later)

Syntax

```
<JntInternalServoData> = GetJntData(<DataNumber> , <JntNumber>)
```

Description

GetJntData gets the internal servo data (specified by <DataNumber>) of a joint specified by <JntNumber> into <JntInternalServoData>.

<JntInternalServoData>.is a floating point type data of the specified joint. <DataNumber> should be any of the following:

| <DataNumber> | <JntInternalServoData> |
|--------------|--|
| 1 | Current motor speed (Actual speed) in rpm |
| 2 | Motor rotation angle error in pulses |
| 4 | Motor current absolute value in ratio (%) to the rated value |
| 5 | Motor torque control value (excluding torque offset) in ratio (%) to the rated value |
| 8 | Joint position or angle control value in mm or degrees |

Macro definition

Not needed

Related commands

GetSrvData

Notes

- (1) Data numbers other than those given above are reserved. Do not use any other number other than the above, although no error will result if you specify any number up to 30.
- (2) If you attempt to fetch the servo data when the single-joint servo data monitor is running, the fetching process may become very slow. Take care when using the single-joint servo data monitor.
- (3) If you change <DataNumber>, the modification may take time. Do not change it so frequently.
- (4) Execute this command in a TAKEARMed task that holds an arm semaphore. If not in a TAKEARMed task, the error "21F7: Cannot take arm semaphore" will result.

Example

```
defsng vel  
vel=GetJntData(1,7)      'Get motor speed of J7.
```

DENSO ROBOT

VM-6083D/VM-60B1D/VS-E/HM-E/HS-E SERIES

Extended-Joints Support

SUPPLEMENT

First Edition November 2002

Second Edition January 2006

DENSO WAVE INCORPORATED
Factory Automation Division

1H**C

The purpose of this manual is to provide accurate information in the handling and operating of the robot. Please feel free to send your comments regarding any errors or omissions you may have found, or any suggestions you may have for generally improving the manual.

In no event will DENSO WAVE INCORPORATED be liable for any direct or indirect damages resulting from the application of the information in this manual.

