

# ***DENSO ROBOT***

Vertical articulated

**V\* SERIES**

Horizontal articulated

**H\*-G SERIES**

Cartesian coordinate

**XYC-4G SERIES**

Integrated compact type

**XR-G SERIES**

**SUPPLEMENT**

**Extended-Joints Support**

Copyright © DENSO WAVE INCORPORATED, 2007-2011

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 manual covers:**

the following robot systems configured with an extended-joint support robot controller.

- 
- VM-G series
  - VS series
  - VP-G series
  - HM-G series
  - HS-G series
  - XYC-4G series
  - XR-G series
- 

## **Important**

To ensure operator safety, be sure to read the precautions and instructions in "SAFETY PRECAUTIONS."

This product is not evaluated for EMC directive. When you export this product to Europe, it needs to be compliant with EMC directive as a facility.



## Contents

### Commands Listed in Alphabetical Order

#### Chapter 1 Overview

1.1	Configuration of Extended-Joint Support System.....	1-1
1.2	Extended-Joint Related Functions.....	1-2

#### Chapter 2 Robot Components of Extended-Joint Support System

2.1	Configuration of Extended-Joint Support System.....	2-1
2.1.1	System Configuration .....	2-1
2.1.2	Precautions for Safe Use of the Robot System.....	2-2
2.2	Extended-Joint Support Controller .....	2-3
2.2.1	Customized Controller.....	2-3
2.2.2	Connectors on Extended-Joint Support Controller .....	2-4
2.2.3	External Dimensions of Extended-Joint Support Controller .....	2-5
2.2.4	Controller Specifications .....	2-6
2.2.5	Total Output Capacity for Extended-Joint Motors.....	2-9
2.2.6	Connecting Cables to the Robot Controller.....	2-13
2.3	Optional Components for Extended-Joint Support System.....	2-14
2.4	Choosing AC Servomotors .....	2-16
2.4.1	AC Servomotors.....	2-16
2.4.2	External Dimensions of AC Servomotors .....	2-23
2.4.3	Safety Rules for Using Extended-Joint Motors .....	2-29

#### Chapter 3 Extended-Joint Related Functions

3.1	Extended-Joint Related Functions.....	3-1
3.1.1	Operating Procedures of the Extended-Joint Related Functions.....	3-1
3.1.2	Commands Supporting Extended-Joints .....	3-39
3.2	Configuring the Extended-Joint Parameters .....	3-72
3.2.1	Calling up "Path Configuration Parameters" and "Servo Configuration Parameters" Windows.....	3-72
3.2.2	Detailed Description of Extended-Joint Parameter Setting.....	3-78
3.2.3	Gain Tuning of Extended-Joints .....	3-84
3.2.4	Extended-Joints Exclusive Operations .....	3-97
3.2.5	Extended-Joint Parameter Setting Commands.....	3-102

## Chapter 4 Getting Started with Extended-Joints (Rack & Pinion)

4.1	Checking the Cable Connection .....	4-1
4.2	Powering On the Controller.....	4-1
4.3	Disabling the Robot Arm.....	4-2
4.4	Restarting the Controller.....	4-2
4.5	Configuring the Extended-Joint Servo Parameters .....	4-3
4.6	Resetting the Extended-Joint Encoder .....	4-5
4.7	Restarting the Controller.....	4-6
4.8	Checking the Encoder Info about Extended-Joints .....	4-6
4.9	Configuring the Extended-Joint Parameters .....	4-7
4.10	Restarting the Controller.....	4-8
4.11	Checking the Encoder Feedback .....	4-8
4.12	Running the Extended-Joint Manually .....	4-9
4.13	Moving to the CALSET Position .....	4-9
4.14	Configuring Other Extended-Joint Parameters .....	4-10
4.15	Restarting the Controller.....	4-11
4.16	Performing CALSET Operation .....	4-12
4.17	Setting up an Arm Group .....	4-13
4.18	Restarting the Controller.....	4-14
4.19	Checking the Extended-Joint Motion .....	4-14
4.20	Performing Auto Tuning .....	4-14
4.21	Checking the Extended-Joint Motion in Running-in.....	4-17
4.22	Enabling the Robot Arm .....	4-18

## Commands Listed in Alphabetical Order

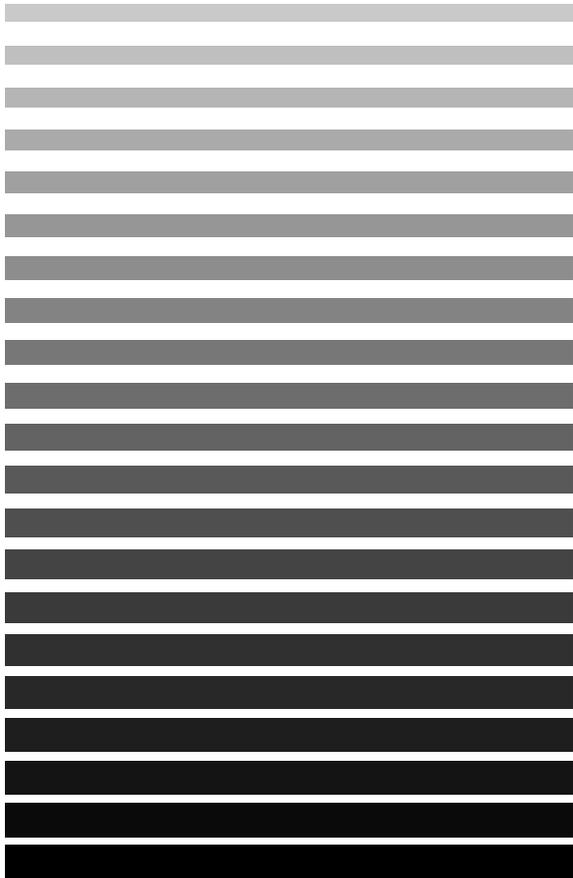
Commands	Functions	Refer to:
<b>A</b>		
ACCEL	Specify the internal composite acceleration/deceleration of joints included in a currently held arm group.	3-52
ARRIVE	Define the motion ratio relative to the programmed full travel distance to the target point in order to make the current program stand by to execute the next step until the robot reaches the defined motion ratio.	3-49
<b>C</b>		
ClearSrvMonitor	Initialize the pointer of data obtained by the single-joint servo data monitor function.	3-105
CURACC	Get the current internal composite acceleration of joints included in a currently held arm group.	3-58
CURDEC	Get the current internal composite deceleration of joints included in a currently held arm group.	3-60
CUREXJ	Get the current angle of an extended-joint into a floating-point variable.	3-47
CURJACC	Get the current internal acceleration of individual joints included in a currently held arm group.	3-59
CURJDEC	Get the current internal deceleration of individual joints included in a currently held arm group.	3-61
CURJSPD	Get the current internal speed of individual joints included in a currently held arm group.	3-57
CURSPD	Get the current internal composite speed of joints included in a currently held arm group.	3-56
<b>D</b>		
DECEL	Specify the internal composite deceleration of joints involved in a currently held arm group.	3-54
DESTEXJ	Get 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).	3-48
DRIVE	Carry out the relative motion of each joint.	3-43
DRIVEA	Carry out the absolute motion of each joint.	3-44
<b>G</b>		
GetJntData	Get the internal servo data of a specified joint.	3-109
GetSrvData	Get the internal servo data of robot joints.	3-108
GIVEARM	Release the currently held arm group.	3-42
<b>I</b>		
INTERRUPT ON/OFF	Interrupt the current robot motion.	3-62

Commands	Functions	Refer to:
<b>J</b>		
JACCEL	Specify the internal acceleration and deceleration of individual joints included in a currently held arm group.	3-53
JDECEL	Specify the internal deceleration ratio of individual joints included in a currently held arm group.	3-55
JSPEED	Specify the internal speed of individual joints included in a currently held arm group.	3-51
<b>M</b>		
MotionComp	Judge whether execution of running motion commands is complete.	3-107
MotionSkip	Abort running motion commands.	3-106
MOVE	Move 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.	3-45
MoveIndexHome	Rotate the index table to move a specified extended-joint to the home position.	3-71
mvResetPulseWidthJnt	Reset the encoder pulse count for an allowable positioning error for a specified extended-joint to the default.	3-66
mvSetPulseWidthJnt	Set the encoder pulse count for an allowable positioning error for a specified extended-joint.	3-65
<b>P</b>		
POSCLR	Forcibly restore the current position of a joint to 0 mm or 0 degree.	3-64
<b>R</b>		
ResetCurLmt	Release the drive current limit set for a specified joint motor.	3-69
ResetCycloidJnt	Cancel the cycloid mode set for a specified extended-joint and restore the normal mode.	3-68
ResetEralw	Reset the positioning error allowance of a specified joint to the initial value.	3-70
<b>S</b>		
SetCycloidJnt	Enter 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.	3-67
SetMonitorCond	Set the monitoring conditions for single-joint servo data monitor.	3-102
SPEED	Specify the internal composite speed of joints included in a currently held arm group.	3-50
StartSrvMonitor	Start monitoring single-joint servo data.	3-103
StopSrvMonitor	Stop monitoring single-joint servo data.	3-104
<b>T</b>		
TAKEARM	Get 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.	3-39

# Chapter 1



## Overview



This chapter outlines the extended-joint support system.

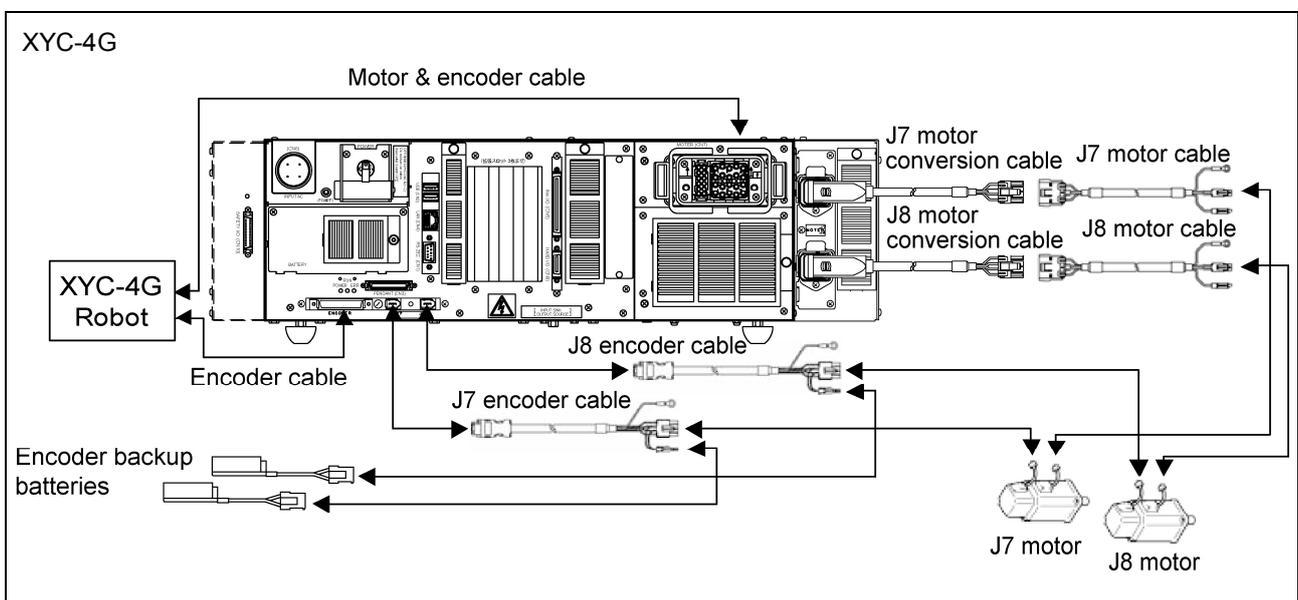
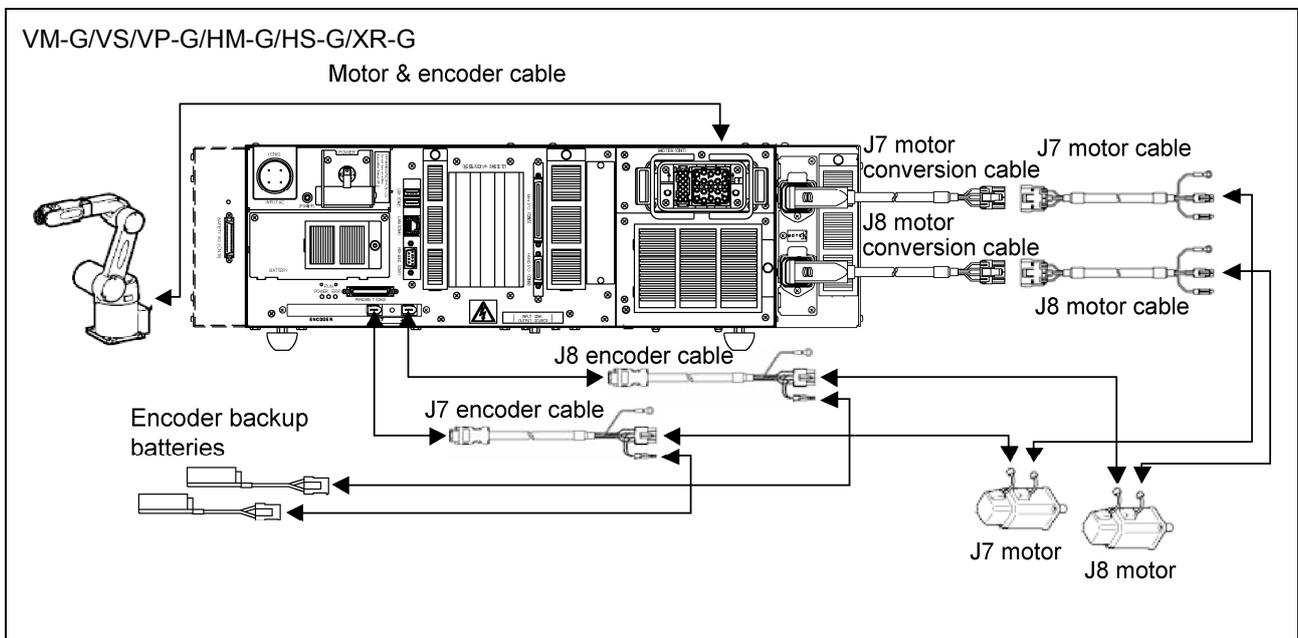


The extended-joint support system, which can be optionally configured for the VM-G, VS, VP-G, HM-G, HS-G, XYC-4G, and XR-G series, enables the extended-joint support robot controller to control up to two extended-joints in addition to the conventional robot joints.

## 1.1 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) The robot controller and extended-joint motors are connected using extended-joint motor cables, motor conversion cables, encoder cables, and encoder backup batteries (all of these are options).
- (3) Extended-joint motors (option) which are equipped with bus-line encoders (Up to 2 motors, motor type selectable)



---

## 1.2 Extended-Joint Related Functions

### (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) Configuring extended-joint parameters

To use extended-joints, you need to configure the "path configuration parameters" and "servo configuration 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 can easily do gain tuning of the extended-joint motors.

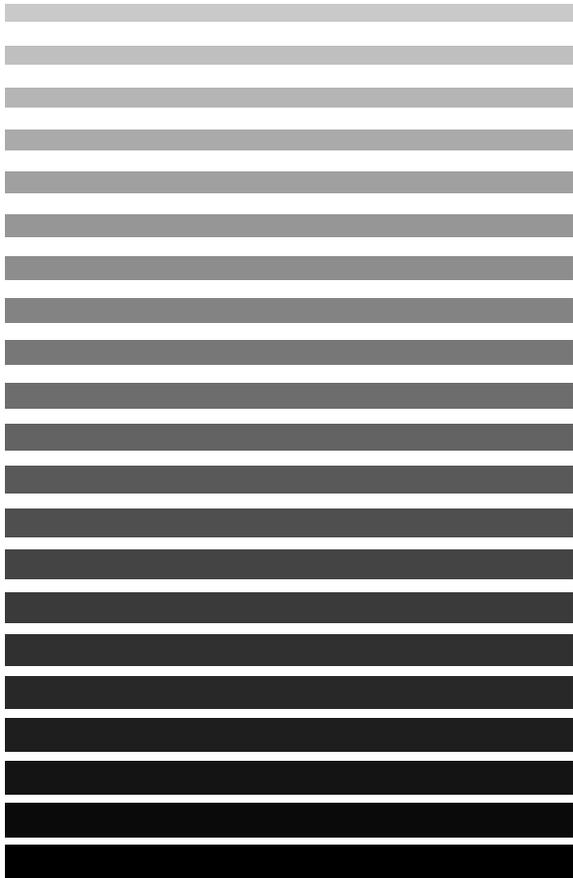
#### - Manual gain tuning

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 can 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.



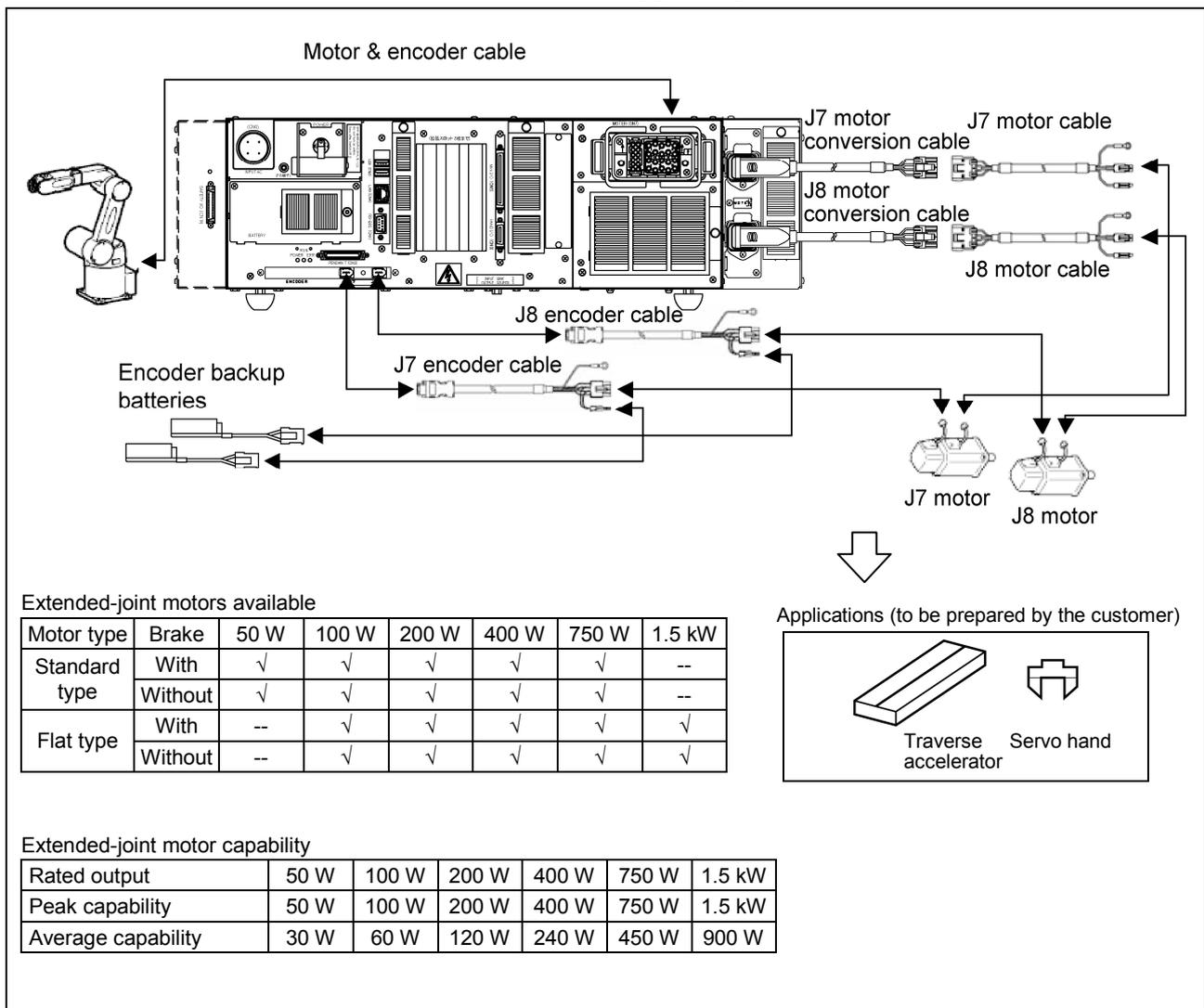
# 2.1 Configuration of Extended-Joint Support System

## 2.1.1 System Configuration

An extended-joint support system can be configured (see the illustration given below) by using a robot controller designed for extended-joint motors to be connected and connecting the following options (DENSO genuine components).

- Extended-joint motors equipped with bus-line encoders (Up to 2 motors, motor type selectable)
- Extended-joint motor cables
- Extended-joint motor conversion cables
- Extended-joint encoder cables
- Encoder backup batteries

**Extended-Joint Support System**



---

## 2.1.2 Precautions for Safe Use of the Robot System

**Note:** This section provides safety precautions to be taken when you configure the robot system with extended-joint motors. For details, refer to the ISO 10218-1:2006, Safety Requirements.

### [ 1 ] For extended-joint motor driven mechanism

- (1) If the mechanism involves a risk of bodily injury to workers, set up a safety fence to prevent danger.
- (2) When the extended-joint in the mechanism has the largest motion range in all joints (including robot joints), prepare adjustable mechanical stops with sufficient rigidity.
- (3) When the extended-joint in the mechanism has the 2nd or 3rd largest motion range in all joints (including robot joints), prepare joint limiters with sufficient rigidity.
- (4) If the mechanism involves a risk of bodily injury to workers when the brakes are released, use warning labels to alert workers to danger.
- (5) If it is not easy to carry the mechanism, prepare any measures for hoisting it.
- (6) Provide markings (e.g. labels) on the driven mechanisms to show which axis corresponds to which mechanism.

### [ 2 ] When connecting extended-joint motors

- (1) When connecting the extended-joint motors using the motor cables and their conversion cables, protect those cables from electrical noise.

### [ 3 ] When mounting extended-joint motors

- (1) If the motor surface gets hot and the workers are at risk of touching the heated surface, attach a warning label, alerting them to the high temperature.

### [ 4 ] For providing data to the end user

- (1) When the motion range of the extended-joint is 3rd largest or below in all joints (including robot joints), provide the end user with the maximum stop time and distance of that joint, using instruction manuals, etc.



## 2.2.2 Connectors on Extended-Joint Support Controller

Front Panel

Safety box  
(Only on the global type with safety box)

Safety board  
(Only on the global type with safety boardff)

Rear panel (Air exhaust side)

Front panel

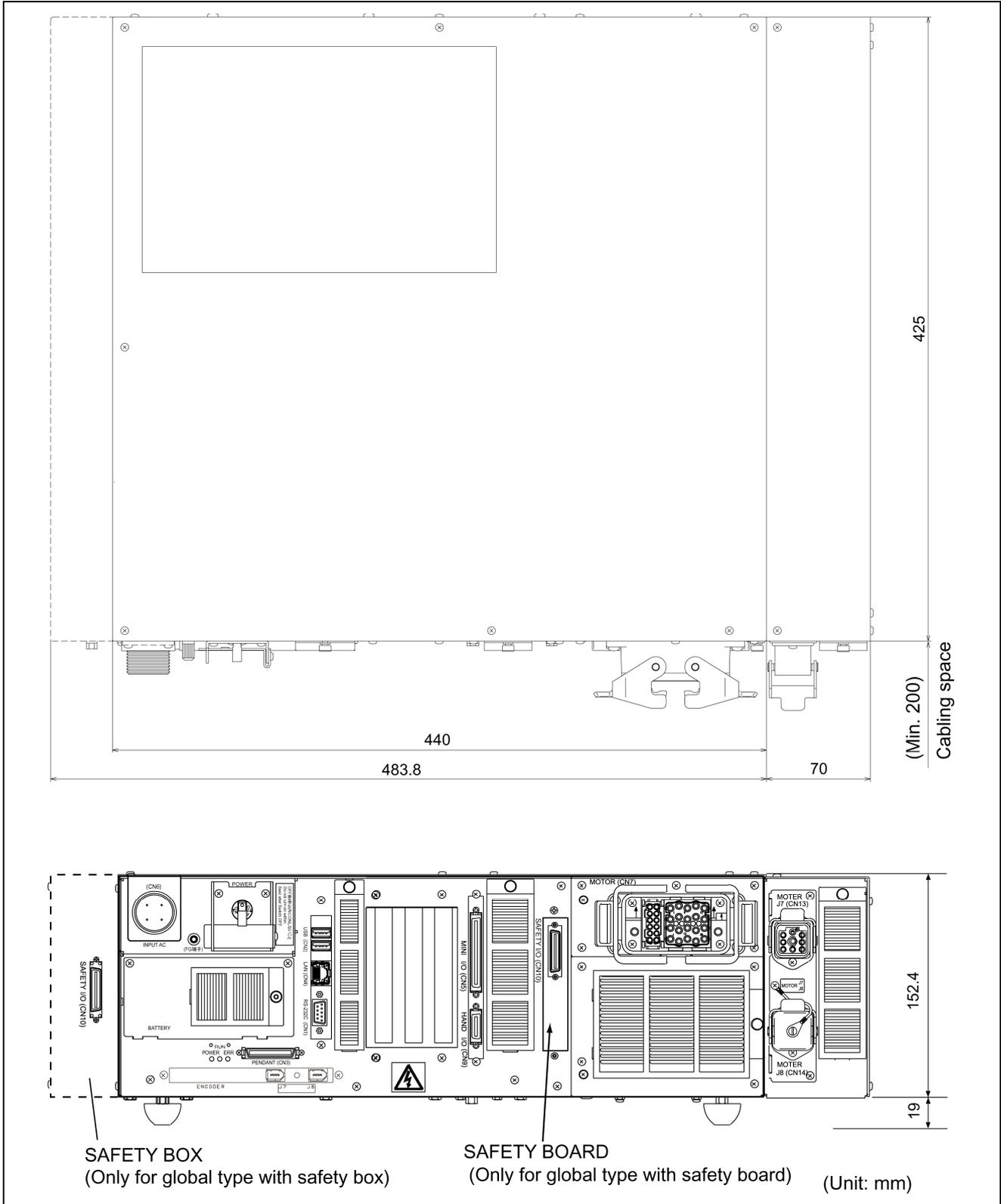
Air intake filters

Connector No.	Marking	Name
CN1	RS-232C	Serial interface connector
CN2	USB	USB connector (2 lines)
CN3	PENDANT	Teach pendant connector
CN4	LAN	Ethernet connector
CN5	Mini I/O	User/system I/O connector
CN6	INPUT AC	Power supply connector
CN7	MOTOR	Motor connector
CN8	ENCODER	Encoder connector (exclusive to XYC-4G)
CN9	HAND I/O	Hand I/O connector
CN10	SAFETY I/O	Safety I/O connector (only on the global type)
CN11	J7	Extended-joint J7 encoder connector
CN12	J8	Extended-joint J8 encoder connector
CN13	MOTOR J7	Extended-joint J7 motor connector
CN14	MOTOR J8	Extended-joint J8 motor connector

**⚠ Caution:** Be sure to turn off the power switch of the robot controller before connecting or disconnecting the connectors to/from the controller. Otherwise, the internal circuitry of the controller may be damaged.

### 2.2.3 External Dimensions of Extended-Joint Support Controller

The figure below shows the outer dimensions of the extended-joint support controller.



External Dimensions of Extended-Joint Support Controller

## 2.2.4 Controller Specifications

The table below lists the specifications of the extended-joint support controller.

Item		Specifications	
Applicable controller		RC7M + Extended-joint options (see Section 2.3)	
Robot series		VM-G, VS, VP-G, HM-G, HS-G, XYC-4G, and XR-G	
Number of controllable axes		VM-G, VS, and VP-G: Up to eight axes simultaneously (6 robot axes and 2 extended-joints) HM-G, HS-G, XYC-4G, and XR-G: Up to six axes simultaneously (4 robot axes and 2 extended-joints)	
Control system		PTP, CP 3-dimensional linear, 3-dimensional circular (Only PTP control for extended-joints)	
Drive system		All axes: Full-digital AC servo	
Language used		DENSO robot language (conforming to SLIM)	
Memory capacity		3.25 MB (equivalent to 10,000 steps, 30,000 points)	
Teaching system		1) Remote teaching 2) Numerical input (MDI)	
External signals (I/O)	Standard I/O	Mini I/O	Input signals: 8 user open points + 11 fixed system points Output signals: 8 user open points + 14 fixed system points <b>(Note: The global type does not use emergency stop input/output out of the fixed system points.)</b>
		HAND I/O	Input signals: 8 user open points Output signals: 8 user open points
	SAFETY I/O (Only for global type)		Input signals: 6 fixed system points Output signals: 5 fixed system points
	Parallel I/O board (Option)	2 boards	Input signals: Additional 80 user open points Output signals: Additional 96 user open points
		1 board	Input signals: Additional 40 user open points Output signals: Additional 48 user open points
	DeviceNet board (Option)	Master & slave	Input signals: 1024 points (Master) + 256 points (Slave) Output signals: 1024 points (Master) + 256 points (Slave)
		Master	Input signals: 1024 points Output signals: 1024 points
		Slave	Input signals: 256 points Output signals: 256 points
CC-Link board (option)	Slave	Input signals: 384 points Output signals: 384 points (including remote registers RWw and RWr)	
External communication		RS-232C: 1 line Ethernet: 1 line USB: 2 lines (supporting USB flash memory drives)	
Extension slots		3 (For adding optional boards)	
Self-diagnosis function		Overrun, servo error, memory error, input error, etc.	
Timer function		0.02 to 10 sec. (in units of 1/60 sec.)	
Error display		- Outputs error codes to the external I/O. - Displays error messages on the teach pendant (option). - Displays error codes on the mini pendant (option).	

## Chapter 2 Robot Components of Extended-Joint Support System

Item		Specifications	
Cables	VM-G, VS, VP-G, HM-G, HS-G, XR-G Motor & encoder cable	2 m, 4 m, 6 m, 12 m (20 m long cable not usable)	
	XYC-4G motor cable	4 m, 6 m	
	XYC-4G encoder cable	4 m, 6 m	
	I/O cable (option)	8 m, 15 m (For Mini I/O, HAND I/O, Optional board for parallel I/O and SAFETY I/O)	
	Power supply cable	5 m	
	Motor cable (option)	Two types available for 1.5 kW and 750 W or below. 4 m, 6 m, 12 m for each type (Connected to the controller via the motor conversion cable)	
	Encoder cable (option)	4 m, 6 m, 12 m	
Environmental conditions (in operation)		Temperature: 0 to 40°C Humidity: 90% RH or less (no condensation allowed)	
Power supply (in full configuration)		Three-phase, 200 VAC-15% to 230 VAC+10%, 50/60 Hz, 3.3 kVA Single-phase, 230 VAC±10%, 50/60 Hz, 3.3 kVA Single-phase, 100 VAC-15% to 110 VAC+10%, 50/60 Hz, 1.0 kVA	
I/O power source	External power source	A 24 VDC ±10% should be supplied from external equipment.	<b>Note:</b> Refer to the RC7M CONTROLLER MANUAL, Sections 4.2.1 and 5.2.1 "Setting up Mini I/O Power Supply."
	Internal power source	A 24 VDC ±10% should be supplied internally in the controller.	
Safety Performance (Note 1)		With safety box (Safety board is not available)	Structure : Category 4 Performance Level : e MTTFd : 2.03×10 <sup>3</sup> years (Emergency stop) 0.76×10 <sup>3</sup> years (Protective stop) DCavg : 99%
Stop category		With safety box: Category 1 (Safety board is not available) (See IEC60204-1 for details of stop category)	
Degree of protection		IP20	
Weight (in full configuration)		Approx. 26 kg (57 lbs.)	

(Note1) Safety performance of robot stopping functions (emergency stop and protective stop). Input devices to be supplied by customers such as Safety door switch are not included in the PL calculation.

Calculation conditions:

- This calculation is based on ISO 13849-1:2006 and IEC 62061:2005.
- Operating time           24 hours/day and 365 days/year
- Frequency of actuation   1 time/day (Emergency stop), 10 times/day (Protective stop)

---

## Controller Handling Notes

### WARNING

- NEVER touch fins. Their hot surfaces may cause severe burns.
- NEVER insert fingers or foreign objects into openings. Doing so may cause bodily injury.
- Before opening the controller cover and accessing the inside of the controller for maintenance, be sure to turn off the power switch, disconnect the power cable, and wait at least 3 minutes. This is for protecting you from electric shock.
- NEVER connect or disconnect connectors to/from the controller when the AC power or the 24 VDC power for I/O is being supplied. Doing so may cause electric shock or controller failure.

### CAUTION IN INSTALLATION

- This controller is not designed to be dust-proof, splash-proof, or explosion-proof.
- Read user's manuals before installation.
- Do not place anything on the controller or apply any impact to the controller.
- Avoid placing the controller in an environment where it may be subjected to excessive vibration.
- When releasing the connector locks of the extended-joint motor cables, take care not to let those locks scratch or damage the controller panel.

### CAUTION

- The robot controller connectors are of a screw-lock type or ring-lock type. Lock the connectors securely. If even one of the connectors is not locked, weak contact may result thereby causing an error.
- Be sure to turn the robot controller OFF before connecting/ disconnecting the power connector or motor connector. Otherwise, the internal circuits of the robot controller may be damaged.

### 2.2.5 Total Output Capacity for Extended-Joint Motors

Up to two extended-joint motors can 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-joint motors as listed below.

To connect extended-joint motors to the robot controller, it is necessary to use dedicated extended-joint cables listed in Section 2.3, "Optional Components for Extended-Joint Support System."

 **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 could cause a power error, overcurrent, overvoltage, overload and other errors, depending upon the use conditions of the extended-joint motors.**  
**Those problems may be solved by adjusting acceleration, deceleration, maximum speed, and gain of each extended-joint servo loop.**

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

Applicable robot and controller			Total output capacity for extended-joints, when both the robot unit and extended-joints operate simultaneously: (*1)	Total output capacity for extended-joints, when robot unit is on halt: (*2)	
Robot series	Motor capacity	Robot controller models		If extended-joints will be subjected to reaction force (e.g., when used as traverse accelerator): (*3)	If extended-joints will not be subjected to reaction force:
VM-G	3150W	RC7M-VMG6BA	Max. 795W	Max. 1065W	Max. 2175W
VS-G	1580W	RC7M-VSG6BA	Max. 1894W	Max. 2030W	Max. 2585W
VP-G	410W	RC7M-VPG5/6CA	Max. 2713W	Max. 2757W	Max. 2845W
HM-G	3100W	RC7M-HMG4BA	Max. 830W	Max. 1110W	Max. 2600W
HS-G	2200W	RC7M-HSG4BA	Max. 1460W	Max. 1646W	Max. 2740W
XYC-4G	900W	RC7M-XYCG4AA	Max. 2370W	Max. 2510W	Max. 2800W
XR-434G	1250W	RC7M-XRG4BA	Max. 2125W	Max. 2405W	Max. 2600W
XR-437G XR-43AG	1600W	RC7M-XRG4BA	Max. 1880W	Max. 2160W	Max. 2600W
VS-050/060	875W	RC7M-VSA1-****_* *_*	Max. 2388W	Max. 2463W	Max. 2763W
VS-068/087	2530W	RC7M-VSA2-****_* *_*	Max. 1229W	Max. 1479W	Max. 2235W

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

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

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

(\*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.

Note: The above table assumes the following operating conditions: Arm folded, 70% robot load, and 30% extended-joint load

## Chapter 2 Robot Components of Extended-Joint Support System

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

Applicable robot and controller			Total output capacity for extended-joints, when both the robot unit and extended-joints operate simultaneously: (*1)	Total output capacity for extended-joints, when robot unit is on halt: (*2)	
Robot series	Motor capacity	Robot controller models		If extended-joints will be subjected to reaction force (e.g., when used as traverse accelerator): (*3)	If extended-joints will not be subjected to reaction force:
VM-G	3150W	RC7M-VMG6BA	—	Max. 65W	Max. 1175W
VS-G	1580W	RC7M-VSG6BA	Max. 894W	Max. 1030W	Max. 1585W
VP-G	410W	RC7M-VPG5/6CA	Max. 1713W	Max. 1757W	Max. 1845W
HM-G	3100W	RC7M-HMG4BA	—	Max. 110W	Max. 1600W
HS-G	2200W	RC7M-HSG4BA	Max. 460W	Max. 646W	Max. 1740W
XYC-4G	900W	RC7M-XYCG4AA	Max. 1370W	Max. 1510W	Max. 1800W
XR-434G	1250W	RC7M-XRG4BA	Max. 1125W	Max. 1405W	Max. 1600W
XR-437G XR-43AG	1600W	RC7M-XRG4BA	Max. 880W	Max. 1160W	Max. 1600W
VS-050/060	875W	RC7M-VSA1-****-** _***	Max. 1388W	Max. 1463W	Max. 1763W
VS-068/087	2530W	RC7M-VSA2-****-** _***	Max. 229W	Max. 479W	Max. 1235W

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

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

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

(\*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.

**Note:** The above table assumes the following operating conditions: Arm folded, 70% robot load, and 30% extended-joint load

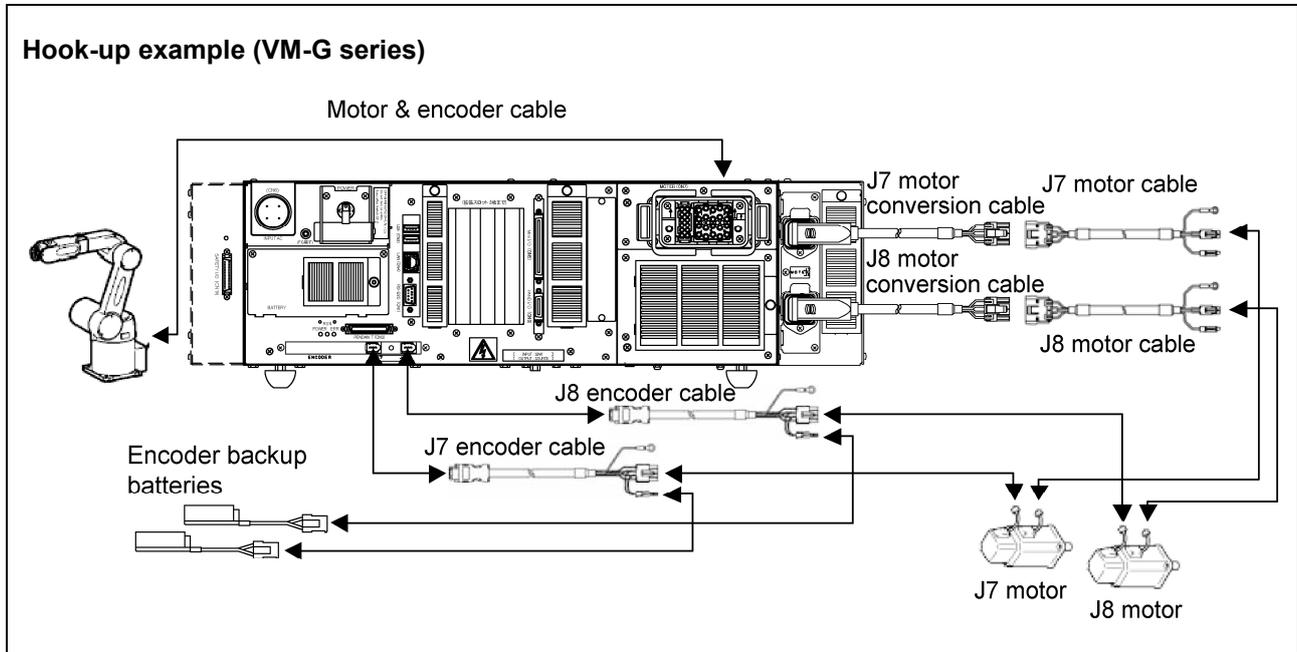
**(3) If supplied with single-phase, 100V power: Max. 400 W (Max. 200 x 2)**

Applicable robot and controller			Total output capacity for extended-joints, when both the robot unit and extended-joints operate simultaneously: (*1)	Total output capacity for extended-joints, when robot unit is on halt: (*2)	
Robot series	Motor capacity	Robot controller models		If extended-joints will be subjected to reaction force (e.g., when used as traverse accelerator): (*3)	If extended-joints will not be subjected to reaction force:
VP-G	410W	RC7M-VPG5/6CAA	Max. 400W	Max. 400W	Max. 400W
<p>(*1) When both the robot unit and extended-joint motors operate simultaneously, the total output capacity for extended-joints can apply as listed above.  <b>Connection example: If the output capacity for extended-joints is 800 W, the robot controller can connect up to two joints of 400 W each.</b></p> <p>(*2) If extended-joint motors are to be driven only when the robot is on halt, these output capacities can apply.</p> <p>(*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.</p> <p><b>Note: The above table assumes the following operating conditions: Arm folded, 70% robot load, and 30% extended-joint load</b></p>					

### 2.2.6 Connecting Cables to the Robot Controller

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

Refer to this example and the connector names given in Section 2.2.2.



#### ⚠ CAUTION

- (1) Use DENSO WAVE-authorized extended-joint cables given in Section 2.3 "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.
- (2) Miswiring of extended-joint cables could cause malfunction the motors.  
To prevent miswiring of extended-joint motors, attach the labels that come with a set of extended-joint cables to those cables.
- (3) Water or dust may get into the cable connection sections depending upon the usage environment. Take any protective measures for cable connectors.
- (4) The extended-joint support controller is shipped with connector caps mounted on the 8th joint motor and encoder connectors. To connect a motor to the 8th joint, remove those connector caps.
- (5) Be sure to connect all of the shield wires of the motor cables and encoder cables and the FG wires of the motors to ground.

#### 2.2.6.1 Replacement of encoder backup batteries

To replace encoder backup batteries, turn on the power of the controller and set the motor to OFF state.

⚠ Caution: The information of the encoder will be lost if you replace an encoder backup battery with the controller power OFF.

## 2.3 Optional Components for Extended-Joint Support System

The following optional components are available. Purchase them as necessary.

Optional Components (1)

Category	Name	Remarks	Part No.
Motors	AC servomotor (Standard type, 50W, w/o brake)	SGMAH-A5A1A-DH1*	410627-0210
	AC servomotor (Standard type, 50W, w/ brake)	SGMAH-A5A1A-DH2*	410627-0160
	AC servomotor (Standard type, 100W, w/o brake)	SGMAH-01A1A-DH1*	410627-0220
	AC servomotor (Standard type, 100W, w/ brake)	SGMAH-01A1A-DH2*	410627-0170
	AC servomotor (Standard type, 200W, w/o brake)	SGMAH-02A1A-DH1*	410627-0230
	AC servomotor (Standard type, 200W, w/ brake)	SGMAH-02A1A-DH2*	410627-0180
	AC servomotor (Standard type, 400W, w/o brake)	SGMAH-04A1A-DH1*	410627-0240
	AC servomotor (Standard type, 400W, w/ brake)	SGMAH-04A1A-DH2*	410627-0190
	AC servomotor (Standard type, 750W, w/o brake)	SGMAH-08A1A-DH1*	410627-0250
	AC servomotor (Standard type, 750W, w/ brake)	SGMAH-08A1A-DH2*	410627-0200
	AC servomotor (Flat type, 100W, w/o brake)	SGMPH-01A1A-DH1*	410627-0310
	AC servomotor (Flat type, 100W, w/ brake)	SGMPH-01A1A-DH2*	410627-0260
	AC servomotor (Flat type, 200W, w/o brake)	SGMPH-02A1A-DH1*	410627-0320
	AC servomotor (Flat type, 200W, w/ brake)	SGMPH-02A1A-DH2*	410627-0270
	AC servomotor (Flat type, 400W, w/o brake)	SGMPH-04A1A-DH1*	410627-0330
	AC servomotor (Flat type, 400W, w/ brake)	SGMPH-04A1A-DH2*	410627-0280
	AC servomotor (Flat type, 750W, w/o brake)	SGMPH-08A1A-DH1*	410627-0340
	AC servomotor (Flat type, 750W, w/ brake)	SGMPH-08A1A-DH2*	410627-0290
	AC servomotor (Flat type, 1500W, w/o brake)	SGMPH-15A1A-DH1*	410627-0350
	AC servomotor (Flat type, 1500W, w/ brake)	SGMPH-15A1A-DH2*	410627-0300

## Chapter 2 Robot Components of Extended-Joint Support System

### Optional Components (2)

Category	Name	Remarks	Part No.	
Extended-joint cables	Extended-joint cable set	1.5 kW	(4 m) <b>(Note 1)</b>	410149-1140
			(6 m) <b>(Note 1)</b>	410149-1150
			(12 m) <b>(Note 1)</b>	410149-1160
		Max. 750W	(4 m) <b>(Note 1)</b>	410149-1170
			(6 m) <b>(Note 1)</b>	410149-1180
			(12 m) <b>(Note 1)</b>	410149-1190
	Extended-joint motor conversion cable		<b>(Note 1)</b>	410149-0980
	Extended-joint motor cables	1.5 kW	(4 m) <b>(Note 1)</b>	410149-0990
			(6 m) <b>(Note 1)</b>	410149-1000
			(12 m) <b>(Note 1)</b>	410149-1010
		Max. 750W	(4 m) <b>(Note 1)</b>	410149-1020
			(6 m) <b>(Note 1)</b>	410149-1030
			(12 m) <b>(Note 1)</b>	410149-1040
	Extended-joint encoder cables		(4 m) <b>(Note 1)</b>	410149-1050
			(6 m) <b>(Note 1)</b>	410149-1060
(12 m) <b>(Note 1)</b>			410149-1070	
Encoder backup battery		One per motor	410611-0030	
Optional boxes	Rubber packing set for controller protection box extended-joint cables		410169-2140	
<p><b>(Note 1)</b> A set of miswiring prevention labels and the instruction sheet "Labeling Extended-Joint Cables for Miswiring Prevention" comes with extended-joint cables, a set per part number.</p>				

## 2.4 Choosing AC Servomotors

There are 20 types of AC servomotors available to the extended-joint support system. According to the reference data given in this section, you choose suitable motors.

### 2.4.1 AC Servomotors

#### 2.4.1.1 List of AC servomotors

The table below lists 20 types of AC servomotors available to the extended-joint support system.

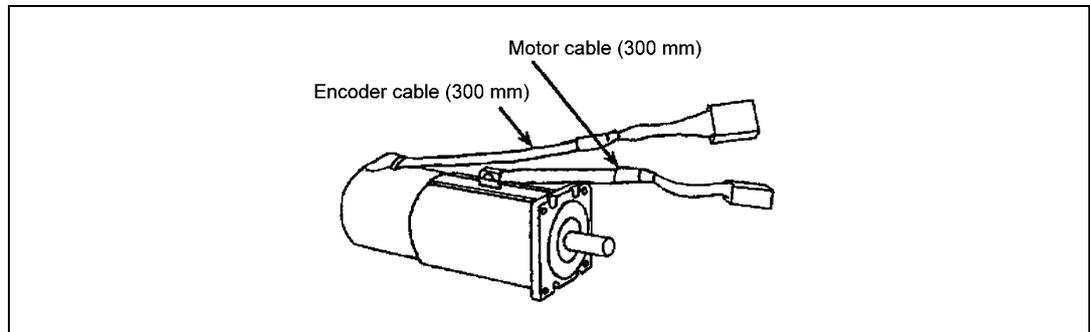
When placing an order for an extended-joint support controller, order AC servomotors selected at the same time. This is because the controller needs to have an IPM board suited to those motors at the factory.

**List of AC Servomotors for Extended-Joint Support System**

Type	Motor capacity	Brake	Motor model	Part No.	
Standard type (For the system requiring torque with small inertia)	50 W	Without	SGMAH-A5A1A-DH1*	410627-0210	
		With	SGMAH-A5A1A-DH2*	410627-0160	
	100 W	Without	SGMAH-01A1A-DH1*	410627-0220	
		With	SGMAH-01A1A-DH2*	410627-0170	
	200 W	Without	SGMAH-02A1A-DH1*	410627-0230	
		With	SGMAH-02A1A-DH2*	410627-0180	
	400 W	Without	SGMAH-04A1A-DH1*	410627-0240	
		With	SGMAH-04A1A-DH2*	410627-0190	
	750 W	Without	SGMAH-08A1A-DH1*	410627-0250	
		With	SGMAH-08A1A-DH2*	410627-0200	
	Flat type (For limited motor space)	100 W	Without	SGMPH-01A1A-DH1*	410627-0310
			With	SGMPH-01A1A-DH2*	410627-0260
200 W		Without	SGMPH-02A1A-DH1*	410627-0320	
		With	SGMPH-02A1A-DH2*	410627-0270	
400 W		Without	SGMPH-04A1A-DH1*	410627-0330	
		With	SGMPH-04A1A-DH2*	410627-0280	
750 W		Without	SGMPH-08A1A-DH1*	410627-0340	
		With	SGMPH-08A1A-DH2*	410627-0290	
1500 W		Without	SGMPH-15A1A-DH1*	410627-0350	
		With	SGMPH-15A1A-DH2*	410627-0300	

### 2.4.1.2 Cable End Treatment of AC Servomotors When Shipped

When shipped, the cable ends of an AC servomotor are treated with connector housings as shown below.



Cable End Treatment of AC Servomotor Shipped

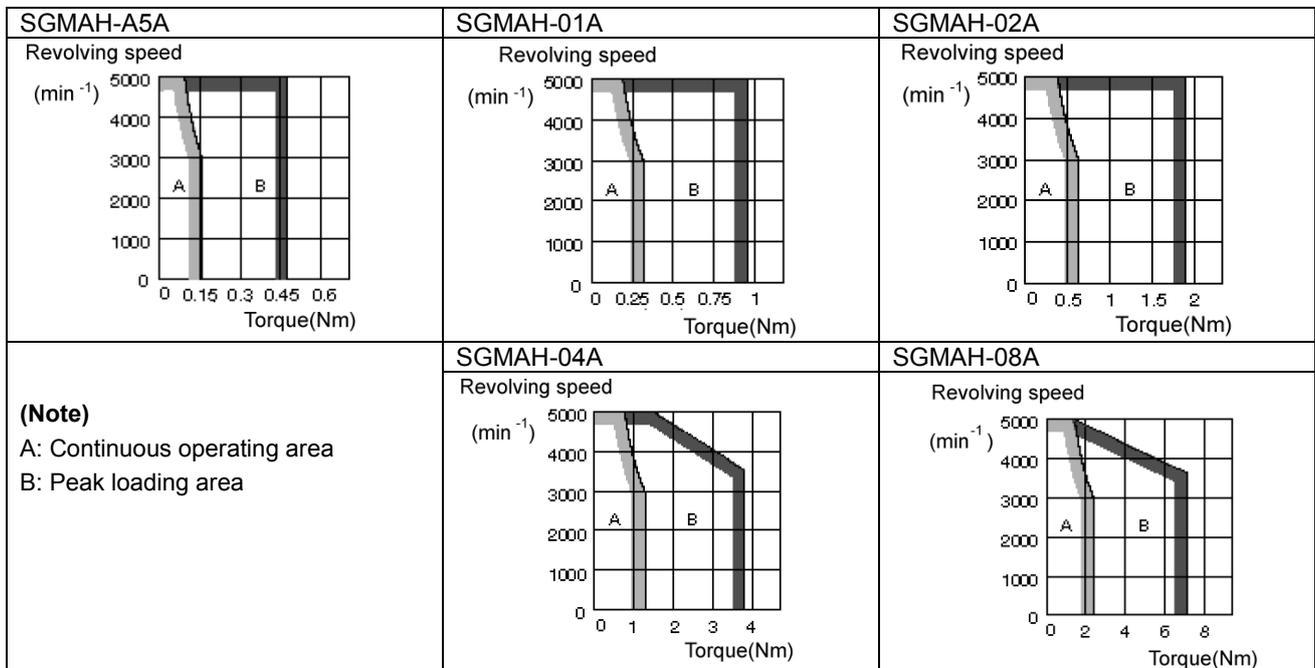
### 2.4.1.3 Motor characteristics lists

The main characteristics of each motor are listed below.

#### (1) Motor characteristics of standard type

Motor model		SGMAH-A5A	SGMAH-01A	SGMAH-02A	SGMAH-04A	SGMAH-08A	
Rated output	W	50	100	200	400	750	
Rated rotation speed	r/min	3000					
Maximum rotation speed	r/min	5000					
Rated output torque	Nm	0.159	0.318	0.637	1.27	2.39	
Peak output torque	Nm	0.477	0.955	1.91	3.82	7.16	
Rotor inertia	( $\times 10^{-4}$ kgm <sup>2</sup> )	0.0220	0.0364	0.106	0.173	0.672	
Allowable load inertia (Max. <i>n</i> times of rotor inertia)		Max. 30 times			Max. 20 times		
Rated current	Arms (RMS value)	0.64	0.91	2.1	2.8	4.4	
Torque constant	Nm/Arms (RMS value)	0.268	0.378	0.327	0.498	0.590	
Weight (without brake)	kg	0.4	0.5	1.1	1.7	3.4	
Weight (with brake)	kg	0.7	0.8	1.6	2.2	4.3	
Brake	Holding torque	Nm	0.159	0.318	0.637	1.27	2.39
	Inertia at revolving part, typical	( $\times 10^{-4}$ kgm <sup>2</sup> )	0.0085		0.058		0.14
	Exciting voltage	VDC	24 $\pm$ 10%				
	Exciting current (at 20°C)	ADC	0.25	0.25	0.29	0.29	0.32

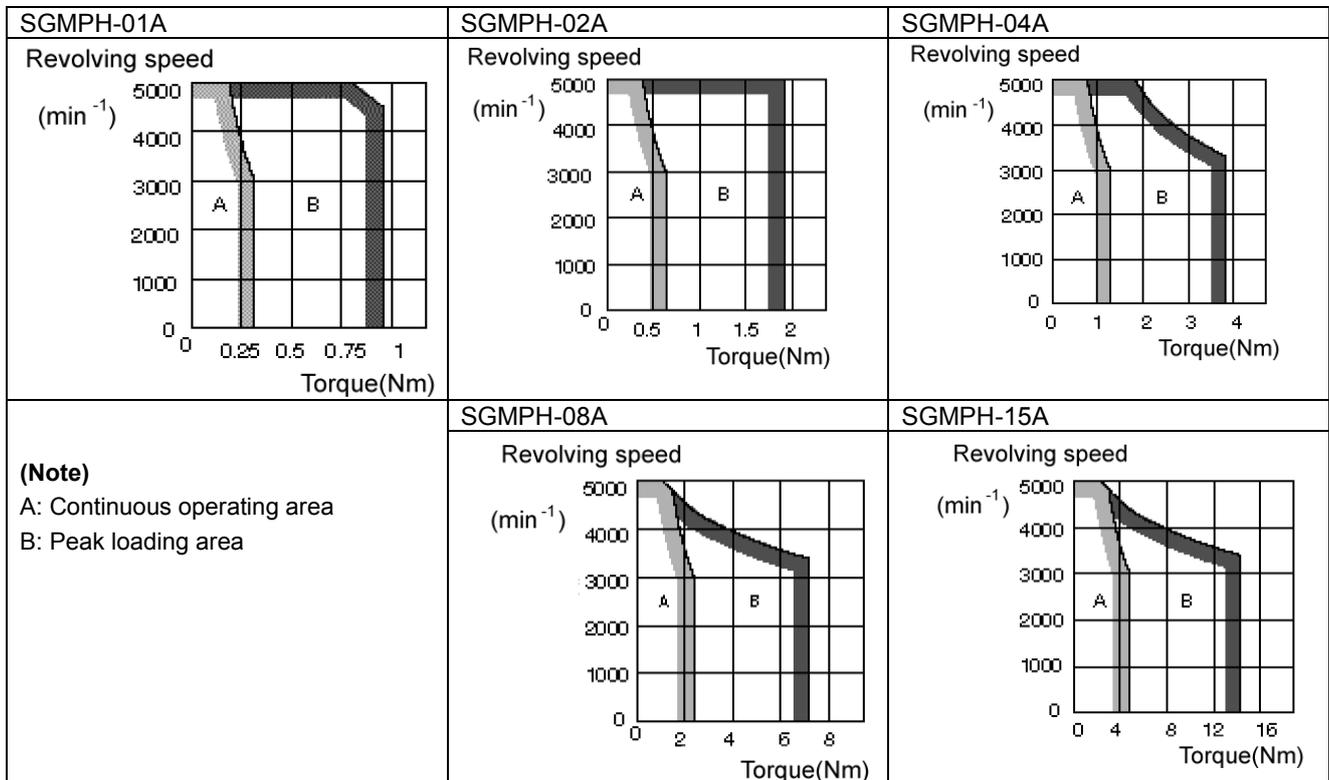
#### Torque vs. Revolving Speed



### (2) Motor characteristics of flat type

Rated output	W	SGMPH-01A	SGMPH-02A	SGMPH-04A	SGMPH-08A	SGMPH-15A	
Rated rotation speed	r/min	100	200	400	750	1500	
Maximum rotation speed	r/min	3000					
Rated output torque	Nm	5000					
Peak output torque	Nm	0.318	0.637	1.27	2.39	4.77	
Rotor inertia	( $\times 10^{-4}$ kgm <sup>2</sup> )	0.955	1.91	3.82	7.16	14.3	
Allowable load inertia (Max. <i>n</i> times of rotor inertia)		0.0491	0.193	0.331	2.10	4.02	
Rated current	Arms (RMS value)	Max. 25 times	Max. 15 times	Max. 7 times	Max. 5 times		
Torque constant	Nm/Arms (RMS value)	0.89	2.0	2.6	4.1	7.5	
Weight (without brake)	kg	0.392	0.349	0.535	0.641	0.687	
Weight (with brake)	kg	0.7	1.4	2.1	4.2	6.6	
Rated output	W	0.9	1.9	2.6	5.7	8.1	
Brake	Holding torque	Nm	0.318	0.637	1.27	2.39	4.77
	Inertia at revolving part, typical	( $\times 10^{-4}$ kgm <sup>2</sup> )	0.029	0.109		0.875	
	Exciting voltage	VDC	24 $\pm$ 10%				
	Exciting current (at 20°C)	ADC	0.32	0.32	0.32	0.31	0.42

**Torque vs. Revolving Speed**



---

## 2.4.1.4 Specification details

### (1) Performance

Item	Specifications
Heat resistance	Allowable ambient temperature Running: 0°C to +40°C In storage: -20°C to +80°C
Dampproof	Allowable ambient humidity 20% to 80% RH (No dew condensation allowed)
Insulation resistance	10 MΩ or more (when cooled), measured by 500 VDC megger (Motor part: Between frame and lead wire)
Withstand voltage	To resist 1500 VAC for one minute (at motor) To resist 1200 VAC for one second (at brake)
Instantaneous maximum rotation speed	100% of maximum rotation speed
Vibration resistance	49m/s <sup>2</sup> max. (In each of X, Y, and Z axes)
Enclosure	Fully-enclosed, non-ventilated IP55 (except the motor shaft penetrating section)

### (2) Assembly accuracy

- ① The assembly accuracy is specified in external dimension drawings given in Section 2.4.2.
  - The motor shaft deviation should be measured in the lateral direction.
  - The perpendicularity of the flange face to the shaft and the eccentricity of the spigot joint should be measured in the shaft upper direction.
- ② The end play (Shaft direction backlash) should be 0.3 mm or less.

### (3) Mounting to the equipment

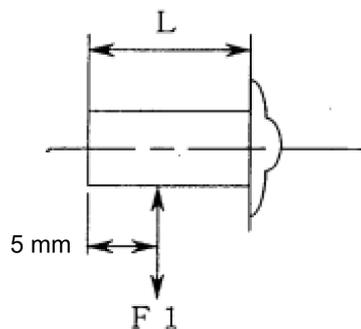
- ① When mounting the motor shaft to the equipment, use a flexible joint.
- ② If a rigid joint is used, even a slight runout of the center applies an excessive force to the shaft, resulting in a shaft break.

If it is not avoidable to use a rigid joint, confirm the mounting accuracy and strength of the motor shaft.
- ③ The motor has a built-in positioning transducer. When mounting the motor to the equipment, take special care not to apply an excessive force to the shaft; otherwise, the positioning transducer may be broken.
- ④ Do not pull motor lead wires. Do not allow them to be bent repeatedly.

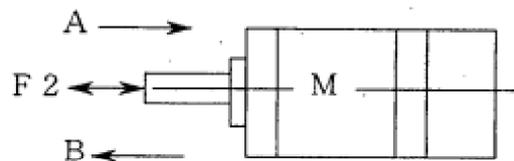
### (4) Allowable shaft load

Motor			Radial load (F1)	Thrust load (F2)
Part No.	Model	Capacity		
410627-0210	SGMAH-A5A1A-DH1*	50 W	68N	54N
410627-0160	SGMAH-A5A1A-DH2*	50 W	68N	54N
410627-0220	SGMAH-01A1A-DH1*	100 W	78N	54N
410627-0170	SGMAH-01A1A-DH2*	100 W	78N	54N
410627-0230	SGMAH-02A1A-DH1*	200 W	245N	74N
410627-0180	SGMAH-02A1A-DH2*	200 W	245N	74N
410627-0240	SGMAH-04A1A-DH1*	400 W	245N	74N
410627-0190	SGMAH-04A1A-DH2*	400 W	245N	74N
410627-0250	SGMAH-08A1A-DH1*	750 W	392N	147N
410627-0200	SGMAH-08A1A-DH2*	750 W	392N	147N
410627-0310	SGMPH-01A1A-DH1*	100 W	78N	49N
410627-0260	SGMPH-01A1A-DH2*	100 W	78N	49N
410627-0320	SGMPH-02A1A-DH1*	200 W	245N	68N
410627-0270	SGMPH-02A1A-DH2*	200 W	245N	68N
410627-0330	SGMPH-04A1A-DH1*	400 W	245N	68N
410627-0280	SGMPH-04A1A-DH2*	400 W	245N	68N
410627-0340	SGMPH-08A1A-DH1*	750 W	392N	147N
410627-0290	SGMPH-08A1A-DH2*	750 W	392N	147N
410627-0350	SGMPH-15A1A-DH1*	1500 W	490N	147N
410627-0300	SGMPH-15A1A-DH2*	1500 W	490N	147N

Radial load (F1) position



Thrust load (F2) direction



---

## (5) Lead wire colors and signals

### ■ Motor lead wires

Motor side	Name
Red	U
White	V
Blue	W
Green/Yellow	E (FG)

### ■ Brake lead wires

Brake side	Name
Black	Brake
Black	Brake

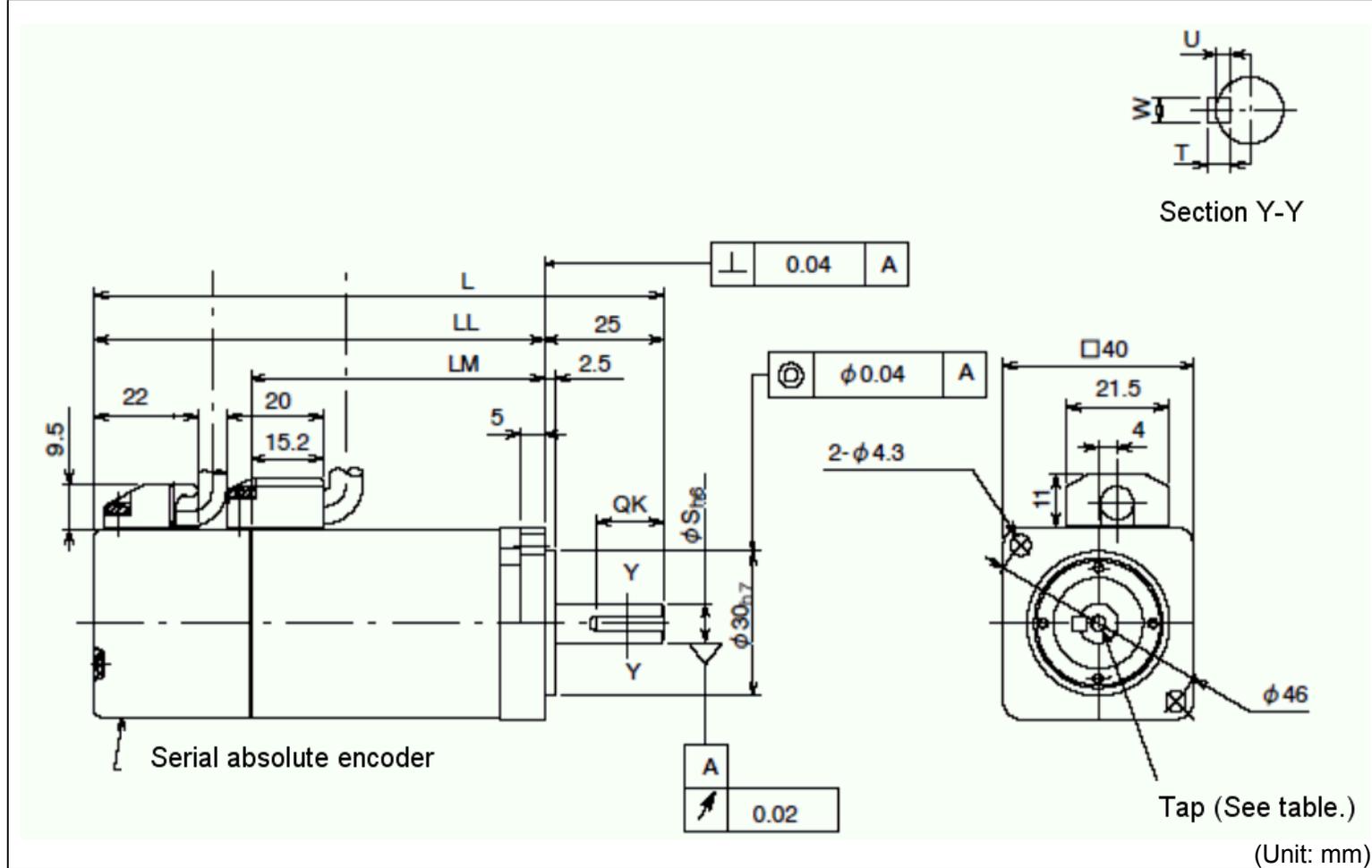
### ■ Encoder lead wires

Encoder side	Name	Remarks
Orange	BAT	Twisted pair
White/Orange	COMMON	
Sky blue	PS	Twisted pair
White/Sky blue	$\overline{\text{PS}}$	
Red	+5 V	Twisted pair
Black	0 V	
Shielding wire	Shield	

### 2.4.2 External Dimensions of AC Servomotors

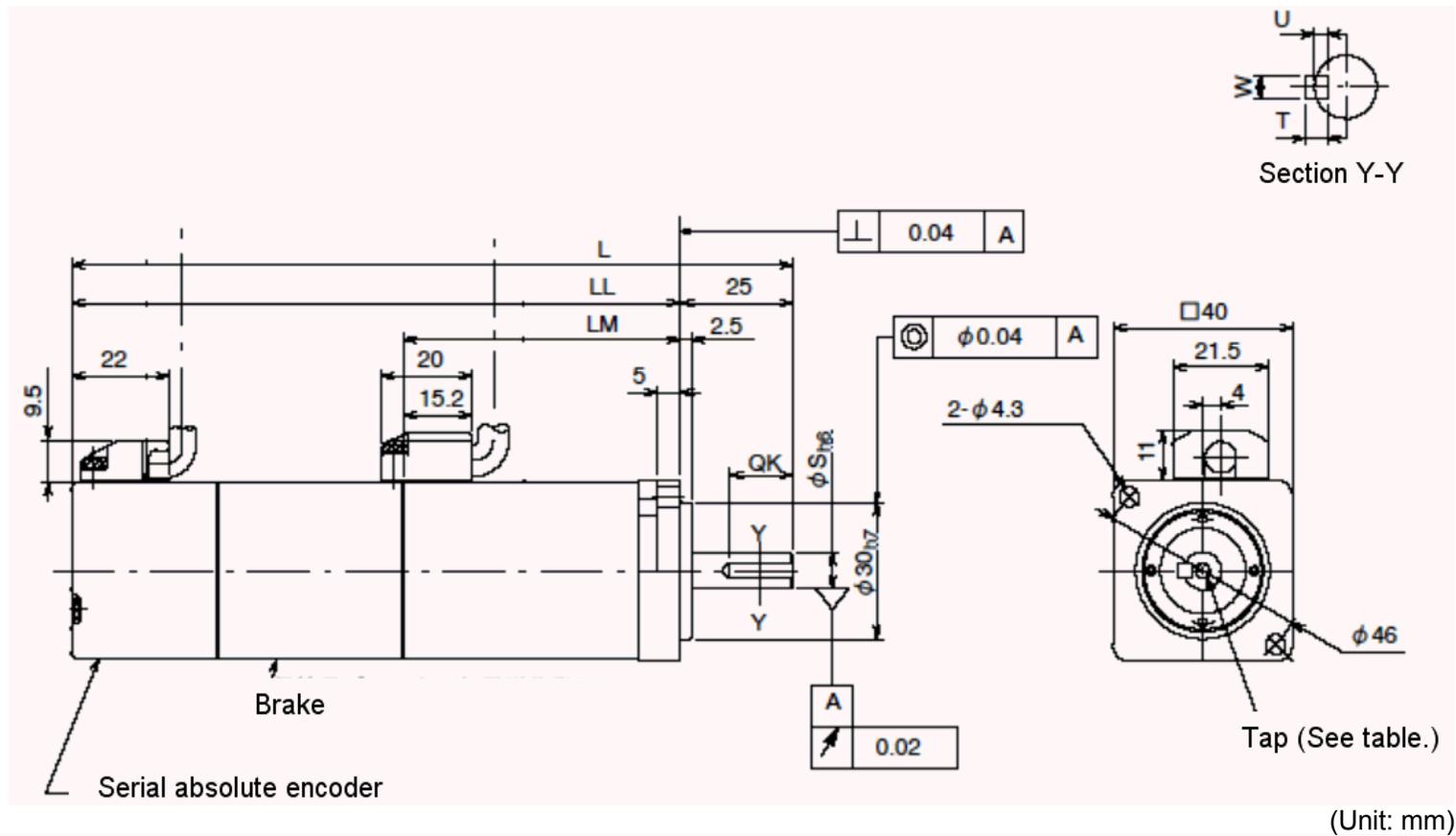
■ Standard type without brake, 50 W or 100 W

Part No.	Motor model SGMAH -	Rated output	L	LL	LM	S	Tap	QK	U	W	T	Approx weight	Remarks
410627-0210	A5A1A - DH1*	50 W	102.0	77.0	44.0	6	Dia. 2.5, Depth 5	14	1.2	2	2	0.4 kg	(1) The assembly accuracy conforms to the Japan Machine Tool Builders' Association standard (MAS402-1981). (TIR value) - Shaft end deflection: 0.03 (at the shaft projection center) - Perpendicularity of flange face to shaft: 0.08 (φ5) - Eccentricity of flange-fitting-outside diameter against shaft: 0.06 (at the center of spigot joint)  (2) Flange fixing bolt to be used: Hexagon socket-head bolt  (3) Motor cable / Encoder cable length: 300 mm
410627-0220	01A1A - DH1*	100 W	119.5	94.5	61.5	8	Dia. 3, Depth 6	14	1.8	3	3	0.5 kg	



## ■ Standard type with brake, 50 W or 100 W

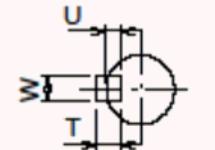
Part No.	Motor model SGMAH -	Rated output	L	LL	LM	S	Tap	QK	U	W	T	Approx. weight	Remarks
410627-0160	A5A1A - DH2*	50 W	133.5	108.5	44.0	6	Dia. 2.5, Depth 5	14	1.2	2	2	0.7 kg	(1) The assembly accuracy conforms to the Japan Machine Tool Builders' Association standard (MAS402-1981). (TIR value) - Shaft end deflection: 0.03 (at the shaft projection center) - Perpendicularity of flange face to shaft: 0.08 (φ45) - Eccentricity of flange-fitting-outside diameter against shaft: 0.06 (at the center of spigot joint)
410627-0170	01A1A - DH2*	100 W	160.0	135.0	61.5	8	Dia. 3, Depth 6	14	1.8	3	3	0.8 kg	



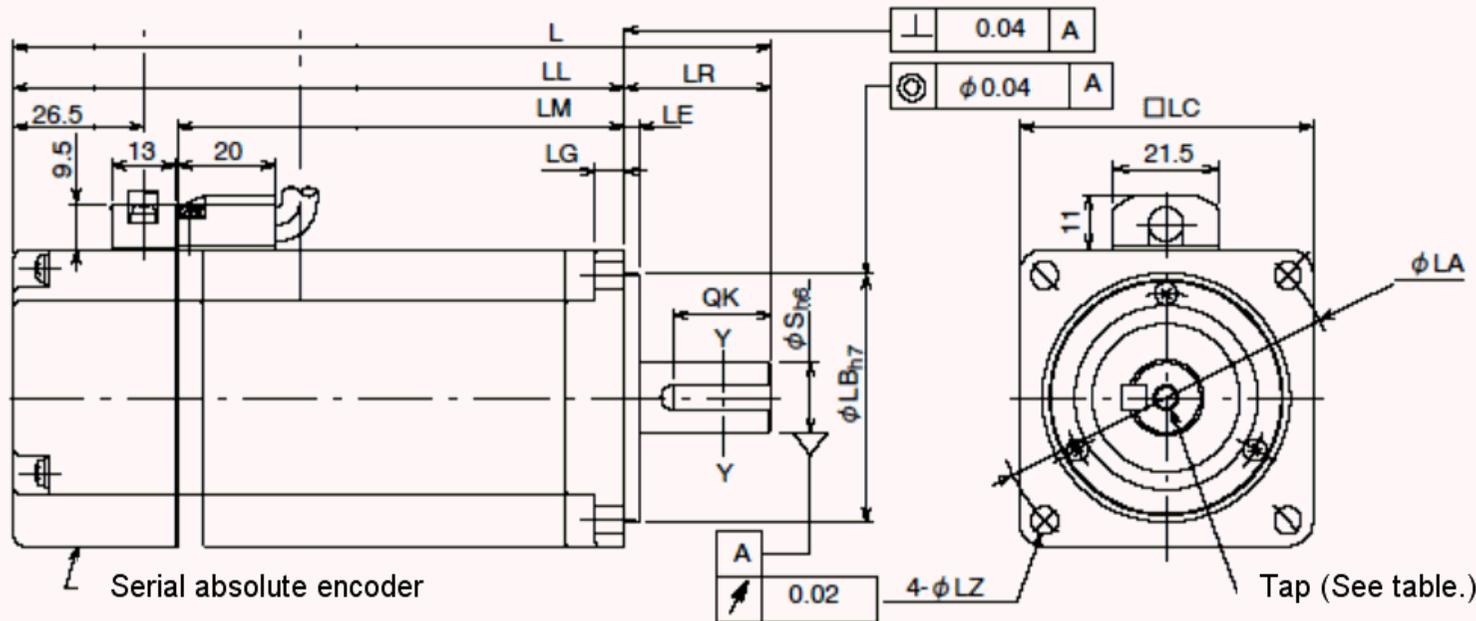
- (1) The assembly accuracy conforms to the Japan Machine Tool Builders' Association standard (MAS402-1981). (TIR value)
  - Shaft end deflection: 0.03 (at the shaft projection center)
  - Perpendicularity of flange face to shaft: 0.08 (φ45)
  - Eccentricity of flange-fitting-outside diameter against shaft: 0.06 (at the center of spigot joint)
- (2) Flange fixing bolt to be used: Hexagon socket-head bolt
- (3) Motor cable / Encoder cable length: 300 mm

### ■ Standard type without brake, 200 W, 400 W or 750 W

Part No.	Motor model SGMAH-	Rated output	L	LL	LM	LR	LE	LG	LC	LA	LZ	S	LB	Tap	Q K	U	W	T	Approx. weight	Remarks
410627-0230	02A1A-DH1*	200W	126.5	96.5	62.5	30	3	6	60	70	5.5	14	50	Dia. 5, Depth 8	20	3	5	5	1.1 kg	(1) The assembly accuracy conforms to the Japan Machine Tool Builders' Association standard (MAS402-1981). (TIR value)  - Shaft end deflection: 0.03 (at the shaft projection center) - Perpendicularity of flange face to shaft: 0.08 (φ45) - Eccentricity of flange-fitting-outside diameter against shaft: 0.06 (at the center of spigot joint)
410627-0240	04A1A-DH1*	400W	154.5	124.5	90.5	30	3	6	60	70	5.5	14	50	Dia. 5, Depth 8	20	3	5	5	1.7 kg	
410627-0250	08A1A-DH1*	750W	185	145	111	40	3	8	80	90	7	16	70	Dia. 5, Depth 8	20	3	5	5	3.4 kg	



Section Y-Y

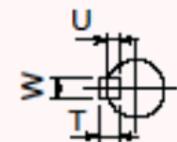


(Unit: mm)

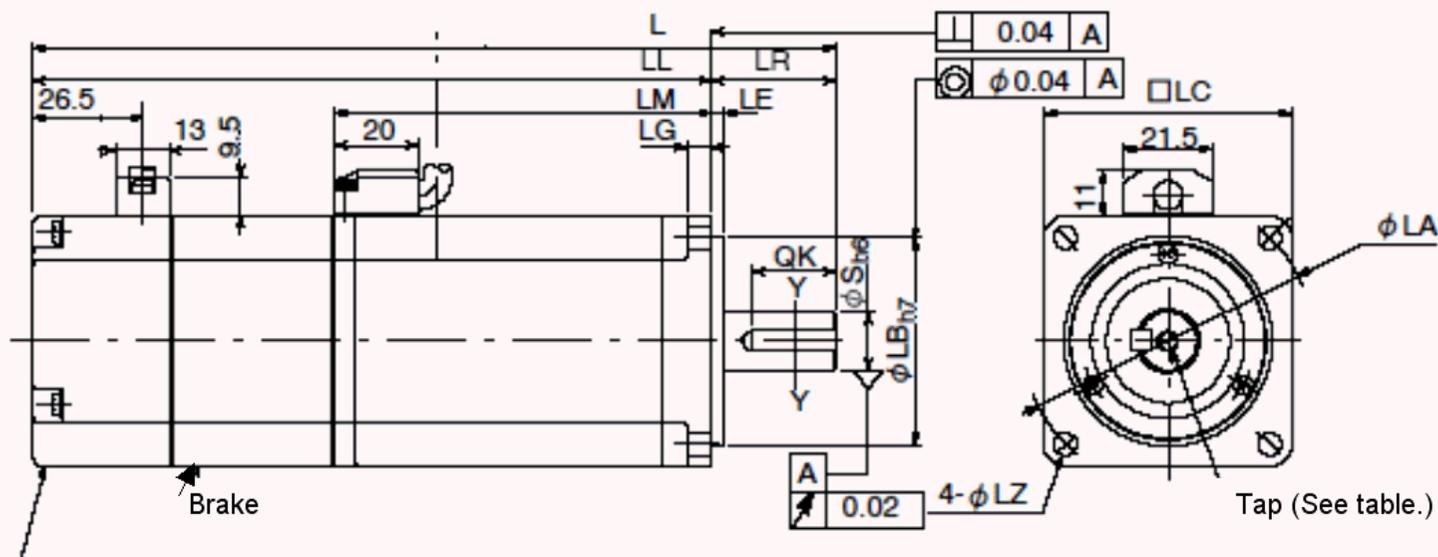
- (1) The assembly accuracy conforms to the Japan Machine Tool Builders' Association standard (MAS402-1981). (TIR value)  
  
- Shaft end deflection: 0.03 (at the shaft projection center)  
- Perpendicularity of flange face to shaft: 0.08 (φ45)  
- Eccentricity of flange-fitting-outside diameter against shaft: 0.06 (at the center of spigot joint)
- (2) Flange fixing bolt to be used: Hexagon socket-head bolt
- (3) Motor cable / Encoder cable length: 300 mm

### ■ Standard type with brake, 200 W, 400 W or 750 W

Part No.	Motor model SGMAH-	Rated output	L	LL	LM	LR	LE	LG	LC	LA	LZ	S	LB	Tap	QK	U	W	T	Approx. weight	Remarks
410627-0180	02A1A-DH2*	200 W	166.0	136.0	62.5	30	3	6	60	70	5.5	14	50	Dia. 5, Depth 8	20	3	5	5	1.6 kg	(1) The assembly accuracy conforms to the Japan Machine Tool Builders' Association standard (MAS402-1981). (TIR value)  -Shaft end deflection: 0.03 (at the shaft projection center) -Perpendicularity of flange face to shaft: 0.08 ( $\phi 45$ ) -Eccentricity of flange-fitting-outside diameter against shaft: 0.06 (at the center of spigot joint)  (2) Flange fixing bolt to be used: Hexagon socket-head bolt  (3) Motor cable / Encoder cable length: 300 mm
410627-0190	04A1A-DH2*	400 W	194.0	164.0	90.5	30	3	6	60	70	5.5	14	50	Dia. 5, Depth 8	20	3	5	5	2.2 kg	
410627-0200	08A1A-DH2*	750 W	229.5	189.5	111	40	3	8	80	90	7	16	70	Dia. 5, Depth 8	20	3	5	5	4.3 kg	



Section Y-Y



Serial absolute encoder

(Unit: mm)

### ■ Flat type without brake, 100 W, 200 W, 400 W, 750 W, or 1.5 kW

Part No.	Motor model SGMPH-	Rated output	L	LL	LM	LR	LE	LG	LF	LC	LA	LZ	S	LB	LH	Tap	QK	U	W	T	Approx. weight	Remarks
410627-0310	01A1A-DH1*	100 W	87.0	62.0	42.5	25	3	6	12.5	60	70	5.5	8	50	10.55	Dia. 3, Depth 6	14	1.8	3	3	0.7 kg	(1) The assembly accuracy conforms to the Japan Machine Tool Builders' Association standard (MAS402-1981). (TIR value)  -Shaft end deflection: 0.03 (at the shaft projection center) -Perpendicularity of flange face to shaft: 0.08 (φ45) -Eccentricity of flange-fitting-outside diameter against shaft: 0.06 (at the center of spigot joint)  (2) Flange fixing bolt to be used: Hexagon socket-head bolt  (3) Motor cable / Encoder cable length: 300 mm
410627-0320	02A1A-DH1*	200 W	97.0	67.0	48.1	30	3	8	11.9	80	90	7	14	70	8.25	Dia. 5, Depth 8	16	3	5	5	1.4 kg	
410627-0330	04A1A-DH1*	400 W	117.0	87.0	68.1	30	3	8	11.9	80	90	7	14	70	8.25	Dia. 5, Depth 8	16	3	5	5	2.1 kg	
410627-0340	08A1A-DH1*	750 W	126.5	86.5	66.7	40	3.5	10	12.8	120	145	10	16	110	10.5	Dia. 5, Depth 8	22	3	5	5	4.2 kg	
410627-0350	15A1A-DH1*	1.5 kW	154.5	114.5	94.7	40	3.5	10	12.8	120	145	10	19	110	10.5	Dia. 6, Depth 10	22	3.5	6	6	6.6 kg	

Section Y-Y

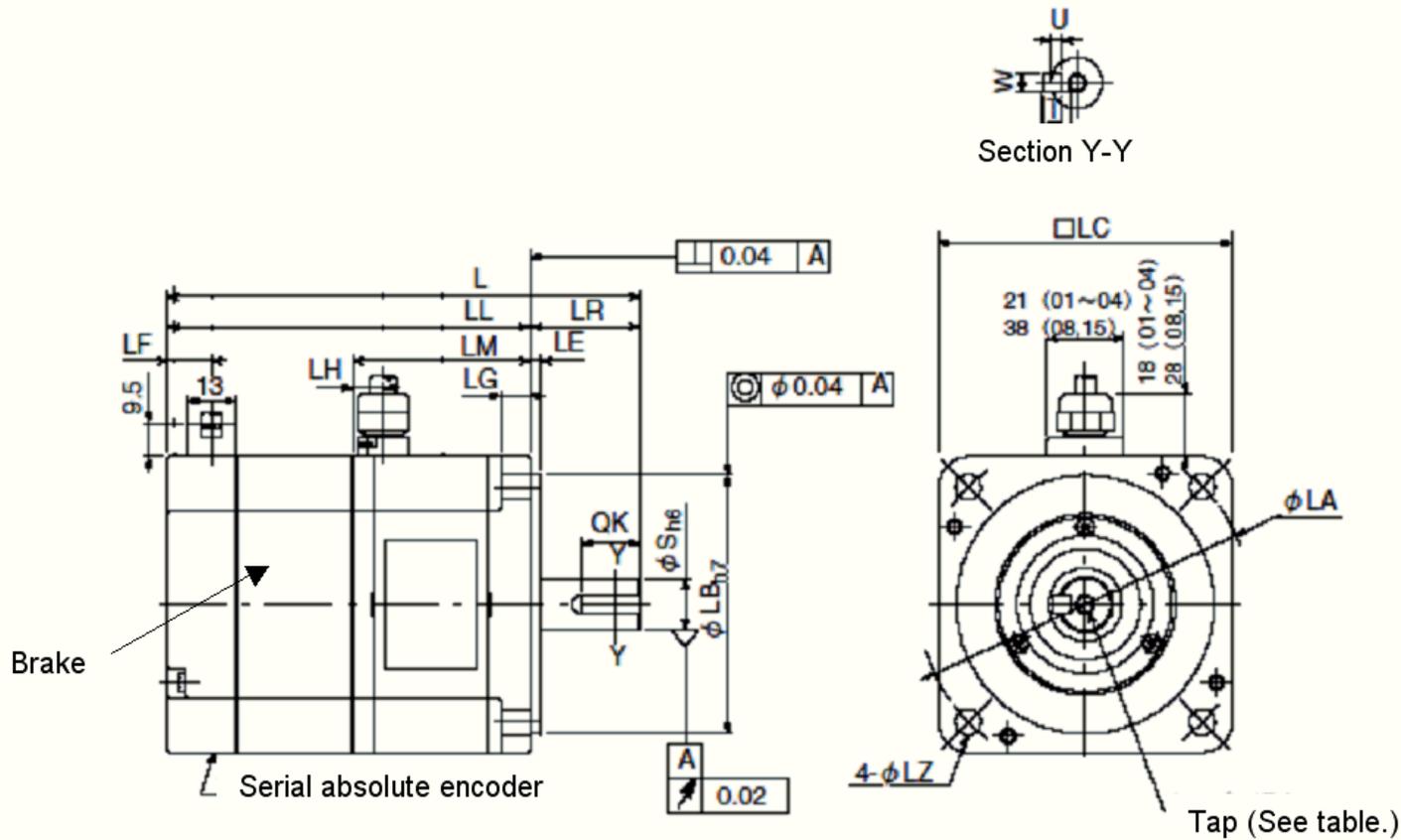
Serial absolute encoder

Tap (See table.)

(Unit: mm)

■ Flat type with brake, 100 W, 200 W, 400 W, 750 W, or 1.5 kW

Part No.	Motor model SGMPH-	Rated output	L	LL	LM	LR	LE	LG	LF	LC	LA	LZ	S	LB	LH	Tap	QK	U	W	T	Approx. weight	Remarks
410627-0260	01A1A-DH2*	100 W	116.0	91.0	42.5	25	3	6	12.5	60	70	5.5	8	50	10.55	Dia. 3, Depth 6	14	1.8	3	3	0.9 kg	(1) The assembly accuracy conforms to the Japan Machine Tool Builders' Association standard (MAS402-1981). (TIR value) -Shaft end deflection: 0.03 (at the shaft projection center) -Perpendicularity of flange face to shaft: 0.08 ( $\phi 45$ ) -Eccentricity of flange-fitting-outside diameter against shaft: 0.06 (at the center of spigot joint) (2) Flange fixing bolt to be used: Hexagon socket-head bolt (3) Motor cable / Encoder cable length: 300 mm
410627-0270	02A1A-DH2*	200 W	128.5	98.5	48.1	30	3	8	11.9	80	90	7	14	70	8.25	Dia. 5, Depth 8	16	3	5	5	1.9 kg	
410627-0280	04A1A-DH2*	400 W	148.5	118.5	68.1	30	3	8	11.9	80	90	7	14	70	8.25	Dia. 5, Depth 8	16	3	5	5	2.6 kg	
410627-0290	08A1A-DH2*	750 W	163.0	123.0	66.7	40	3.5	10	12.8	120	145	10	16	110	10.5	Dia. 5, Depth 8	22	3	5	5	5.7 kg	
410627-0300	15A1A-DH2*	1.5 kW	188.0	148.0	94.7	40	3.5	10	12.8	120	145	10	19	110	10.5	Dia. 6, Depth 10	22	3.5	6	6	8.1 kg	



(Unit: mm)

### 2.4.3 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 individual 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 a slight runout of the center within the allowable range, use flexible couplings exclusively designed for servomotors and as rigid as possible.
    - When using rigid couplings, mount them with extra care. If applied in installation, excessive bending load will break motor shafts or shorten the service life of bearings.
  - (6) When mounting 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.

Perform centering 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.

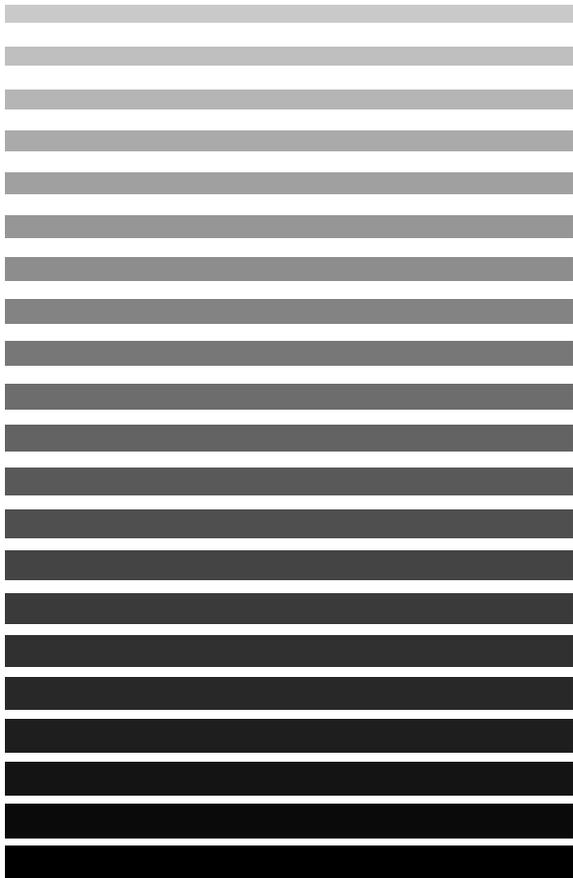
### **[ 3 ] Motor storage notes**

- (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 within the specified temperature/humidity range (-20°C to 80°C, RH 90% or less, without dew condensation).

# Chapter 3



## Extended-Joint Related Functions



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



## 3.1 Extended-Joint Related Functions

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

The extended-joint related functions include the following.

- [ 1 ] Operating extended-joints manually
- [ 2 ] Getting extended-joint positions
- [ 3 ] Getting extended-joint positions in conjunction with robot joint positions
  - Successive position intake procedure
  - Concurrent position intake procedure
- [ 4 ] Operating extended-joints by specifying a desired variable
- [ 5 ] Operating extended-joints by program  
(Several commands are extended and added.)
- [ 6 ] Setting the boundless rotation of extended-joints
- [ 7 ] Setting the link info
- [ 8 ] Setting the radius of gyration

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

### 3.1.1 Operating Procedures of the Extended-Joint Related Functions

This section describes the operating procedures of the extended-joint related functions listed above.

## [ 1 ] Operating extended-joints manually

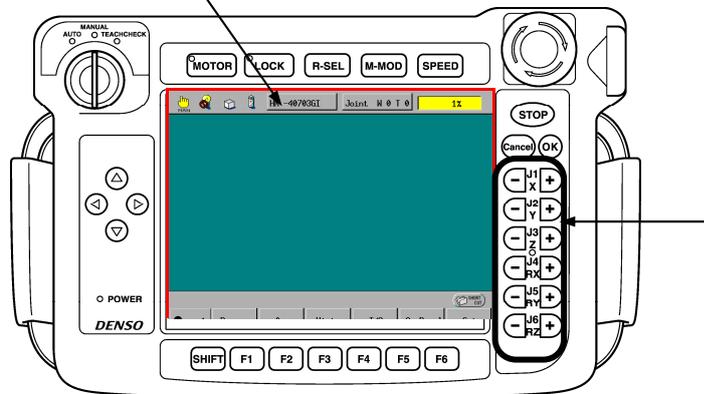
This section describes items relating to the extended-joints. For other details, refer to the SETTING-UP MANUAL, Section 3.2 "Manual Mode."

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

### ■ From the teach pendant

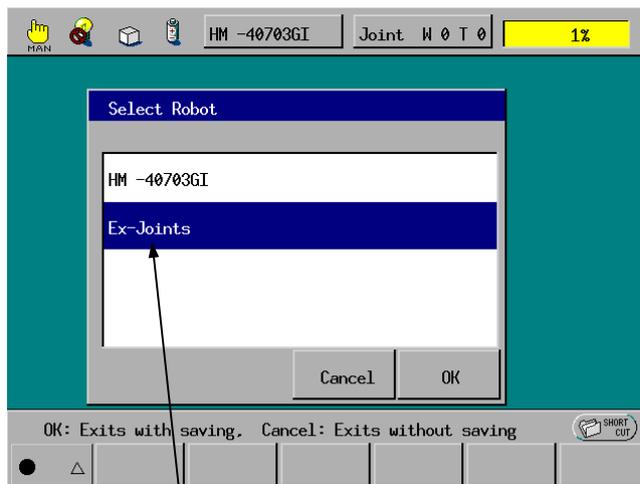
**Step 1** Set the mode switch to the MANUAL position and press the MOTOR key to turn the motor on.

**Step 2** Press the robot select button (or R-SEL key).



The Select Robot window will display as shown below.

**Step 3** Select **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 automatically switched from robot joints to extended-joints.

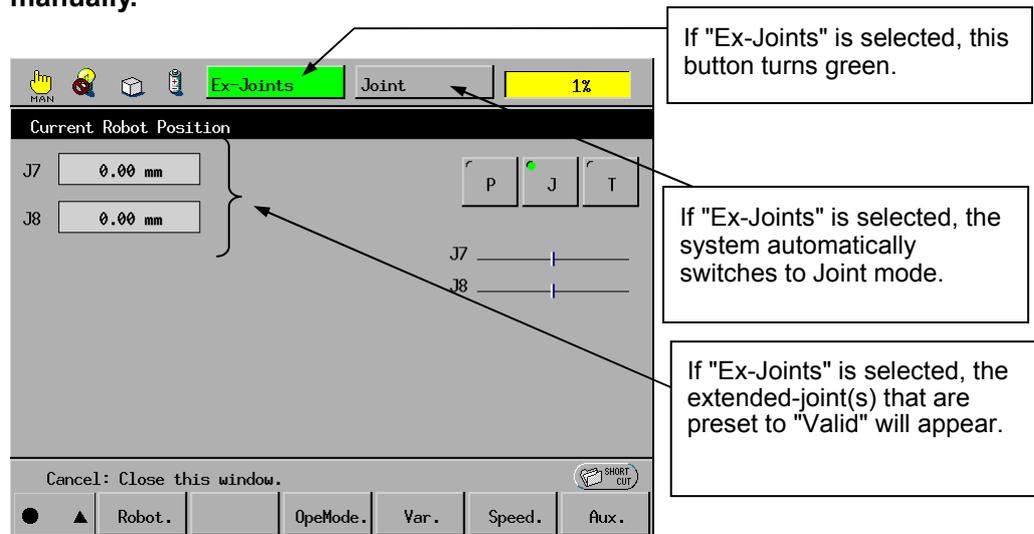
**Note:** Extended-joints can operate only in Joint mode. Selecting Ex-Joints in the above window will automatically switch to Joint mode.

**NOTE:** In the Select Robot window, the "Ex-Joints" appears only when any extended-joint is set to "Valid" in the servo configuration parameters window. If no "Ex-Joints" appears, access [F2 Arm]—[F12 Maint.]—[F7 Joints]—[F8 Servo] and then enable the desired extended-joint.

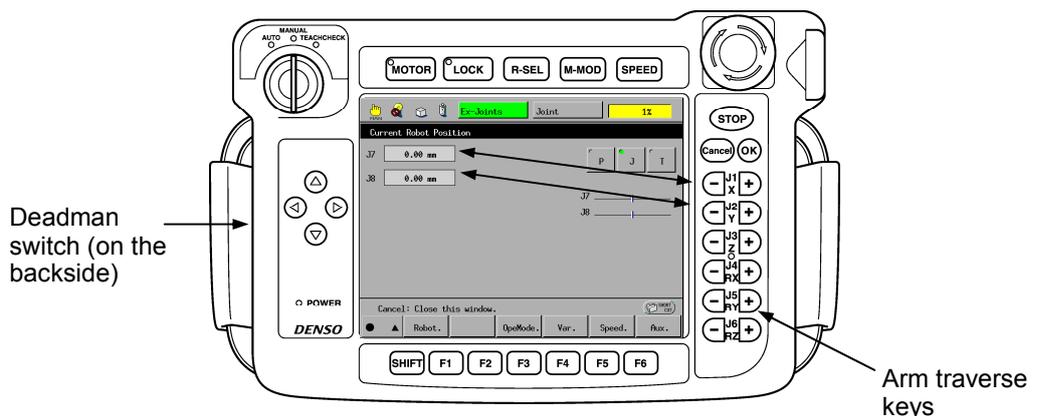
For more details, refer to Section 3.2 "Configuring the Extended-Joint Parameters."

**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 can operate the extended-joints manually.



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

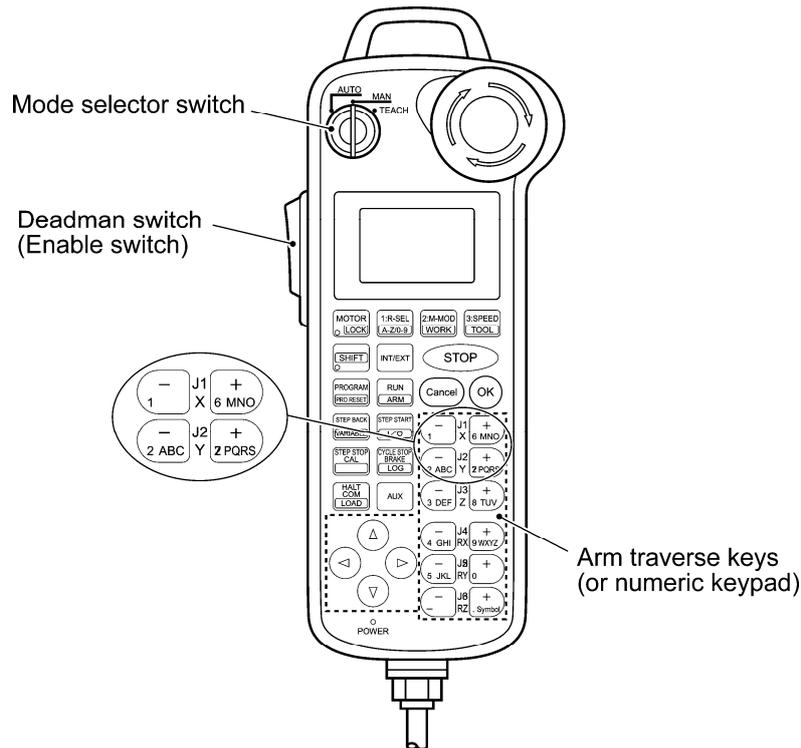


**Note:** Assignment of joints to the arm traverse keys is arranged from the top line first, as shown above. In this figure, you can move the 7th joint by using the [+ J1 X] and [- J1 X] keys.

■ **From the mini-pendant**

- (1) Press the [1:R-SEL] key and select "Ex-Joints."
- (2) Turn the mode selector switch to the MAN position.
- (3) Press the MOTOR key to turn the motor on.
- (4) While holding down the deadman switch, press one of the arm traverse keys to run the motor.

To move the 1st extended joint, use the [+ J1 X] and [- J1 X] keys; to move the 2nd extended joint, use the [+ J2 X] and [- J2 X] keys.

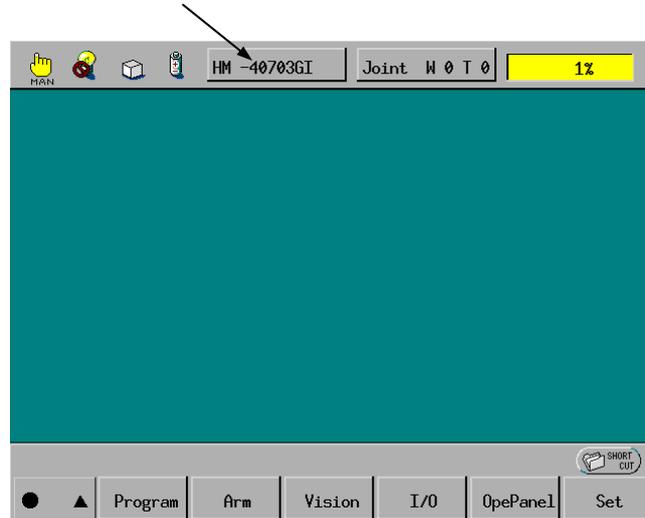


### [ 2 ] Getting extended-joint positions

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

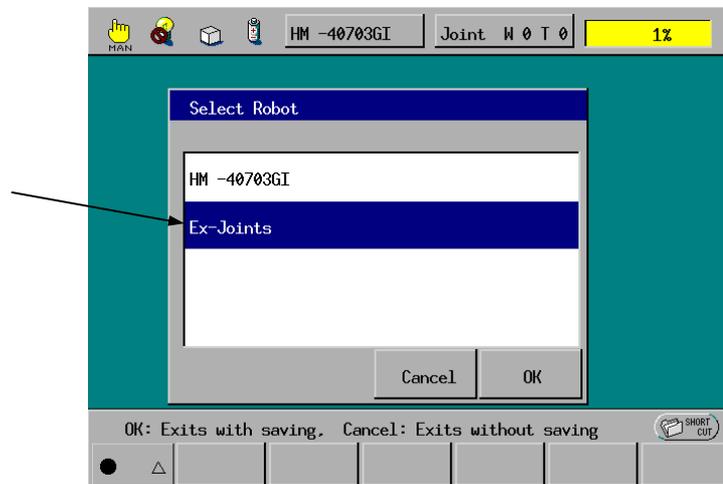
■ **From the teach pendant**

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



The Select Robot window will display as shown below.

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

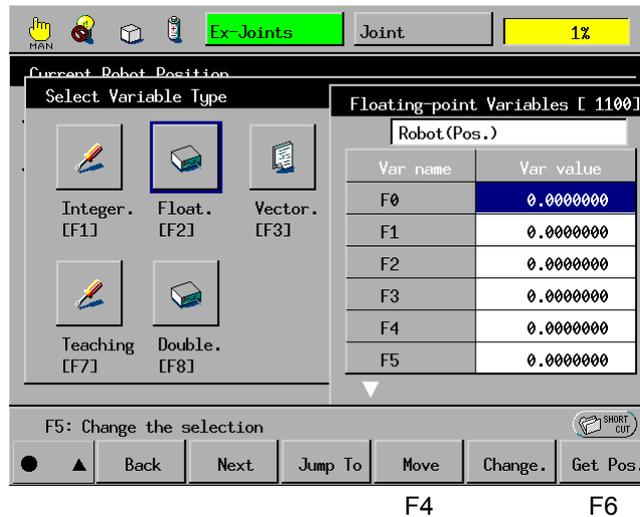


**NOTE:** In the Select Robot window, the "Ex-Joints" appears only when any extended-joint is set to "Valid" in the servo configuration parameters window. If no "Ex-Joints" appears, access [F2 Arm]—[F12 Maint.]—[F7 Joints]—[F8 Servo] and then enable the desired extended-joint.

For more details, refer to Section 3.2 "Configuring 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.

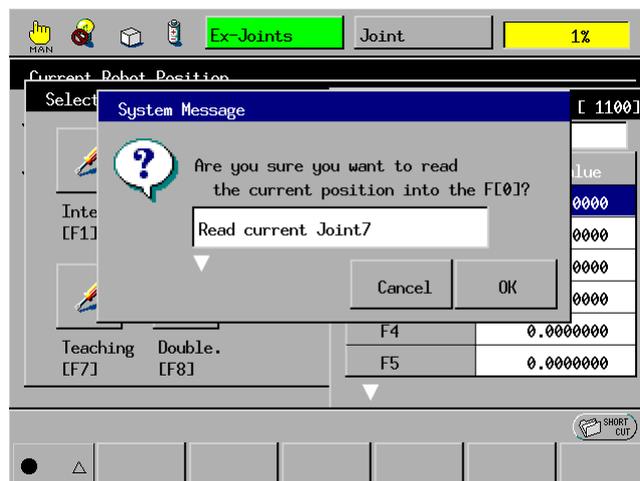


This example will get the extended-joint position value into variable F0.

**NOTE:** The [Move] and [Get Pos.] function buttons for the floating-point variables appear 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 configuration parameters 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 mini-pendant**

**Step 1** Press [VARIABLE] to call up the screen which lists variables used in your robot controller.

M	Ex	J		100
I	[	0/	100]	
				0
I	[	1/	100]	
				0
I	[	2/	100]	
				0

When the robot controller is turned on, the integer variables are displayed by default as shown above.

**Step 2** Use the right and left cursor keys to select the desired variable type.  
(The screen cycles through "I ↔ F ↔ D ↔ V ↔ P ↔ J ↔ T ↔ I.")

M	Ex	J		100
T	[	0/	50]	
X	:	0.	00000000	
Y	:	0.	00000000	
Z	:	0.	00000000	
Ox	:	0.	00000000	

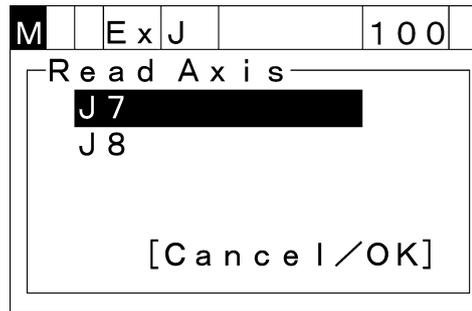
**Step 3** With the desired variable type displayed, press OK to call up the "Function" screen to handle the selected variable.

To make the "Function" screen disappear, press Cancel.

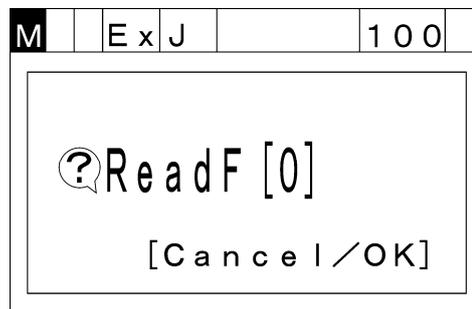
**NOTE:** There are two types of "Function" screens depending upon the variable type. For I, D, V, P, J, and T variables, four items (JumpTo, Move, Change, and Copy) appear; for F variables, five items (JumpTo, Move, Change, GetPos, and Copy) appear.

M	Ex	J		100
Function				
• JumpTo				
• Move				
• <b>Change</b>				
• Copy				
[Cancel/OK]				

**Step 4** Use the up and down cursor keys to select "GetPos" and then press OK, and the joint selection screen appears as shown below.



Use the up and down cursor keys to select the desired joint and then press OK, and the current position reading screen appears as shown below.



Press OK to get the current robot position into the specified variable.

To cancel the new setting, press Cancel, and the screen returns to the variable list one shown in Step 1.

### [ 3 ] Getting extended-joint positions in conjunction with robot joint positions

This feature enables the robot controller to get the current positions of extended-joints in conjunction with those of robot joints into variables. It is useful in teaching for the concurrent drive of extended-joints or the drive of extended-joints in conjunction with robot joints, as programmed below.

```
drivea (7,F107),(8,F108)
move p,p5 exa ((7,F107),(8,F108))
```

The teach pendant provides two position intake procedures--successive and concurrent position intake procedures.

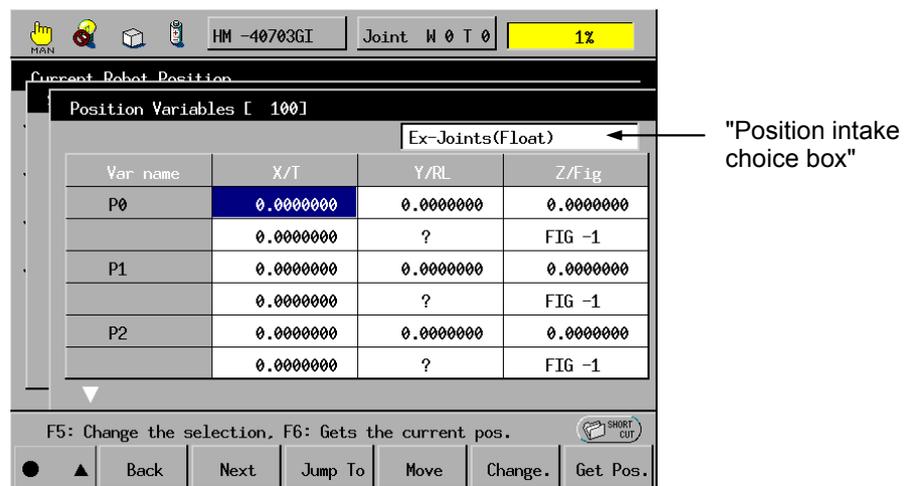
The concurrent position intake requires configuring the combination of variables for robot joint positions and those for extended-joint positions as a teach point beforehand. It gets robot joint positions and extended-joint positions concurrently into respective variables.

#### ■ Successive position intake procedure using the teach pendant

**(Example)** Get the current position of the robot joints into the specified position variable and then get that of extended-joints in succession

**Step 1** From the top screen, call up the Position Variables window by accessing:

**[F2 Arm]—[F4 Var.]—[F4 Pos.]**



F6

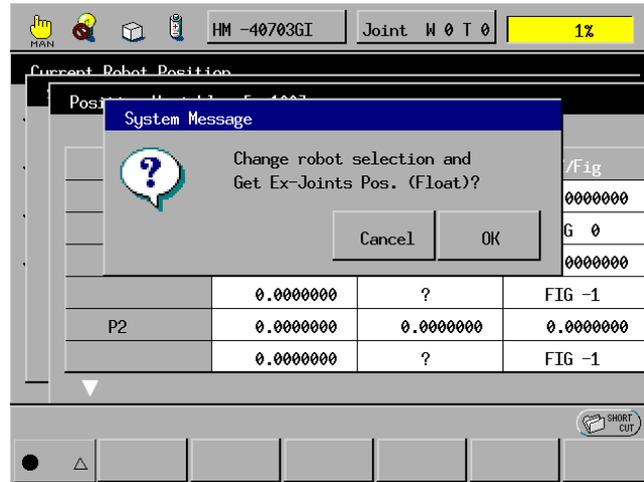
**NOTE:** In either of the following cases, the "Position intake choice box" appears when the extended-joint robot system is selected.

- With Ex-Joints being listed in the Select Robot window (see item [ 2 ], Step 2), any of the "Pos.," "Joint." and "Tran." is selected.
- With the Ex-Joints selected in the Select Robot window, the "Float." is selected.

**Step 2** Select a variable into which the position data should be gotten and then press [F6 Get Pos.].

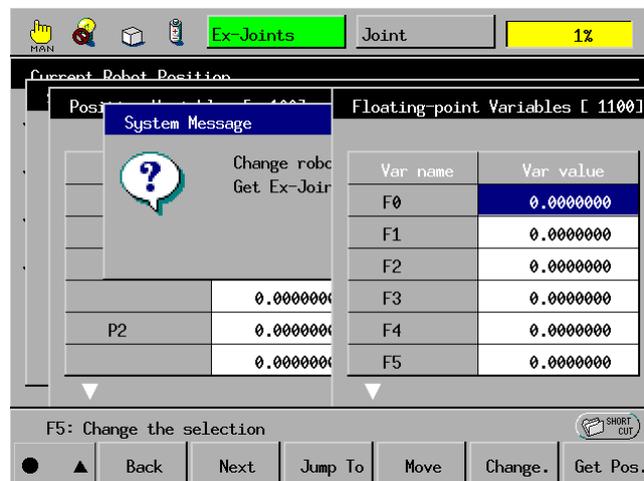
The current position of the robot joints will be gotten into the variable.

- Step 3** The following system message appears, prompting you to confirm whether or not to proceed to an intake of the current position of the extended-joints in succession. Press OK to get the current position of the extended-joints. Pressing Cancel returns to Step 1.



**NOTE:** If "No intake of extended-joint position" is selected in the "Position intake choice box," the system message shown above does not appear, causing no successive data intake of the extended-joint position.

- Step 4** Wait for the robot selection area to show Ex-Joints and the floating-point variables window to appear as shown below.



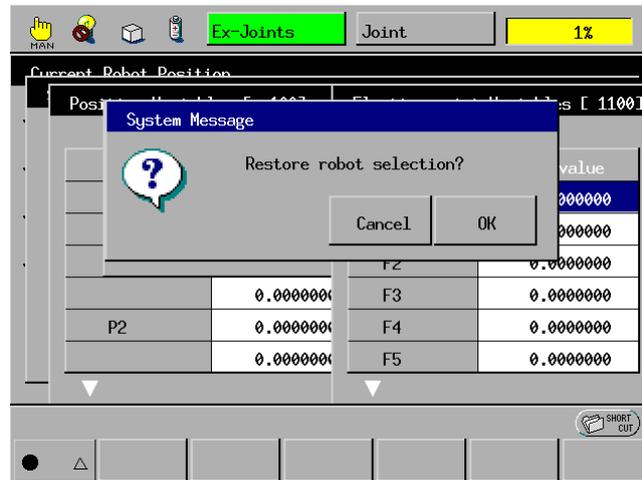
- Step 5** Select a variable into which the position data should be gotten and then press [F6 Get Pos.]. The current position of the extended-joint will be gotten into the variable.

**Step 6** Press Cancel to close the floating-point variables window.

The following system message appears, prompting you to confirm whether or not to return to robot selection state.

Pressing OK returns to the robot selection state.

Pressing Cancel keeps the extended-joint selection state.



**Step 7** With the robot or extended-joint selection state specified in Step 6, the screen returns to the Position Variables window.

Either of the sequences "Getting robot joint positions first and then extended-joint positions" and "Getting extended-joint positions first and then robot joint positions" is available.

■ **Concurrent position intake procedure using the teach pendant**

First, it is necessary to configure the combination of variables (position, joint, and homogeneous transform matrix variables) to be used for getting robot joint positions and variables for getting extended-joint positions. The combination uses a set of 10 consecutive floating-point variables per teaching point.

Next, specify a variable into which the position data should be gotten and a target joint, to an individual teaching point.

Last, select a desired teaching point and press [Get Pos.] to get individual joint data to all of the variables specified.

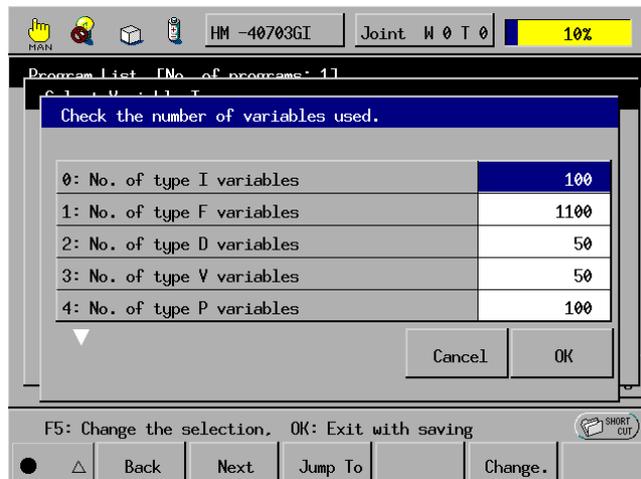
**Step 1** Reserving floating-point variables

Reserve the sufficient number of floating-point variables beforehand since getting position data uses those variables 10 times the number of teaching points to be used.

To reserve 100 teaching points, for example, specify the number of floating-point variables so that 1000 floating-point variables can be used for teaching points.

To reserve 100 teaching points, for example, specify 1000 floating-point variables to be used for teaching points.

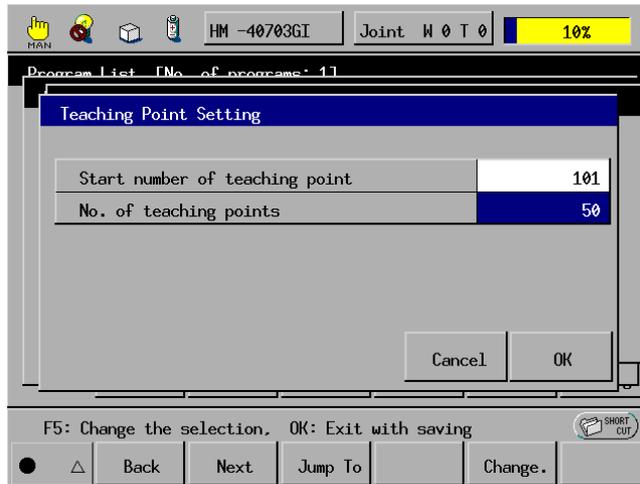
**Access: [F1 Program]--[F4 Var.]--[F12 VarsUsed.]**



### Step 2 Configuring teaching points with variables

Specify the range of floating-point variables to be used as teaching points

**Access:** [F1 Program]--[F4 Var.]--[F7 Teaching]--[F8 SetPoint]



Start number of teaching point: Specify the start number of floating-point variables to be used for 10 variables per teaching point.

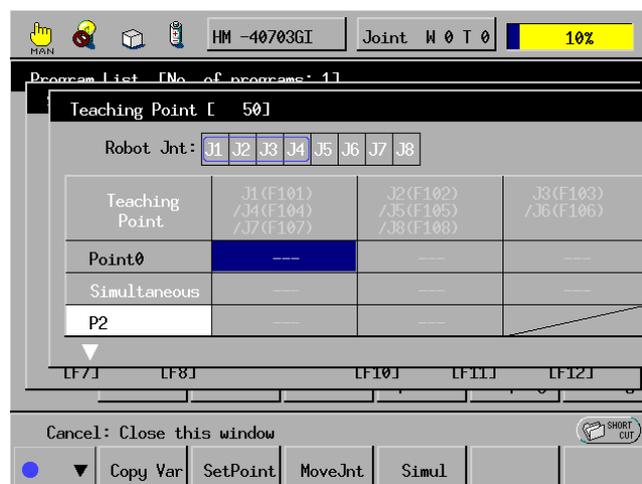
No. of teaching points: Specify how many teaching points should be configured.

(Example) The screen given on the previous page specifies floating-point variables F101 to F600 to use for teaching points. Teaching point 0 uses F101 to F110.

### Step 3 Configuring teaching points with joints

Specify the target joints whose position data should be gotten at each of teaching points.

Robot joint position data can be gotten into not only floating-point variables but also position, joint, and homogeneous transform matrix variables.



## Step 4

Associating variables with joints using teaching points

Floating-point variables are automatically associated with joints 1 to 8, starting with the first variable in a set of 10 variables.

(Example) Associating variables using teaching point 0 when the start number of teaching point is 101

Joint 1: F101

Joint 2: F102

Joint 3: F103

Joint 4: F104

Joint 5: F105

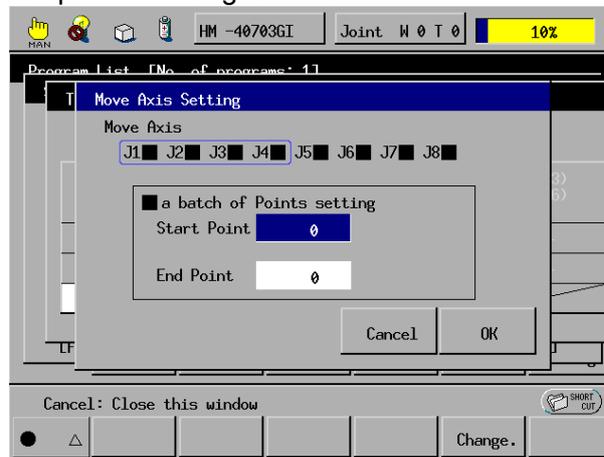
Joint 6: F106

Joint 7: F107

Joint 8: F108

F109 and F110 are reserved for teaching point configuration data. Do not edit those variables.

In the Teaching Point window, select the teaching point to configure and then press [F9 Select.J] to call up the following window.



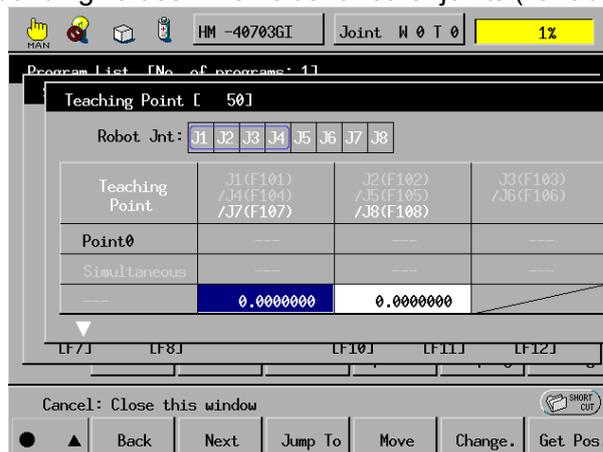
In the "Selected joints" area, press the target joints. Joints enclosed in a box are robot joints so that they cannot be individually selected as a target joint.

Robot joint data can be gotten into not only floating-point variables but also other types of variables. Note that specifying other types of variables no longer associates robot joints with floating-point variables.

Press "Batch point setting" to turn the left-hand box green. It enables you to specify target joints for two or more teaching points at the same time.

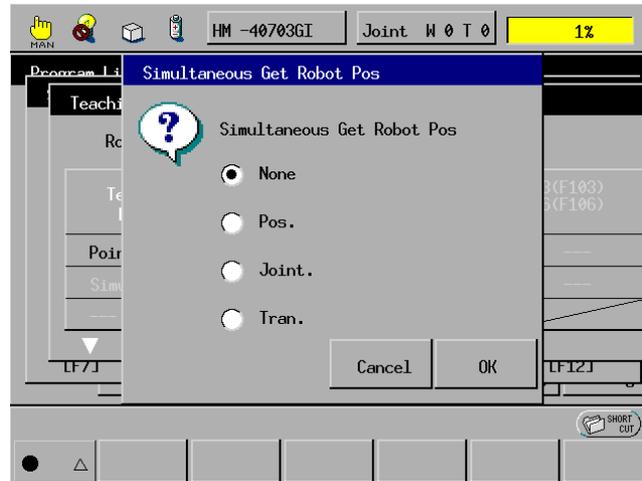
Fill in the "Start point" and "End point" to specify the range of the teaching points, and then press OK.

The following window appears where the fields for the specified joints (variables) show the corresponding values. The fields for other joints (variables) are dimmed.



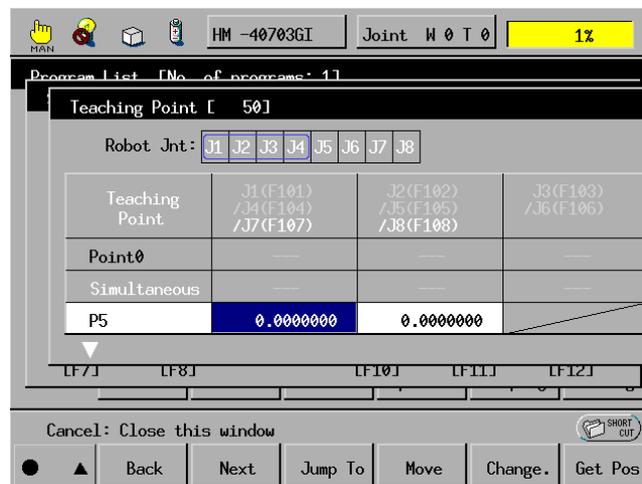
**Step 5** Associating robot joints with variables (position, joint, and homogeneous transform matrix variables) except floating-point ones

In the Teaching Point window, select the desired teaching point and press [F10 Ver type] to display the following window.



In the window shown above, select the desired variable type and press OK to call up the numerical keypad. Enter the desired variable number and press OK to define the variable.

As shown below, the specified variable name appears in the corresponding variable field.



Even if robot joints have been associated with floating-point variables, specifying variables other than floating-point ones for robot joints above automatically keeps robot joints out of targets of floating-point variables.

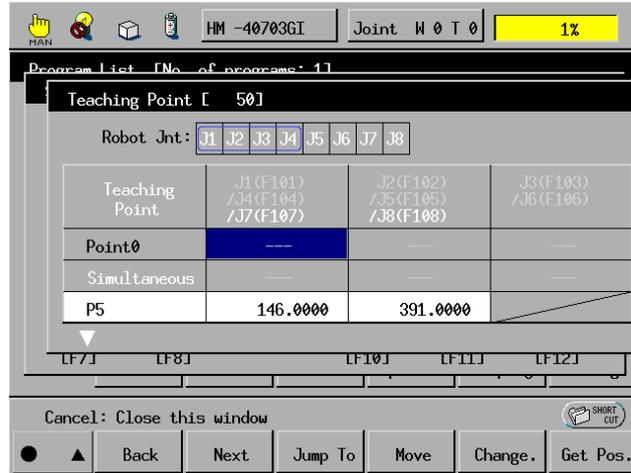
Steps 4 and 5 configure individual teaching points with target joints and variables.

## Step 6

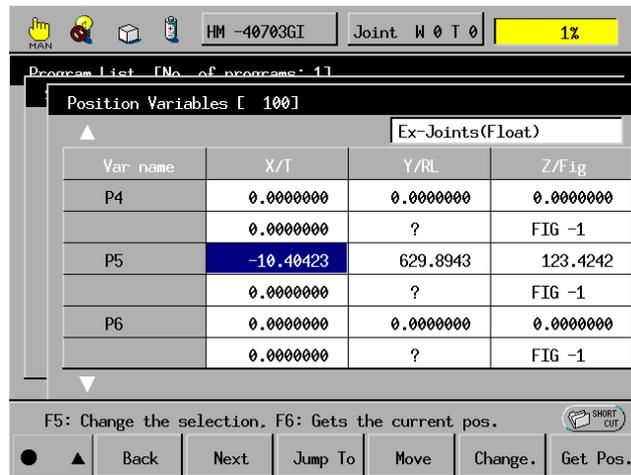
Getting position data of robot joints and extended-joints concurrently at a teaching point

Call up the teaching point at which the position data should be gotten into the previously specified variables.

Press [F6 Get Pos.] to get position data.

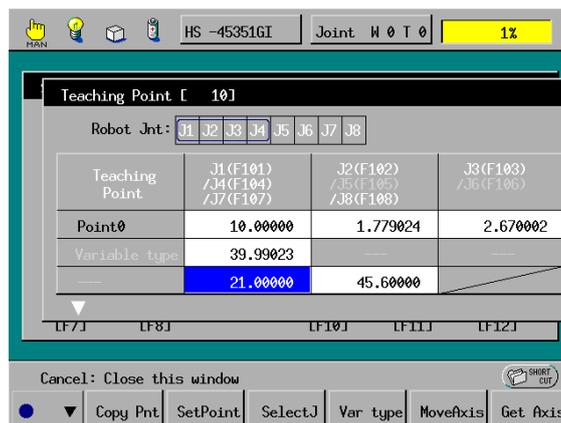


As shown below, position data has been also gotten into position variables.



### Note

In the Teaching Point window, it is possible to get position data of an individual joint. Select the target joint with the cursor and press [F12 Get Axis] to get the position data of the selected joint.



■ **From the mini-pendant**

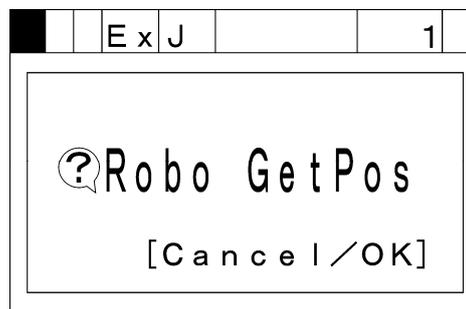
**(Example)** Get the current position of the extended-joint into the specified variable and then get that of the robot joints in succession

**Step 1** Select a floating-point variable and get the current position of the extended-joint into it.

**Step 2** The following system message appears, prompting you to confirm whether or not to proceed to an intake of the robot joints in succession.

Press OK to get the current position of the robot joints.

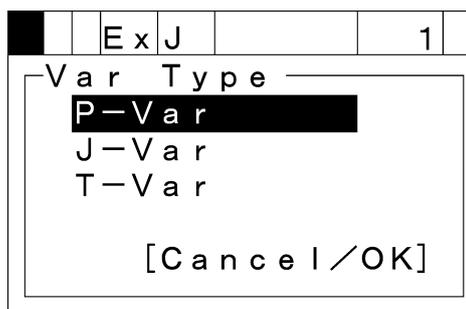
Pressing Cancel returns to the floating-point variables screen.



**Step 3** Wait for the variable type choice screen to appear and then select the desired variable type.

Press OK to continue an intake of the current position of the robot joints.

Pressing Cancel returns to the floating-point variables screen.

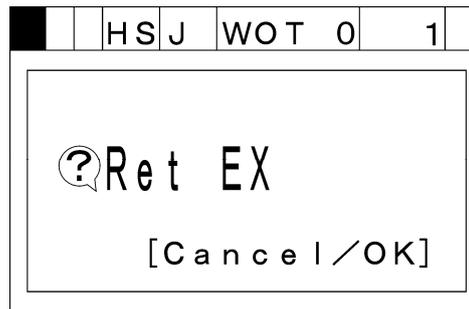


**NOTE:** The variable type choice screen shown above does not appear if an intake of the current robot position starts with "Pos.," "Joint." or "Tran." being selected.

---

**Step 4** | Wait for the status bar to show the robot selection state and the variable selected in Step 3 to appear.  
The current position of the robot joints will be gotten into the variable.

**Step 5** | Press Cancel to close the variable selected in Step 3.  
The following system message appears, prompting you to confirm whether or not to return to the extended-joint selection state.  
Pressing OK returns to the extended-joint selection state.  
Pressing Cancel keeps the robot selection state.



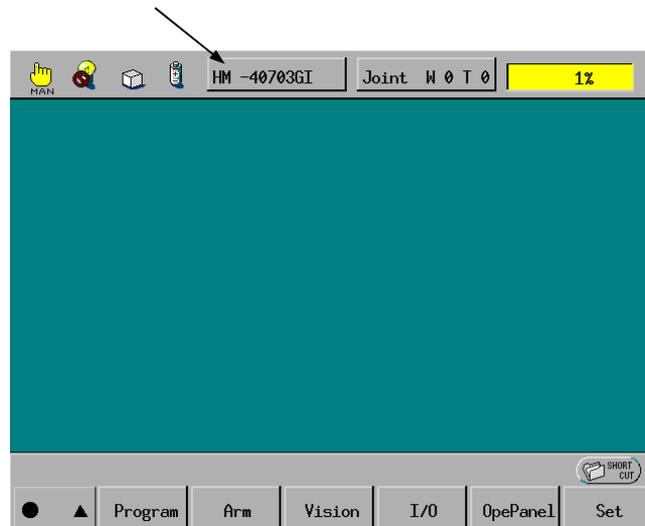
**Step 6** | With the robot or extended-joint selection state specified in Step 5, the screen returns to the floating-point variables screen.

### [ 4 ] Operating extended-joints using variables

Robots equipped with extended-joints can be operated with variables in two ways--moving an individual extended-joint with a floating-point variable and moving both robot joints and extended-joints concurrently with the teaching point feature.

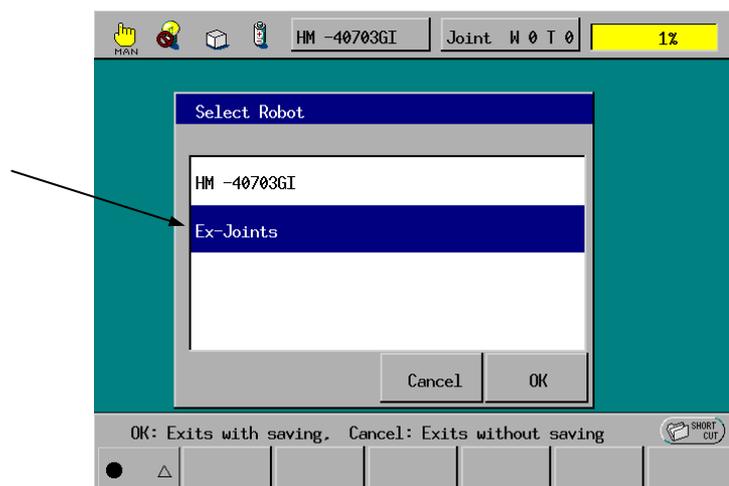
■ **Moving an individual extended-joint with a floating-point variable, using the teach pendant**

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



The Select Robot window will display.

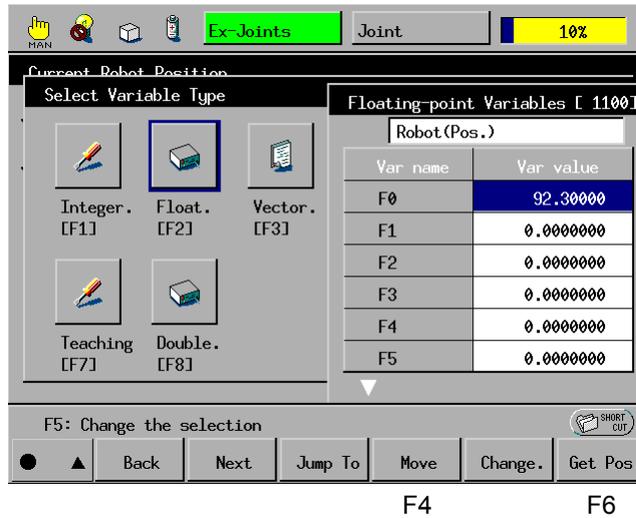
**Step 2** Select the "Ex-Joints" and then press the OK button.



**NOTE:** In the Select Robot window, the "Ex-Joints" appears only when any extended-joint is set to "Valid" in the servo configuration parameters window. If no "Ex-Joints" appears, access [F2 Arm]—[F12 Maint.]—[F7 Joints]—[F8 Servo] and then enable the desired extended-joint.

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

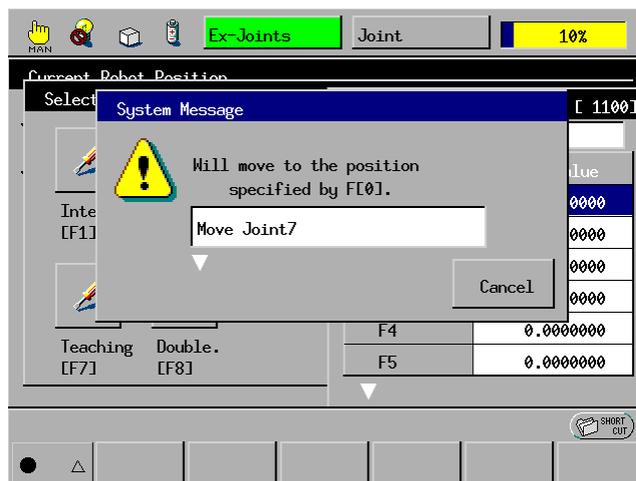
- Step 3** From the top screen of the teach pendant, access [F2 Arm]—[F4 Var.]—[F2 Float.] to call up the Floating-point Variables window. Select a variable having the desired movement value.



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 appear only when the "Ex-Joints" is selected in the Select Robot window.

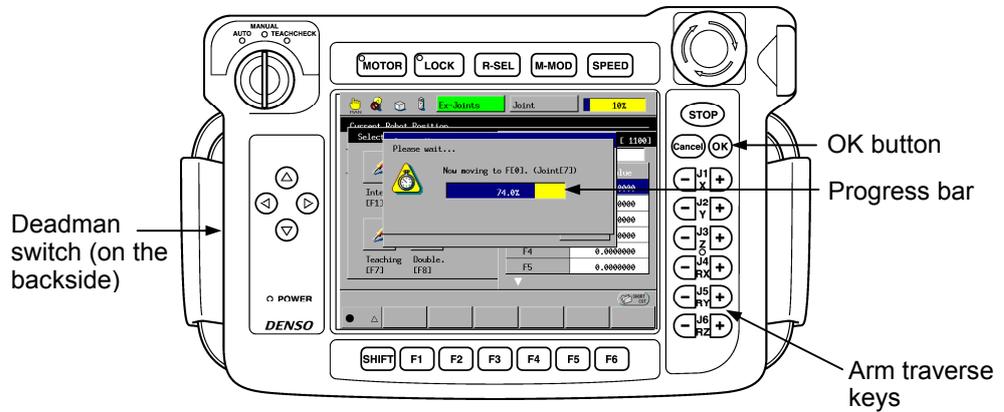
- Step 4** Press [F4 Move]. The window below appears. Select an extended-joint you want to move by using the floating-point variable specified in Step 3.



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 configuration parameters window may be read into floating-point variables. No robot joints can be read.

- Step 5** While holding down either one of the deadman switches, keep on pressing the OK button until the progress bar shows 100%.

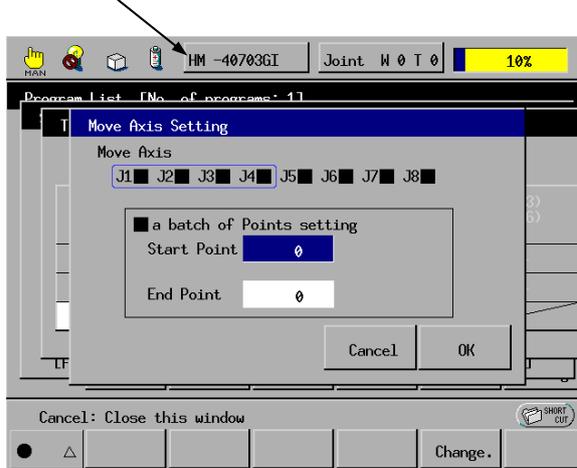


The above procedure moves the 7th joint to the position 92.30 assigned to floating-point variable F0.

**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 section "[ 5 ] Operating extended-joints by program."

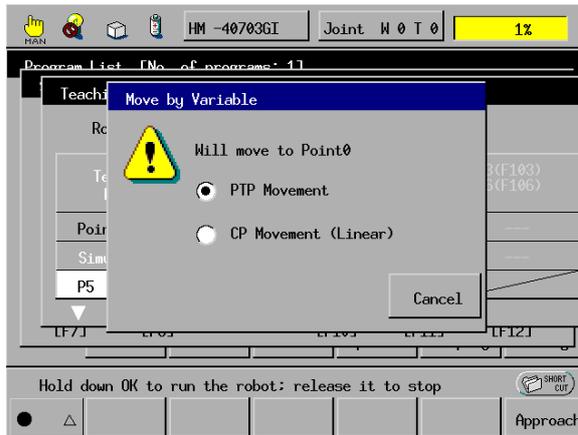
■ **Moving both robot joints and extended-joints concurrently with the teaching point feature**

**Step 1** Call up the previously defined teaching point and press [F4 Move].



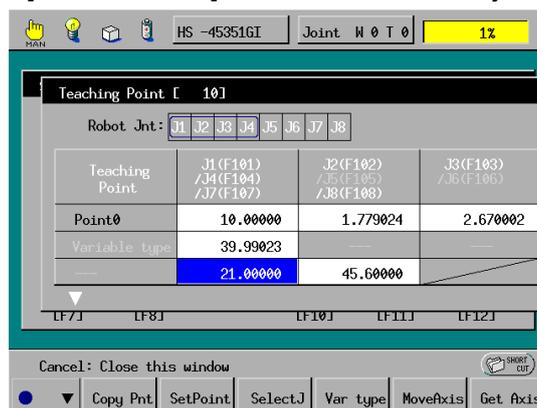
**Step 2** Select the PTP or CP motion and then press the OK button with the deadman switch held down.

While the OK button is being pressed, the end-of-arm tooling moves to the teaching point displayed in the Teaching Point window.



**Note**

In the Teaching Point window, it is possible to move an individual joint using a variable. Select the target joint with the cursor and press [F11 MoveAxis] to move the selected joint using a variable.



■ **From the mini-pendant**

**Step 1** Press [VARIABLE] to call up the screen which lists variables used in your robot controller.

M	E x	J		1 0 0
I [ 0 / 1 0 0 ]				
0				
I [ 1 / 1 0 0 ]				
0				
I [ 2 / 1 0 0 ]				
0				

When the robot controller is turned on, the integer variables are displayed by default as shown above.

**Step 2** Use the right and left cursor keys to select the desired variable type.  
(The screen cycles through "I ↔ F ↔ D ↔ V ↔ P ↔ J ↔ T ↔ I.")

M	E x	J		1 0 0
T [ 0 / 5 0 ]				
X : 0. 0 0 0 0 0 0 0				
Y : 0. 0 0 0 0 0 0 0				
Z : 0. 0 0 0 0 0 0 0				
O x : 0. 0 0 0 0 0 0 0				

**Step 3** With the desired variable type displayed, press OK to call up the "Function" screen to handle the selected variable.

To make the "Function" screen disappear, press Cancel.

**NOTE:** There are two types of "Function" screens depending upon the variable type. For I, D, V, P, J, and T variables, four items (JumpTo, Move, Change, and Copy) appear; for F variables, five items (JumpTo, Move, Change, GetPos, and Copy) appear.

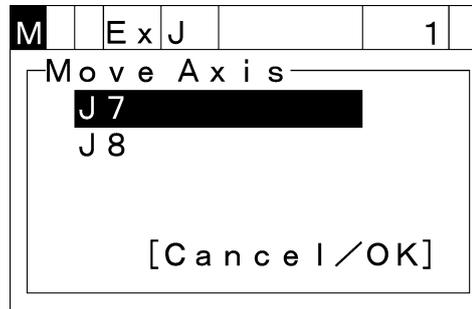
M	E x	J		1 0 0
Function				
• JumpTo				
• Move				
• Change				
• Copy				
[Cancel/OK]				

---

**Step 4**

Use the up and down cursor keys to select "Move" and then press OK, and the joint selection screen appears as shown below.

Pressing Cancel instead of OK cancels the new setting and returns to the variable list shown in Step 1.



Use the up and down cursor keys to select the desired joint.

While holding down the deadman switch, keep on pressing the OK button, and the selected extended-joint moves the specified position. (Only in Manual or Teach check mode, an extended-joint can move.)

### [ 5 ] 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 Configuration (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 can operate the extended-joints.

In the above configuration 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 Deadman switch (on the backside)

■ **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**

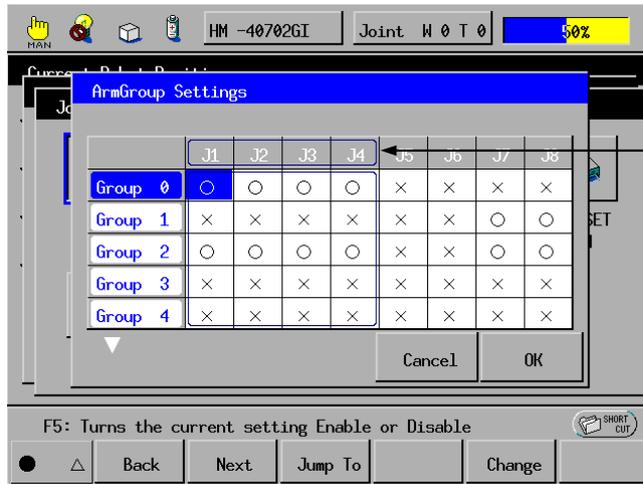
Executing a GIVEARM command releases the currently held arm group.

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 Joints]—[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.

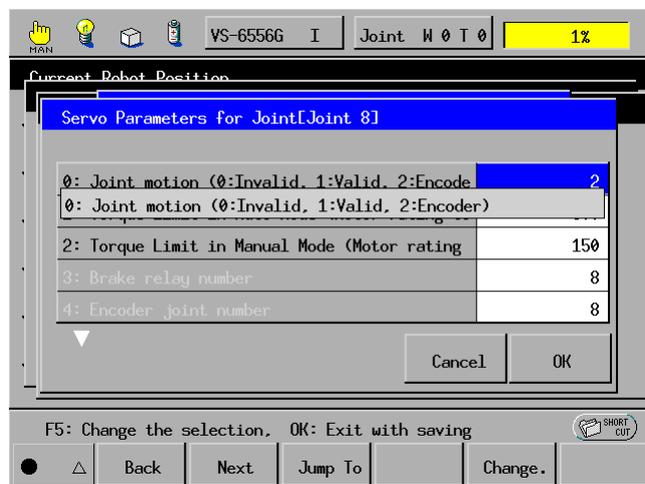
### Notes for setting up an arm group

- (1) New settings will go into effect when the controller is turned off and then on after the settings are 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 configuration parameters window beforehand. If not, an error will occur when you are attempting to set up an arm group.

Access to the servo configuration parameters window is:

**[F2 Arm]—[F12 Maint.]—[F7 Joints]—[F8 Servo]**

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



### [ 6 ] 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. You need to set the boundless rotation parameter to [1: Boundless] in the path configuration parameters window.

#### Notes for allowing boundless rotation on extended-joints

- (1) When an extended-joint is used as a rotary joint, an absolute motion command (`DRIVEA` or `MOVE` with `EXA` option) can drive it within the range of  $\pm 360^\circ$ . When it is used as a sliding joint, the allowable motion range is  $\pm 32768$  in the number of motor rotations from the reference position (CALSET position).
- (2) If a movement of a rotary joint exceeding the range of  $\pm 360^\circ$  is commanded, the joint rotates the specified angle and then it automatically returns to the position within the range of  $\pm 360^\circ$ . This correction operation changes the reference position (CALSET position). Therefore, the Step Back function cannot return the program control back to the steps preceding the change of the reference position.
- (3) When an extended-joint keeps on rotating in the same direction, the current value might jump (overflow) suddenly and greatly. Performing an absolute motion in this state moves the extended-joint to the position different from the specified one.
- (4) In a boundless rotation motion command, the effective number of digits is 7. 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` so that `5555` will be trimmed 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.
- (6) When an extended-joint is used as a rotary joint requiring positioning, e.g., index table, observe the following instructions.
  - For the reduction gear, enter an integer multiple. Entering a non-integer multiple will result in a positioning error after a lot of rotations.
  - If a relative motion command specifies a motion amount using a decimal, the extended-joint could reach the position slightly different from the specified one. Using such a relative motion command repeatedly will result in a positioning error after a lot of rotation.

To avoid such a positioning error, correct the difference from the specified position, for example, using an absolute motion command or `MoveIndexHome` library after completion of one rotation to return the extended-joint to the home position.
- (7) Rotating an extended-joint exceeding  $\pm 32768$  in the number of motor rotations from the reference position (CALSET position) with the controller power being off will require CALSET operation when the controller is powered on at the next time.

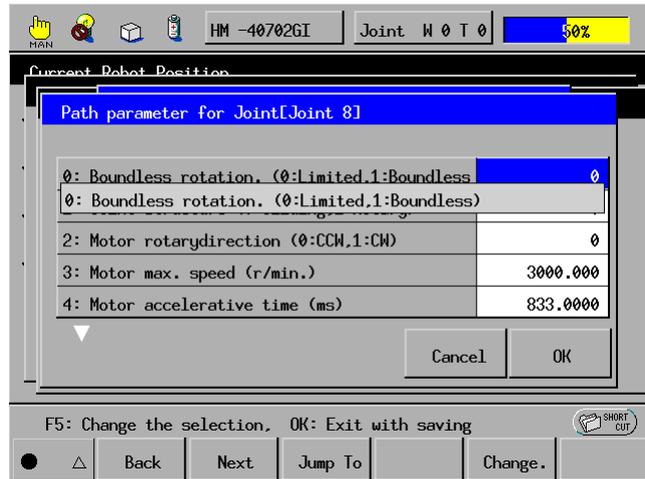
### ■ *Allowing boundless rotation*

From the top screen of the teach pendant, access:

**[F2 Arm]—[F12 Maint.]—[F7 Joints]—[F7 Path]**

In the path configuration parameters window, set the boundless rotation parameter to [1: Boundless].

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



### [ 7 ] Setting the link info

The global type of the controller always monitors the motion speed of the robot joints being driven and limits it to a maximum of 250 mm/sec in Manual or Teach check mode.

When driving extended-joints also in Manual or Teach check mode, the controller applies the same speed limit to them.

If the link info is set up between robot joints and extended-joint(s), the controller can limit the total motion speed of the linked joints to a maximum of 250 mm/sec. Note that the total motion speed differs from the actual composite speed since the controller does not take the motion directions of individual joints into account.

#### ■ Concept of the link info

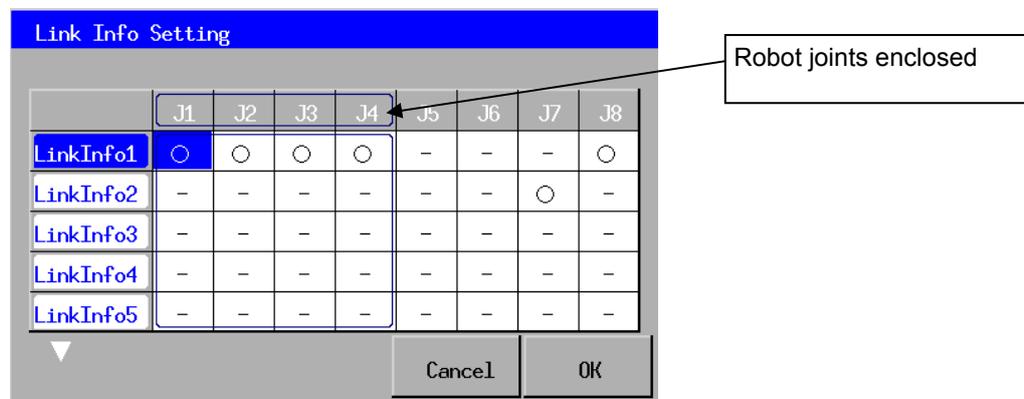
"Link info" defines whether or not robot joints and extended-joints affect with each other when they are being driven.

If the link info is set up, the linked joints are mutually affected; if not, they are driven independently.

All of robot joints are configured in a batch so that all of them are either linked or not.

An example of the link info setup window is given below.

Link info setup window (ex. 4-joint robots. J5 and J6 are invalid.)



○: Linked, -: Not linked

In the above example,

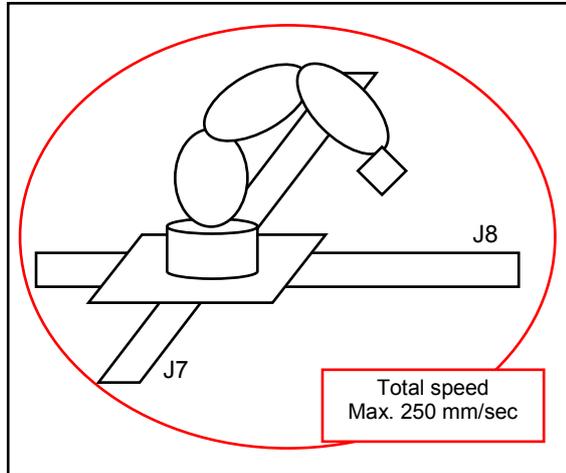
Link info 1 links robot joints J1 to J4 and extended-joint J8 together so that the robot joints and J8 are mutually affected in motion.

Link info 2 does not link extended-joint J7 with any joints so that J7 affects neither robot joints nor J8 in motion.

■ **Configuration samples of robot joints and extended-joints**

Given below are configuration samples and their link info to be set up.

< **Configuration sample 1** > Linking robot joints and extended-joints (J7 and J8) together



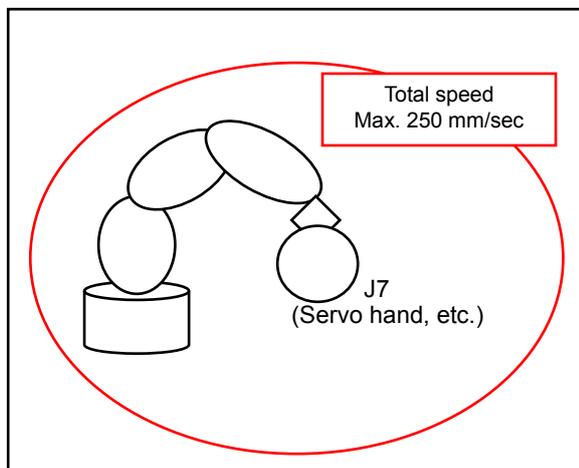
In the configuration shown at left, the motions of extended-joints J7 and J8 affect those of the robot joints, so link the robot joints and extended-joints together.

The speed limit applies to the total motion speed.

Link info setup sample (4-joint robots)

	J1	J2	J3	J4	J5	J6	J7	J8
LinkInfo1	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	-	-	<input type="radio"/>	<input type="radio"/>

< **Configuration sample 2** > Linking robot joints and extended-joint (J7) together



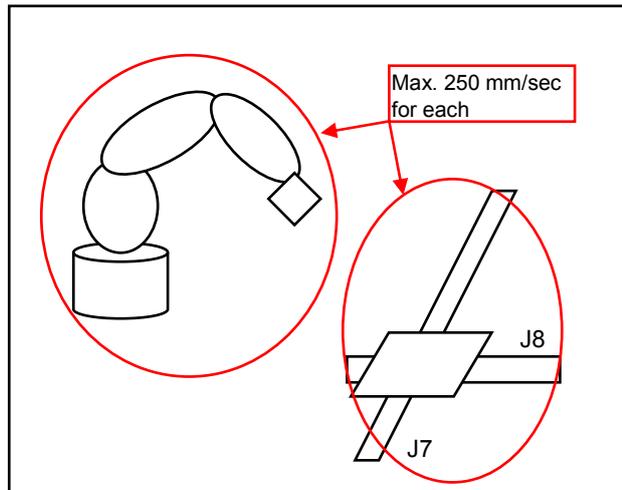
In the configuration shown at left, the motions of robot joints affect that of extended-joint J7, so link the robot joints and J7 together.

The speed limit applies to the total motion speed.

Link info setup sample (4-joint robots)

	J1	J2	J3	J4	J5	J6	J7	J8
LinkInfo1	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	-	-	<input type="radio"/>	-

**< Configuration sample 3 >** No link between robot joints and extended-joints, but linking extended-joints J7 and J8 together



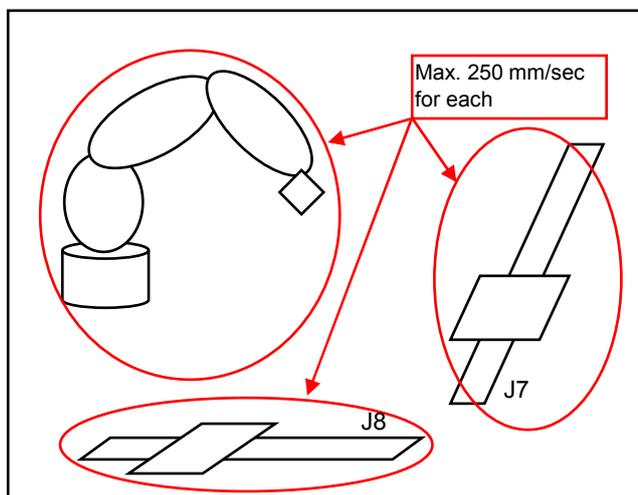
In the configuration shown at left, the motions of extended-joints J7 and J8 does not affect those of the robot joints, so do not link the robot joints and extended-joints together. But, the motions of J7 and J8 are mutually affected, so link them together.

The speed limit applies to each of the robot joints and extended-joints (J7 and J8) independently. But for J7 and J8, it applies to the total motion speed of them.

Link info setup sample (4-joint robots)

	J1	J2	J3	J4	J5	J6	J7	J8
LinkInfo1	○	○	○	○	-	-	-	-
LinkInfo2	-	-	-	-	-	-	○	○

**< Configuration sample 4 >** No link between robot joints and extended-joints or between extended-joints J7 and J8



In the configuration shown at left, the motions of extended-joints J7 and J8 does not affect those of the robot joints, so do not link the robot joints and extended-joints together. Neither do the motions of J7 and J8 mutually, so do not link them together.

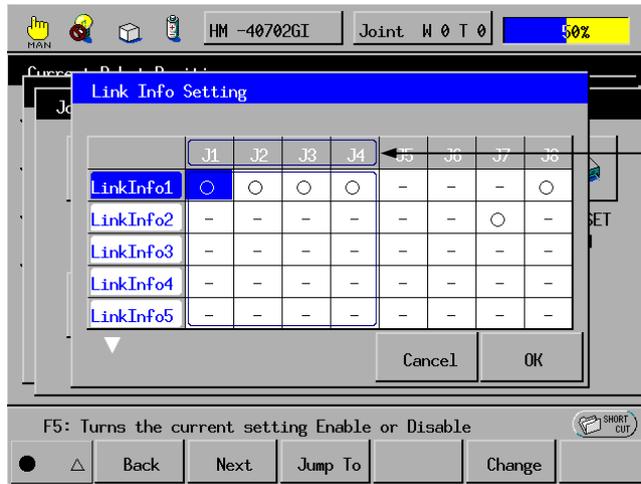
The motion speed limit applies to each of the robot joints, J7, and J8 independently.

Link info setup sample (4-joint robots)

	J1	J2	J3	J4	J5	J6	J7	J8
LinkInfo1	○	○	○	○	-	-	-	-
LinkInfo2	-	-	-	-	-	-	○	-
LinkInfo3	-	-	-	-	-	-	-	○

■ **Setting up link info**

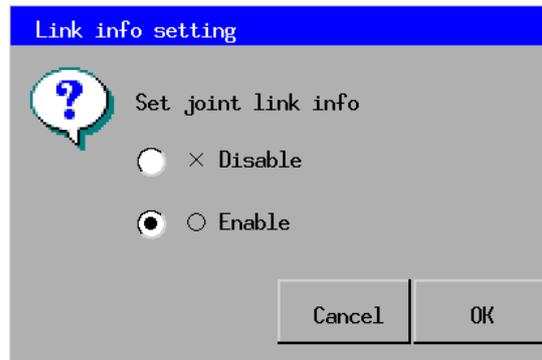
**Step 1** Call up the Link Info Setting window by accessing:  
**[F2 Arm]—[F12 Maint.]—[F7 Joints]—[F4 LinkInfo]**



Robot joints enclosed in this box

F5

**Step 2** Move the cursor to a joint to be linked and then press [F5 Change].  
 The setup window appears as shown below.



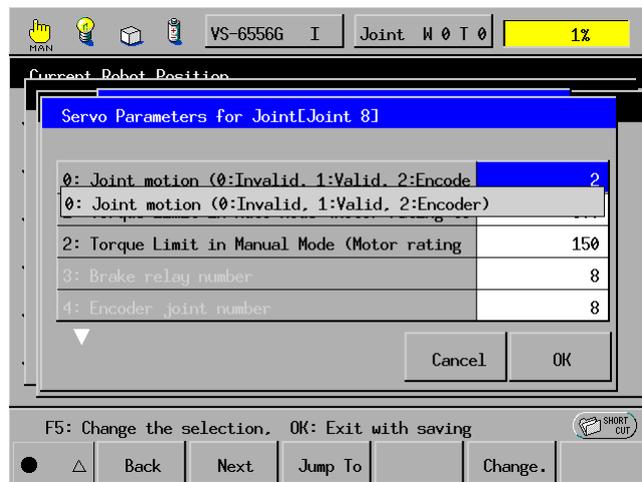
**Step 3** After selecting "O" or "X," press OK.

### Notes for setting up link info

- (1) The newly configured link info goes into effect when the controller is restarted.
- (2) All of robot joints are configured in a batch so that all of them are either linked or not.
- (3) Extended-joints to be selected for linking should be set to "Valid" in the servo configuration parameters window beforehand.

If not, selecting any extended-joint for linking causes an error (see System message window 1 on the next page). If such an error occurs, access [F2 Arm]—[F12 Maint.]—[F7 Joints]—[F8 Servo] and then enable the desired extended-joint.

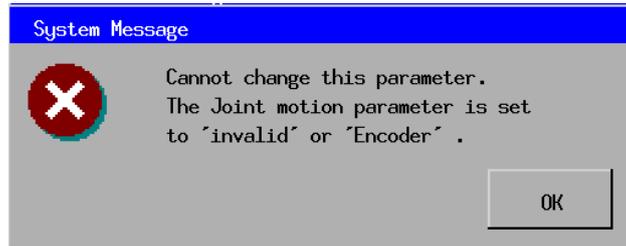
For more details, refer to Section 3.2 "Configuring the Extended-Joint Parameters."



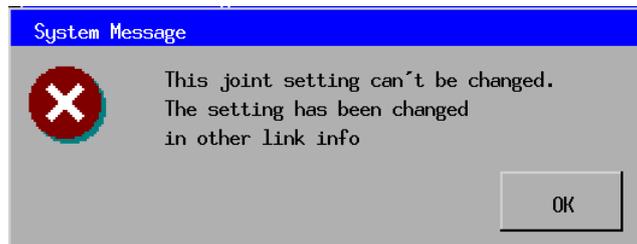
- (4) In the Link Info Setting window shown in Step 1, "-" denotes joints not specified as "Valid" or joints linked in any other joint info. Attempting to change the setting of joints marked with "-" will cause an error (see System message window 2 on the next page).
- (5) Link info should be set up for all of robot joints and extended-joints specified as "Valid." With any joint not selected for linking, pressing OK in Step 2 will cause an error (see System message window 3 on the next page).

---

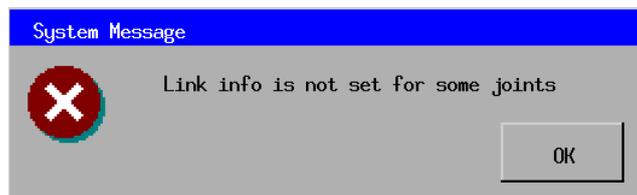
**System message window 1: Joint not specified as "Valid"**



**System message window 2: Joint already linked**



**System message window 3: Any joint not selected for linking**



### [ 8 ] Setting the radius of gyration

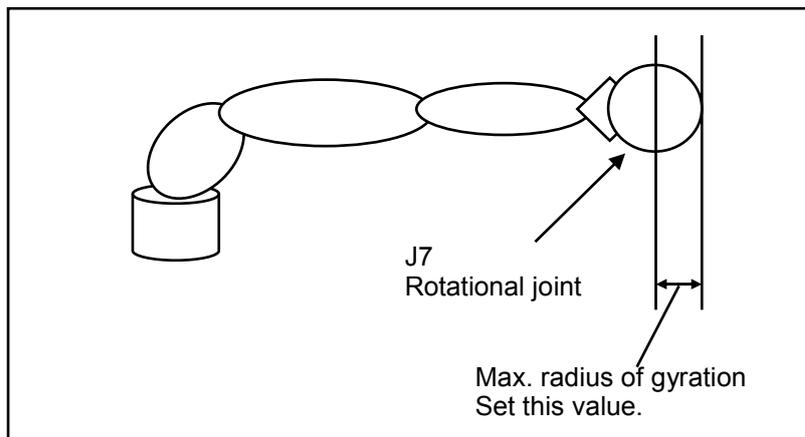
The global type of the controller limits the motion speed of both robot joints and extended-joints being driven to a maximum of 250 mm/sec in Manual or Teach check mode, as described in "[ 7 ] Setting the link info."

Using an extended-joint as a rotational joint requires specifying the radius of gyration of the extended-joint as a parameter.

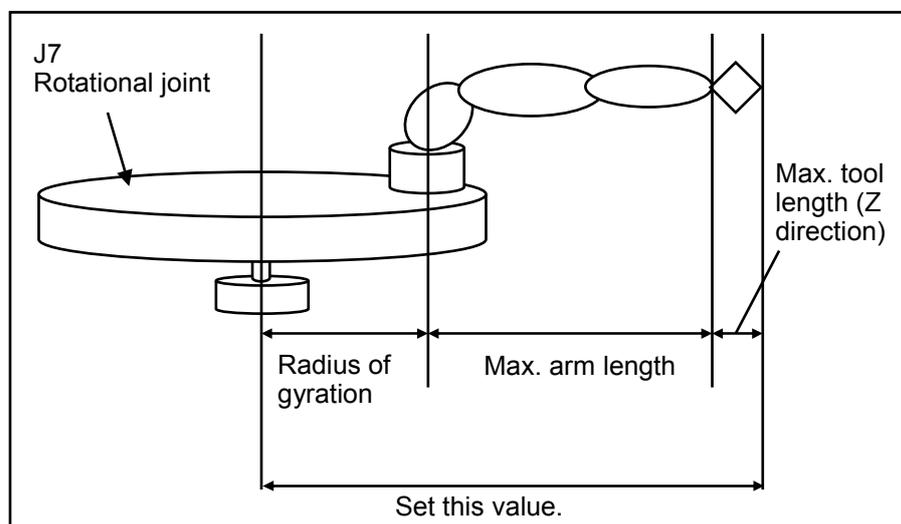
#### Notes for setting up the radius of gyration

- (1) When the robot is equipped with a tool (e.g., servo hand) having rotational joint J7 as shown in Configuration sample 1, specify the radius of gyration of the rotational joint as a parameter.
- (2) When the robot is mounted on rotational joint J8 (e.g., turntable) as shown in Configuration sample 2, it is necessary to specify the total value of the "Radius of gyration of J8," "Maximum arm length of the robot (Table 3.1)," and "Maximum Z-direction length of the tool" as a parameter.
- (3) In the combination of Configuration samples 1 and 2, that is, when the robot is mounted on rotation joint J8 (e.g., turntable) and it is equipped with a tool (e.g., servo hand) having rotational joint J7, it is necessary to specify the related parameters for each joint.

**< Configuration sample 1 >** Robot equipped with a tool (servo hand) having rotational joint J7



**< Configuration sample 2 >** Robot mounted on rotation joint J8 (turntable)



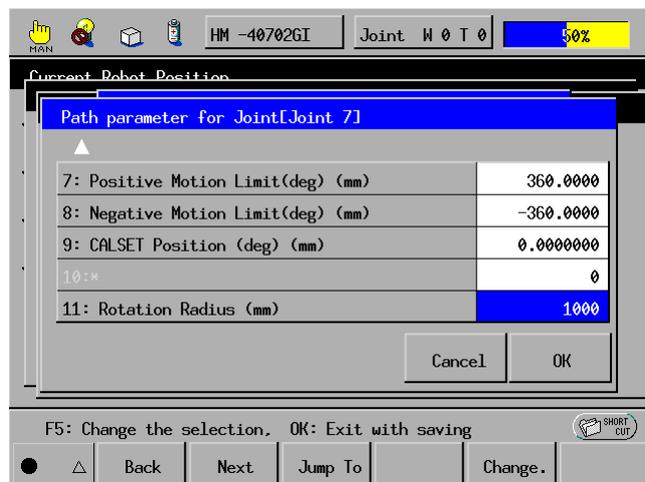
**Table 3.1 Robot Models and Max. Arm Length**

Robot series	Robot model	Max. arm length (mm)
VP-G	VP-5243G	430
	VP-6242G	432
VS-G	VS-6556G	653.42
	VS-6577G	854.88
	VS-6556G-B	653.42
	VS-6577G-B	854.88
VM-G	VM-6083G	1021.09
	VM-60B1G	1298.41
HS-G	HS-4535*G	350
	HS-4545*G	450
	HS-4555*G	550
HM-G	HM-4060*G	600
	HM-4A60*G	600
	HM-4070*G	700
	HM-4A70*G	700
	HM-4085*G	850
	HM-4A85*G	850
	HM-40A0*G	1000
	HM-4AA0*G	1000

■ **Specifying a "radius of gyration" parameter**

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

Enter the desired value to the "Radius of gyration" in the path configuration parameters.  
For more details, refer to Section 3.2 "Configuring the Extended-Joint Parameters."



### 3.1.2 Commands Supporting Extended-Joints

**NOTE:** Commands described in this section can be used for general robots, not exclusively used for extended-joints.

## TAKEARM (Statement)

### Function

Get an arm group.

Upon execution of this statement, the programmed internal 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[<ArmGroupNumber>] [<KEEP=DefaultValue>]

### Description

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

If <ArmGroupNumber> 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.
```

## Chapter 3 Extended-Joint Related Functions

- (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

Release the currently held arm group.

## Syntax

GIVEARM

## Description

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

Carry 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 MANUAL I, 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 0	○	○	○	○	×	×	×	×
Group 1	×	×	×	×	×	×	○	○
Group 2	○	○	○	○	×	×	○	○
Group 3	×	×	×	×	×	×	×	×
Group 4	×	×	×	×	×	×	×	×

```
PROGRAM PR01
TAKEARM 1 '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

Carry 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 MANUAL I, 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 PRO1  
TAKEARM 1 '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

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.

# MOVE (Statement)

## Function

Move 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 section explains the functions relevant to extended-joints. For others, refer to the PROGRAMMER'S MANUAL I, 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 I, 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) 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

Get 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 PR01
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

Get 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 lfl 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.
lfl = DESTEXJ(7)         'Assign target angle 100 of previously
                          'commanded motion into lfl.
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

Define 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

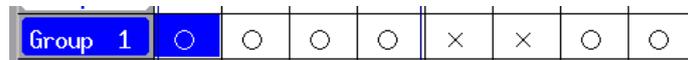
This section explains the functions relevant to extended-joints. For others, refer to the PROGRAMMER'S MANUAL I, Section 12.1 "ARRIVE."

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 PR01
```

```
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 works.

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

Specify 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

Specify 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:

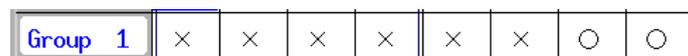
```
JACCEL 25 (JSPEED*JSPEED÷100)
JDECEL 25 (JSPEED*JSPEED÷100)
```

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

Specify 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 PRO1
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

Specify 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

Specify 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

Specify 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.**

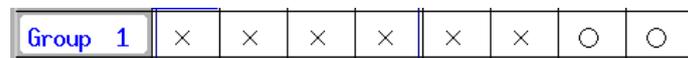
The actual deceleration is equal to (external x internal deceleration ÷ 100).

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

Get 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**

Get 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

Get 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**

Get 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

Get 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**

Get 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

Interrupt 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.

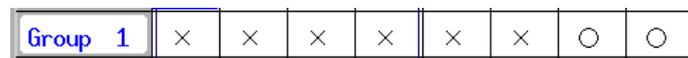
To execute the INTERRUPT command, the task must have gotten an arm group semaphore.

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	×	×	×	×	×	×	×	×

<pre> PRO1 TAKEARM 0 INTERRUPT ON MOVE P,P0      'Motion command : : INTERRUPT OFF END                     </pre>	<pre> PRO2 TAKEARM 1 INTERRUPT ON DRIVE(7,30)   'Motion command : : INTERRUPT OFF END                     </pre>
---	--

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 restore 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).

Executing this command sets a new reference position (CALSET position).

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 POSCLR.

### **mvSetPulseWidthJnt** (Library)

#### **Function**

Set 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**

Reset 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

Enter 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

Cancel the cycloid mode set for a specified extended-joint and restore 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** Release 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

Reset 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.
```

# MoveIndexHome (Library)

## Function

Rotate the index table to move a specified extended-joint to the home position.

## Syntax

```
MoveIndexHome (<JntNumber>,<motion direction>,  
<reference update>)
```

## Description

`MoveIndexHome` rotates the index table to move an extended-joint specified by `<JntNumber>` in the direction specified by `<motion direction>`.

`<JntNumber>` should be an extended-joint used in the index table.

`<motion direction>` is any of 0 to 2. Specifying 0 moves the extended-joint to the home position near to the current position; specifying 1 moves it in the negative (-) direction; specifying 2 moves it in the positive (+) direction. When the current position is 90°, specifying 0 or 1 moves the extended-joint to position 0°; specifying 2 moves it to position 360°.

If `<reference update>` is 1, `MoveIndexHome` moves the extended-joint to the home position and then executes the `POSCLR` command to set the reference position (0°) after the movement. Usually `<reference update>` should be 1.

## Related commands

`POSCLR`

## Notes

- (1) Before calling this library, you need to hold an arm group semaphore. The `TAKEARM` should execute beforehand.
- (2) This library is applicable only to joints specified for boundless rotation.
- (3) If `<reference update>` is 1, the `POSCLR` command executes after movement to the home position. Therefore, even using the Step Back function cannot return the program control back to any command preceding `MoveIndexHome`.
- (4) If `<motion direction>` is 1 or 2, the extended-joint might hardly move if the current position is near to the home position.

## Example

```
PROGRAM PR01  
TAKEARM 1 'Get Arm Group 1 including J8  
CALL MoveIndexHome (8,0,1) 'Move J8 back to the home position  
'Then execute POSCLR to set 0 degree  
DRIVEA (8,45) 'Move J8 to the 45 degree position  
DRIVEA (8,90) 'Move J8 to the 90 degree position  
:  
:  
END
```

## 3.2 Configuring the Extended-Joint Parameters

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

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

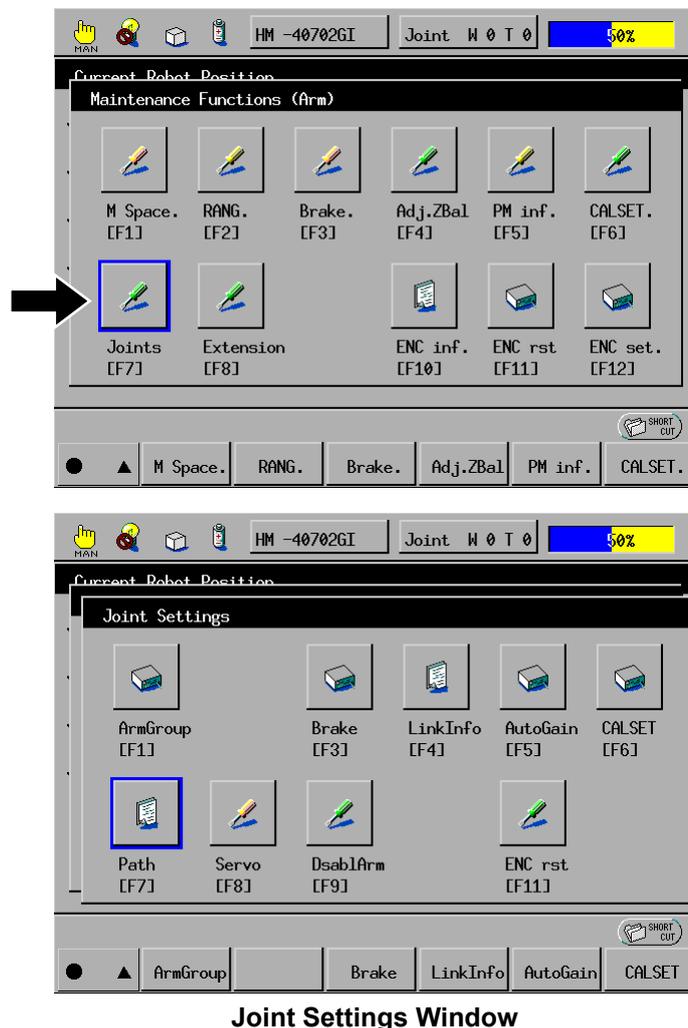
The extended-joint parameters (path configuration, servo configuration, arm group, link info, and arm disabled) can be configured also in WINCAPSIII.

Log on as a Programmer, select **Project | Joints**, and configure the related parameters. Logging on as an Operator allows you to browse the parameter values only.

### 3.2.1 Calling up "Path Configuration Parameters" and "Servo Configuration Parameters" Windows

- (1) Calling up the Joint Settings window

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

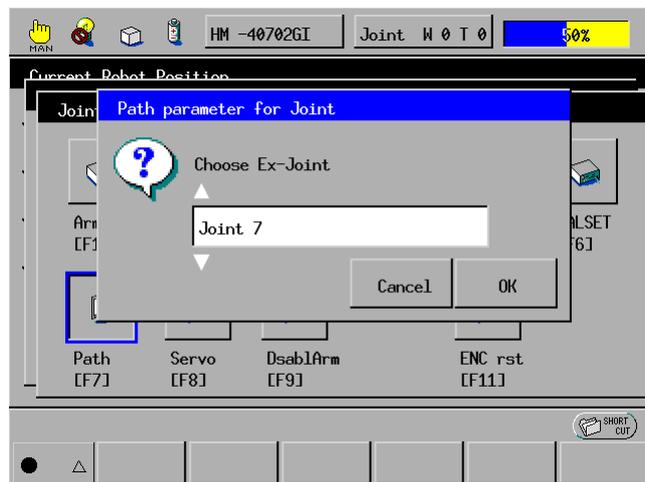


## (2) Changing the path configuration parameters



On the Joint Settings window shown above, press [F7 Path].

The path configuration window appears as shown below. Select the target joint (J7 in this example) by using the cursor keys or jog dial.



Press OK. The path configuration parameters window appears as shown below. Change the path configuration parameters and press OK. (For detailed procedure, refer to Section 3.2.2.)



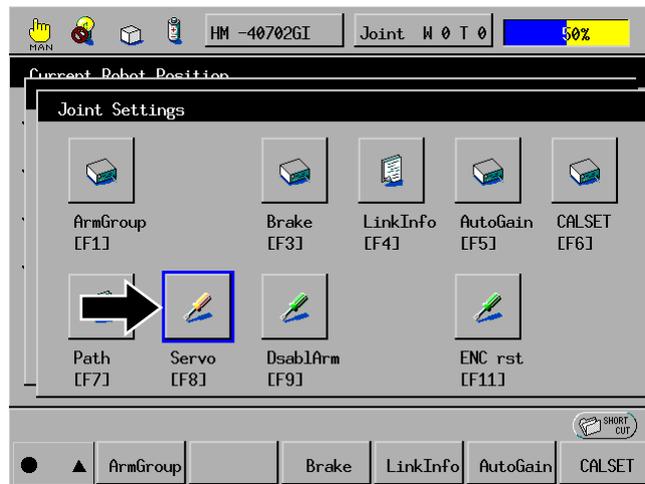
The path configuration parameters are listed in Table 3.2 below.

**NOTE:** Some parameters will take effect after the controller power is restarted.

**Table 3.2 List of Path Configuration Parameters**

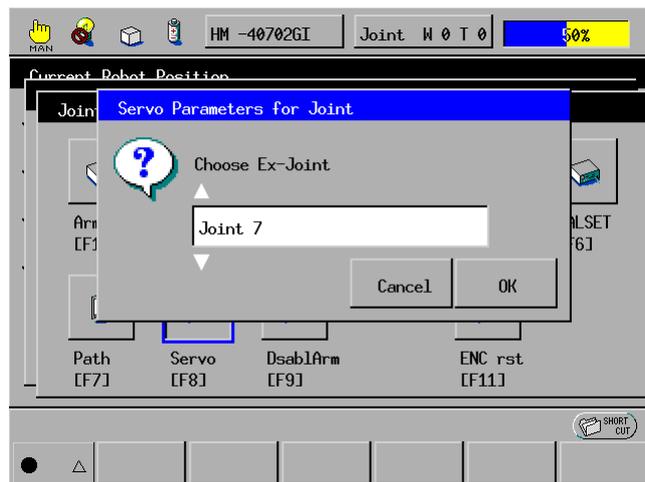
Parameter name	Entry range	Factory default	Unit	Description	Remarks	Controller restart
Boundless rotation (0: Limited, 1: Boundless)	0 or 1	0		To rotate the motor 32768 times or more in the same direction, set this parameter to 1.	Setting this parameter to "1: Boundless" requires the Motion limit detection parameter to be set to "0: Invalid."	Needed
Joint structure (0: Sliding, 1: Rotary)	0 or 1	1		If your optional mechanism to be connected to the specified motor has a sliding 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.	50	For lead: mm/r	For rotary joints, set the deceleration ratio (motor rotation/joint rotation). For sliding 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.	Setting the Boundless rotation parameter to "1: Boundless" requires this parameter to be set to "0: Invalid."	Needed
Positive motion limit (deg.) (mm)		500	For rotary joints: degrees For sliding joints: mm	Set the positive motion limit.		Not needed
Negative motion limit (deg.) (mm)		0	For rotary joints: degrees For sliding joints: mm	Set the negative motion limit.		Not needed
CALSET position		0	For rotary joints: degrees For sliding joints: mm	Set the CALSET reference position.		Not needed
Radius of gyration (mm)	0 to 100000	1000	mm	For rotary joints, set the maximum radius of rotation. For sliding joints, no setting is required.	For details, see Section 3.1.1, "[ 8 ] Setting the radius of gyration."	Needed

## (3) Changing the servo configuration parameters



On the Joint Settings window shown above, press [F8 Servo].

The servo configuration window appears as shown below. Select the target joint (J7 in this example) by using the cursor keys or jog dial.



Press OK. The servo configuration parameters window appears as shown below. Change the servo configuration parameters and press OK. (For detailed setting procedure, refer to Section 3.2.2.)



The servo configuration parameters are listed in Table 3.3 below.

**NOTE:** Some parameters will take effect after the controller power is restarted.

**Table 3.3 Servo Configuration 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 axis number	1 to 8			Displays the encoder axis 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 Section 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 Section 3.2.3.	Not needed

## Chapter 3 Extended-Joint Related Functions

Parameter name	Entry range	Factory default	Unit	Description	Remarks	Controller restart
Speed proportional gain	1 min.	200 (for 400W or less) 400 (for 750W 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 Section 3.2.3.	Not needed
Speed integral gain	0 min.	5		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 Section 3.2.3.	Not needed
Filter parameter	0 to 15	10		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  (SGMAH 50W : 1, 100W : 2, 200W : 3, 400W : 4, 750W : 5  SGMPH 100W : 12, 200W : 13, 400W : 14, 750W : 15, 1.5 kW : 16)	1 to 6			Display the connected motor capacity	No change allowed.	

## 3.2.2 Detailed Description of Extended-Joint Parameter Setting

The path configuration parameters and servo configuration 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.

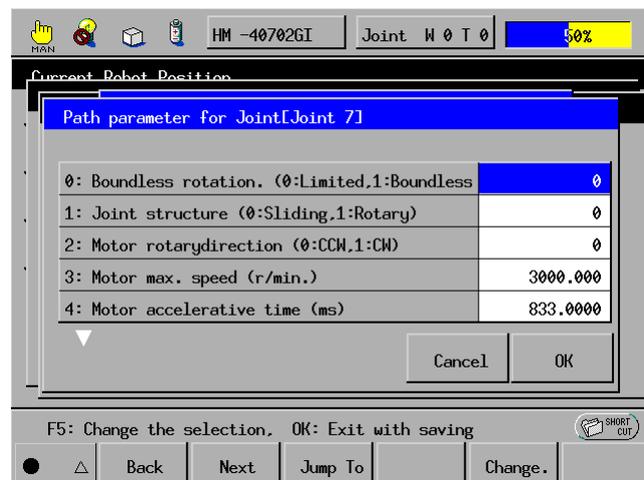
If this happens, reset the encoder (refer to Section 3.2.4.3) and restart the robot controller.

**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 configuration parameters

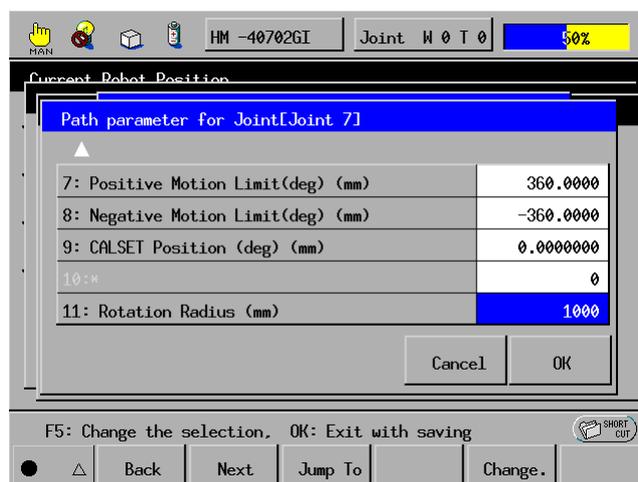
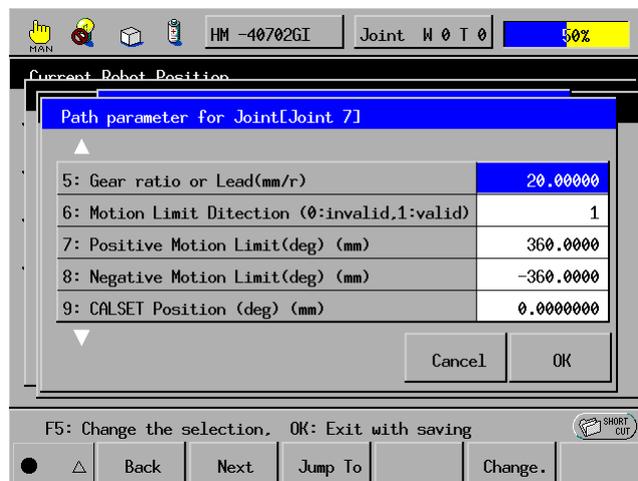
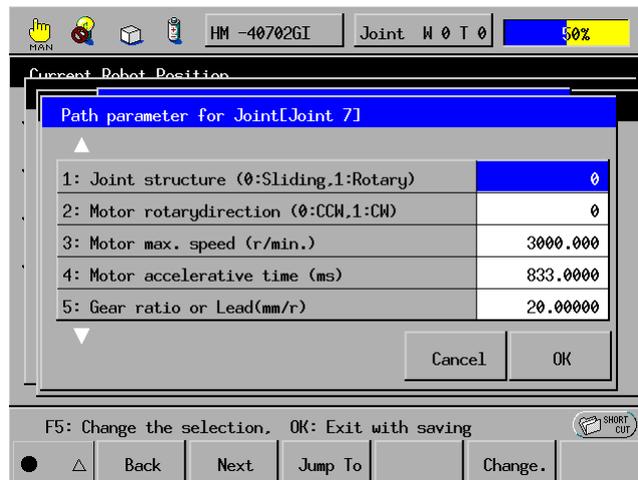
For the calling-up procedure of the path configuration parameters window, refer to Section 3.2.1.

#### (2-1) Boundless rotation



To rotate the motor 32768 times or more in the same direction, set the boundless rotation to "1: Boundless."

## (2-2) Setting the motion conditions



Set the motion-relation parameters--sliding/rotary 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.

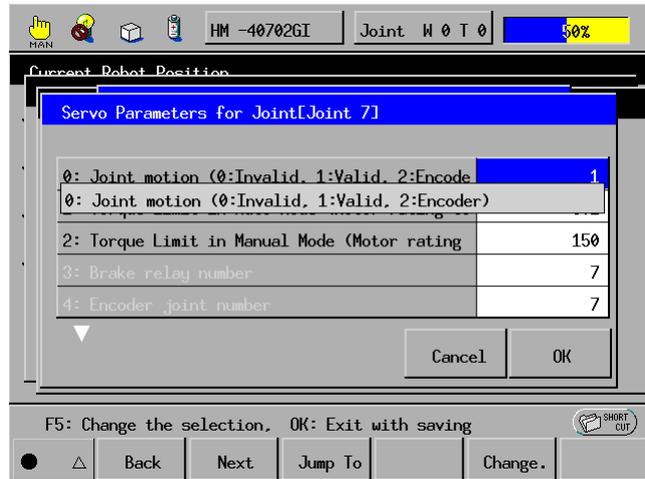
**NOTE:** When using an extended-joint as a rotary one with the boundless rotation selected, enter an integer multiple of the gear ratio. Entering a non-integer multiple will result in a positioning error in multiple rotation.

### (3) Setting the servo configuration parameters

For the calling-up procedure of the servo configuration parameters window, refer to Section 3.2.1.

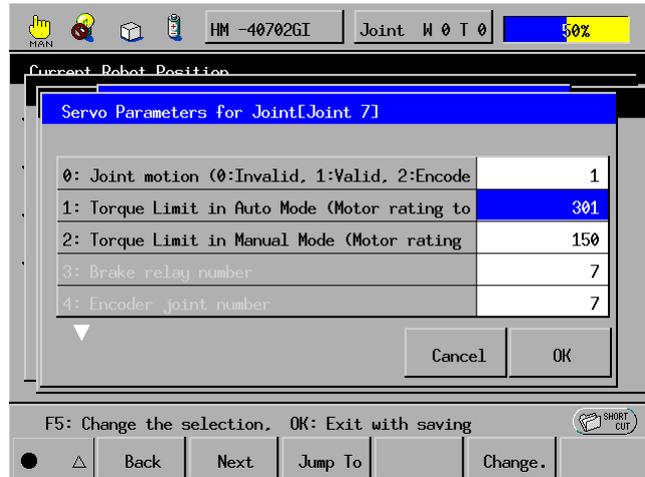
#### (3-1) Setting the joint motion

Set the joint motion to "1: Valid."



#### (3-2) Setting the torque limits

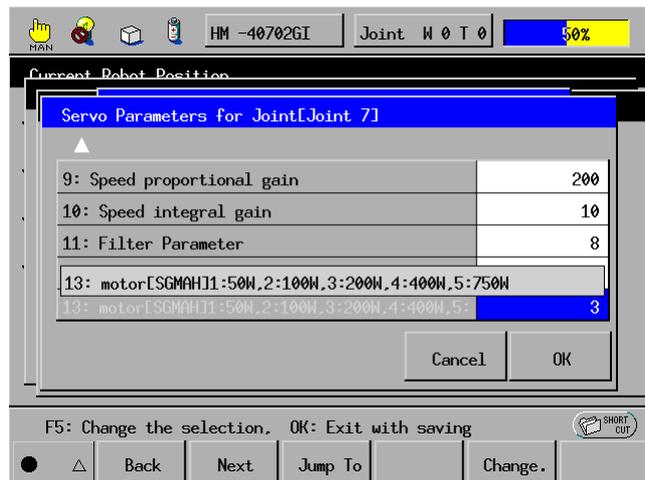
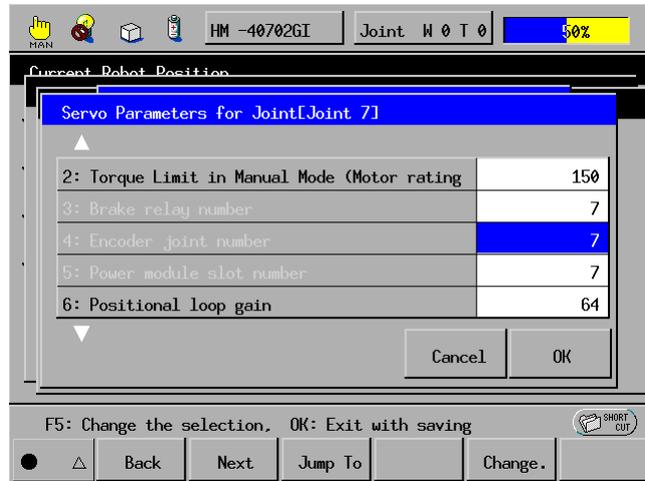
Set the torque limits in each of Auto and Manual modes.



## Chapter 3 Extended-Joint Related Functions

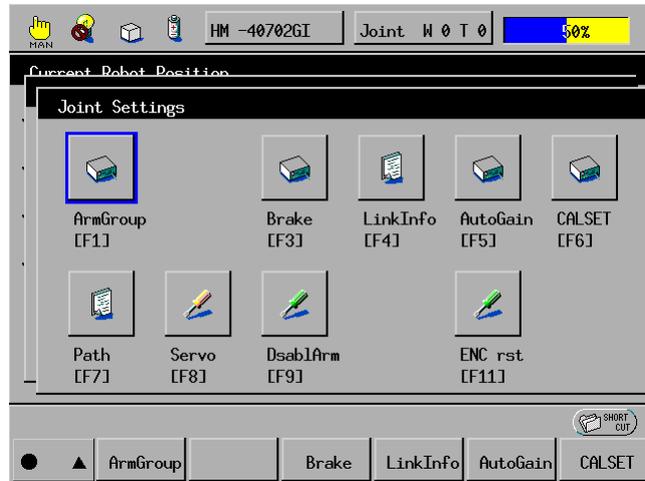
(3-3) Checking the encoder axis number, power module slot number, and motor capacity

Checks that the encoder axis 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 Section 3.1.1 "Operating Procedures of the Extended-Joint Related Functions, [4] Operating extended-joints using variables."



After completion of steps (1) to (4) above, 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 Section 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 servo configuration 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 Section 3.2.4.1 "Performing CALSET operation on extended-joints."

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

### (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 Section 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:

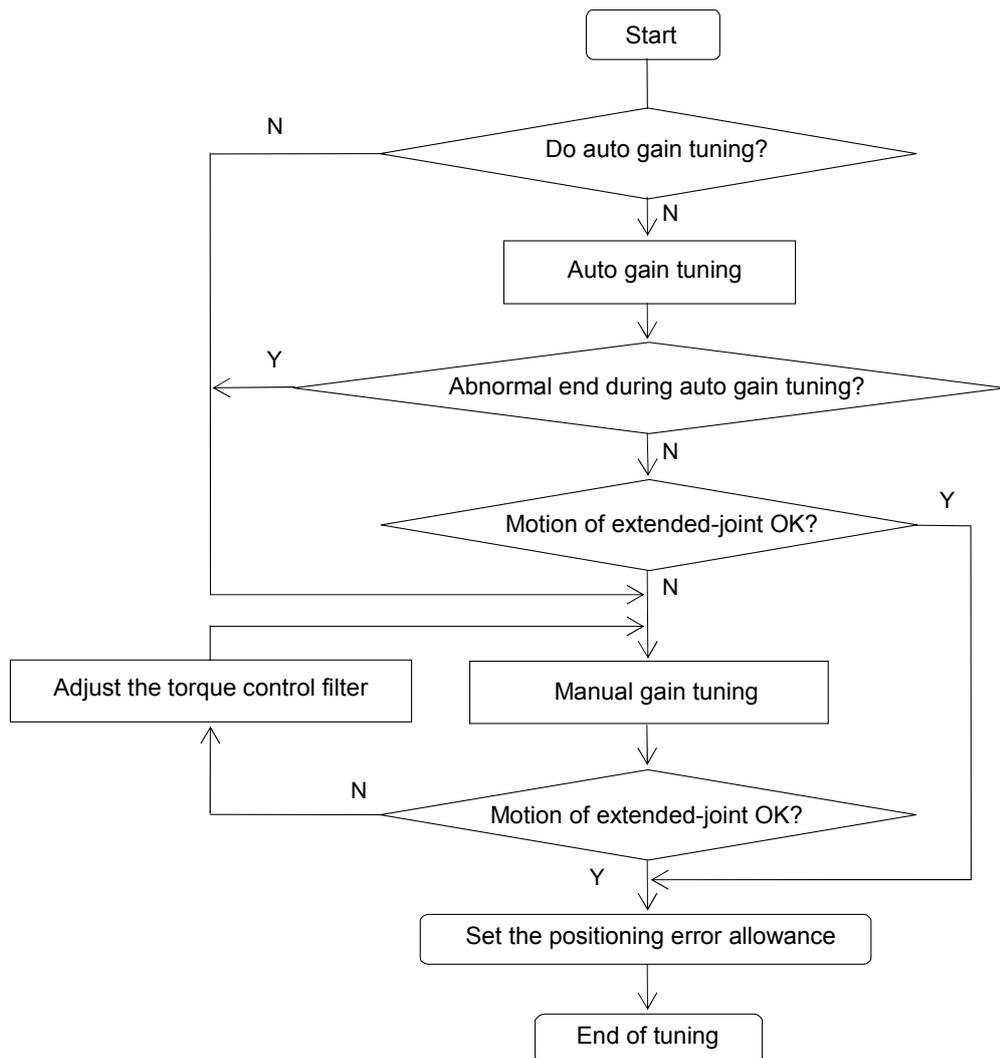
#### (1) Auto gain tuning

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

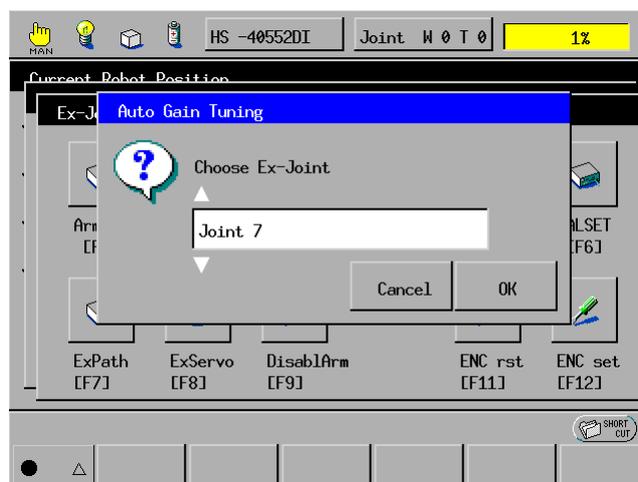
To implement auto gain tuning, your optional mechanism to be connected to the extended-joint motor should satisfy the requirements given in [ 1 ] below. Otherwise, some errors may occur and the auto gain tuning process may be interrupted. If such happens, implement manual gain tuning.

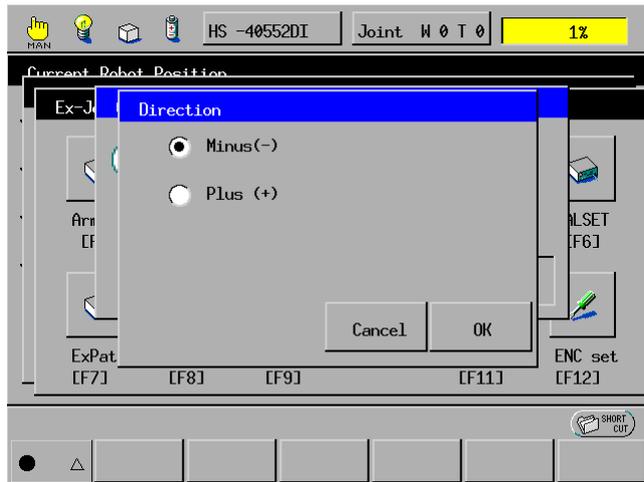
#### [ 1 ] Requirements for implementing auto gain tuning

- (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.

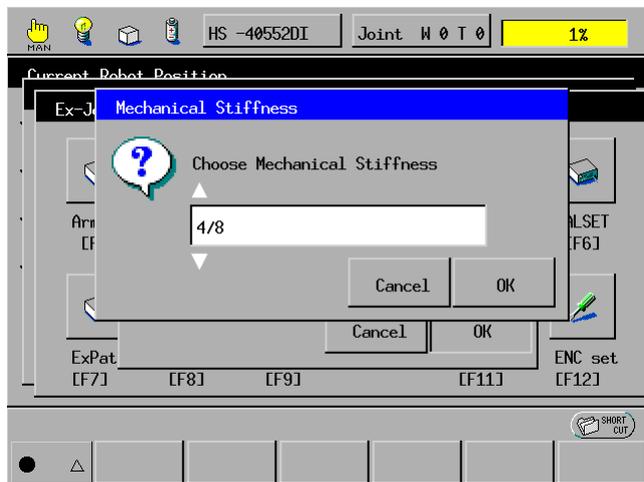
#### [ 2 ] Auto gain tuning procedure

- (1) Turn the motor power on and perform CAL  
**NOTE:** If the robot is in Auto 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 Joint Settings window.  
Access: Top Screen—[F2 Arm]—[F12 Maint.]—[F7 Joints]
- (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.



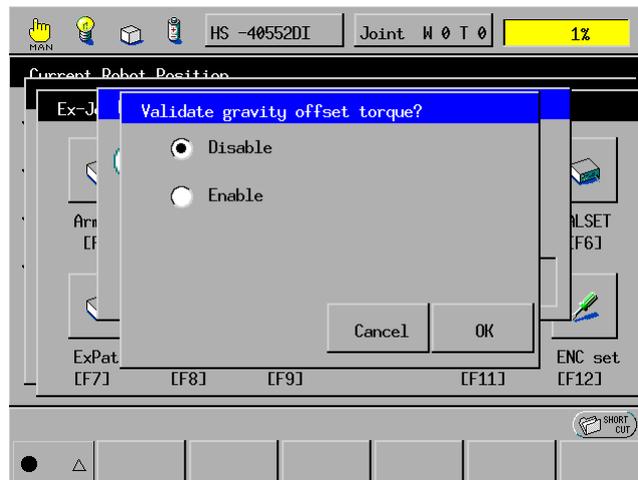


(5) Select the mechanical rigidity, referring to the rigidity reference values listed below.



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 servo configuration parameters. On the Joint Settings window, press [F8 Servo] to call up the servo configuration parameters 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 either one of the deadman switches 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 switches. Doing so will interrupt auto gain tuning.

**NOTE:** If the joint motion has been set to [2:Encoder] on the servo configuration parameters window, then the error message "Not executable" will appear during auto gain tuning.

- (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.

**Caution:**

**For a boundless rotation joint, CALSET needs to be performed after each auto gain tuning, which deletes CALSET values.**



- (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.

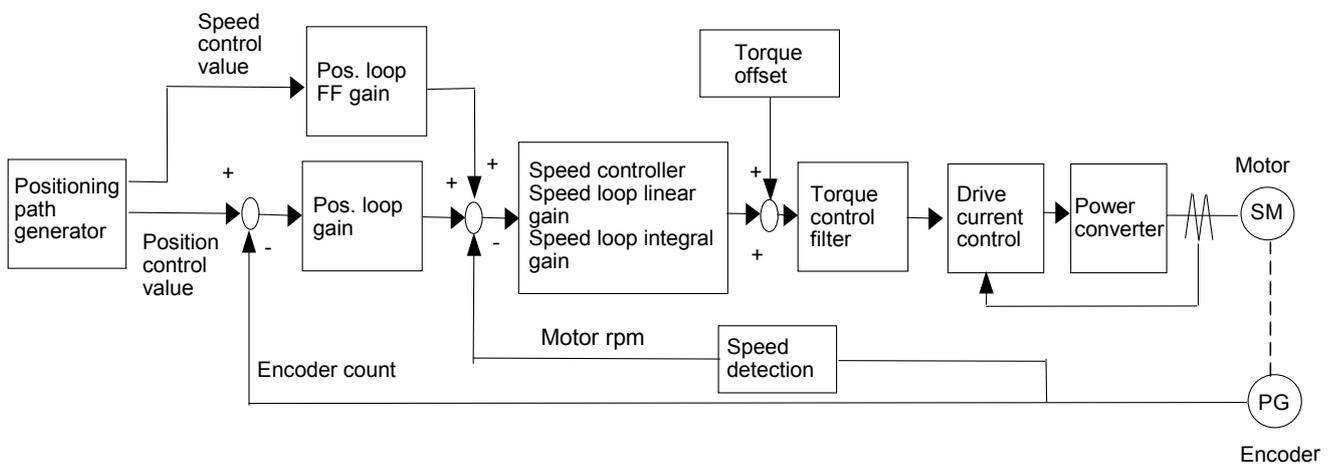
- (10) If you set higher mechanical rigidity for transmission mechanism having lower rigidity and vice versa, then an error may occur during auto gain tuning. Change the mechanical rigidity setting and retry auto gain tuning.

### 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:

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

The drive current system gain is as listed below.

Motor Model	Motor Rating	Drive Current System Gain
SGMAH-A5A1A	50 W	1.852E-05
SGMAH-01A1A	100 W	3.004E-05
SGMAH-02A1A	200 W	7.144E-05
SGMAH-04A1A	400 W	1.093E-04
SGMAH-08A1A	750 W	1.979E-04
SGMPH-01A1A	100 W	3.421E-05
SGMPH-02A1A	200 W	7.845E-05
SGMPH-04A1A	400 W	1.209E-04
SGMPH-08A1A	750 W	2.150E-04
SGMPH-15A1A	1500 W	5.819E-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:

$$\text{Integral speed loop gain time constant (ms)} = 0.25 \times [\text{Speed control linear gain}] / [\text{Speed loop integral gain}] \text{ (Formula 3.2.3-4)}$$

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 value and the band.

Filter set value	3	4	5	6	7	8	9	10	11	12	13	14	15
Band (Hz)	2450	1080	843	682	559	460	377	305	241	184	133	85	41

### (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 Section 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 Section 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 Section 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**

WINCAPSIII can graph the monitored data on a PC screen.

Adjust the scale and offset of the graph and check the graphed data.

**(5) Saving the monitored data into a file**

WINCAPSIII can save the monitored data into a CSV file.

Select Eile | Export to display the Export window. Select Log | Single Joint servo log and then press the Export button to save the data into a CSV file.

---

### [ 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 3 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.

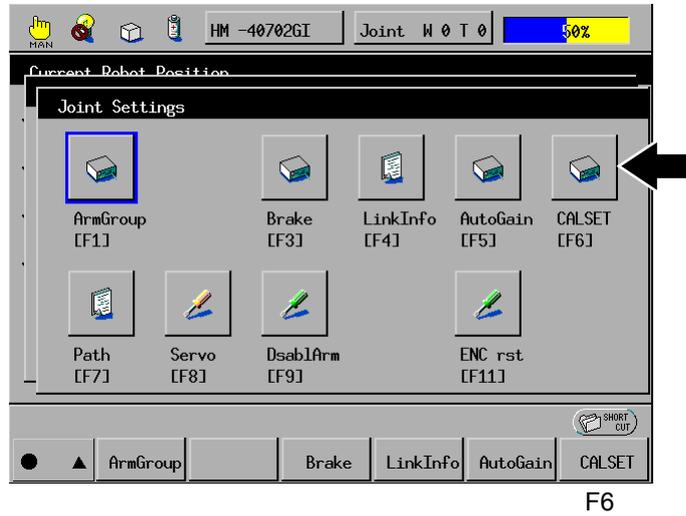
Set Value	-5	-4	-3	-2	-1	0	1	2	3	4	5
Tuning Ratio	1.5	1.4	1.3	1.2	1.1	1	0.9	0.8	0.7	0.6	0.5

### 3.2.4 Extended-Joints Exclusive Operations

#### 3.2.4.1 Performing CALSET operation on an extended-joint

(1) Calling up the Joint Settings window

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



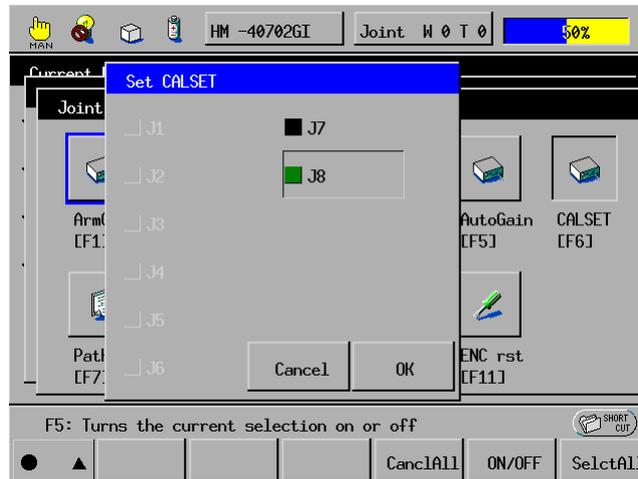
F6

(2) Calling up the Set CALSET window

In the 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

#### (1) Calling up the Joint Settings window

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



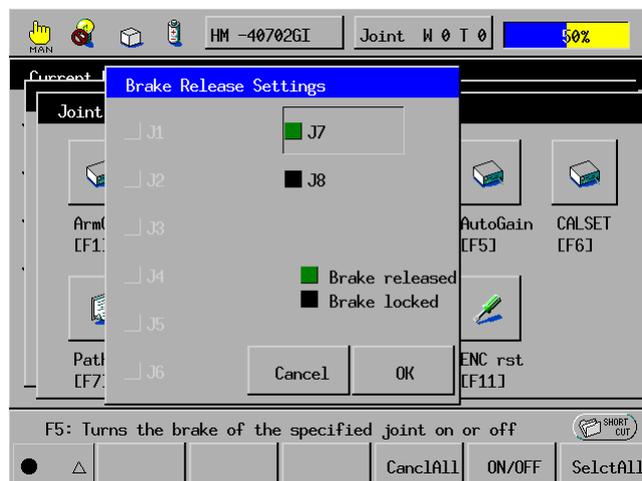
F3

#### (2) Calling up the Brake Release Settings window

In the 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.

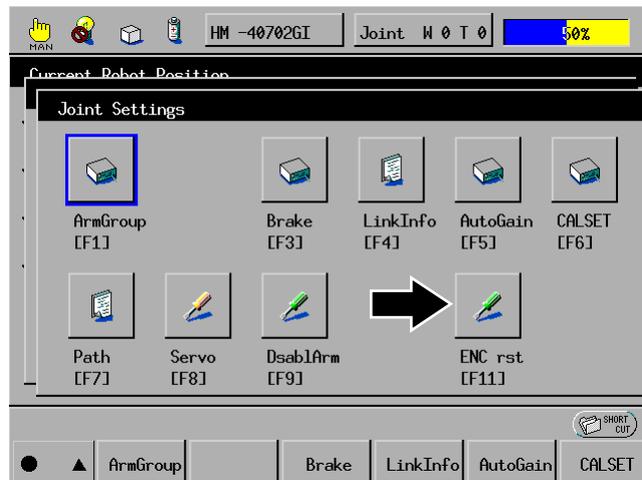


F5

### 3.2.4.3 Resetting an extended-joint encoder

#### (1) Calling up the Joint Settings window

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



F11

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

In the 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 Enabling/disabling the robot arm

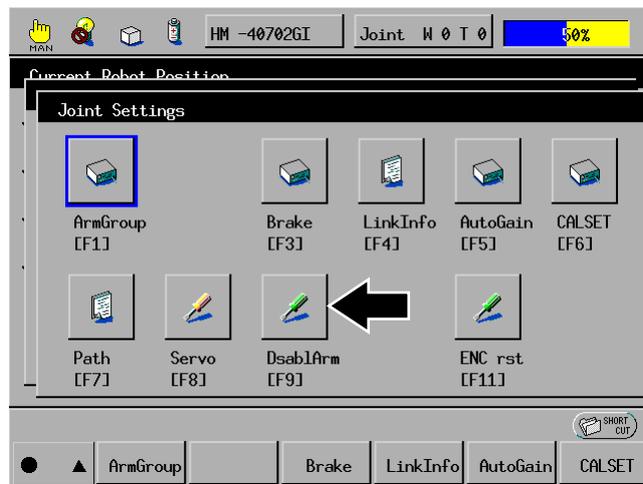
In adjustment of extended-joints, if you want to turn only extended-joint motors on for operational check without turning robot arm motor on, then disable the robot arm.

**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 Joint Settings window

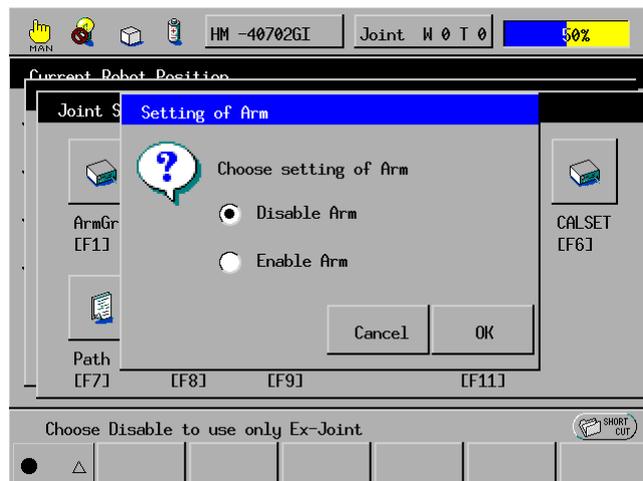
Access: Top Screen—[F2 Arm]—[F12 Maint.]—[F7 Joints]



F9

### (2) Enabling/disabling the robot arm

In the Joint Settings window, press [F9 DsablArm], 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

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

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

## SetMonitorCond

### Function

Set 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>`:

<code>&lt;MonitorData1&gt;</code> and <code>&lt;MonitorData2&gt;</code>	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

Start 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

Stop 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

Initialize 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

Abort 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
defsnrg 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

Judge 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

Get 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

Get 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

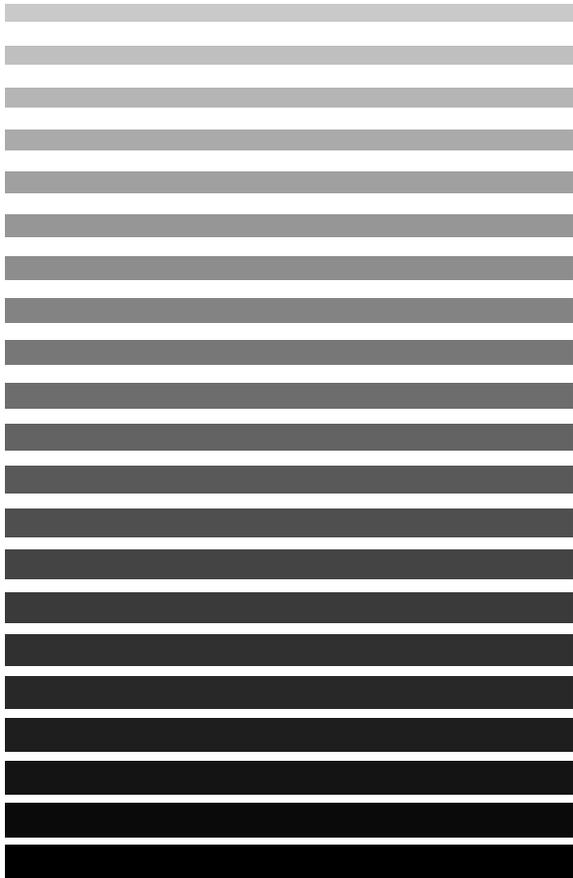
```
defsnsg vel
vel=GetJntData(1,7)      'Get J7 motor speed
```



# Chapter 4

---

## Getting Started with Extended-Joints (Rack & Pinion)



This chapter guides you through the typical start-up procedure for driving extended-joints, taking a rack & pinion as an example.



The typical start-up procedure for driving extended-joints is given below, taking a rack & pinion as an example. Follow Sections 4.1 through 4.22. For details for each step, refer to the related items given in the previous chapters or in the related manuals.

### 4.1 Checking the Cable Connection

Check that the following cables are connected to the controller.

- Power cable
- Motor & encoder cable (Robot motor and encoder cables for XYC-4G series)
- Extended-joint motor cable(s) and their conversion cable(s)
- Extended-joint encoder cable(s)

### 4.2 Powering On the Controller

Turn the controller power switch on.

**Caution: Never turn the motor power on.**

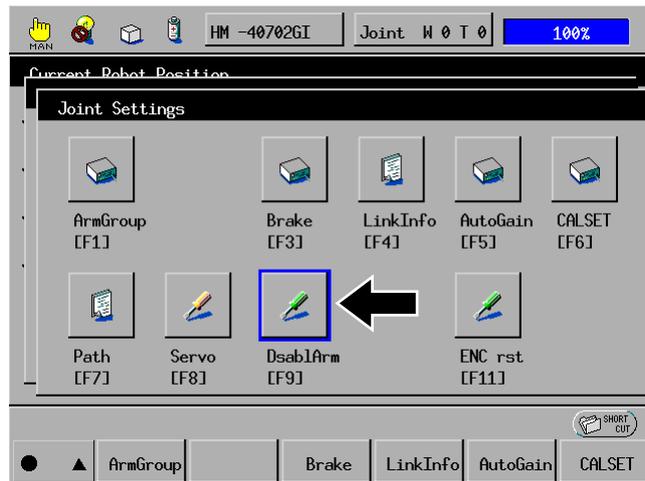
The "J7 Encoder System Down Error" message appears, so press the Cancel button on the teach pendant to clear the error.

## 4.3 Disabling the Robot Arm

For adjustment of extended-joints, you need to power on the extended-joint motor(s) for operational check without powering on the robot arm motor. Therefore, disable the robot arm using the following procedure.

- (1) Call up the Joint Settings window.

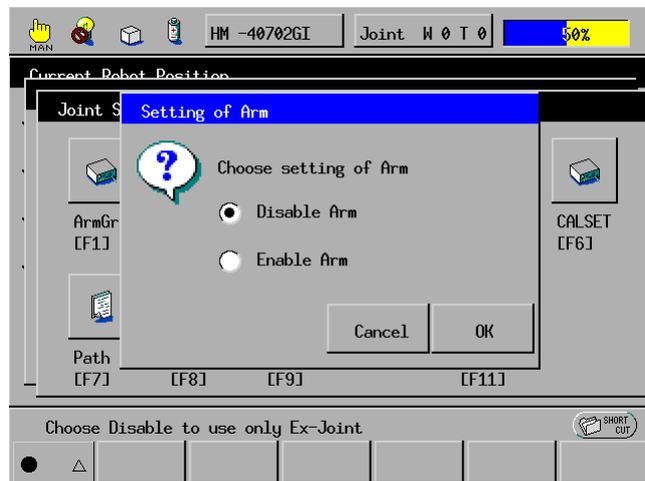
Access: Top Screen—[F2 Arm]—[F12 Maint.]—[F7 Joints]



- (2) Press [F9 DsablArm] on the above window.

The current arm settings appear as shown below.

Select "Disable Arm" and press OK.



## 4.4 Restarting the Controller

Restart the controller.

**Caution: Never turn the motor power on.**

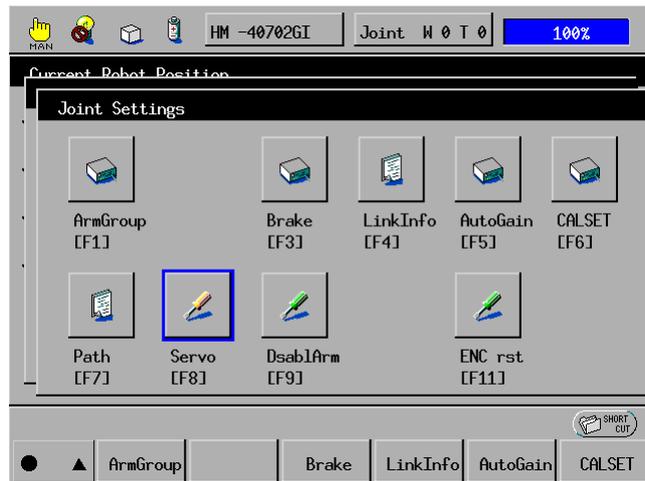
The "J7 Encoder System Down Error" message appears, so press the Cancel button on the teach pendant to clear the error.

## 4.5 Configuring the Extended-Joint Servo Parameters

Select an extended-joint to be used with the following procedure.

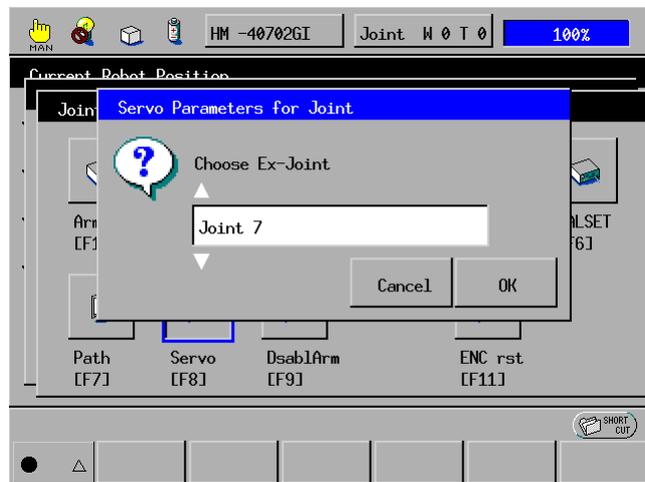
- (1) Call up the Joint Settings window.

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



- (2) Press [F8 Servo] on the above window.

The "Servo Parameters for Joint" window appear as shown below.



- (3) On the "Servo Parameters for Joint" window in step (2) above, select the target joint using the cursor keys or jog dial, and then press OK.

The servo configuration parameters window appears as shown below.



- (4) In the "Joint motion" field, select "1:Valid" and press OK.

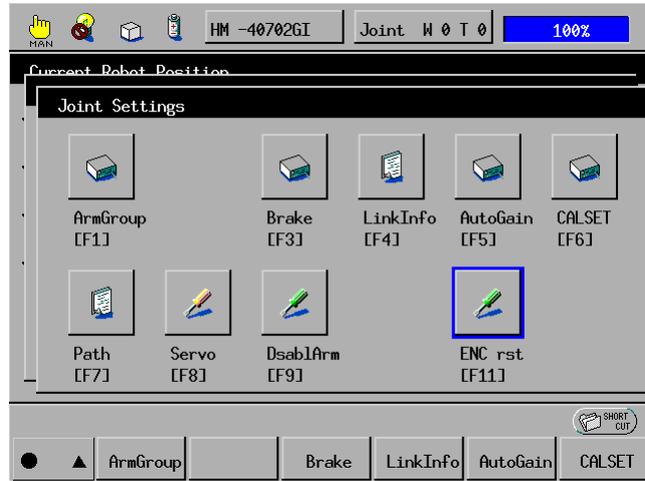
The screen returns to the Joint Settings window shown in step (1).

- (5) Repeat steps (2) through (4) for each joint to be used.

## 4.6 Resetting the Extended-Joint Encoder

- (1) Call up the Joint Settings window.

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

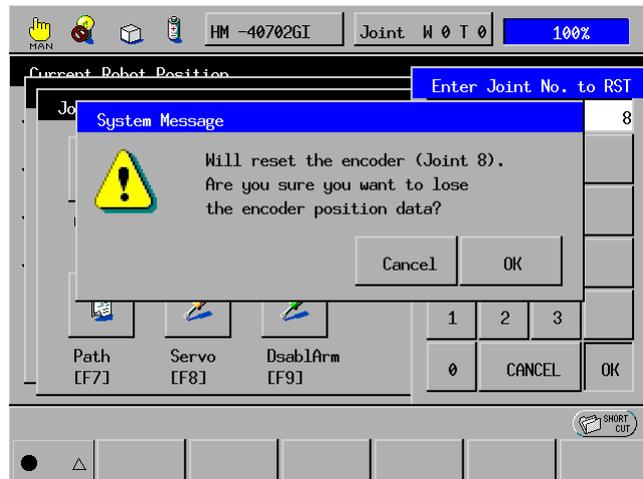


- (2) In the above window, press [F11 ENC rst], and the numeric keypad for entering the joint number appears as shown below.

Enter the joint number of an encoder to be reset and then press OK.



(3) Wait for the following confirmation message to appear and then press OK.



## 4.7 Restarting the Controller

Restart the controller.

**Caution: Never turn the motor power on.**

The "J7 Encoder System Down Error" message appears, so press the Cancel button on the teach pendant to clear the error.

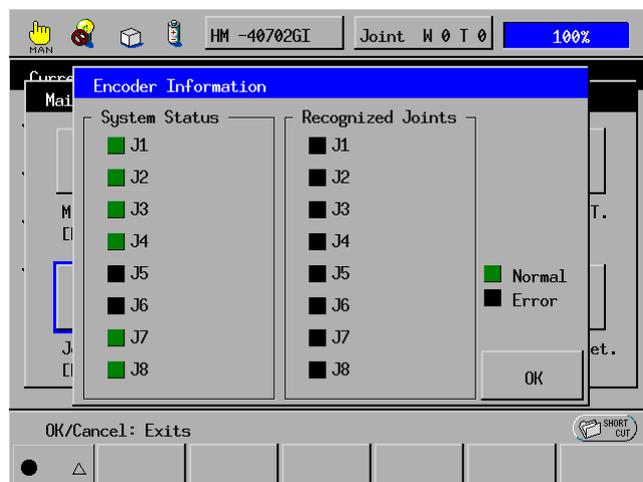
## 4.8 Checking the Encoder Info about Extended-Joints

Check that the extended-joint encoders are normal with the following procedure.

Call up the Encoder Information window.

Access: Top Screen—[F2 Arm]—[F12 Maint.]—[F10 ENCinfo.]

Check the boxes preceding extended-joints are green (normal).



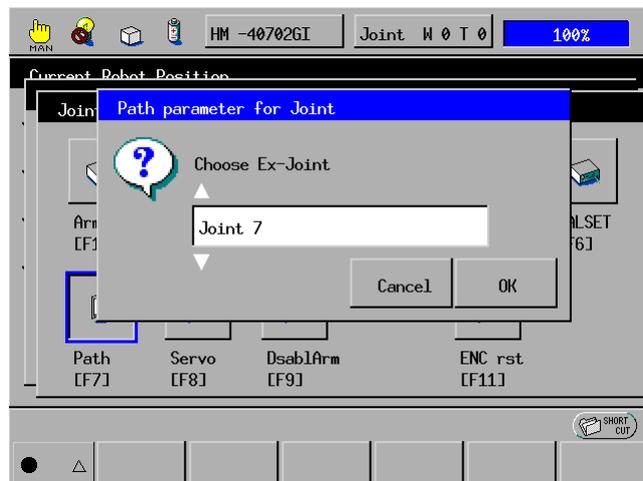
## 4.9 Configuring the Extended-Joint Parameters

Sections 4.9 and 4.14 instruct you to enter the minimum parameters required for manually moving extended-joints.

Select the desired axis and enter the parameters.

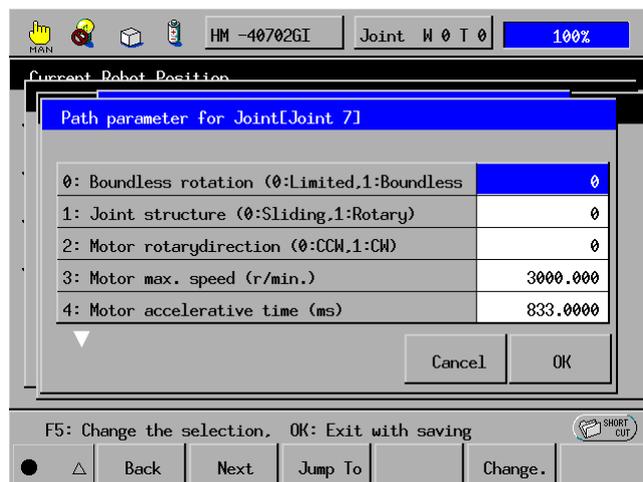
- (1) Call up the "Path parameter for Joint" window.

Top Screen—[F2 Arm]—[F12 Maint.]—[F7 Joints]—[F7 Path]



- (2) Select the target joint and press OK.

The path configuration parameters window appears as shown below.



- (3) Configure the following parameters in the above window and then press OK.

- 0: Boundless rotation (0: Limited, 1: Boundless) (Usually enter "0.")
- 1: Joint structure (0: Sliding, 1: Rotary)
- 5: Gear ratio or lead (mm/r)

---

## 4.10 Restarting the Controller

Restart the controller.

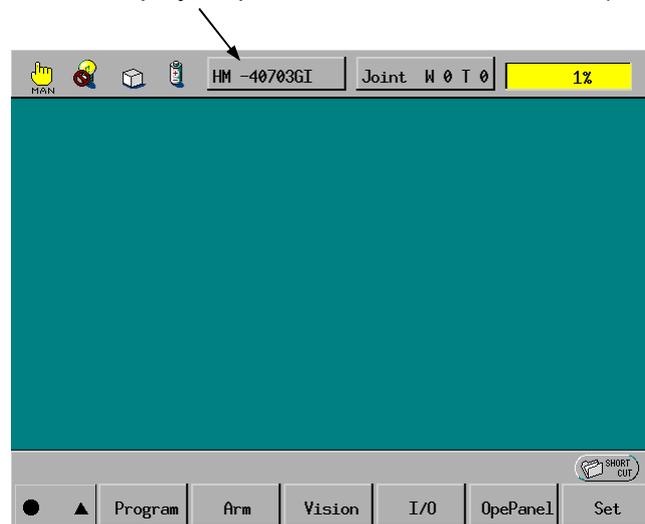
**Caution: Never turn the motor power on.**

The "J7 Encoder System Down Error" message appears, so press the Cancel button on the teach pendant to clear the error.

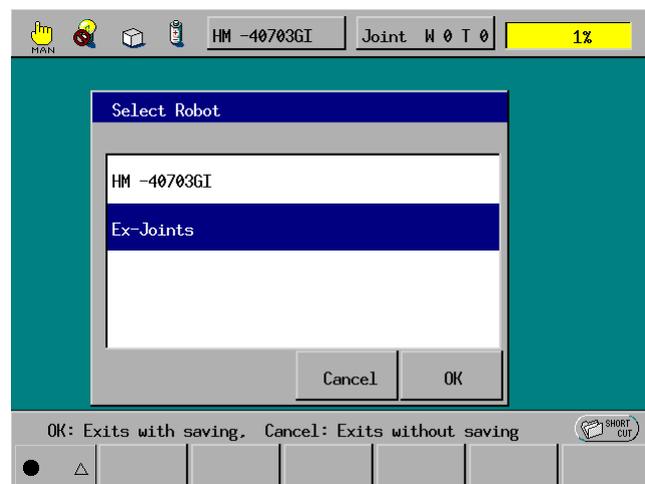
## 4.11 Checking the Encoder Feedback

Check that the controller receives a correct response from the extended-joint encoder.

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

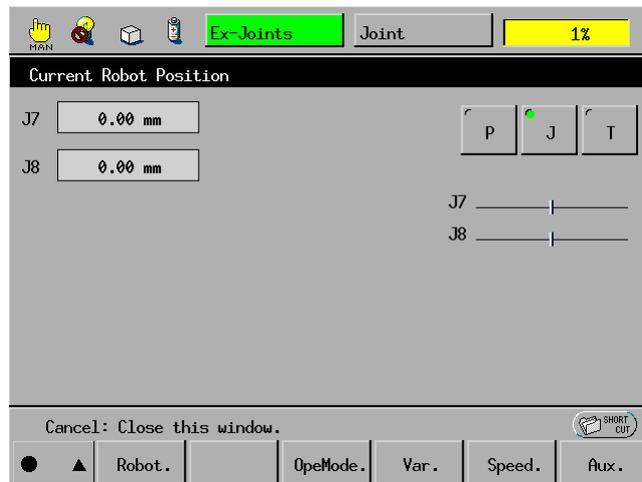


The Select Robot window appears as shown below.



- (2) Select **Ex-Joints** and then press OK.  
The display returns to the top screen.

- (3) Press [F2 Arm] to call up the Current Robot Position window as shown below.



- (4) Move the extended-joint by hand to check that the encoder follows the joint movement.

**Note:** For an extended-joint having a brake, release the brake using the following procedure beforehand.

### Releasing brakes

- 1) Access: Top Screen—[F2 Arm]—[F12 Maint.]—[F3 Brake.]
- 2) Select the joint to release the brake and then press OK.

Press OK again to lock the brake except when absolutely necessary.

## 4.12 Running the Extended-Joint Manually

- (1) Turn the motor ON.
- (2) Use the teach pendant to move the extended-joint in the positive (+) and negative (-) directions in Manual mode for checking for normal operation.

**Note:** No large movement is required in this step.

**Note:** Check that the movement direction specified with the arm traverse key matches the actual one of the extended-joint. If the direction is opposite, change the "2: Motor rotation direction" parameter" in Section 4.14.

## 4.13 Moving to the CALSET Position

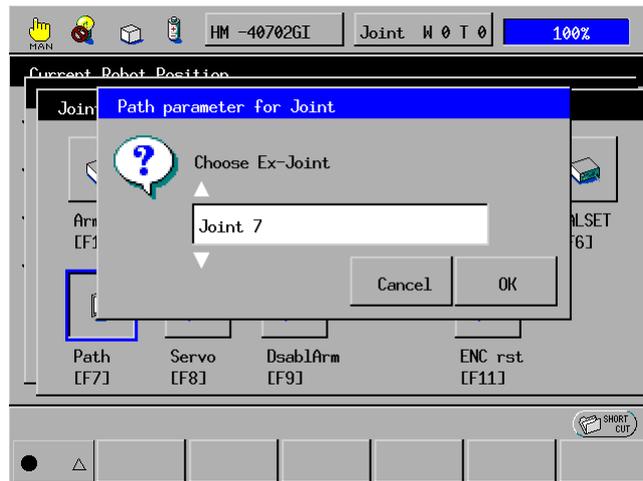
- (1) Turn the motor OFF.
- (2) Hold the joint by hand and move it to the CALSET position.

## 4.14 Configuring Other Extended-Joint Parameters

Enter other minimum required parameters not configured in Section 4.9.

- (1) Call up the "Path parameter for Joint" window.

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

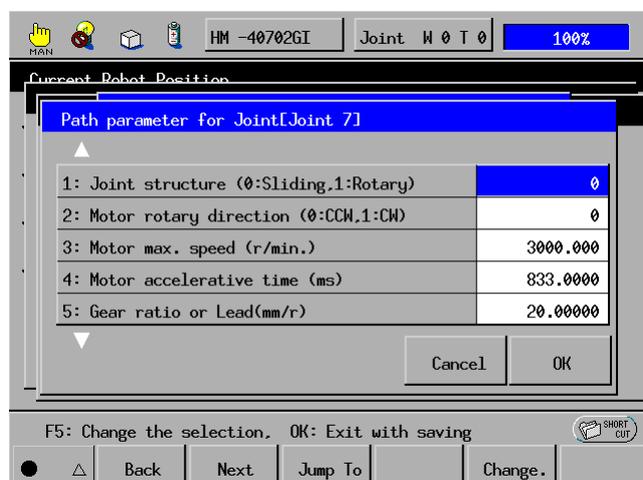


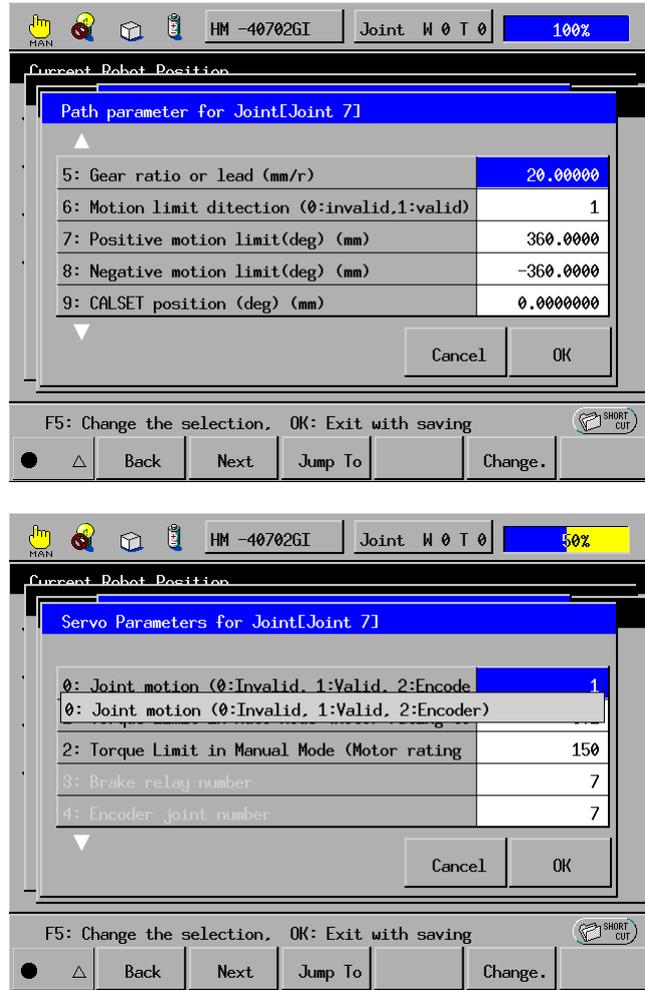
- (2) Select the target joint and press OK.

The path configuration parameters window appears as shown below and on the next page.

Configure the following parameters and then press OK.

- 2: Motor rotation direction (0: CCW, 1: CW)
- 3: Motor max. speed (r/min.)
- 4: Motor acceleration time (ms)
- 7: Positive motion limit (deg) (mm)
- 8: Negative motion limit (deg) (mm)
- 9: CALSET position (deg) (mm)
- 11: Rotation radius (mm)





### 4.15 Restarting the Controller

Restart the controller.

**Caution: Never turn the motor power on.**

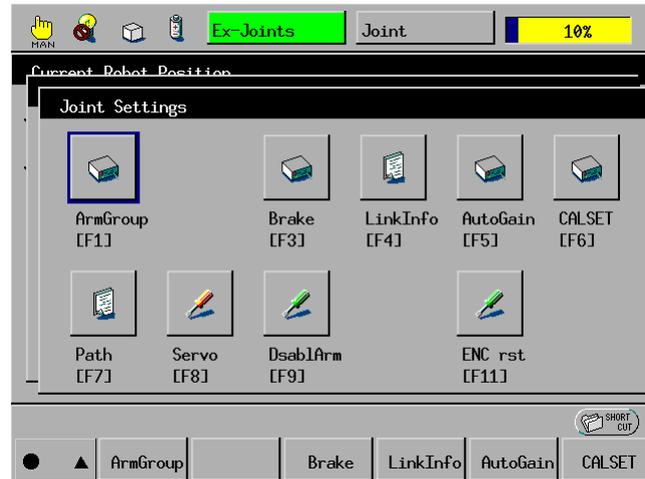
The "J7 Encoder System Down Error" message appears, so press the Cancel button on the teach pendant to clear the error.

## 4.16 Performing CALSET Operation

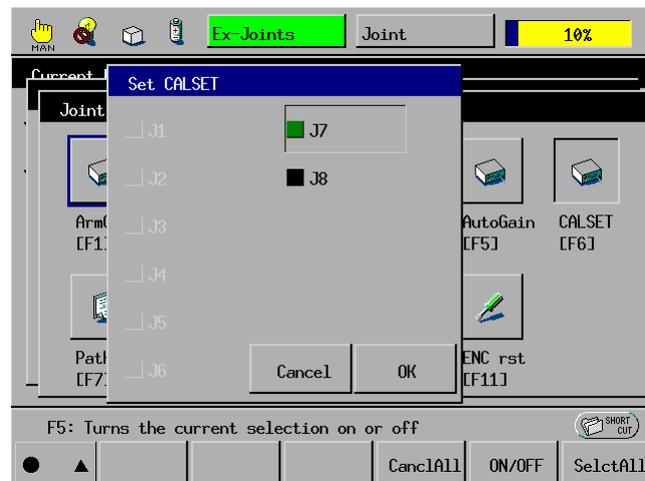
Perform CALSET operation extended-joints using the procedure given below.

- (1) Call up the Joint Settings window.

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



- (2) Press [F6 CALSET] in the above window to call up the Set CALSET window as shown below.

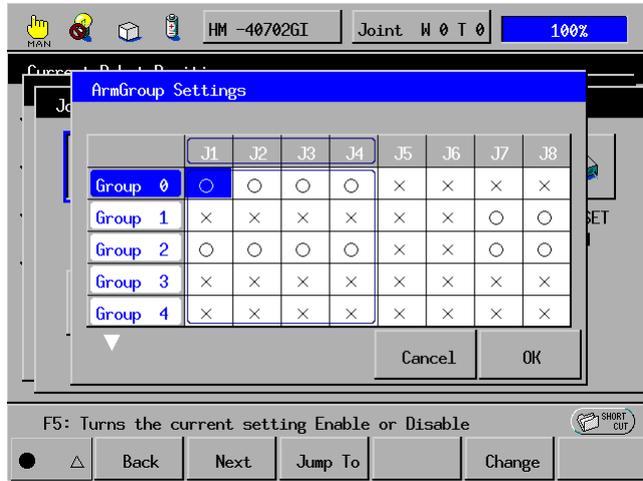


- (3) Select the extended-joint to perform CALSET and then press OK.  
The controller performs CALSET operation for the specified joint.  
In the same way, perform CALSET for other extended-joints.
- (4) After completion of CALSET operation, move the extended-joint(s) to the safe position by hand.

## 4.17 Setting up an Arm Group

(1) Call up the Arm Group Settings window.

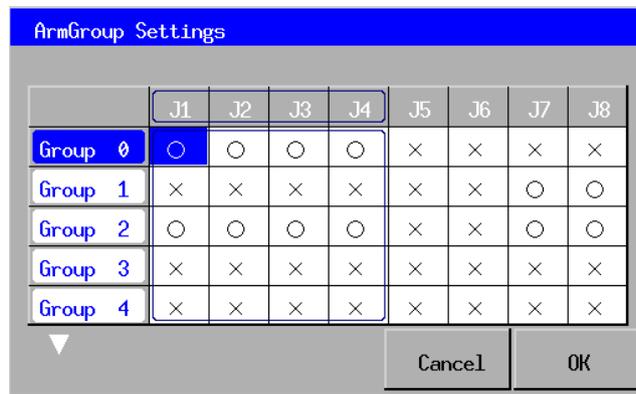
Access: Top Screen—[F2 Arm]—[F12 Maint.]—[F7 Joints]—[F1 ArmGroup]



(2) Move the cursor to a joint to be grouped and then press [F5 Change].

(3) After marking joints to be grouped or not with "O" or "X", respectively, press OK.

### ■ Arm Group Setup Example



This example sets up each group as shown below.

Group 0: Only robot joints involved.

TAKEARM 0

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

TAKEARM 1

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

TAKEARM 2

In programs, specify group numbers as shown below.

```

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

---

## 4.18 Restarting the Controller

Restart the controller.

**Caution: Never turn the motor power on.**

The "J7 Encoder System Down Error" message appears, so press the Cancel button on the teach pendant to clear the error.

## 4.19 Checking the Extended-Joint Motion

Create a sample program given below and run it to check the actual motion of extended-joints.

- (1) Create the following motion program sample.

```
PROGRAM EX_TEST1
TAKEARM 1
DRIVEA @0(7,100),S=100 'Move extended-joint
DELAY 100 'Timer
DRIVEA @0(7,6000),S=100 'Move extended-joint
DELAY 100 'Timer
GIVEARM
END
```

- (2) Set the external speed to about 10%, run the above program, and check the actual motion of extended-joints.

If any error occurs, investigate the sliding resistance of the extended-joint(s) or other probable causes.

## 4.20 Performing Auto Tuning

Perform auto tuning using the following procedure.

- (1) Turn the motor power ON and perform calibration.

**Note:** If the controller is in Auto or Teach check mode, switch it to Manual mode.

- (2) Move the extended-joint(s) to the position where rotating the extended-joints in the forward and reverse directions (in the counterclockwise and clockwise directions) each two times will not cause any problem.

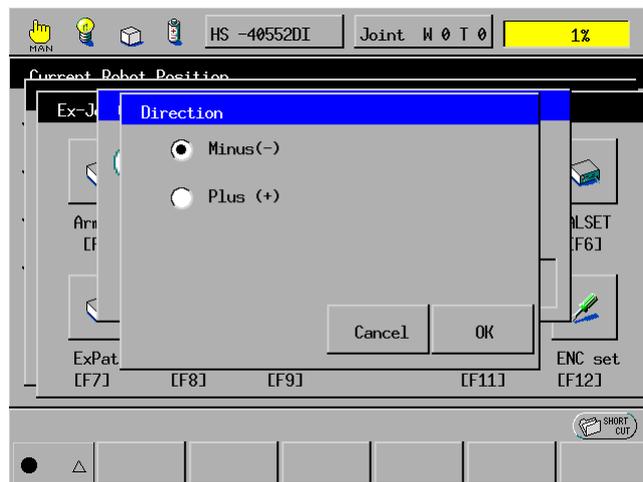
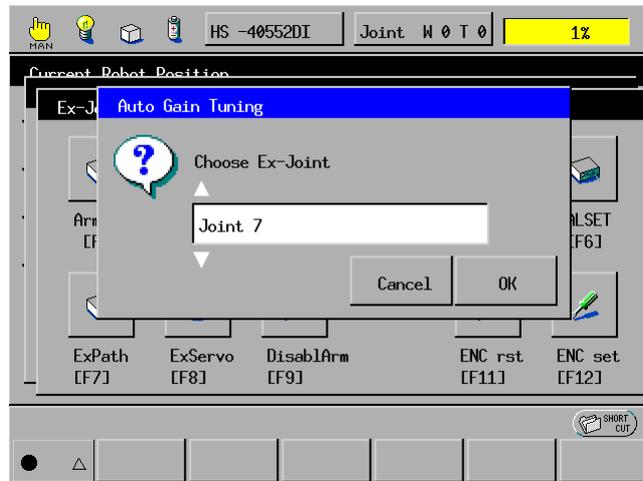
- (3) Call up the Joint Settings window on the teach pendant.

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

## Chapter 4 Getting Started with Extended-Joints (Rack & Pinion)

- (4) Press [F5 AutoGain] to call up the Auto Gain Tuning window.

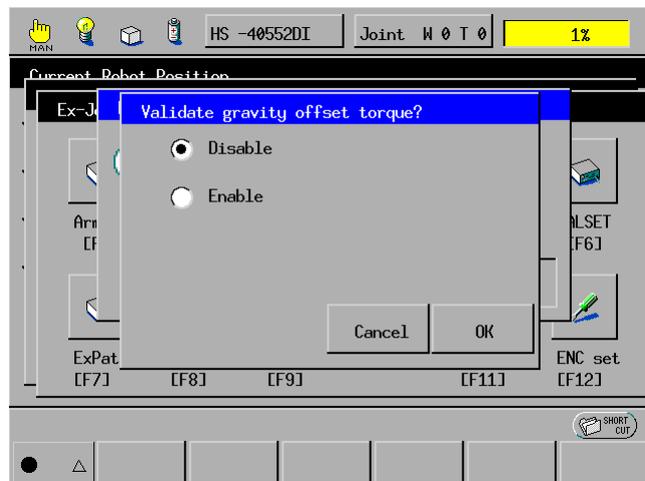
In the windows below, select the joint number to perform auto gain tuning and the movement direction.



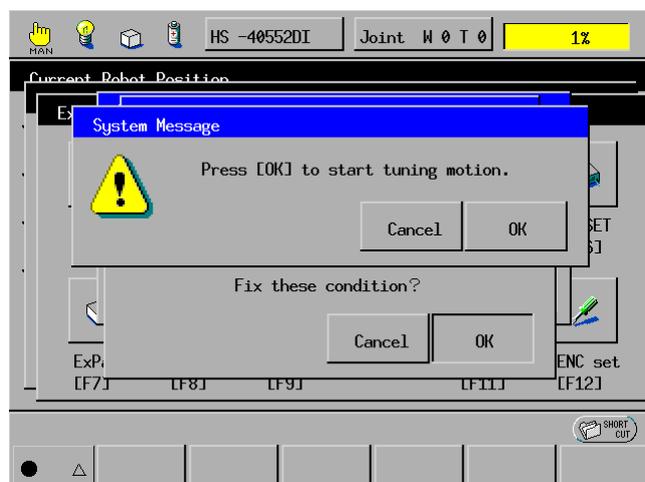
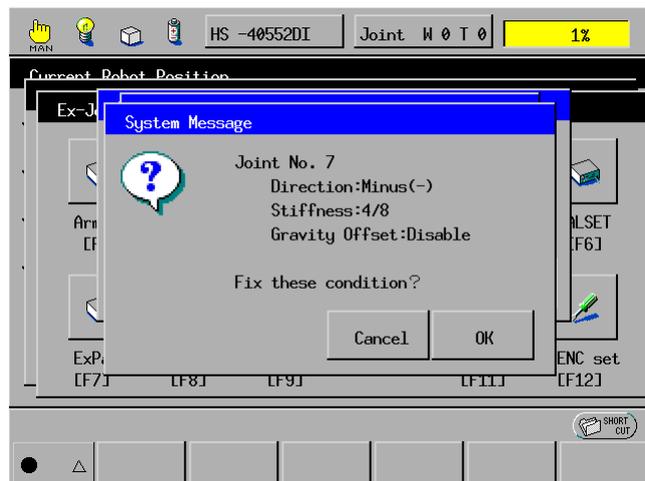
- (5) Select the machine rigidity, referring to the table below.

If the target machine is driven by:	Machine rigidity value
Ball-and-thread, direct-coupled	4 to 8
Reduction gear configured as a ball-and-thread drive	3 to 7
Timing belt	3 to 6
Gear, Rack & pinion	2 to 6
Other low rigidity machines	1 to 3

- (6) Select whether or not to enable the gravity offset torque.  
Only for the mechanism in which the extended-joint vertically undergoes gravitational load, enable this function.



- (7) While holding down either one of the deadman switches, press OK.  
Auto tuning starts.



- (8) Wait for auto tuning to finish successfully.

## 4.21 Checking the Extended-Joint Motion in Running-in

Create a sample program for running-in and run it to check the actual motion of extended-joints while increasing the external speed.

- (1) Create the following program sample.

Read in a CALL program from the library.

```
'!TITLE "Reading in extended-joint torque and absolute current values"
PROGRAM EX_TEST2
TAKEARM 1

CALL SetMonitorCond(7,1,4,4) 'Set monitoring conditions for single-
                                'joint servo data monitor
CALL StartSrvMonitor          'Start monitoring the single-joint servo
                                'data
DRIVEA @0(7,100),S=100        'Move extended-joint
DELAY 100                     'Timer
DRIVEA @0(7,6000),S=100       'Move extended-joint
DELAY 100                     'Timer
CALL StopSrvMonitor           'End monitoring
END
```

- (2) First, set the external speed to about 10%, run the above program, and check the actual motion of extended-joints.

Gradually increase the external speed up to 100%.

If an acceleration error or any other error occurs during acceleration, decrease the "3: Motor max. speed (r/min)" out of the path configuration parameters by 500 rpm and go back to Section 4.20 "Performing Auto Tuning."

- (3) If no error occurs when the extended-joint is being driven with the external speed 100%, import the data to WINCAPSIII and display the graph from the single-joint servo data log.



---

(4) Confirm the auto tuning state on the graph in step (3).

Note: The WINCAPSIII export function allows you to export the log data to the robot controller and save it in CSV format.

## **4.22 Enabling the Robot Arm**

Enable the robot arm that has been disabled in Section 4.3.

# **DENSO ROBOT**

V\* / H\*-G / XYC-4G/XR-G SERIES

## **Extended-Joints Support**

---

### **SUPPLEMENT**

First Edition      July 2007  
Seventh Edition    April 2011  
Eighth Edition     October 2011

DENSO WAVE INCORPORATED

10N\*\*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.

