ERC Robot Controller

ERC Advanced Programming Training Manual

NOT FOR RESALE

June 13, 1997

MOTOMAN
805 Liberty Lane
West Carrollton, OH 45449
TEL: (937) 847-6200 FAX: (937) 847-6277
24-HOUR SERVICE HOTLINE: (937) 847-3200

PREFACE

PURPOSE OF THIS MANUAL

This ERC Advanced Training Manual is to be used in conjunction with Motoman's ERC Advanced Programming Class and is not intended for use as a training tool in itself. This manual is not for resale and will not be sold separately.

WHO SHOULD USE THIS MANUAL

This manual should be used by attendees of Motoman's ERC Advanced Class. Do not use this manual as a reference tool unless you have attended the ERC Advanced Class and have received certification through Motoman, Inc.

HOW TO USE THIS MANUAL

This manual has been written in accordance with Motoman's ERC Advanced Classes daily course structure. It is designed to assist students in understanding ERC functions for a Motoman robot. Use this manual as a step-by-step guide through the ERC Advanced Classes.

DISCLAIMER

Be aware that keystrokes described in this manual may vary with software versions, and purchased software options.

TABLE OF CONTENTS

PART 1 - STANDARD SOFTWARE

Section Page

Section Page

Section Page

MOTOMAN, INC

ERC ADVANCED PROGRAMMING TRAINING MANUAL

PART 1:
STANDARD SOFTWARE

1.0 OP2 KEY FEATURES

The ERC OP2 key is only functional when the operator's panel is in **TEACH** mode.

An 8-digit USER ID password must be entered to access the OP2 menu for authorized maintenance personnel only.

1.1 CHANGING THE REGISTERED USER ID NUMBER

User functions are protected by an 8-digit User ID number. This permits only authorized personnel who know the User ID number to modify settings. The preset factory number is 00000000 (8 zeros).

To change the registered user ID number, complete the following steps in TEACH mode:

1. Press OP2.

   The user ID display appears

NOTE: *If parameters, home calibration position, or concurrent I/O ladder program are modified after delivery, "CHANGE" is displayed on the lines where the changes have been made. The date of the change is also displayed.*

2. Enter the current 8-digit ID number using the number keys on the operator's panel.
3. Press ENTER.

   The soft key labels for user functions appear at the bottom of the CRT screen to call up corresponding displays.
4. Press the ARROW RIGHT key two times.
5. Press **U.ID chg**(F5).

   The user ID registration display appears.
6. Enter the desired 8-digit number using the number keys on the operator's panel.
7. Press ENTER.

The new number will be registered as the new ID number. Always keep a record of the user ID number after changing it.

1.2 SETTING TIME/DATE

The date and time settings on the ERC are in numeric format. The time displayed is in military (24-hour clock) time, and the date format is Year-Month-Day.

To set the time or date, complete the following steps in TEACH mode:

1. Press OP2.
2. Enter the 8-digit ID number using the number keys on the operator's panel.
3. Press ENTER.
4. Press **Clock set** (F4).

   The clock set display appears.
5. Press **Date set** (F4) or **Time set** (F5).
6. Enter the new date or time using the number keys on the operator's panel. For example, to enter the date as June 13, 1997, press "1997.6.13" and ENTER. To set the time as 2:56 p.m., press "14.56" and ENTER.

NOTE: An error will occur if all figures are not entered or if erroneous dates or times are entered.

7. Press ENTER.

1.3 ACCESSING/CHECKING SYSTEM INFORMATION

The Robot Axis Configuration and Version No. Display contains information about the manipulator type, the axis configuration, and information regarding the CPU version.

To view this information, proceed as follows in TEACH mode:

1. Press OP2.
2. Enter the 8-digit ID number using the number keys on the operator's panel
3. Press ENTER.
4. Press the ARROW RIGHT key two times.
5. Press **VERSION** (F2).

1.4 REGISTERING I/O NAMES

Each Universal Input and Universal Output should be given a name description of up to 16 characters for ease of identification when the Diagnosis screen is displayed.

To register I/O names, complete the following steps in TEACH mode:

1. Press OP2.
2. Enter the 8-digit ID number with the number keys on the operator's panel.
3. Press ENTER.
4. Press the ARROW RIGHT key.
5. Press **IN name** (F4) for User Input Name Display, or **OUT name** (F5) for User Output Name Display. The USER input name display or USER output name display appears.
6. **Page↓** (F1) or **Page↑** (F2) if necessary, then move the cursor to the desired number. Up to 96 input names and 64 output names can be registered.
7. Press **Rename** (F5).
8. Press **ABC.A B C** (F1). Enter the input/output name using the alphabet, symbol, and/or number characters. An I/O name must be 16 characters or less.
9. Press **EXIT** (F5).
10. Press ENTER. The I/O name is registered.

NOTE: To clear I/O name, simply press CANCEL then ENTER.

1.5 SETTING ERC PLAY SPEED KEY

The PLAY SPD key on the ERC teach pendant can be set by SC parameters to 8 preset, frequently used Linear speeds in units of 0.01 mm/sec. The Linear values are set in either cm/min or mm/sec. To change the values from cm/min to mm/sec, or from mm/sec to cm/min, see Section 1.5.2.

The SC Parameters SC060-SC067 correspond to the PLAY SPEED key's choices numbered 1-8.

To change any or all of the 8 preset choices, complete the following steps in TEACH mode:

1. Press OP2.
2. Enter the 8-digit ID number using the number keys on the operator's panel.
3. Press ENTER.
4. Press **Parameter** (F1).
5. Press **SC** (F1).
6. **Page**↓ (F1) or **Page**↑ (F2) if necessary, then cursor to the desired parameter [SC060-SC067].
7. Press **Data chg** (F3).
8. Press CANCEL.
9. Enter the desired setting in units of 0.01mm/sec.
10. Press ENTER.
11. Turn the controller OFF, wait 5 seconds, then power ON.

To change from cm/min to mm/sec the appropriate SC parameter must be set.

To set this parameter, complete the following steps in TEACH mode:

1. Press OP2.
2. Enter the 8-digit ID number with the number keys on the operator's panel.
3. Press ENTER.
4. Press **Parameter** (F1).
5. Press **SC** (F1).
6. **Page**↓ (F1) or **Page**↑ (F2) if necessary, then move cursor to SC203.
7. Press **Data chg** (F3).
8. Press CANCEL.
9. Enter 1 for cm/min, or 0 for mm/sec.
10. Press ENTER.
11. Turn the controller OFF, wait 5 seconds, then power ON.

1.6 SOFT LIMITS

Soft limits restrict the work envelope of the manipulator. Software monitors the manipulator before it reaches the hard limit switch. The manipulator working range is restricted with the following three soft limits:

- Individual maximum working range for individual axes

  [MAX: S L U R B T = SC000-SC005; MIN: S L U R B T = SC006-SC011]

  (Unit: Pulse count)
- Mechanical interference area between axes

  [Motor Limit: +B & +T = SC012 & SC013; -B & -T = SC014 & SC015]

  (Unit: Pulse count)

  [Intrf. Limit L & U axis: MAX Angle = SC016; MIN Angle = SC017]

  (Unit: 0.1 degree)
- Allowable cubic working range set parallel to robot coordinate axis

  [MAX: X Y Z = SC018-SC020; MIN: X Y Z = SC021-SC023]

  (Unit: 0.1 mm, set with absolute value on base coordinate to include travel axis)

These soft limits are continuously monitored by the system, and the manipulator stops automatically if it reaches the maximum working range of any soft limit.

To determine if the SLURBT soft limits need to be changed to values other than the default settings, begin with the robot at its usual "safe" start/end position. Using JOINT coordinates, jog each axis to the desired positive operating limit, then the negative operating limit, and write down the pulse position data to be entered into the SC parameter files. The S+ and S- pulse values should be set just within the hard-stop limit switch boundaries established by the placement of the bolts. The L+ and U+/U- should be given as much "freedom" as possible; the L- soft limit should not be set below the rearward horizontal position for any Motoman robot (0 on a K-Series robot). It is also important for the positive and negative soft limit settings for R, B, and T axes to be set to values appropriate for the end-effector(s) to be used.

To display the robot's current position in order to write down the value settings at each +/- operating limit, complete the following steps:

1. Press DISP.
2. Press **Position** (F2).
3. Press **Pulse** (F1).

To access the SC parameters to set the soft limits, complete the following steps in TEACH mode:

1. Press OP2.
2. Enter the 8-digit ID number with the number keys on the operator's panel.
3. Press ENTER.
4. Press **Parameter** (F1).
5. Press **SC** (F1).
6. **Page**↓ (F1) or **Page**↑ (F2) if necessary, then cursor to the desired parameter.
7. Press **Data chg** (F3).
8. Press CANCEL.
9. Enter pulse counts for the axis.
10. Press ENTER.
11. Continue by moving the cursor to each axis parameter to be modified and repeat Steps 7 through 10.
12. Turn the controller OFF, wait 5 seconds, then power ON.

In order to operate the robot beyond the soft limits for maintenance purposes, etc., the soft limits can temporarily be released.

Use the following procedure to temporarily release the soft limits:

1. Press TEACH.
2. Press the ARROW RIGHT key.
3. Press **Limit open** (F2).

   The message line will read ==>"Soft limits has been released."

Choose <u>one</u> of the following three ways to cancel the soft limit release:

- Press PLAY

  or

- Turn the main power switch OFF on the ERC, wait 5 seconds, then turn it ON again,

  or

- Complete the following steps:

1. Press TEACH.
2. Press the ARROW RIGHT key.
3. Press **Resume** (F4).

1.7 HOME POSITION CALIBRATION

Home position calibration is an operation in which the home position and encoder zero position coincide. Although this operation is performed at the factory prior to shipment, the following are times when home position calibration must be performed again:

- Changing the combination of the manipulator and controller
- After replacing a motor or encoder
- After the clearing of stored memory (cause by the replacement of MM14 board, a weak battery, etc.)
- After a home position deviation (caused by crashing the manipulator into a work piece, etc.)

Home position is pulse position "0" for each axis.

The deviation values between home position and control reference position are set to parameters. The deviation values are specified by an angle in units of 1/1000° and vary for different manipulator types.

There are two ways to calibrate home position:

- Batch registration of all axes (all the axes can be moved at the same time)
- Registering individual axis (each axis is moved individually)

NOTE: Teaching and playback are not possible before home position calibration is complete.

Batch registration is performed when the robot and controller are initially set-up together, or perhaps when major changes have been made. Usually, individual axis registration is used after motor replacement, etc. (see section 1.7.2).

To batch register <u>all</u> axes, complete the following steps in TEACH mode:

1. Press OP2.
2. Enter the 8-digit ID number using the number keys on the operator's panel.
3. Press **Calibrate** (F3). The home position display appears.
4. Turn Servo Power ON.
5. Press **Setting** (F5).
6. Press **All axes** (F5).
7. Press **Register** (F5) to "capture" the displayed value of all axes as the home position.

   NOTE: Pressing **Resume** *(F2) at Step 7 stops the operation.*
8. Turn the controller OFF, wait 5 seconds, then power ON.

To register individual axis, complete the following steps in TEACH mode:

1. Press OP2.
2. Enter the 8-digit ID number using the number keys on the operator's panel, then press ENTER.
3. Press **Calibrate** (F3).
4. Turn Servo Power ON.
5. Press **Setting** (F5).
6. Press **Each axes** (F3).
7. Move the cursor to the right-hand column where "ON OFF" is displayed for "Select SW".
8. Move the cursor up or down to highlight the "ON OFF" row for the desired S, L, U, R, B, or T axis.
9. Press **select SW** (F4) to change the setting from OFF "➡ (not selected)" to ON "■ (selected)".

NOTE: Repeat steps 8 & 9 for each axis to be selected.

10. Press **Register** (F5) to capture the current value of each axis displayed as ON "■ (selected)" for home position.
11. Turn the controller OFF, wait 5 seconds, then power ON.

To edit the absolute data of each axis, complete the following steps in TEACH mode:

1. Press OP2.
2. Enter the 8-digit ID number using the number keys on the operator's panel, then press ENTER.
3. Press **Calibrate** (F3).
4. Press the ARROW RIGHT key.
5. Move the cursor to the axis for which absolute data is to be changed.
6. Press **Data chg** (F5).
7. Press CANCEL to clear data on the input line.
8. Enter the desired absolute data value using the number keys.
9. Press ENTER.
10. Repeat Steps 5 through 9 as needed.
11. Turn the controller OFF, wait 5 seconds, then power ON.

1.8 REGISTERING CONCURRENT I/O USER MESSAGES

Up to sixteen User Messages (0-15) can be registered for use in Concurrent I/O.

To register a message, complete the following steps in TEACH mode:

1. Press OP2.
2. Enter the 8-digit ID number using the number keys on the operator's panel.
3. Press ENTER.
4. Press the ARROW RIGHT key.
5. Press **User Msg** (F2).
6. **Page↓** (F1) or **Page↑** (F2), then move the cursor to the desired message number.
7. Press **Msg chg** (F5).
8. Press CANCEL to clear existing message, or press the ARROW RIGHT key and use (F2), (F3), & Backspace (F4) to edit and enter up to 32 characters using alphabet (CAP/LC), symbols, or numbers.
9. Press EXIT (F5).
10. Press ENTER.

1.9 REGISTERING CONCURRENT I/O USER ALARMS

Up to sixteen User Alarm messages (0-15) can be registered and are numbered 2010-2160.

To register an alarm message, complete the following steps in TEACH mode:

1. Press OP2.
2. Enter the 8-digit ID number using the number keys on the operator's panel.
3. Press ENTER.
4. Press the ARROW RIGHT key.
5. Press **User Alarm** (F3).
6. **Page↓** (F1) or **Page↑** (F2), then move the cursor to the desired message number.
7. Press **Msg chg** (F5).
8. Press CANCEL to clear existing message, or press the ARROW RIGHT key and use (F2), (F3), & Backspace (F4) to edit and enter up to 32 characters using alphabet (CAP/LC), symbols, or numbers.
9. Press EXIT (F5).
10. Press ENTER.

1.10 DISPLAYING THE I/0 MONITOR

The I/O Monitor displays the status of various signals related to concurrent I/O. It is available in DISPLAY mode where no user ID number is required so that it can be used for normal I/O status check.

To display the I/O Monitor, complete the following steps:

1. Press DISP.
2. Press **Diagnosis** (F4).
3. Press the ARROW RIGHT key.
4. Press **IO monitor** (F5).
5. Choose **Univ. I/O** (F1), **Specified** (F2), **Inner-sig.** (F3), **Cntl-sig.** (F4), or **CNTR/TMR** (F5), *or press the ARROW RIGHT key and choose* **IO CHANNEL** (F5).

1.11 EDITING CONCURRENT I/O OPERANDS

To EDIT Operands in Node 1: IN-connect or Node 3: OT-connect, complete the following in TEACH mode:

1. Press OP2.
2. Enter the 8-digit password.
3. Press ENTER.
4. Press the ARROW RIGHT key.
5. Press **Concur. I/O** (F1).
6. Choose **IN-connect** (F2) or **OT-connect** (F3).
7. Using **Page↓** (F1) or **Page↑** (F2) and the cursor keys, move to the desired step address.

Note: *Node 1: IN-connect => Only OUT#XXXX operands for STR#2XXX can be edited;*

Node 3: OT-connect => Only STR#XXXX operands for OUT#3XXX can be edited.

8. Press **Line edit** (F5).
9. Enter the desired 4-digit operand.
10. Press ENTER.
11. Repeat steps 7-10 for each operand change in the chosen Node.
12. Press **Disp chg** (F3).
13. Press **Compile** (F5). [Servos must be OFF] If for any reason you need to cancel the editing, rather than Compile, then press the ARROW RIGHT key, press **EDIT-CANCL** (F5), and then press **Execute** (F5).

Note: *When compiling is complete, the flashing word "EDITING" will disappear. However, if logic errors are found, the cursor will stop at the faulty step that needs correction.*

Table 1-1 Classification of Concurrent I/O Signals

SIGNAL	CLASSIFICATION	REMARKS	SIGNAL RANGE
0XXX	ROBOT universal input	Referenced with input instructions of the job	0010 to 0127 (96 signals)
1XXX	ROBOT universal output	Output with output instructions of the job	1010 to 1087 (64 signals)
2XXX	C. I/O input	Signal no. corresponding to input terminal	2010 to 2127 (96 signals)
3XXX	C. I/O output	Signal no. corresponding to output terminal	3010 to 3087 (64 signals)
4XXX	Specified control input	Signal to change the operating condition of the robot	4010 to 4057 (40 signals)
5XXX	Specified status output	Signal notifying the operating condition of the robot	5010 to 5107 (80 signals)
6XXX	Counter/Timer	Timer to be used in the concurrent I/O.	6010 to 6013 (4 signals)
		Counter to be used in the concurrent I/O.	6014 to 6017 (4 signals)
7XXX	Inner Relay signal	Auxiliary relay in the concurrent I/O	7010 to 7207 (160 signals)
80XX	Inner Control signal	Monitoring of the hardware signal status of the robot control section	8010 to 8027 (16 signals)

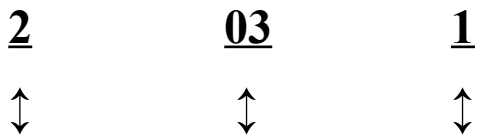
Notes:

① *The first digit (0-9) of the four-digit operand number designates the signal type.*

② *The middle two digits represent the group number (e.g., 01-08 for Universal Outputs.*

③ *The last digit (0-7) denotes the signal classification for each bit in the group of eight.*

EXAMPLE: Signal "2031" is broken out in the following way:



① External *Inputs* ② *Group Number* 3 ③ *Input #2*

2.0 OP1 KEY FEATURES

2.1 CUBIC INTERFERENCE ZONES

The ERC has four possible cubes available. Cubes 1 & 2 are standard and cubes 3 & 4 can be purchased as an option. The cubes are internally tied to specified inputs and outputs:

The specified outputs will turn on when the robot is inside the cubes. There are four methods to teach cubes: (1) defining cubes by MIN-MAX method (see Section 2.1.1), and (2) defining cubes by CUBE LENGTH method (see Section 2.1.2). (3) PANEL method for defining or modifications (see Section 2.1.3) and (4) Setting SC PARAMETERS (see Section 2.1.4).

Defining cubes by Manual (Servo power with teach pendant) Min-Max method sets minimum and maximum positions for the cube with servo power ON.

To define a cube using MANUAL MIN/MAX method, complete the following steps in TEACH mode:

1. Press OP1.
2. Press the ARROW RIGHT key.
3. Press **CUBIC** (F5).
4. Press **Manual** (F5).
5. Press **MIN/MAX** (F4).
6. Turn servo power ON and ENABLE the teach pendant.
7. Press + or - to choose the desired cube MIN.
8. Jog robot to MINIMUM position for cube.
9. Press MODIFY on teach pendant.
10. Press RECORD.
11. Press + or - to choose the desired cube MAX.
12. Jog the robot to the MAXIMUM position for cube.
13. Press MODIFY on teach pendant.
14. Press RECORD.

Defining cubes by MANUAL RECORD Center Point and SET the Cube Length creates dimensions for the cube's edge lengths in X, Y, & Z based on the center position point.

To define a cube using CUBE LENGTH method, complete the following steps in TEACH mode:

1. Press OP1.
2. Press the ARROW RIGHT key.
3. Press **CUBIC** (F5).
4. Press **Manual** (F5).
5. Press **CUBIC LEN.** (F5).
6. Press **SET LENGTH** (F3).
7. Move the cursor to the dimension to be modified (X, Y, or Z).
8. Press **Data chg** (F4).
9. Press CANCEL.
10. Enter the numeric value in millimeters for the desired cube dimension.
11. Press ENTER.
12. Repeat Steps 7 through 11 for each dimension to be modified.
13. Turn servo power ON and ENABLE the teach pendant.
14. Press + or - to select the desired cube #1-4.
15. Jog the robot to the desired center point of the cube.
16. Press MODIFY on the teach pendant..
17. Press RECORD.
18. Press **EXIT** (F5).

To define or make modifications to existing cube dimensions using the number key pad on the operator's panel, complete the following steps in TEACH mode:

1. Press OP1.
2. Press the ARROW RIGHT key.
3. Press **CUBIC** (F5).
4. Press **Page↓** (F1) or **Page↑** (F2) to choose the desired cube #1-4.
5. Cursor to the Maximum/Minimum X, Y, or Z dimension to be changed.
6. Press **Panel** (F4).
7. Press CANCEL.
8. Enter the desired numeric value in millimeters.
9. Press ENTER.
10. Repeat Steps 5 through 9 for each dimension to be modified.

To define or make modifications to existing cube dimensions by setting the SC parameters at the operator's panel, complete the following steps in TEACH mode:

1. Press OP2.
2. Enter the 8-digit ID number with the number keys on the operator's panel.
3. Press ENTER.
4. Press **Parameter** (F1).
5. Press **SC** (F1).
6. **Page↓** (F1) or **Page↑** (F2) if necessary, then cursor to the desired parameter. [SC026-SC031 are the maximum X,Y,Z and minimum X,Y,Z for CUBE 1; SC032-SC037 are CUBE 2 parameters; SC088-SC093 are Cube 3 parameters; and SC094-SC099 are Cube 4 parameters]
7. Press **Data chg** (F3).
8. Press CANCEL.
9. Enter value in units of 0.1 millimeter.
10. Press ENTER.
11. Continue by moving the cursor to each parameter to be modified and repeat Steps 7 through 10.
12. Turn the controller OFF, wait 5 seconds, then power ON.

To display a cube's Specified Output Status, complete the following steps:

1. Press DISP.
2. Press **Diagnosis** (F4).
3. Press the ARROW RIGHT key.
4. Press **Specif.OUT** (F4).
5. Press **Page↓** (F1) or **Page↑** (F2) to the specified output number. [SOUT#027-030 = Cubes #1-4].

When the robot is inside the cube zone, the specified output will turn ON. The square will switch to the ON setting.

Note: The corresponding Concurrent I/O OPERANDs are #5042, #5043, #5044, and #5045.

2.2 Position Monitoring Function by Feedback Pulse

Rather than monitoring an entire cube area, the POSOUT instruction executes Position Monitoring for any specified *step*, and consequently can control the interlock signal for external peripheral devices, etc. There are 8 POSOUT Files available and each must be set up in advance.

To set up the POSOUT FILES, complete the following steps in TEACH mode:

1. Press OP1.
2. Press the ARROW RIGHT key.
3. Press **POSOUT F.** (F1).
4. Press **Page↓** (F1) or **Page↑** (F2) to elect the desired file.
5. Cursor to the data to be changed.
6. Press **Data chg** (F5).
7. Press CANCEL
8. Enter the correct value.
9. Press ENTER.
10. Repeat steps 5 through 8 for each data change.

To program the POSOUT instruction, complete the following steps in TEACH mode:

1. Move the cursor to the line in the job where the instruction is to be inserted.
2. Press EDIT.
3. Press **Insert** (F1).
4 Press **In/Out** (F1).
5. Press the ARROW RIGHT key.
6. Press **POSOUT** (F1).
7. Enter File No.(1-8) for position monitor.
8. Press ENTER.

When the instruction is executed, position monitoring starts after the output relay in the specified file is turned on. If the robot position (checked by feedback pulse) is shifted in excess of allowable pulses of the file specified by the instruction execution step position, then the pause position signal is turned on.

The Position monitoring status is checked in the diagnosis display.

To access the Position Monitoring Diagnosis Display, complete the following steps:

1. Press DISP.
2. Press **Diagnosis** (F4).
3. Press the ARROW RIGHT key.
4. Press **POSOUT DSP** (F2).

3.0 PROGRAMMING SHORTCUTS

3.1 USE OF "SAME COMMAND" TO REPEAT AN INSTRUCTION

When a job is in Edit mode the soft key **Same comd.** (F5) is available to repeat a desired instruction or motion type immediately below the cursor.

To use the *Same comd.*, complete the following in TEACH EDIT mode:

1. Press **Insert** (F1).
2. Press **Same comd.** (F5)
3. Complete the line or step by following the menu choices.

NOTE: If a motion step command was chosen, then **POS-VAR** *(F4) may be used, or* **NOW POS** *(F5) which will accept the robot's current position (whether servos are ON or OFF).*

4. Press ENTER to insert the command into the job.

3.2 RESERVED JOB LIST

The Reserved Job Names are job names that will be used frequently. Each Reserved Job Name is registered on the Reserve Job List which holds ten names or prefixes. Using a Reserved Job Name simplifies entry of new job names.

To register a reserved name on the list, complete the following steps:

1. Press TEACH.
2. Press **New job** (F3).
3. Press **reserve JOB** (F5).
4. Press **Prog. name** (F5).
5. Press ABC (F1).
6. Move the cursor to each desired character, then press ENTER.

NOTE: Job names are limited to eight characters; therefore, the reserved prefix must contain 6 characters or less.

7. Press **EXIT** (F5).
8. Press ENTER.

To use a Reserved Job Name to name a new job, complete the following steps:

1. Press TEACH.
2. Press **New job** (F3).
3. Press **reserve JOB** (F5).
4. Move the cursor to the desired reserved job name.
5. Press **Prog. job** (F4).

NOTE: The reserved job name at the cursor position is displayed on the input line. It can be used once "as is" for a job name, but from then on as only a prefix.

6. Enter any additional characters using the numeric keypad only.
7. Press ENTER.

3.3 SEARCH OPTIONS

When the content of a job is displayed on the screen in TEACH mode, it is possible to search for a specific instruction, line, step, or label. The soft key **Search** menu appears as follows:

(F2) INST Carries out an instruction search

(F3) LINE Carries out a line address number search

(F4) STEP Carries out a step address number search

(F5) LABEL Carries out a label search

To search for an instruction within a job's content selected in TEACH mode, complete the following steps:

1. Press EDIT.
2. Press the ARROW RIGHT key.
3. Press **Search** (F3).
4. Press **INST** (F2).
5. Choose the desired instruction-type soft key. (ARROW RIGHT if necessary to find desired INST-type)

   The soft key labels for the instruction type appear.
6. Choose the desired instruction from the soft key labels F1-F5. (ARROW RIGHT if necessary to find more choices)
7. Press ENTER.

The cursor will advance to the first line it finds with the chosen instruction.

To search for a line within a job's content selected in TEACH mode, complete the following steps:

1. Press EDIT.
2. Press the ARROW RIGHT key.
3. Press **Search** (F3).
4. Press **Line** (F3).
5. Using the number keys, enter the line number for which you are searching.
6. Press ENTER.

The cursor moves to the line number address.

To search for a step within a job's content selected in TEACH mode, complete the following steps:

1. Press EDIT.
2. Press the ARROW RIGHT key.
3. Press **Search** (F3).
4. Press **Step** (F4).
5. Using the number keys, enter the step number for which you are searching.
6. Press ENTER.

The cursor moves to the step number address.

To search for a label within a job's content selected in TEACH mode, complete the following steps:

1. Press EDIT.
2. Press the ARROW RIGHT key.
3. Press **Search** (F3).
4. Press **Label** (F5).
5. Enter the characters of the label.
6. Press ENTER.

The cursor will move to the desired label.

3.4 DEFAULT SPEED INSTRUCTION

It is possible to program a position and motion type without designating a speed. However, prior to any steps *without* speeds, the *default* SPEED instruction must be inserted in the program that states the desired velocities.

To access the SPEED setting, complete the following steps in TEACH EDIT mode:

1. Press **Insert** (F1).
2. Press the ARROW RIGHT key.
3. Press **Motion** (F1).
4. Press the ARROW RIGHT key.
5. Press SPEED (F3).
6. Chose **VJ speed** (F2), **V of tcp** (F3), **V of pose** (F4), or **V of ex** (F5).
7. Enter the desired default speed.

   Ranges:

   VJ = 0.01-100.00%.

   V= 1-9000 cm/min. or 0.1-1500.0 mm/sec.

   VR= 1-360 deg./sec.

   VE= 0.01-100.00%.
8. Press ENTER (the instruction moves to the buffer line).
9. Press ENTER (the instruction moves into the job).

The following is a section sample of a job with a default speed. In Steps 003 through 006, the robot will move at 4500 cm/min for the MOVC & MOVL, and in Step 007 the robot will move at 50% for the MOVJ.

```
LINE:   STEP:   INSTRUCTION:
0000                    NOP
0001              001   MOVJ    VJ=25.00    CONT
0002              002   MOVL    V=276             CONT
0003                SPEED  V=4500  VJ=50.00
0004              003   MOVC    CONT
0005              004   MOVC    CONT
0006              005   MOVC    CONT
0007              006   MOVL    PL=0
0008              007   MOVJ    CONT
0009                    END
```

NOTE: Each motion step that is intended to use the default SPEED must have the RECORDed speed

removed by doing a **Data dlt** *(F2) in Line edit.*

4.0 NWAIT, CWAIT, & UNTIL INSTRUCTIONS

4.1 PROGRAMMING THE NWAIT TAG

The No Wait (NWAIT) motion tag will allow all instruction lines programmed between the tag and the next step to be executed while the robot moves to the NWAIT tagged position step.

In the example below, while the robot travels to Step 009, instructions on lines 0013 through 0018 will be executed.

```
LINE:    STEP:    INSTRUCTION:
0011              008  MOVL     V=138 CONT
0012              009  MOVL     V=138 CONT NWAIT
0013                   DOUT OT#02    1
0014                   TIMER     T=1.5
0015                   DOUT OT#02    0
0016                   DOUT OT#03    1
0017                   TIMER     T=2.0
0018                   DOUT OT#03    0
0019              010  MOVL     V=276 CONT
```

NOTE: If the instructions of the NWAIT (lines 0013-0018 above) take a longer period of time than the robot's travel to the NWAIT step (from step 008 to 009 above), then the robot will hesitate at this step (step 009 above) until the instructions are completed before going to the next step.

To access the NWAIT tag, complete the following steps with the job in TEACH EDIT mode:

1. Move the cursor to the desired step to add the NWAIT tag.
2. Press **Line edit** (F4).
3. Press **Data add** (F3).
4. Press **NWAIT** (F5).
5. Press ENTER.

NOTE: The NWAIT command tag can also be added while programming with position variables or incremental moves in the MOTION menu. To add the NWAIT command tag after the position variable address is entered press NWAIT (F5) and ENTER.

4.2 PROGRAMMING THE CWAIT INSTRUCTION

The Cancel No Wait (CWAIT) is a separate **Sequence** instruction that will resume normal line by line job execution.

In the example below, only instructions on lines 0013 through 0015 will be executed while the robot travels to Step 009.

LINE:	STEP:	INSTRUCTION:
0011	008	MOVL V=138 CONT
0012	009	MOVL V=138 CONT NWAIT
0013		DOUT OT#02 1
0014		TIMER T=1.5
0015		DOUT OT#2 0
0016		**CWAIT**
0017		DOUT OT#03 1
0018		TIMER T=2.0
0019		DOUT OT#03 0
0020	010	MOVL V=276 CONT

NOTE: If the instructions of the NWAIT (lines 0013-0015 above) take a longer period of time than the robot's travel to the NWAIT step (from step 008 to 009 above), then the robot will hesitate at this step (step 009 above) until the instructions are completed before continuing with the next instruction or step.

To program the CWAIT Sequence instruction, complete the following steps with the job in TEACH EDIT mode:

1. Move the cursor to the line where the CWAIT will be inserted.
2. Press **Insert** (F1).
3. Press **Sequence** (F2).
4. Press the ARROW RIGHT key.
5. Press **CWAIT** (F2).
6. Press ENTER.

4.3 PROGRAMMING THE UNTIL TAG

An UNTIL tag allows the robot to move toward a programmed position until an input signal is received. A Linear or Joint motion step can be programmed with an UNTIL tag.

In the example below, the robot will abort moving toward the position in Step 020 when Input #3 is ON and proceed directly to Step 021. If the job contains no other motion steps after the UNTIL tag, the robot will stop when the input is ON.

LINE:	STEP:	INSTRUCTION:			
0024	019	MOVL	V=1125	CONT	
0025	020	MOVL	V=2250	CONT	UNTIL IN#03=1
0026	021	MOVL	V=4500	CONT	

To program an UNTIL tag on a MOVL or MOVJ, complete the following steps:

1. Display the job on the screen in Teach and Edit mode.
2. Press **Line edit** (F4).
3. Press **Data add** (F3).
4. Press **UNTIL** (F4).
5. Enter the input number.
6. Press ENTER, and proceed via ONE of the following possibilities:

7. Press ON (F3) or OFF (F4),
8. Press ENTER.

or

7. Press **Variable** (F5).
8. Enter Byte variable number.
9. Press Enter (onto the buffer line).
10. Press ENTER (into job).

NOTE: When programming with position variables or incremental moves in the MOTION menu, the UNTIL command tag can be added after the position variable address is entered simply by pressing UNTIL (F4) and continuing as above in Steps 5 through 8 (or 5 through 10).

5.0 VARIABLES

5.1 ARITHMETIC VARIABLES

There are 4 basic types Arithmetic Variables available for programming with the ERC; **Byte, Integer, Double** Precision Integer, & **Real .**

Table 5-1 Arithmetic Variables

DATA TYPE	VARIABLE ADDRESS NO.	RANGE
BYTE	B00 to B99	O to 255
INTEGER	I00 to I99	-32,768 to 32,767
DOUBLE PRECISION	D00 to D99	-2,147,483,648 to 2,147,483,647
REAL	R00 to R99	-1.70141E+38 to 1.70141E+38 (precision for values between -1&+1: -9.99999E-38 to 9.99999E-38)

5.2 POSITION VARIABLES

There are 2 data formats for **Robot** Position Variables: **Pulse** & **XYZ**. There are 4 data types for XYZ: **BF**, **RF**, **UF**, **& TF. External Station** position variables are Pulse type only. In EDIT, Position data may be recorded with servo power ON for all types except **TOOL** (see Sections 6.2 and 6.3). Data for **TOOL,** and all other types as well, may be hand-entered in EDIT for purposes of shifting, etc. (see Sections 6.5 and 6.6).

Table 5-2 Position Variables

DATA TYPE	DATA FORMAT	VARIABLE NO.
Pulse	SLURBT	P000 to P063
Base (BF)	XYZ	P000 to P063
Robot (RF)	XYZ	P000 to P063
User (UF#)	XYZ	P000 to P063
Tool	XYZ	P000 to P063
External Station	Pulse	EX00 to EX63

5.3 SYSTEM VARIABLES ($)

There are 2 types of System Variables used with the ERC. All are designated by the "$" symbol. The ARITHMETIC type $B-status and the POSITION $P-type or $XE-type are "fetched" using the SET Arithmetic Function and stored in the corresponding User type. Each System Variable address is dedicated for a specific use.

The $B-status type variable file may be displayed as follows:

1. Press DISP.
2. Press **Diagnosis** (F4).
3. Press the ARROW RIGHT key four times.
5. Press **$B-status** (F4).

Some samples for dedicated use of $-type variables are included in the table below:

Table 5-3 System Variables

SYSTEM VARIABLES	USER TYPE	DEDICATED USE
$B-type	B-type	$B002 Return code for SRCH instruction $B006 Return code for HSEN instruction $B008 Return code for SYSTART instruction
$P-type	P-type	$P000 Current PULSE position data $P001 Current XYZ (BF) $PX011 REFP1 Pulse $PX012 REFP2 Pulse $PX013 REFP3 Pulse $PX014 REFP4 Pulse $PX015 REFP5 Pulse $PX016 REFP6 Pulse $PX017 REFP7 Pulse $PX018 REFP8 Pulse
$XE-type	EX-type	$XE00 Current PULSE position data of External Axis

6.0 POSITION VARIABLES

6.1 ACCESS TO POSITION VARIABLE DISPLAY

The ERC supports 64 Position Variable addresses #P000-P063 for storing position data, shift amounts, and incremental move data.

To access the Position Variable display, complete the following steps:

1. Press DISP.
2. Press **File** (F3).
3. Press **Variable** (F4).
4. Press the ARROW RIGHT key.
5. Press **Robot** (F4). The Position Variable Display appears
6. Press **Page**↓ (F1) or **Page**↑ (F2) to desired number,

   or

   press SEARCH (F4), enter the number, then press ENTER.

To delete the data of the displayed variable, complete the following steps:

1. Press EDIT.
2. Press **Ers data** (F3).
3. Press **Execute** (F5).

NOTE: **Erase data** *must be executed to access menu to specify data-type (Pulse or XYZ) if a* change *of data-type is desired. The data type must be established before the position variable can be used.*

6.2 STORE POSITION VARIABLE IN PULSE COUNTS

To store the robot's current position as the displayed Position Variable in Pulse Counts, complete the following steps: (see Section 6.1 to display and/or **Ers data** type)

1. Turn ON the servo power.
2. Press EDIT.
3. Press **Teach pend** (F5).
4. Press **Pulse-type** (F4).
5. Press ENABLE on the Teach Pendant.
6. Press the + or **-** key until desired address appears on the LED display with **PLS** data-type.
7. Jog the robot to the desired position.
8. Press RECORD. (If just changing existing data, press MODIFY on the pendant before pressing RECORD.)

6.3 STORE POSITION VARIABLE FOR XYZ

To store the robot's current position as the displayed Position Variable for XYZ, complete the following steps: (see Section 6.1 to display and/or **Ers data** type)

1. Turn ON the servo power.
2. Press EDIT.
3. Press **Teach pend** (F5).
4. Press **XYZ-type** (F5).
5. Press ENABLE on the Teach Pendant.
6. Press the + or **-** key until desired address appears on the LED display with **XYZ** data-type .
7. Jog the robot to the desired position.
8. Press RECORD. (If just changing existing data, press MODIFY on the pendant before pressing RECORD.)

6.4 CHECK POSITION VARIABLE

To check the position set to the displayed Position Variable, complete the following steps: (see Section 6.1 to display)

1. Turn ON servo power.
2. Press EDIT.
3. Press **Teach pend** (F5).
4. Press ENABLE on the Teach Pendant.
5. Press the +/- keys to display the desired address.
6. WARNING! The following step will cause the robot to move from its current position directly to the Position Variable location. Ensure that the direct path is clear before proceeding. After safe operation is ensured, press FWD until the position variable number stops flashing.

6.5 EDIT POSITION VARIABLE IN PULSE COUNTS

To edit the displayed Position Variable in Pulse Counts, complete the following steps: (see Section 6.1 to display and/or Ers data type)

1. Press EDIT.
2. Press **Pnl. input** (F4).
3. Move the cursor to the chosen coordinate axis.
4. Press **Data chg** (F4).
5. Enter the correct numeric value using the number key pad.
6. Press ENTER.
7. Repeat Steps 3 through 6 for each desired setting or modification of S, L, U, R, B, and T.
8. Press **QUIT** (F5).

6.6 EDIT POSITION VARIABLE FOR XYZ AND Tx, Ty, Tz

Whenever a position variable is used for incremental moves or shift amounts, the data can be entered manually in units of millimeters and degrees.

To edit the displayed Position Variable for XYZ in mm and Tx, Ty, and Tz in degrees, complete the following steps: (see Section 6.1 to display and/or **Ers data** type)

1. Press EDIT.
2. Press **Pnl. input** (F4).
3. Move the cursor to the dimension to be set or modified.
4. Press **Data chg** (F4).
5. Enter X, Y, Z (-8388.608 to +8388.607mm) or Tx, Ty, Tz (-180.00 to 180.00 degrees).
6. Press ENTER.

7. Repeat Steps 3 through 6 for each dimension to be set or modified.

6.7 PROGRAMMING MOTION WITH POSITION VARIABLES

To program Motion in a job using position variables, complete the following steps with the job in TEACH EDIT mode:

1. Press **Insert** (F1).
2. Press the ARROW RIGHT key.
3. Press **Motion** (F1).
4. Choose **MOVJ** (F1), **MOVL** (F2), **MOVC** (F3), or ARROW RIGHT twice to choose **MOVS** (F1).
5. Press **Pos-var** (F4).
6. Enter the Position Variable number (0-63).
7. Press ENTER.
8. Press **Speed** (F2/F4).
9. Press **V of tcp** (F3) if MOVL, MOVC, or MOVS.
10. Enter the **tcp** speed for MOVL, MOVC, MOVS or **Joint** speed for a MOVJ.
11. Press ENTER, then continue with other menu choices if desired (CONT, PL, UNTIL, NWAIT).
12. Press ENTER.

A job example is shown below:

LINE:	STEP:	INSTRUCTION:
0000	000	NOP
0001	001	MOVJ P000 VJ=25.00 CONT
0002	002	MOVJ VJ=100.00 CONT
0003	003	MOVJ VJ=12.50 CONT
0004	004	MOVL P017 V=76 CONT
0005	005	MOVL P018 V=76 PL=0
0006	006	MOVL P023 V=138 PL=0
0007	007	MOVL P026 V=138 CONT
0008	008	MOVJ VJ=12.50 CONT
0009	009	MOVJ P000 VJ=100.00 CONT
0010		END

NOTE: In Steps 001 and 009, above, the same position variable has been used.

6.8 INCREMENTAL MOVES (IMOV)

An INCREMENTAL MOVE (IMOV) is a **linear** move of a designated amount and direction based on a position variable. The position variable must be set up in advance in XYZ to a specific number of millimeters in one or more directions and orientations in degrees (refer to Section 6.6 to do this). For example, if P027 is set for Z=+50 mm, then the robot will move linearly 50 mm in the positive direction.

To program an INCREMENTAL MOVE (IMOV) based on a Position Variable's previously set data, complete the following steps with the job in TEACH EDIT mode:

1. Press **Insert** (F1).
2. Press the ARROW RIGHT key.
3. Press **Motion** (F1).
4. Press **IMOV** (F4).
5. Enter the Position Variable number.
6. Press ENTER.
7. Press **Speed** (F2).
8. Press **V of tcp** (F3),
   or
   press **V of pose** (F4) if data in Position Variable is only degrees orientation.
9. Enter the **V=** (Control-point-speed) or enter the **VR=** (Angle-speed).
10. Press ENTER.
11. Press the ARROW RIGHT key.
12. Press **SEL. coord** (F4).
13. Choose **Base coord** (F2), **Tool coord** (F4), or **Robt coord** (F5)
    or
    choose **User coord** (F3), then enter number of frame and press ENTER.
    Continue with EDIT menu choices for PL level, UNTIL or NWAIT if desired.
14. Press ENTER (the contents of the buffer line move into the job).

In the job sample below, the robot will move in a stair-step pattern five times due to the IF statement on the JUMP instruction. The stair-step pattern is created by setting P002 in XYZ format to Z=50 mm (all other elements remain zero millimeters/degrees), and P003 in XYZ format to Y= -50 mm (all other elements remain zero millimeters/degrees).

```
LINE:   STEP:   INSTRUCTION:
0000            NOP
0001    001     MOVJ  VJ=25.00  CONT
0002            SET I00  0
0003            *3
0004    002     IMOV P002 V=2250  CONT  UF#8
0005    003     IMOV P003 V=2250  CONT  UF#8
```

```
0006             INC I00
0007             JUMP *3 IF I00 < 6
0008      004    MOVJ  VJ=100.00  CONT
0009             END
```

7.0 ARITHMETIC FUNCTIONS

There are nineteen Arithmetic Functions available on the ERC. They appear in the **Arithmetic** soft key menu as follows:

[F1]	[F2]	[F3]	[F4]	[F5]	
SET	**INC**	**DEC**	**SETE**	**GETE**	*(ARROW RIGHT)*
	ADD	**SUB**	**MUL**	**DIV**	*(ARROW RIGHT)*
	AND	**OR**	**NOT**	**XOR**	*(ARROW RIGHT)*
CNVRT CLEAR				*(ARROW RIGHT)*	
	SQRT	**SIN**	**COS**	**ATAN**	

NOTE: *In* **all** *ARITHMETIC FUNCTIONS, the* ***results*** *are stored in the variable address* immediately following the instruction.

7.1 VALUE SETTING FUNCTIONS

1. **Clear** a section of any existing values to zero in consecutive Byte, Integer, Double, or Real variable addresses.

   Examples:

   CLEAR B06 5 (Clears B06, B07, B08, B09, & B10)

   CLEAR R07 ALL (Clears R07 through end at R99)

2. **Clear** the **STACK** for nested CALL jobs. The STACK is limited to 8 levels of nesting; by clearing the Stack within a nested Call Job, additional levels are made available.

CAUTION! *Once the stack is cleared of previous levels, the controller will* NOT *RETURN to any of these jobs, so programmer must consider safe robot path for application.*

Example:

CLEAR STACK (This job becomes level 1 of 7. When controller gets to END of this job, cycle is finished at level 0).

1. **Set** a designated Byte, Integer, Double, or Real variable address to a <u>constant</u> value.

   Examples:

   SET R09 452.2756 (Sets constant into R09)

   SET D08 -81726354 (Sets constant into D08)

   SET B00 255 (Sets constant into B00)

2. **Set** a designated Byte, Integer, Double, or Real variable address to the same value that's stored in <u>another</u> variable's address <u>of the same type</u>.

   Examples:

   SET R39 R09 (Sets value stored in R09 into R39)

   SET B01 B03 (Sets value stored in B03 into B01)

3. **Set** a designated Byte variable address to a LOW or HIGH value resulting from the first or second byte of an Integer address value expressed in binary. Since the range of Integer type variables is -32,768 to 32,767, all integer values are set into the Byte address as an absolute [+] value from 0-255.

NOTE: TWO BYTES = 1 Word (16 bits).

H I G H								L O W							
128	*64*	*32*	*16*	*8*	*4*	*2*	*1*	*128*	*64*	*32*	*16*	*8*	*4*	*2*	*1*
0	0	0	0	1	0	1	0	1	1	1	1	1	1	0	1
32768	*16384*	*8192*	*4096*	*2048*	*1024*	*512*	*256*	*128*	*64*	*32*	*16*	*8*	*4*	*2*	*1*

Examples for table above where 0000101011111101=2,813:

SET B17 I02 LOW (If I02 = 2,813, then I02 LOW = 253 in B17)

SET B17 I02 HIGH (If I02 = 2,813, then I02 HIGH = 10 in B17)

NOTE: If I02 = -2813 (1111010100000010), then I02 LOW = 2 and I02 HIGH = 245.

4. **Set** all 6 elements of a designated Position variable equal to another Position variable's elements of the same type (Pulse, BF, RF, UF).

NOTE: Data type must be previously defined in position variable file.

Example:

SET P055 P040 (Sets all 6 elements of P040 into P055)

5. **Set** SYSTEM variable ($) information into a user's variable file of the same type.

Examples:

SET P000 $P000 (Sets system's current Pulse position data in P000)

SET B05 $B02 (Sets system's current return code of SRCH in B05)

1. **Convert** a Position variable from Pulse or any Frame type to only one of the Frame types BF, RF, UF, TF and stores the converted data in the same or other designated Position variable address.

Examples:

CNVRT P003 P003 RF (Converts data of P003 into RF type)

CNVRT P004 P005 UF#7 (Converts data of P005 into UF type in P004)

CNVRT P006 P000 BF (Converts data of P000 into BF type in P006)

2. **Convert** the value stored in a Byte variable address into an Integer variable address.

Example:

CNVRT I05 B03 (Converts data from B03 into I05)

3. **Convert** the value stored in an Integer or Double integer variable address into a Real address.

Examples:

CNVRT R07 D03 (Converts data from D03 into R07)

CNVRT R08 I75 (Converts data from I75 into R08)

4. **Convert** the value stored in a Byte, Integer or Real variable address into a Double integer variable address.

Examples:

CNVRT D08 B03 (Converts data from B03 into D08)

CNVRT D09 I75 (Converts data from I75 into D09)

CNVRT D10 R98 (Converts data from R98 into D10)

7.2 POSITION VARIABLE FUNCTIONS

Set Element sets the designated *element* in a Position variable equal to a constant or a value stored in a D-variable address.

Examples:

SETE P007 (1) -500000 (Sets constant into *first element* of P007)

SETE P005 (2) D03 (Sets value of D03 into *second element* of P005)

Get Element gets the designated *element* in a Position variable and stores the value as **Pulse counts** or **microns** in a D-variable address.

NOTE: *XYZ type Position variable elements are* **mm** *and degrees; 1,000 microns = 1 mm and degrees are in 0.01 units.*

Example:

GETE D04 P007 (3) (Gets *third element* of P007 and stores value in D04)

7.3 COUNTING FUNCTIONS

Increment the value stored only in a Byte or Integer variable address by adding one.

Example:

INC B36 (Increases by one the value stored in B36)

Decrement the value stored only in a Byte or Integer variable address by subtracting one.

Example:

DEC I08 (Decreases by one the value stored in I08)

7.4 CALCULATING FUNCTIONS

1. **Add** a constant or the value stored in the same type arithmetic variable address to the designated arithmetic variable address.

   Examples:

   ADD D08 -81726354 (Adds constant to value stored in D08)

   ADD B06 B18 (Adds value in B18 to value in B06)

2. **Add** the 6 elements stored in a Position variable address to the designated Position variable's elements. (Data type must be same; Pulse, BF, RF, or UF.) Suggestion: This function is useful for combining XYZ shift amounts.

   Example:

   ADD P025 P024 (Adds 6 elements of P024 to values in P025)

1. **Subtract** a constant or the value stored in the same type arithmetic variable address from the designated arithmetic variable address.

   Examples:

   SUB I07 -12345 (Subtracts constant from value stored in I07)

   SUB R06 R18 (Subtracts value stored in R18 from value in R06)

2. **Subtract** the 6 elements stored in a Position variable address from the designated Position variable's elements. (Data type must be same; Pulse, BF, RF, or UF.)

NOTE: This function is useful for "zeroing" all 6 elements.

Examples:

SUB P027 P027 (Subtracts all elements from self to "zero" all values)

SUB P056 P023 (Subtracts all elements of P023 from P056)

1. **Multiply** a designated arithmetic variable address by a constant or value stored in another variable address of the same type.

   Examples:

   MUL I05 25 (Multiplies 25 times value in I05)

   MUL R05 R02 (Multiplies value in R02 times value in R05)

2. **Multiply** a single element stored in the position variable address by an *integer* constant.

   Example:

   MUL P009 (3) -4 (Multiplies *integer* constant -4 times third element of P009)

1. **Divide** a constant or the value stored in another arithmetic variable address of the same type *into* designated arithmetic variable address.

   Examples:

   DIV R05 67.854 (Divides constant into value of R05)

   DIV D14 D02 (Divides value in D02 into value of D14)

2. **Divide** a n *integer* constant into a single element stored in the designated Position variable address.

   Example:

   DIV P026 (1) 5 (Divides *integer* constant 5 into first element of P026)

1. Stores the **square root** of a *positive* constant or the *positive* value stored in another Real variable address into only a designated Real variable address.

Examples:

SQRT R08 3 ($\sqrt{3}$ = 1.732051E+00 is stored in R08)

SQRT R14 81726.354 (Square root of *positive* constant is stored in R14)

SQRT R09 5.87427E+16 (Square root of *positive* constant is stored in R09)

SQRT R10 1.2487E-05 (Square root of *positive* constant is stored in R10)

7.5 TRIGONOMETRIC FUNCTIONS

Stores the **sine** of a constant *degrees* or the value stored in another Real variable address into only a designated Real variable address.

Examples:

SIN R08 90	(SIN 90° = 1.000000E+00 is stored in R08)
SIN R01 -60	(SIN -60° = -8.660254E-01 is stored in R01)
SIN R09 R12	(Sine of the *degrees* value in R12 is stored in R09)

Stores the **cosine** of a constant *degrees* or the value stored in another Real variable address into only a designated Real variable address.

Examples:

COS R08 90	(COS 90° = -2.05103E-10 stored in R08)
COS R02 -60	(COS -60° = 5.000000E-01 stored in R02)
COS R09 R12	(Cosine of the *degrees* value in R12 stored in R09)

Stores the **arc tangent** in *degrees* for a constant ratio or for the value stored in another Real variable address into only a designated Real variable address.

Examples:

ATAN R08 1	(ATAN 1 = 45°; 4.500000E+01 is stored in R08)
ATAN R00 - 1.73205E+00	(ATAN -√3 = -60°; -6.000000E+01 is stored in R00)
ATAN R09 R12	(Arc tangent of value in R12 is stored in R09)

7.6 BOOLEAN LOGIC FUNCTIONS

Stores the logical resultant of **AND** for only a Byte variable address *combined with* either a *positive* constant or the value stored in any other Byte variable address.

NOTE: This is useful for masking I/O. To mask inputs 6,7 & 8 of IG#1 after a DIN B00 IG#1, use next line of instruction as AND B00 31 (31=16+8+4+2+1, sum of inputs 1 to 5).

Examples:

AND B00 7
(If B00=63 0011_1111 [63]
AND 0000_0111 [7]
0000_0111
then 7 is stored in B00)

AND B00 B01
(If B00=63 0011_1111 [63]
B01=63 AND 0011_1111 [63]
0011_1111
then 63 is stored in B00)

AND B00 83
(If B00=63 0011_1111 [63]
AND 0101_0011 [83]
0001_0011
then 19 is stored in B00)

AND B00 209
(If B00=63 0011_1111 [63]
AND 1101_0001 [209]
0001_0001
then 17 is stored in B00)

Stores the logical resultant of **OR** for only a designated Byte variable address *combined with* either a *positive* constant or the value stored in any other Byte variable address.

Examples:

OR B01 7
(If B01=63 0011_1111 [63]
OR 0000_0111 [7]
0011_1111
then 63 is stored in B01)

OR B01 B02
(If B01=63 0011_1111 [63]
B02=63 OR 0011_1111 [63]
0011_1111
then 63 is stored in B01)

OR B01 83
(If B01=63 0011_1111 [63]
OR 0101_0011 [83]
0111_1111
then 127 is stored in B01)

OR B01 209
(If B01=63 0011_1111 [63]
OR 1101_0001 [209]
1111_1111
then 255 is stored in B01)

Stores the logical resultant of **NOT** for a *positive* constant or the value stored in any Byte variable address into only a designated Byte variable address.

NOTE: The initial/existing value in the Byte address where the resultant is to be stored is insignificant because it is not used as part of the NOT logic process.

Examples:

```
NOT B02 7          (If B02=63      0011_1111   [63] (insignificant)
                         NOT       0000_0111   [7]
                                   1111_1000   [255-7]
                   then 248 is stored in B02)

NOT B02 B03        (If B02=63      0011_1111   [63] (insignificant)
                   B03=63  NOT     0011_1111   [63]
                                   1100_0000   [255-63]
                   then 192 is stored in B02)

NOT B02 83         (If B02=63      0011_1111   [63] (insignificant)
                         NOT       0101_0011   [83]
                                   1010_1100   [255-83]
                   then 172 is stored in B02)

NOT B02 209        (If B02=63      0011_1111   [63] (insignificant)
                         NOT       1101_0001   [209]
                                   0010_1110   [255-209]
                   then 46 is stored in B02)
```

Stores the logical resultant of **XOR ("EXCLUSIVE OR")** for only a designated Byte variable address *combined with* either a *positive* constant or the value stored in any other Byte variable address.

Examples:

```
XOR B03 7          (If B03=63      0011_1111   [63]
                         XOR       0000_0111   [7]
                                   0011_1000
                   then 56 is stored in B03)

XOR B03 B04        (If B03=63      0011_1111   [63]
                   B04=63  XOR     0011_1111   [63]
                                   0000_0000
                   then 0 is stored in B03)

XOR B03 83         (If B03=63      0011_1111   [63]
                         XOR       0101_0011   [83]
                                   0110_1100
                   then 108 is stored in B03)

XOR B03 209        (If B03=63      0011_1111   [63]
                         XOR       1101_0001   [209]
                                   1110_1110
                   then 238 is stored in B03)
```

8.0 SHIFT FUNCTIONS

8.1 TEMPORARY SHIFT WITHIN A JOB

The Shift Function provides the ability to transfer (shift) a single or group of programmed positions by a specified distance and orientation. It uses position variables for offsets in the X, Y, Z, Tx, Ty, and Tz directions. The shift value is the difference of X, Y, or Z in the specified coordinate system between the programmed position, the shifted position, and the tool displacement of degrees Tx, Ty, and Tz.

By using the Shift Function, a minimal number of robot positions are programmed in order to provide execution at multiple locations. This simplifies and reduces the time required to teach a job. This function is extremely useful for palletizing, stacking, loading/unloading, and other applications that perform repeated functions at multiple locations.

The Shift Function is activated by issuing a SFTON (shift on) command and specifying a position variable and relative coordinate system. The shift must be set up in advance in a position variable. (Refer to Section 6.6 for XYZ.) The shift data can be in any of the following coordinate systems: Base, Robot, Tool, or one of the 8 User Frames. When the Shift function is executed, programmed positions are shifted by the distance(s) indicated in the specified position variable. The Shift Function will affect all programmed positions including points in called child jobs until a SFTOF (shift off) command is executed.

To access the Shift function SFTON, complete the following steps with the job in TEACH EDIT mode:

1. Press **Insert** (F1).
2. Press **SHIFT** (F4).
3. Press **SFTON** (F4).
4. Enter the position variable address number.
5. Press ENTER.
6. Choose **Base coord** (F2), **Tool coord** (F4), or **Robt coord** (F5)

   or

   choose **User coord** (F3), then enter number of frame and press ENTER.
7. Press ENTER (the contents of the buffer line move into the job).

To cancel the shift within the program using SFTOF, complete the following steps with the job in TEACH EDIT mode:

1. Press **Insert** (F1).
2. Press **SHIFT** (F4).
3. Press **SFTOF** (F5). [This will immediately enter the instruction into the job; pressing ENTER is not necessary]

In the job example, below, the motions in Steps 006, 007, and 008 will be shifted during playback.

LINE:	STEP:	INSTRUCTION:
0011	005	MOVJ VJ=25 CONT
0012		SFTON P027 UF#(5)
0013	006	MOVJ P000 VJ=100 CONT
0014	007	MOVL V=1250 CONT
0015	008	MOVL V=1250 CONT
0016		SFTOF
0017	009	MOVJ VJ=100 CONT

8.2 MSHIFT

The MSHIFT function will calculate a desired offset using the difference between two position variables and store it in a third that can be attached to a coordinate frame. This allows the amount of shift in millimeters/degrees to be determined during the execution of a job. The two position variables used to calculate the difference can be set up in PULSE or XYZ; however, if the positions were stored in an XYZ format, then the instruction line must state their data type as well. The data will always be stored in the resultant variable address in XYZ format (see Section 6.0).

In the example below, the MSHIFT in line 0001 will subtract the original Pulse position data (P001) from the new Pulse position data (P002) and store the difference in P003 in Robot Frame XYZ format. Only the motion steps between the SFTON in line 0003 and the SFTOF in line 0009 will be shifted the calculated amount that is stored in P003.

```
LINE:   STEP:   INSTRUCTION:
0000    000     NOP
0001            MSHIFT  P003  RF  P001  P002
0002    001     MOVJ VJ=25  CONT
0003            SFTON  P003
0004    002     MOVJ  VJ=50  CONT
0005    003     MOVL  V=66  PL=0
0006    004     MOVL  V=4500  CONT
0007    005     MOVL  V=4500  CONT
0008    006     MOVL  V=66  PL=1
0009            SFTOF
0010    007     MOVJ  VJ=100  CONT
0011            END
```

NOTE: *If the reference and aimed positions were in an XYZ frame type, then the instruction must also state this data type as shown in the bold print for the example below.*

```
0001        MSHIFT  P008  UF#5  P011  P012  BF
```

To program the MSHIFT function, complete the following steps with the job on screen in TEACH EDIT mode:

1. Press **Insert** (F1).
2. Press **SHIFT** (F3).
3. Press **MSHIFT** (F1).
4. Enter the P variable number for stored shift data.
5. Press ENTER.
6. Choose **BF** (F1), **RF** (F2), **TF** (F3),

   or

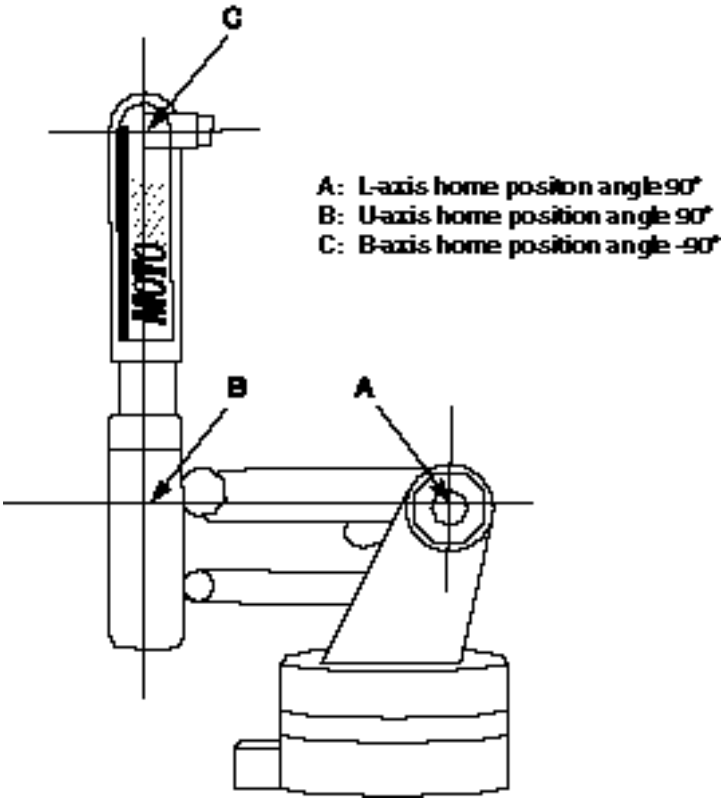
   choose **UF** (F4), then enter the number of the User Frame and press ENTER.
7. Enter the P variable number of the original position.
8. Press ENTER.
9. Enter the P variable number of the desired new shifted position.
10. Press ENTER.
11. If the reference and aimed positions are not in Pulse, then enter the frame type.
12. Press ENTER (to enter MSHIFT into the job from the buffer line).

MOTOMAN, INC

ERC ADVANCED PROGRAMMING TRAINING MANUAL

PART 2: SOFTWARE OPTIONS

RESERVED JOB LIST, 16


C
A: L-axis home positon angle:90°
B: U-axis home position angle:90°
C: B-axis home position angle:-90°
B
A