School of Computing,
Communication and Electronics

Mitsubishi RV-2AJ Industrial Robot Programming and Calibration

Lab Notes

Joerg Wolf
Paul Robinson

Version 0.6 Nov 2005

Table of Contents

page

1. Introduction

This manual provides a hands-on introduction to the Mitsubishi RV-2AJ, 5-axis industrial robot. For further information please consult the Mitsubishi manual [1].

1.1 Safety

REFER TO THE SAFETY SECTION OF YOUR MODULE NOTES – THESE ARE ON THE PORTAL.

The following safety rules must ALWAYS be obeyed. Failure to do so may result in exclusion from the laboratory.

1. **NEVER enter the robot work space when servo power is on!**
   i.e., the green LED (light emitting diode) at the "SVO ON" switch is on.

2. **Always** make sure there is no one in the work cell before starting a program.

3. **Always** give an audible warning before running the program.

4 . **Always** keep one hand at the emergency stop and observe the robot during its full operational cycle

5. **Always** write "**OVRD 20**" as the first line in any robot program. This limits the maximum speed

6. **Never** load the robot with more than 2 kg. If the arm is operated at full stretch limit the maximum load to 0.5 kg.

7. **Never** leave the robot unattended when it is powered up.

1.2 Operational rules

In order to prevent damage to the robot:

- Do not switch off the controller power while the robot has power on the servo
- Avoid collisions. If you think the robot will cause a collision press the emergency stop button immediately.
- Do not change the tool-end coordinates

2. Operation modes

Identify the four parts of the robot system, namely;

1. The Mitsubishi RV-2AJ robot arm
2. The robot controller
3. A Teach Pendant (or TeachBox).
4. The PC

Figure1. The Controller

Emergency Stop Button

Figure 2.

The Teach Pendant

Make sure the emergency stop buttons of both the teach pendant and controller are pulled out.
Switch on the Controller. (Power up takes approx. 20 seconds)

2.1 Teach pendant mode

Turn the key on the Controller to TEACH.
Turn the key on the Teach Pendant to ENABLE.

2.1.1 Moving the robot

To move the robot two buttons must be pushed, i.e., the button under the teach pendant (dead-man's-handle) and the button "STEP/MOVE". (Keeping these two buttons pressed down turns on the servos). A low beeping sound from the PWM can be heard and the LED "SVO ON" goes green. Stay out of the work space when the servos are on (Safety Rule 1 – above)

The robot can be now moved in joint-space (Θ_1 to Θ_5) or Cartesian-space (XYZ).
For Cartesian-space operation press the XYZ button once (top row 3rd across).
The buttons labelled –X,+X,-Y,+Y,-Z,+Z can now be used to move the robot.
The buttons labelled -A,+A are to rotate the robot's end-effector around the end effector Z axis. .
The buttons labelled,-B,+B are to rotate the robot's end-effector around the end effector …… axis?

Pressing the XYZ button at any time displays the position of the end-effector.
To open and close the hand keep the dead-man's handle pressed, release the "STEP/MOVE" button and keep the "HAND" button pressed instead.
To open the hand press "+C" to close it "-C"

2.2 Automatic Operation

Automatic Operation is used to run programs that are stored in the robot controller memory. The program will cycle, i.e. run continuously once started.

Turn the key on the Teach Pendant to DISABLE.
Turn the key on the Controller to AUTO(Op.) .
Press the "Change Display" button until the status display shows "P.xxxxxx"
Press the "Up" or "Down" button to select your program.
Switch on the servos by pressing the "SVO ON" button.
Press "START"
If the "END" button is pressed, the robot will finish the current program cycle and stop.
If the "STOP" button is pressed, the robot stops immediately.

2.3 Automatic Operation, controlled externally (via PC)

Turn the key on the Teach Pendant to DISABLE.
Turn the key on the Controller to AUTO(Ext.).

To communicate with the robot start up:
"C:\Mitsubishi Progs\RoboExplorer\Mits_Soft\RoboExplorer.exe"

In the background a second program will automatically start up called "RoboCom.exe" RoboCom has a light blue background when communication with the robot is established.

2.3.1 Creating and downloading a MELFA-BASIC file

A MELFA-BASIC file is a program to control the robot. A typical MELFA-BASIC program consists of a sequence of move commands which move the robot's end-effector, sometimes called the gripper, to predefined target positions (Teach Points). A MELFA-BASIC file must end with an empty line.

Use the RoboExplorer to select your MELFA-BASIC file from the hard disk. A MELFA-BASIC file should have the extension ".MB4". An example file can be found with this manual: TEST.MB4.

Drag the MELFA-BASIC file across onto the RV-2AJ robot icon, see Figure 3.

Drag the MELFA-Position file across to the left onto the MELFA-BASIC file that has just been downloaded.

A progress-bar will pop-up during downloading of the MELFA-BASIC and MELFA-Position file.

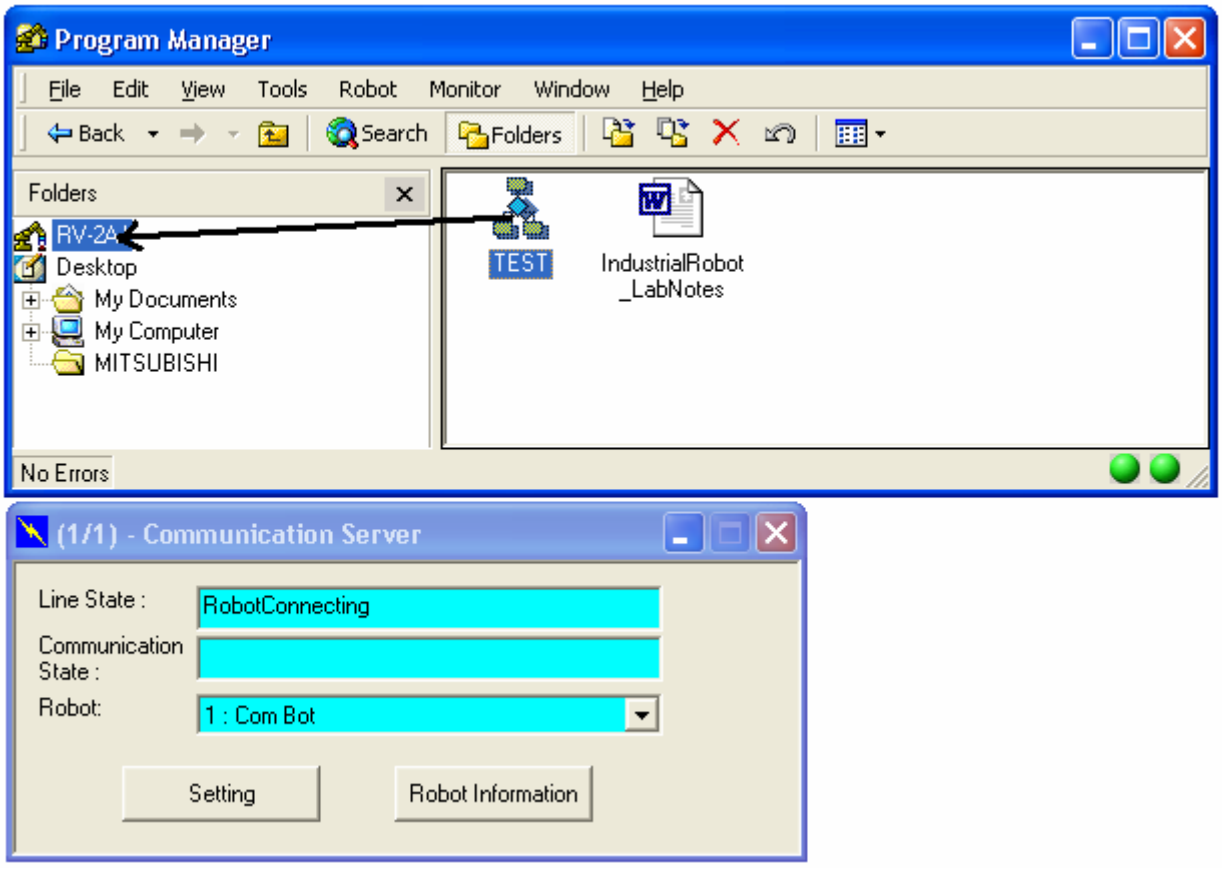


Figure 3 – Running a MELFA Basic programme from the PC

2.3.2 Creating and downloading a teach-point file

Teach points (TP's) define target positions for the robot. A Teach point for the RV-2AJ robot consists of 5 values, namely:

1. Cartesian X position
2. Cartesian Y position
3. Cartesian Z position
4. A – wrist rotation A
5. B – wrist rotation B

Hence the Mitsubishi RV-2AJ robot has only 5 degrees of freedom. We know that in general to both position (x,y,z) and orientate (roll, pitch, yaw) an object in space requires SIX degrees of freedom. Therefore this robot, in common with many other industrial robots, has reduced functionality. In practice this does not seriously limit it's the range of its application.

To teach the robot a new teach point (TP) position;

- Switch the robot into "Teach pendant mode" and drive the arm to the desired target position.
- Once the robot is in its target position open "Tools/TP Editor" from the menu on the PC software.
- Press the right mouse button and choose the option "  New Teach Point"
- Right click on the newly created teach-point and select the option "  Learn Robot's Position"
- The teach-point can be renamed using the option "  Edit Teach Point". The first letter of the name of the new TP position should be a "P". e.g. "P1", "P10" , "PSAVE"
- Right mouse-click the option save. The Teach Point file is saved to the hard drive. A teach point file must have the extension ".POS"

A teach point file can only be downloaded to a robot after the corresponding MELFA-BASIC file (.MB4) has been downloaded. To download a teach point file drag the file onto the MELFA-BASIC file in the robot (right hand side of the screen). By dragging the teach point file onto the MELFA-BASIC program in the robot rather than onto the robot itself you indicate that the teach points belong to a particular program in the robot. A progress bar should flash during the download similar to downloading a program.

3. Introduction to MELFA-BASIC

3.1 Instruction Reference

Command	Explanation	Example
MOV	Moves the robot to a teach-point position	MOV P1
MVS	Moves the robot to a teach-point in a straight line	MVS P1
	Moves the robot to a position above the teach-point in a straight line. (Z-Axis distance in Tool-frame)	MVS P1, -50
OVRD	Override speed limit (0 to 100%) (never use more than 30 for safety ! make this the first line of your program)	MVS 20
DLY	Delay in seconds – Robot waits	DLY 0.5
HOPEN	Opens the gripper	HOPEN 1
HCLOSE	Closes the gripper	HCLOSE 1
GOSUB	Calls a subroutine	GOSUB *PICK
RETURN	Returns from the subroutine	RETURN
DEF POS	Defines a position variable	DEF POS PTMP
END	End of program	END

3.2 Example programs:

3.2.1 Placing of an object

The example program Pr1 puts down an object at position P10:

```
OVRD   20          'set speed to 20%
MOV     P10, -50   'go within 50mm of teach-point P10
OVRD   5           'set speed to 5%
MVS     P10        'go to P10
DLY     0.5        'wait 0.5 seconds to make sure the robot stopped
HOPEN  1           'open gripper
DLY      0.5       'wait 0.5 seconds
OVRD   20          'set speed to 20%
MVS      P10, -50  'move up, to leave position P10
END                'end of program
```

Pr1: placing of an object

As shown in the program Pr1 a good program should consider the following:

- use as few teach-points as possible
- Always use an approach point before going to the target point, to ensure you are not hitting the target from the wrong direction
- use delays to let the robot settle
- set the speed limit appropriately

When using the interpolation move MVS, the robot sometimes cannot create an interpolation path and returns with an out of range error. Use MOV instead of MVS when moving between approach points.

3.2.2 Subroutines

Example program Pr2 demonstrates how to use GOSUB and return.

```
P10  =  P1                          'P10 is set to P1
GOSUB *PICKUP                       'call subroutine *PICKUP
P10  =  P2                          'P10 is set to P2
GOSUB *PLACE                        'call subroutine *PLACE
END                                 'end of program


*PICKUP                            'subroutine *PICKUP
MVS     P10,-50
…..HCLOSE 1…..                      'incomplete subroutine
RETURN                              'end of subroutine *PICKUP

*PLACE
MVS   P10,-50
….HOPEN 1……..                       'incomplete subroutine
RETURN
```

Pr2: subroutine for pick and place

3.2.3 External input

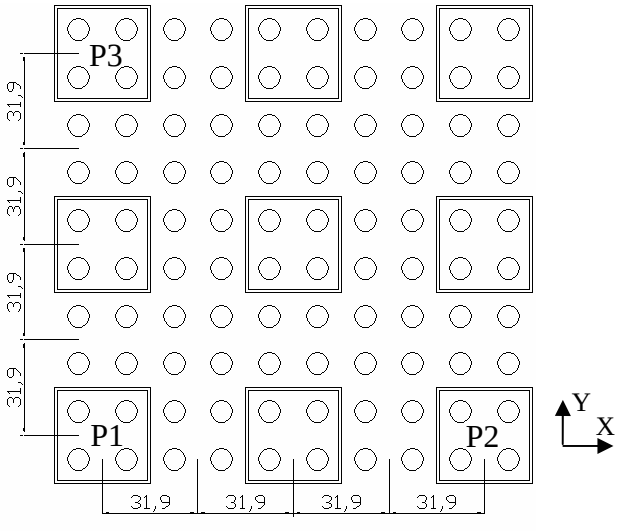
Example program Pr3 demonstrates the use of the external inputs. The robot has 16 I/O lines. For example if bit 8 is connected to a light beam that monitors a conveyor belt. If an object comes along the robot picks it up.

```
*LOOP

MOV     PREST                       'Move to a position above the conveyor belt
                                    'ready to grab object that may come along

WAIT    M_IN ( 8 ) = 0              'Waits for the input signal bit 8 to turn OFF

                                    '(light beam has been interrupted by the object)

GOSUB *PICKUP                       'calls subroutine which could pick up the object

GOSUB *PUTSOMEWHERE

GOTO *LOOP
```

Pr3: External input

3.2.4 Palletising

Example program Pr4 shows how to create a matrix/grid of target points out of 3 corner points. This is very useful if the robot has to palletise objects. It can also be used to define the grid of the Duplo board. A square Duplo brick is approximately 31.9mm x 31.9mm. The example program shows how to define a grid of 3 by 3 bricks using the corner point P1, P2 and P3 (see figure on the right)




```
DEF POS P10
P2      = P1                    'copy coordinates from P1 into P2
P3      = P1                    'copy coordinates from P1 into P3
P2.X    = P2.X + 31.9 * 4.0     'modify the copy P2 : extend the X component
P3.Y    = P3.Y + 31.9 * 4.0     'modify the copy P3 : extend the Y component
DEF PLT 1,P1,P2,P3,  ,3,3,1     'define a pallet called "PLT 1"   made of 9 points (3x3)
                                'use P1,P2 and P3 to determine the end points of the pallet

M8 = 5                          'set the integer variable M8 to 5
P10 = PLT 1,M8                  'select the fifth position in the pallet and copy it to P10

MOV   P10                       'go to that selected position
```

Pr3: Defining and using a grid of coordinates

3.3 Variables

Variables can be of different types. The most useful variable type is a position.
A position variable should start with "P". The position variables P1…P99 are defined, therefore a DEF POS command is not required for these.
Position variables can be modified. Suppose a teach-point P1 is stored in memory. To increase the x-component 10 mm use:

```
P1.X = P1.X + 10
```

To change P1, rotating around the end-effector's y-axis 0.07 radians use:

```
P1.B = P1.B - 0.07
```

4. Common Error Messages

When an error occurs the controller will beep. To recover from an error press the reset button on the teach pendant or on the robot controller. If the emergency stop button was pushed, it must be pulled out before pressing reset.

In some cases an error a code will appear on the screen. Common error codes are listed below:

Code	Meaning
L2800 – L2803	position data is inadequate. *
L2600 – L2603	position is out of range *
H0060	Emergency Stop on controller was pushed
H0070	Emergency Stop on T/B was pushed
H5000	The T/B Enable key was validated in the automatic mode.
H1010	Collision
C1350	Overload (possibly Collision)
C4340	Variable not defined (you forgot DEF POS or you forgot to download the teach-point file)

If the code is not listed please consult [2].

***** On an error message while trying to go to a pre-programmed teach point there several possible causes:

- The target position is outside the robots workspace (joint reaches the limit while trying to reach the target position). Try adjusting the target position slightly. A fraction of a millimetre can be enough.
- the teach point file (.POS) is corrupt.
  Edit the .POS file with MS Notepad (not RoboExplorer!) and confirm that the last bracket of each position is (6,0) and not (0,0)
- An interpolation or inverse kinematics error can occur if you use MVS (move linear) commands and a calculated target position rather than a learned teach-point. Also avoid angles at exactly -/+180.0°.

5. Working Envelope Limits

Joint space limits:

Joint	Limit
J1	-150° to +150°
J2	- 60° to +120°
J3	-110° to +120°
J5	- 90° to + 90°
J6	-200° to +200°

Z > 44 mm	To prevent the end effector from hitting the work surface (table)
X > - 220 mm	To prevent the robot from reaching behind

References:

[1] “INSTRUCTION MANUAL, Detailed explanations of functions and operations”, Mitsubishi Electric Corporation, Japan, 2001

[2] “INSTRUCTION MANUAL, Troubleshooting”, Mitsubishi Electric Corporation, Japan, 2001