COMPLETE OUR ONLINE SURVEY

Motoman is committed to total customer satisfaction! Please give us your feedback on the technical manuals you received with your Motoman robotic solution.

To participate, go to the following website:

TABLE OF CONTENTS

Section Page

TABLE OF CONTENTS .................................................................................i
LIST OF FIGURES ........................................................................................v
LIST OF TABLES ..........................................................................................vi

1 INTRODUCTION
1.1 MotoSim Description ........................................................................ 1-1
1.2 Features ........................................................................................... 1-1
1.3 Supported Equipment/Operation ...................................................... 1-2
1.4 System Requirements ...................................................................... 1-2
1.5 MotoSim Components ...................................................................... 1-2
1.6 Care of Components ........................................................................ 1-3
1.7 MotoSim Terminology ...................................................................... 1-3
1.8 Reference to Other Documentation.................................................. 1-4
1.9 Customer Service Information .......................................................... 1-4

2 SAFETY
2.1 Introduction ....................................................................................... 2-1
2.2 Standard Conventions ...................................................................... 2-2
2.3 General Safeguarding Tips .............................................................. 2-3
2.4 Mechanical Safety Devices .............................................................. 2-3
2.5 Installation Safety ............................................................................. 2-4
2.6 Programming Safety ......................................................................... 2-4
2.7 Operation Safety .............................................................................. 2-5
2.8 Maintenance Safety .......................................................................... 2-6

3 INSTALLATION
3.1 Installing the Hardware Key ............................................................. 3-1
3.2 Installing MotoSim ............................................................................ 3-1
3.3 Installing and Viewing Manuals ........................................................ 3-3
3.4 Installing Inovate™ (MotoSim Full Package Only) ........................... 3-3
3.5 Uninstalling MotoSim and Inovate™ ................................................ 3-4

4 CONTROL FEATURES
4.1 Menus ............................................................................................... 4-1
  4.1.1 File Menu .............................................................................. 4-1
  4.1.2 Edit Menu .............................................................................. 4-1
  4.1.3 Robot Menu .......................................................................... 4-2
  4.1.4 Camera Menu ....................................................................... 4-4
  4.1.5 Model Menu .......................................................................... 4-7
  4.1.6 Tools Menu ........................................................................... 4-8
  4.1.7 View Menu ......................................................................... 4-9
  4.1.8 Window Menu ....................................................................... 4-9
  4.1.9 Help Topics ......................................................................... 4-10
4.2 Tool Bar .......................................................................................... 4-11
4.3 Camera Bar ................................................................. 4-12
4.4 Control Dialogs .......................................................................... 4-14
  4.4.1 Programming Pendant .................................................. 4-14
  4.4.2 OLP (Off-Line Programming) ......................................... 4-15
  4.4.3 CAD Tree/Model Selection Window .............................. 4-16
  4.4.4 Add Model ................................................................. 4-16
  4.4.5 New Part .................................................................... 4-17
  4.4.6 Basic Modeler Editing Dialogs ...................................... 4-18
  4.4.7 Posture Setting for Basic Modeler ................................. 4-21
  4.4.8 Operation .................................................................... 4-22
  4.4.9 Pulse ........................................................................... 4-22
  4.4.10 Display Reference Frame ............................................ 4-23
  4.4.11 Job View ..................................................................... 4-23
  4.4.12 Job Select .................................................................... 4-24
  4.4.13 Change JOB Holder ................................................... 4-25
  4.4.14 Play Panel .................................................................... 4-25
  4.4.15 Heart Beat ..................................................................... 4-26
  4.4.16 Results ......................................................................... 4-26
  4.4.17 I/O Monitor .................................................................. 4-27
  4.4.18 Measure ....................................................................... 4-27
  4.4.19 Reach View ................................................................... 4-27
  4.4.20 Robot Information ....................................................... 4-28
  4.4.21 Trace DLG Property .................................................... 4-29
  4.4.22 Trace Property ............................................................... 4-29
  4.4.23 Camera ......................................................................... 4-30
  4.4.24 Camera Attach .............................................................. 4-31
  4.4.25 Setting Unit Dialogs .................................................... 4-29
  4.4.26 Auto Place .................................................................... 4-32
  4.4.27 Process Angle (Frame Op.) .............................................. 4-35

5 BASIC OPERATIONS
  5.1 Opening a Cell File ............................................................... 5-1
  5.2 Loading a “Previous Version” Cell ........................................ 5-1
  5.3 Creating a New Cell File ..................................................... 5-2
  5.4 Adding Robots .................................................................. 5-3
  5.5 Modeling Equipment and Production Parts ....................... 5-5
  5.6 Copying Parts .................................................................... 5-6
  5.7 Exporting and Importing AutoCAD DXF Files .................... 5-7
    5.7.1 Exporting a DXF File .................................................. 5-7
    5.7.2 Importing a DXF File .................................................. 5-7
  5.8 Creating a Robot Job in MotoSim ........................................ 5-8
    5.8.1 Programming Points Using the Programming Pendant ... 5-8
  5.9 Axis6 Models .................................................................... 5-10
    5.9.1 Automatically Creating Axis6 Job Models ................. 5-10
5.10 Job Playback and Settings ............................................................. 5-10
  5.10.1 Selecting a Job ................................................................... 5-10
  5.10.2 Playing a Job ..................................................................... 5-11
  5.10.3 Multiple Robot Job Selection .............................................. 5-11
  5.10.4 Opening a Cell File .............................................................. 5-11
  5.10.5 Adding/Deleting Robots ...................................................... 5-11
  5.10.6 Creating a Teaching Sheet ................................................. 5-11
  5.10.7 Displaying Teaching Sheet ................................................. 5-12
  5.10.8 Creating a Trace ................................................................. 5-12
  5.10.9 Displaying/Hiding Axis6 models ......................................... 5-12
  5.10.10 Changing Axis6 Length ..................................................... 5-12
  5.10.11 Calculating Cycle Time ...................................................... 5-13
  5.10.12 Displaying/Using Pulse Record Data ................................. 5-13
  5.10.13 Displaying Maximum Reach ............................................. 5-13
  5.10.14 Making Models from Jobs ............................................... 5-14
  5.10.15 Making Jobs from Models ............................................... 5-14
  5.11 Creating a Catalog List ............................................................ 5-14

6 ADVANCED OPERATIONS
  6.1 Process Angle ........................................................................... 6-1
    6.1.1 Teaching Robot Jobs Using Microsoft Excel ......................... 6-1
    6.1.2 Converting Intersection Frame to Job ................................. 6-2
    6.1.3 Converting Inform-script to Job ......................................... 6-6
    6.1.4 Copying a Job from an SK120 to a UP130 .......................... 6-11
    6.1.5 Mirroring the UP130 Job .................................................... 6-13
  6.2 Modeling Extension ................................................................... 6-14
    6.2.1 Creating Model Files from 3-D CAD Files ......................... 6-14
    6.2.2 Creating Arc Welding Tool Models .................................... 6-15
    6.2.3 Device Models for Saving/Loading .................................... 6-20
  6.3 Simulation Extension ................................................................ 6-22
    6.3.1 Non-Motion Programming ............................................... 6-22
    6.3.2 Part Motion Simulation ...................................................... 6-22
    6.3.3 Manipulator Features ....................................................... 6-28
    6.3.4 Setting Step-Motion for Off-Line Programming ................. 6-41
  6.4 External Axis ............................................................................ 6-42
    6.4.1 Modeling External Axes .................................................... 6-42
    6.4.2 Motoman Parameter All.PRM Editor ................................. 6-43
    6.4.3 Using VCON/VCOF .......................................................... 6-49
  6.5 Utilities .................................................................................... 6-50
    6.5.1 Robot Autoplacement ....................................................... 6-50
    6.5.2 Collision Detection ........................................................... 6-57
    6.5.3 Setting Up Inputs/Outputs for Multiple Robots .................. 6-59
    6.5.4 Monitoring Robot Variable Data ........................................ 6-63
    6.5.5 Changing Motion Module .................................................. 6-65
    6.5.6 Axis6 Mathematics Programming ...................................... 6-65
## LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table 5-1</td>
<td>Z-Offset for Various Motoman Robots</td>
</tr>
<tr>
<td>Table 5-2</td>
<td>Play Control Buttons</td>
</tr>
<tr>
<td>Table 6-1</td>
<td>File Import Methods</td>
</tr>
<tr>
<td>Table 6-2</td>
<td>Basic Motion Commands</td>
</tr>
<tr>
<td>Table 6-3</td>
<td>Sample Robot Job with Motion Commands</td>
</tr>
<tr>
<td>Table 6-4</td>
<td>Sample Robot Job with Part Transfer Instructions</td>
</tr>
<tr>
<td>Table 6-5</td>
<td>Reference</td>
</tr>
<tr>
<td>Table 6-6</td>
<td>Model Names for RM2 Motion</td>
</tr>
<tr>
<td>Table 6-7</td>
<td>I/O</td>
</tr>
<tr>
<td>Table 6-8</td>
<td>Sample RM2 Sweep to A Job (WITH2A.JBI)</td>
</tr>
<tr>
<td>Table 6-9</td>
<td>Sample RM2 Sweep to B Job (WITH2B.JBI)</td>
</tr>
<tr>
<td>Table 6-10</td>
<td>Sample RM2 Sweep-to-A Job</td>
</tr>
<tr>
<td>Table 6-11</td>
<td>I/O</td>
</tr>
<tr>
<td>Table 6-12</td>
<td>INF2JBI Commands</td>
</tr>
</tbody>
</table>
SECTION 1

INTRODUCTION

1.1 MotoSim Description
MotoSim (Motoman Simulation) is a graphically intensive off-line teaching system for Motoman robots using the YASNAC XRC, MRC, and ERC controllers. MotoSim is one in a family of MotoSoft™ quality software products.

1.2 Features

- **Import Capabilities** — Imports DXF, IGES, Faro Arm data, 3-D CAD, MotoSim MDL, and many other file formats readable by Inovate.
- **Optional 3-D Design Package** — Inovate is low cost, easy-to-use 3D desktop product for creating, modifying, communicating and interrogating 3D designs. It is the ideal design collaboration and communication tool from concept to production.
- **External Teach Capability** — MotoSim can be programmed using an external 6-degree-of-freedom positioning device.
- **Cycle Time Calculation** — Accurately calculates program cycle times.
- **Off-line Programming** — Mouse-click a point in space and the rendered robot moves the TCP to that location and enters the move command into the robot’s program.
- **Collision Detection** — Detects collisions between robot and other objects.
- **Programming Pendant** — Resembles Programming Pendant on the XRC. Makes off-line programming very intuitive for user.
- **Full-control Cameras** — View your robot model from any point in space. Camera controls are easy to use.
- **Automatic Job Creation** — Attaches jobs to parts instead of robots.
- **Trace** — Displays each update point as a path while the robot program executes. The points can be copied to the clipboard in a spreadsheet format.
- **OLE** — Allows other programs to share information with MotoSim.
- **Process Angle** — Adjusts series of frames (6 degree freedom model) for Robot TCP.
- **Mirror Transfer Job** — Transfers Job, mirroring location with any reference model.
- **Intersection Line** — Creates series of frames from intersection lines on robot model.
- **Teaching from Excel Spread sheet** — Teaches Robot TCP using XYZ, Rxyz spread sheet in Excel.
- **Teaching Job with Robot Holding Parts** — Supports EIMOV.
- **Symmetry TOOL** — Effective teaching technique for minimum motion while Robot Tool is symmetry type.
- **Import 3D-Studio 3DS files** — Import surface models using 3DS files directly.
- **Create JOB with Excel Spreadsheet**
1.3 **Supported Equipment/Operation**

Refer to the MotoSim Robot Library and MotoSim Tool Library addendum sheets included with your installation disks for a complete listing of robot models and other equipment supported by this release of MotoSim.

Motoman constructs robot and cell models on an ongoing basis. If your copy of MotoSim does not include model files for your Motoman-supplied equipment, contact Customer Service to check if new model files are available.

This release of MotoSim does not support spline moves or dual-robot controllers.

1.4 **System Requirements**

To run MotoSim, the following hardware and software are required:

- **Computer** IBM-compatible personal computer (PC)
- **Processor** Intel Pentium, 120 MHz minimum
- **Memory** 128 MB minimum (256 MB recommended)
- **Hard Disk** 250 MB available
- **CD-ROM Drive** Needed for installation only
- **Monitor** VGA, 800x600 resolution, 16-bit
- **Input Device**
  - Mouse
  - Six-degree-of-freedom input devices required for external programming input
- **Operating System**
  - Windows 98®, Windows NT® 4.0 or later, Service Pack 3 or higher
  - Windows 2000®, Windows XP®
- **Programs**
  - Internet Explorer 4.0 or later

**For Maximum Performance Results:**

- Install MotoSim on a 120 MHz (or faster), Pentium-based computer;
- Remember that communications software such as FDE (Floppy Disk Emulator) is required to download jobs to the controller;
- Use MotoCal to calibrate jobs before running them on the robot;
- Be familiar with PC usage and robot programming (this manual was written from the perspective that you are).

1.5 **MotoSim Components**

- Motoman CD Browser (P/N 141720-1)
- Hardware key (P/N 137215-1)
- MotoSim User’s Manual (P/N 141831-1)
1.6 Care of Components

CAUTION!
Do not lose the hardware key. Keep the hardware key in a safe place. If you lose the hardware key, MotoSim will not work and you will have to purchase a new copy of the software. If the key is accidentally damaged, return it to Motoman for replacement.

Take precautions to avoid scratching the CD’s. Always store the CD’s in their cases and avoid placing the CD’s on any surface when not in its protective case.

Keep this user’s manual in a safe place and refer to it whenever necessary. Additional copies of this manual are available from Motoman.

1.7 MotoSim Terminology

Axis6
A six-degree-of-freedom frame that can be used:
- When programming motion for non-robotic equipment in the cell.
- To define locations in the robot’s envelope when you move the robot using off-line programming.

Heart Beat
The refresh and update intervals used in the robot program simulation.

Local Coordinates
Describes an object’s position relative to its parent object.

OLP
Off-line programming.

Refresh
The frequency at which MotoSim redraws the screen display.

Robot Accuracy
How precisely the robot can move to a point in space without having been to the point previously.

Robot Repeatability
How reliably the robot can return to a previously recorded position.

Trace
Visible update points in a job that permit the user to view the path of the robot.

Teacher
An Axis6 point (or set of points) used as reference points for programming and shifting jobs.

Teaching Sheet
Displays all the programmed points on the screen, numbered and connected with straight lines to display the programmed path of the robot.

Update (time)
The frequency at which MotoSim calculates the robot’s position.

World Coordinates
Describes an object’s position relative to the world origin of the cell.
1.8 Reference to Other Documentation

For additional information refer to the following:

- Manipulator Manual for your robot model
- Application-specific Motoman manuals
- Vendor manuals for system components not manufactured by Motoman

1.9 Customer Service Information

If you are in need of technical assistance, contact the Motoman service staff at (937) 847-3200. Please have the following information ready before you call:

- MotoSim version
- Operating system (Windows 98®, Windows 2000®, Windows NT® (SP3 or higher), Windows XP®)
- System configuration (hard disk capacity, memory, software, etc.)
- Description of difficulty (make a note of any error messages)
SECTION 2
SAFETY

2.1 Introduction

It is the purchaser's responsibility to ensure that all local, county, state, and national codes, regulations, rules, or laws relating to safety and safe operating conditions for each installation are met and followed.

We suggest that you obtain and review a copy of the ANSI/RIA National Safety Standard for Industrial Robots and Robot Systems. This information can be obtained from the Robotic Industries Association by requesting ANSI/RIA R15.06. The address is as follows:

Robotic Industries Association
900 Victors Way
P.O. Box 3724
Ann Arbor, Michigan 48106
TEL: 313/994-6088
FAX: 313/994-3338

Ultimately, the best safeguard is trained personnel. The user is responsible for providing personnel who are adequately trained to operate, program, and maintain the robot cell. The robot must not be operated by personnel who have not been trained!

We recommend that all personnel who intend to operate, program, repair, or use the robot system be trained in an approved Motoman training course and become familiar with the proper operation of the system.

This safety section addresses the following:

- Standard Conventions (Section 2.2)
- General Safeguarding Tips (Section 2.3)
- Mechanical Safety Devices (Section 2.4)
- Installation Safety (Section 2.5)
- Programming Safety (Section 2.6)
- Operation Safety (Section 2.7)
- Maintenance Safety (Section 2.8)
2.2 Standard Conventions

This manual includes information essential to the safety of personnel and equipment. As you read through this manual, be alert to the four signal words:

- DANGER
- WARNING
- CAUTION
- NOTE

Pay particular attention to the information provided under these headings which are defined below (in descending order of severity).

⚠️ DANGER!

*Information appearing under the DANGER caption concerns the protection of personnel from the immediate and imminent hazards that, if not avoided, will result in immediate, serious personal injury or loss of life in addition to equipment damage.*

⚠️ WARNING!

*Information appearing under the WARNING caption concerns the protection of personnel and equipment from potential hazards that can result in personal injury or loss of life in addition to equipment damage.*

⚠️ CAUTION!

*Information appearing under the CAUTION caption concerns the protection of personnel and equipment, software, and data from hazards that can result in minor personal injury or equipment damage.*

NOTE: Information appearing in a NOTE caption provides additional information which is helpful in understanding the item being explained.
2.3 General Safeguarding Tips

All operators, programmers, plant and tooling engineers, maintenance personnel, supervisors, and anyone working near the robot must become familiar with the operation of this equipment. All personnel involved with the operation of the equipment must understand potential dangers of operation. General safeguarding tips are as follows:

- Improper operation can result in personal injury and/or damage to the equipment. Only trained personnel familiar with the operation of this robot, the operator's manuals, the system equipment, and options and accessories should be permitted to operate this robot system.

- Do not enter the robot cell while it is in automatic operation. Programmers must have the teach pendant when they enter the robot cell.

- Improper connections can damage the robot. All connections must be made within the standard voltage and current ratings of the robot I/O (Inputs and Outputs).

- The robot must be placed in Emergency Stop (E-STOP) mode whenever it is not in use.

- In accordance with ANSI/RIA R15.06, section 6.13.4 and 6.13.5, use lockout/tagout procedures during equipment maintenance. Refer also to Section 1910.147 (29CFR, Part 1910), Occupational Safety and Health Standards for General Industry (OSHA).

2.4 Mechanical Safety Devices

The safe operation of the robot, positioner, auxiliary equipment, and system is ultimately the user's responsibility. The conditions under which the equipment will be operated safely should be reviewed by the user. The user must be aware of the various national codes, ANSI/RIA R15.06 safety standards, and other local codes that may pertain to the installation and use of industrial equipment. Additional safety measures for personnel and equipment may be required depending on system installation, operation, and/or location. The following safety measures are available:

- Safety fences and barriers
- Light curtains
- Door interlocks
- Safety mats
- Floor markings
- Warning lights

Check all safety equipment frequently for proper operation. Repair or replace any non-functioning safety equipment immediately.
2.5 Installation Safety

Safe installation is essential for protection of people and equipment. The following suggestions are intended to supplement, but not replace, existing federal, local, and state laws and regulations. Additional safety measures for personnel and equipment may be required depending on system installation, operation, and/or location. Installation tips are as follows:

- Be sure that only qualified personnel familiar with national codes, local codes, and ANSI/RIA R15.06 safety standards are permitted to install the equipment.
- Identify the work envelope of each robot with floor markings, signs, and barriers.
- Position all controllers outside the robot work envelope.
- Whenever possible, install safety fences to protect against unauthorized entry into the work envelope.
- Eliminate areas where personnel might get trapped between a moving robot and other equipment (pinch points).
- Provide sufficient room inside the workcell to permit safe teaching and maintenance procedures.

2.6 Programming Safety

All operators, programmers, plant and tooling engineers, maintenance personnel, supervisors, and anyone working near the robot must become familiar with the operation of this equipment. All personnel involved with the operation of the equipment must understand potential dangers of operation. Programming tips are as follows:

- Any modifications to PART 1 of the MRC controller PLC can cause severe personal injury or death, as well as damage to the robot! Do not make any modifications to PART 1. Making any changes without the written permission of Motoman will VOID YOUR WARRANTY!
- Some operations require standard passwords and some require special passwords. Special passwords are for Motoman use only. YOUR WARRANTY WILL BE VOID if you use these special passwords.
- Back up all programs and jobs onto a floppy disk whenever program changes are made. To avoid loss of information, programs, or jobs, a backup must always be made before any service procedures are done and before any changes are made to options, accessories, or equipment.
- The concurrent I/O (Input and Output) function allows the customer to modify the internal ladder inputs and outputs for maximum robot performance. Great care must be taken when making these modifications. Double-check all modifications under every mode of robot operation to ensure that you have not created hazards or dangerous situations that may damage the robot or other parts of the system.
- Improper operation can result in personal injury and/or damage to the equipment. Only trained personnel familiar with the operation, manuals, electrical design, and equipment interconnections of this robot should be permitted to operate the system.
- Inspect the robot and work envelope to be sure no potentially hazardous conditions exist. Be sure the area is clean and free of water, oil, debris, etc.
- Be sure that all safeguards are in place.
- Check the E-STOP button on the teach pendant for proper operation before programming.
- Carry the teach pendant with you when you enter the workcell.
- Be sure that only the person holding the teach pendant enters the workcell.
- Test any new or modified program at low speed for at least one full cycle.

2.7 Operation Safety

All operators, programmers, plant and tooling engineers, maintenance personnel, supervisors, and anyone working near the robot must become familiar with the operation of this equipment. All personnel involved with the operation of the equipment must understand potential dangers of operation. Operation tips are as follows:

- Be sure that only trained personnel familiar with the operation of this robot, the operator's manuals, the system equipment, and options and accessories are permitted to operate this robot system.
- Check all safety equipment for proper operation. Repair or replace any non-functioning safety equipment immediately.
- Inspect the robot and work envelope to ensure no potentially hazardous conditions exist. Be sure the area is clean and free of water, oil, debris, etc.
- Ensure that all safeguards are in place.
- Improper operation can result in personal injury and/or damage to the equipment. Only trained personnel familiar with the operation, manuals, electrical design, and equipment interconnections of this robot should be permitted to operate the system.
- Do not enter the robot cell while it is in automatic operation. Programmers must have the teach pendant when they enter the cell.
- The robot must be placed in Emergency Stop (E-STOP) mode whenever it is not in use.
- This equipment has multiple sources of electrical supply. Electrical interconnections are made between the controller, external servo box, and other equipment. Disconnect and lockout/tagout all electrical circuits before making any modifications or connections.
- All modifications made to the controller will change the way the robot operates and can cause severe personal injury or death, as well as damage the robot. This includes controller parameters, ladder parts 1 and 2, and I/O (Input and Output) modifications. Check and test all changes at slow speed.
2.8 Maintenance Safety

All operators, programmers, plant and tooling engineers, maintenance personnel, supervisors, and anyone working near the robot must become familiar with the operation of this equipment. All personnel involved with the operation of the equipment must understand potential dangers of operation. Maintenance tips are as follows:

- Do not perform any maintenance procedures before reading and understanding the proper procedures in the appropriate manual.
- Check all safety equipment for proper operation. Repair or replace any non-functioning safety equipment immediately.
- Improper operation can result in personal injury and/or damage to the equipment. Only trained personnel familiar with the operation, manuals, electrical design, and equipment interconnections of this robot should be permitted to operate the system.
- Back up all your programs and jobs onto a floppy disk whenever program changes are made. A backup must always be made before any servicing or changes are made to options, accessories, or equipment to avoid loss of information, programs, or jobs.
- Do not enter the robot cell while it is in automatic operation. Programmers must have the teach pendant when they enter the cell.
- The robot must be placed in Emergency Stop (E-STOP) mode whenever it is not in use.
- Be sure all safeguards are in place.
- Use proper replacement parts.
- This equipment has multiple sources of electrical supply. Electrical interconnections are made between the controller, external servo box, and other equipment. Disconnect and lockout/tagout all electrical circuits before making any modifications or connections.
- All modifications made to the controller will change the way the robot operates and can cause severe personal injury or death, as well as damage the robot. This includes controller parameters, ladder parts 1 and 2, and I/O (Input and Output) modifications. Check and test all changes at slow speed.
- Improper connections can damage the robot. All connections must be made within the standard voltage and current ratings of the robot I/O (Inputs and Outputs).
SECTION 3
INSTALLATION

3.1 Installing the Hardware Key

The hardware key supplied with MotoSim must be installed on your computer or MotoSim will not function properly. The hardware key attaches to the computer’s parallel port. This port is commonly used to connect printers and other peripheral devices to your computer. To attach the hardware key:

1. Disconnect any device currently connected to your computer’s parallel port.
2. Carefully insert the hardware key into the parallel port. If the key does not fit, do not force it. The key should fit snugly but does not require significant force to insert.
3. Connect your peripheral cable to the free end of the hardware key. The key will not interfere with the operation of your printer or other peripheral devices.

If you are using two or more MotoSoft™ products that require the use of a hardware key, you can “stack” the keys (connecting them in series).

3.2 Installing MotoSim

The MotoSim installation files are located on the Motoman CD Browser (P/N 141720-1). Please refer to the CD-Browser for detailed installation instructions.

MotoSim application files are installed in c:\Program Files\Motoman\MotoSim. To install MotoSim, proceed as follows:

1. Insert the CD-ROM Browser into the CD-ROM drive on the computer.
2. The setup program begins automatically and the introduction screen appears.

Note: Setup executes automatically when the CD is inserted into the CD-ROM drive unless autoexecute has been disabled on your computer. If the setup program does not autoexecute, select demo32.exe from the CD-ROM Open Properties dialog. Demoshield Player appears and prompts you for a .dbd file. Double-click *.dbd to begin the install process.

![MotoSim CD Browser](image)
3. Click the button marked Simulation Software. The list of Motoman simulation software appears.

4. Select the MotoSim package you wish to install.

NOTE: You must have a hardware security key installed for each package you purchased. If the hardware key is not installed, you will not be able to run the software you purchased.

5. Follow the install instructions as you are guided through the installation process.
3.3 Installing and Viewing Manuals

An electronic version of this manual is included on the install disk. Adobe® Acrobat® Reader¹ is required to view the files. If your computer already has this program installed, you may skip to Step 4. If your computer does not have Acrobat® Reader installed, proceed as follows:

1. Click Utilities.
2. Click Acrobat Reader.

NOTE: An internet connection is required to download this software.

3. Follow the online directions to download and install this software.
4. To view the MotoSim manual, you will need to browse to the file on the Motoman CD Browser.
5. Double-click the My Computer icon on your desktop.
6. Right-click the CD-ROM icon for the drive containing the Motoman CD Browser and select Open from the dropdown menu.
7. Double-click the Installs folder.
8. Double-click the desired software package.
10. Open the Acrobat .pdf file labeled 141831-1.

NOTE: If you would like to install a manual on your computer for later viewing, simply copy the file to a directory on your hard drive.

3.4 Installing Inovate™ (MotoSim Full Package Only)

The Inovate™ option allows users to create and modify advanced models. MotoSim and Inovate™ use OLE (Object Linking and Embedding) to work together seamlessly. Inovate™ also enables MotoSim users to import IGES, DXF, STEP, STL, SLA, and 3DS files.

NOTE: Inovate™ must be installed if you wish to create advanced models for use in MotoSim.

Consult the Visionary Design Systems documentation included with MotoSim (full package) for instructions on installing and using Inovate™.

¹. Adobe® and Acrobat® are registered trademarks of Adobe Systems Incorporated.
3.5 **Uninstalling MotoSim and Inovate™**

In the event you wish to remove MotoSim, or Inovate™ software from your computer, do not simply delete the program folders. Doing so will leave unwanted files in your Windows directory. Use the Add/Remove Programs feature in Windows to ensure a complete removal of the software components from your hard disk.

1. Select Settings Control Panel from the Start menu.
2. Double-click *Add/Remove Programs*.
3. Select the software to be uninstalled.
4. Follow the prompts to remove the software.
4.1 Menus

This section contains an overview of all menu commands.

4.1.1 File Menu

New
Creates a new cell project file.

Open…
Opens existing cell project files.

Close
Closes the currently opened cell project file.

Save
Saves the opened cell project file.

Save As…
Saves the opened cell project file as another file with a different name.

[Recent cell files]
After one or more cell project files have been created, the names and paths of recently opened files will appear in the File menu beneath the “Print Setup” command. Select the desired cell project to open the file.

Edit Cel…
Allows text-level editing of cell files.

NOTE: You can use MRC Editor, XRC Job Editor or Notepad to edit cell and job files.

Relative Path
When this option is active, robot and part models are created relative to the location of the cell file. When this option is inactive, files will be saved to default locations in the MotoSim directory.

Load Cel…
Loads the contents of a cell file into the open cell project file. This feature is convenient for copying cell setups from an existing cell file to a new cell file. It also upgrades a .cel file to an .mcl file.

Exit
Closes the opened cell file, prompts you to save changes (if necessary), and exits the MotoSim program.

4.1.2 Edit Menu

Edit Model…
Opens an edit dialog for selected basic models and open Inovate for selected advanced models.

Cut, Copy, Paste, Find, and Delete
Basic operations will be applied to selected models.
4.1.3 **Robot Menu**

**Controller**
Contains commands to select robot jobs, change the heart beat, and get offset information on the current robot.

*Select JOB*
Selects different robot job (.jbi) files.

*Select Holder JOB...*
Selects different holder robot job (.jbi) files.

**Change Robot**
Switches between different robots if more than one robot is in the cell.

**Heart Beat**
Changes the refresh and update interval MotoSim uses to display robot motion. When selected, “Heart Beat” opens a dialog that allows you to select Detail Mode or Skip Mode. Detail Mode refreshes the display frequently making the robot motion appear smoother. Skip Mode causes the simulation to skip screen updates to make the program run faster. Skip Mode is usually used when MotoSim is running on a slow PC.

**Information**
Provides information on the current robot being used by MotoSim and allows you to set robot offset and controller start-up time.

**PlayBack**
Playback functions allow you to control the job program being played.

*Cycle Time...*
Runs the presently loaded job and calculates the job cycle time.

*Quick Cycle Time Cal...*
Calculates job cycle time without running the MotoSim simulation. Quick Cycle Time Cal... is just as accurate as Cycle Time... but is much faster when calculating cycle times for large jobs or when running MotoSim on a slow computer.

**Start**
Starts the robot’s job.

**Hold**
Stops the program in progress and holds at the current program step.

**Pause**
Pauses the program in progress. If the current step is not complete when the pause command is issued, the job will stop in mid-step.

**Step Next**
Steps the program forward through the job.

**Step Back**
Steps the program backward through the job.

**NOTE:** Variables set at the beginning of a job will be ignored if you step through a job. Variables are only recognized when playing a job using the Start command.
Reset Job
Stops all program execution and resets the job back to “NOP.”

Pulse Record…
Records and displays registered encoder pulse locations as the robot moves through the selected job.

Modify Data
Adjusts tool and robot user frames.

Tool Data…
Adjusts the tool center point (TCP) of the tool. Figures are in millimeters. The TCP on screen will move after the new tool data is entered.

Tool Property
Adjusts the tool data including whole tool data (Mass...)

Set User Frame…
Makes modifications to the robot’s user frame.

User Frame Property…
Makes modifications to the user frame properties.

I/O Monitor…
Displays the current status of I/O within the job. Also allows you to manually turn I/O on and off for testing purposes and to reset I/O.

Teaching Sheet
Create
Creates the Teaching Sheet for the current job.

Delete
Deletes the Teaching Sheet from the display.

Break
Cancels the Teaching Sheet operation if I/O logic problems cause the simulation to stall.

Make Axis6 Model
Creates Axis6 models from the active job.

Color
Changes the color of the Teaching Sheet path.

Move Dependent
Color codes the program path by the type of move. For example, linear moves are drawn in a different color than joint moves.

User Defined
Sets the color of the standard Teaching Sheet display. Use this command to set the Teaching Sheet to a different color that is easier to see against the color of the robot and background.

With Motion
Does the same thing as “Create” but moves the robot to each point as it generates the teaching sheet display.
Maximize
Enlarges or reduces the screen view to best accommodate the complete extent of the Teaching Sheet display.

Center
Centers the Teaching Sheet in the center of the screen.

Trace
Contains commands for creation and deletion of robot trace models.

Clipboard…
 Copies the trace data to the clipboard so that you can paste the data into a spreadsheet or data analysis program. The copied values represent the pulse counts of each point in the trace.

Clear
Clears the trace model.

Property…
Opens the Trace Property dialog which enables you to set options related to the trace model such as color, detail level, etc.

OLP
Controls Off-line Programming options.

To Point…
When active, this option forces the TCP to match the orientation of the selected Axis6 point.

Reach View…
Open the Reach View dialog which provides options for displaying the maximum reach of the TCP or a specified robot link.

Add Robot…
Opens the Add Robot dialog for addition of robots to the cell.

Delete Robot…
Used to delete robot models from the cell.

4.1.4 Camera Menu

Camera Scope
Contains commands that change how the mouse controls camera motion, position, lighting, and model handling.

Scroll & Zoom
Rotates around the cell or increases/decreases magnification. Move the mouse right or left to rotate the cell. Move the mouse up or down to pull out or zoom in, respectively.

Free Zoom
Enables the Free Zoom mode allowing you to enlarge objects by encircling them with the cursor.
**Rotate XY**
Rotates the cell view about the X- and Y-axes. Clicking and dragging the mouse across the screen rotates the model about whichever axis is aligned vertically. Clicking and dragging the mouse up and down rotates the model about whichever axis runs across the screen.

**Sliding**
Click and drag the cell across the screen to view adjacent areas of the cell while working at high magnification.

**Center**
Moves clicked object to the center of the screen.

**Flying Camera**
Click and drag the cursor on the screen in the direction you want the camera to fly. Click and drag the mouse right or left to move in the those directions. Click and drag the mouse up or down to fly out or fly in, respectively.

**Position**
Activates camera motion controls.

**Lighting**
Places MotoSim in the lighting mode which allows you to control direction and intensity of lighting. See the Camera Bar descriptions for detailed usage instructions.

**Direction**
Controls the direction from which the MotoSim model is displayed by the computer.

- **View**
  Resets the view direction.

- **Side**
  Side view of the model.

- **Top**
  Top view of the model.

- **Front**
  Front view of the model.

**NORM**
“Normalizes” the current view, straightening/leveling it to make viewing or further movement easier.

**NORM:INIT**
Returns the screen to the default 3-D view.

**Back Color**
Controls the color of the background in the MotoSim simulation window.

**Camera**

- **Configuration**
  Opens the Camera configuration dialog.

- **Surface Mode**
  Displays the models as solid objects. If Surface Mode is turned off, models are displayed as wireframes.
**CONTROL FEATURES**

*Set Unit of Object…*
Contains options for setting units of measurement displayed in measurement and result dialogs.

*Over Draw Wireframe*
Displays wireframe line over top of the solid model but with hidden lines removed.

**Model**

*Select Model*
Selects in the CAD Tree window parts or models which you click with the mouse. This helps you determine what part of the model you are editing when models are comprised of many parts.

*Shift*
Allows you to drag an object directly in the simulation window.

*Rotate*
Allows you to rotate an object directly in the simulation window.

*Drag in Z*
Allows you to click and drag an object to rotate it about its Z-axis.

*With Wireframe*
Displays the model with its wireframe.

**Line Width**

*Pen=1,2,3*
Controls the thickness of the Axis6 indicators. Set to a thicker value (3) when using large, high-resolution displays on which a fine line may be hard to see.

**Frame Length…**
Controls the length of the Axis6 frames displayed in the MotoSim window.

**Mouse Picking**
Controls how the OLP mouse picking will select objects when you click on them.

*Free*
Moves the robot to the point clicked on the model.

*Vertex*
Moves the robot to the edge where two objects meet.

*Center*
Moves the robot to the center of the face clicked.

**Measure**
Measures the distance between two or more points. Click on the two points and MotoSim will display a dialog box showing the distance between the points, the relative position of the second point, and a box that allows you to change the number of points sampled before the measurement is displayed (2 is the default value).
4.1.5  **Model Menu**

**New Model**
Opens the Add Part dialog and sets the default modeling tool to the basic modeler in MotoSim. Subsequent basic parts can be created by clicking the New button in the CAD Tree window.

**Edit Model…**
Opens an edit dialog for selected basic models and open Inovate™ for selected advanced models.

**Hide/See**
Allows you to Hide/Display models, frames, names, and wire view or any combination of the same.

**Model Attribute**

**Set Parent…**
Changes a model’s parent without moving the model. After a parent change, position values will change to reflect the position relative to the new parent.

**Move Parent…**
Changes a model’s parent and moves the model to the same local coordinates used with the former parent. This will cause the model to move in the cell but retain the same position relative to its parent.

**Set File Path…**
Opens the Model Load dialog which allows you to specify a directory other than the default directory for model files. By default, MotoSim will place model files in the same directory with the cell file unless specified otherwise in the Model Load dialog.

**Rename…**
Allows you to rename an object.

**Set Position…**
Allows you to change part position relative to the part’s parent.

**Set I/O Object…**
Opens the I/O Monitor panel

**Property…**
Contains four options for model display:

**Opacity**
Allow you to set the opacity of the selected model. By decreasing the opacity value, the model becomes increasingly transparent, allowing you to see objects otherwise obscured from view.
Recursible Call for Model Tree
When this option is active, all opacity changes to an object carry to all child parts. If this option is inactive, opacity changes will only affect selected objects.

Model I/O
Allows you to define the I/O information for the selected model.

Type
Allows you to define the category of the selected model.

Model Tree View

Refresh View
Causes the display to update changes made in the CAD Tree window.

Expand Tree
Expands the selected tree.

AutoCAD Support

AutoCAD Export…
Exports the selected model to AutoCAD DXF.

AutoCAD Import…
Imports an AutoCAD DXF file.

Models List…
Operates models to delete, or change a parent.

Import 3D Model…
Imports 3D models (*.rwz, *.3ds).

Relocate Models…
Relocates model location from a parent using AXIS6 type shape data.

4.1.6 Tools Menu

CAD Tool

CAD
Opens the CAD Tree window for creation or editing of basic models.

TriSpectives Resident
Opens the TriSpectives modeler for creation and edition of advanced models.

Robot

T-Pendant
Opens the Programming Pendant dialog

Position and Job View
Opens the Pulse and Job View windows.

Play Panel
Opens the Play Panel dialog.

Auto Place…
Automatically places robots in the optimum location(s) for working on a part.
**StageMaster…**
Opens the StageMaster dialog.

**Servo**
Contains one menu command for servo configuration.

**Configuration…**
Opens the Servo Configuration dialog which has options for servo settings, etc.

### 4.1.7 View Menu

**Toolbar**
Alternately displays and hides the tool bar.

**Status Bar**
Alternately displays and hides the status bar.

**Robot Bar**
Alternately displays and hides the robot bar.

**Camera Bar**
Alternately displays and hides the camera bar.

**Logo Bar**
Alternately displays and hides the Motoman logo bar.

### 4.1.8 Window Menu

**New Window**
Opens a new window with no models.

**Cascade**
Arranges multiple, open windows so that the title bars are visible for easy switching between windows.

**Tile**
Arranges multiple, open windows horizontally, one above the other.

**Arrange Icons**
Neatly arranges the icons of minimized windows along the bottom of the screen.

**Split**
shows split window.

**[open cells]**
A list of all opened cel files. Select items from this list to bring the respective cel file window to the front.
4.1.9 Help Topics

Help...
Opens a standard Windows Help file designed for MotoSim.

About MotoSim
Opens the About MotoSim dialog containing version, copyright, memory statistics, and system information.

HINT: Nearly all of the commands above can be accessed by clicking in the window or on a model with the right mouse button. Right-button clicking will create a floating menu from which you can select various commands described above.

NOTE: Use caution when editing with the right-mouse-button commands. If using OLP Pick to select objects in the simulation window, you may unknowingly select an object whose visibility attribute is off and is located in front of the desired object. You could end up editing or deleting the wrong object. When in doubt, select models from the CAD Tree window and carefully read all warning pop-ups before clicking OK.
4.2 **Tool Bar**

The following buttons/pull down menus appear on MotoSim’s tool bar. Many of the tool bar commands are the same as certain menu commands. Tool bar commands are described below.

- **New Project**
  Creates a new cell project file.

- **Open...**
  Opens existing cell project files.

- **Save**
  Save the open cell project file.

- **Robot Bar**
  Allows you to select a robot.

- **Quick Cycle Time**
  Runs the robot job in quick cycle time mode.

- **Detail Mode**
  Refreshes the display frequently making the robot motion appear smoother.

- **Pulse Record**
  Displays a window that contains a “transcript” of every robot position used to perform the current job.

- **Reset Job**
  Resets the present job to the first step.

- **Start**
  Starts the currently loaded job.

- **Pause**
  Stops the running program after the current step has been completed. The program will remain at the present step until the program play is resumed or the job is reset.

- **Hold**
  Stops the currently loaded job.

- **Back Step**
  Moves the robot to the previous step in the program. This button does not function when the program is at the first step.

- **Next Step**
  Moves the robot to the next step in the program. This button does not function when the program is at the last step.

- **Select Job**
  Opens the previously saved or imported job files.

- **Programming Pendant**
  Displays MotoSim’s programming pendant.

- **Position and Job View**
  Displays the current position of the robot in pulse counts and a window showing the current job.

- **OLP**
  Opens the OLP dialog and moves the robot to different positions when you click on points on or about the cell.
CONTROL FEATURES

I/O Monitor
Opens the I/O Monitor panel.

Auto Place
Places the selected robot in the optimum location for working on surrounding work pieces.

Change JOB Holder
JOB Holders allow you to organize your jobs and model files into various combinations without having to move or duplicate files. Click this button to add a file to a new JOB Holder or change the JOB Holder.

4.3 Camera Bar

The following buttons appear on MotoSim’s camera bar. Many of the camera bar commands are the same as some Camera menu commands. Camera bar commands are described below.

Scroll & Zoom
Rotates around the cell or increases/decreases magnification. Move the mouse right or left to rotate the cell. Move the mouse up or down to pull out or zoom in, respectively.

Free Zoom
Enables the Free Zoom mode which allows you to enlarge objects by encircling them with the cursor.

Rotate XY
Rotates about the X- and Y-axes of the cell. Clicking and dragging the mouse across the screen rotates the model about whichever axis is aligned vertically. Clicking and dragging the mouse up and down rotates the model about whichever axis runs across the screen.

Sliding
Enables user to slide the camera up, down, and sideways in the same plane to view different parts of the cell.

Center
Positions the selected model to the center of the screen.

Measure
Enables the Measure tool described in Section 4.1.4.

Select Model
This feature makes editing objects easier by allowing you to find and highlight an object in the CAD Tree window by simply clicking on the object in the MotoSim window. The CAD window must be open for this feature to work.

Lighting
Enables the lighting control. Mouse movements change the direction and intensity of the lighting.

View Frames
Toggles display of the Axis6 frame indicators.

Direction X
Changes the direction of screen sliding on wheel-equipped mice. Clicking this button will cause the screen view to slide sideways when the mouse wheel is used.

Drag
Moves objects out of the way for easier viewing of hidden objects. Moves made with this tool are not permanent.
Spin
Works much like "Rotate XY" but only rotates the selected object in the cell rather than the whole cell view.

Flying Camera
Invokes Flying Camera mode. Click and drag the cursor on the screen in the direction you want the camera to fly. Click and drag the mouse right or left to move in those directions. Click and drag the mouse up or down to fly out or fly in, respectively.

Position
Allows the user to "fly" through and around the robot cell.

Panorama
Toggles the cell display between flat and 3-D perspective.

CAD
Opens the CAD Tree window to view the structure of the cell model and make changes to model components.

Collision Detect
Opens the Collision Free dialog which allows you to select objects in the cell that may collide with each other. When collision detection is active, any object that collides with another will turn red. Correct the robot position or path to prevent the collision.

Configuration
Opens the Camera settings box. From this box, you can specify camera position and motion, and lighting intensity.

Normal View
Returns the cell view to a normal orientation after using Rotate XY or other view tools to change the view perspective. This feature also centers the TCP in the middle of the screen.

Initial Scale
Reverts the view to the default magnification and centers the cell in the screen.

Top View
Changes the current view to a standard top view.

Front View
Switches the current view to the default front view.

Side View
Switches the current view to the default side view.

Isometric View
Changes the current view to a standard isometric view.
4.4 **Control Dialogs**

This section contains descriptions and basic instructions for MotoSim’s most commonly used or complicated dialog boxes.

4.4.1 **Programming Pendant**

The Programming Pendant represents the robot programming pendant. Its features are easy to recognize for users already familiar with programming the Motoman controller.

**Joint, Wld, Tool, User**
These controls correspond to the coordinate buttons on the real programming pendant.

**Ext**
External Axis control. Use selector above this button to select desired external axis.

**Job window**
Displays the currently loaded job.

**Fast, Med, Slow**
These buttons control teaching speed. Used for making fine adjustments to the robot before programming a point. Not used with OLP.

**Sync**
When this option is checked, the robot will move to position when the corresponding step is selected in the job view window.

**Motion keys**
These keys will vary depending upon which coordinate system you are using. “Joint” will display S, L, U, R, B, T keys. All other coordinate systems will display X, Y, Z, Rx, Ry, Rz keys. External axis displays E1, E2, E3, E4, E5, E6.

**Move…**
Sets move type, speed, and position level.

**Edit Functions**
The Add, Delete, and Modify buttons are used to edit the program.

**Close**
Closes the programming pendant dialog.
4.4.2 OLP (Off-Line Programming)

Contains options for off-line job programming. OLP controls the type of objects that are selected and how they are selected. It also controls movement of external axes and objects.

Pick Type
Determines how MotoSim will pick points. “Free” causes the robot to move to the surface you click on. “Vertex” causes the robot to move only to intersections or edges of surfaces. “Center” causes the robot to move to the center of the part surface.

Pose
Causes the robot to move with posture.

Symmetry
Causes the robot to move with minimum Rz motion of the tool.

Pick Object
Pick object determines when OLP will be used to select parts, Axis6 frames, or both.

OLP Pick
OLP works only when this option is checked. Zooming, scrolling, and other camera operations that use mouse input require this option to be disabled until you are finished changing views.

Normal
Once OLP Pick is turned on and a point has been selected, turning on “Normal” causes the TCP to change its alignment to the part. The TCP will remain in the same location but the tool will rotate about the point to make the Z-axis perpendicular to the face of the object selected. The surface does not need an Axis6 for this feature to work.

Operation Object
Changes the programming pendant to control the robot, a teaching point, or a selected model, teacher model, EIMOV Target.

XYZ
Opens the Position Data (Pulse) dialog to allow moving of robot or objects by pulse counts.

EI Move
When this option is active, use the Position Data dialog or programming pendant to control the external axes. Select the reference model to be EIMoved.

Model…
Gives the name of the part or model which EIMOV is referencing.

Other Solution
Opens the Pulse Rec dialog and provides four possible solutions for orientation about the specified point.

All Pose
Opens the Pulse Rec dialog and provides all* possible solutions for orientation about the specified point. (* Up to the maximum specified in the box next to the button.)
**Symmetry**
Specifies the degree increment used to calculate additional solutions for Other Solution and All Pose.

**Daxis6**
Provides data on the selected Axis6 or model.

**Undo**
Places the robot in the previous posture.

### 4.4.3 CAD Tree/Model Selection Window

The CAD Tree window (also known as the Model Selection window when adding models) displays the complete structure of the cell file. From this window, you build and modify models used in your cell. It also allows you to control which models are visible on the screen.

Models are represented differently in the CAD Tree window.

Right click the selected model to bring up the “Model menu”.

![Basic Model](image)
Basic models are depicted with this icon in the CAD Tree.

![ Dummy Model](image)
Dummy models are depicted as dimmed Basic models in the CAD Tree.

![Robot](image)
Robot models are depicted with this icon in the CAD Tree.

![3-D CAD Model](image)
Models created in TriSpectives are depicted with this icon in the CAD Tree.

By default, the CAD Tree window displays models as shown above. If you select Model Hide/See Status from the View menu, additional icons will appear to depict the status of each model in the CAD Tree.

### 4.4.4 Add Model

This dialog is used to add new models to the cell file. You can also create dummy models to use as an invisible parent for other models to simplify the task of moving complex models.

**Name**
Enter the plain-text name of the model here.

**File Name**
Opens an existing model file or change the name of the default filename created from the plain-text name before saving the model name. The “…” button allows you to browse for a model file.
Dummy Model
Creates a dummy model with no shape or size. It is used to define position and orientation only. It is used as a parent upon which you can build complex models. Building a dummy model and then attaching other parts allows you to move groups of parts by moving the dummy model only.

External CAD
Checking this option will cause MotoSim to open Inovate™ to create an advanced model instead of the standard MotoSim basic modeler. Advanced modeler environment will be opened when you click OK.

4.4.5 New Part
This dialog appears when you select Edit Model... from the Model menu. This dialog allows you to create and edit the size, type, color, and position of parts relative to the parent object.

Add Parts
Select the desired part type from the dropdown list and click Add to create the part.

Add
Creates whichever part is currently displayed in the Add Parts dropdown list. New parts will appear in the parts list in the lower left-hand corner of this dialog.

Edit
Changes attributes and position of the selected part.

Delete
Deletes the selected part.

Move Org
Redefines the origin frame of the part. This may be used for controlling the tool model.

Name
Sets a unique name for each shape.

Clipboard...
Exchanges data between programs (i.e. Excel data sheets).

Copy to
Copy X,Y,Z, Rx,Ry,Rz data to clipboard. From here it can be pasted into other applications.

Paste from
Copy X,Y,Z, Rx,Ry,Rz data from clipboard. Data can be copied from Excel spread sheet.
4.4.6 Basic Modeler Editing Dialogs

Each time you open the New Part dialog from the CAD Tree window, a window at the top of the dialog will offer several part type options. When you select a part type and click Add or Edit, a different Edit box will appear depending on which part type you selected. The following dialogs control the size, position, color, and other aspects of the part’s appearance.

Box Edit
Used to create and edit basic boxes.

Size Controls
Used to set the width, depth, and height of the box.
Highlight the desired data field and enter in the desired value using the numeric keys on the keyboard.

Press Tab to switch between data boxes.

Speed
Controls how much the size changes each time an arrow button is clicked. To make large changes, set speed to FAST. To make finer adjustments, set the speed to SLOW.

Color…
Changes the color of the box.

Posture…
Position Setting. See Section 4.4.7.

Cone Edit
Used to create and edit conical objects.

NOTE: This dialog is titled “Cone 2 Edit” to differentiate it from “Cone Edit,” a model type used in earlier versions of MotoSim.

Size Controls
Used to set the base diameter and height of the cone.

Number
Controls how smoothly the shading of the cone is shown on the computer screen. Higher numbers will produce smoother surfaces but draw time will increase. Use a lower number when simulation speed is more important than model appearance or when using a slow computer.

Speed
Controls how much the size changes each time an arrow button is clicked. To make large changes, set speed to FAST. To make finer adjustments, set the speed to SLOW.

Color…
Changes the color of the cone.

Posture…
Position Setting. See Section 4.4.7.
Cube Edit
Used to create and edit extruded shapes from points in space. Enter all but the last point with a Z position of zero. If only one point is inserted, nothing will be shown on the screen. When the second point is entered, a line appears connecting the two points. When you add a third point, the line will turn into a plane. When you add the last point, leave the X and Y values at zero and specify a Z-position. This will set the thickness of the extrusion. Additional points added before the last point will further define the part’s shape.

Index
Displays a listing of all the points inserted.

Synch. Teacher
Moves the teacher to the currently selected Index point.

Show All
Shows all element data.

Insert
Enters the values in the Position box into a new index value shown in the Index window.

Delete
Deletes the selected point from the Index window.

Color…
Changes the color of the model.

Posture…
Position Setting. See Section 4.4.7.

Cylinder Edit
Used to create and edit cylinders or conical objects.

Size Controls
Used to set the upper/lower diameter and height of the cylinder.

Number
Controls how smoothly the shading of the cylinder is shown on the computer screen. Higher numbers will produce smoother surfaces but draw time will increase. Use a lower number when simulation speed is more important than model appearance or when using a slow computer.

Speed
Controls how much the size changes each time an arrow button is clicked. To make large changes, set speed to FAST. To make finer adjustments, set the speed to SLOW.

Straight
Makes the upper and lower diameter controls identical thus creating a cylinder. Turn this option off if you want to make a conical object.

Color…
Changes the color of the cylinder.

Posture…
Position Setting. See Section 4.4.7.
**Floor Edit**
Used to create and edit a floor model in an empty cell file or used to edit an existing floor’s size and color.

**Size Controls**
Controls the dimensions of the floor. X(mm) and Y(mm) control the dimensions of the floor.

**Number**
The X and Y values control how many tiles are displayed along each respective dimension. Experiment with different settings until the desired floor pattern is achieved. The default value for both X and Y is 8. If both X and Y are set to 1, the floor will appear as one solid color.

**Speed**
Controls how much the size changes each time an arrow button is clicked. To make large changes, set speed to FAST. To make finer adjustments, set the speed to SLOW.

**Center**
Checking this option will make the floor center on the world origin. Leaving this option turned off will cause the floor to be positioned with the world origin on the midpoint of one edge of the floor.

**Color**
Changes the color of the floor.

**Axis6 Edit**
This edit dialog is used to create and edit Axis6 points.

**Index**
Displays the numbers of all Axis6 points already created.

**Synch. Teacher**
Moves the teacher to the currently selected Index point.

**Show All**
Shows all elements of part.

**Insert**
Enter the values in the Position box into a new index value shown in the Index window. Each point entered alters the shape of the part.

**Delete**
Deletes the selected point from the Index window.

**Color...**
Changes the color of the shape.

**Posture...**
Position Setting (see Section 4.4.7).

**Frame Op.**
Frame operation (see Section 4.4.8).
Pipe Edit
Used to create and edit Pipe (hollow cylinder) objects.

NOTE: This dialog is titled “Pipe 2 Edit” to differentiate it from “Pipe Edit,” a model type used in earlier versions of MotoSim.

Size Controls
Used to set the upper/lower diameter, upper/lower thickness, and height of the pipe.

Number
Controls how smoothly the shading of the pipe is shown on the computer screen. Higher numbers will produce smoother surfaces but draw time will increase. Use a lower number when simulation speed is more important than model appearance or when using a slow computer.

Speed
Controls how much the size changes each time an arrow button is clicked. To make large changes, set speed to FAST. To make finer adjustments, set the speed to SLOW.

Straight
Makes the upper and lower diameter controls identical, thus creating a straight pipe. Turn this option off if you want to make a pipe with unequal end diameters.

Color…
Changes the color of the pipe.

Posture…
Position Setting. See Section 4.4.7.

4.4.7 Posture Setting for Basic Modeler
The following dialog is used in most basic modeler dialogs to set the position of a new or existing model part relative to the world or the part’s parent.

Posture Setting
This function is abbreviated in most dialogs as Posture and is used to set the posture of the selected model part. “Absolute position” refers to the model’s coordinates relative to the world origin.

Pxyz (mm)
Changes X-, Y-, and Z-axes positions in millimeters.

Rxyz (deg)
Changes the X-, Y-, and Z-axes rotation in one-degree increments.

Speed
Controls how much the position changes each time an arrow button is clicked. To move the part a long distance, set speed to FAST. To make finer position or size adjustments, set the speed to SLOW.
4.4.8 Operation

Absolute
Posture data is related to its parent model. Relative Posture data is related to itself.

Relative
Posture data is related to itself.

Sel Model
Posture data is related to selected model.

Teacher

Go
Causes the part to move to a selected teacher (Axis6) position. Use the Select Model tool to select the desired teacher axis and click Goto.

Pose
Moves the part to the teacher like “Go” but forces the frame to adjust its position to the teacher’s frame. If the teacher’s axes are rotated to a different orientation than the part being moved, the part will match the teacher’s Rx, Ry, and Rz orientation.

4.4.9 Pulse

Used to view and control the exact position of robots and external axes by pulse count rather than millimeters.

Axis Controls
The arrows to the right of each axis control adjust the position of robots and external axes by pulse count. The “thermometer” display represents the limit of travel.

NOTE: Values may be manually entered into the box for each axis. You can switch to the next box by pressing the Tab key.

Extern
Switches the axis controls from robot to external axis. S,L,U,R,B, and T correspond to external axes 0,1,2,3,4, and 5, respectively.

! Refresh. Moves the robot on the screen to the position indicated when pulse counts have been directly entered in the axis control windows.

Frame
Opens the Display Reference Frame dialog.

Fixed TCP
Keeps the robot TCP at the specified model.
4.4.10 Display Reference Frame

This dialog is only used in conjunction with the Pulse dialog and displays the position of the TCP relative to any of the listed coordinate systems.

**Pulse**
Changes the TCP position displayed in the Position Data dialog to its present position in pulse counts.

**Robot Frame**
Changes the TCP position displayed in the Position Data dialog to its present position in millimeters relative to the robot’s center point.

**Base Frame**
Changes the TCP position displayed in the Position Data dialog to its present position in millimeters relative to the world origin.

**User Frame**
Changes the TCP position displayed in the Position Data dialog to its present position in millimeters relative to the origin of the selected user frame.

**External Axis**
Changes the TCP position displayed in the Position Data dialog to its present position in millimeters.

**Joint (Angle, Length)**
Changes the TCP position displayed in the Position Data dialog to its present position in joint angle or joint stroke.

**User Frame Selector**
Allows changing between user frames when more than one frame is established.

**NOTE:** User Frame is not available if no user frames have been created.

**Speed**
Sets the speed at which the position controls shuttle the robot on the display.

4.4.11 Job View

This dialog displays the currently opened job file. It also allows you to edit move types and position levels.

**Del, Add, Mod**
Deletes or modifies the highlighted step or adds a new step after the highlighted step.

**Enter**
Completes any changes made to the highlighted step.

**File Menu**

**Move Tag…**
Used to set motion type, speed, and position level.

**Replace Position Only**
If selected, changes position data only when you modify a point. Speed and position level settings remain unchanged.

**Freeze Robot Position**
Freezes the robot so that modifications can be made to other steps.

**Select Job…**
Opens the previously saved or imported job files.
OLP Menu

*Make Model…*
Creates an Axis6 for each taught position in the currently active job.

*Make Jobs…*
Creates a job from Axis6 points to be used to shift jobs to compensate for changes in the cell environment. Points are created in the same order as the index points in the Axis6 model edit dialog.

*Make Mirror Job…*
The Job is converted to a mirror location. This function is performed using two combined functions; Process Angle, and Make Jobs.

*Paste Clip to Position…*
Creates JOB points from clipboard data. The data can be XYZ, Rxyz, or Pulse. Excel spreadsheets can also be used to calculate custom points. You can also define XYZ, Rxyz data by referring to any model frame desired.

*Copy Job to Robot…*
Copies a job from one robot to another robot.

**Option**

*Virtual…*
Used to access Faro Arm data display and setup.

**Window Menu**

*Refresh Window*
Refreshes the contents of the job display window.

### 4.4.12 Job Select

Allows you to select available jobs or copy the entire contents of one job to another. You can also create a new job by entering the desired job name in the JBI window and clicking OK.

*Editor…*
Opens selected JOB with NotePad.EXE as text allowing you to see or edit JOB text file directly.

*Copy…*
Displays a copy dialog that prompts you for the name of a destination job file. When you click Copy, the contents of the selected job are copied into the new job.

*Refreshes the job display if changes have been made via the programming pendant while the Job Select box is open.*

*Delete*
Deletes the selected job.

**Job Listing Window**
Lists all available jobs in the current directory.
Job Display Window
Displays the contents of the selected job.

4.4.13 Change JOB Holder
Used to select existing folders containing Jobs, or create new folders containing categorized Jobs. This allows you to quickly and easily find individual Job files.

Look In
Enables selection of folder wanted to be used to hold jobs.
All other functions are identical to a standard job select dialog.

4.4.14 Play Panel
Used to manually control job playback.

Reset
Resets the list box.

1 Cycle
Runs the job one time, then stops.

Step
Plays job one step at a time. Each time you click Play, the job advances one step.

Auto
Plays job continuously—resets automatically and plays job continuously until the program is stopped.

Line
Plays job one line at a time. Each time you click “Play”, the job advances one line.

Copy
Copies time values to clipboard to be pasted into a text document or spreadsheet.

Step Only
Displays the Play Option dialog.

Play Statistics
Displays real-time statistics on move time and play time as the job is running.

Data window
Displays job data in the following format: Line, Start, Length, Ratio, State

Line=Job line
Start=Cycle time at the start of the first move
Length=Move length in mm
Ratio=Relative portion of total job cycle time spent on the move
State=Status
  0-Accelerating
  1-Moving
  2-Decelerating
  3-Finish point

Delay bar
Allows you to view the simulation at a slower speed.
CONTROL FEATURES

4.4.15 Heart Beat

Heart beat is the interval at which MotoSim refreshes the display. Short duration heart beat settings make the simulation appear smoother but also makes it run slower. Longer heart beat settings make the simulation run faster but the display simulation will appear jerky. Default is 0.15 second.

Refresh Interval

Skip Mode
Sets the refresh interval to 1.5 seconds.

Default Mode
Sets the refresh interval to 0.15 seconds.

Detail Mode
Sets the refresh interval to 0.015 seconds.

Segment Clock
Controls how long MotoSim waits before it calculates the position of a trace point and displays it on the screen. Points will be calculated but not drawn until the next refresh. This is used most often to ensure accurate trace paths when using long refresh intervals as with Skip Mode.

4.4.16 Results

At the end of each program run, MotoSim will display job statistics in the status bar at the bottom of the MotoSim screen. Alternatively, you can select Playback · Cycle Time... from the Robot menu to display job statistics. The following dialog will appear.

Cycle Time

Play Time
This is the approximate time the program will take on a real system. It includes time for I/O and servo warm up. Changing position levels within the job will also affect this time.

Move Time
This is the sum time of all robot moves. This does not include I/O delays or warm up time.

Simulation
This represents how long it took MotoSim to run the program. The time will differ depending on the speed of the computer, the refresh settings, and trace options. This time has no relevance to actual cycle-time on a real robot.

Start Up Time
This is the time it takes for the robot interpreter to reach standby. Start up time is set automatically when you load up the parameter file from the controller. If this number is changed, be aware that cycle time calculations may not be as accurate.
4.4.17 I/O Monitor

Model I/O function sets up inputs and outputs for multiple robots. Select a robot from the Combo box on the Robot Toolbar, then select the I/O Monitor icon from the toolbar. The Model I/O Input dialog will appear.

**I/O Index**
Displayed inputs and outputs of robots listed in top pull down menu.

**Clr**
Selecting CLR will turn off all outputs of the robot whose I/O box is active.

4.4.18 Measure

This dialog appears when you click on two or more points while using the measure tool. It displays linear distance between two points.

**Distance**
Displays distance in millimeters between two points.

**Number Selected**
Sets how many points must be clicked before the result dialog appears. A setting of “2” will produce both a linear measurement in millimeters and relative position data. A setting of “3” or more will provide relative position data but no linear measurement.

**Copy…**
Copies the result of the measurement to the clipboard.

4.4.19 Reach View

Access Reach View through the OLP command in the Robot menu. This option displays the maximum reach envelope of the robot. This measurement is taken from the robot’s flange then drawn on the screen in an arc extending from the lowest point the robot can reach in front of itself to its maximum reach behind itself. The arc appears as a colored line.

**Trace**

**Link6 Reachability**
Creates the trace based on the robot’s selected link.

**Tool Center Point (TCP)**
Creates the trace based on the robot’s tool center point (TCP).

**Color…**
Specifies the color used to represent Link6 Reachability and TCP maximum reach.

**Option Divided**

**Vertical**
Specifies the number of vertical segments used to represent the reach model. The higher the number, the smoother the model will appear.
CONTROL FEATURES

*Horizontal*
Specifies the number of horizontal segments used to represent the reach model. The higher the number, the smoother the model will appear.

*Opacity*
Controls the opacity of the reach trace.

*Surface*
Displays the trace as a solid plane rather than a wireframe.

*Sphere*
Displays the reach area as a 3-D model superimposed over the cell and robot models.

*Dual Face*
Reach models are normally modeled by MotoSim with one-way visibility; that is, they are only visible from one side. In this case, the outside of the reach model is visible, but if you attempt to view the reach model from the inside, the model is not visible.

If Dual Face is active, the model is visible from both sides. This increases memory and processor usage but improves the appearance of the reach model.

*Smoothing*
Causes the model to be drawn using the highest quality settings. You may wish to turn this feature off if you have a slow computer.

*Delete…*
Deletes the selected reach trace.

4.4.20 *Robot Information*

This dialog provides information for, and allows editing of, robot frame position data. It allows you to adjust the robot frame when the real robot position differs from the MotoSim robot’s position relative to its base location.

*Offset of Robot Frame*
This enters offset values to correct model-versus-actual robot position.

*Start Up Time*
This is the time it takes for the robot interpreter to reach standby. Start up time is set automatically when you load up the parameter file from the controller. If you change this number, be aware that your cycle time calculations may not be accurate.

![Robot Property Dialog](image)

**WARNING!**

**DO NOT** change the parameter file in MotoSim unless it is absolutely necessary because of changes in your actual robot cell. If you change the robot offset in MotoSim, all your jobs will be created according to the new offset. If the job is then loaded to a real robot that does not have the same offset, damage or injury can result.
4.4.21 **Trace DLG Property**

The Trace DLG Property dialog will appear when `Trace · Property...` is selected from the Robot menu. It provides options for creating, deleting, and editing trace elements.

**Registered Trace**
Lists all traces calculated for the cell.

**Property**
Open the Trace Property dialog. See Section 4.4.22.

**Delete-->**
Places selected traces in the “Will be Deleted…” window.

**<--Add**
Removes selected items from the “Will be Deleted…” window.

**New**
Opens the Trace Property dialog so you can create a new trace.

**Robot’s TCP**
Creates a trace based on the robot’s TCP. This is the most common type of trace.

**Will be Deleted…**
Traces left in this window will be deleted when you click OK.

4.4.22 **Trace Property**

This dialog controls how the trace is drawn, its color, and which part of the robot the trace is based upon.

**Parent Model**
Opens the Model Select dialog to allow you to select the parent model to which the trace will be attached.

**Number of Points**
Sets the maximum number of points used by the trace. This number of points is not distributed evenly along the trace. Heart beat determines the interval. This option simply controls the maximum number of trace points that will be displayed from the start of the job.

**Color**

**Auto Color**
Automatically selects a color for the trace. If one or more traces already exists, the trace will be assigned a different color.

**Color**
Changes the color of the trace line.
CONTROL FEATURES

4.4.23 Camera

This dialog controls camera option defaults. Select Camera · Configuration… from the Camera menu to display this dialog.

Near Clipping
Changes camera default view settings.

Pos…
Allows you to specify fixed positions for the camera.

Note: The distance between the camera and object can be adjusted using the Spin button.

Panorama
Toggles the cell display between flat and 3-D perspective. At angles close to the floor, “Panorama” may cause the perspective to appear exaggerated.

Distance
Determines the camera’s distance from the world origin. The higher the number, the greater the distance and the smaller the cell will appear.

Width
Determines the camera’s field of view. The higher the number, the greater the field of view.

Light
Specifies the ambient lighting cast on the model.

Surface
Toggles the cell models between wireframe and hidden-line solid views. It is recommended that you leave this option turned on unless you have a slow computer.

Shading
Controls the shading of solid models. Turning this option on will smooth the appearance of the model, but it will also slow down display generation and the simulation time.

Refresh Timer
Adjusts the display generation and simulation time.

ms
Specifies the time interval for screen refresh. The default is 0ms. Set to 30ms or more for a smooth refresh rate.

Cycle
Indicates number of robot cycles shown every time screen is refreshed. The default is 20.

Scale

Pan Speed
Sets the default speed at which camera pans across the screen.

Loc. Speed
Controls the speed of the flying camera.

Magnification
Sets the default magnification.
**Default Buttons**
Stores three Scale settings. Each of the different settings can be used as the default by selecting the desired option before closing.

**Update**
Updates screen display to reflect any changes made in camera control dialog before closing dialog.

### 4.4.24 Camera Attach

This dialog controls camera attachment.

**Model**

**Follow**
Causes the camera to follow the assigned part from the current distance and angle. The camera will remain in the same plane and will not zoom in or out as the selected part moves toward or away from the screen.

**Attach**
“Mounts” the camera to the selected part. Allows you to see what the part sees as the program runs. Enter the name of the model or robot link to which you want to attach the camera before you click OK.

**Active**
Toggles the camera attachment options on and off.

### 4.4.25 Setting Unit Dialogs

The Setting Unit dialog appears when you select Set Unit of Object… subcommand from the Camera command in the Camera menu. This provides options for camera position and behavior.

**Unit String**
This box displays options for selecting the unit of measurement used in result dialogs and data display windows. The options will change between length, position, angle, and weight units depending on the tab selected.

**Connect**
Makes all dialogs use the same unit of measurement. Turning this option off allows independent settings in each of the four tab screens.

**Data**

**Min**
Minimum value (0 for absolute values, -n for relative values).

**Max**
Maximum data value. Helpful for limiting the size of your cell. Setting a value too low (less than 100000) may limit your ability to move models within your cell.
**Division Number**
Increasing the division number increases the jog resolution. For example: If Data Max is set to 1000 and the Division Number is set to 250, then the model will move in increments of 4 (mm or pulses) when you use the position editing dialogs.

**Speed**
Preset speeds for slow, medium, and fast. Click desired speed to insert the corresponding number in the Data box.

### 4.4.26 Auto Place

This dialog controls options for automatic placement of a robot to minimize cycle time for a given cell layout and selected robot program. INF files can also be used individually with this dialog. To use Auto Place, proceed as follows:

1. Select a Job to analyze. A job using simple robot instructions, such as MOVL and MOVJ, is recommended. Do not include CALL, JUMP, WAIT, etc.
2. Select “Load From Job”, so that you have AXIS6 in the list.
3. Click the “Job…” button. The “Making JOB” screen appears. Follow all sequence.
4. Click the “Search” button, so all possible positions are displayed.
File Menu

Open…
Activates an open dialog from which you can select Auto Place files.

Save…
Saves the current Auto Place file.

Edit File
Opens the current Auto Place file in NotePad for editing.

Exit
Exits Auto Place.

Edit Menu

Copy Search Location
Copies search location to the clipboard.

Teach Menu

Add
Adds a position point for the robot.

Delete a Point
Deletes the selected point.

Delete All Points
Deletes all position points.

Load From JOB
Loads teach points from an existing INF (MotoSim job) file.

Check Reach
Checks reach from all possible robot positions within specified limits.

Making JOB…
Creates a JBI (job) file from an INF file (MotoSim’s job file format).

Operation

Org Position and Pulse
Moves the robot to the last saved position after job motion or the robot position/posture has changed.

Org Position
Returns the base of the robot to its original position.

Org Pulse
Returns the robot to its original posture.

Status

Org Position
When this option is on, the search area is constrained relative to the original position. When it is off, the search area is recalculated based on the current position, and new Auto Place locations are generated.

Org Pulse
Returns the robot to the original posture before proceeding to the next Auto Place location. This resets the posture between each location and reduces posture drift as the robot tests each location.

Location
This dialog contains various commands for determining the optimum location for your robot from all possible calculated positions.
**Search Start**
Searches for all possible robot positions within reach of the work envelope.

**Search Area Def…**
Enables you to limit the Auto Place search area and density.

**Load from Model File…**
Enables use of Axis6 model index points as the basis for an Auto Place search.

**Delete a Location**
Deletes the selected robot location.

**Delete All Locations**
Deletes all generated robot locations.

**Quick Cycle Time**
Runs robot job in quick cycle time mode and then jumps to the next Auto Place location without displaying the robot performing the entire job. This feature is useful for multi-step jobs or testing a high number of possible robot positions.

- **Start**
  Starts the Quick Cycle Time operation.

- **Stop**
  Stops the Quick Cycle Time operation.

- **Define INF File…**
  Select INF file to perform AutoPlace or cycle-time calculations.

**Status**

- **Name Number**
  Displays sequence number for each Axis6 point created during Auto Place operation.

- **Synchronize**
  The robot moves to each possible location while the TCP remains at the current job step. This allows viewing of robot posture when reaching a single point from multiple robot locations. The robot moves to each position while remaining in the same posture.

**Select Teach**

- **Add**
  Adds the currently selected teacher to the Select Teach box. These are the points from which the Auto Place will be calculated.

- **Check**
  Performs a quick check of all Teach points without executing a simulation of robot movement. Invalid or out-of-reach points are prefixed with “NG>” in the Select Teach listing.

- **Job…**
  Used to open a different job so you can perform Auto Place on it.

**Search Location**
Displays a list of all Auto Place locations (in pulse counts). The current location is highlighted in the list as the search operation is in progress.

- **Search**
  Starts the Auto Place search operation.

- **Del**
  Deletes the highlighted Auto Place location.
4.4.27 Process Angle (Frame Op.)

Cube posture is modified from the Frame Op portion of the Cube Edit window. The Rx, Ry, and Rz posture of a cube can be modified by selecting “Arithmetic”. Mirror transfer of a cube is accomplished by selecting “Mirror-Trans”.

Arithmetic

Mathematically modifies cube posture.

Primary Axis

Shows methods for transferring AXIS6.

“Use Default Frame” generates the same AXIS6 posture (Rx,Ry,Rz) as its model. “Sel Model...” is required for defining the default frame.

“Use Path Frame” generates an AXIS6 posture using a Frame coordinated with the nearest three points.

“Use Near Polygon” generates an AXIS6 posture perpendicular to the nearest polygon. “Sel Model...” is required for defining the polygon model.

“Use Cylindrical Frame” generates an AXIS6 posture that is cylindrical coordinated. “Sel Model...” is required for defining center of cylinder.

Primary Axis also creates AXIS6 postures that a specific axis directs to the next point. “Sel Model...” specifies the available models needed when required.

Mirror-Trans

Modifies AXIS6 to be mirror transferred. Several options are available for making mirror transfer. Select the “Mirror plane” and “Keep plane” of the TCP frame.

Translate/Rotate

Creates matrix transfer for each AXIS6 model.

Trans(X), Trans(Y), and Trans(Z) make matrix transfer.

Rotate(X), Rotate(Y), and Rotate(Z) make matrix rotation.

This option can be used with any combination and multi line. This operation is performed after primary translations.
To start MotoSim:
Select *Programs · Motoman · MotoSim* from the Windows Start menu.

### 5.1 Opening a Cell File

An attractive feature of MotoSim is that it contains accurate models of standard Motoman products. To use MotoSim with a standard Motoman system, simply open a single cell project file (“.mcl” file) and all of the system’s components will appear on the screen, ready for programming. With custom systems, cell files must be manually constructed to represent the equipment used. See “” on page 1., for information on creating a new cell file.

To open a cell file:
1. Select *Open…* from the File menu.
2. Open the *Cells* directory.
3. Double-click the desired cell (“.mcl”) file. The cell file will open and the specified components within that cell will appear on the screen. See Section 5.7, Exporting and Importing AutoCAD DXF Files, for information on programming the cell to perform a job.

### 5.2 Loading a “Previous Version” Cell

If you want to use “.cel” files that were used by MotoSim version 4.3 or before, you must update them to “.mcl” files first. To update a “.cel” file, proceed as follows:

1. Select *New...* from the File menu. The Select Motoman Cell file window will appear. Select *Cancel*.
2. Select *Load Cell...* from the File menu.
5.3 Creating a New Cell File

If you are using MotoSim to model custom cells or you are using equipment not sold by Motoman, you must manually build the cell file to represent the equipment.

NOTE: Motoman constructs robot and cell models on an ongoing basis. If your copy of MotoSim does not include model files for your Motoman-supplied equipment, contact Customer Service to check if new model files are available.

To create a new cell project file, proceed as follows:

1. Select New... from the File menu. The Select Motoman Cell file window will appear.
2. Enter a name for the new cell file in the File Name box.
3. Click Open. MotoSim will create a new directory using the name of the cell you just created. It will also create a directory for models.

NOTE: When you select OK, the new cell is created and opened. It is not necessary to select Open from the File menu. Using the "Drive" box and the "Directory" box you may specify where the new cell and associated sub-directories will be created.

CAUTION!

Cell files are over-written when a new version of MotoSim is installed. If you have modified a “standard” supplied cell, you may lose important data unless you first copy your cell files to another location.

4. The cell filename was created from the name of the cell. For example, if you named the cell “robot1,” the cell file will be “robot1.mcl.”
5. Your new cell file will open. It will contain a model of a floor that looks like a checkerboard. This is a good time to save your cell file. The next step is to add a robot.
5.4 Adding Robots

To properly program robots and equipment with external axes, MotoSim requires a parameter file obtained from the controller of your system. Preconfigured MotoSim cell files for Motoman turnkey systems have all the external axes already in place. Motoman systems have the appropriate parameter files for all equipment installed in the system.

NOTE: It is strongly recommended that you use the parameter file from your robot. You may have additional options enabled that standard included files don’t include and you could ultimately disable a purchased function in your robot by loading the all.prm used in MotoSim.

The first step is to add the robot:

1. Select Add Robot… from the Robot menu. An Open dialog box will appear that requests the location of the parameter (“all.prm”) file for the desired robot. The parameter file is obtained from your controller using FDE (floppy disk emulator). Consult your Motoman FDE manual if you are not familiar with FDE operation and use. You may also obtain the parameter file from the controller via FC1, FC2, and VDE.

NOTE: MotoSim comes with basic “all.prm” parameter files for demonstration purposes. It is recommended that you immediately retrieve the parameter file from your controller and overwrite the parameter files in the MotoSim directory.

CAUTION!

DO NOT copy the parameter files included with MotoSim to your controller! Doing so will destroy all software options you may have purchased.

2. Once you have located the parameter file and selected it, the Install Robot dialog will appear. It will ask you for a robot name. Give the robot any name you wish.

3. The Filename box contains a space to enter the model of Motoman robot you will be using. Click Model.

NOTE: It is very important that you select only the file for the robot you are using. Improper robot selection will cause obvious modeling errors on screen and make programming impossible. If your robot model looks collapsed or exploded on the MotoSim screen, load correct model file.

4. Select the “.mdl” file for the model of robot you are using and click OK. The robot will appear in the center of the cell floor. You will be viewing it from the top. Switch views to examine the robot from different directions.
NOTE: Most MotoSim robot models have the Z-offset of the robot set so that the robot sits squarely on the floor. If for any reason you move the robot and want to return it to the original floor height, refer to Table 5-1 to get the Z-offset for your robot. Select your robot from the CAD Tree, click POS, and enter the Z position for your robot.

<table>
<thead>
<tr>
<th>Model</th>
<th>Z-Offset</th>
<th>Model</th>
<th>Z-Offset</th>
</tr>
</thead>
<tbody>
<tr>
<td>K10</td>
<td>585 mm</td>
<td>S604</td>
<td>0 mm</td>
</tr>
<tr>
<td>K100</td>
<td>830 mm</td>
<td>SK120</td>
<td>850 mm</td>
</tr>
<tr>
<td>K1004</td>
<td>830 mm</td>
<td>SK16</td>
<td>585 mm</td>
</tr>
<tr>
<td>K100R</td>
<td>550 mm</td>
<td>SK45</td>
<td>760 mm</td>
</tr>
<tr>
<td>K205</td>
<td>900 mm</td>
<td>SK506</td>
<td>1830 mm</td>
</tr>
<tr>
<td>K3</td>
<td>384 mm</td>
<td>SK506-C1</td>
<td>1830 mm</td>
</tr>
<tr>
<td>K30</td>
<td>730 mm</td>
<td>SK6</td>
<td>530 mm</td>
</tr>
<tr>
<td>K30-C01</td>
<td>730 mm</td>
<td>SK6-C02</td>
<td>530 mm</td>
</tr>
<tr>
<td>K506</td>
<td>1830 mm</td>
<td>SP100</td>
<td>880 mm</td>
</tr>
<tr>
<td>K6</td>
<td>550 mm</td>
<td>SV035</td>
<td>350 mm</td>
</tr>
<tr>
<td>K60</td>
<td>720 mm</td>
<td>SV3</td>
<td>300 mm</td>
</tr>
<tr>
<td>K604</td>
<td>750 mm</td>
<td>UP120</td>
<td>525 mm</td>
</tr>
<tr>
<td>NY300</td>
<td>1000 mm</td>
<td>UP130</td>
<td>730 mm</td>
</tr>
</tbody>
</table>

* All robot versions are -C00 unless otherwise specified

The next step is to model your other equipment. By default, MotoSim will use the Basic Modeler to create new models unless you specify otherwise.

New parts or equipment will be created at the world origin (the middle of the floor), right where the robot appeared. First, you have to move the robot away from the center of the floor to avoid confusing overlap of model parts.

1. Select the robot in the CAD Tree.
2. Choose Set Position in the Attribute menu or in the right mouse-button menu.
3. Click the \(X(mm)\) and \(Y(mm)\) arrows in the Position box until the robot moves away from the center of the floor. Adjust the Step number higher to move the robot in greater increments.
4. Click OK when finished.
5. Save your cell file.
5.5 **Modeling Equipment and Production Parts**

Regardless of your application (welding, material handling, cutting, etc.), the next step is to add models that represent equipment and production parts to your cell file.

MotoSim features preconstructed models for Motoman products and third party products sold with Motoman cells. Modeling of equipment may not be necessary if you have a standard Motoman product.

MotoSim (full package) features a powerful advanced modeler called Inovate™. It can be used to import 2-D and 3-D CAD files and to create rendered 3-D models from scratch. Inovate™ is called “advanced modeler” in the MotoSim program.

**NOTE:** Please refer to the TriSpectives documentation provided with your MotoSim package for instructions on using TriSpectives to create models.

Perform the following steps to model equipment and parts in your cell:

1. Select “world” to create a model at the world origin. You may also select any other object if you wish that object to be the parent of the new part. Next, decide whether to use MotoSim’s Advanced Modeler (the preferred method) or Basic Modeler to generate your part model(s).

**NOTE:** Advanced Modeler will generate more attractive models with greater ease but may run unacceptably slow on pre-Pentium computers. The advanced modeler will not work with existing models created using the basic modeler.

2. Select **New Model** from the File menu of the CAD Tree window. The Add Model dialog will open.

3. Type a name for the new model and click **OK**. An advanced model will be created by default. If you want a basic model, be sure to turn off the External CAD option. The CAD Tree window will open. You will see that your new model has been added to the cell’s tree.

**NOTE:** If the new model is the child of another part, the tree may need to be expanded before the new part is visible.

4. Click on the model with the right mouse button and a floating menu will appear. Select **Edit Model**. MotoSim will open TriSpectives. Consult TriSpectives documentation for instructions on how to create models. If you turned off the External CAD option, you will be prompted with an Add Parts dialog to build up your equipment model from basic geometric shapes. Collect all the dimensional data for the position and size of the equipment or part before you begin.

The “Add Parts” dropdown list contains selections to create a box, cone, cube, cylinder, face, pipe, Axis6, or floor elements. (You would only create a floor if you had started with an empty cell.)

5. Select the type of part you want to add and click **Add**.
6. A part edit dialog will appear. The actual contents of the dialog will differ depending on the type of part you added but it will always have settings for position, color, and size. Make all desired settings and then click **OK**.

**NOTE:** When you create a new part, it will appear at the origin of the parent part unless you specify its position from the start. If you are building up models from many parts, you need only change the position of the highest level element. All the child parts will follow because they are “attached” to the original part.

**NOTE:** You must click **OK** for changes to be saved. Clicking the “X” in the upper right-hand corner will exit the dialog but changes will not be saved.

7. Repeat Steps 1 through 6 to add additional parts.

**NOTE:** If a new part is created at the world origin and it is smaller than a part located at the world origin, you may not see the new part because it has been created “inside” the larger part. Adjust the Z(mm) position to move the new part up or the X(mm) or Y(mm) to move the part away from other parts. You can also set the Visible attribute of the larger part to Off. Remember to turn it back on after the smaller part has been moved.

8. Once you have created and positioned all necessary parts to represent your positioner, save your cell file.

### 5.6 Copying Parts

If a tool or part has already been modeled, it is possible to copy the model so you don’t have to model from scratch.

1. Open the CAD Tree and locate the part you would like to copy.

**NOTE:** In complex cells, it is sometimes difficult to locate the model which represents a particular part. Click the Select Model button on the tool bar (it looks like a pointing hand) and then click the part in the MotoSim window. The part will be highlighted in the **CAD Tree**.

2. Select **Copy** from the Edit menu.
3. Click to select the desired parent model.
4. Select **Paste** from the Edit menu.
5. Click **OK** to copy the model. The new model will appear in the CAD Tree in the specified location.
6. Select **Edit Model**… or other editing functions from the floating menu to change the part attributes or position as desired.
5.7 **Exporting and Importing AutoCAD DXF Files**

5.7.1 **Exporting a DXF File**

This function exports a .dxf file containing all cell data.

If you select a model, you can export DXF/DWG file of the specified model.

1. Select AutoCad Export… from the File menu of the CAD dialog.
2. Select “All Models.”
3. Click Convert.

5.7.2 **Importing a DXF File**

This function converts DXF/DWG (AutoCad data) into an .mdl file. MotoSim can convert line data only. If you require face data, use the Trispectives CAD Import function.

1. Select AutoCad Import… from File menu of the CAD dialog.
2. Click File… in the Import File category.
3. The model file name is automatically assigned.
4. Using Cell model path creates the default of model file name. If “Copy to Model holder” is selected, the model file name defaults to.mdl. Otherwise, the model file name will have the same path with DXF file extension.
5. Click Convert.
6. An .mdl file will be created in the specified directory.
7. Select New Model… from the File menu of the CAD dialog.
8. Input a model name and select a file name.
5.8 Creating a Robot Job in MotoSim

MotoSim’s interface is designed to closely resemble that of an actual programming pendant and environment.

⚠️ CAUTION!
Do not attempt to download and run MotoSim jobs if you are unfamiliar with programming on the XRC/MRC controller.

⚠️ WARNING!
Ensure that you are programming your job using the proper parameter files for the intended system. If a job programmed for one system is run on a different system, the robot may malfunction and cause injury or damage.

Creating jobs consists of two basic operations—programming moves and entering commands. MotoSim is designed to create, edit, and simulate robot moves only. MotoSim can monitor I/O functions but cannot program I/O. I/O, JUMP, CALL, and other commands must be manually added to the job. See Section 6.3.1, Non-Motion Programming, for information on performing I/O programming using a text editor.

The following programming instructions assume you have already created a robot cell with a robot model appropriate for the application.

5.8.1 Programming Points Using the Programming Pendant

The most common tool you will use to program in MotoSim is the programming pendant. MotoSim’s programming pendant resembles a robot programming pendant.

Open a cell file you have created and spend a few minutes familiarizing yourself with the controls. As you click on the motion buttons, the robot on the screen will move according to your commands.

Try the different move types and speeds until you are comfortable shuttling the robot.

To program a point, proceed as follows:

1. Move the robot to the desired safe position. Typically, each job will begin and end with a move to a safe position. Use the motion buttons to move the robot to the desired safe position.

2. Make sure “NOP” is highlighted in the programming pendant job window.

3. Make sure “Add” is marked at the bottom of the programming pendant and click Enter. The move will appear as Step 001 in the programming pendant display.

4. Move the robot using the motion buttons or mouse pick an object. After each move, click Add, Enter to add the move.

NOTE: You can insert a move in the middle of the job by highlighting the step above where you want the new step to appear. When you click Enter, the new step will appear below the highlighted step. You cannot add steps...
To change a point or set default values for all following points:
Click Move… on the programming pendant to change the speed, move type, and position level of a move before you enter the move. If you have already entered a move, highlight the line to be edited, make your changes, and click Modify, Enter.

To delete a point, proceed as follows:
1. Highlight the step to be deleted.
2. Click Delete, Enter. The step will disappear from the job and the following steps will be renumbered automatically.

If you have accurately modeled parts, you can use “Pick” features to assist moving the robot to parts. This greatly speeds up programming and reduces off-line programming errors.

To program a move to the edge, surface, or corner of a part, proceed as follows:
1. Click OLP to open the OLP dialog.
2. Click Free, Vertex, or Center to specify how you want the mouse to pick points.
   - “Free”—click any point on the object and the robot will move to that point.
   - “Vertex”—find edges and corners easier by snapping the robot to the exact intersection of two or more lines. Use this mode when you are programming points along the edges of parts.
   - “Center”—snap to the center of part surface.

**NOTE:** When OLP is enabled, the robot will move wherever you click. In order to use certain tools and camera functions, you must first click “Active” to disable it. Enable “Active” to continue picking points.

3. Once you have completed entering all the robot’s moves, insert desired I/O and other manual commands into the job.
4. Filter the MotoSim job with MotoCal.
5. Save the job and use FDE to transfer the job to your controller.

**WARNING!**
Ensure that you are transferring your job to the intended system. If a job programmed for one system is run on a different system, the robot may malfunction and cause injury or damage.
5.9  **Axis6 Models**

Axis6 models are used as reference points that include position and orientation data. When POSE is checked in the OLP dialog and an Axis6 is selected with the OLP tool, the TCP on the robot will adjust its orientation to match that of the Axis6. This is useful for programming multiple points with precise control over tool angle and position.

There are several types of Axis6 models. Dark blue Axis6 models represent origins of cells and robots, and tool center points. Cyan Axis6 models represent user frames. Yellow Axis6 models represent points in a job model.

5.9.1  **Automatically Creating Axis6 Job Models**

MotoSim can automatically generate Axis6 job models which correspond to programmed points.

**HINT:** By creating Axis6 job models in a part model, you are actually storing the job data in the part model, not the robot. If you run a wide variety of different jobs, you can simplify your job-model management by storing the job data within its respective part model by using Axis6 job models.

1. Open the Job View panel.
2. Select **Make Model** from the OLP menu.
3. Select the desired parent and model color.
4. Click **OK**.

The new model will appear in the CAD Tree named “[robotname]x Jx [jobname]” where “default” represents the name of the job into which points were programmed. For more information about defining Axis6 points mathematically.

5.10  **Job Playback and Settings**

There are several operations you can perform with MotoSim during and after programming a job.

5.10.1  **Selecting a Job**

To open a job:

1. Select **Controller · Sel Job**... from the Robot menu. The Select box will appear containing a list of jobs for the active robot.

2. Select the desired job and click **OK**.

3. Run the job as described in Section 5.10.1.
5.10.2 Playing a Job

After you have programmed a job, use the play control buttons on the tool bar to:

<table>
<thead>
<tr>
<th>Table 5-2</th>
<th>Play Control Buttons</th>
</tr>
</thead>
<tbody>
<tr>
<td>⏪</td>
<td>Reset Job</td>
</tr>
<tr>
<td>⏯</td>
<td>Play</td>
</tr>
<tr>
<td>⏯</td>
<td>Pause</td>
</tr>
</tbody>
</table>

5.10.3 Multiple Robot Job Selection

If your cell has multiple robots, this procedure determines which job will be run on each robot.

1. Select Robot · Controller · Change Robot.
2. Select Robot · Controller · Sel JOB… to choose the desired job.
3. When finished, click OK.

5.10.4 Opening a Cell File

Select File · Open or select recently opened cell files listed in the File menu.

5.10.5 Adding/Deleting Robots

Select Robot · Add Robot… or Delete Robot…. You will be prompted to locate the all.prm file that represents the robot you want to add or delete. Select the desired file and click OK.

5.10.6 Creating a Teaching Sheet

A teaching sheet is a set of reference points that can be manipulated together, allowing you to shift jobs or make repetitive changes to multiple programmed points. A teaching sheet is made from a job file (INF or JBI). To create a teaching sheet from a job, perform the following steps:

1. Open the desired job by selecting Controller · Sel JOB from the Robot menu.
2. Select Teaching Sheet · Create from the Robot menu. A colored line with numbered points will appear to represent the programmed job.

The teacher can be moved as a unit to allow easy adjustment of a job to reflect changes such as a work table being moved in the real cell. Move the job to the desired location and then use OLP Pick to click on the teacher points and reprogram a new job.

To identify and reposition the teacher in the CAD Tree window, look for a model named "[robotname]x Jx [jobname]."
5.10.7 Displaying Teaching Sheet

A teaching sheet displays program points connected by a colored line. The program points can be labeled with a step number. This feature is useful for analysis calculations and job shifting.

To create a teaching sheet, proceed as follows:

1. Load a job.
2. Select Robot · Teaching Sheet · Create.

To delete a teaching sheet, proceed as follows:

Select Robot · Teaching Sheet · Delete.

To change the color of the teaching sheet, proceed as follows:

1. Select Robot · Teaching Sheet · Create.
2. Select the desired color and click OK.

5.10.8 Creating a Trace

Trace creates a visible dotted line which represents the actual path the robot takes during execution of the current job.

To create a trace, proceed as follows:

1. Select Robot · Trace · Property. The Trace DLG Property dialog will appear and prompt you for information needed to create the trace.
2. Click New.
3. Select the model that you want to trace (typically the TCP) and
4. Click Property… to change color or number of points representing the trace.
5. Click OK. Next time you run the job, the trace will appear.

To delete a trace, proceed as follows:

Select Robot · Trace · Property. The Trace DLG Property dialog will appear and prompt you for information needed to delete the trace.

NOTE: You can use the trace clipboard feature to copy trace data to the clipboard. You can then paste this data into a spreadsheet program to perform cycle time and move analysis.

5.10.9 Displaying/Hiding Axis6 models

Selecting Camera · Model · With Wire frame toggles all Axis6 models on/off.

5.10.10 Changing Axis6 Length

Select Camera · Frame Length.
5.10.11 Calculating Cycle Time

Cycle time is calculated each time a job is run and is displayed in the status bar at the bottom of the screen.

Cycle time can also be displayed for the currently loaded job by selecting *Playback · Cycle Time...* from the Robot menu.

Quick Cycle Time calculates cycle times in the same manner but does not execute simulation of the robot moves on the screen. This is helpful when performing cycle time calculations on large or slow-moving jobs or when running MotoSim on a pre-Pentium computer.

*NOTE:* Cycle time is calculated from the present step to the end of the job. Click the Reset Job button on the tool bar to ensure the job is starting from the beginning.

5.10.12 Displaying/Using Pulse Record Data

The Pulse Record window contains a "transcript" of every robot position used to perform the current job. This data can be copied to a spreadsheet program for analysis or saved to a file for use at a later time.

To display the Pulse Record Window, proceed as follows:

- Click *Pulse Record* on the tool bar
- Select *Playback · Pulse Record...* from the Robot menu

Clicking a specific line in the Pulse Record window will cause the robot to move to that position in the job.

*NOTE:* Pulse Record points correspond to the individual trace points displayed as a broken line in the MotoSim window. You may have to increase view magnification or detail level to see the individual points.

5.10.13 Displaying Maximum Reach

The maximum reach of the robot’s TCP or Link6 can be displayed graphically on the screen. The reach model can assist you in placement of models in the cell and help prevent collisions between the robot and cell components or other robots.

The reach model can be displayed in different colors and opacity.

To create a Maximum Reach model:

1. Select *OLP · Reach View...* from the Robot menu.
2. Select Link6 Reachability or Tool Center Point (TCP) and the desired color for the model.
3. Select desired appearance options (see Reach View on page 4-30 for a description of each option).
4. Click *OK*. The reach model will appear in the cell.

You can change reach model attributes at any time by returning to the Reach View screen as described is Step 1.

*NOTE:* If you have more than one robot in your cell, you must first switch to the desired robot before you can add, delete, or modify its reach model.
5.10.14 Making Models from Jobs

This operation creates an AXIS6 model as an ".mdl" file for a specified robot. After conversion, you can modify the new model’s properties (location, color, pose, etc.).

1. Select OLP · Make Model… from the JobView dialog box.
2. The Parent model name box defaults to the parent model for the specified robot, but you can change the parent model name by entering a new name and pressing Select...
3. You can also change the color of the AXIS6 model by choosing a new color and pressing Select...
4. Click OK.

5.10.15 Making Jobs from Models

This operation creates an ".jbi" file for a specified robot from a AXIS6 model. The Make Job from Model dialog box tells you the name of the model, the name of the job, the name of the model file, the name of the parent model, and the number of positions.

1. Select OLP · Make Job… from the Job View dialog box.
2. From the pull down menu, select desired model.
3. Click Make Job.
4. An information box will appear, telling you the job was created and the number of positions which were converted. Click OK.

5.11 Creating a Catalog List

You can create a catalog list of cell files by importing .mcl files into an Excel worksheet. The screen below shows how .mcl files appear in the catalog.
6.1 **Process Angle**

6.1.1 **Teaching Robot Jobs Using Microsoft Excel**

1. Select data on your Excel spreadsheet and copy the data (Ctrl+C). Columns A, B, C, D, E, and F correspond to robot axes S, L, U, R, B, and T) respectively.

2. Click **Add**, and set MOVJ VJ=50.00.

3. Select **Paste Position**... from the OLP menu in the JobView dialog box. The following dialog box will appear.

Data is displayed in pulse counts. Click **XYZ,Rxyz** to display values in millimeters and degrees.

If the dialog box reads “1(x),” conversion has failed. Data displayed in this form cannot be converted to a job.

If the dialog box reads “3(o),” conversion has succeeded. Data can be converted to a JOB.

If “**XYZ,Rxyz**” is active, select “**RefFrame...**” model, and use this data as the original frame.
6.1.2 Converting Intersection Frame to Job

The MotoSim Process Angle feature is compatible with MotoSim version 6.2. This function creates JOB files from intersection data.

1. Create a new cell.
2. Add a UP6 robot model (Robot · Add Robot...)
3. Create a new model. In the CAD tree, right click the FLOOR folder and select New Model...
4. Name the new model “TableCenter” and check the dummy box.
5. Set position of TableCenter as follows:
   \[ \text{Pxyz} = 800, 0, 800 \]
   \[ \text{Rxyz} = 180, 0, 0 \]
6. Create a new model. In the CAD tree, right click the TableCenter folder and select New Model...
7. Name new model “Table01”.
8. Add a BOX model in Table01. In the CAD tree, double click the new Table01 folder.
9. Select BOX and click Add.
10. Set values as follows:
    \[ \text{Width} = 100 \]
    \[ \text{Depth} = 100 \]
    \[ \text{Height} = 10 \]
11. Create a new model. In the CAD tree, right click the Table01 folder and select New Model...
12. Name new model “Obj01”.
13. Add a CYLINDER model in Obj01. In the CAD tree, double click the new Obj01 folder.
14. Select CYLINDER and click Add.
15. Set values as follows:
    \[ \text{U.Diameter} = 50 \]
    \[ \text{L.Diameter} = 50 \]
    \[ \text{Height} = 10 \]
16. Click Posture... and set values as follows:
    \[ \text{Pxyz(mm)} = 0, 0, -10 \]
    \[ \text{Rxyz(deg)} = 0, 0, 0 \]
17. Click the Collision Detect button on the Camera toolbar.
18. Create a new collision pair.
19. Add target models (Obj01, Table01, etc.)

20. Select Create Line Models... from the Collision Pair dropdown menu.

21. Select Table01 and select the Frame (Axis6) and Delete Same Point options.

22. Click OK. Table01x0000 Model Name appears.

23. Click OK again.

24. Select Camera · Frame Length...

25. Set Length to 3 mm.

26. In the CAD tree, double click the Table01x0000 model.

27. Select AXIS6 and click Edit.

28. Click Adjust...

29. Select X-axis to Next-Point for Primary Axis and click OK.

30. Click Adjust... again.

31. Select None for Primary Axis. Enter “1” for Translate/Rotate Angle (deg) and select the Y radio button. Click Add. This adjusts the width of the cutting bead.

32. Enter 30 for Translate/Rotate Angle (deg) and select the Rx radio button. This adjusts the angle of the cutting brad.
33. Click OK.
34. Click OK again. A Frame (AXIS6) model series appears.

35. Click Select Job button from Robot toolbar.
36. Enter “MDJOB” in JBI field and click OK.
37. Click Position and Job View button from Robot toolbar.
38. Select File · Move Tag...
39. Modify Interpolation data and click OK.
40. From the CAD tree, right click the Table01x0000 folder and select Edit Model.
41. Click the Clipboard button.

42. Click Copy to and click OK.
43. Click OK.

44. Select the Add radio button.
45. Select OLP · Paste Clip to Position...
46. Click the *RefFrame* button.
47. Select the Table01x0000 folder and click *OK*.

48. Select *Sym Tool* check box and click *OK*. The job created from the intersection data appears.

49. The result of the job simulation appears as follows:
6.1.3 Converting Inform-script to Job

The Inf2Jbi function is part of the Auto Place tool bar button. This function supports AXIS6 based job creation. Inform-script files can be easily be created from Pulse Jobs. In addition, Pulse jobs can be created from Inf files by simply modifying the data using Excel.

Creating a JOB file from an Inform-script file

1. Create a JOB1.JBI file at any cell. This is a pulse type job.

2. Click on the Auto Place toolbar button. The Auto Place dialog appears.

3. Select **Load from JOB** from the Teach menu. The Up50-a00xJxJOB1 model appears under the same parent as the Up50-a00 robot model.

4. Click **Job**... in the Auto Place window. The **Making JOB Page 1/2** window appears.

5. Type JOB2 in the **Job:** window and press **Next >**. The **Making JOB Page 2/2** window appears.
6. Click **Convert >>**.
7. Click **Finish**.

8. Open the JOB2.jbi file. Notice that all motion types were converted to MOVJ. This is a pulse type job and is related to the JOB2.inf file.

9. Click **JOB...** in the Auto Place window again. This time select the *Existing* radio button and select the JOB2.inf file.

10. Click **Next**.

11. Click **Editor...** This allows you to edit the JOB2.inf file in NotePad.
12. Modify the motion types as follows:
   TBOXTEACH=(MOVJ,Up50-a00xJxJOB1:AXIS6[0][0],NULL,NULL);
   TBOXTEACH=(MOVJ,Up50-a00xJxJOB1:AXIS6[0][1],NULL,NULL);
   TBOXTEACH=(MOVL,Up50-a00xJxJOB1:AXIS6[0][2],NULL,NULL);
   TBOXTEACH=(MOVL,Up50-a00xJxJOB1:AXIS6[0][3],NULL,NULL);
   TBOXTEACH=(MOVL,Up50-a00xJxJOB1:AXIS6[0][4],NULL,NULL);
   TBOXTEACH=(MOVL,Up50-a00xJxJOB1:AXIS6[0][5],NULL,NULL);
   TBOXTEACH=(MOVJ,Up50-a00xJxJOB1:AXIS6[0][6],NULL,NULL);
   TBOXTEACH=(MOVJ,Up50-a00xJxJOB1:AXIS6[0][7],NULL,NULL);

13. Save the file and close NotePad.
15. Click Finish. The MOVL instructions are now in the job.

Velocity can also be modified in a similar manner using the inform-script and NotePad.
Modifying Using Microsoft Excel®

1. Double click the “Up50-a00xJxJOB1” model in the CAD tree.

2. Click Clipboard.... The Shape Exchange Clipboard appears.

3. Click Copy to.

4. Open Excel and paste the text data into the spreadsheet.

5. Add 30mm to the Z axis data.

6. Select the data area and copy to clipboard.

7. Click Paste from in the Shape Exchange Clipboard. Notice the Z-axis values are changed.

8. Click OK.

9. Click OK once more in the Up50-a00xJxJOB1 window. The job data is shifted 30mm higher.
10. Click on the Auto Place toolbar button.
11. Click Job....

12. Select the Existing radio button, and select JOB2.inf.
13. Click Next >.
14. Click Convert >>.
15. Click Finish.

16. Open JOB2.jbi again. Notice all positions are shifted 30mm higher.

### 6.1.4 Copying a Job from an SK120 to a UP130

1. Open a Cell.
2. Add a SK120 robot with MRC control (Robot · Add Robot...).
3. Add second robot (UP130 with XRC control).
4. Select the SK120 job you want copied to the UP130.

5. Set the TCP for both robots to the same values (Robot · Modify Data · Tool Data).

6. Click the Position and Job View button from the SK120 robot toolbar.

7. Select OLP · Copy Job to Robot... and select the UP130 robot.

8. Click OK. The SK120 job has now been copied to the UP130.
6.1.5 **Mirroring the UP130 Job**

1. Open Cell containing UP130 robot.
2. Click *Select Job* button
3. Highlight job to be mirrored and click *Copy...*. The Copy Job window appears.
4. Name the Destination of Copy and click *Copy* and *OK*.
5. Click the *Position and Job View* button from the UP130 robot toolbar.
6. Select *OLP · Make Mirror Job*...

7. The job can be mirrored about any object that exists in the cell. To change the object, click *Sel Model*...

8. Select desired object from CAD tree and click *OK*. 

---

**ADVANCED OPERATIONS**

---

**MOTOMAN** 6-13 *MotoSim User’s Manual*
9. Select model to be used to create the mirrored job.
10. Click Make Job.
11. New job is successfully created. Click OK.

### 6.2 Modeling Extension

#### 6.2.1 Creating Model Files from 3-D CAD Files

Files are imported into MotoSim using the Inovate™ Advanced Modeler. The method you use depends on what type of file you are importing. Table 6-2 lists some of the formats that can be imported into MotoSim and the method of import.

<table>
<thead>
<tr>
<th>File Type</th>
<th>Import Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAT (Acis)</td>
<td>• Import with Inovate™ advanced modeler</td>
</tr>
<tr>
<td>STEP</td>
<td>• Import with Inovate™ advanced modeler</td>
</tr>
<tr>
<td>TrueSpace</td>
<td>• Import with Inovate™ advanced modeler</td>
</tr>
<tr>
<td>IGES</td>
<td>• Import with Inovate™ advanced modeler</td>
</tr>
<tr>
<td>3DS (3D Studio)</td>
<td>• Import with Inovate™ advanced modeler</td>
</tr>
<tr>
<td>MDL (MotoSim model)</td>
<td>• Add from the CAD Tree window in MotoSim</td>
</tr>
<tr>
<td></td>
<td>• Drag file into MotoSim window</td>
</tr>
<tr>
<td>DXF (AutoCAD)</td>
<td>• Import with Inovate™ advanced modeler</td>
</tr>
<tr>
<td>RWX (RenderWare)</td>
<td>• Add from the CAD Tree window in MotoSim</td>
</tr>
<tr>
<td></td>
<td>• Drag file into MotoSim window</td>
</tr>
</tbody>
</table>

**NOTE:**

- CAD objects converted using advanced modeler are placed in cell according to origin position in original CAD file. If origin of object in original document is in or near the middle of the object, the object appears on the MotoSim screen at or close to the cell’s world origin. If, on the other hand, the object origin in the original document is a significant distance from the object itself, the object appears farther away from world origin when it appears on the MotoSim screen.
- Be sure units are set correctly before importing a file. If units are set incorrectly, the imported object will appear incorrectly sized.
- All imported files have a 3DS (3D Studio) file for editing and an RWX (RenderWare) file for viewing. Both files are located in the models directory. MotoSim manages the files transparent to the user. If only RWX files are supplied, the model cannot be edited. This enables companies to provide equipment models without disclosing proprietary CAD data to the end user.
- If you experience out-of-memory problems while importing IGES files, increase virtual memory setting in Windows.
Importing Models with Inovate™
The External CAD option must be included with MotoSim to import models using Inovate™. To import models using Inovate™, proceed as follows:

1. Create a new model in MotoSim or open an existing advanced model. **You cannot import CAD data into a basic model.**
2. Select Insert Part from the Inovate™ Assembly menu.
3. Start the External CAD option before selecting this function.
4. Select the desired model type.
5. Double-click the desired model file.

The model will be opened in the Inovate™ window for editing. When you are done making desired changes, exit Inovate™ and the model will appear in the MotoSim window.

Importing 3DS Files with MotoSim
To import models with MotoSim, proceed as follows:

1. Create a new model in MotoSim.
2. Click on the "…” button in "File Names" area.
3. Change "Files of type" to "*.3ds".
4. Select the *.3ds file.

6.2.2 Creating Arc Welding Tool Models
The following procedures demonstrate how to make a typical arc welding torch model. These procedures include setting the tool model at TCP, and shows how to change the original model frame using Teacher.

1. Create a new cell named “ArcWeldTorch.”
2. Add a UP6 robot model (Robot · Add Robot...)
3. Click Cad Tree button.
4. Right click Up6-a00 folder and hide the UP6 robot model.
5. Right click the floor and select Property... The FLOOR Property window appears.
6. Set Opacity of Floor to 0.2.
7. In the CAD tree, right click the FLOOR folder and select New Model... .

8. Set the new model name as “ArcWeldTorch.”
9. MotoSim verifies creation of new folder. Click OK.
10. Add a cylinder. In the CAD tree, double click the new ArcWeldTorch folder.
11. Select CYLINDER and click Add.
12. Set values as follows:
   U.Diameter = 80.00
   L.Diameter = 80.00
   Height = 50.00

13. Click Posture... and set values as follows:
    Pxyz(mm) = 0, 0, 0
    Rxyz = 0, 0, 0
14. Click OK in the Posture window.
15. Click OK in the Cylinder window.
16. Click OK in the ArcWeldTorch Add Parts window.
17. Add another cylinder. In the CAD tree, double click ArcWeldTorch folder.
18. Select CYLINDER and click Add.
19. Set values as follows:
   U.Diameter = 30.00
   L.Diameter = 30.00
   Height = 100.00

20. Click Posture... and set values as follows:
    Pxyz(mm) = 100, 0, 30
    Rxyz = 0, -10, 0
21. Click OK in the Posture window.
22. Click OK in the Cylinder window.
23. Click OK in the ArcWeldTorch Add Parts window.

24. Add a third cylinder. In the CAD tree, double click ArcWeldTorch folder.
25. Select CYLINDER and click Add.
26. Set values as follows:
   U.Diameter = 30.00
   L.Diameter = 30.00
   Height = 100.00

27. Click Posture... and set values as follows:
    Pxyz(mm) = 85, 0, 120
    Rxyz = 0, -30, 0
28. Click OK in the Posture window.
29. Click OK in the CYLINDER window.
30. Click OK in the ArcWeldTorch Add Parts window.

31. Add a Cone. In the CAD tree, double click ArcWeldTorch folder.
32. Select CONE2 and click Add.
33. Set values as follows:
   L.Diameter = 30.00
   Height = 30.00
34. Click *Posture...* and set values as follows:
   \[P_{xyz}(\text{mm}) = 155, 0, 0\]
   \[R_{xyz} = 0, -30, 0\]
35. Click *OK* in the Posture window.

36. Click *Posture...* again. Set *Operation* to Relative, and set values as follows:
   \[P_{xyz}(\text{mm}) = 0, 0, 240\]
   \[R_{xyz} = 0, 0, 0\]
   \[Operation = \text{Relative}\]
37. Click *OK* in the Posture window.
38. Click *OK* in the CONE2 window.
39. Click *OK* in the ArcWeldTorch Add Parts window.
40. Add a Box. In the CAD tree, double click ArcWeldTorch folder.
41. Select BOX and click *Add*.
42. Set values as follows:
   \[Width = 150.00\]
   \[Depth = 25.00\]
   \[Height = 25.00\]
43. Click *Posture...* and set values as follows:
   \[P_{xyz}(\text{mm}) = 50, 0, 60\]
   \[R_{xyz} = 0, 0, 0\]
44. Click *OK* in the Posture window.
45. Click *OK* in the BOX window.
46. Click *OK* in the ArcWeldTorch Add Parts window. The arc-welding torch is complete.
47. Select OLP from the Robot tool bar.
48. Select *Teacher* in Operation field.
49. Check the *Pose* box in the “Pick Type” area.
50. Click the cursor at the very tip of the torch to move the teacher.
51. In the CAD tree, right click the Teacher folder and select Set Position....
52. Verify values as follows and click OK.
   Pxyz = 20, 0, 233.827
   Rxyz = 30.001, 0, -90.001
53. Close OLP window.
54. In the CAD tree, right click the ArcWeldTorch folder and select Edit Model....
55. Click Move Org.
56. Select Teacher and click OK.
57. Click OK in the model edit window.
58. In the CAD tree, right click the ArcWeldTorch folder and select Model Attribute · Move Parent....
59. Select the “Up6-a00_tcp” model and click OK.
60. In the CAD tree, right click the Up6-a00 folder and select Hide/See · DModel.
61. Right click the Teacher folder and select Set Position....
62. From the main menu, select Robot · Modify Data · Tool Data...
63. Set Tool Up6-a00(XRC) values the same as Teacher Posture and click OK.
   X = 20, Y = 0, Z = 233.827
   Rx = 30, Ry = 0, Rz = -90
64. In the CAD tree, select Up6-a00_tcp.
65. Right click the Up6-a00_tcp file and select Hide/See · DFrame. The ArcWeldTorch and TCP is now located on the robot.
6.2.3 Device Models for Saving/Loading

A device is a group of models that can be handled as a component. Save Device saves child-model tree information to the target folder. Load Device loads model tree information from the target folder.

All model files are copied to the target cell/Models folder so that the source model files are safe from being modified.

**Save Device**

1. Create a new cell.
2. Select "New Model" and add a model to the cell.
3. Name model names with same prefix (ex. "Dev1").
4. Repeat steps 2 and 3 to add required Models using same prefix.
5. Create parent/child relationships.
6. Set position for each model.
7. Select highest parent (root) model ("Dev1").
8. Select "Save Device...". The Save Device window appears.

9. Use same prefix as device name.
10. You may need to change the model prefix. Click "TAB" key after changing name, so prefix in tree is updated.
11. Click "OK" button.
Load Device

1. Open MotoSim cell.
2. Select "Load Device…". The Load Device window appears.

4. You may need to change the model prefix. Click "TAB" key after changing name, so prefix is updated.
5. Click "OK" button.
6. You should now be able to load a device keeping the same parent/child relationship and relative location to each parent.
6.3 **Simulation Extension**

6.3.1 **Non-Motion Programming**

MotoSim does not have non-motion programming capability. CALL, JUMP, and other non-motion functions must be manually inserted within the job using MRC Editor, XRC Job Editor, or a text editor such as Notepad.

If you just want to add one or two lines, you can also edit a job by opening the Job View dialog.

1. Double-click the line after which you want to insert a line. The Job Edit dialog will appear.
2. Click Add Line.
3. Type the desired job entry in the window.
4. Click OK.

To replace a line, proceed as follows:

1. Double-click the line after which you want to insert a line. The Job Edit dialog will appear.
2. Click Replace.
3. Type the desired job entry in the window.
4. Click OK. The line will be replaced.

6.3.2 **Part Motion Simulation**

Since the robot’s programming language does not include commands for simulating part motion, commands must be manually entered into the job file so that MotoSim can create a realistic simulation.

**Basic Motion Commands**

Simulation of part handling or processes which involve multiple moving parts requires several model operations, namely **copying**, **moving**, and **deleting**. These commands are described in Table 6-2 below.

**NOTE:** “M1” and “M2” represent the names of the models.

<table>
<thead>
<tr>
<th>Command</th>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model Copy 1</td>
<td>‘MDL:DUP M1 M2</td>
<td>Copies M1 to create M2. M2 is written to the cell file</td>
</tr>
<tr>
<td>Model Copy 2</td>
<td>‘MDL:REF M1 M2</td>
<td>Copies M1 to create M2. M2 is not written to the cell file. This reduces cell file size</td>
</tr>
<tr>
<td>Model Move 1</td>
<td>‘MDL:MOV M1 M2</td>
<td>Moves M1 by changing its parent. M2 represents the destination parent model. M1 will have the same local coordinates</td>
</tr>
</tbody>
</table>
MSET6
'MDL:MSET6 M1=10,20,30,0,0,0
Changes the local coordinates of M1 to the specified coordinates.

MPRE6
'MDL:MPRE6 M1=10,20,30,0,0,0
Changes the local coordinates of M1 with multiply the specified matrix value before it.
MPRE6(Matrix Pre multiply with Axis6 type)

MPST6
'MDL:MPST6 M1=10,20,30,0,0,0
Changes the local coordinates of M1 with multiply the specified matrix value after it.
MPST66(Matrix Post multiply with Axis6 type)

EMPTY
'MDL:EMPTY M1
Deletes all shapes in M1 model.

COPY
'MDL:COPY M1 M2
Copies M1 model to M2 model. The original M2 shapes are kept and add M1 shapes to M2 model.
COPY2
'MDL: COPY2 M1 M2
Copies M1 model to M2 model. The original M2 shapes are kept and add M1 shapes to M2 model. M2 shapes are shown at the same location with M1 shapes, even if M2 model is located at the different location from M1 model.

DUMP
'MDL: DUMP M1 M2
Copies M1 model to M2 model. The original M2 shapes are kept and add M1 shapes to M2 model. M2 shapes are shown at the same location with M1 shapes, even if M2 model is located at the different location from M1 model. DUMP is same as COPY2.

LOADF
'MDL: LOADF M1
Loads M1 shapes data from the model file.

SAVEF
'MDL: SAVEF M1
Saves M1 shapes data to the model file.

DELSHAPEBYTYPE
'MDL: DELSHAPEBYTYPE M1 AXIS6
Deletes the specified shape in M1 model by a shape type, such as AXIS6, BOX, and etc.

DELSHAPEBYNAME
'MDL: DELSHAPEBYNAME M1 AAA
Deletes the specified shape in M1 model by a shape name, defined by users.

DELSHAPEBYTYPEXOR
'MDL: DELSHAPEBYTYPEXOR M1 AXIS6
Deletes the specified shape in M1 model by an exclusive shape type, such as AXIS6, BOX, and etc.

DELSHAPEBYNAMEXOR
'MDL: DELSHAPEBYNAMEXOR M1 AAA
Deletes the specified shape in M1 model by an exclusive shape name, defined by users.
Table 6-3 illustrates the use of several motion commands.

**Table 6-3  Sample Robot Job with Motion Commands**

<table>
<thead>
<tr>
<th>Line</th>
<th>Step</th>
<th>Instruction</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td></td>
<td>NOP</td>
<td></td>
</tr>
<tr>
<td>001</td>
<td>001</td>
<td>MOVJ C0000 VJ=90</td>
<td>sd1 - setdown point 1</td>
</tr>
<tr>
<td>002</td>
<td>002</td>
<td>MOVJ C0001 VJ=90</td>
<td>sd2 - Get ready to drop</td>
</tr>
<tr>
<td>003</td>
<td>003</td>
<td>MOVL C0002 V=500</td>
<td>sd3 - Set it down</td>
</tr>
<tr>
<td>004</td>
<td></td>
<td>TIMER T=1.00</td>
<td></td>
</tr>
<tr>
<td>005</td>
<td></td>
<td>'MDL:MOV Boxs Pallet</td>
<td>Changes the parent of “Boxs” to “Pallet”</td>
</tr>
<tr>
<td>006</td>
<td></td>
<td>'MDL:REF Boxs Boxs0</td>
<td>Makes a copy of “Boxs” and calls it “Boxs0”</td>
</tr>
<tr>
<td>007</td>
<td></td>
<td>'MDL:SEE Boxs0</td>
<td>Displays “Boxs0”</td>
</tr>
<tr>
<td>008</td>
<td></td>
<td>'MDL:MOV Boxs SK120_tcp</td>
<td>Moves “Boxs” to the SK120 tool center point</td>
</tr>
<tr>
<td>009</td>
<td></td>
<td>'MDL:AXIS6 Boxs 0,0,63,5,0,0</td>
<td>Moves “Boxs” to local coordinates 0,0,63,5,0,0</td>
</tr>
<tr>
<td>010</td>
<td></td>
<td>'MDL:HID Boxs</td>
<td>Hides “Boxs”</td>
</tr>
<tr>
<td>011</td>
<td></td>
<td>MOVJ C0003 VJ=90</td>
<td>sd4 - Pull up slightly</td>
</tr>
<tr>
<td>012</td>
<td></td>
<td>RET</td>
<td></td>
</tr>
<tr>
<td>013</td>
<td></td>
<td>END</td>
<td></td>
</tr>
</tbody>
</table>

**Simulating Part Transfer**

Transfer of a part requires a “handoff” of the part from one model to another or, in MotoSim terms, changing the part’s parent.

This is usually used to simulate material handling where the robot is picking up objects from one location, moving them to a different location, and then setting the part back down.

A robot typically performs the following sequence of steps when picking up a part:

- Approaches the part
- Moves to a pickup position
- Picks up the part
- Moves the part
- Sets the part down
- Releases the part
- Departs the set down location

To simulate a handoff in a job, perform the following steps:

1. Program the robot moves as you would normally.
2. Save the job.
3. Open the job using MRC Editor, XRC Job Editor, or a text editor such as Notepad. Scroll down to the end of the header information. You will see the program steps for each of the robot’s moves. Locate the program step that is executed right before the robot picks up the part.

4. Insert two blank lines beneath the step identified in Step 3.

5. Type the following text in the two lines (italics represent model names):

```
'MDL:MOV part_name new_parent_name
'MDL:AXIS6 part name=0,0,0,0,0,0
```

**NOTE:** Text should appear as boldface text shown in the example below. Header information has been removed from this job example for clarity.

```
NOP
MOVJ C0000 000 VJ=100.00 CONT
MOVL C0001 001 V=400.0 CONT
'MDL:MOV tool1 sK120_flange
'MDL:AXIS6 tool1=0,0,0,0,0,0
MOVL C0002 002 V=400.0 CONT
RET
END
```

Make sure each line begins with an apostrophe (’). This mark will tell the robot to ignore this line when it runs the job. Steps beginning with the apostrophe are for MotoSim only and have no function when downloaded to the robot controller.

The first line identifies the name of the part you are moving (tool1) and the parent to which you want to connect the part (SK120_flange). The name of the tool and the parent must be typed exactly as they appear in the CAD Tree window. The second line creates a dummy Axis6 to which the part is attached. The Axis6 is what is actually attached to the flange.

**NOTE:** The six zeros (…0,0,0,0,0,0) that follow the second line are position data that will be automatically set when the parent change occurs. There is no need to enter the numbers.

Table 6-7 contains a practical example of how part attachment can be used in programming. This job models Robot 1 picking up a piece and placing it on a press top. The simulation shows the press top rising to represent the press. Robot 2 removes the piece while Robot 1 adds a new piece.

**Table 6-4 Sample Robot Job with Part Transfer Instructions**

<table>
<thead>
<tr>
<th>Line</th>
<th>Step</th>
<th>Instruction</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td></td>
<td>NOP</td>
<td></td>
</tr>
<tr>
<td>001</td>
<td></td>
<td>'MDL:AXIS6 part2=0,6579,990,20,0,0</td>
<td>Moves part2 to this position</td>
</tr>
<tr>
<td>002</td>
<td></td>
<td>'MDL:AXIS6 part=-0.41,21,986,0,0,180</td>
<td>Moves part to this position</td>
</tr>
<tr>
<td>003</td>
<td>001</td>
<td>MOVJ VJ=100.00</td>
<td></td>
</tr>
<tr>
<td>004</td>
<td>002</td>
<td>MOVJ VJ=80.00 PL=0</td>
<td></td>
</tr>
<tr>
<td>005</td>
<td></td>
<td>TIMER T=0.2</td>
<td></td>
</tr>
</tbody>
</table>
The HIDSEE command toggles a single model view using I/O bit value. The sample script follows:

`'MDL:HIDSEE Tube00 #[IOBIT\Tube00\INPUT01\01]`

Tube00 is HID (hidden), if I/O value of Tube00 model equals 0.
Tube00 is SEE (seen), if I/O value of Tube00 model equals 1.

See HideSeeTube.mcl in “Examples\Using Model Drive\HideSeeModel” holder.

Indexed Model Name with HIDSEE Command

The HID and SEE commands can be used to index a group of models. The eight bits in the I/O byte represent a binary number (0-255). Each of these numbers can be tied to a particular model. The command indexes the model using the I/O byte value. For example, in the sample script below, the HID and SEE commands will only effect the Tube model number that equals the IOBYTE value in the brackets.

<table>
<thead>
<tr>
<th>Line</th>
<th>Step</th>
<th>Instruction</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>006</td>
<td></td>
<td>'MDL:MOV part2 K506_tcp</td>
<td>Attaches part2 to the robot’s tool center point</td>
</tr>
<tr>
<td>007</td>
<td>003</td>
<td>MOVJ VJ=100.00</td>
<td></td>
</tr>
<tr>
<td>008</td>
<td></td>
<td>'MDL:AXIS6 press-top=0,0,0,0,0</td>
<td>Moves press top to this position, at 0,0,0,0,0 it will most likely take it out of view</td>
</tr>
<tr>
<td>009</td>
<td></td>
<td>DOUT OT#01=1</td>
<td>Indication to second robot</td>
</tr>
<tr>
<td>010</td>
<td>004</td>
<td>MOVJ VJ=100.00</td>
<td>Allows second robot to begin a job</td>
</tr>
<tr>
<td>011</td>
<td></td>
<td>DOUT OT#01=0</td>
<td>Waits until second robot is done</td>
</tr>
<tr>
<td>012</td>
<td></td>
<td>WAIT IN#01=1</td>
<td></td>
</tr>
<tr>
<td>013</td>
<td>005</td>
<td>MOVJ VJ=100.00</td>
<td></td>
</tr>
<tr>
<td>014</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>015</td>
<td></td>
<td>TIMER T=0.2</td>
<td></td>
</tr>
<tr>
<td>016</td>
<td></td>
<td>'MDL:MOV part2 world</td>
<td></td>
</tr>
<tr>
<td>017</td>
<td>006</td>
<td>MOVJ VJ=80.00 PL=0</td>
<td></td>
</tr>
<tr>
<td>018</td>
<td></td>
<td>'MDL:AXIS6 press-top= 0,0,-900,0,0,0</td>
<td>Moves press top up to this position</td>
</tr>
<tr>
<td>019</td>
<td>007</td>
<td>MOVJ VJ=100.00</td>
<td></td>
</tr>
<tr>
<td>020</td>
<td></td>
<td>'MDL:AXIS6 part=0,6579,990,20,0,0</td>
<td>Moves part to this position</td>
</tr>
<tr>
<td>021</td>
<td>008</td>
<td>MOVJ VJ=100.00</td>
<td></td>
</tr>
<tr>
<td>022</td>
<td>009</td>
<td>MOVJ VJ=80.00 PL=0</td>
<td></td>
</tr>
<tr>
<td>023</td>
<td></td>
<td>END</td>
<td></td>
</tr>
</tbody>
</table>
Tube00 is indexed, if the value of byte \([\text{IOBYTE}\backslash\text{Tube00}\backslash\text{INPUT01}\backslash0]\) is 0.
Tube01 is indexed, if the value of byte \([\text{IOBYTE}\backslash\text{Tube00}\backslash\text{INPUT01}\backslash1]\) is 1.
See HideSeeTube2.mcl in "Examples\Using Model Drive\HideSeeModel" holder. There is a small difference between \([\text{IOBYTE}\backslash\text{Tube00}\backslash\text{INPUT01}\backslash0]\) and \([\text{IOBYTE}\backslash\text{Tube00}\backslash\text{INPUT01}\backslash1]\)
\([\text{IOBYTE}\backslash\text{Tube00}\backslash\text{INPUT01}\backslash0]\) indexes Tube01.
\([\text{IOBYTE}\backslash\text{Tube00}\backslash\text{INPUT01}\backslash1]\) indexes Tube1.
So that
\([\text{IOBYTE}\backslash\text{Tube00}\backslash\text{INPUT01}\backslash0]\) index Tube001.

### 6.3.3 Manipulator Features

Equipment such as indexing tables, conveyor belts, diverters, doors, and other equipment that moves but is not servo controlled must be modeled using motion DLL’s. These types of motion are controlled by I/O in the real world but they cannot be represented in MotoSim without a DLL to tell MotoSim when and how to move the object.

DLL scripting is very complex and is not documented in this manual. This section will address how to tie existing linear, indexing, and RM2 DLL’s into your job to simulate motion.

**Linear32 Motion**

Linear32.DLL (Linear DLL) is used to simulate Servo-Gun or any linear motion related to robot axis.

1. Create the models in MotoSim that you want the DLL to control. Make sure the model is attached to an Axis6 model. Create one if necessary. Make a note of the Axis6 number. You will need it later to link the DLL to the part.
2. Open the cell file with Notepad.
3. Go to the section of the file that describes the part you want to move.
4. Enter the TASK command as shown in the example below.

   ```plaintext
   TASK=LINER32.DLL ROBOT=Sk6 LINK=6 FUNCA1=1.0 FUNCA2=10.0 FUNCB=31.0 MODEL=conveyer AXIS6NO=0;
   ```

The syntax for linear motion is:

- **TASK**—The DLL filename.
- **ROBOT**—The robot name exactly as it appears in the CAD Tree window.
- **LINK**—A robot axis used to trigger the DLL. In the “manipulator” example above, this axis is set to “6.” Robot axes are numbered 0-5 and correspond to the S, L, U, R, B, T axes of the robot. Axis number 6 is the first available external axis.

It is possible to create an empty robot model file and use all six axes to control up to six linear motion DLL’s. In this case, you would set the axis values of the six TASK commands to 0 through 5.
FUNCA1—A number value representing how many millimeters the model will be moved each time FUNCA2 cycles.

FUNCA2—A number value representing how many pulse counts (time) will elapse while the equipment is moving the distance specified by FUNCA1. FUNCA1 does not wait for FUNCA2 to cycle. Instead, the two FUNCA variables work together to form a millimeters/time rate of motion.

FUNCB—A number value representing the starting position in millimeters of the moving part from its Axis6 origin.

MODEL—The name of the model that will be moved according to values defined by FUNCA1, FUNCA2, and FUNCB.

AXIS6NO—The number of the Axis6 to which your model is attached. 0,1,2,3,4, and 5 correspond to the X,Y, Z, Rx, Ry, and Rz axes respectively.

Index32 Manipulator Operations

Index32.DLL (Index DLL) is used to simulate rotating indexing positioners, hinged doors, diverters, etc.

1. Create the models in MotoSim that the DLL will control. Make sure model is attached to an Axis6 model. Create one if necessary.
2. Open cell file with Notepad.
3. Go to section of file that describes the part to be moved.
4. Enter TASK command as shown in the example below.
   — Index32.DLL performs Index32 Manipulator
   — This DLL is family DLL of MotoSim.Exe.
   — This DLL is installed by writing script as follows in the Cel File.
     See "Using Index32" sample cells.

In the "Index32Sam01.cel"

This example shows how to link Robot I/O automatically.

Note: This I/O script format is retained in this version for backwards compatibility. However, this I/O script format may not be supported in future versions.

```plaintext
MANIPULATOR
{
    TASK=Index32.DLL MODEL=Box02 STROKE=180 TIME=3 AXIS6NO=5
    ROBOT=Up6 ROBOT(OUTA)=1 ROBOT(OUTB)=2 ROBOT(INA)=1 ROBOT(INB)=2;
}
```

In the "Index32Sam02.cel"

This example shows the simplest description of Index32.DLL.

In this case, create I/O-link information between the model and robot. Refer to Section 4.4.17.

```plaintext
MANIPULATOR
{
    TASK=Index32.DLL MODEL=Box02 STROKE=180 TIME=3 AXIS6NO=5;
}
```

In the "Index32Sam03.cel"

This example shows how to link Robot I/O automatically.

Model moves Z-direction by 100(mm).
**Note:** This I/O script format is retained in this version for backwards compatibility. However, this I/O script format may not be supported in future versions.

```plaintext
MANIPULATOR
{
    TASK=Index32.DLL MODEL=Box02 STROKE=100 TIME=3 AXIS6NO=2
    ROBOT=Up6 ROBOT(OUTA)=1 ROBOT(OUTB)=2 ROBOT(INA)=1 ROBOT(INB)=2;
}
```

In the "Index32Sam04.cel"

This example shows how to change color according to model location.

**Note:** This I/O script format is retained in this version for backwards compatibility. However, this I/O script format may not be supported in future versions.

```plaintext
MANIPULATOR
{
    TASK=Index32.DLL MODEL=Box02 STROKE=100 TIME=3 AXIS6NO=2
    ROBOT=Up6 ROBOT(OUTA)=1 ROBOT(OUTB)=2 ROBOT(INA)=1 ROBOT(INB)=2
    IONAME=Box02 COLOR(MOVE)=RGB(255,0,0) COLOR(OUT1)=RGB(0,255,0) COLOR(OUT2)=RGB(0,0,255);
}
```

In the "Index32Sam05.cel"

This example shows different usages of Index32.DLL

```plaintext
MANIPULATOR
{
    TASK=Index32.DLL MODEL=Box02 STROKE=100 TIME=3 AXIS6NO=0
    IONAME=Box02 COLOR(MOVE)=RGB(255,0,0);
    TASK=Index32.DLL MODEL=Box02 STROKE=100 TIME=3 AXIS6NO=1
    IONAME=Box02x COLOR(MOVE)=RGB(255,0,0);
}
```

In the "Index32Sam06.cel"

This example illustrates conveyer simulation.

```plaintext
MANIPULATOR
{
    TASK=Index32.DLL MODEL=Box02 STROKE=1000 TIME=10 AXIS6NO=0;
}
```

### Table 6-5   Reference

<table>
<thead>
<tr>
<th>Input 1</th>
<th>Command to go 0 distance (command-1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input 2</td>
<td>Command to go STROKE distance (command-2)</td>
</tr>
<tr>
<td>Output 1</td>
<td>status of location 0 (status-1)</td>
</tr>
<tr>
<td>Output 2</td>
<td>status of location STROKE (status-2)</td>
</tr>
<tr>
<td>Input 3</td>
<td>Command to reset distance (Conveyer function)(command-3)</td>
</tr>
<tr>
<td>MODEL=Box02</td>
<td>Specifies Model name</td>
</tr>
<tr>
<td>STROKE=180</td>
<td>Specifies stroke of moving in degrees</td>
</tr>
<tr>
<td>TIME=3</td>
<td>Specifies time of moving in seconds</td>
</tr>
</tbody>
</table>
RM2 Motion

RM2X32.DLL (RM2 DLL) supports Motoman RM2-series positioners.

1. Place the DLL file in the MotoSim directory.
2. The cell file must be edited.
   This DLL, like other DLL's, must be identified in the .cel file in the manipulator section and formatted as follows:
   **TASK**=Rm2x32.dll **RM2NAME**=Rm2x500 **DRIVERNAME**=robot_ex1 **SIDEANGLE**=93.24
   - **TASK** specifies which dll to use, in the case of the RM2 positioner, it should be “RM2x32.dll.”
   - **RM2NAME** specifies which positioner is connected to the specified robot. The name should appear exactly as displayed in the CAD tree (necessary in case the cell has more than one positioner or robot).
   - **DRIVERNAME** specifies which external axis drives the RM2 positioner. The name should appear exactly as displayed in the CAD tree. This external axis should have sufficient pulse deviation to allow the positioner to sweep 180°.

When creating a new positioner model, there are several model naming schemes that must be followed closely. In order for the DLL to move the headstocks, the sweep axis, the sweep lock pins, or the servo motor each has to be created as a separate model. These models then need to be named specifically. All of these models need to contain the RM2 name as was entered into the .cel file.

### Table 6-5  Reference

<table>
<thead>
<tr>
<th>AXIS6NO=5</th>
<th>Specifies direction of moving. 0, 1, 2, 3, 4, and 5 correspond to the X, Y, Z, Rx, Ry, and Rz axes respectively</th>
</tr>
</thead>
<tbody>
<tr>
<td>ROBOT=Up6</td>
<td>Specifies Robot linked with I/O. This I/O script format may not be supported in future versions.</td>
</tr>
<tr>
<td>ROBOT(OUTA)=1</td>
<td>Specifies I/O address that is linked with command-1 of model. This I/O script format may not be supported in future versions.</td>
</tr>
<tr>
<td>ROBOT(OUTB)=2</td>
<td>Specifies I/O address that is linked with command-2 of model. This I/O script format may not be supported in future versions.</td>
</tr>
<tr>
<td>ROBOT(INA)=1</td>
<td>Specifies I/O address that is linked with status-1 of model. This I/O script format may not be supported in future versions.</td>
</tr>
<tr>
<td>ROBOT(INB)=2</td>
<td>Specifies I/O address that is linked with status-2 of model. This I/O script format may not be supported in future versions.</td>
</tr>
<tr>
<td>IONAME=Box02</td>
<td>Specifies I/O object name</td>
</tr>
<tr>
<td>COLOR(MOVE)=RG B(255,0,0)</td>
<td>Specifies color of Model at moving</td>
</tr>
<tr>
<td>COLOR(OUT1)=RG B(0,255,0)</td>
<td>Specifies color of Model at status-1</td>
</tr>
<tr>
<td>COLOR(OUT2)=RG B(0,0,255)</td>
<td>Specifies color of Model at status-2</td>
</tr>
</tbody>
</table>
Table 6-6  Model Names for RM2 Motion

<table>
<thead>
<tr>
<th>RM2NAMExA</th>
<th>Rotates when the headstock is rotated</th>
</tr>
</thead>
<tbody>
<tr>
<td>RM2NAMExB</td>
<td>Rotates when the headstock is rotated</td>
</tr>
<tr>
<td>RM2NAMExC</td>
<td>Moves when the positioner sweeps from side to side</td>
</tr>
<tr>
<td>RM2NAMExCxBase</td>
<td>Rotation-base of RM2NAMExC</td>
</tr>
<tr>
<td>RM2NAMExCxPin</td>
<td>Lock pin model that locks headstock A</td>
</tr>
<tr>
<td>RM2NAMExCxPinxR</td>
<td>Lock pin model that locks the main sweep arm</td>
</tr>
<tr>
<td>RM2NAMExCxBase1</td>
<td>Rotation-base of headstock</td>
</tr>
</tbody>
</table>

The I/O control of this RM2.DLL is the most important part of the control. RM2x32DLL uses 5 inputs and 7 outputs.

Table 6-7  I/O

<table>
<thead>
<tr>
<th>IN#001</th>
<th>Cycle Latched. This input needs to be ON before changing servo axis</th>
</tr>
</thead>
<tbody>
<tr>
<td>IN#002</td>
<td>Withdraw Servo. This input is a command to withdraw the servo.</td>
</tr>
<tr>
<td>IN#003</td>
<td>Advance Servo. This input is a command to advance the servo.</td>
</tr>
<tr>
<td>IN#004</td>
<td>Move Sweep (or Index) Lock to On. This input is a command to move the sweep (or index) arm lock pin to the on position.</td>
</tr>
<tr>
<td>IN#005</td>
<td>Move Sweep (or Index) Lock to Off. This input is a command to move the sweep (or index) arm lock pin to the off position.</td>
</tr>
<tr>
<td>OUT#001</td>
<td>Cycle Start. This output is same status as IN#001. This tells the RM2 working on Withdraw operation.</td>
</tr>
<tr>
<td>OUT#003</td>
<td>A Lock On. This Out tells Lock of headstock A is ON.</td>
</tr>
<tr>
<td>OUT#004</td>
<td>B Lock On. This Out tells Lock of headstock B is ON.</td>
</tr>
<tr>
<td>OUT#005</td>
<td>Withdraw Servo ON. This Out tells Servo of Withdraw is ON.</td>
</tr>
<tr>
<td>OUT#006</td>
<td>Advanced Servo ON. This Out tells Servo of headstock is ON.</td>
</tr>
<tr>
<td>OUT#007</td>
<td>Sweep (or Index) Lock is On. This output indicates the Sweep (or Index) Lock is On</td>
</tr>
<tr>
<td>OUT#008</td>
<td>Sweep (or Index) Lock is Off. This output indicates the Sweep (or Index) Lock is Off</td>
</tr>
</tbody>
</table>
In the "Rm2x500.mcl" cell in example holder, using "CALL JOB: WITH2A" to operate Withdraw is the best way.
You may operate same things by manually.

**Changing Servo from Headstock B to Withdraw.**
1. Set Ex1 axis pulse as "50000".
2. Set "Rm2x500xCxBase1" location as "(0,0,0,0,0,0)".
3. Set RobotOUT#9 as ON.
4. Set RobotOUT#10 as OFF.
5. Operate Ex1 axis to move Withdraw Servo.

**Changing Servo from Withdraw to Headstock A.**
1. Set Ex1 pulse as "-50000".
2. Set "Rm2x500xCxBase1" location as "(0,0,38.1,0,0,0)".
3. Set RobotOUT#9 as OFF.
4. Set RobotOUT#10 as ON.
5. Operate Ex1 axis to move to Headstock A Servo.

**Note:** Here Z=38.1(mm) is the stroke specification of RM2 positioner.

**IO assignment between Rm2 and robot**
There are IO connections between Rm2 and robot in Rm2x500.mcl file.
Robot(OUT #1)-Rm2(IN #1)
Robot(OUT #9)-Rm2(IN #2)
Robot(OUT #10)-Rm2(IN #3)
Robot(OUT #11)-Rm2(IN #4)
Robot(OUT #12)-Rm2(IN #5)

Robot(IN #1)-Rm2(OUT #1)
Robot(IN #9)-Rm2(OUT #3)
Robot(IN #10)-Rm2(OUT #4)
Robot(IN #11)-Rm2(OUT #5)
Robot(IN #12)-Rm2(OUT #6)
Robot(IN #13)-Rm2(OUT #7)
Robot(IN #14)-Rm2(OUT #8)

**Controlling external axis**
Use "WITH2A.JBI" job for sweeping from B to A.
Use "WITH2B.JBI" job for sweeping from A to B.

There are sample programs in cell.
Execute "WITHDRAW.JBI" for checking basically motion.
Programming A side
Use "WITH2A.JBI" for locating A-side on robot.
Make new job for teaching.
Teach position and program desirable sequence.
Call this job from master job.

Programming B side
Use "WITH2B.JBI" for locating B-side on robot.
Make new job for teaching.
Teach position and program desirable sequence.
Call this job from master job.

BSIDEANGLE
Set BSIDEANGLE value of the “Task=” line to the initial angle of the large rotation axis. The default value is 0.
### Table 6-8  Sample RM2 Sweep to A Job (WITH2A.JBI)

<table>
<thead>
<tr>
<th>Job Commands</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NOP</td>
<td></td>
</tr>
<tr>
<td>MOVJ C0000 EC0000 VJ=65.00 PL=0</td>
<td></td>
</tr>
<tr>
<td>DOUT OT#(1)=ON</td>
<td></td>
</tr>
<tr>
<td>WAIT IN#(1)=ON</td>
<td></td>
</tr>
<tr>
<td>WAIT IN#(9)=ON</td>
<td></td>
</tr>
<tr>
<td>DOUT OT#(10)=OFF</td>
<td></td>
</tr>
<tr>
<td>DOUT OT#(9)=ON</td>
<td></td>
</tr>
<tr>
<td>WAIT IN#(11)=ON</td>
<td></td>
</tr>
<tr>
<td>DOUT OT#(11)=OFF</td>
<td>Turn off Withdraw Lock On.</td>
</tr>
<tr>
<td>DOUT OT#(12)=ON</td>
<td>Turn on Withdraw Lock Off.</td>
</tr>
<tr>
<td>WAIT IN#(14)=ON</td>
<td>Wait for Withdraw Lock Off being ON.</td>
</tr>
<tr>
<td>WAIT IN#(10)=ON</td>
<td></td>
</tr>
<tr>
<td>MOVJ C0001 EC0001 VJ=65.00 PL=0</td>
<td></td>
</tr>
<tr>
<td>DOUT OT#(12)=OFF</td>
<td>Turn off Withdraw Lock Off.</td>
</tr>
<tr>
<td>DOUT OT#(11)=ON</td>
<td>Turn on Withdraw Lock On.</td>
</tr>
<tr>
<td>WAIT IN#(13)=ON</td>
<td></td>
</tr>
<tr>
<td>DOUT OT#(9)=OFF</td>
<td></td>
</tr>
<tr>
<td>DOUT OT#(10)=ON</td>
<td></td>
</tr>
<tr>
<td>WAIT IN#(12)=ON</td>
<td></td>
</tr>
<tr>
<td>WAIT IN#(9)=OFF</td>
<td></td>
</tr>
<tr>
<td>DOUT OT#(1)=OFF</td>
<td></td>
</tr>
<tr>
<td>END</td>
<td></td>
</tr>
</tbody>
</table>
### Table 6-9  Sample RM2 Sweep to B Job (WITH2B.JBI)

<table>
<thead>
<tr>
<th>Job Commands</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NOP</td>
<td></td>
</tr>
<tr>
<td>MOVJ C0000 EC0000 VJ=65.00 PL=0</td>
<td></td>
</tr>
<tr>
<td>DOUT OT#(1)=ON</td>
<td></td>
</tr>
<tr>
<td>WAIT IN#(1)=ON</td>
<td></td>
</tr>
<tr>
<td>WAIT IN#(10)=ON</td>
<td></td>
</tr>
<tr>
<td>DOUT OT#(10)=OFF</td>
<td></td>
</tr>
<tr>
<td>DOUT OT#(9)=ON</td>
<td></td>
</tr>
<tr>
<td>WAIT IN#(11)=ON</td>
<td></td>
</tr>
<tr>
<td>DOUT OT#(11)=OFF</td>
<td></td>
</tr>
<tr>
<td>DOUT OT#(12)=ON</td>
<td></td>
</tr>
<tr>
<td>WAIT IN#(14)=ON</td>
<td></td>
</tr>
<tr>
<td>WAIT IN#(9)=ON</td>
<td></td>
</tr>
<tr>
<td>MOVJ C0001 EC0001 VJ=65.00 PL=0</td>
<td></td>
</tr>
<tr>
<td>DOUT OT#(12)=OFF</td>
<td></td>
</tr>
<tr>
<td>DOUT OT#(11)=ON</td>
<td></td>
</tr>
<tr>
<td>WAIT IN#(13)=ON</td>
<td></td>
</tr>
<tr>
<td>DOUT OT#(9)=OFF</td>
<td></td>
</tr>
<tr>
<td>DOUT OT#(10)=ON</td>
<td></td>
</tr>
<tr>
<td>WAIT IN#(12)=ON</td>
<td></td>
</tr>
<tr>
<td>WAIT IN#(10)=OFF</td>
<td></td>
</tr>
<tr>
<td>DOUT OT#(1)=OFF</td>
<td></td>
</tr>
<tr>
<td>END</td>
<td></td>
</tr>
</tbody>
</table>
All of these models must be created at specific locations. The DLL is hard coded to recognize locations as either engaged and disengaged.

- The sweep model Rz-position should be 180 when robot is working side B.
- The HeadALock and HeadBLock Z-positions should equal zero when locked into the headstock to prevent rotation.
- Both HeadA and HeadB must be modeled so they rotate about the Z axis.
- The Servo model must be created so that its Y-position equals zero when engaged to one of the headstocks.
- The sweeplock1 and sweeplock2 pins must be modeled so that their Y-positions equal zero when engaged in the sweep axis.

Zero pulse counts is set in the middle of the rotation (90 degrees). It is critical that these pulse counts be accurately recorded because they determine when the positioner is in the correct position for locking pins and sweep-lock pins to engage and disengage. The maximum allowable deviation is 15 pulse counts.

**NOTE:** Despite the allowable deviation, the exact point above should be programmed and a position level of 0 should be used to make sure the ideal position is reached.

The sweep model should be modeled so that the rotation around Z is 180 degrees when at Side B and it should be 0 degrees when at Side A. Rotation can be adjusted by adding a dummy model to the sweep model. In this manner the sweep model can be set to 180 degrees and adjusted into position by rotating the dummy model. (The DLL checks the position of the models to determine the active side of the positioner so the sweep model position is critical.)

Headstocks are created to permit all rotation within its parameters. The DLL will allow the headstocks to rotate until a softlimit on the motor is reached. In order to engage the servo motor into the headstocks, the motor must be positioned at a specific location. To engage the Side B headstock, the position of the motor must be to its positive, upper limit (23800, 47600, and 56400 respectively for the RM2-250, -500, and -750). To engage the Side A headstock, the motor must be positioned to the lowermost limit. A 15-pulse deviation is allowed but the positions should be taught with a position level of zero.

**NOTE:** A change in I/O will not register with the DLL unless the robot or one other external axis is moved. Include a Sweep-to-A and a Sweep-to-B instruction in your job. An example sweep job is shown on the following page.
CAUTION!

Motoman reserves the right to change positioner specifications without notice. Refer to the documentation shipped with your RM2 positioner before downloading jobs. The following job is intended for illustrative purposes only.

Table 6-10  Sample RM2 Sweep-to-A Job

<table>
<thead>
<tr>
<th>Line</th>
<th>Step</th>
<th>Instruction</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td>000</td>
<td>NOP</td>
<td></td>
</tr>
<tr>
<td>001</td>
<td>001</td>
<td>MOV J 0000</td>
<td>Robot to safe position, headstock to home.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EC 0000</td>
<td>VJ=65.00 PL=0</td>
</tr>
<tr>
<td>002</td>
<td></td>
<td>DOUT OT(1)</td>
<td>Turn on Station Ready lamp.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ON</td>
<td></td>
</tr>
<tr>
<td>003</td>
<td></td>
<td>WAIT IN(1)=ON</td>
<td>Wait for Cycle Start palm buttons.</td>
</tr>
<tr>
<td>004</td>
<td></td>
<td>WAIT IN(9)=ON</td>
<td>Verify engagement of locking pins on Side A.</td>
</tr>
<tr>
<td>005</td>
<td></td>
<td>DOUT OT(10) OFF</td>
<td>Turn off ADVANCE SERVO signal.</td>
</tr>
<tr>
<td>006</td>
<td></td>
<td>DOUT OT(9) ON</td>
<td>Turn on WITHDRAW SERVO signal to retract servomotor.</td>
</tr>
<tr>
<td>007</td>
<td></td>
<td>WAIT IN(11)=ON</td>
<td>Wait for servomotor and drive to be withdrawn.</td>
</tr>
<tr>
<td>008</td>
<td></td>
<td>DOUT OT(11) OFF</td>
<td>Turn off “advance sweep arm lock pins” signal.</td>
</tr>
<tr>
<td>009</td>
<td></td>
<td>DOUT OT(12)=ON</td>
<td>Turn on “withdraw sweep arm lock pins” signal.</td>
</tr>
<tr>
<td>010</td>
<td></td>
<td>WAIT IN(14)=ON</td>
<td>Wait for disengagement of sweep arm lock pins.</td>
</tr>
<tr>
<td>011</td>
<td></td>
<td>WAIT IN(10)=ON</td>
<td>Wait for engagement of B side headstock lock pins.</td>
</tr>
<tr>
<td>012</td>
<td>002</td>
<td>MOV J 0001</td>
<td>External axis indexes until side A lines up with</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EC 0001</td>
<td>drive unit and index arm lock pins.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>VJ=65.00 CONT PL=0</td>
<td></td>
</tr>
<tr>
<td>013</td>
<td></td>
<td>DOUT OT(12)=OFF</td>
<td>Turn off the &quot;withdraw sweep arm lock pins&quot; signal.</td>
</tr>
<tr>
<td>014</td>
<td></td>
<td>DOUT OT(11) ON</td>
<td>Turn on the &quot;advance sweep arm lock pins&quot; signal.</td>
</tr>
<tr>
<td>015</td>
<td></td>
<td>WAIT IN(13)=ON</td>
<td>Wait for the sweep arm to be locked</td>
</tr>
<tr>
<td>016</td>
<td></td>
<td>DOUT OT(9) OFF</td>
<td>Turn off the signal to Withdraw the servomotor and is drive unit.</td>
</tr>
<tr>
<td>017</td>
<td></td>
<td>DOUT OT(10) ON</td>
<td>Turn on the signal to Advance the servomotor and is drive unit.</td>
</tr>
<tr>
<td>018</td>
<td></td>
<td>WAIT IN(12)=ON</td>
<td>Wait for the Servomotor and its drive unit to be advanced.</td>
</tr>
<tr>
<td>019</td>
<td></td>
<td>WAIT IN(9)=OFF</td>
<td>Wait for the A headstock locking pins to be disengaged.</td>
</tr>
<tr>
<td>020</td>
<td></td>
<td>DOUT OT(1) OFF</td>
<td>Turn off the Station Ready lamp.</td>
</tr>
<tr>
<td>021</td>
<td></td>
<td>END</td>
<td></td>
</tr>
</tbody>
</table>
Table 6-11  I/O

<table>
<thead>
<tr>
<th>RM2 I/O #</th>
<th>Robot I/O #</th>
<th>Signal Name</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input #1</td>
<td>Output #1</td>
<td>Cycle Latched</td>
<td>Must be on before External Axis can move.</td>
</tr>
<tr>
<td>Input #2</td>
<td>Output #9</td>
<td>Withdraw Servo</td>
<td>Command to withdraw Servomotor, and its drive.</td>
</tr>
<tr>
<td>Input #3</td>
<td>Output #10</td>
<td>Advance Servo</td>
<td>Command to advance Servomotor, and its drive.</td>
</tr>
<tr>
<td>Input #4</td>
<td>Output #11</td>
<td>Advance Lock Pin</td>
<td>Command to engage the sweep arm lock pin.</td>
</tr>
<tr>
<td>Input #5</td>
<td>Output #12</td>
<td>Withdraw Lock Pin</td>
<td>Command to disengage the sweep arm lock pin.</td>
</tr>
<tr>
<td>Output #1</td>
<td>Input #1</td>
<td>Cycle Start</td>
<td>This output is the same as Input #1. This means the RM2 is working.</td>
</tr>
<tr>
<td>Output #3</td>
<td>Input #9</td>
<td>A Lock On</td>
<td>Indicates A headstock is locked.</td>
</tr>
<tr>
<td>Output #4</td>
<td>Input #10</td>
<td>B Lock On</td>
<td>Indicates B headstock is locked.</td>
</tr>
<tr>
<td>Output #5</td>
<td>Input #11</td>
<td>Servo Withdrawn</td>
<td>Indicates Servomotor and its drive are withdrawn.</td>
</tr>
<tr>
<td>Output #6</td>
<td>Input #12</td>
<td>Servo Advanced</td>
<td>Indicates Servomotor and its drive are advanced.</td>
</tr>
<tr>
<td>Output #7</td>
<td>Input #13</td>
<td>Arm Lock On</td>
<td>Indicates sweep arm is locked.</td>
</tr>
<tr>
<td>Output #8</td>
<td>Input #14</td>
<td>Arm Lock Off</td>
<td>Indicates sweep arm is not locked.</td>
</tr>
</tbody>
</table>

**NOTE:** The sweep positions are programmed in position variables. They are factory set and should not require editing. Care should be taken not to use these variables in other programs. You can edit position variables from the file display.
**MsimIoDecoder DLL**
MsimIoDecoder converts Input IO data to Output data.
See example:
"\Examples\Using Model Drive\HideSeeModel\MsimIoDecoder.mcl"

**Scripts in Cel file**
MANIPULATOR

```
{  
    TASK=MSimIoDecoder.DLL MODEL=TubeBox  
    FUNCTION=DECODExLONG IONAMEIN=INPUT01  
    IONAMEOUT=OUTPUT01 COLORCHANGE=YES  
}
```

**How MsimIoDecoder works**
MsimIoDecoder gets Input (INPUT01) of “TubeBox” model and converts this value to Output (OUTPUT01).
If input is 0x01, output will be turn #1 on and other off.
If input is 0x02, output will be turn #2 on and other off.
If input is 0x03, output will be turn #3 on and other off.
If input is 0x04, output will be turn #4 on and other off.
If input is 0x05, output will be turn #5 on and other off.
If input is 0x06, output will be turn #6 on and other off.
With this function, you can turn on only one output according to the input value.
See changes IO dialog box of Sp100_RIO and TubeBox IO.
Play back job and notice the changes.

Robot I/O
Tube Box I/O dialog box
6.3.4 Setting Step-Motion for Off-Line Programming

The "Step Motion" function can help recognize Robot movement during off-line programming. It is interpolated from the current posture to the target posture.

This function initiates when you push "<<", ">>", or "|<" on the Robot Tool bar.

To setup step motion for off-line programming select Robot > Step Motion from the Tools dropdown window.

The Set Step Motion window appears.

The "Jump to" radio button indicates that robot motion isn’t interpolated from the current posture to the target posture. Move immediately.

The "Move to" radio button indicates that robot motion is interpolated from the current posture to the target posture.

The number is the number of interpolation.

A smaller number indicates faster robot movement.

A larger number indicates slower robot movement.

This setting will be saved in cell file.
### 6.4 External Axis

#### 6.4.1 Modeling External Axes

Modeling external axes is necessary to simulate the motion of positioners that have full servo control (not index tables) such as headstock/tailstock positioners.

**NOTE:** *Indexing tables and other nonservo-driven devices must be modeled using DLL motion control. See Section 6.5, Using DLL's to Model Motion, for information on modeling motion in MotoSim.*

In order to model an external axis, the parameter file for the system must contain data for external axis control. If you purchased your system with a Motoman external axis positioner, your parameter file will already contain the appropriate data.

To model a positioner or other equipment with an external axis:

1. Model all non-moving components.
2. Locate and select the external axis in the CAD Tree window. External axes have an “_ex[n]” ending (n represents the number of the external axis).
3. Perform a parent change on the external axis dummy to make the positioner base (nonmoving parts) the parent.
4. Click New to open the Add Model dialog.
5. Attach all moving parts to the external axis dummy. Give it a name like “Tooling plate” or “Headstock.”
6. Create the moving parts of your positioner. Make sure you create the parts with the external axis dummy as the parent.

By attaching the moving components to the external axis dummy, you can now move all the moving parts in unison by moving only the dummy.

The dummy model is now attached to the external axis. If you move the external axis with the programming pendant, the dummy model (and all of its attached parts) will also move.

Click Ext on the programming pendant during programming to move the external axis. You will see that all of your modeled moving parts rotate/move while the positioner’s non-moving components remain stationary. This type of modeling can be used to simulate any motion that is servo controlled.

**NOTE:** *If, when you open your cell file, the external axis has moved to the world origin, perform the following steps to fix the cell file. Rename the “_ex[n]” model to something else (e.g. “robot _ex1A”). Create a dummy model using the same name as the external axis (e.g. “robot_ex1”), position the dummy model in the external axis position, and delete the old model (e.g.”robot_ex1A”). Save the cell file. Close and reopen the cell file. The external axis will appear in the correct location when the cell file is reopened.*
6.4.2 Motoman Parameter All.PRM Editor

MotoPrmAll.exe provides basic file editing for the XRC and MRC controllers. Parameters can be edited using clipboard techniques. The ALL.PRM structure can also be viewed as a tree diagram. External axis data can be added to normal robot ALL.PRM files in the XRC controller only. Special dialog boxes allow you to set up Group Coordination information like base axis and external axis.

⚠️ DANGER!
Do not download any ALL.PRM files that have been edited with this software to the controller. This software will not prevent mechanical damage if ALL.PRM files are loaded to the controller.

Viewing ALL.PRM Files
1. Select File · Open from the main menu.
2. Select //RC(8050) from left window.
3. The data included in //RC is shown in the right window.

The “8050” indicates the number of data in this file. The View Size field selects the amount of this data shown in the window. “20” shows 20 of the 8050 total. Selecting a larger number redraws the window to accommodate the larger data set.

4. Click for //RC(8050) and select the RCD(50) file. The RCD data appears in the window.

Editing PRM Data
The Motoman Parameter All.PRM editor allows you to edit parameter data and copy and paste with Microsoft Excel.

1. Double-click an item in the right window. The Edit Dialog box appears allowing you to edit this value.
2. Select Edit · Copy to copy data to the clipboard. This data can then be pasted into an Excel spreadsheet, modified, and then copied and pasted back into MotoPrmAll.

Adding a Base Axis
The MotoSimAll software makes adding x-, y-, and z-axis data for robot base very easy. The following illustrates how to add an x-axis to a UP20 robot.

1. Create a new MotoSim cell with UP20 robot.
2. Save this cell and then close it.
3. Using MotoPrmALL, open the ALL.PRM file for this UP20 robot holder.
4. Select Tool · Change XRC Basic Configuration.

5. Select “x” from the Base Axis window and click OK. The title bar changes to reflect the “7(axis)” and “r1b1s1s2s3s4s5s6” coordinate data.
   - r1 — Robot 1 group
   - b1 — Base axis group for Robot 1
   - s1 — Station axis group

6. Locate and view data at S1D 150,152, 165, and 185. This data is related to the additional x-axis data. You must determine the x-axis features; How long is the stroke?, How fast can it operate?, etc. This data can be copied from existing data.

7. From MotoPrmAll, open ALL.PRM located at “C:\Program Files\Motoman\MotoSim\Robots\Motoman XRC\Up6 Family\Up6-a00\x” Both ALL .PRM parameters are shown at once.
8. Adjust the screens to appear side by side with left view showing UP20, and right view UP6 with X-Base Axis.

9. Select RC2G from right UP6 view and copy data to clipboard (Ctrl+C).

10. Select RC2G in left UP20 view and paste data from clipboard (Ctrl+V). The Copy Prompt window appears.

11. Click OK. The Copy dialog appears confirming “copied 1 block data.”

12. Click OK.


14. Save the UP20 ALL.PRM files.

15. Open the first UP20 cell created in these steps. The UP20 appears with x-base axis, and “Up20-a00_ex1” appearing in the CAD tree view.

Adding External Station Axis

Adding an external station axis is very similar to the adding x-base axis operation.

1. In MotoPramALL, open the ALL.PRM file for the UP20 robot.

2. Open the ALL.PRM file in “C:\Program Files\MotoSim\Robots\Motoman XRC \Up6 Family\Up6-a00\s1(S200-A00,B00)” holder. This ALL.PRM file includes “s1” (External Station data) as Group 2.
3. Select Tool → Change XRC Basic Configuration. The Group Property window appears.
4. Select “s1” in the Number of Stations box.
5. Select “1” in the Number of Axis box.
6. Click OK.

7. From MotoPrmAll, select RC2G from the Up6 view and copy data to clipboard (Ctrl+C).
8. Select RC3G in left UP20 view and paste data from clipboard (Ctrl+V). The Copy Prompt window appears.
9. Click OK. The Copy dialog appears confirming “copied 1 block data.”
10. Perform the same operation for the following files:
    Copy UP6 R02G to UP20 R03G.
    Copy UP6 S1C2G to UP20 S1C3G.
    Copy UP6 SV2G to UP20 SV3G.
    Copy UP6 SVM2G to UP20 SVM3G.
    Copy UP6 AMC2G to UP20 AMC3G.
    Copy UP6 SVP2G to UP20 SVP3G.
    Copy UP6 MF2G to UP20 MF3G.
12. Open the first UP20 cell created in these steps. The UP20 appears with External Station axis, and “Up20-a00_ex2” appearing in the CAD tree view.

To add a rotating model with ex2 external axis, proceed as follows:
1. Add Ex2Base model.
2. Right click on Up20-a00_ex2 in the CAD tree and select Model Attribute → Rename....
3. Rename model Up20-a00_ex2x.
5. Create a new model Ex2Table.
6. Add cylinder model in Ex2Table model.
8. Delete Up20-a00_ex2x model.
9. Save Cell and close.
10. Open cell again. The cell shows the UP20 robot cell with ex1 (Base) and ex2 (Station).

**Viewing Joint Information**

MotoPrmALLView.exe allows you to view Robot Grouped Joint data.

1. Execute MotoPrmALLView.exe.
2. Open UP20 robot ALL.PRM or drag and drop from Windows Explorer.

![Joint 0 data in robot group](image1)

![Link 0 data in robot group](image2)
Joint 0 data in Base-X axis group

Joint Type 0 — X-direction
Joint Type 1 — Y-direction
Joint Type 2 — Z-direction
Joint Type 3 — X-rotation
Joint Type 4 — Y-rotation
Joint Type 5 — Z-rotation

Joint 0 data in Station External axis

ALL.PRM has r1 as robot, b1 as X-Base-axis and s1 as Station External axis in left tree view. Controller type and robot type in right view.
6.4.3 Using VCON/VCOF

For more information, please refer to the T-Axis Speed Control Function Manual (147429-1).

MotoSim adds a link-model (Robot_link6_link1) as the child model of the T-Axis frame (Robot_link6) for simulation.

If the Robot_link6_link1 model already exists, MotoSim will use the existing model for simulation (see Using VCON.mcl).

**VCON**

The T-axis starts rotating continuously with the execution of the VCON instruction. However, there is a delay of 200 (ms) before rotation starts. If the next operation requires the rotation of the T-axis, use the TIMER instruction to wait for the T-axis to begin rotating at a constant speed before executing the next step. The simulator doesn’t calculate acceleration and deceleration velocity. The simulator will create a link-model (Robot_link6_link1) if required.

Showing a rotating-model can be done to add a visible model as the child model of the link-model (Robot_link6_link1).

**VCOF**

The rotation will stop without delay. This instruction will hold the robot for 500 (ms). During this delay time, the simulator will rotate back the link-model (Robot_link6_link1) to Rz=0.
### 6.5 Utilities

#### 6.5.1 Robot Autoplacement

Robot autoplacement (Auto Place) is a method for determining the optimum location for the robot in a given workspace. “Optimum” is defined as robot placement that minimizes job cycle time, moves the robot within reach of points in the job, or both.

The first and most logical requirement is that the robot can reach all points in the programmed job. Once all possible robot locations have been determined, you can then select the one location that minimizes cycle time.

To start Auto Place, proceed as follows:

1. Create or open a job to which you want to Auto Place the robot.
2. If your cell has more than one robot, make sure the correct robot is selected using the Change Robot… command.
3. Click the Auto Place button on the toolbar or select Auto Place… from the Tools menu. The Auto Place dialog will appear.
4. Click Job… to convert a MotoSim job (INF file) to an Auto Place file. When the Making Job dialog appears, either create a new job using the active teacher model and the specified move settings (left side) or convert an existing job (right side).
5. Click Next>.
6. The second screen of the Making Job… dialog will appear. The contents of the INF file is displayed in the left window. You may view the INF using the scroll bars, mouse wheel, or up and down keys; or edit the INF by clicking Editor… below the job display window.
7. Click Convert>> to convert the INF to a JBI.

**NOTE:** If a JBI with the same name already exists, MotoSim will prompt you to overwrite it or cancel.

8. Click Finish.
9. Make sure OLP Pick is turned on by clicking the button with the hand icon on the toolbar.
10. Click the small window in the top of the Select Teach box. A flashing cursor will appear. If the window already contains text, select it all and delete it.
11. Click the desired Axis6 (Teach) point. The Axis6 model’s CAD Tree name will appear in the Select Teach box.
12. Click Search. Search locations are generated automatically within the specified search area.
13. Click Quick Cycle Time Start to start the Auto Place search.
14. Scroll through the list to determine which robot position yields the best cycle time. Cycle time is displayed at the beginning of each line.

The Auto Place model (containing Axis6 models for each location) is visible unless it is hidden or deleted from the CAD Tree window. The Auto Place model is named [robot name/xAutoplace].
INF2JBI ("INF to JBI")
INF2JBI is used to create jobs relative to work pieces rather than the robot cell. This operation is necessary when fixturing or the robot is moved in the cell. INF2JBI modifies the job file from existing data so you don’t have to rewrite the entire job if you decide to move a work piece or robot.
INF2JBI is a program that uses scripts called INF files to modify .jbi (job) files based on commands and data entered into the INF file.

List of INF2JBI Commands
The following 14 commands are used in INF2JBI:

- **TOOL**
  Specifies tool number.

- **JOBTYPE**
  Specifies job type.

- **POSCONE**
  Specifies position type.

- **GROUP**
  Specifies move control group.

- **POSC**
  Defines robot position.

- **POSE**
  Defines external axis position.

- **PUTP**
  Makes the values set with the POSC, POSE default positions that are used with TBOX-TEACH command.

- **TBOXMOV**
  Specifies parameter after each move command.

- **TBOXTEACH**
  Adds an actual POS MOVE or POS line to the job file.

- **INFORM**
  Adds a supplementary command to the job file.

- **ERRORLIST**
  Creates a list of conversion errors (if any occur).

- **MODELDRV**
  Works the same as “Part Motion Simulation” while creating a job.

- **GENERALOPR**
  Enables or disables robot servo effect.
### Table 6-12 INF2JBI Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
<th>Input (INF)</th>
<th>Result</th>
<th>Output (JBI)</th>
</tr>
</thead>
<tbody>
<tr>
<td>TOOL</td>
<td>Specifies the tool number. After specifying the tool number, it will remain that way until changed again.</td>
<td>TOOL=(0)</td>
<td>The tool number is specified as “0.”</td>
<td>/// TOOL 0</td>
</tr>
<tr>
<td>JOB TYPE</td>
<td>Specifies the job type. After specifying the job type, it will remain that way until changed again. PULSE—Pulse type (default) RECTAN—Rectangular coordinate</td>
<td>JOBTYPE (&quot;PULSE&quot;);</td>
<td>The job type is specified as PULSE.</td>
<td>/// PULSE</td>
</tr>
<tr>
<td>POSCONF</td>
<td>Specifies position type. This needs to be specified when a new job file is created. After specifying the position type, it will remain that way until changed again. If unspecified, the default value (PULSE) is used. PULSE: default BASE: Base coordinate ROBOT: Robot coordinate TOOL: Tool coordinate USER: User coordinate</td>
<td>POSCONF=&quot;PULSE&quot;</td>
<td>The position type is specified as PULSE.</td>
<td>/// POSTYPE PULSE</td>
</tr>
<tr>
<td>GROUP</td>
<td>Specifies the MOVE control group. This needs to be specified when a new job file is created. If unspecified, the default 1, “RB1” is used. First parameter: Group Range (1~8) Second parameter: Group Specification (RB1, RB2, BS1, BS2, ST1, ST2, ST3, ST4, ST5, ST6)</td>
<td>GROUP=(1, “RB1”)</td>
<td>Control group 1, RB1 is specified.</td>
<td>/// GROUP1 RB1</td>
</tr>
<tr>
<td>Command</td>
<td>Purpose</td>
<td>Input (INF)</td>
<td>Result</td>
<td>Output (JBI)</td>
</tr>
<tr>
<td>---------</td>
<td>---------</td>
<td>-------------</td>
<td>--------</td>
<td>--------------</td>
</tr>
<tr>
<td><strong>POSC</strong></td>
<td>Defines a robot position (PULSE, RECTAN).</td>
<td>POSC[999]=(-308, 7097, -19163, -30, 3809, -89); The values in the above example are set based on Axis6 SLURBT position.</td>
<td>Contains position information for recall by PUTP or TBOXTEACH.</td>
<td>Stores a robot position. These positions may be made current via the PUTP command or used directly in the TBOXTEACH command.</td>
</tr>
<tr>
<td><strong>POSE</strong></td>
<td>Defines an external axis position (PULSE).</td>
<td>POSE[999]=(-33520, 1000, 24936);</td>
<td>Contains position information for recall by PUTP or TBOXTEACH.</td>
<td>Stores a robot position. These positions may be made current via the PUTP command or used directly in the TBOXTEACH command.</td>
</tr>
<tr>
<td><strong>PUTP</strong></td>
<td>Makes POSC and POSE positions the default positions for calls with a NULL parameter in the TBOXTEACH command.</td>
<td>PUTP(POSC[999]);</td>
<td>Stores POSC internally. The robot will assume this pose.</td>
<td>Next time a robot position is specified via the TBOXTEACH command with a “NULL” parameter, the last POSC will be used to determine the robot’s position.</td>
</tr>
<tr>
<td><strong>TBOXMOV</strong></td>
<td>Sets a parameter after a MOVE command.</td>
<td>TBOXMOV=(MOVJ, VJ=50.00, PL=1);</td>
<td>Sets the move to the type and velocity listed.</td>
<td>MOVJ VJ=50.00</td>
</tr>
<tr>
<td><strong>TBOXTEACH</strong></td>
<td>Adds a line to the job file. (Move type, Robot position, External axis position, Station axis position) Values: NULL-Uses last POSC set in PUTP •POSC[999] •Model: Axis6 [0][0]</td>
<td>TBOXTEACH=(MOVJ, NULL, NULL, NULL);</td>
<td>A MOVE command is added to the job. Parameters and positions are taken from TBOCMOV, POSC, POSE, and PUTP.</td>
<td>MOVJ C000 VJ=50.00</td>
</tr>
<tr>
<td><strong>INFORM</strong></td>
<td>Adds supplementary commands to the job file.</td>
<td>INFORM (“TIMER T=1.0”)</td>
<td>Timer command added to file.</td>
<td>TIMER T=1.0</td>
</tr>
<tr>
<td><strong>ERRORLIST</strong></td>
<td>If there is a conversion error, a description of the error will be placed in the *.log file.</td>
<td>*.log</td>
<td></td>
<td>NA</td>
</tr>
</tbody>
</table>
### Table 6-12 INF2JBI Commands - continued

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
<th>Input (INF)</th>
<th>Result</th>
<th>Output (JBI)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>GENERALOPR</strong></td>
<td>Enables and disables robot servo effect.</td>
<td>GENERALOPR(&quot;Servo Off&quot;); Or GENERALOPR(&quot;Servo On&quot;)</td>
<td>Disables and enables servo effect of robot.</td>
<td>GENERALOPR(&quot;Servo Off&quot;); is usually inserted in the INF file. Output (JBI) none</td>
</tr>
<tr>
<td><strong>MODELDRV</strong></td>
<td>Works the same as “Part Motion Simulation” while creating a job.</td>
<td>MODELDRV(&quot;SET 111MasterBox SP100_UF3&quot;);</td>
<td>Part Motion added to file.</td>
<td>Output (JBI) ‘MDL: SET 111MasterBox SP100_UF3’</td>
</tr>
<tr>
<td><strong>TCPSYMME-TRY</strong></td>
<td>Allows Symmetry teaching for robot TCP.</td>
<td>COMMAND(“TcpSymmetry=On”); COMMAND(“TcpSymmetry=Off”)</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>TCPEIMOV</strong></td>
<td>Defines User Frame number for EIMOV operation (see “CuttingCurveEIMOV.mcl in example folder).</td>
<td>COMMAND(“TcpEIMOV=UF#(1)” )</td>
<td>User Frame #1 is used for EIMOV command.</td>
<td></td>
</tr>
</tbody>
</table>

---

**Note:**
- General operations enable and disable robot servo effects.
- Model drive works the same as “Part Motion Simulation” during job creation.
- TCP symmetry allows teaching for robot TCP.
- TCPEIMOV defines user frame numbers for EIMOV operations.
Using INF2JBI requires the following steps:

1. Create a cell including part(s) and a robot.
2. Create Axis6 points or note other references such as vertices, corners, etc.
3. Create an INF file using Notepad. Be sure to save the file with an “INF” extension or INF2JBI won’t recognize it. Refer to the INF example below when creating your INF file.
4. Select Robot · Auto Place… from the Tools menu.
5. Select Making JOB from the Teach menu in the Auto Place dialog.

Sample INF file using above-listed commands:

```plaintext
ROBOT(0);
{
  CELNAME=DEMO.CEL;
  ERRORLIST=./DEMO.LOG;
}
//
SUBJOB("H");
{
  TOOL=(1);
  JOBTYPE("PULSE");
  GROUP = (1,"RB1");
  //
  POSC[999]=(0,0,0,0,-14000,0);
  //define a position
  PUTP(POSC[999]);
  GENERALOPR ("Servo Off");
  //make a position active
  POSE[999]=(0,....,);
  PUTE(POSE[999]);
  TBOXMOV=(MOVJ,VJ=100,PL=1);
  //set default velocity for joint moves to 100%
  TBOXMOV=(MOVL,V=400,PL=1);
  //set default velocity for linear moves to 400mm/sec.
  POSCONF=(PULSE);
  TBOXTEACH=(MOVL,ATC1:AXIS6[0][0],NULL,NULL);
  //program a linear move at the Axis6 location (0,0) on object ACT1
  TBOXTEACH=(MOVL,ATC1:AXIS6[0][1],NULL,NULL);
  TBOXTEACH=(MOVL,ATC1:AXIS6[0][2],NULL,NULL);
  TBOXTEACH=(MOVL,ATC2:AXIS6[0][0],NULL,NULL);
  TBOXTEACH=(MOVL,ATC2:AXIS6[0][1],NULL,NULL);
  TBOXTEACH=(MOVL,ATC2:AXIS6[0][2],NULL,NULL);
  TBOXTEACH=(MOVL,ATC3:AXIS6[0][0],NULL,NULL);
  TBOXTEACH=(MOVL,ATC3:AXIS6[0][1],NULL,NULL);
  TBOXTEACH=(MOVL,ATC3:AXIS6[0][2],NULL,NULL);
  TBOXTEACH=(MOVL,ATC4:AXIS6[0][0],NULL,NULL);
  TBOXTEACH=(MOVL,ATC4:AXIS6[0][1],NULL,NULL);
```

MOTOMAN

6-55

MotoSim User’s Manual
TBOXTEACH=(MOVL,ATC4:AXIS6[0][2],NULL,NULL);
POSC[999]=(0,0,0,0,-14000,0);
PUTP(POSC[999]);
TBOXMOV=(MOVJ,VJ=100,CONT);
TBOXTEACH=(MOVJ,NULL,NULL,NULL);
}

Sample JBI file created by processing the above INF file with INF2JBI:

/JOB
//NAME H
//POS
///NPOS 17,17,0,0
///TOOL 1
///PULSE
C000=0,0,0,0,-14000,0
C001=-41035,13545,-12170,0,-7544,1651
C002=-40833,14254,-11974,0,-7653,1602
C003=-41035,13545,-12170,0,-7544,1651
C004=-41035,13545,-12170,0,-7544,1651
C005=-45715,15986,-11400,0,-7970,2781
C006=-45414,16630,-11155,0,-8106,2708
C007=-45715,15986,-11400,0,-7970,2781
C008=-45715,15986,-11400,0,-7970,2781
C009=-49464,19508,-9867,0,-8819,3687
C010=-49123,20125,-9555,0,-8992,3604
C011=-49464,19508,-9867,0,-8819,3687
C012=-49464,19508,-9867,0,-8819,3687
C013=-54089,17455,-10816,0,-8294,-3456
C014=-53638,17924,-10612,0,-8406,-3565
C015=-54089,17455,-10816,0,-8294,-3456
C016=0,0,0,0,-14000,0
E000=0,....
E001=0,....
E002=0,....
E003=0,....
E004=0,....
E005=0,....
E006=0,....
E007=0,....
E008=0,....
E009=0,....
E010=0,....
E011=0,....
E012=0,....
E013=0,....
E014=0,....
E015=0,....
E016=0,....
//INST
//DATE 1995/2/27 16:17
///ATTR 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
///FRAME BASE
NOP

MOVEJ C000 E000 VJ=100.0 PL=1
MOVEJ C001 E001 VJ=100.0 PL=1
MOVEJ C002 E002 V=400.0 PL=1
MOVEJ C003 E003 V=400.0 PL=1
MOVEJ C004 E004 VJ=100.0 PL=1
MOVEJ C005 E005 VJ=100.0 PL=1
MOVEJ C006 E006 V=400.0 PL=1
MOVEJ C007 E007 V=400.0 PL=1
MOVEJ C008 E008 VJ=100.0 PL=1
MOVEJ C009 E009 VJ=100.0 PL=1
MOVEJ C010 E010 V=400.0 PL=1
MOVEJ C011 E011 V=400.0 PL=1
MOVEJ C012 E012 VJ=100.0 PL=1
MOVEJ C013 E013 VJ=100.0 PL=1
MOVEJ C014 E014 V=400.0 PL=1
MOVEJ C015 E015 V=400.0 PL=1
MOVEJ C016 E016 VJ=100.0

RET

END

6.5.2 Collision Detection

Collision detection allows you to determine if an object will collide with one or more objects in the cell. When collision detection is active, any object that is impacted will turn red. This signals you that a collision has occurred. Correct the robot position or path to prevent the collision.

Collision Detection also has the ability to draw models to represent the three-dimensional space and/or parts involved in the collision. To turn this option on, select Show Intersection Lines from the Collision Pair menu.

1. Select New Collision Pair.
2. Select Collision01@1.
3. Select Collision Pair · Add Model...
4. Select a model for collision check.

5. Do the same operations for Collision01@2.
6. You can also use the CAD tree to set a collision model.
7. Click \textit{CTL}+\textit{C} at the CAD tree.

8. Click \textit{CTL}+\textit{V} to add a model.

Select \textit{Create Line Models} from the Collision Pair menu.
Several Intersection-Line options are available.

\textbf{Frame (Axis 6)}
Allows you to create an AXIS model not a LINE model.

\textbf{Delete Same Point}
Deletes the same point, if it exists.

\textbf{Make Normal Frame}
Allows you to create a normal frame at the individual face, if it exists.

\textbf{Set X-axis to Nextpoint}
Allows you to force the X-direction to the next point.

\textbf{Min Dist.(mm)}
Allows you to define the minimum distance between each point. If the distance between two points is less than this value, MotoSim treats these points as the same point.

You can also select which model you want to relate the new model. The parent of the new model will be the same as the parent of the selected model.
6.5.3 Setting Up Inputs/Outputs for Multiple Robots

Use the following procedure to set up inputs and outputs for multiple robots.

**NOTE:** 
I/O is set up for general application and cannot be changed to material handling, welding, etc.

1. Open a MotoSim .mcl file containing two or more robots.
2. From the Combo box on the Robot Toolbar, select a robot.
3. Select the I/O Monitor icon \( \text{\text{I/O Monitor}} \) from the toolbar.

4. Repeat steps 2 and 3 for the second robot.
5. The Model I/O Input window is displayed after each robot is selected.
6. From the pull down menu for robot 1, select ROUT01 and the output group desired (Model I/O Output window).
7. Highlight the output to be connected.
8. Click **Add To**. The Model I/O Input window is displayed, with three pull down menus and a Set Count window.
9. In the top pull down menu, select the robot that the first robot’s output will be connected to.
10. The second pull down menu defaults to RIN01. You cannot change this default.
11. In the third pull down menu, select the desired input group.
12. Highlight the Input that the first robot’s output is to be connected to by selecting it as shown.
13. Click **OK**.
14. The Output is now connected to the second robot’s Input. The connection is displayed in the “Connect To” and “Connect From” sections of each robot’s I/O box.
**Verifying I/O Connections**

1. Display the Outputs of the first robot and the Inputs of the second.

![Model I/O (Contnr_RIO) Output](image)

![Model I/O (Contnr_RIO) Input](image)

2. To verify that Output #1 on the first robot is connected to Input #1 of the second robot, click the box next to Output#1. If the connection is correct, the box next to Input#1 of the second robot will display a check mark.

**NOTE:** Selecting “Clr” will turn off all outputs of the robot whose I/O box is active.

**Setting I/O Group**

The *Set Count* edit box can be set to the number of I/O connections you need in the Connection dialog. For example; if you set this number to 8, 8 series of I/O connections are set in one action. This is useful for setting group I/O combining 8 bits as 1 group.

**Setting I/O Names**

I/O names can be set to each terminal. Double click any terminal listed and an I/O Name window appears. Simply type in the desired name and click *OK*.

This function works with the IONAME.DAT file, if the robot folder contains an IONAME.DAT file.

The reading sequence is as follows:

1. Load IO name information from MDL file.
2. Load IONAME.DAT information if IONOAME.DAT exists.

The writing sequence is as follows:

1. Write IONAME.DAT information if IONAME.DAT file exists.
2. Write IO name information to MDL file.

The following is an example of an MDL file that includes IO names.

//Up6_RIO
//Model Memory DATA
MEMORYBLOCK
{
    MEMINPUTLONG(128,1)
    {
        NAME=RIN01
        DATA=0 0 0 0
        IONAME(0)=Cycle Start
        IONAME(1)=ManualXZ
        IONAME(16)=At Side A
        IONAME(17)=At Side B
        IONAME(18)=Decel Side A
IONAME(19)=Decel Side B
IONAME(20)=Zero Speed
IONAME(21)=Inverter Fault
IONAME(62)=Conveyor Rdy Ld
IONAME(64)=Job Bit 0 Side A
IONAME(65)=Job Bit 1 Side A
IONAME(66)=Job Bit 2 Side A
IONAME(67)=Job Bit 3 Side A
IONAME(68)=Job Bit 4 Side A
IONAME(69)=Job Bit 5 Side A
IONAME(70)=Job Bit 6 Side A
IONAME(71)=Job Bit 7 Side A
IONAME(72)=Job Bit 0 Side B
IONAME(73)=Job Bit 1 Side B
IONAME(74)=Job Bit 2 Side B
IONAME(75)=Job Bit 3 Side B
IONAME(76)=Job Bit 4 Side B
IONAME(77)=Job Bit 5 Side B
IONAME(78)=Job Bit 6 Side B
IONAME(79)=Job Bit 7 Side B
IONAME(80)=Riveter Ready
IONAME(81)=In Cycle
IONAME(82)=Riveter Finish
IONAME(83)=Home
IONAME(84)=Partial Retracte
IONAME(85)=Fault 1
IONAME(86)=Fault 2
IONAME(87)=Fault 4
IONAME(88)=Rivet Prsnt T1
IONAME(89)=Rivet Prsnt T2
IONAME(90)=Vac/Prox Made
IONAME(91)=Henrob Reset
IONAME(92)=Reset Inverter
}
MEMOUTPUTLONG(128,1)
{
NAME=ROUT01
DATA=255 0 0 0
IONAME(0)=Table Relay
IONAME(1)=Vacuum Check
IONAME(8)=Partial Retract
IONAME(9)=Cut Tape T1
IONAME(10)=Motor Start
IONAME(11)=Cut Tape T2
IONAME(12)=Motor Stop
IONAME(16)=FWD run/stop
IONAME(17)=REV run/stop
IONAME(18)=Impact stop
IONAME(19)=Fault reset
IONAME(20)=pin 19 G5 pc bd
IONAME(21)=Speed 1
IONAME(22)=Jog FWD A
IONAME(23)=Jog REV B
IONAME(24)=Unclamp A
IONAME(48)=Robot Ready
IONAME(60)=Run Time Clock
IONAME(61)=Count A Side
IONAME(62)=Count B Side
IONAME(64)=Rbt Load Complet
IONAME(80)=Riveter Advanced
IONAME(81)=Riveter Home
IONAME(82)=Seq Sel 1
IONAME(83)=Seq Sel 2
IONAME(84)=Seq Sel 4
IONAME(85)=Seq Sel 8
IONAME(86)=Gun Sel 1
IONAME(87)=Gun Sel 2
IONAME(92)=Inverter Fault
The following is an example of a IONAME.DAT file downloaded from an XRC.

/IONAME
//IN
Cycle Start,ManualXZ,,
...
...
...
At Side A,At Side B,Decel Side A,Decel Side B
Zero Speed,Inverter Fault,,
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
Robot Ready,,
...
...
Run Time Clock, Count A Side, Count B Side,
Rbt Load Complet,,
...
...

Riveter Advanced, Riveter Home, Seq Sel 1, Seq Sel 2
Seq Sel 4, Seq Sel 8, Gun Sel 1, Gun Sel 2
...
Inverter Fault,,
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...

6.5.4 Monitoring Robot Variable Data

MotoSim can monitor Robot variables including B-variables (Byte), I-variables (Integer), L-variables (Long), and R-variables (Real).

1. Select Controller · Information… from the Robot menu.
2. Check "MotoDDE" and click OK.

3. A MotoDDE software screen will appear.
4. When you play a job, variable data will be displayed in the window.

5. Double-click a variable to edit its data. Make desired change and click OK. Changed variables will appear red.

You can also use Microsoft Excel DDE function. If you are want to view the Int000 variable, you can link this value to Excel by inserting the following command in Excel: (=MOTODDE]MotDDE1]gmemory!Int000)
6.5.5 Changing Motion Module

The internal motion module speeds simulation time but lessens the accuracy of cycle time analysis somewhat. MotoSim is designed to use the RRS (Realistic Robot Simulation) module for simulating the XRC controller. Using the RRS module slows down simulation time but yields very accurate cycle time calculation results (less than 1% error).

1. Select Controller · Information… from the Robot menu.
2. Click Select…
3. Select the desired Motion module and click OK.

6.5.6 Axis6 Mathematics Programming

Here is a C/C++ example program of converting roll-pitch-yaw angle and Euler angle and Matrix.

**AXIS6** means Roll-Pitch-Yaw angle.

**AXIS6T** means Euler angle.

**FRAME4** means 4*4 Matrix.

**Converting Matrix to AXIS6 Data**

```c
AXIS6 *FRAME42AXIS6(FRAME4 *f,AXIS6 *p) {
  int i;
  double c,s;
  p->p[5]=atan22(f->N[1],f->N[0]);
  c = cos(p->p[5]);
  s = sin(p->p[5]);
  p->p[3] = atan22(s*f->A[0]-c*f->A[1],-s*f->O[0]+c*f->O[1]);
  for(i=0;i<3;i++) p->p[i]=f->P[i];
  return(p);
}
```

**Converting Axis6 Data to Matrix**

```c
FRAME4 *AXIS62FRAME4(AXIS6 *p,FRAME4 *f) {
  int i;
  double s[3],c[3];
  for(i=0;i<3;i++)
  {
    s[i]=sin(p->p[i+3]);
    c[i]=cos(p->p[i+3]);
    f->P[i]=p->p[i];
  }
```
ADVANCED OPERATIONS

f->N[0] = c[2]*c[1];
f->N[1] = s[2]*c[1];
f->N[2] = -s[1];
f->O[0] = c[2]*s[1]*s[0] - s[2]*c[0];
f->O[1] = s[2]*s[1]*s[0] + c[2]*c[0];
f->O[2] = c[1]*s[0];
f->A[0] = c[2]*s[1]*c[0] + s[2]*s[0];
f->A[1] = s[2]*s[1]*c[0] - c[2]*s[0];
f->A[2] = c[1]*c[0];
return(f);
}

Converting Euler data to Matrix

FRAME4 *AXIST62FRAME4(AXIS6 *p, FRAME4 *f)
{
int i;
FRAME4 f1,f2,f3;
axisRframe4(3,p->p[3],&f1);
axisRframe4(4,p->p[4],&f2);
mulframe4(&f1,&f2,&f3);
axisRframe4(5,p->p[5],&f2);
mulframe4(&f3,&f2,&f);
for(i=0;i<3;i++)f->P[i]=p->p[i];
return(f);
}

FRAME4 *axisRframe4(int no,double d, FRAME4 *f)
{
AXIS6 p1;
p1=normaxis6();
p1.p[no]=d;
AXIS62FRAME4(&p1,f);
return(f);
}

FRAME4 *mulframe4(FRAME4 *a, FRAME4 *b, FRAME4 *c)
{
int i;
for(i=0;i<3;i++)
{
c->O[i] = a->N[i]*b->O[0] + a->O[i]*b->O[1] + a->A[i]*b->O[2];
c->P[i] = a->N[i]*b->P[0] + a->O[i]*b->P[1] + a->A[i]*b->P[2] + a->P[i];
}
return(c);
}

Converting Matrix to Euler data

AXIS6 *FRAME42AXIST6(FRAME4 *f, AXIS6 *p)
{
/* Tool */ /* Not Implemented */
int i;
double s[3],c[3];
for(i=0;i<3;i++)
{
    s[i]=sin(p->p[i+3]);
    c[i]=cos(p->p[i+3]);
}

f->N[0] = c[2]*c[1]*c[0]-s[2]*s[0];
return(p);
}

For complete understanding data structures

typedef struct frame4st FRAME4;
struct frame4st
{
    double N[3],O[3],A[3],P[3];
};

typedef struct axis6_st AXIS6;
struct axis6_st
{
    double p[6];
};

6.5.7 Converting a Robot Model from ROTSY

To convert a robot model from ROTSY to MotoSim, proceed as follows:
1. Create a temporary folder and copy the .mdl, .dat, and .prm files to this folder.

2. Start the “MSimConvertRobotModelsFromYec.exe” program in MotoSim. The MSimConvertRobotModels window appears.
3. Drag the “all.prm” file from the temporary folder and drop into this window.
4. Click Convert All. The following confirmation windows appear.

5. Click OK for each.

MotoSim creates a new folder with the same name as the .mdl file. You may want to rename the robot file and folder names after this process.

ModelMacro
You may see the following “MODELMACRO” script in the “Robotinf.dat” file.

MODELMACRO
{

ADD %RBNAME%_link3 wire "%RBPATH%wire.MDL"
ADD %RBNAME%_flange Torch "%RBPATH%torch.MDL"
}

This can be done in MotoSim cell after creating or adding robot.
See "Ea1400-a00xStdCell".

In a few case, there are *.Rwx files attached to the Robot link. All *.Rwx files are
included in the robot model as reference files.

Be sure to copy all *.Rwx files to the robot folder.

6.5.8 Using Model I/O Viewer

1. Select Model IO View... from the Tools menu.

The Model IO View window appears.

2. The Cut, Copy, and Paste commands from the Edit dropdown window can be
used for all connections.

These commands can also be used to copy and paste connections to and from an
Excel spreadsheet for editing.
6.6 Calibration

6.6.1 Calibration

Calibration is necessary to ensure that jobs created in MotoSim accurately represent real-world conditions in your actual robot cell. Refer to the MotoSoft™ MotoCal manual for detailed instruction on robot job calibration.

6.6.2 Showing CompuGuage Data

1. Open cell.
2. Make a model and add a robot.
3. Right click within the world screen and select New Model.
4. Name this model AAA and click OK.
5. From the CAD tree, double click the AAA folder.
6. Select LINE from Add Parts menu and click Add.
7. Click Insert button and to insert 1 point in this Line.
8. Click OK.
9. Select LINE and click Clip Board....

10. Open Microsoft Excel®.
11. Open *.txt file created by Compu-Guage.
12. Select the *Fixed width* radio button and click *Next*.

13. Select the area you want to show, and Copy to clipboard (Ctrl-C).
14. Click *Paste* from MotoSim window.

15. Click *OK*. The new line appears, created from the Compu-Guage .txt file.
APPENDIX A

A.1 Programming Examples

Programming examples are included with MotoSim to help you create your own programs and cell files. The files are located in the “Examples” subdirectory inside your MotoSim directory.

<table>
<thead>
<tr>
<th>Example Name</th>
<th>Programming Samples of…</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spot-weld</td>
<td>• Linear DLL</td>
</tr>
<tr>
<td></td>
<td>• Hide</td>
</tr>
<tr>
<td></td>
<td>• Multiple work pieces</td>
</tr>
<tr>
<td></td>
<td>• Long program</td>
</tr>
<tr>
<td>Tank-liner</td>
<td>• Trace</td>
</tr>
<tr>
<td>RM2x500 Example</td>
<td>• RM2 DLL</td>
</tr>
<tr>
<td>Press example</td>
<td>• I/O</td>
</tr>
<tr>
<td>Using Index32</td>
<td>• Indexed Motion</td>
</tr>
<tr>
<td>Using Intersection</td>
<td>• Cylinder</td>
</tr>
<tr>
<td>Using Model Drive</td>
<td>• Axis6 Model</td>
</tr>
<tr>
<td></td>
<td>• Hide/See Model</td>
</tr>
<tr>
<td>Using Inform JOB</td>
<td>• Inf2JBI</td>
</tr>
<tr>
<td></td>
<td>• EIMOVL</td>
</tr>
<tr>
<td>Compu Guage</td>
<td>• Showing CompuGuage data</td>
</tr>
</tbody>
</table>
A.2 OLE Sample Code

Open architecture is very important for software products being released today. MotoSim can be used by other Motoman software, off-the-shelf software, and customized user applications to perform and simplify a variety of complicated robot tasks.

OLE presents multiple advantages for the MotoSim user.

- OLE enables users to create macros to simplify complex or repetitive tasks.
- OLE allows MotoSim to share data with other OLE compliant applications, opening a world of possibilities for automating robot job generation and sharing robot information with other robotics support software.
- MotoSim’s OLE functionality permits control of MotoSim functions via remotely networked workstations.

OLE Client source code is included with MotoSim and you may use it to build your own custom applications to interface with MotoSim. This C++ code is subject to change without notice. If you are a software developer and you want to use the supplied code, always use the file MotoSim.tlb located in the same directory as the MotoSim executable.

- Pick and Place
- Tool Effect
- DXF2JBI
- Bending Sheet Metal
- Multi Cylinder
- MotoSim Collision
A.3 Motoman Part Files

Motoman Parts files are included with MotoSim to help you create your own cell files. These files are located in the "\Program Files\Motoman\MotoSim\Parts\MISTD Components" subdirectory inside your MotoSim directory.

The following is an example procedure for importing a UP6 6 inch riser.

1. Select Import 3D Model… from the Model dropdown menu. The Model file appears..

2. Select the 6in riser.3ds file and click Open.

You must now set the parent of the UP6 robot to the riser. Proceed as follows:

1. Right click on the Up6-a00 file in the CAD tree and select Model Attribute > Set Parent....

2. Right click on the 6in_riser file in the CAD tree and select Edit Model....
3. Change AXIS6 data, Z-Axis (height) to 152.4mm (6 in).

**Note:** Refer to Standard Components.xls in \Program Files\Motoman\MotoSim\Parts\ MI STD Components for height. Some Motoman standard risers are not exact millimeter equivalents to inch (Ex: the UP6, 6 inch riser is 155 millimeters tall).


5. Make sure both the riser and robot appear in the list.

6. Click *OK*. The robot now appears in the correct location atop the riser.
# INDEX

**Numerics**

3-D CAD Model, 4-16

**A**

Active, 4-31
Add, 4-17, 4-29, 4-33, 4-34
Add Parts, 4-17
Add Robot..., 4-4
Adding Robots, 5-3
Adding/Deleting Robots, 5-11
All Pose, 4-15
ANSI/RIA, 2-1
Arrange Icons, 4-9
Attach, 4-31
Attribute Menu, 4-7
Auto, 4-25
Auto Color, 4-29
Auto Place, 4-32
Auto Place..., 4-8
AutoCAD Support, 4-8
Automatic Job Creation, 1-1
Automatically Creating Axis6 Job Models, 5-10
Axis Controls, 4-22
Axis6, 1-3
Axis6 Edit, 4-20
Axis6 Models, 5-10
AXIS6NO, 6-29

**B**

Back Color, 4-5
Base Frame, 4-23
Basic Model, 4-16
Basic Modeler Editing Dialogs, 4-18
Basic Motion Commands, 6-22
Box Edit, 4-18
Brake Release, 4-8
Break, 4-3

**C**

CAD, 4-8, 4-13
CAD Too, 4-8
CAD Tree/Model Selection Window, 4-16
Calculating Cycle Time, 5-13
Calibration, 6-63, 6-70
Camera, 4-5, 4-30
Camera Attatch, 4-31
Camera Bar, 4-9, 4-12
Camera Scope, 4-4
Care of Components, 1-3
Cascade, 4-9
Center, 4-4, 4-5, 4-6, 4-12, 4-20
Change JOB Holder, 4-25
Change Robot, 4-2
Changing Axis6 Length, 5-12
Changing Motion Module, 6-65
Check, 4-34
Check Reach, 4-33
Clear, 4-4
Clipboard, 4-17
Clipboard..., 4-4
Close, 4-1, 4-14
Clr, 4-27
Collision Detect, 4-13
Collision Detection, 1-1, 6-57
Color, 4-3, 4-19, 4-20, 4-29
Color..., 4-18, 4-19, 4-20, 4-21, 4-27
Cone Edit, 4-18
Configuration, 4-5, 4-13
Configuration..., 4-9
Connect, 4-31
Control Dialogs, 4-14
Controller, 4-2
Converting Intersection Frame to Job, 6-2
Copy, 4-24, 4-25
Copy Job..., 4-24
Copy..., 4-27
Copying a Job from an SK120 to a UP130, 6-11
Copying Parts, 5-6
Create, 4-3
Creating a Catalog List, 5-14
Creating a Teaching Sheet, 5-11
Creating a Trace, 5-12
Creating Model Files from 3-D CAD Files, 6-14
Cube Edit, 4-19
Customer Service Information, 1-4
Cut, Copy, Paste, and Delete, 4-1
Cycle, 4-30
Cycle Start, 4-9
INDEX

C - continued
Cycle Time, 4-26
Cycle Time Calculation, 1-1
Cycle Time..., 4-2
Cylinder Edit, 4-19

D
Data, 4-31
Data window, 4-25
Daxbis6, 4-16
Default Buttons, 4-31
Default Mode, 4-26
Define INF File..., 4-34
Del, 4-34
Del, Add, Mod, 4-23
Delete, 4-3, 4-17, 4-19, 4-20, 4-24, 4-29
Delete a Location, 4-34
Delete a Point, 4-33
Delete All Locations, 4-34
Delete All Points, 4-33
Delete Robot..., 4-4
Delete..., 4-28
Detail Mode, 4-26
Direction, 4-5, 4-12
Display Reference Frame, 4-23
Displaying Maximum Reach, 5-13
Displaying Teaching Sheet, 5-12
Displaying/Hiding Axis6 models, 5-12
Displaying/Using Pulse Record Data, 5-13
Distance, 4-27, 4-30
Division Number, 4-32
Drag, 4-12
Drag in Z, 4-6
Dual Face, 4-28
Dummy Model, 4-16, 4-17

E
Edit, 4-17
Edit Cel..., 4-1
Edit File, 4-33
Edit Menu, 4-1
Edit Model..., 4-1, 4-7
Emergency Stop (E-STOP), 4-9
Enter, 4-23

Equipment Description, 4-1, 5-1
ERRORLIST, 6-51, 6-53
Exit, 4-1, 4-33
Ext, 4-14
Ext El Move, 4-15
Extern, 4-22
External Axis, 4-23, 6-42
External CAD, 4-17
External Teach Capability, 1-1

F
Fast, Med, Slow, 4-14
Features, 1-1
File Menu, 4-1, 4-23, 4-33
File Name, 4-16
Floor Edit, 4-20
Flying Camera, 4-5, 4-13
Follow, 4-31
Frame, 4-22
Frame Length..., 4-6
Free, 4-6
Free Zoom, 4-4, 4-12
Freeze Robot Position, 4-23
Front, 4-5
Front View, 4-13
Full-control Cameras, 1-1
FUNCA1, 6-29
FUNCB, 6-29
FUNCTA2, 6-29

G
General Safeguarding Tips, 2-3
GENERALOPR, 6-51, 6-54
Goto, 4-22
GROUP, 6-51, 6-52

H
Hard Stop/Shock Absorbers, 4-16
Heart Beat, 1-3, 4-2, 4-26
Hide/See, 4-7
Hold, 4-2
Horizontal, 4-28
INDEX

I
I/O Index, 4-27
I/O Monitor, 4-27
I/O Monitor..., 4-3
Import Capabilities, 1-1
Importing Models with TriSpectives, 6-15
INDEX, 4-19, 4-20
INF2JBI (“INF to JBI”), 6-51
INFORM, 6-51, 6-53
Information, 4-2
Initial Scale, 4-13
Insert, 4-19, 4-20
Installation, 3-1
Installation Safety, 2-4
Installing and Viewing Manuals, 3-3
Installing MOTOSIM, 3-1
Installing the Hardware Key, 3-1
Installing TriSpectives™, 3-3
Introduction, 1-1
Isometric View, 4-13

J
Job Display Window, 4-25
Job Listing Window, 4-24
Job Playback and Settings, 5-10
Job Select, 4-24
!., 4-24
JOB TYPE, 6-52
Job View, 4-23
Job window, 4-14
Job..., 4-34
JOBTYPE, 6-51
Joint, Wld, Tool, User, 4-14

L
Light, 4-30
Lighting, 4-5, 4-12
Line Width, 4-6
Linear Motion, 6-28
LINK, 6-28
Link6 Reachability, 4-27
Load Cel..., 4-1
Load From JOB, 4-33
Load from Model File..., 4-34
Loading a Previous Version Cell, 5-1
Loc. Speed, 4-30
Local Coordinates, 1-3
Location, 4-33

M
Magnification, 4-30
Maintenance Safety, 2-6
Make Axis6 Model, 4-3
Make Jobs..., 4-24
Make Model..., 4-24
Making JOB..., 4-33
Making Jobs from Models, 5-14
Making Models from Jobs, 5-14
Manipulator Features, 6-28
Master Job Start, 4-13
Max, 4-31
Maximize, 4-4
Measure, 4-6, 4-27
Measure0, 4-12
Mechanical Safety Devices, 2-3
Menus, 4-1
Min, 4-31
MODEL, 6-29
Model, 4-6, 4-31
Model I/O, 4-8
Model Menu, 4-7
MODELDRV, 6-51, 6-54
Modeling Equipment and Production Parts, 5-5
Modeling Extension, 6-14
Modeling External Axes, 6-42
Modify Data, 4-3
Monitoring, 6-63
Motion keys, 4-14
MOTOSIM, 1-3
MOTOSIM Components, 1-2
MOTOSIM Description, 1-1
Move Dependent, 4-3
Move Org, 4-17
Move Parent..., 4-7
Move Tag..., 4-23
Move Time, 4-26
Move..., 4-14
Ms, 4-30
Multiple Robot Job Selection, 5-11
INDEX

N
Name, 4-16, 4-17
Name Number, 4-34
Near Clipping, 4-30
New, 4-1, 4-29
New Model, 4-7
New Window, 4-9
Non-Motion Programming, 6-22
NORM, 4-5
  INIT, 4-5
Normal, 4-15
Normal View, 4-13
Number, 4-18, 4-19, 4-20, 4-21
Number of Points, 4-29
Number Selected, 4-27

O
Off-line Programming, 1-1
Offset of Robot Frame, 4-28
OLE, 1-1
OLP, 1-3, 4-4
OLP Menu, 4-24
OLP Pick, 4-15
Opacity, 4-7, 4-28
  , 4-9
Open..., 4-1, 4-33
Opening a Cell File, 5-1, 5-11, 6-6
Operation, 4-33
Operation Object, 4-15
Operation Safety, 2-5
Operator Station
  Cycle Start, 4-9
  Emergency Stop (E-STOP), 4-9
  Master Job Start, 4-13
  Operator Station Enable/Disable, 4-14
  Positioner Auto/Manual, 4-10
Reset, 4-15
Option Divided, 4-27
Org Position, 4-33
Org Position and Pulse, 4-33
Org Pulse, 4-33
Other Solution, 4-15
Over Draw Wireframe, 4-6

P
Pan Speed, 4-30
Panorama, 4-13
Parent Model, 4-29
Part Motion Simulation, 6-22
Pause, 4-2, 5-11
Pen=1,2,3, 4-6
Pick Type, 4-15
Pipe Edit, 4-21
Play, 5-11
Play Panel, 4-8, 4-25
Play Statistics, 4-25
Play Time, 4-26
PlayBack, 4-2
Playing a Job, 5-10
Pneumatic Headstocks, 4-17
Pos 1..., 4-18, 4-19, 4-20, 4-21
Pos..., 4-30
POSC, 6-51, 6-53
POSCONE, 6-51
POSCONF, 6-52
POSE, 6-51, 6-53
Pose, 4-22
Position, 4-5, 4-13
Position and Job View, 4-8
Positioner
  Hard Stop/Shock Absorbers, 4-16
  Pneumatic Headstocks, 4-17
Positioner Auto/Manual, 4-10
Process Angle, 6-1
Process Angle (Frame Op.), 4-35
Programming Pendant, 1-1, 4-4
Programming Points Using the Programming
  Pendant, 5-8
Programming Safety, 2-4
Property, 4-29
Property..., 4-4, 4-7
Pulse, 4-22, 4-23
  !, 4-22
Pulse Record..., 4-3
PUTP, 6-51, 6-53
## INDEX

**Q**  
Quick Cycle Time, 4-34  
Quick Cycle Time Calc., 4-2

**R**  
Reach View..., 4-4  
, 4-1  
Recursible Call for Model Tree, 4-8  
Reference to Other Documentation, 1-4  
Refresh, 1-3  
Refresh Interval, 4-26  
Refresh Timer, 4-30  
Refresh View, 4-8  
Refresh Window, 4-24  
Registered Trace, 4-29  
Relative Path, 4-1  
Rename..., 4-7  
Replace Position Only, 4-23  
Reset, 4-15, 4-25  
Reset Job, 4-3, 5-11  
Results, 4-26  
RM2 Motion, 6-31  
ROBOT, 6-28  
Robot, 4-16  
Robot Accuracy, 1-3  
Robot Autoplacement, 6-50  
Robot Bar, 4-9  
Robot Frame, 4-23  
Robot Information, 4-28  
Robot Menu, 4-2  
Robot Repeatability, 1-3  
Robot’s TCP, 4-29  
Robotic Industries Association, 2-1  
Rotate, 4-6  
Rotate XY, 4-5, 4-12  
Rx, Ry, Rz(deg), 4-21

**S**  
Safety, 2-1  
Save, 4-1  
Save As..., 4-1  
Save..., 4-33  
Scale, 4-30  
Scroll & Zoom, 4-4, 4-12  
Search, 4-34  
Search Area Def..., 4-34  
Search Location, 4-34  
Search Start, 4-34  
Segment Clock, 4-26  
Select JOB, 4-2  
Select Model, 4-6, 4-12  
Select Teach, 4-34  
Selecting and Playing a Job, 5-11  
Servo, 4-9  
Set File Path..., 4-7  
Set I/O Object..., 4-7  
Set Parent..., 4-7  
Set Position..., 4-7  
Set Unit of Object..., 4-6  
Set User Frame..., 4-3  
Setting Unit Dialogs, 4-31  
Setting Up Inputs/Outputs for Multiple Robots, 6-59  
Shading, 4-30  
Shift, 4-6  
Showing CompuGauge Data, 6-70  
Side, 4-5  
Side View, 4-13  
Simulating Part Transfer, 6-25  
Simulation, 4-26  
Simulation Extension, 6-22  
Size Controls, 4-18, 4-19, 4-20, 4-21  
Skip Mode, 4-26  
Sliding, 4-12  
Smoothing, 4-28  
Speed, 4-18, 4-19, 4-20, 4-21, 4-32  
Sphere, 4-28  
Spin, 4-13  
StageMaster..., 4-8  
Standard Conventions, 2-2  
Start, 4-2, 4-34  
Status, 4-33, 4-34  
Status Bar, 4-9  
Step, 4-25  
Step Back, 4-2, 5-11  
Step Next, 4-2, 5-11  
Step Only, 4-25  
Stop, 4-34, 5-11
INDEX

S - continued
Straight, 4-19, 4-21
Supported Equipment/Operation, 1-2
Surface, 4-28, 4-30
Surface Mode, 4-5
Symmetry, 4-16
Sync, 4-14
Synch, 4-19, 4-20
Synchronize, 4-34
System Overview, 1-3
System Requirements, 1-2

T
TASK, 6-28
TBOXMOV, 6-51, 6-53
TBOXTEACH, 6-51, 6-53
Teach Menu, 4-33
Teacher, 1-3, 4-22
Teaching Robot Jobs Using Microsoft Excel, 6-1
Teaching Sheet, 1-3, 4-3
Terminology, 1-3
Tile, 4-9
To Point…, 4-4
TOOL, 6-51, 6-52
Tool Bar, 4-11
Tool Center Point (TCP, 4-27
Tool Data…, 4-3
Toolbar, 4-9
Top, 4-5
Top View, 4-13
T-Pendant, 4-8
Trace, 1-1, 1-3, 4-4, 4-27
Trace DLG Property, 4-29
Tree View, 4-8
TriSpectives Resident, 4-8
Type, 4-8

U
UF#Model…, 4-15
Uninstalling MOTOSIM and TriSpectives™, 3-4
Unit String, 4-31
Update, 4-31
Update (time), 1-3

User Defined, 4-3
User Frame, 4-23
User Frame Property…, 4-3
Using DLL’s to Model Motion, 6-43
Utilities, 6-50

V
Verifying I/O Connections, 6-60
Vertex, 4-6
Vertical, 4-27
View, 4-5
View Frames, 4-12
Virtual…, 4-24

W
Warm Up Time, 4-26
Width, 4-30
With Motion, 4-3
With Wireframe, 4-6
World Coordinates, 1-3

X
XYZ, 4-15