FANUC America Corporation SYSTEM R-30*i*B and R-30*i*B Mate Controller Line Tracking Setup and Operations Manual

MAROBLNTK04121E REV E

To be used with software V8.10 and later.

FANUC America Corporation
3900 W. Hamlin Road
Rochester Hills, Michigan 48309–3253

About This Manual

Copyrights and Trademarks

FANUC America Corporation
3900 W. Hamlin Road
Rochester Hills, MI 48309-3253

The descriptions and specifications contained in this manual were in effect at the time this manual was approved. FANUC America Corporation, hereinafter referred to as FANUC America, reserves the right to discontinue models at any time or to change specifications or design without notice and without incurring obligations.

FANUC America's manuals present descriptions, specifications, drawings, schematics, bills of material, parts, connections and/or procedures for installing, disassembling, connecting, operating and programming FANUC America Corporation's products and/or systems. Such systems consist of robots, extended axes, robot controllers, application software, the KAREL® programming language, INSIGHT® vision equipment, and special tools.

FANUC America recommends that only persons who have been trained in one or more approved FANUC America Training Course(s) be permitted to install, operate, use, perform procedures on, repair, and/or maintain FANUC America's products and/or systems and their respective components. Approved training necessitates that the courses selected be relevant to the type of system installed and application performed at the customer site.

Warning

This equipment generates, uses, and can radiate radio frequency energy and if not installed and used in accordance with the instruction manual, may cause interference to radio communications. As temporarily permitted by regulation, it has not been tested for compliance with the limits for Class A computing devices pursuant to subpart J of Part 15 of FCC Rules, which are designed to provide reasonable protection against such interference. Operation of the equipment in a residential area is likely to cause interference, in which case the user, at his own expense, will be required to take whatever measure may be required to correct the interference.

FANUC America conducts courses on its systems and products on a regularly scheduled basis at its headquarters in Rochester Hills, Michigan. For additional information contact

FANUC America Corporation
3900 W. Hamlin Road
Rochester Hills, MI 48309-3253

www.fanucamerica.com

For customer assistance, including Technical Support, Service, Parts & Part Repair, and Marketing Requests, contact the Customer Resource Center, 24 hours a day, at 1-800-47-ROBOT (1-800-477-6268). International customers should call 011-1-248-377-7159.

Send your comments and suggestions about this manual to: product.documentation@fanucamerica.com

Patents

One or more of the following U.S. patents might be related to the FANUC America products described in this manual.

FRA Patent List

4,630,567 4,639,878 4,707,647 4,708,175 4,708,580 4,942,539 4,984,745 5,238,029 5,239,739 5,272,805 5,293,107 5,293,911 5,331,264 5,367,944 5,373,221 5,421,218 5,434,489 5,644,898 5,670,202 5,696,687 5,737,218 5,823,389 5,853,027 5,887,800 5,941,679 5,959,425 5,987,726 6,059,092 6,064,168 6,070,109 6,086,294 6,122,062 6,147,323 6,204,620 6,243,621 6,253,799 6,285,920 6,313,595 6,325,302 6,345,818 6,356,807 6,360,143 6,378,190 6,385,508 6,425,177 6,477,913 6,490,369 6,518,980 6,540,104 6,541,757 6,560,513 6,569,258 6,612,449 6,703,079 6,705,361 6,726,773 6,768,078 6,845,295 6,945,483 7,149,606 7,149,606 7,211,978 7,266,422 7,399,363

FANUC LTD Patent List

4,571,694 4,626,756 4,700,118 4,706,001 4,728,872 4,732,526 4,742,207 4,835,362 4,894,596 4,899,095 4,920,248 4,931,617 4,934,504 4,956,594 4,967,125 4,969,109 4,970,370 4,970,448 4,979,127 5,004,968 5,006,035 5,008,834 5,063,281 5,066,847 5,066,902 5,093,552 5,107,716 5,111,019 5,130,515 5,136,223 5,151,608 5,170,109 5,189,351 5,267,483 5,274,360 5,292,066 5,300,868 5,304,906 5,313,563 5,319,443 5,325,467 5,327,057 5,329,469 5,333,242 5,337,148 5,371,452 5,375,480 5,418,441 5,432,316 5,440,213 5,442,155 5,444,612 5,449,875 5,451,850 5,461,478 5,463,297 5,467,003 5,471,312 5,479,078 5,485,389 5,485,552 5,486,679 5,489,758 5,493,192 5,504,766 5,511,007 5,520,062 5,528,013 5,532,924 5,548,194 5,552,687 5,558,196 5,561,742 5,570,187 5,570,190 5,572,103 5,581,167 5,582,750 5,587,635 5,600,759 5,608,299 5,608,618 5,624,588 5,630,955 5,637,969 5,639,204 5,641,415 5,650,078 5,658,121 5,668,628 5,687,295 5,691,615 5,698,121 5,708,342 5,715,375 5,719,479 5,727,132 5,742,138 5,742,144 5,748,854 5,749,058 5,760,560 5,773,950 5,783,922 5,799,135 5,812,408 5,841,257 5,845,053 5,872,894 5,887,122 5,911,892 5,912,540 5,920,678 5,937,143 5,980,082 5,983,744 5,987,591 5,988,850 6,023,044 6,032,086 6,040,554 6,059,169 6,088,628 6,097,169 6,114,824 6,124,693 6,140,788 6,141,863 6,157,155 6,160,324 6,163,124 6,177,650 6,180,898 6,181,096 6,188,194 6,208,105 6,212,444 6,219,583 6,226,181 6,236,011 6,236,896 6,250,174 6,278,902 6,279,413 6,285,921 6,298,283 6,321,139 6,324,443 6,328,523 6,330,493 6,340,875 6,356,671 6,377,869 6,382,012 6,384,371 6,396,030 6,414,711 6,424,883 6,431,018 6,434,448 6,445,979 6,459,958 6,463,358 6,484,067 6,486,629 6,507,165 6,654,666 6,665,588 6,680,461 6,696,810 6,728,417 6,763,284 6,772,493 6,845,296 6,853,881 6,888,089 6,898,486 6,917,837 6,928,337 6,965,091 6,970,802 7,038,165 7,069,808 7,084,900 7,092,791 7,133,747 7,143,100 7,149,602 7,131,848 7,161,321 7,171,041 7,174,234 7,173,213 7,177,722 7,177,439 7,181,294 7,181,313 7,280,687 7,283,661 7,291,806 7,299,713 7,315,650 7,324,873 7,328,083 7,330,777 7,333,879 7,355,725 7,359,817 7,373,220 7,376,488 7,386,367 7,464,623 7,447,615 7,445,260 7,474,939 7,486,816 7,495,192 7,501,778 7,502,504 7,508,155 7,512,459 7,525,273 7,526,121

VersaBell, ServoBell and SpeedDock Patents Pending.

Conventions

This manual includes information essential to the safety of personnel, equipment, software, and data. This information is indicated by headings and boxes in the text.

Warning

Information appearing under WARNING concerns the protection of personnel. It is boxed and in bold type to set it apart from other text.

Caution

Information appearing under **CAUTION** concerns the protection of equipment, software, and data. It is boxed to set it apart from other text.

Note Information appearing next to **NOTE** concerns related information or useful hints.

Contents

List of Figures

List of Tables

Safety

FANUC America Corporation is not and does not represent itself as an expert in safety systems, safety equipment, or the specific safety aspects of your company and/or its work force. It is the responsibility of the owner, employer, or user to take all necessary steps to guarantee the safety of all personnel in the workplace.

The appropriate level of safety for your application and installation can best be determined by safety system professionals. FANUC America Corporation therefore, recommends that each customer consult with such professionals in order to provide a workplace that allows for the safe application, use, and operation of FANUC America Corporation systems.

According to the industry standard ANSI/RIA R15-06, the owner or user is advised to consult the standards to ensure compliance with its requests for Robotics System design, usability, operation, maintenance, and service. Additionally, as the owner, employer, or user of a robotic system, it is your responsibility to arrange for the training of the operator of a robot system to recognize and respond to known hazards associated with your robotic system and to be aware of the recommended operating procedures for your particular application and robot installation.

Ensure that the robot being used is appropriate for the application. Robots used in classified (hazardous) locations must be certified for this use.

FANUC America Corporation therefore, recommends that all personnel who intend to operate, program, repair, or otherwise use the robotics system be trained in an approved FANUC America Corporation training course and become familiar with the proper operation of the system. Persons responsible for programming the system-including the design, implementation, and debugging of application programs-must be familiar with the recommended programming procedures for your application and robot installation.

The following guidelines are provided to emphasize the importance of safety in the workplace.

CONSIDERING SAFETY FOR YOUR ROBOT INSTALLATION

Safety is essential whenever robots are used. Keep in mind the following factors with regard to safety:

- The safety of people and equipment
- Use of safety enhancing devices
- Techniques for safe teaching and manual operation of the robot(s)
- Techniques for safe automatic operation of the robot(s)
- Regular scheduled inspection of the robot and workcell
- Proper maintenance of the robot

Keeping People Safe

The safety of people is always of primary importance in any situation. When applying safety measures to your robotic system, consider the following:

- External devices
- Robot(s)
- Tooling
- Workpiece

Using Safety Enhancing Devices

Always give appropriate attention to the work area that surrounds the robot. The safety of the work area can be enhanced by the installation of some or all of the following devices:

- Safety fences, barriers, or chains
- Light curtains
- Interlocks
- Pressure mats
- Floor markings
- Warning lights
- Mechanical stops
- EMERGENCY STOP buttons
- DEADMAN switches

Setting Up a Safe Workcell

A safe workcell is essential to protect people and equipment. Observe the following guidelines to ensure that the workcell is set up safely. These suggestions are intended to supplement and **not** replace existing federal, state, and local laws, regulations, and guidelines that pertain to safety.

- Sponsor your personnel for training in approved FANUC America Corporation training course(s) related to your application. Never permit untrained personnel to operate the robots.
- Install a lockout device that uses an access code to prevent unauthorized persons from operating the robot.
- Use anti-tie-down logic to prevent the operator from bypassing safety measures.
- Arrange the workcell so the operator faces the workcell and can see what is going on inside the cell.

- Clearly identify the work envelope of each robot in the system with floor markings, signs, and special barriers. The work envelope is the area defined by the maximum motion range of the robot, including any tooling attached to the wrist flange that extend this range.
- Position all controllers outside the robot work envelope.
- Never rely on software or firmware based controllers as the primary safety element unless they comply with applicable current robot safety standards.
- Mount an adequate number of EMERGENCY STOP buttons or switches within easy reach of the operator and at critical points inside and around the outside of the workcell.
- Install flashing lights and/or audible warning devices that activate whenever the robot is operating, that is, whenever power is applied to the servo drive system. Audible warning devices shall exceed the ambient noise level at the end-use application.
- Wherever possible, install safety fences to protect against unauthorized entry by personnel into the work envelope.
- Install special guarding that prevents the operator from reaching into restricted areas of the work envelope.
- Use interlocks.
- Use presence or proximity sensing devices such as light curtains, mats, and capacitance and vision systems to enhance safety.
- Periodically check the safety joints or safety clutches that can be optionally installed between the robot wrist flange and tooling. If the tooling strikes an object, these devices dislodge, remove power from the system, and help to minimize damage to the tooling and robot.
- Make sure all external devices are properly filtered, grounded, shielded, and suppressed to prevent hazardous motion due to the effects of electro-magnetic interference (EMI), radio frequency interference (RFI), and electro-static discharge (ESD).
- Make provisions for power lockout/tagout at the controller.
- Eliminate *pinch points* . Pinch points are areas where personnel could get trapped between a moving robot and other equipment.
- Provide enough room inside the workcell to permit personnel to teach the robot and perform maintenance safely.
- Program the robot to load and unload material safely.
- If high voltage electrostatics are present, be sure to provide appropriate interlocks, warning, and beacons.
- If materials are being applied at dangerously high pressure, provide electrical interlocks for lockout of material flow and pressure.

Staying Safe While Teaching or Manually Operating the Robot

Advise all personnel who must teach the robot or otherwise manually operate the robot to observe the following rules:

- Never wear watches, rings, neckties, scarves, or loose clothing that could get caught in moving machinery.
- Know whether or not you are using an intrinsically safe teach pendant if you are working in a hazardous environment.
- Before teaching, visually inspect the robot and *work envelope* to make sure that no potentially hazardous conditions exist. The work envelope is the area defined by the maximum motion range of the robot. These include tooling attached to the wrist flange that extends this range.
- The area near the robot must be clean and free of oil, water, or debris. Immediately report unsafe working conditions to the supervisor or safety department.
- FANUC America Corporation recommends that no one enter the work envelope of a robot that is on, except for robot teaching operations. However, if you must enter the work envelope, be sure all safeguards are in place, check the teach pendant DEADMAN switch for proper operation, and place the robot in teach mode. Take the teach pendant with you, turn it on, and be prepared to release the DEADMAN switch. Only the person with the teach pendant should be in the work envelope.

> **Warning**
>
> **Never bypass, strap, or otherwise deactivate a safety device, such as a limit switch, for any operational convenience. Deactivating a safety device is known to have resulted in serious injury and death.**

- Know the path that can be used to escape from a moving robot; make sure the escape path is never blocked.
- Isolate the robot from all remote control signals that can cause motion while data is being taught.
- Test any program being run for the first time in the following manner:

> **Warning**
>
> **Stay outside the robot work envelope whenever a program is being run. Failure to do so can result in injury.**

  — Using a low motion speed, single step the program for at least one full cycle.

  — Using a low motion speed, test run the program continuously for at least one full cycle.

  — Using the programmed speed, test run the program continuously for at least one full cycle.

- Make sure all personnel are outside the work envelope before running production.

Staying Safe During Automatic Operation

Advise all personnel who operate the robot during production to observe the following rules:

- Make sure all safety provisions are present and active.
- Know the entire workcell area. The workcell includes the robot and its work envelope, plus the area occupied by all external devices and other equipment with which the robot interacts.
- Understand the complete task the robot is programmed to perform before initiating automatic operation.
- Make sure all personnel are outside the work envelope before operating the robot.
- Never enter or allow others to enter the work envelope during automatic operation of the robot.
- Know the location and status of all switches, sensors, and control signals that could cause the robot to move.
- Know where the EMERGENCY STOP buttons are located on both the robot control and external control devices. Be prepared to press these buttons in an emergency.
- Never assume that a program is complete if the robot is not moving. The robot could be waiting for an input signal that will permit it to continue activity.
- If the robot is running in a pattern, do not assume it will continue to run in the same pattern.
- Never try to stop the robot, or break its motion, with your body. The only way to stop robot motion immediately is to press an EMERGENCY STOP button located on the controller panel, teach pendant, or emergency stop stations around the workcell.

Staying Safe During Inspection

When inspecting the robot, be sure to

- Turn off power at the controller.
- Lock out and tag out the power source at the controller according to the policies of your plant.
- Turn off the compressed air source and relieve the air pressure.
- If robot motion is not needed for inspecting the electrical circuits, press the EMERGENCY STOP button on the operator panel.
- Never wear watches, rings, neckties, scarves, or loose clothing that could get caught in moving machinery.
- If power is needed to check the robot motion or electrical circuits, be prepared to press the EMERGENCY STOP button, in an emergency.
- Be aware that when you remove a servomotor or brake, the associated robot arm will fall if it is not supported or resting on a hard stop. Support the arm on a solid support before you release the brake.

Staying Safe During Maintenance

When performing maintenance on your robot system, observe the following rules:

- Never enter the work envelope while the robot or a program is in operation.
- Before entering the work envelope, visually inspect the workcell to make sure no potentially hazardous conditions exist.
- Never wear watches, rings, neckties, scarves, or loose clothing that could get caught in moving machinery.
- Consider all or any overlapping work envelopes of adjoining robots when standing in a work envelope.
- Test the teach pendant for proper operation before entering the work envelope.
- If it is necessary for you to enter the robot work envelope while power is turned on, you must be sure that you are in control of the robot. Be sure to take the teach pendant with you, press the DEADMAN switch, and turn the teach pendant on. Be prepared to release the DEADMAN switch to turn off servo power to the robot immediately.
- Whenever possible, perform maintenance with the power turned off. Before you open the controller front panel or enter the work envelope, turn off and lock out the 3-phase power source at the controller.
- Be aware that an applicator bell cup can continue to spin at a very high speed even if the robot is idle. Use protective gloves or disable bearing air and turbine air before servicing these items.
- Be aware that when you remove a servomotor or brake, the associated robot arm will fall if it is not supported or resting on a hard stop. Support the arm on a solid support before you release the brake.

Warning

Lethal voltage is present in the controller WHENEVER IT IS CONNECTED to a power source. Be extremely careful to avoid electrical shock. HIGH VOLTAGE IS PRESENT at the input side whenever the controller is connected to a power source. Turning the disconnect or circuit breaker to the OFF position removes power from the output side of the device only.

- Release or block all stored energy. Before working on the pneumatic system, shut off the system air supply and purge the air lines.
- Isolate the robot from all remote control signals. If maintenance must be done when the power is on, make sure the person inside the work envelope has sole control of the robot. The teach pendant must be held by this person.
- Make sure personnel cannot get trapped between the moving robot and other equipment. Know the path that can be used to escape from a moving robot. Make sure the escape route is never blocked.

- Use blocks, mechanical stops, and pins to prevent hazardous movement by the robot. Make sure that such devices do not create pinch points that could trap personnel.

> **Warning**
>
> **Do not try to remove any mechanical component from the robot before thoroughly reading and understanding the procedures in the appropriate manual. Doing so can result in serious personal injury and component destruction.**

- Be aware that when you remove a servomotor or brake, the associated robot arm will fall if it is not supported or resting on a hard stop. Support the arm on a solid support before you release the brake.
- When replacing or installing components, make sure dirt and debris do not enter the system.
- Use only specified parts for replacement. To avoid fires and damage to parts in the controller, never use nonspecified fuses.
- Before restarting a robot, make sure no one is inside the work envelope; be sure that the robot and all external devices are operating normally.

KEEPING MACHINE TOOLS AND EXTERNAL DEVICES SAFE

Certain programming and mechanical measures are useful in keeping the machine tools and other external devices safe. Some of these measures are outlined below. Make sure you know all associated measures for safe use of such devices.

Programming Safety Precautions

Implement the following programming safety measures to prevent damage to machine tools and other external devices.

- Back-check limit switches in the workcell to make sure they do not fail.
- Implement "failure routines" in programs that will provide appropriate robot actions if an external device or another robot in the workcell fails.
- Use *handshaking* protocol to synchronize robot and external device operations.
- Program the robot to check the condition of all external devices during an operating cycle.

Mechanical Safety Precautions

Implement the following mechanical safety measures to prevent damage to machine tools and other external devices.

- Make sure the workcell is clean and free of oil, water, and debris.
- Use DCS (Dual Check Safety), software limits, limit switches, and mechanical hardstops to prevent undesired movement of the robot into the work area of machine tools and external devices.

KEEPING THE ROBOT SAFE

Observe the following operating and programming guidelines to prevent damage to the robot.

Operating Safety Precautions

The following measures are designed to prevent damage to the robot during operation.

- Use a low override speed to increase your control over the robot when jogging the robot.
- Visualize the movement the robot will make before you press the jog keys on the teach pendant.
- Make sure the work envelope is clean and free of oil, water, or debris.
- Use circuit breakers to guard against electrical overload.

Programming Safety Precautions

The following safety measures are designed to prevent damage to the robot during programming:

- Establish *interference zones* to prevent collisions when two or more robots share a work area.
- Make sure that the program ends with the robot near or at the home position.
- Be aware of signals or other operations that could trigger operation of tooling resulting in personal injury or equipment damage.
- In dispensing applications, be aware of all safety guidelines with respect to the dispensing materials.

  Note Any deviation from the methods and safety practices described in this manual must conform to the approved standards of your company. If you have questions, see your supervisor.

ADDITIONAL SAFETY CONSIDERATIONS FOR PAINT ROBOT INSTALLATIONS

Process technicians are sometimes required to enter the paint booth, for example, during daily or routine calibration or while teaching new paths to a robot. Maintenance personal also must work inside the paint booth periodically.

Whenever personnel are working inside the paint booth, ventilation equipment must be used. Instruction on the proper use of ventilating equipment usually is provided by the paint shop supervisor.

Although paint booth hazards have been minimized, potential dangers still exist. Therefore, today's highly automated paint booth requires that process and maintenance personnel have full awareness of the system and its capabilities. They must understand the interaction that occurs between the vehicle moving along the conveyor and the robot(s), hood/deck and door opening devices, and high-voltage electrostatic tools.

Caution

Ensure that all ground cables remain connected. Never operate the paint robot with ground provisions disconnected. Otherwise, you could injure personnel or damage equipment.

Paint robots are operated in three modes:

- Teach or manual mode
- Automatic mode, including automatic and exercise operation
- Diagnostic mode

During both teach and automatic modes, the robots in the paint booth will follow a predetermined pattern of movements. In teach mode, the process technician teaches (programs) paint paths using the teach pendant.

In automatic mode, robot operation is initiated at the System Operator Console (SOC) or Manual Control Panel (MCP), if available, and can be monitored from outside the paint booth. All personnel must remain outside of the booth or in a designated safe area within the booth whenever automatic mode is initiated at the SOC or MCP.

In automatic mode, the robots will execute the path movements they were taught during teach mode, but generally at production speeds.

When process and maintenance personnel run diagnostic routines that require them to remain in the paint booth, they must stay in a designated safe area.

Paint System Safety Features

Process technicians and maintenance personnel must become totally familiar with the equipment and its capabilities. To minimize the risk of injury when working near robots and related equipment, personnel must comply strictly with the procedures in the manuals.

This section provides information about the safety features that are included in the paint system and also explains the way the robot interacts with other equipment in the system.

The paint system includes the following safety features:

- Most paint booths have red warning beacons that illuminate when the robots are armed and ready to paint. Your booth might have other kinds of indicators. Learn what these are.
- Some paint booths have a blue beacon that, when illuminated, indicates that the electrostatic devices are enabled. Your booth might have other kinds of indicators. Learn what these are.
- EMERGENCY STOP buttons are located on the robot controller and teach pendant. Become familiar with the locations of all E-STOP buttons.
- An intrinsically safe teach pendant is used when teaching in hazardous paint atmospheres.
- A DEADMAN switch is located on each teach pendant. When this switch is held in, and the teach pendant is on, power is applied to the robot servo system. If the engaged DEADMAN switch is released during robot operation, power is removed from the servo system, all axis brakes are applied, and the robot comes to an EMERGENCY STOP. Safety interlocks within the system might also E-STOP other robots.

Warning

An EMERGENCY STOP will occur if the DEADMAN switch is released on a bypassed robot.

- Overtravel by robot axes is prevented by software limits. All of the major and minor axes are governed by software limits. DCS (Dual Check Safety), limit switches and hardstops also limit travel by the major axes.
- EMERGENCY STOP limit switches and photoelectric eyes might be part of your system. Limit switches, located on the entrance/exit doors of each booth, will EMERGENCY STOP all equipment in the booth if a door is opened while the system is operating in automatic or manual mode. For some systems, signals to these switches are inactive when the switch on the SOC is in teach mode.When present, photoelectric eyes are sometimes used to monitor unauthorized intrusion through the entrance/exit silhouette openings.
- System status is monitored by computer. Severe conditions result in automatic system shutdown.

Staying Safe While Operating the Paint Robot

When you work in or near the paint booth, observe the following rules, in addition to all rules for safe operation that apply to all robot systems.

Warning

Observe all safety rules and guidelines to avoid injury.

Warning

Never bypass, strap, or otherwise deactivate a safety device, such as a limit switch, for any operational convenience. Deactivating a safety device is known to have resulted in serious injury and death.

Warning

Enclosures shall not be opened unless the area is known to be nonhazardous or all power has been removed from devices within the enclosure. Power shall not be restored after the enclosure has been opened until all combustible dusts have been removed from the interior of the enclosure and the enclosure purged. Refer to the Purge chapter for the required purge time.

- Know the work area of the entire paint station (workcell).
- Know the work envelope of the robot and hood/deck and door opening devices.
- Be aware of overlapping work envelopes of adjacent robots.
- Know where all red, mushroom-shaped EMERGENCY STOP buttons are located.
- Know the location and status of all switches, sensors, and/or control signals that might cause the robot, conveyor, and opening devices to move.
- Make sure that the work area near the robot is clean and free of water, oil, and debris. Report unsafe conditions to your supervisor.
- Become familiar with the complete task the robot will perform BEFORE starting automatic mode.
- Make sure all personnel are outside the paint booth before you turn on power to the robot servo system.
- Never enter the work envelope or paint booth before you turn off power to the robot servo system.
- Never enter the work envelope during automatic operation unless a safe area has been designated.
- Never wear watches, rings, neckties, scarves, or loose clothing that could get caught in moving machinery.

- Remove all metallic objects, such as rings, watches, and belts, before entering a booth when the electrostatic devices are enabled.
- Stay out of areas where you might get trapped between a moving robot, conveyor, or opening device and another object.
- Be aware of signals and/or operations that could result in the triggering of guns or bells.
- Be aware of all safety precautions when dispensing of paint is required.
- Follow the procedures described in this manual.

Special Precautions for Combustible Dusts (powder paint)

When the robot is used in a location where combustible dusts are found, such as the application of powder paint, the following special precautions are required to insure that there are no combustible dusts inside the robot.

- Purge maintenance air should be maintained at all times, even when the robot power is off. This will insure that dust can not enter the robot.
- A purge cycle will not remove accumulated dusts. Therefore, if the robot is exposed to dust when maintenance air is not present, it will be necessary to remove the covers and clean out any accumulated dust. Do not energize the robot until you have performed the following steps.

1. Before covers are removed, the exterior of the robot should be cleaned to remove accumulated dust.
2. When cleaning and removing accumulated dust, either on the outside or inside of the robot, be sure to use methods appropriate for the type of dust that exists. Usually lint free rags dampened with water are acceptable. Do not use a vacuum cleaner to remove dust as it can generate static electricity and cause an explosion unless special precautions are taken.
3. Thoroughly clean the interior of the robot with a lint free rag to remove any accumulated dust.
4. When the dust has been removed, the covers must be replaced immediately.
5. Immediately after the covers are replaced, run a complete purge cycle. The robot can now be energized.

Staying Safe While Operating Paint Application Equipment

When you work with paint application equipment, observe the following rules, in addition to all rules for safe operation that apply to all robot systems.

> **Warning**
>
> **When working with electrostatic paint equipment, follow all national and local codes as well as all safety guidelines within your organization. Also reference the following standards: *NFPA 33 Standards for Spray Application Using Flammable or Combustible Materials* , and *NFPA 70 National Electrical Code* .**

- **Grounding**: All electrically conductive objects in the spray area must be grounded. This includes the spray booth, robots, conveyors, workstations, part carriers, hooks, paint pressure pots, as well as solvent containers. Grounding is defined as the object or objects shall be electrically connected to ground with a resistance of not more than 1 megohms.
- **High Voltage**: High voltage should only be on during actual spray operations. Voltage should be off when the painting process is completed. Never leave high voltage on during a cap cleaning process.
- Avoid any accumulation of combustible vapors or coating matter.
- Follow all manufacturer recommended cleaning procedures.
- Make sure all interlocks are operational.
- No smoking.
- Post all warning signs regarding the electrostatic equipment and operation of electrostatic equipment according to NFPA 33 Standard for Spray Application Using Flammable or Combustible Material.
- Disable all air and paint pressure to bell.
- Verify that the lines are not under pressure.

Staying Safe During Maintenance

When you perform maintenance on the painter system, observe the following rules, and all other maintenance safety rules that apply to all robot installations. Only qualified, trained service or maintenance personnel should perform repair work on a robot.

- Paint robots operate in a potentially explosive environment. Use caution when working with electric tools.
- When a maintenance technician is repairing or adjusting a robot, the work area is under the control of that technician. All personnel not participating in the maintenance must stay out of the area.
- For some maintenance procedures, station a second person at the control panel within reach of the EMERGENCY STOP button. This person must understand the robot and associated potential hazards.
- Be sure all covers and inspection plates are in good repair and in place.
- Always return the robot to the “home” position before you disarm it.

- Never use machine power to aid in removing any component from the robot.
- During robot operations, be aware of the robot’s movements. Excess vibration, unusual sounds, and so forth, can alert you to potential problems.
- Whenever possible, turn off the main electrical disconnect before you clean the robot.
- When using vinyl resin observe the following:
  - — Wear eye protection and protective gloves during application and removal
  - — Adequate ventilation is required. Overexposure could cause drowsiness or skin and eye irritation.
  - — If there is contact with the skin, wash with water.
  - — Follow the Original Equipment Manufacturer’s Material Safety Data Sheets.
- When using paint remover observe the following:
  - — Eye protection, protective rubber gloves, boots, and apron are required during booth cleaning.
  - — Adequate ventilation is required. Overexposure could cause drowsiness.
  - — If there is contact with the skin or eyes, rinse with water for at least 15 minutes. Then, seek medical attention as soon as possible.
  - — Follow the Original Equipment Manufacturer’s Material Safety Data Sheets.

Chapter 1

OVERVIEW

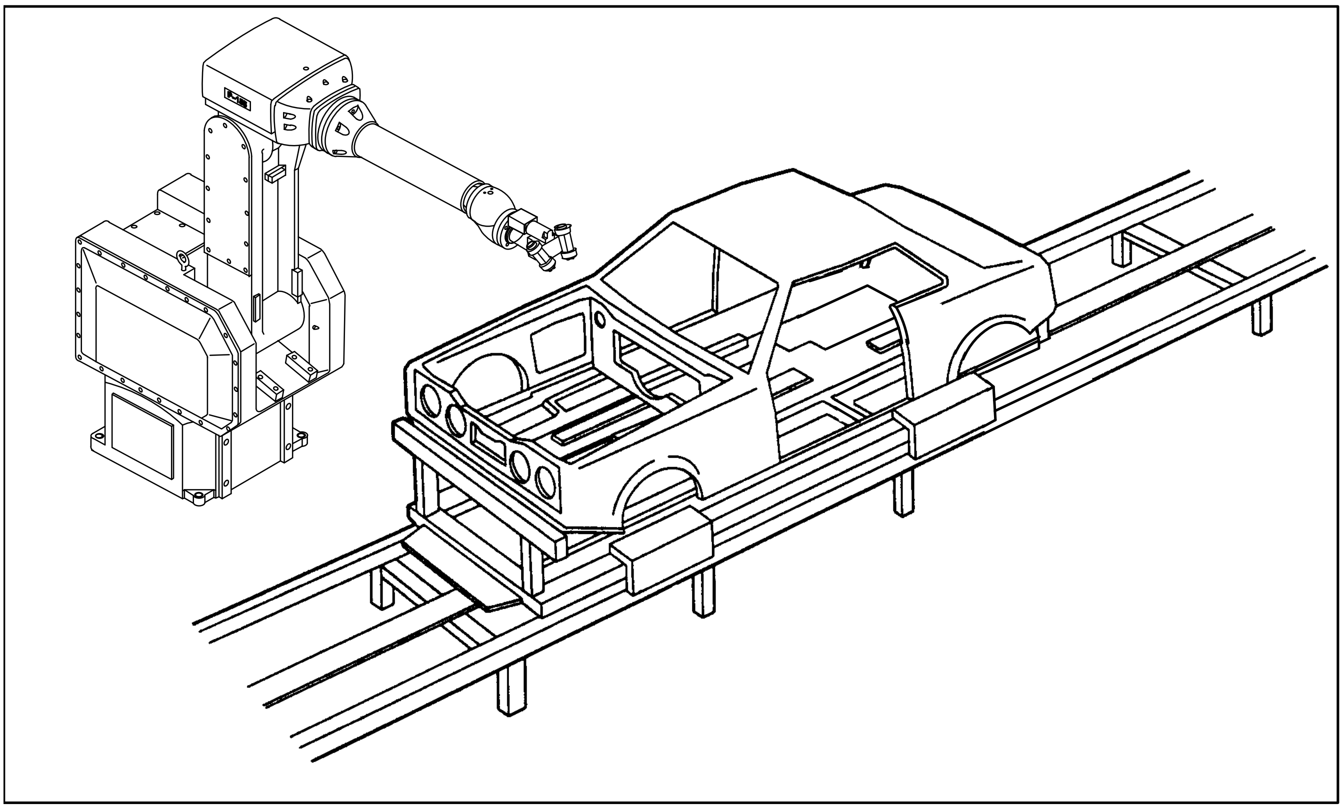
Contents

1.1 OVERVIEW

Tracking is an optional feature that enables a robot to treat a moving workpiece as a stationary object. The option is used in conveyor applications, where the robot must perform tasks on moving workpieces without stopping the assembly line. See Figure 1–1 .

This user guide provides information for the installation and operation of the FANUC America Corporation SYSTEM R-30*i*A Teach Pendant Program (TPP) Line Tracking option. This feature provides a complete stand-alone environment for all teach pendant-based line tracking, with teach pendant SETUP screen access to tracking parameters and teach pendant instructions for tracking program execution.

Figure 1–1. Typical Tracking Workcell



1.2 GENERAL TRACKING DESCRIPTIONS

Tracking refers to the option for tracking an assembly line. In this environment, the robot must track and manipulate a workpiece which is moving through its workspace on a conveyor, platform, or other mechanism.

Tracking saves production time by allowing workpieces to continue to move on the conveyor, instead of requiring them to be removed from the conveyor and placed in a stationary fixture. Tracking can also increase the working volume of the robot's workspace if you carefully segment the program into various regions or windows. Each region lies within the robot's workspace, at some time, as the workpiece moves past the robot.

Tracking can be accomplished in two ways:

- Single axis line tracking
- Cartesian line or circular tracking

1.3 SINGLE-AXIS (RAIL) TRACKING

In single axis tracking, the position of the robot's extended axis (an integrated or non-integrated base axis) is adjusted to track the motion of a linear conveyor. The conveyor motion direction must be parallel to that of the tracking axis.

This single-axis tracking is known as *rail tracking* , since the typical application uses a rail or platform to perform the tracking motion. With rail tracking, the robot arm configuration (excluding the tracking axis) remains as programmed. All types of motion (Linear, Circular, and Joint) are allowed.

Rail tracking is a simple method of dealing with a constantly moving workpiece. Rail tracking is used in large systems that can occupy a large amount of floor space. It is easy to teach and works with almost any application. This option allows a large volume of work to be accomplished by one system.

1.4 CARTESIAN TRACKING

Cartesian tracking refers to a stationary robot whose Tool Center Point (TCP) position is adjusted to track the motion of a conveyor. You should use Cartesian tracking whenever floor space is a primary concern, or if you cannot install a rail axis for tracking.

You can increase the work capacity of a robot by teaching paths efficiently. You can also reduce overall cycle time by using the motion of the conveyor to increase the robot workspace and decrease the time needed to complete a path.

With Cartesian tracking, the arm configuration of the main robot axes (not including any extended axes which might be present) is changed to achieve the tracking motion. Because of this, Cartesian tracking is restricted to Linear and Circular program motions. Joint motions are not supported.

There are two kinds of Cartesian tracking: Line and Circular (not to be confused with Linear and Circular motions). These are described in Section 1.4.1 and Section 1.4.2 respectively.

Note Program path planning and teaching is critical for Cartesian tracking. Inefficient paths can restrict robot movement around the workpiece, possibly reducing the workspace. In addition, the joint trajectories of the robot will rarely be the same during program execution as during program teaching due to the motion of the conveyor. Refer to Chapter 4 *PLANNING AND CREATING A PROGRAM* and Chapter 5 *ADVANCED TECHNIQUES* before you attempt to teach a tracking path.

Note Cartesian tracking only supports integrated extended axes.

1.4.1 LINE Tracking

Cartesian line tracking consists of a robot and a linear conveyor which moves parts past a robot. The robot is usually mounted on a stationary pedestal beside the conveyor, where it can easily reach the parts as they move past it. The robot can also be mounted above or below the conveyor, or on a rail or other integrated extended axis depending on the needs of the application.

1.4.2 CIRCULAR Tracking

Cartesian circular tracking consists of a circular conveyor or rotary table which moves parts past a robot. The robot can be located either inside or outside the circle of the conveyor. The robot can also be mounted above or below the conveyor, or on a rail or other integrated extended axis depending on the needs of the application.

Note Boundary function will be supported when the function is enabled. To enable boundary checking set $LNCFG.$COMP_SW2 bit 0x10 on (OR this system variable with 0x10)

Note Only Linear program motion is supported for circular tracking. Circular and Joint program motion is not permitted.

Chapter 2

HARDWARE AND SOFTWARE

Contents

2.1 REQUIREMENTS

When the αS-1000 encoder is used, it can be directly plugged into the Main CPU board. You might not need a line tracking interface board. When the αS-1000 encoder is not used, the line tracking system requires a line tracking interface board within the controller, and a fiber optic FSSB connection cable.

Additionally, external hardware (a tracking encoder) and the associated interconnections (a pulse coder cable) are required to track the line (conveyor, platform, table, and so forth). See Figure 2–5 .

Finally, another external mechanism (a sensor or part detect switch) must be installed to detect the presence of a part traveling on the conveyor as it approaches the robot workspace. This must be wired into a controller digital input card.

Hardware

The line tracking system requires the items shown in Table 2–1 .

Table 2–1. Requirements

Required Components	R-30*i*B Mate Controller	R-30*i*B Controller A-Cabinet	R-30*i*B Controller B-Cabinet	Comments
Line Tracking Interface Board	A05B-2650-J035 (mini slot) 2ch	A05B-2600-J035 (wide-mini slot) or A05B-2600-J036 (mini slot)	A05B-2600-J035 (wide-mini slot) or A05B-2600-J037 (mini slot)	This order specification includes Line tracking I/F board and optical fiber cable.
Separate detector interface unit (SDU1)	N/A	A02B-0323-C205	A02B-0323-C205	If you use 5 or more Pulsecoders, it needs not only SDU1 (A02B-0323-C205) but also SDU2 (A02B-0323-C204) and the cable connection between basic unit (SDU1) and expansion unit (SDU2) (A02B-0236-K831) Also it is required to consider optical fiber according to SDU's installation position. **Note** SDU requires retrofit work to mount in the container (see Figure 2–6 through Figure 2–14).

Table 2–1. Requirements (Cont'd)

Required Components	R-30*i*B Mate Controller	R-30*i*B Controller A-Cabinet	R-30*i*B Controller B-Cabinet	Comments
Pulsecoder Cable (Incremental)	N/A	for A20B-8101-0421 (wide-mini slot): A05B-2601-J380 (7M) A05B-2601-J381 (14M) A05B-2601-J382 (20M) A05B-2601-J383 (30M)	for A20B-8101-0421 (wide-mini slot): A05B-2603-J380 (7M) A05B-2603-J381 (14M) A05B-2603-J382 (20M) A05B-2603-J383 (30M)	
	(mini slot) (one Pulsecoder): A05B-2650-J200 (7M) A05B-2650-J201 (14M) A05B-2650-J202 (20M)	for A20B-8101-0601 (mini slot) (one Pulsecoder): A05B-2601-J370 (7M) A05B-2601-J371 (14M) A05B-2601-J372 (20M) A05B-2601-J373 (30M)	for A20B-8101-0601 (mini slot) (one Pulsecoder): A05B-2603-J370 (7M) A05B-2603-J371 (14M) A05B-2603-J372 (20M) A05B-2603-J373 (30M)	
	(mini slot) (two Pulsecoder): A05B-2650-J210 (7M) A05B-2650-J211 (14M) A05B-2650-J212 (20M)	for A20B-8101-0601 (mini slot) (two Pulsecoder): A05B-2601-J360 (7M) A05B-2601-J361 (14M) A05B-2601-J362 (20M) A05B-2601-J363 (30M)	for A20B-8101-0601 (mini slot) (two Pulsecoder): A05B-2603-J360 (7M) A05B-2603-J361 (14M) A05B-2603-J362 (20M) A05B-2603-J363 (30M)	

Table 2–1. Requirements (Cont'd)

Required Components	R-30*i*B Mate Controller	R-30*i*B Controller A-Cabinet	R-30*i*B Controller B-Cabinet	Comments
Pulsecoder Cable (αA1000S Pulsecoder)	N/A	for A05B-2600-J035 (wide-mini slot): A05B-2601-J220 (7M) A05B-2601-J221 (14M) A05B-2601-J222 (20M) A05B-2601-J223 (30M)	for A05B-2600-J035 (wide-mini slot): A05B-2603-J220 (7M) A05B-2603-J221 (14M) A05B-2603-J222 (20M) A05B-2603-J223 (30M)	In case of a αA1000S Pulsecoder as an absolute Pulsecoder, cables in this list can not be used.
	A05B-2650-J205 (7M) A05B-2650-J206 (14M) A05B-2650-J207 (20M)	for A05B-2600-J036 (mini slot) (one Pulsecoder): A05B-2601-J210 (7M) A05B-2601-J211 (14M) A05B-2601-J212 (20M) A05B-2601-J213 (30M)	for A05B-2600-J037 (mini slot) (one Pulsecoder): A05B-2603-J210 (7M) A05B-2603-J211 (14M) A05B-2603-J212 (20M) A05B-2603-J213 (30M)	
	A05B-2650-J215 (7M) A05B-2650-J216 (14M) A05B-2650-J217 (20M)	for A20B-2600-J036 (mini slot) (two Pulsecoder): A05B-2601-J260 (7M) A05B-2601-J261 (14M) A05B-2601-J262 (20M) A05B-2601-J263 (30M)	for A20B-2600-J037 (mini slot) (two Pulsecoder) : A05B-2603-J260 (7M) A05B-2603-J261 (14M) A05B-2603-J262 (20M) A05B-2603-J263 (30M)	
	Main Board CRS41 Port: A05B-2650-J220 (7M) A05B-2650-J221 (14M) A05B-2650-J222 (20M)	Main Board JD17 Port: A05B-2601-J270 (7M) A05B-2601-J271 (14M) A05B-2601-J272 (20M) A05B-2601-J273 (30M)	Main Board JD17 Port: A05B-2603-J270 (7M) A05B-2603-J271 (14M) A05B-2603-J272 (20M)	

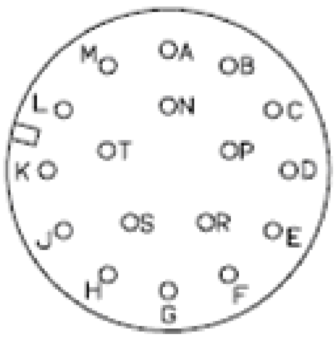
Table 2–1. Requirements (Cont'd)

Required Components	R-30*i*B Mate Controller	R-30*i*B Controller A-Cabinet	R-30*i*B Controller B-Cabinet	Comments
			A05B-2603-J273 (30M)	
Multiplexer cable	N/A	for A20B-8101-0421 (wide-mini slot): A05B-2601-J385 (7M) A05B-2601-J386 (14M) A05B-2601-J387 (20M) A05B-2601-J388 (30M)	for A20B-8101-0421 (wide-mini slot): A05B-2603-J385 (7M) A05B-2603-J386 (14M) A05B-2603-J387 (20M) A05B-2603-J388 (30M)	
	N/A	for A20B-8101-0601 (mini slot): A05B-2601-J375 (7M) A05B-2601-J376 (14M) A05B-2601-J377 (20M) A05B-2601-J378 (30M)	for A20B-8101-0601 (mini slot): A05B-2603-J375 (7M) A05B-2603-J376 (14M) A05B-2603-J377 (20M) A05B-2603-J378 (30M)	When using multiple robots to track parts on the conveyor, use a Multiplexer to Controller cable, or use Ethernet encoder function (option).
Hardware				
Incremental Pulsecoder to Multiplexer cable		A05B-2451-K102(7M) A05B-2451-K103(14M)		

Note When using multiple robots to track parts on the conveyor, use Ethernet encoder function (option). (Pulsecoder can not be used in using αA1000S Pulsecoder, A860-0372-T001.)

Figure 2–1. αA1000S pulsecoder (A860-0372-T001) Connection Signal Information

A860-0372-T001

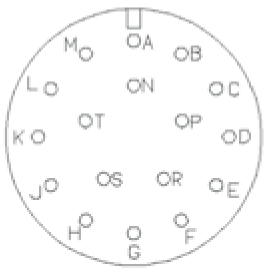


3102A20-29PW

Signal Name	Pin No. α A1000S 3102A20-29P
SD	A
*SD	D
REQ	F
*REQ	G
+5V	J,K
0V	N,T
Shield	H
+6VA	R
0VA	S

Figure 2–2. Pulsecoder (A860-0301-T001) Connection Signal Information

A860-0301-T001



3102A20-29P

Signal Name	Pin No. 310A20-29P
A	A
*A	D
B	B
*B	E
Z	F
*Z	G
C1	-
C2	-
C4	-
C8	-
+5V	C,J,K
0V	N,P,T
Shield	H
OH1	
OH2	
REQ	
+6VA	
0VA	

Table 2–2. Pulsecoder Specifications

Item	αA1000S Pulsecoder (A860-0372-T001)	Incremental Pulsecoder (A860-0301-T001)
Power voltage	5 (V) ±5%	5 (V) 5%
Current comsumption	Up to 0.3 (A)	Up to 0.35 (A)
Working temperature range	0 to +60 (deg C)	0 to +60 (deg C)
Resolution	1 000 000[/rev]	N/A
Maximum speed of revolution	4000[min^{-1}]	N/A

Table 2–2. Pulsecoder Specifications (Cont'd)

Item		αA1000S Pulsecoder (A860-0372-T001)	Incremental Pulsecoder (A860-0301-T001)
Maximum response frequency		N/A	100x 10^3 (Hz)
Input shaft inertia		Up to 1 x 10^{-4}[kg m^2]	Up to 5x 10^{-3} (kg • m^2)
Input shaft startup torque		Up to 0.1[N m]	Up to 0.8 (Nm)
Rated loads	Radial	2.0[N]	20 (N)
	Axial	1.0[N]	10 (N)
Shaft diameter runout		0.02x10^3[m]	0.02x 10^{-3} (m)
Configuration		Dust proof and drip-proof (equivalent to IP55: when using waterproof connector)	N/A
Vibration Proof Acceleration		5[G] (50 to 2000[Hz])	N/A
Weight		Approx. 0.75[kg]	Approx. 2.0 (kg)

Figure 2–3. αA1000S Pulsecoder Dimensions (A860-0372-T001)

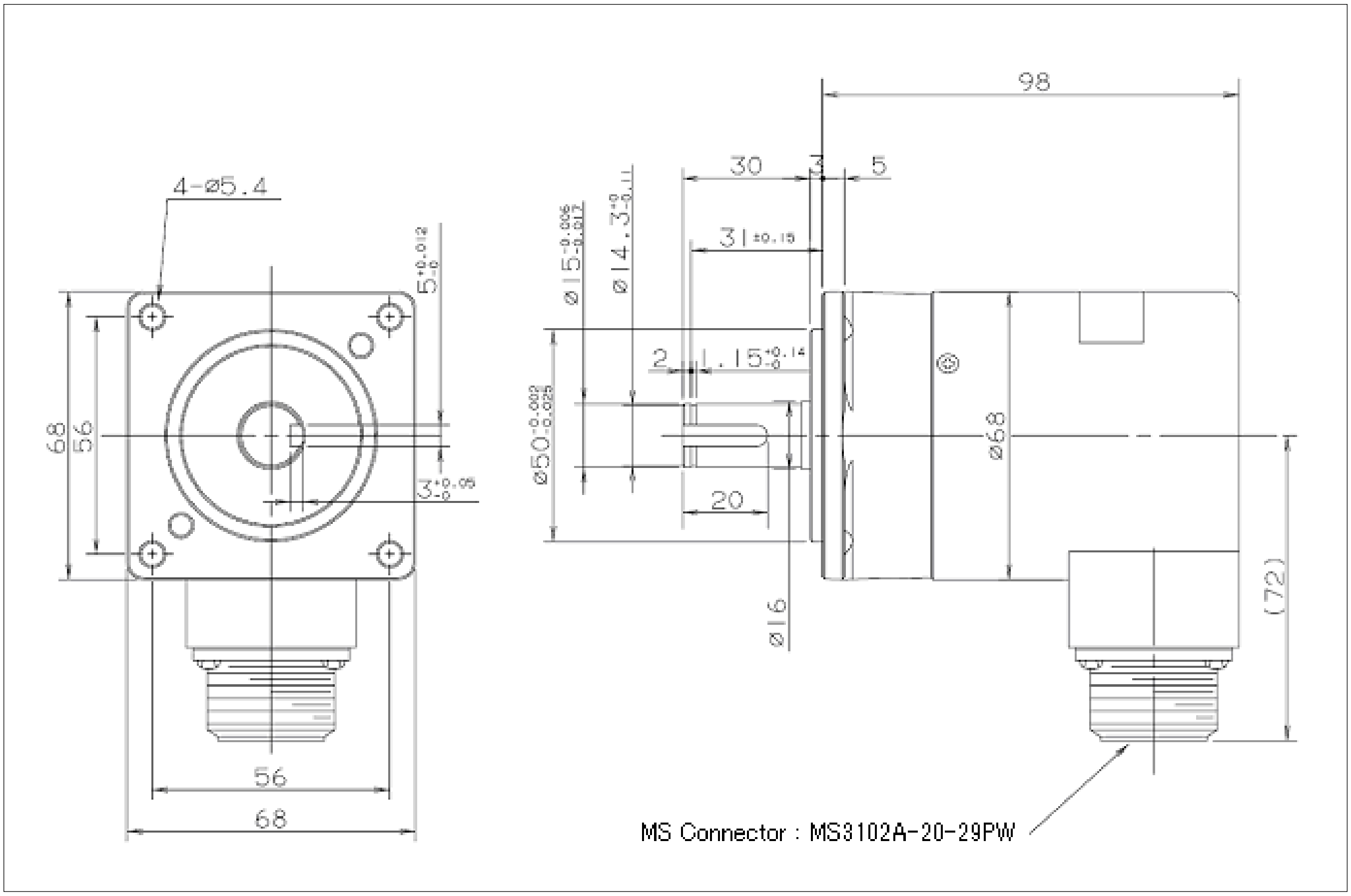


Figure 2–4. Pulsecoder Dimensions (A860-0301-T001)

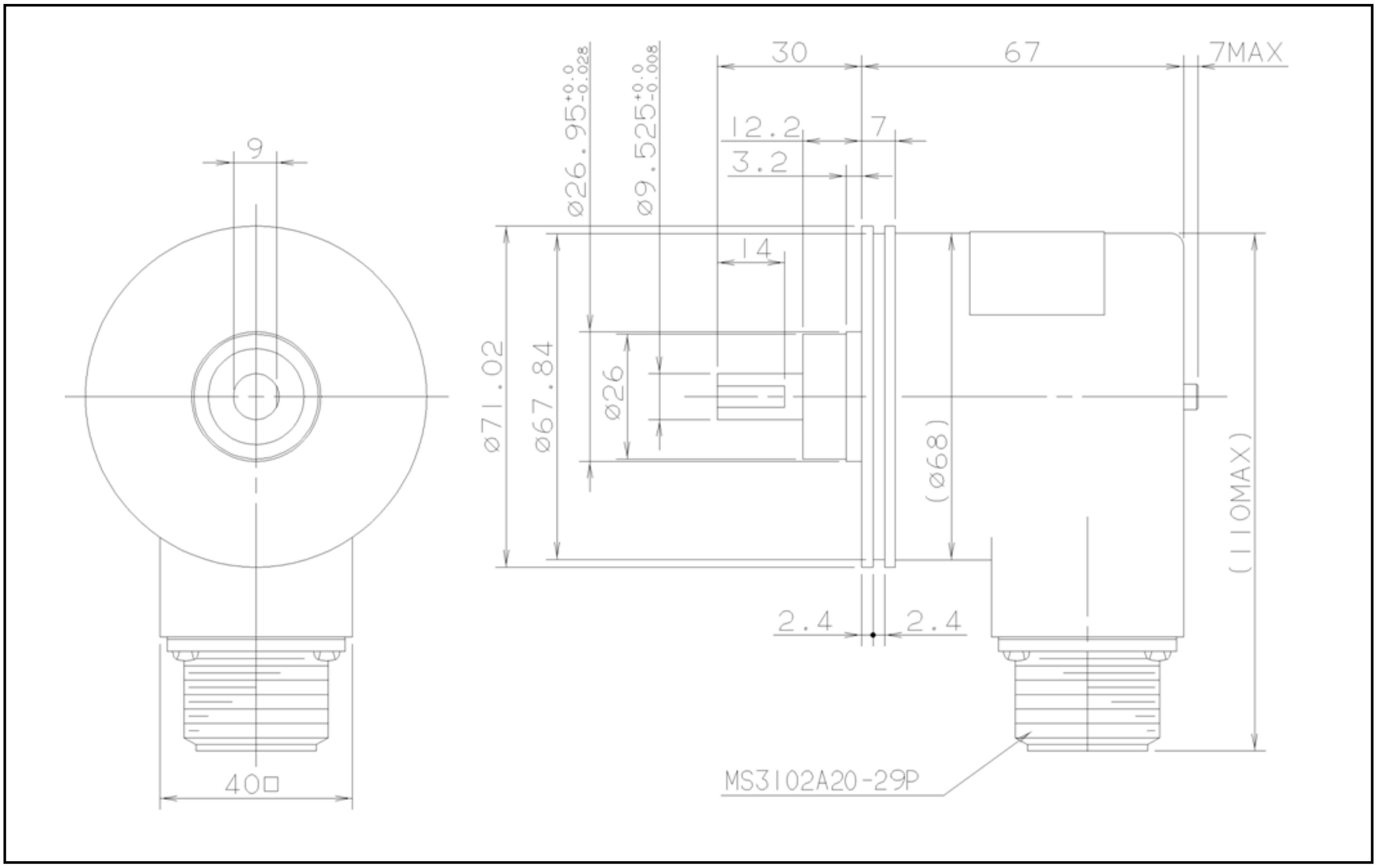
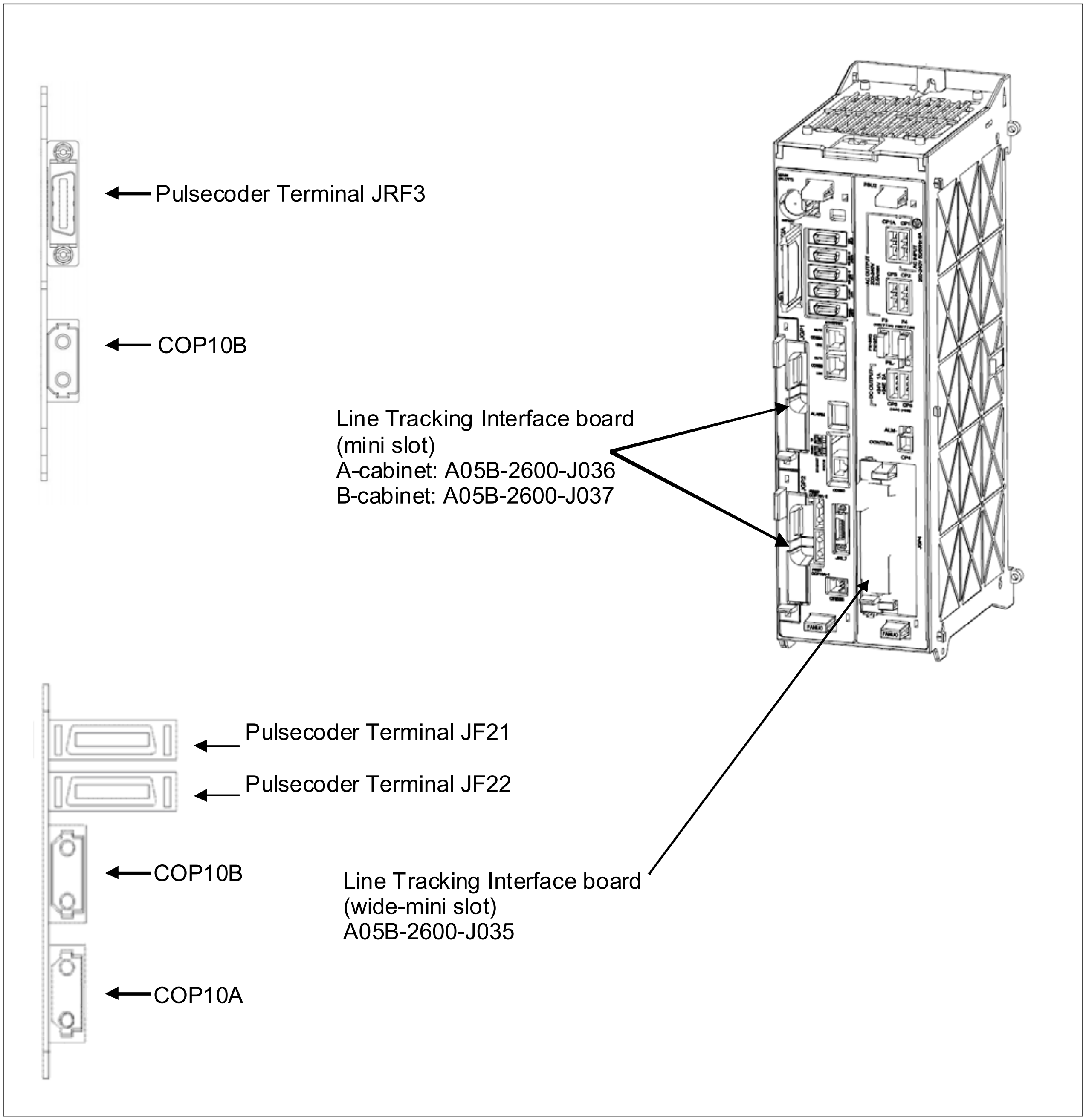


Figure 2–5. Controller with Line Tracking Connections



Note If the line tracking interface board cannot be used or is not available, you can use the SDU shown in Figure 2–6 .

Figure 2–6. External Dimensions of Separate Detector Interface Unit

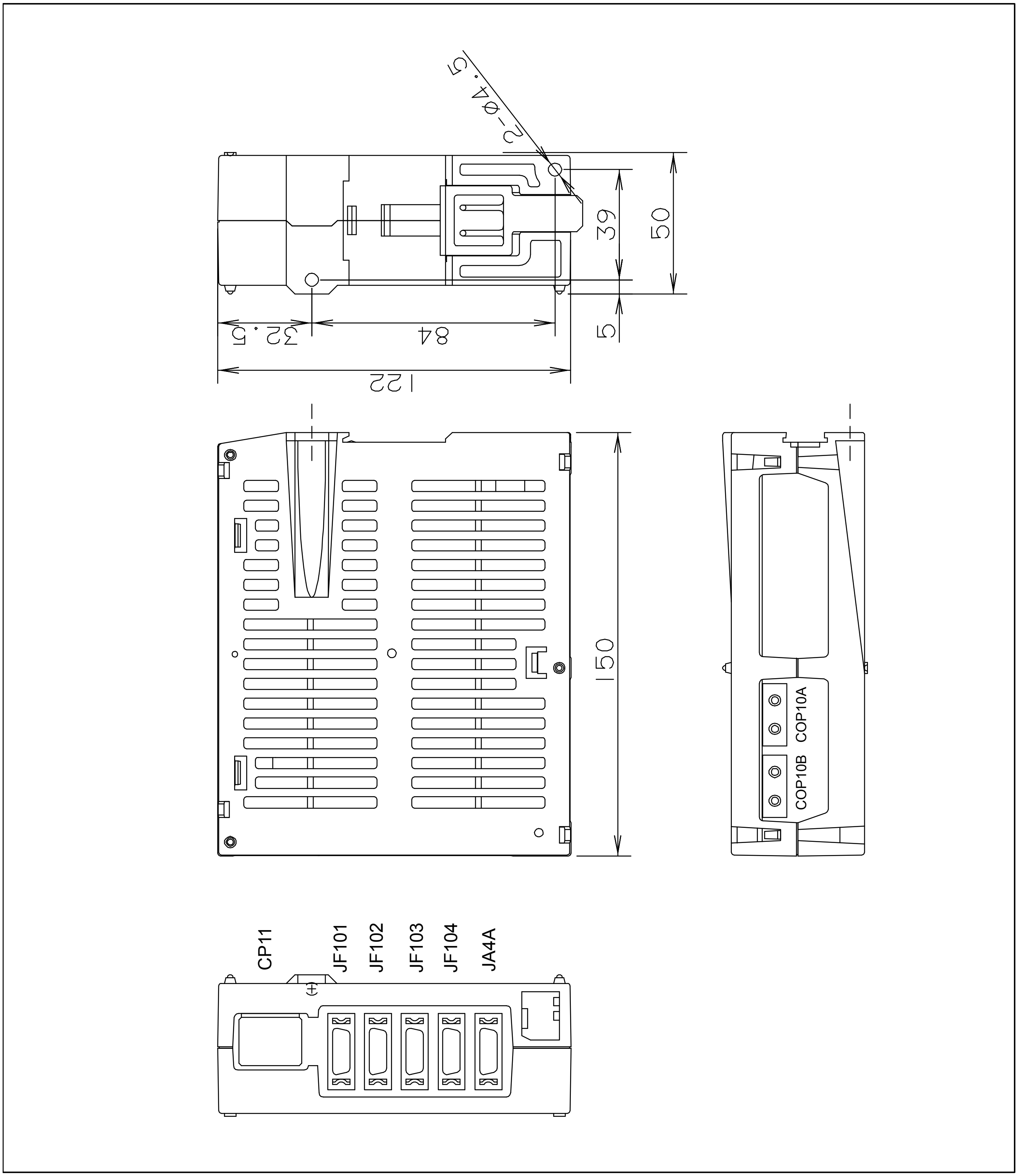


Figure 2–7. Cable Connection between Basic Unit and Expansion Unit

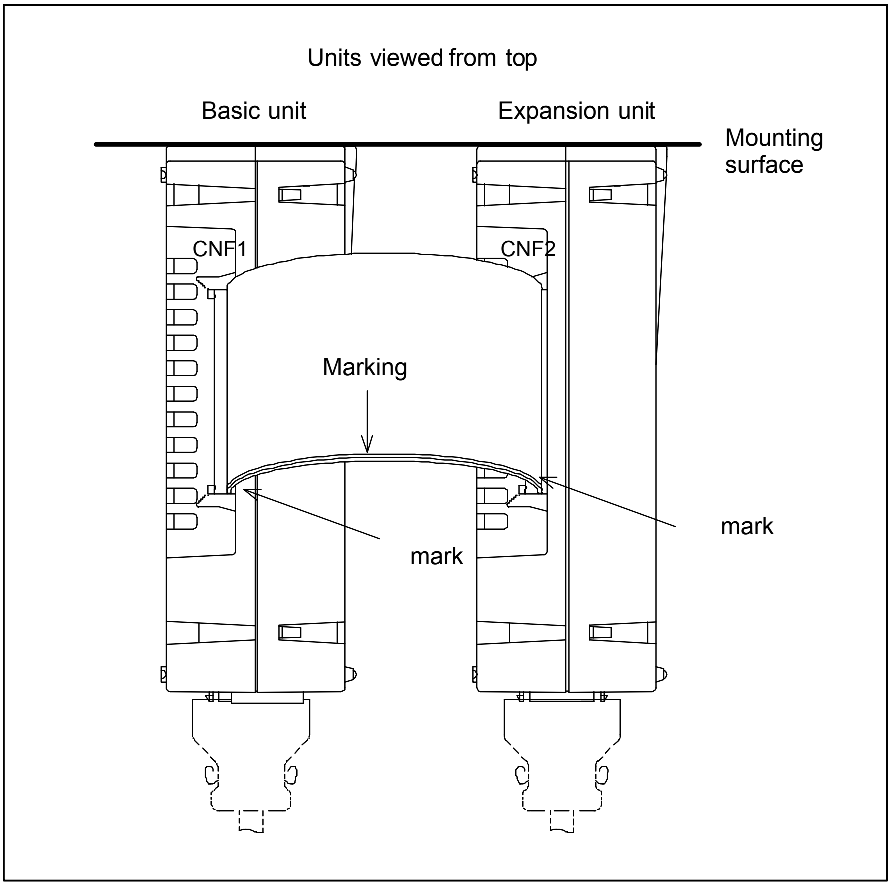


Figure 2–8. Connector Locations on the Basic Unit

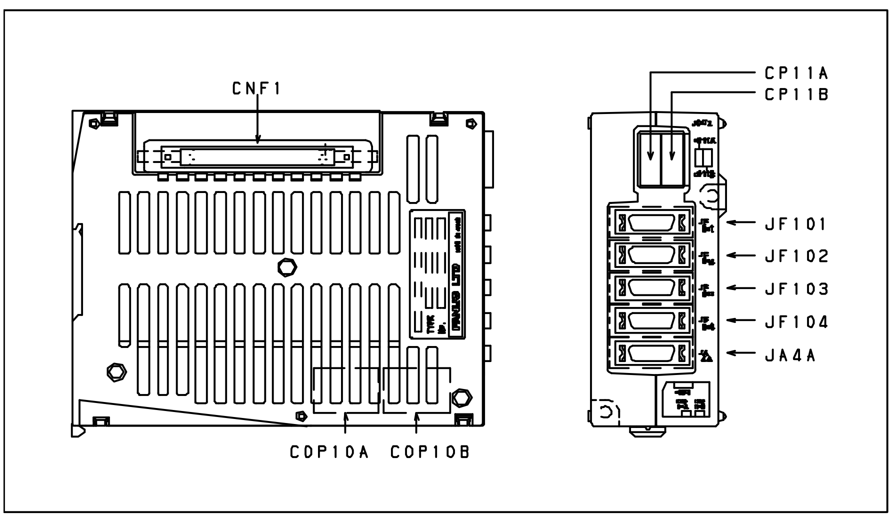


Figure 2–9. Connector Locations on the Expansion Unit

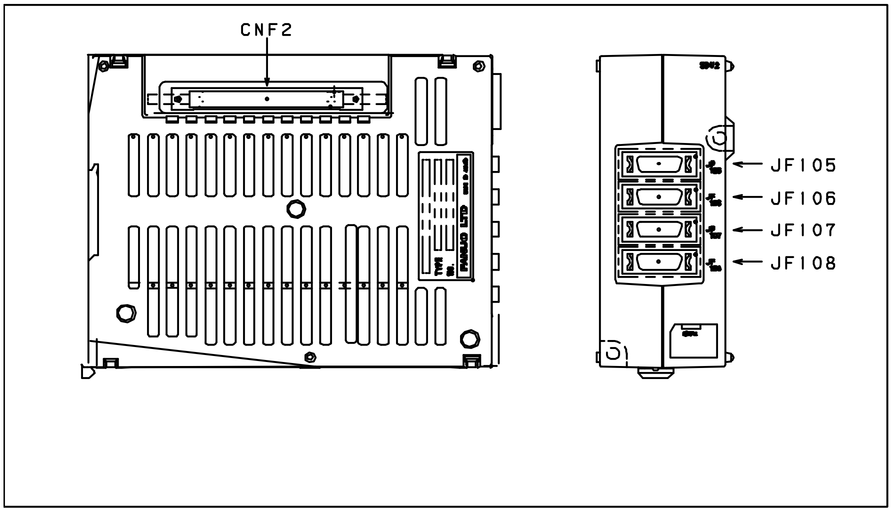


Figure 2–10. Flat Cable Placement During Installation

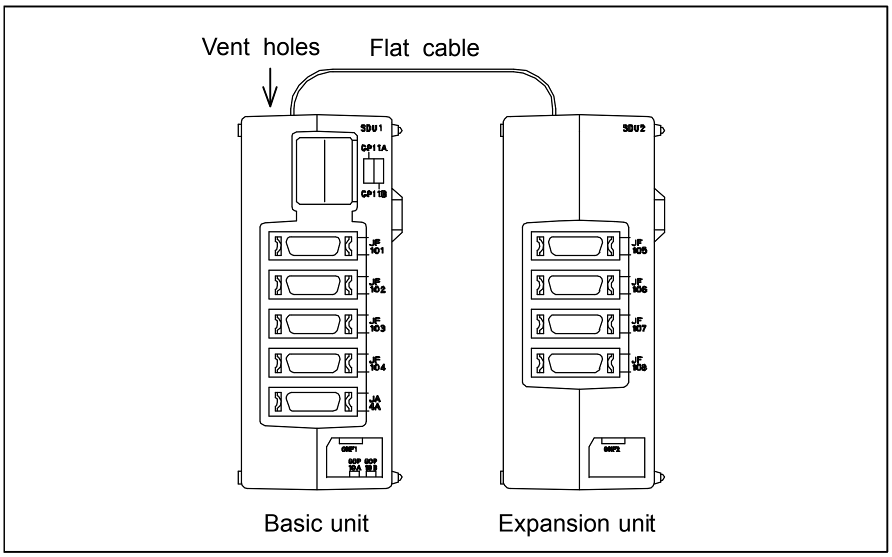
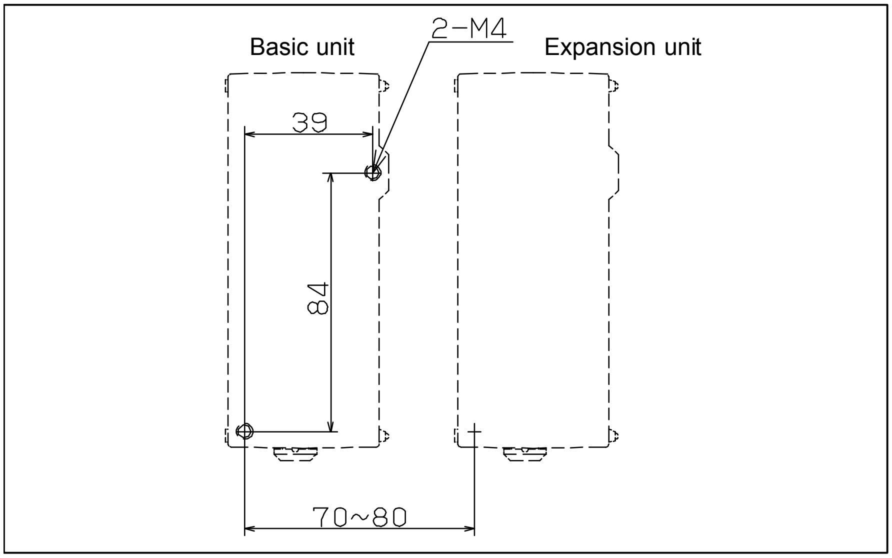


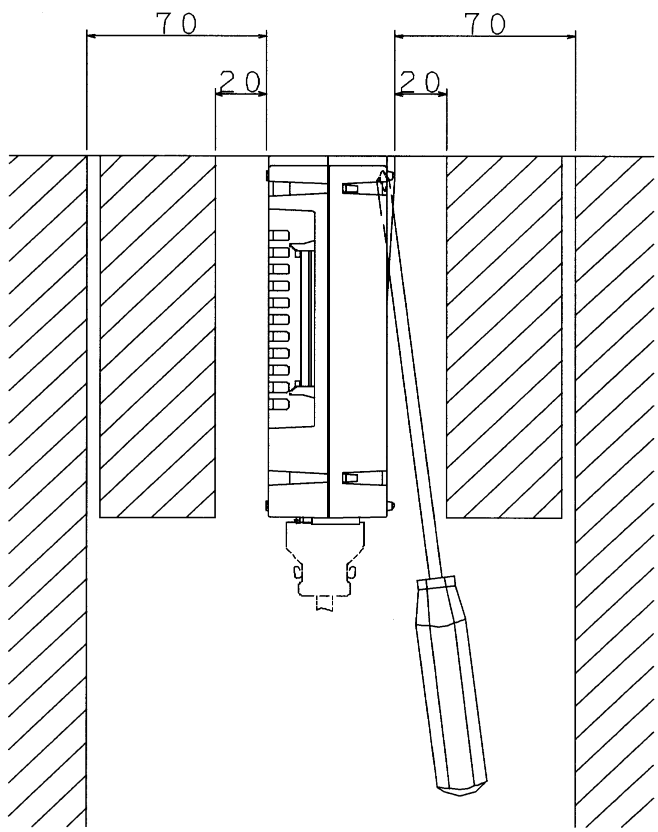
Figure 2–11. Horizontal Separation of Mounting Holes During Installation



Caution

To install or remove the unit, you must insert a screwdriver obliquely. Therefore, you must have sufficient access clearance on both sides of the units. As a general guideline, if the front of an adjacent unit appears flush with the unit or slightly set back, allow a clearance of about 20 mm between the two units. If the front of an adjacent unit protrudes beyond the front of the unit, allow a clearance of about 70 mm between the two units. Also, when you are installing the unit near the side of a cabinet, you must allow a clearance of about 70 mm between the unit and the side of the cabinet.

Figure 2–12. Accessing the Unit



Access clearance near a separate detector interface unit

Caution

When you are removing the unit, be careful not to damage the lock by applying excessive force. When you are installing and removing the unit, hold the upper and lower ends of the unit so that stress is not applied to the side of the unit (the surface with slits).

Figure 2–13. Installing and Removing the Unit

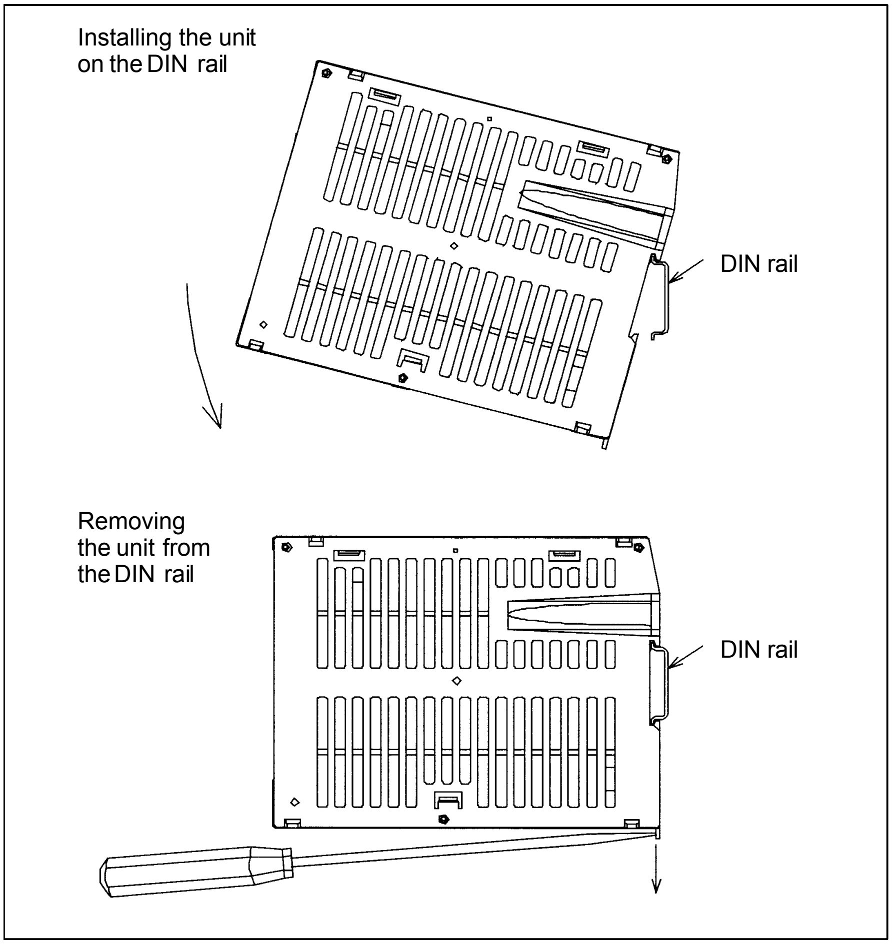
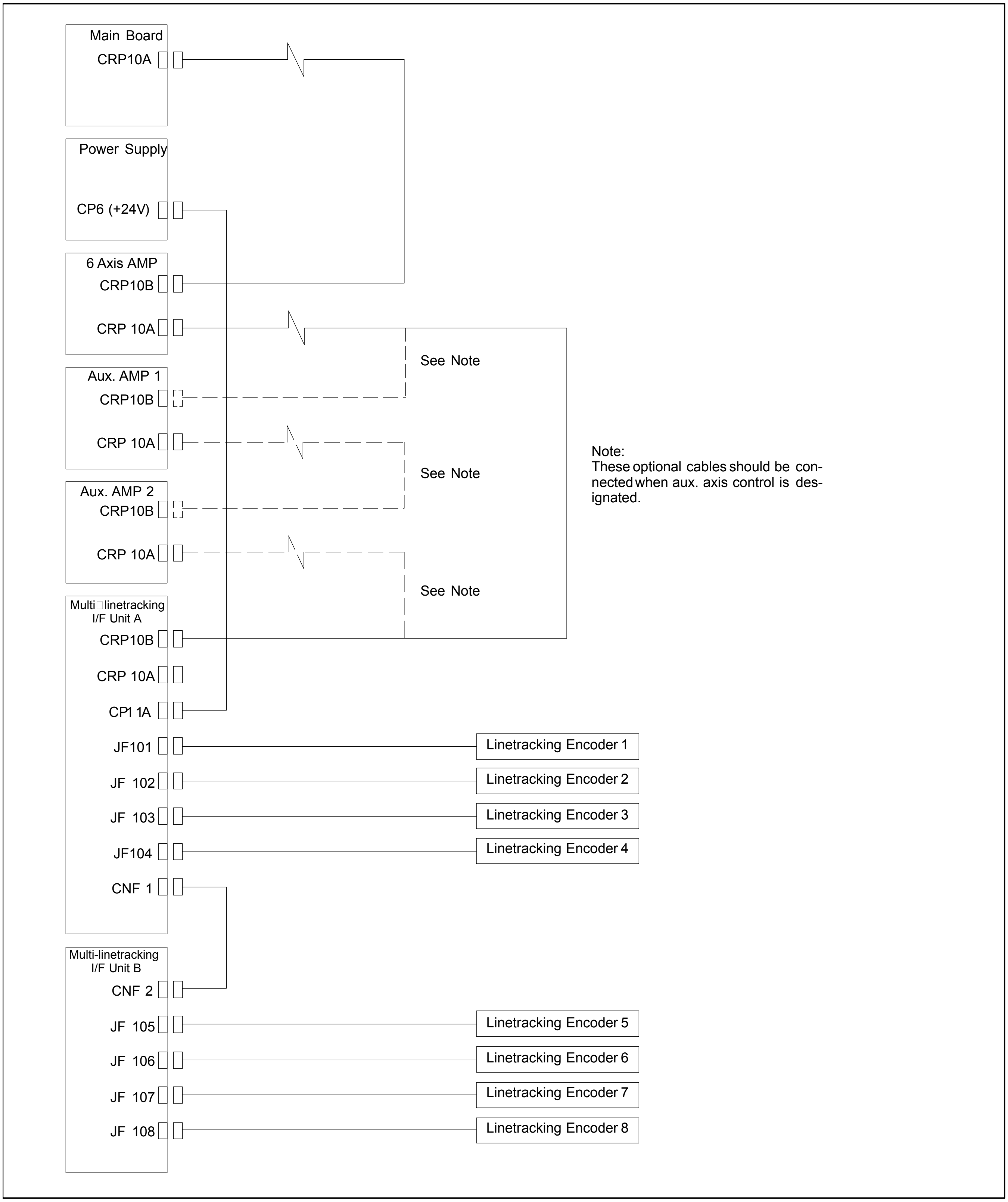


Figure 2–14. Connection Diagram



2.2 INSTALLATION

Line tracking requires both hardware and software installation.

2.2.1 Hardware

A tracking encoder must be installed to monitor the speed of the line or conveyor. A part detect switch must also be installed to detect the approach of a new part.

Line Tracking Interface Board

The line tracking interface board should be inserted into the applicable slot of the power supply unit or main CPU board. See Figure 2–5 . If Separate Detector Interface units are used, they can be mounted in the cabinet separately from the main CPU board. See Figure 2–6 through Figure 2–14.

Fiber Optic FSSB Connector

When a line tracking interface board, A20B-8101-0421 (Wide mini slot) is used The original fiber optic FSSB cable connector that connects to the COP10A connector of the main CPU board should be moved to COP10A of the line tracking interface board. The additional fiber optic FSSB cable should connect COP10A of the main CPU board to COP10B of the line tracking interface board.

When a line tracking interface board, A20B-8101-0601 (mini slot) is used The additional fiber optic FSSB cable should connect COP10A of the 6-axis servo amplifier or aux axis servo amplifier to COP10B of the line tracking interface board. (See Figure 2–5 .)

If Separate Detector Interface units are used, see Figure 2–14 for a connection diagram.

Tracking Encoder

Two kinds of Encoders (Pulsecoders) can be used on R-30*i*B robots (shown below).

- αA1000S Pulsecoder A860-0372-T001 (available as both incremental and absolute)
- Incremental Pulsecoder A860-0301-T001 to T004 Make sure you use appropriate gear or reducer to get desirable resolution (typically 30-80 pulses per mm for Line Tracking).

Part Detect Switch

A part detect switch must be installed, as a digital input, to monitor when a part on the conveyor is approaching the robot workspace. Refer to your application-specific *Setup and Operations Manual* for more information on setting up a digital input.

This switch might be one of numerous types including a contact switch, proximity switch, or optical beam device.

Note You must be aware of the exact location along the conveyor, at which the part will trigger the switch. This location will be used for tracking.

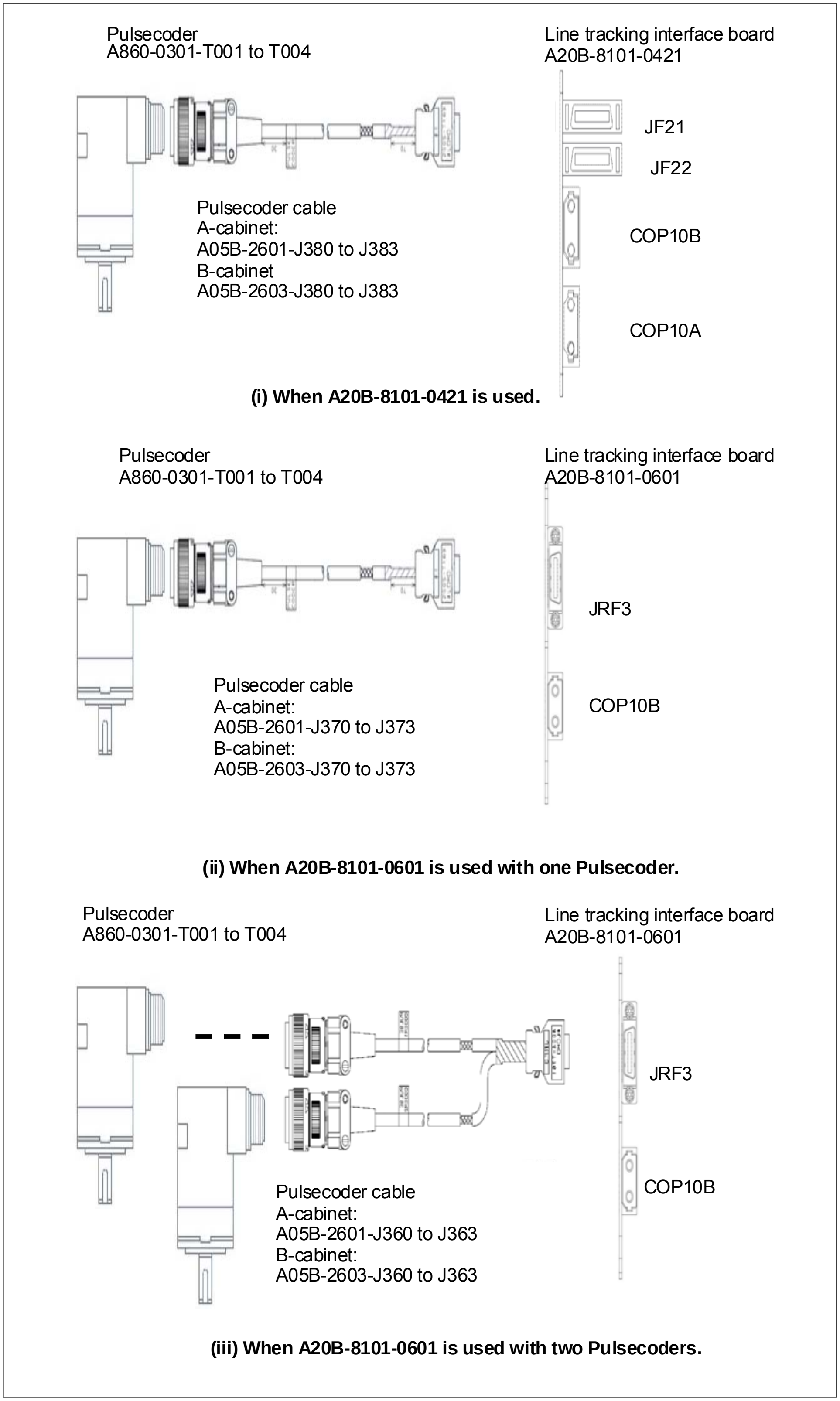
Note Tracking accuracy depends on the precision of the trigger switch. A faster part detect switch gives a more precise trigger value.

Pulse Multiplexer

When using multiple robots, input the value of the encoder to each robot controller via the pulse multiplexer. Connect the line tracking cables and power cable to the pulse multiplexer as shown in Figure 2–15 and Figure 2–16.

When the pulse multiplexer is used, αA1000S Pulsecoder A860-0372-T001 can not be used. If you want to use multiple robots with αA1000S Pulsecoder, use the Ethernet encoder function (option).

Figure 2–15. Connecting cables (Pulsecoder, A860-0301-T001 to T004)



(i) When A20B-8101-0421 is used.

(ii) When A20B-8101-0601 is used with one Pulsecoder.

(iii) When A20B-8101-0601 is used with two Pulsecoders.

Figure 2–16. Pulse Multiplexer and Connecting Cables 1 (Pulsecoder, A860-0301-T001 to T004)

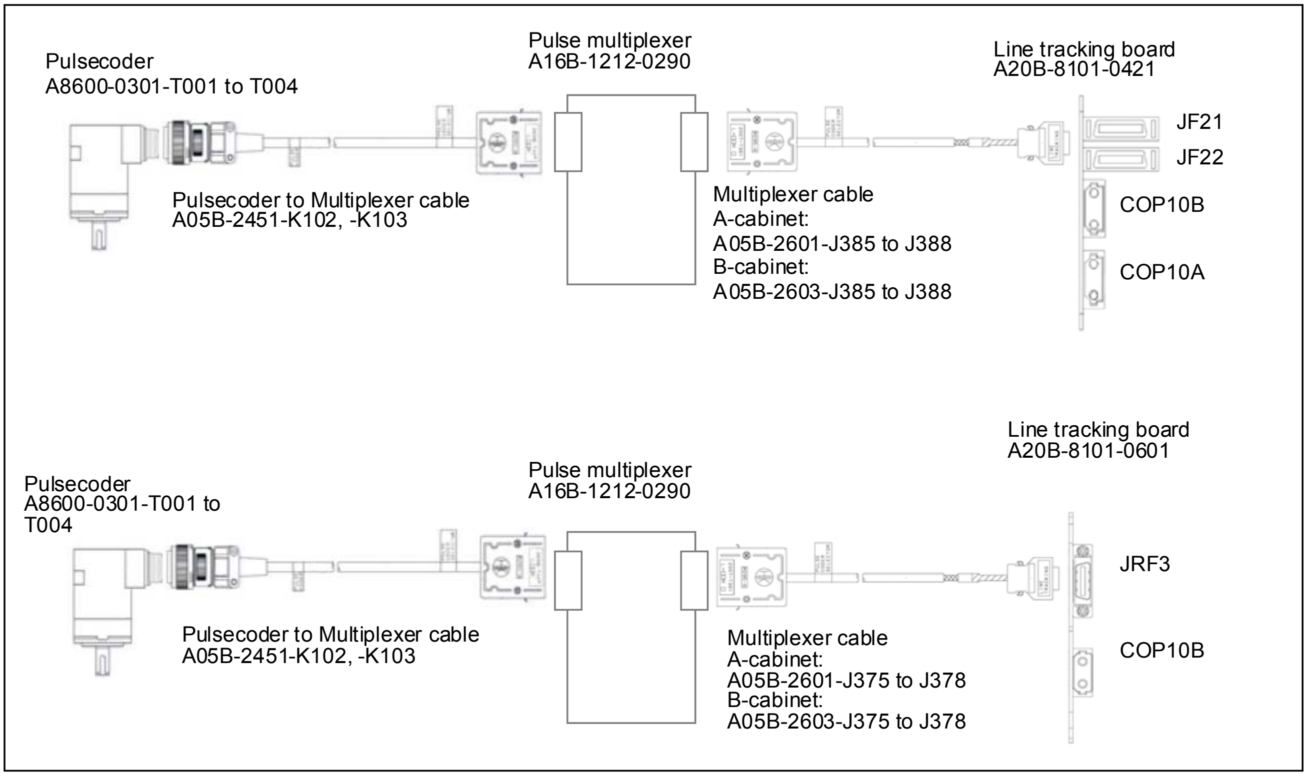
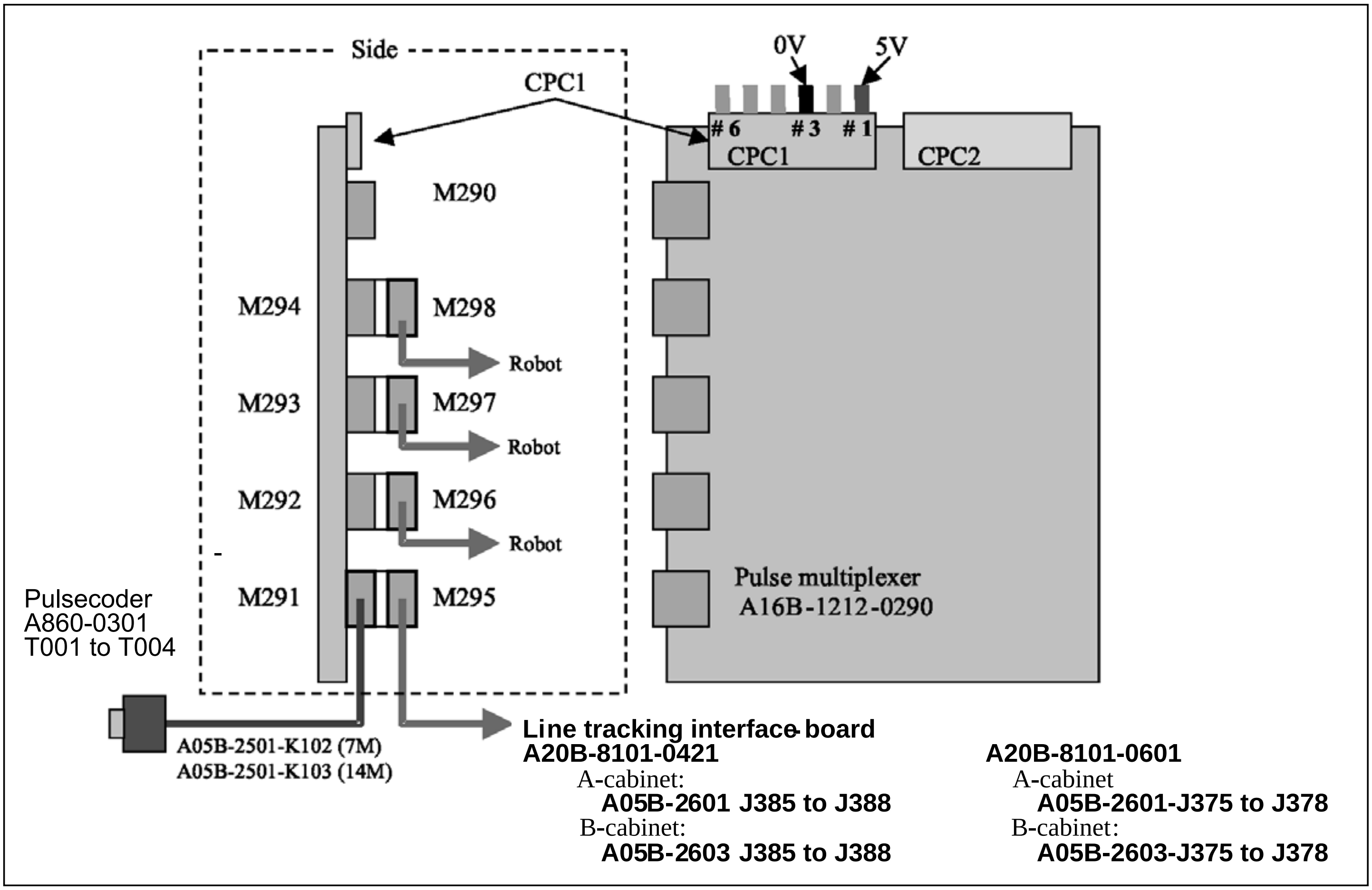


Figure 2–17. Connecting cables to the multiplexer 2 (Pulsecoder, A860-0301-T001 to T004)



Note You can connect up to four cables (pulse multiplexer to/from controller) to one pulse multiplexer.

2.2.2 Software

Line tracking software is distributed as an option. Refer to the *Software Installation Manual* to install a software option.

Line tracking can not be used with the following functions.

- Coordinated Motion
- Space Check
- Remote TCP
- Singularity Avoidance
- Finishing Function Package

- Servo Gun Change function
- Robot Link Function
- Basic/Intelligent Interference Check
- Arc Sensor (TAST)
- AVC
- RPM
- Touch Sensor
- Restart Position check function

Note The Constant Path option must be loaded with the Tracking software option for all applications except painting.

Note The restart position check function cannot be used with the Line Tracking function. Therefore, please set $USERTOL_ENB=FALSE in the system variable screen in order to disable the Restart position check function.

Chapter 3

LINE TRACKING SETUP

Contents

Note If you need to modify any system variable information, refer to Appendix A for descriptions of each line tracking system variable.

3.1 ENCODER SETUP

3.1.1 Encoder Setup Overview

An encoder helps the robot track the job on the conveyor correctly. It provides the robot with a number of pulse counts for every millimeter the conveyor moves. Setting up the encoder establishes the physical relationship between the robot and the conveyor.

Use Procedure 3-1 to set up the encoder parameters. Refer to Section 3.1.2 to verify that you have set up the encoder correctly.

The Encoder Setup Screen parameters are contained in the system variable structure $ENC_STAT. Refer to Table 3–1 for an overview of each encoder setup item and its related system variable.

Note Encoders must be set up before tracking information is set up.

Table 3–1. Encoder Setup Items

ENCODER SETUP SCREEN PARAMETERS	DESCRIPTION	RELATED SYSTEM VARIABLE
Encoder Number Value: 1 - 8 Default:	This item is the schedule selection number of the encoder you are setting up.	N/A
Encoder Axis Value: 0 - 16 Default:	This item allows you to select the encoder axis to set up.	$SCR.$enc_axis[ ]
Encoder Type Value: 0 = incremental 1 = Serial absolute 2 = Servoconv 3 = Virtual 4 = Serial INC 5 = Main Serial INC 6 = Main Serial ABS Default: 0	This item specifies the type of tracking to be used.	$SCR.$enc_type[ ]

Table 3–1. Encoder Setup Items (Cont'd)

ENCODER SETUP SCREEN PARAMETERS	DESCRIPTION	RELATED SYSTEM VARIABLE
Encoder Enable Value: 0 = off 1 = on Default: 0	This item allows you to turn the specified tracking encoder ON or OFF.	$ENC_STAT.$enc_enable
Current Count (cnts) Value: Integer	This item displays the current value for the specified encoder.	$ENC_STAT.$enc_count
Multiplier (ITP/update) Value: 1 - 100 Default:	This item allows you to specify how often the multiplier looks at the conveyor, which can save processor time.	$ENC_STAT.$enc_multipl
Average (updates) Value: 1 - 100 Default:	This item is a value that will help to smooth robot motion when tracking the conveyor.	$ENC_STAT.$enc_average
Stop Threshold(cnt/updt) Value: Positive Integer Default:	This item is the number of encoder counts per encoder update. If the encoder counts per update goes below this number, the system will consider the conveyor stopped.	$ENC_STAT.$enc_thresh
Simulate Enable Value: 0 = off 1 = on Default: 0	This item allows you to turn simulation of the specified tracking encoder ON or OFF.	$ENC_STAT.$enc_sim_on
Simulate Rate(cnt/updt) Value: Integer Default:	This item is the desired number of encoder counts per encoder update. This field is used when encoder simulation is enabled.	$ENC_STAT.$enc_sim_spd
Master RIPE (option) Value: Integer Default: 0	This item is the	$ENC_STAT.$enc_sim_spd
Master Encoder (option) Value: Integer Default: 0	This item is the	$ENC_STAT.$enc_sim_spd

Procedure 3-1 Encoder Setup

Steps

1. Press MENU.
2. Select SETUP.
3. Press F1, [TYPE].
4. Select Encoders. You will see a screen similar to the following.

```
SETUP Encoders
       Encoder Number:  1
1 Encoder Axis:                          1
2 Encoder Type:               INCREMENTAL
3 Encoder Enable:                      OFF
  Current Count (cnts):                  1
4 Multiplier (ITP/update):               1
5 Average (updates):                     1
6 Stop Threshold (cnt/updt):             0
7 Simulate:       Enable:              OFF
8                 Rate (cnt/updt):       0
9  Master Ripe
10 Master Encoder
[TYPE]             ENCODER
```

5. **To display the encoder information for another encoder number,** press F3, ENCODER. This is the schedule selection number of the encoder you are setting up. The default value is 1.

   Note There are two encoders available if you are using a line tracking interface board (part number A20B-8101-0421, A20B-8101-0601). There is one encoder available if you are using the main CPU board (only αA1000S Pulsecoder). There are up to eight encoders available if you are using separate detector interface units, SDU1 (part number A02B-0303-C205) and SDU2 (part number A02B-0303-C204).

6. Select Encoder Axis. Type the channel number of the servo axis board to be used for the tracking encoder. Valid values for this field are 1 through 32.

   For single axis tracking, this field is set to 1. For dual axis tracking, encoder 1 (connected to Line 1 on the CPU) is set to encoder axis = 1. Encoder 2 is set to encoder axis = 2.

   Note You must perform a cold start for this change to take effect. Refer to Section 3.2 after you complete this procedure.

7. Move the cursor to Encoder Type. This specifies the type of tracking encoder that is to be used. There are 6 choices to choose:

   - 1 = INCREMENTAL When Pulsecoder (A860-0301-T001) is used
   - 2 = Serial absolute When αA1000S pulsecoder (A860-0372-T001) is used
   - 3 = Servoconv This will be automatically updated when setup servo conveyor.
   - 4 = Virtual Allow user to simulate encoder without actual encoder.
   - 5 = Serial INC When αA1000S pulsecoder (A860-0372-T001) is used
   - 6 = Main Serial INC When αA1000S pulsecoder (A860-0372-T001) is used

- 7 = Main Serial ABS When αA1000S pulsecoder (A860-0372-T001) is used

Note You must perform a cold start for this change to take effect. Refer to Section 3.2 after you complete this procedure.

8. Move the cursor to Encoder Enable. This allows you to turn the specified tracking encoder ON or OFF.

   - **To turn ON the encoder,** press F4. When turned ON, the encoder will update the count value. The encoder must be turned ON for use with both the actual encoder and under simulation.
   - **To turn OFF the encoder,** press F5.

   Caution

   The Encoder Enable field will automatically reset to OFF after each COLD start. Verify it is set correctly before you run production. Otherwise, your system will not operate correctly.

   Note You can also turn the encoder ON or OFF from within a teach pendant program, by using the LINE instruction. For more information about the LINE instruction, refer to Section 4.5 .

   Current Count (cnts) displays the current value for the specified encoder. You cannot modify this value.

9. Select Multiplier (ITP/update). Enter a value for the encoder update multiplier. This field allows you to specify how often the multiplier looks at the conveyor, which can save processor time. There will be one encoder update for every interpolation (ITP) time increment. For example: *Multiplier×ITP_time(ms) = encoder update(ms)*

   - This field should be set **equal to 1** for most applications. 1×16ms = 16 ms
   - This field can be set to a value **greater than 1** to save processor time for conveyors that maintain fairly constant speeds. 2×16ms = 32 ms

10. Select Average (updates). Enter a value that will help to smooth robot motion when tracking the conveyor.

    If you have a conveyor that does not move smoothly, set this field to a larger value to make robot motion smooth. A typical encoder average value is 10.

11. Select Stop Threshold (cnt/updt). Type the number of encoder counts per encoder update. If the encoder counts per update goes below this number, the system will consider the conveyor stopped.

12. Move the cursor to Simulate Enable. This allows you to turn simulation of the specified tracking encoder ON or OFF (Step 8). The default is OFF. This field is typically used for testing purposes.

Note You **do not** have to plug in a real encoder to simulate. However, if you do not have a real encoder connected, you might get a SRVO-82 error code. This error will not affect the operation of the robot or the simulated line tracking. However, some line tracking instructions (DEFENC, LINESIM, and LINE, for example) might function differently than expected, if you simulate without a real encoder connected. For more information about line tracking instructions, refer to Section 4.5 .

- To simulate the tracking encoder, press F4. When turned ON, the encoder counts will be generated based upon the simulation rate value.
- To use actual encoder counts, press F5. When turned OFF, the encoder counts will be read from the actual encoder when the conveyor is moved.

Note The encoder itself must also be turned ON to allow encoder simulation.

13. Select Simulate Rate (cnt/updt). Type the desired number of encoder counts per encoder update. This field is used when encoder simulation is enabled.

14. If you have the Ethernet Tracking option installed, set up the Master RIPE and the Master Encoder. Refer to Section 5.9 for more information.

15. You must perform a Cold start if you changed Encoder Axis (Step 6) or Encoder Type (Step 7). This must be done before you setup Tracking (Section 3.3).

16. Verify that you have set up the encoder correctly. Refer to Section 3.1.2 . This must be done before you set up Tracking (Section 3.3).

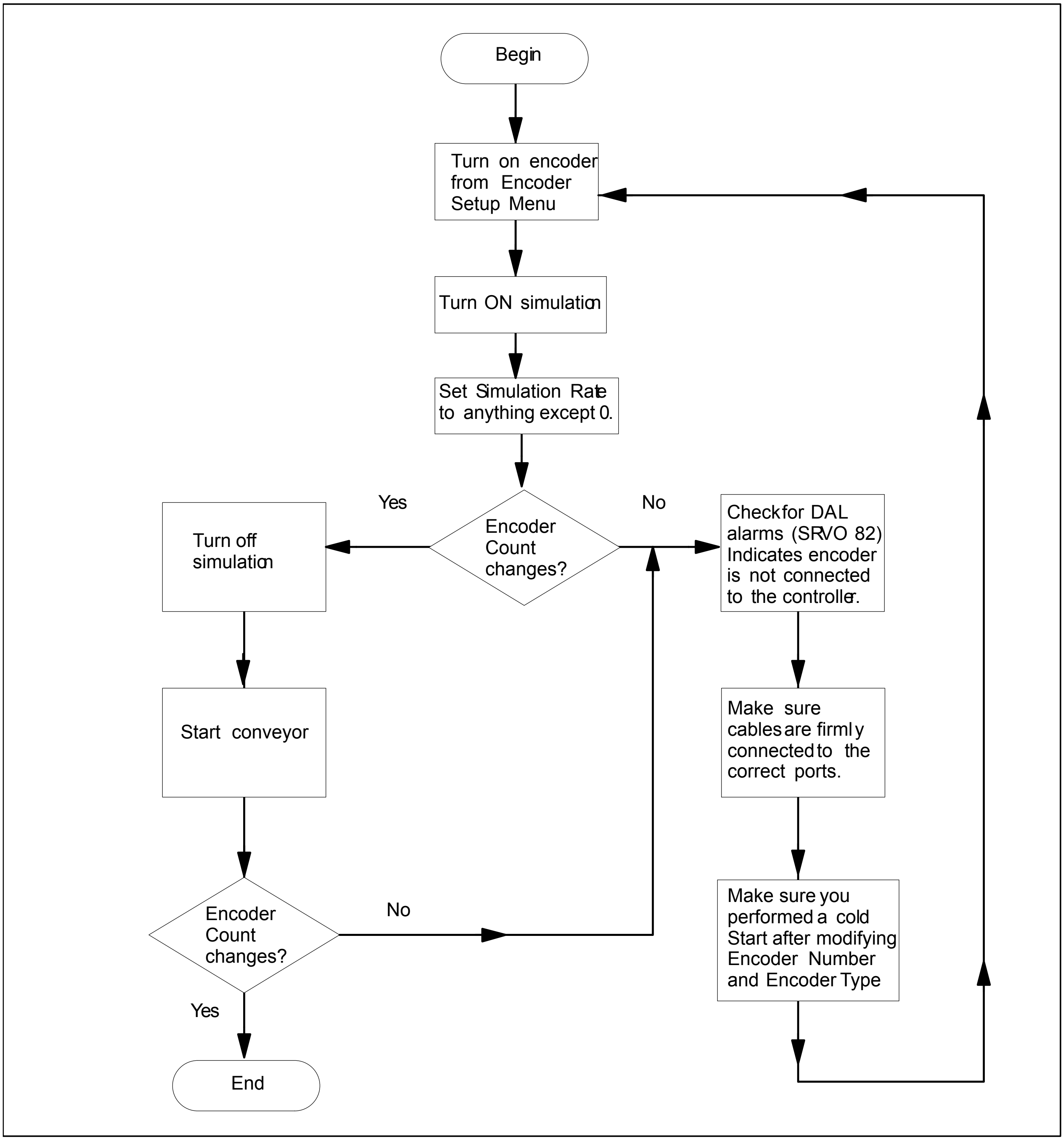
You have completed Encoder Setup.

- **If you have modified Encoder Axis or Encoder Type,** you must perform a COLD start before setting up Tracking. Proceed to Section 3.2 .
- **If you have not modified Encoder Axis or Encoder Type,** you can now proceed to Section 3.3 , Tracking Setup.

3.1.2 Verify Encoder Setup is Correct

Figure 3–1 uses a flowchart to show you how to verify that you have set up the line tracking encoder correctly. For a complete sample test program that you can use to verify line tracking operations, refer to Section 3.4 .

Figure 3–1. Verify Encoder Setup



3.2 COLD START

If you have modified Encoder Axis or Encoder Type on the Encoder Setup Screen (Procedure 3-1), you must perform a Cold start in order for the changes to take effect.

A *Cold start* is the standard method for turning on power to the robot and controller. If your robot is set up to perform a semi hot start, you can force a Cold start using either Procedure 3-2 or Procedure 3-3 . A Cold start does the following:

- Initializes changes to system variables
- Initializes changes to I/O setup
- Displays the UTILITIES Hints screen

A Cold start will be complete in approximately 30 seconds.

Use Procedure 3-2 to perform a Cold start. Use Procedure 3-3 to perform a Cold start from the Configuration Menu.

Procedure 3-2 Performing a Cold Start

Conditions

- All personnel and unnecessary equipment are out of the workcell.

> **Warning**
>
> **DO NOT turn on the robot if you discover any problems or potential hazards. Report them immediately. Turning on a robot that does not pass inspection could result in serious injury.**

- The controller is plugged in and is working properly.
- The teach pendant ON/OFF switch is OFF and the DEADMAN switch is released.
- The REMOTE/LOCAL setup item in the System Configuration Menu is set to LOCAL.

Steps

1. **If the controller is turned on,** turn off the power disconnect circuit breaker.
2. **On the teach pendant** , press and hold the SHIFT and RESET keys.
3. Turn the power disconnect circuit breaker to ON.
4. Release all of the keys. You will see a screen similar to the following.

```
UTILITIES Hints                     ^
             ApplicationTool
    Vx.xxx/xx                XXXX/XX

  Copyright xxxx, All Rights Reserved
            FANUC CORPORATION
       FANUC America Corporation
Licensed Software: Your use constitutes
your acceptance. This product protected
by several U.S. patents.
```

By performing a Cold start, the data you modified in Encoder Axis or Encoder Type has been saved. You can now proceed Section 3.3 , Tracking Setup.

Procedure 3-3 Performing a Cold Start from the Configuration Menu

Steps

1. **If the controller is turned on,** turn off the power disconnect circuit breaker.
2. **On the teach pendant** , press and hold PREV and NEXT.
3. Turn the power disconnect circuit breaker to ON.
4. Release all of the keys. You will see a screen similar to the following.

```
---------- CONFIGURATION MENU ---------
-
1 Hot start
2 Cold start
3 Controlled start
4 Maintenance
Select >
```

5. Select Cold Start and press ENTER. When the Cold start is complete, you will see a screen similar to the following.

```
UTILITIES Hints                          ^
             ApplicationTool
    Vx.xxx/xx                XXXX/XX

  Copyright xxxx, All Rights Reserved
            FANUC CORPORATION
       FANUC America Corporation
Licensed Software: Your use constitutes
your acceptance. This product protected
by several U.S. patents.
```

By performing a Cold start, the data you modified in Encoder Axis or Encoder Type has been saved. You can now proceed Section 3.3 , Tracking Setup.

3.3 TRACKING SETUP

This section describes how to set up tracking parameters for your line tracking application. Tracking setup has been separated into several procedures, to make setup easier for you.

- For general Tracking Setup, use Procedure 3-4
- For Nominal Track Frame Setup
  - Three-point method, use Procedure 3-5
  - Direct entry of frame, use Procedure 3-7
- For Scale Factor Setup, use Procedure 3-8
- Verify that you have set up tracking correctly, Section 3.3.3

Tracking setup allows you to set the parameters, listed on the Tracking Setup Screen, for up to six different schedules or jobs.

The Tracking Setup Screen parameters are contained in the system variable structure $LNSCH. Refer to Table 3–2 for an overview of each tracking setup item and its related system variable. Refer to Appendix A for more detailed information about these and other line tracking system variables.

Table 3–2. Tracking Setup Items

TRACKING SETUP PARAMETERS	DESCRIPTION	RELATED SYSTEM VARIABLES
Schedule Number Value: 1 - 6 Default: 1	This item is the schedule number for a tracking program.	N/A
Robot Tracking GroupValue: 1 Default: 1	This item specifies the robot motion group associated with the current tracking schedule.	$LNSCH.$trk_grp_num
Tracking Type Value: 0 = LINE 1 = RAIL 2 = CIRC Default: 0	This item specifies the type of tracking application.	$LNSCH.$trk_type
Visual Tracking	This item indicates whether the vision system will be used as the trigger mechanism.	Only used when the vision system is loaded
Use Vision Part Queue	This item indicates whether the vision system will be used to set up the part queue.	Only used when the vision system is loaded
Use Tracking Uframe	This item indicates whether the tracking Uframe will be used in the current tracking schedule.	$LNSCH.$use_trk_ufm
Nominal Tracking Frame Value: Position (status) Default: Uninit.	This item allows you to specify the nominal tracking frame used within Cartesian tracking systems.	$LNSCH.$trk_frame
Track (Ext) Axis Num Value: 0 - 3 Default: 0	This item specifies the extended axis which will be used for tracking the conveyor within RAIL tracking systems.	$LNSCH.$trk_axs_num
Track Axis Direction Value: 1 = TRUE (positive direction) 0 = FALSE (negative direction) Default: 1	This item specifies the normal forward motion of the conveyor by comparing it to the motion of the extended axis.	$LNSCH.$trk_axs_dir
Tracking Encoder NumValue: 1 or 2 Default: 1	This item specifies the encoder which will be used for all tracking programs that use the current Tracking Schedule Number.	$LNSCH.$trk_enc_num

Table 3–2. Tracking Setup Items (Cont'd)

TRACKING SETUP PARAMETERS	DESCRIPTION	RELATED SYSTEM VARIABLES
Encoder Scale Factor(cnt/mm) or (cnt/deg) Value: -3.0E38 to 3.0E38 Default: 1.0 Must not = 0.0	**For line and rail tracking,** this item specifies the number of encoder counts per millimeter (counts/mm) of conveyor motion. **For circular tracking,** this item specifies the number of encoder counts per degree (counts/degree) of conveyor motion.	$LNSCH.$scale
Part Detect Dist./Degrees(mm) or (deg) Value: Integer Default: 0	This item allows you to enter the distance (in millimeters for Line and Rail tracking and in degrees for Circular tracking) from the part detect switch to a user-chosen location relative to the robot world frame.	$LNSCH.$teach_dist
Vision Uframe Distance	This item allows you to enter a distance (in millimeters for Line and Rail tracking) from the part detect switch to a location you select where the snap shot of the part is taken.	$LNSCH.$visufm_dist
Trigger INPUT Number Value: 0 - 256 Default: 0	This item allows you to enter a number to specify the digital input (DI[n] where "n" is a number), which is to be used for the part detect switch input signal.	$LNSCH.$trg_din_num
Trigger Value (cnts) Value: Integer Default: 0 (uninit)	This item displays the value of the encoder count at the time of the last part detect (as stored by the teach pendant SETTRIG instruction).	$LNSCH.$trig_value
Encoder Count (cnts) Value: Integer	This item displays the current count value for the specified encoder.	$ENC_STAT.$enc_count

Table 3–2. Tracking Setup Items (Cont'd)

TRACKING SETUP PARAMETERS	DESCRIPTION	RELATED SYSTEM VARIABLES
Selected Boundary Set Value: 1 - 10 Default: 1	This item specifies which of the boundary window sets (pairs of $LNSCH.$BOUND1[n] and $LNSCH.$BOUND2[n]) are used for all position boundary checking, within programs using the current Tracking Schedule Number.	$LNSCH.$sel_bound
Bndry Set n Up Bndry Set n Dn Value: -3.0E38 to 3.0E38 Default: 0.0	This item specifies the up-stream (IN-BOUND) location of a boundary window set.	$LNSCH.$bound1[$LNSCH.$bound2[ ]

Use Procedure 3-4 to set up the tracking parameters.

Note Encoders must be set up before tracking information is set up. Refer to Section 3.1 if you have not set the encoder items.

Procedure 3-4 Tracking Setup

Conditions

- Encoder setup has been performed. Refer to Section 3.1 .

Steps

1. Press MENU.
2. Select SETUP.
3. Press F1, [TYPE].
4. Select 0 –NEXT–.
5. Select Tracking. You will see a screen similar to the following.

```
SETUP Tracking
       Track Schedule Number: 1
1 Robot Tracking Group:                    1
2 Tracking Type:                        Rail
3 Visual Tracking:                        NO
4 Use Vision Part Queue:                  NO
5 Use Tracking Uframe:                    NO
6 Nominal Track Frame:  Stat:          WORLD
7 Track (Ext) Axis Num:                    2
8 Track Axis Direction:             POSITIVE
```

6. **To display the tracking information for another track schedule number,** press F3, SCHED. This specifies which one of the six schedules is displayed. You can choose any one of six tracking schedule numbers for a tracking program, by specifying the desired schedule number in the program header data.

   Note Be sure to select the correct schedule number for the tracking program so that the correct variables are set during production.

7. **To select one or more motion groups** ,

   a. If you are setting more than one motion group, set the system variable $LNCFG.$group_mask to a value greater than one (1). Refer to the *Software Reference Manual* for detailed information on system variables.

   b. Select Robot Tracking Group. Type a number that specifies the robot motion group associated with the current tracking schedule.

8. Move the cursor to Tracking Type. This specifies the type of tracking application.

9. Press F4, [CHOICE].

10. Select the type of tracking for your application.

    - 1 = Line Tracking. This corresponds to the value of the system variable $LNSCH[].$TRK_TYPE = 0.
    - 2 = Rail Tracking. This corresponds to a value of the system variable $LNSCH[].$TRK_TYPE = 1. **If you are using Rail Tracking, go to Step 14 .**
    - 3 = Circular Tracking. This corresponds to a value of the system variable $LNSCH[].$TRK_TYPE = 2.

    Note Changing the tracking type changes the values of the Nominal Tracking Frame, Track Axis Number, and Track Axis Direction. The previous values will be stored until either another schedule number is selected, or this SETUP menu is exited. If the Tracking Type is returned to its previous value, before you select another schedule number or exit the SETUP menu, the previous values will be restored.

11. **If you are using Tracking User frame,** set Use Tracking Uframe to YES. Otherwise, set it to NO.

12. **If you are using Line or Circular tracking** , move the cursor to Nominal Track Frame.

> **Caution**
>
> Do not set the nominal tracking frame for any schedule that specifies RAIL tracking. The nominal tracking frame is automatically set to the (0,0,0,0,0,0) WORLD frame for RAIL tracking systems.

13. Press F2, DETAIL.

- **If you are using Line tracking** , you will see a screen similar to the following.

```
SETUP Frames
 Track Frame Setup (Line)
 Track Frame of Schedule:      1
  X:    0.00  Y:    0.00  Z:    0.00
  W:    0.00  P:    0.00  R:    0.00
 Teach Data:
  Origin:UNINIT      Enc_cnt           0
  X:    0.00  Y:    0.00  Z:    0.00
  +X dir:UNINIT      Enc_cnt           0
  X:    0.00  Y:    0.00  Z:    0.00
  +Y dir:UNINIT
  X:    0.00  Y:    0.00  Z:    0.00
 Scale (cnt/mm):     500.00

         TEACH  COMPUTE SCALE
```

- **If you are using Circular tracking** , you will see a screen similar to the following.

```
SETUP Frames
 Track Frame Setup (Circ)
 Track Frame of Schedule:      1
  X:    0.00  Y:    0.00  Z:    0.00
  W:    0.00  P:    0.00  R:    0.00
 Teach Data:
  +X dir:UNINIT      Enc_cnt           0
  X:    0.00  Y:    0.00  Z:    0.00
  +Y dir:UNINIT      Enc_cnt           0
  X:    0.00  Y:    0.00  Z:    0.00
  Assist:UNINIT
  X:    0.00  Y:    0.00  Z:    0.00
 Scale (cnt/mm):     500.00
```

```
TEACH   COMPUTE SCALE
```

The Track Frame SETUP menu provides a means for you to specify the nominal tracking frame used within Cartesian tracking systems. You can either enter a value for the nominal tracking frame directly, or teach the frame using the three-point method

- Use Procedure 3-5 if you are using the *three-point method* to set the nominal tracking frame. This is the method of choice.
- Use Procedure 3-7 if you are using the *direct entry method* to set the nominal tracking frame.

Note Refer to Section 3.3.1 for more detailed information about setting the Nominal Tracking Frame.

14. **If you are using Rail tracking,** select Track (Ext) Axis Num. Enter a number that specifies the extended axis which will be used for tracking the conveyor within RAIL tracking systems. This number will automatically be set to 0 for Line and Circular tracking systems. Valid values are 1-3.
15. **If you are using Rail tracking,** move the cursor to Track Axis Direction. This specifies the normal forward motion of the conveyor, by comparing it to the motion of the extended axis.

    - If motion is the same as the extended axis, press F4, POSITIVE.
    - If motion is opposite the extended axis, press F5, NEGATIVE.

    Note The extended axis is used for tracking the conveyor within RAIL tracking systems. The Track Axis Direction is automatically set to POSITIVE for Line and Circular tracking systems.

16. **When you have finished setting the Nominal Track Frame (** Section 3.3.1 **),** select Tracking Encoder Num. Enter a number that specifies the encoder which will be used for all tracking programs that use the current Tracking Schedule Number. Valid choices are encoder 1 or 2.
17. Move the cursor to Encoder Scale Factor.

    - **For Line and Rail tracking** , this specifies the number of encoder counts per millimeter (counts/mm) of conveyor motion.
    - **For Circular tracking** , this specifies the number of encoder counts per degree (counts/degree) of conveyor motion.

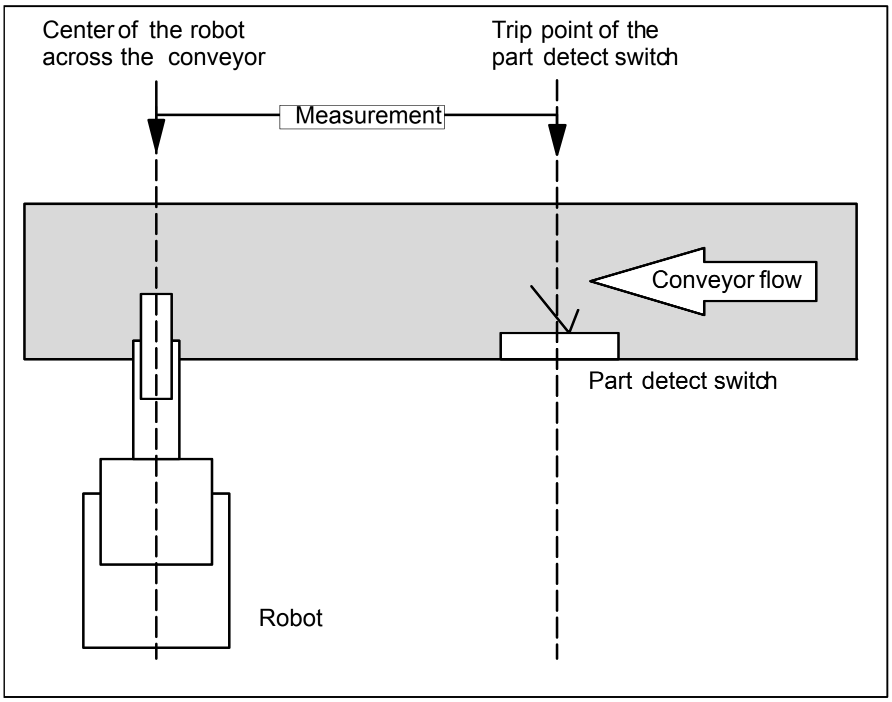
    This number can be any real number **except (0.0)** .

18. Press F2, TEACH. You will be taken to the Scale Factor Setup screen. **Refer to Section 3.3.2 for detailed information about teaching the Scale Factor.**
19. **After you have taught the Scale Factor (** Procedure 3-8 **)** , select Part Detect Dist. Enter the distance (in millimeters for Line and Rail tracking and in degrees for Circular tracking) from the part detect switch to a user-chosen location relative to the robot world frame. This is usually the world X-axis, which is perpendicular to the tracking conveyor when the robot is at its home position.

This number creates a reference between the nominal tracking frame and the part detect switch. The program paths can then be copied from one robot to another, as long as the individual part detect distances are correctly specified for each robot. This compensates for varying part detect switch positions within a multi-robot application.

Note This parameter relies on a correct value for the Encoder Scale Factor Step 17). Refer to Figure 3–2 .

Figure 3–2. Part Detect Switch (Line and Rail Tracking)



20. If you are using Tracking User frame instruction, VISUFRAME, set Vision Uframe Dist. to a proper value.

21. Move the cursor to Trigger INPUT Number. Type a number to specify the digital input (DI[n] where "n" is a number), which is to be used for the part detect switch input signal. This input is monitored during conveyor synchronization for programs which specify the current Tracking Schedule Number. Valid values range from 0-256.

Trigger Value displays the value of the encoder count at the time of the last part detect (as stored by the teach pendant SETTRIG instruction). You cannot modify this value.

Encoder Count displays the current count value for the specified encoder. You cannot modify this value.

22. Select Selected Boundary Set. Enter a number to specify which of the boundary window sets (pairs of $LNSCH.$BOUND1[n] and $LNSCH.$BOUND2[n]) are used for all position boundary checking, within programs using the current Tracking Schedule Number. Refer to Figure 3–3 .

This number is used as an index into each of the two arrays. The index values are used in line tracking programs, to determine when the robot should begin and end work on a part.

Note The teach pendant SETBOUND instruction can be used to change this value from within a teach pendant program. Refer to Section 4.5 for more information.

For Circular tracking the Selected Boundary Set fields will be used only when the circular tracking boundary function is enabled. To enable circular tracking boundary function set $LNCFG.$COMP_SW2 bit 0x10 on (OR the system variable with 0x10)

23. Move the cursor to Bndry Set n Up. This specifies the *up-stream* (IN-BOUND) location of a boundary window set, where the number (n) is a number from 1 through 10 used to index which boundary is being set or selected. Refer to Figure 3–3 .

    - Conveyor positions further up-stream of this position are considered IN-BOUND. The robot cannot work on the part.
    - Conveyor positions further down-stream of this position are either IN-WINDOW or GONE.

    The Selected Boundary Set number (entered in Step 22) is an index into this array. This value must be further *up-stream* than the value of the corresponding down-stream boundary, otherwise a warning message will be displayed.

    - **To record the current position of the robot TCP** (relative to the nominal tracking frame), press SHIFT and F2, RECORD simultaneously. The appropriate boundary value will be extracted and stored as the selected boundary.

    OR

    - **To initialize the currently selected boundary value to 0.0** (regardless of the value of the corresponding *down-stream* boundary value), press F4, INIT-BND. Refer to Figure 3–3 .

24. Move the cursor to Bndry Set n Dwn. This specifies the *down-stream* (OUT-BOUND) location of a boundary window set, where the location is a position along the direction of the conveyor relative to the nominal tracking frame. Refer to Figure 3–3 .

    - Conveyor positions further up-stream of this position are considered either IN-WINDOW or IN-BOUND.
    - Conveyor positions further down-stream of this position are considered GONE. The robot cannot work on the part.

    The Selected Boundary Set number is an index into this array. This value must be further *down-stream* than the value of the corresponding up-stream boundary, otherwise a warning message will appear.

    - **To record the current position of the robot TCP** (relative to the nominal tracking frame), press SHIFT and F2, RECORD simultaneously. The appropriate boundary value will be extracted and stored as the selected boundary.

OR

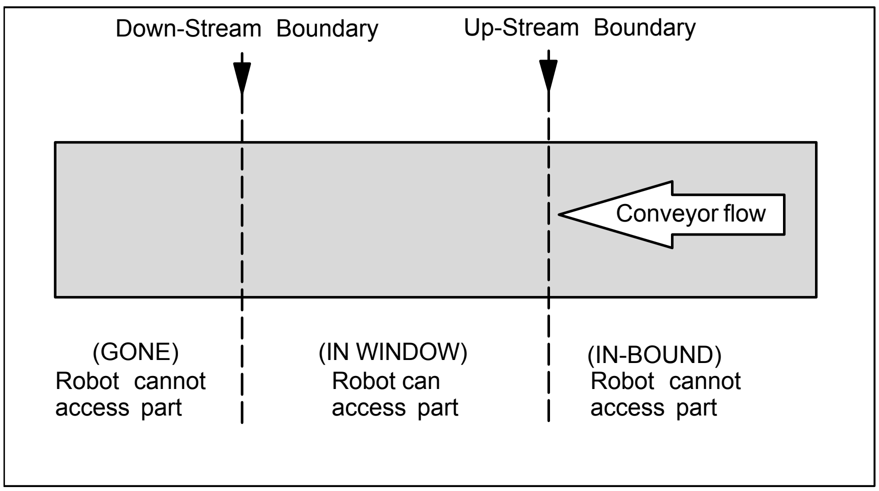
- **To initialize the currently selected boundary value to 0.0** (regardless of the value of the corresponding *up-stream* boundary value), press F4, INIT-BND. Refer to Figure 3–3 .

25. Verify that you have set up tracking correctly. Refer to Section 3.3.3 . This should be done before you run production.

During Production

During production, the system will wait until the part travels past the *up-stream* boundary before the robot will start processing the part. If the part travels past the *down-stream* boundary, the part cannot be processed and an error will be displayed.

Figure 3–3. Boundary Window



The following lists some example values for a line tracking system. All Downbounds values are greater than their corresponding Upbounds values: **Upbounds #1 -**1500 mm **Downbounds #1 -** 100 mm **Upbounds #2** - 500 mm **Downbounds #2** 500 mm **Upbounds #3** - 100 mm **Downbounds #3** 1200 mm

3.3.1 Nominal Tracking Frame Setup

The *nominal tracking frame* is used in a tracking application to provide a coordinate reference frame for all positions and motions referenced with respect to the conveyor.

For line tracking applications :

- Use Procedure 3-5 if you are using the *three-point method* to set the nominal tracking frame for line tracking. This is the method of choice.

- Use Procedure 3-7 if you are using the *direct entry method* to set the nominal tracking frame for line tracking. This method is used when copying from another schedule.

For circular tracking applications :

- Use Procedure 3-6 if you are using the *three-point method* to set the nominal tracking frame for circular tracking. This is the method of choice.
- Use Procedure 3-7 if you are using the *direct entry method* to set the nominal tracking frame for circular tracking. This method is used when copying from another schedule.

For rail tracking applications the system automatically sets this value to be the WORLD (0,0,0,0,0,0) frame.

Caution

Do not set any USER frame (UFRAME) values for tracking programs. Setting a UFRAME could cause unexpected motion during tracking. If you try to set a UFRAME, you will receive an error message when you try to record a tracking position. The Tracking frame is used (instead of the UFRAME) for all tracking motions.

Three Point Method

The three-point method is used to teach the nominal tracking frame. During teaching, you move the cursor to each of the three data positions listed under Teach Method Data. A status value is displayed for each of these positions, and will be one of two values:

- UNINIT - indicates that the position is un-initialized
- RECORDED - indicates that the position has been recorded but not yet used during processing

When any of these positions is selected, the word TEACH appears above the F2 function key. Pressing SHIFT and TEACH simultaneously will record the current robot TCP position (to be used during later processing) and will update the position status to RECORDED.

Warning

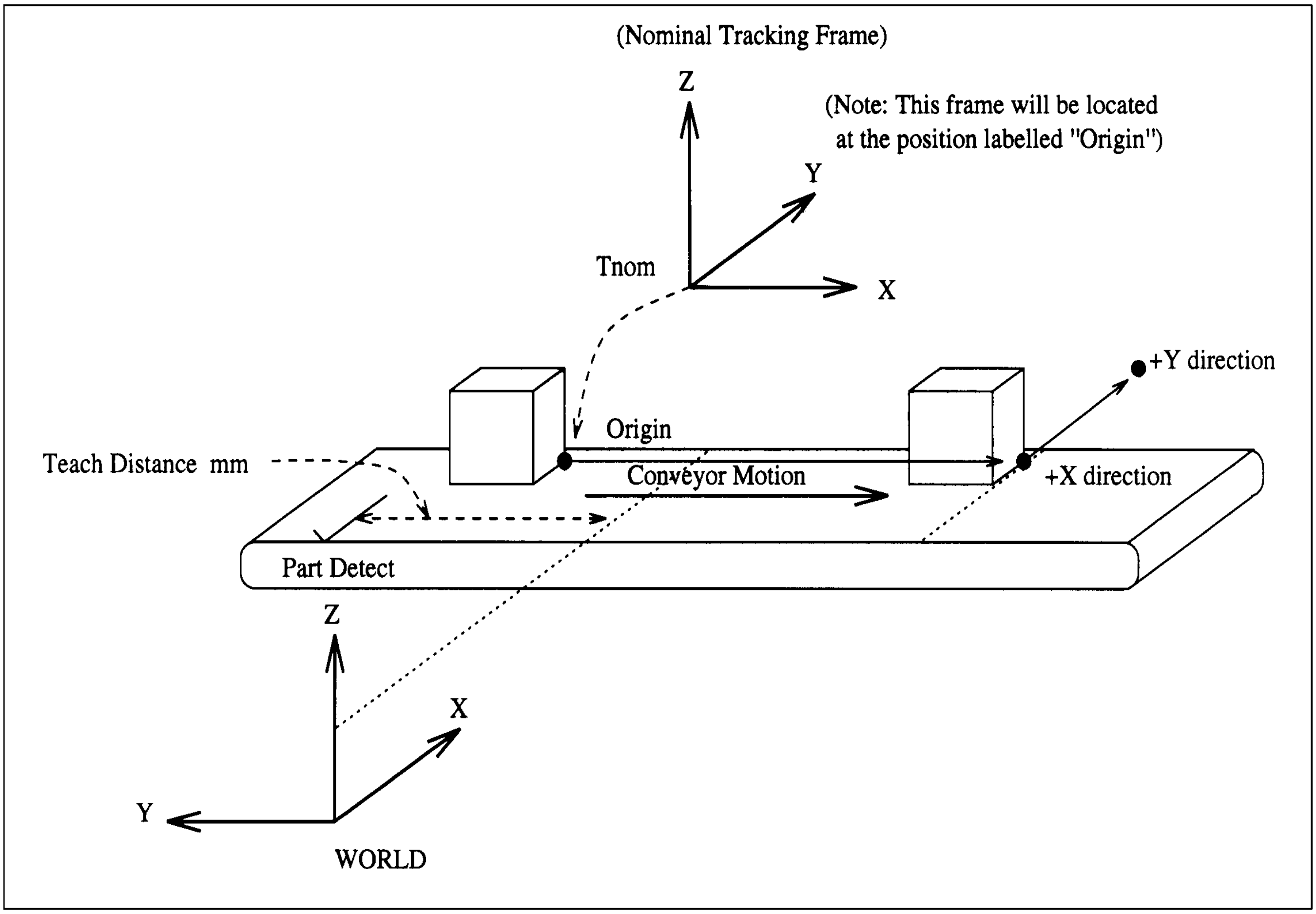
Be sure the robot UTOOL is properly defined before performing this procedure. Otherwise, you could injure personnel or damage equipment. Refer to your application-specific *Setup and Operations Manual* for more information.

For Line Tracking Applications

When setting the nominal tracking frame for a line tracking application you must be aware of the following:

- **The x-axis of this frame must point in the direction of conveyor FORWARD motion.** Use Procedure 3-5 to set this, and all other axes of the nominal tracking frame. The y and z-axes are user-definable, but are typically set so that the z-axis points upward from the surface of the conveyor. Refer to Figure 3–4 .
- The origin location of the nominal tracking frame is arbitrary. You might prefer to set this to the World origin (0,0,0). However, the orientation is very important and should be left as taught using Procedure 3-5 . **After you have set this value and recorded either boundary or motion positions, do not change this value.**
- All boundary locations are recorded relative to this frame.
  - — line tracking boundary values are locations in millimeters along the x-axis of the nominal tracking frame.
  - — rail tracking boundary values are tracking (extended) axis locations relative to the World frame.

Figure 3–4. Nominal Track Frame - Line Conveyor Motion



Procedure 3-5 Three Point Method to Teach the Nominal Tracking Frame for Line Tracking

Conditions

- You have selected Nominal Track Frame from the Tracking Setup Screen. Refer to Procedure 3-4 , Step 12 .
- You are currently at the Track Frame Screen.

Steps

1. Move the cursor to the ORIGIN Teach Method Data position. See the following screen for an example.

```
SETUP Frames
 Track Frame Setup (Line)          1/5
 Track Frame of Schedule:      2
  X:     0.00  Y:     0.00  Z:     0.00
  W:     0.00  P:     0.00  R:     0.00
 Teach Data:
  Origin:RECORDED    Enc_cnt          2356
  X: 2241.80  Y:  754.09  Z:   30.00
  +X dir:RECORDED    Enc_cnt         20356
  X: 2241.80  Y:  954.09  Z:   30.00
  +Y dir:UNINIT
  X:     0.00  Y:     0.00  Z:     0.00
 Scale (cnt/mm):      500.00
          TEACH   COMPUTE  SCALE
```

2. Move the robot TCP to a convenient position along the conveyor. (This position should be an easily distinguishable location either on the conveyor or on a part riding on the conveyor.)
3. Record this position by pressing SHIFT and TEACH simultaneously. The status of the ORIGIN position should change to RECORDED. The screen will also update X, Y, Z and Enc_cnt data with current robot TCP position and encoder count value
4. Move the cursor to select the +x Direction Teach Method Data position.
5. Move the robot away from the part so that the conveyor (and the part) can be moved without running into the robot.
6. Move the conveyor FORWARD (in the direction of normal part flow) for a distance of at least several hundred millimeters (the farther the better, as long as the robot will still be able to reach the new location of the part.)
7. Stop the conveyor.
8. Move the robot to the same location relative to the conveyor (or part) that was used for the ORIGIN position.

9. Record this position by pressing SHIFT and TEACH. (The status of the +x Direction position should change to RECORDED.) The screen will also update X, Y, Z and Enc_cnt data with current robot TCP position and encoder count value

10. Move the cursor to select the +y Direction Teach Method Data position.

11. **Without moving the conveyor** (or the part), move the robot at least 50mm in the direction perpendicular to the conveyor.

    Typically this is toward the left side of the conveyor, when viewing along the direction of forward conveyor flow such that the resulting z-axis of the nominal tracking frame will point upward from the conveyor.

12. Record this position by pressing SHIFT and TEACH simultaneously. (The status of the +y Direction position should change to RECORDED.)

13. **To process all of the data positions and compute a new nominal tracking frame** , press F3, COMPUTE. When the processing is complete, the status of the three Teach method Data positions will be set to PROCESSED, and the Frame Components data values will be updated to display the new nominal tracking frame. See the following screen for an example.

```
SETUP Frames
 Track Frame Setup (Line)          1/5
 Track Frame of Schedule:      2
  X: 2241.80  Y:  754.09  Z:    30.00
  W:    0.00  P:    0.00  R:    90.00
 Teach Data:
  Origin:RECORDED    Enc_cnt          2356
  X: 2241.80  Y:  754.09  Z:   30.00
  +X dir:RECORDED    Enc_cnt         20356
  X: 2241.80  Y:  954.09  Z:   30.00
  +Y dir:RECORDED
  X: 2341.80  Y:  954.09  Z:   30.00
 Scale (cnt/mm):     500.00

          TEACH   COMPUTE  SCALE
```

14. You can setup the encoder scale for this line tracking schedule here or at scale item in the Tracking Setup main menu as in R30iA platform. If you want to setup the encoder scale at this time follow the next steps

15. Move the cursor to Origin or +X dir. The SCALE function will be display. Calculate the encoder scale by pressing SHIFT and SCALE. The scale value will be updated.

You have completed setup of the nominal tracking frame using the three point method. You can now go back to Tracking Setup at Procedure 3-4 , Step 16 .

For Circular Tracking Applications

When setting the Nominal Tracking Frame for a Circular Tracking application, you must be aware of the following.

- The three points are used to compute the CENTER of the circular conveyor, which is then used as the origin of the Nominal Tracking Frame for Circular tracking.
- The +y position relative to the + x position, must point in the direction of *forward* conveyor motion. This establishes the orientation of the Nominal Tracking Frame.
- The Assistant position of the nominal tracking frame is arbitrary, but should be located as shown in Figure 3–5 or Figure 3–6 .
- For counter clockwise conveyor motion, the z-axis of the Nominal Tracking frame must point *up* . Refer to Figure 3–5 .
- For clockwise conveyor motion, the z-axis must point *down* . Refer to Figure 3–6 .
- The x-axis of the Nominal Tracking Frame always points to the +x position used to teach the frame.
- All tracking positions are automatically recorded relative to this frame.
- The circular tracking boundary value should not be used when the boundary function for circular tracking is not enabled. Refer to Section 1.4.2 .

Figure 3–5. Nominal Tracking Frame - Counter Clockwise Circular Tracking

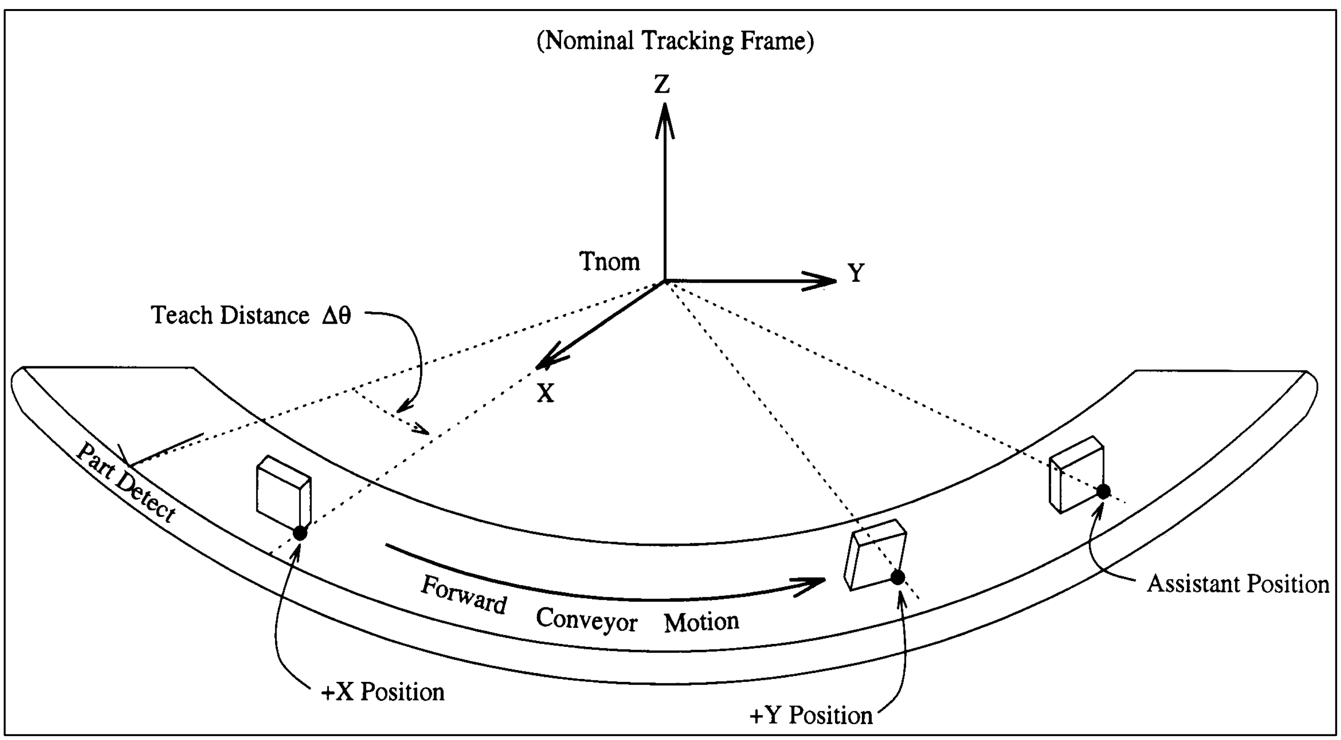
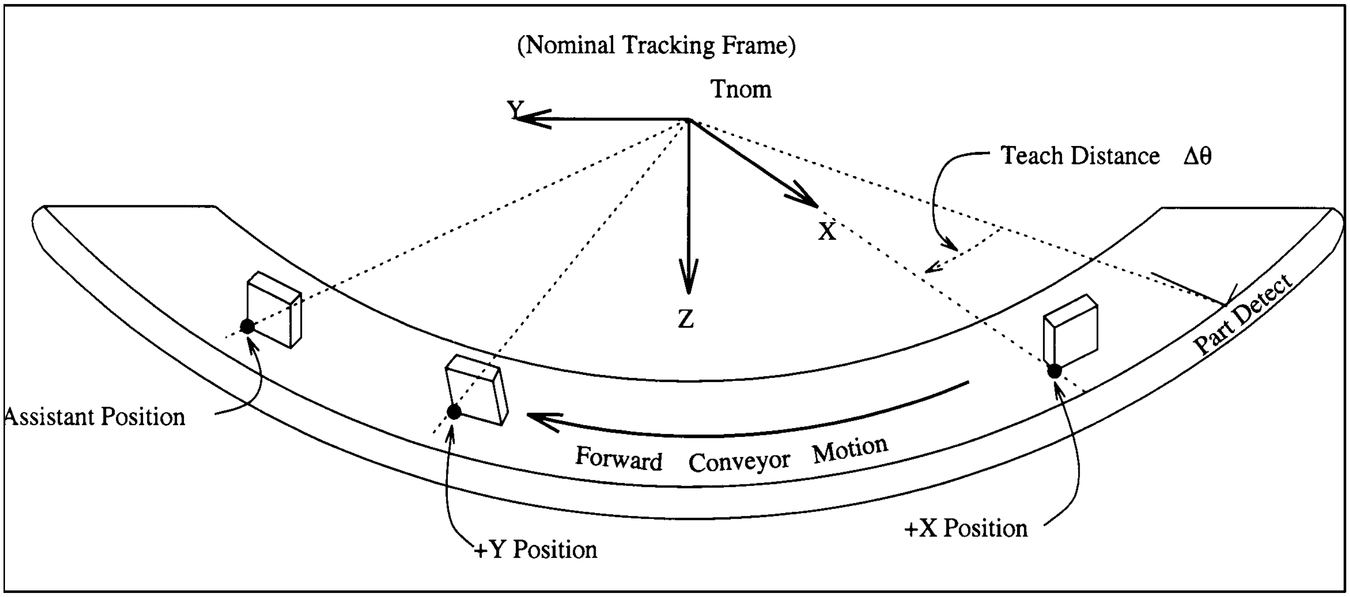


Figure 3–6. Nominal Tracking Frame - Clockwise Circular Tracking



Procedure 3-6 Three Point Method to Teach the Nominal Tracking Frame for Circular Tracking

Conditions

- You have selected Nominal Track Frame from the Tracking Setup Screen. Refer to Procedure 3-4 , Step 12 .
- You are currently at the Track Frame Screen.

Steps

1. Move the cursor to select the +x Direction Teach Method Data position. Refer to Figure 3–5 or Figure 3–6 for an illustration showing how to teach the nominal tracking frame for circular tracking.

```
SETUP Frames
 Track Frame Setup (Cir)          1/5
 Track Frame of Schedule:      2
  X:     0.00  Y:     0.00  Z:    0.00
  W:     0.00  P:     0.00  R:    0.00
 Teach Data:
  +X:RECORDED    Enc_cnt              2356
  X: 2241.80  Y:  754.09  Z:   30.00
  +Y dir:RECORDED    Enc_cnt         20356
  X: 2241.80  Y:  954.09  Z:   30.00
  Assist:UNINIT
  X:     0.00  Y:     0.00  Z:    0.00
 Scale (cnt/deg):     500.00
          TEACH   COMPUTE  SCALE
```

2. Move the robot TCP to a convenient position along the conveyor. (This position should be an easily distinguishable location either on the conveyor or on a part riding on the conveyor.)
3. Record this position by pressing SHIFT and TEACH simultaneously. The status of the +X Direction position should change to RECORDED.
4. Move the cursor to select the +Y Direction Teach Method Data position.
5. Move the robot away from the part so that the conveyor (and the part) can be moved without running into the robot.
6. Move the conveyor FORWARD (in the direction of normal part flow) for a distance of at least 30 to 40 degrees (the farther the better, as long as the robot will still be able to reach the new location of the part.)
7. Stop the conveyor.
8. Move the robot to the same location relative to the conveyor (or part) that was used for the +X Direction position.
9. Record this position by pressing SHIFT and TEACH. The status of the +Y Direction position should change to RECORDED.
10. Move the cursor to select the Assistant position.
11. Move the conveyor FORWARD (in the direction of normal part flow) for a distance of at least 30 to 40degrees. (The farther the better, as long as the robot will still be able to reach the new location of the part.)
12. Record this position by pressing SHIFT and TEACH simultaneously. The status of the Assistant position should change to RECORDED.
13. **To process all of the data positions and compute a new nominal tracking frame** , press F4, COMPUTE. When the processing is complete, the status of the three Teach method Data positions will be set to PROCESSED, and the Frame Components data values will be updated to display the new nominal tracking frame. See the following screen for an example.

```
SETUP Frames
 Track Frame Setup (Circ)          1/5
 Track Frame of Schedule:      2
  X: 1541.80  Y:  564.09  Z:    30.00
  W:    0.00  P:    0.00  R:    0
 Teach Data:
  +X:RECORDED    Enc_cnt          2356
  X: 2241.80  Y:  754.09  Z:   30.00
  +Y dir:RECORDED    Enc_cnt         20356
  X: 2241.80  Y:  954.09  Z:   30.00
  Assist:RECORDED
  X: 2341.80  Y:  1035.09  Z:   30.00
 Scale (cnt/deg):     500.00

          TEACH   COMPUTE  SCALE
```

14. You can setup the encoder scale for this line tracking schedule here or at scale item in the Tracking Setup main menu as in R30iA platform. If you want to setup the encoder scale at this time follow the next steps

15. Move the cursor to Origin or +X dir. The SCALE function will be display. Calculate the encoder scale by pressing SHIFT and SCALE. The scale value will be updated.

You have completed setup of the nominal tracking frame using the three point method. You can now go back to Tracking Setup at Procedure 3-4 , Step 16 .

Direct Entry

This method allows you to modify any of the frame component values (x, y, z, w, p, r) directly. This method is usually used when you copy data from another schedule.

Procedure 3-7 Directly Entering the Nominal Tracking Frame

Conditions

- You have selected Nominal Track Frame from the Tracking Setup Screen. Refer to Procedure 3-4 , Step 12 .
- You are currently at the Track Frame Screen.

Steps

1. Move the cursor to one of the Frame Component values.
2. Press ENTER to select a Frame Component.
3. Enter a new value. The nominal tracking frame system variable $LNSCH[n].$trk_frame is directly updated when you enter a new value, for the current tracking schedule.
4. Repeat Step 1 - Step 3 for each value (x, y, z, w, p, r) you want to set.

   You have finished the nominal tracking frame setup using the direct entry method. You can now go back to Tracking Setup at Procedure 3-4 , Step 16 .

3.3.2 Scale Factor Setup

The *encoder scale factor* is the conversion value used to correlate conveyor encoder count value information with conveyor motion.

This value is a real number (in units of encoder counts per millimeter or degrees) representing FORWARD conveyor motion. The sign (+/-) of this value is EXTREMELY important, since the encoder might be wired into the controller in such a way as to provide either increasing or decreasing count values for conveyor FORWARD motion. The sign of this value should not be confused with the value of the Track Axis Direction used for RAIL Tracking systems.

The encoder scale factor can be taught instead of computed manually. Use Procedure 3-8 to teach the encoder scale factor.

> **Warning**
>
> **Be sure that the robot's tool frame is properly defined before performing this procedure. Otherwise, you could injure personnel or damage equipment. Refer to your application-specific *Setup and Operations Manual* for more information.**

Teaching Hints

During this procedure, the two robot positions (the same position relative to the conveyor or part at two different conveyor positions) and the two corresponding conveyor positions, are recorded internally. The following equation is computed by the controller to determine the encoder scale factor value.

$$scale = \frac{change\ in\ encoder\ counts}{change\ in\ robot\ location}$$

Both conveyor distance and robot positioning accuracy are very important in the above computation. The conveyor should begin at the farthest up-stream end of the robot workspace, positioned so that the robot can still reach the part or marked location on the conveyor, and move to the farthest down-stream end of the robot workspace which meets the same constraints.

You should be very careful to position the robot TCP at the marked position on the part or conveyor, and should be equally precise when repositioning the robot at the second conveyor location. This will provide the highest possible resolution and accuracy for the encoder scale factor computation.

Note For rail tracking systems that use a non-integrated external axis (rail), only the rail position should be changed during this procedure. Otherwise, the result will be inaccurate.

> **Warning**
>
> **Move the robot directly away or up from the conveyor for clearance, when you are instructed to do so. Do not move the robot along the direction of the conveyor. Otherwise, you could injure personnel or damage equipment.**

Procedure 3-8 Teaching the Scale Factor

Conditions

- **This procedure can be done in Track Frame setup procedure for line tracking and circular tracking. If it has been done, You can skip this procedure. However for rail tracking, there is no tracking frame setup so scale set up have to be done in this menu**

You have selected Scale Factor then F2, TEACH and are currently at the Scale Factor screen. See the following screen (for line/rail tracking) for an example.

```
SETUP Encoder Scale
                                   3/3
 Track Schedule:               2
  Track Scale  (cnt/mm):       500.000

   Start Point:    RECORDED
    TCP X: 2241.80 Y: 754.09 Z:   30.00
    Encoder Count:                 2356
   End Point:      RECORDED
    TCP X: 2241.80 Y: 954.09 Z:   30.00
    Encoder Count:                13567



        TEACH   COMPUTE
```

Steps

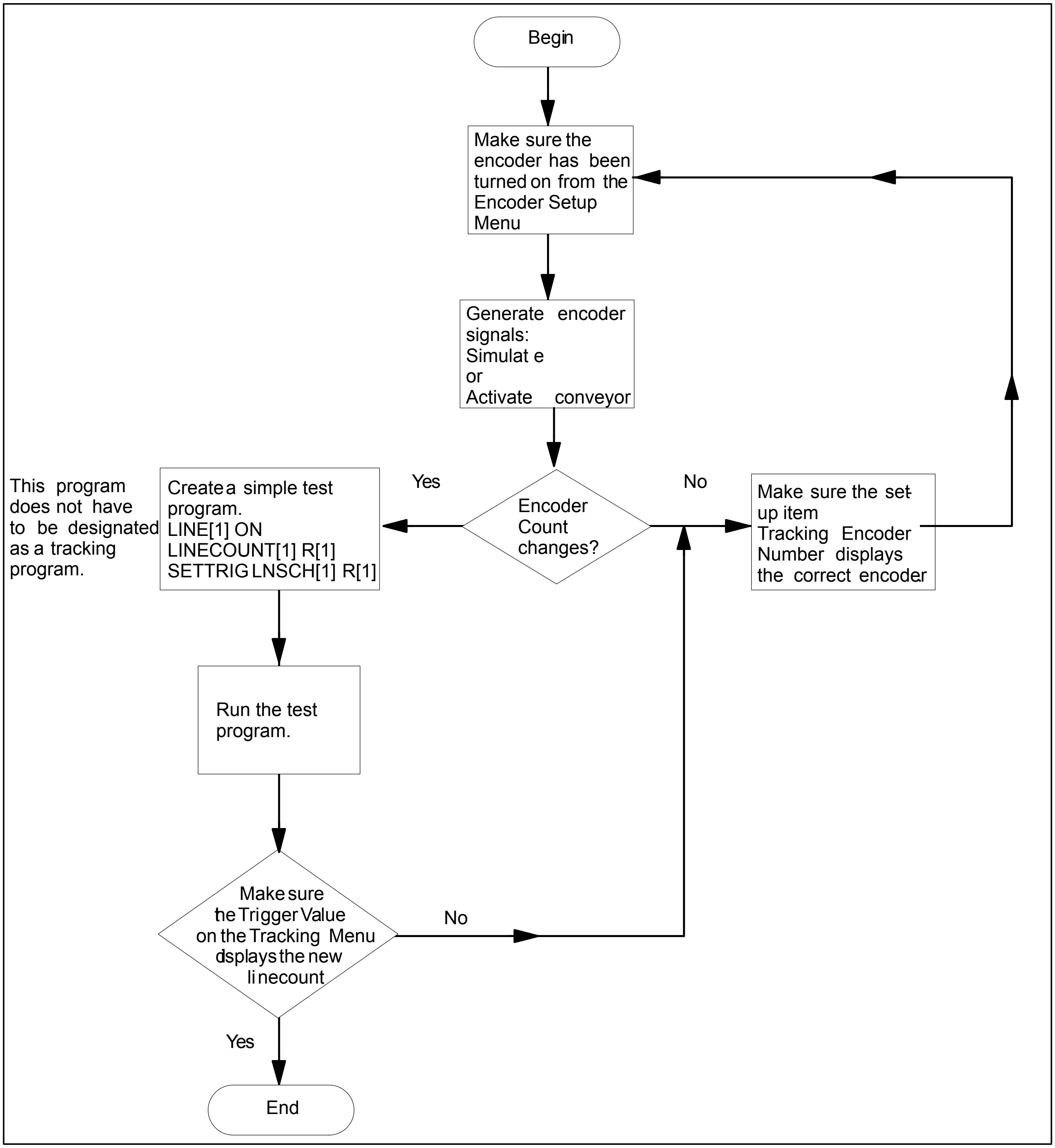
1. Move cursor to Start Point. Jog the robot TCP to a marked location on the part. Press SHIFT and TEACH simultaneously. The status of the Start Point will change from UNIINT to RECORDED. The TCP location and Encoder Count will be updated.
2. Move the TCP out of way so that conveyor can be moved without interference.
3. Move the conveyor FORWARD to position that part at the DOWN-STREAM end of the robot workspace.
4. Move cursor to End point. Jog the robot TCP to a marked location on the part. Press SHIFT and TEACH simultaneously. The status of the End Point will change from UNIINT to RECORDED. The TCP location and Encoder Count will be updated.
5. Press SHIFT and COMPUTE simultaneously. The Encoder Scale will be calculated and updated
6. Move the TCP out of way so that conveyor can be moved without interference

You have finished teaching the scale factor. You can now go back to Tracking Setup Procedure 3-4 Step 19 by pressing PREV key

3.3.3 Verify Tracking Setup is Correct

Figure 3–7 uses a flowchart to show you how to verify that you have setup tracking correctly. For a complete sample test program that you can use to verify line tracking operations, refer to Section 3.4 .

Figure 3–7. Verify Tracking Setup



3.4 VERIFY TRACKING SETUP SAMPLE PROGRAMS

This section contains a sample test program that you can use to help verify that you have setup your line tracking application correctly. Use this program after you have verified Encoder Setup (Section 3.1.2) and Tracking Setup (Section 3.3.3).

Note For more detailed information on planning, writing, or modifying a program, refer to Chapter 4 *PLANNING AND CREATING A PROGRAM* .

You can use this program to check basic line tracking functions. Three positions must be defined:

1. A safe home position for the robot.
2. A safe approach point for the robot to use when tracking the part
3. A point on the part for the robot to track.

3.4.1 MAIN Program Example

In the MAIN Program example in Main Program :

- Header data: Schedule 0 (Non-tracking)
- DIN[1] is used for the part detect input.
- R[1] is available for use.
- Bound[1] boundary values have already been set to reasonable values (ex. set to 0 and 1000).
- P[1] is a safe home position.

Main Program

```
PROG  MAIN
/MN
   1:  ! MOVE TO HOME
   2:  J P[1] 50% FINE ;
   3:  ! ENABLE THE ENCODER
   4:  LINE[1] ON ;
   5:  ! WAIT FOR A PART DETECT
   6:  WAIT DI[1] ON ;
   7:  ! GET TRIGGER VALUE
   8:  LINECOUNT[1] R[1] ;
   9:  ! SET TRIGGER VALUE
  10:  SETTRIG LNSCH[1] R[1] ;
  11:  ! SELECT A BOUNDARY
  12:  SELBOUND LNSCH[1] BOUND[1] ;
  13:  ! CALL TRACKING PROGRAM
```

```
  14:  CALL TRACK ;
  15:  ! MOVE TO HOME
  16:  J P[1] 50% FINE ;
/END
```

3.4.2 TRACK Sub Program Example

In the TRACK Program example in Sub Program: TRACK :

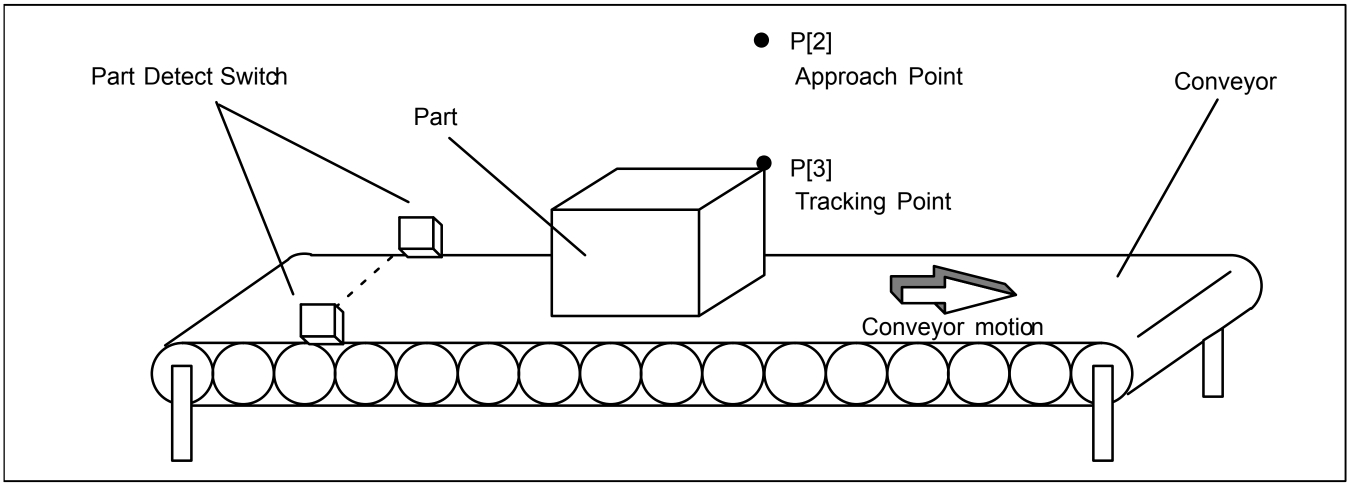
- Header data: Schedule 1, Cont. Track = FALSE, SELBND=0.
- P[2] is a safe approach point, typically above the part (see Figure 3–8).
- P[3] is a known location on the part (see Figure 3–8).

> **Caution**
>
> Do not use a PAUSE instruction in your TRACK program or in a subprogram that is called by the TRACK program. Doing so could result in unexpected motion when the TRACK program resumes.

Sub Program: TRACK

```
/PROG  TRACK
/MN
   1:  ! MOVE TO APPROACH POS
   2:  L P[2] 500mm/sec FINE ;
   3:  ! MOVE TO PART
   4:  L P[3] 500mm/sec FINE ;
   5:  ! WAIT FOR 5 SECONDS
   6:  WAIT 5.00(sec) ;
   7:  ! MOVE TO APPROACH POS
   8:  L P[2] 500mm/sec FINE ;
/END
```

Figure 3–8. Approach and Tracking Points



3.4.3 Verify Sample Program Execution is Correct

When the program MAIN (Main Program and Sub Program: TRACK) is run, the robot should:

1. Move to the home position P[1].
2. Wait until the part is detected, and has moved inside of the boundary 1 tracking window (IN-WINDOW).
3. Move to the approach point P[2].
4. Move to the taught location on the part P[3].
5. Track that point (P[3]) on the part for 5 seconds.
6. Move away from the part, back to P[2].
7. Return to its home position P[1].

Robot Does Not Move as Planned

If the robot does not move as planned.

1. Ensure the nominal tracking frame for schedule 1 has been taught correctly. (See Section 3.3.1)
2. Ensure the encoder scale factor for schedule 1 has been taught correctly. (See Section 3.3.2)
3. Ensure the part detect distance has been measured and set correctly. (See Figure 3–2)
4. Confirm the tracking system was correctly synchronized before the positions were taught. (See Procedure 4-3.)
5. Ensure encoder simulation is OFF.

Robot Does Not Move to Tracking Positions

If the robot fails to move to tracking positions.

1. Ensure the encoder scale factor for schedule 1 has been taught correctly. (See Section 3.3.2)
2. Ensure the boundary values for schedule 1 are correct, particularly if the error message **TRAK-005 Track Destination Gone** is displayed. (See Figure 3–3)
3. While the program is running, go to the Tracking Setup Menu.
   a. Verify the trigger value displayed updates once at the beginning of the program.
   b. Verify the encoder currency count is updating while the program is running.
   c. Verify the value is reasonable for the encoder used. (See Procedure 3-4)

```
SETUP Encoders
       Encoder Number:  1
1 Encoder Axis:                          1
2 Encoder Type:                INCREMENTAL
3 Encoder Enable:                      OFF
  Current Count (cnts):                  1
4 Multiplier (ITP/update):               1
5 Average (updates):                     1
6 Stop Threshold (cnt/updt):             0
7 Simulate:       Enable:              OFF
8                 Rate (cnt/updt):       0
[TYPE]             ENCODER
```

Chapter 4

PLANNING AND CREATING A PROGRAM

Contents

4.1 OVERVIEW

A *FANUC America Corporation program* includes a series of commands, called instructions, that tell the robot and other equipment how to move and what to do to perform a specific task. For example, a program directs the robot and controller to:

1. Move the robot in an appropriate way to required locations in the workcell.
2. Perform an operation, such as spot weld, paint, or arc weld.
3. Send output signals to other equipment in the workcell.
4. Recognize and respond to input signals from other equipment in the workcell.
5. Keep track of time, part count, or job number.

This chapter describes how to perform each of the following, as related to line tracking.

- Plan a program, Section 4.2
- Write and Modify a program, Section 4.3
  - — Synchronize the conveyor, Section 4.3.2
  - — Re-synchronize the conveyor, Section 4.3.5
  - — Set up and use predefined positions in a program, Section 4.3.6
- Use line tracking program instructions, Section 4.5

Planning a Program

Before you write a program, you should **plan the program** . Planning involves considering the best way possible to perform a specific task before programming the robot to complete that task. Planning before creating a program will help you choose the appropriate instructions to use when writing the program.

Writing a Program

You **write a program** using a series of menus on the teach pendant that allow you to select and add each instruction to your program. If the program sequence requires you to define the current location of the robot, you jog, or move, the robot to the desired location and execute the appropriate instruction.

Modifying a Program

After you create a program, you can **modify the program** . You can use a series of teach pendant screens to change or remove an instruction, add a new instruction, move instructions from one location in the program to another, or find specific sections of the program.

If the instruction requires defining the current location of the robot, you jog, or move, the robot to the desired location and add the appropriate instruction.

4.2 PLANNING A PROGRAM

This section describes basic tracking teach pendant programming, including discussions of program organization, use of tracking schedules, special programming situations, and example programs.

4.2.1 Programming a Typical Line Tracking System

A typical tracking system will consist of a single robot that tracks parts along a single conveyor system using one or more tracking schedules, one main teach pendant program to monitor incoming parts, and one or more sub teach pendant programs that control all robot motions.

Use Procedure 4-1 as a guideline when programming a typical line tracking system.

Procedure 4-1 Programming a Typical Set of Tracking Programs

Conditions

- All required hardware has been set up.
- All required parameters on the Encoders and Tracking SETUP screens have been defined. All SETUP information must be set before you begin any programming.

Note Be sure to set all detail program header information before you begin programming.

Steps

1. To begin programming, you should create a main non-tracking program (also referred to as a "job"). The line track program header data should have the line track schedule number set to 0 for all non-tracking programs. Refer to Section 4.3 for more information on creating and modifying a program.

   This job/program will complete the following steps:

   a. Turn "ON" the tracking encoder.

   b. Call or run a (non-tracking) program to move the robot to a home or rest position.

   c. Monitor the part detect switch for the approach of a part.

   d. Record the conveyor count at the time of the part detect.

   Note Step 1d must be done immediately after detection of the change in state of the part detect switch to ensure proper synchronization between the robot and the moving part.

   e. Store this count (the part "trigger value").

   f. Select a boundary set.

g. Call or run one or more other (tracking) programs to move the robot through the desired processing task.

h. Call or run another (non-tracking) program to move the robot away from the moving conveyor. (This step is optional.)

i. Return to Step 1 to return the robot to its rest position to wait for the approach of the next part.

Note Be sure that all of the line tracking instructions used within this program specify the desired line track schedule number, that you set up in the program header, wherever appropriate. This is the number that will be used in the detail program header data for all corresponding tracking motion programs. This is critical to all tracking program motions.

2. Move the robot to a rest position to create the program for use in Step 1 . This also moves the robot out of the way to prepare for the next step of creating one or more tracking programs to carry out the processing task.

3. Display the Encoders SETUP menu to make sure that the tracking encoder has been enabled (turned "ON"). The encoder must be enabled in order for the system to properly perform the robot-conveyor synchronization, prior to recording path positions.

This step should be repeated each time before creating a tracking program. If the encoder is "OFF" at the time of a tracking program creation, you will be instructed to exit the edit session to enable the encoder.

4. You should now create a sub tracking program (also referred to as a "process"), to perform tracking motions.

The line track program header data (for the *sub* program) should have the line track schedule number set to the number of the schedule whose parameters were set up prior to programming, and whose number was specified within the *main* program.

5. Upon entering the program edit session for any tracking program, you will be prompted to *synchronize* the conveyor with the robot. This generally consists of moving the conveyor so that a part passes the part detect switch and then enters the robot workspace.

Note For more information on synchronizing the robot and conveyor, refer to Section 4.2.2.

The line tracking system will take care of monitoring the part detect switch and recording and storing the encoder count/trigger value (provided you have properly set up the Encoders and Tracking SETUP menu parameters) to be used during the path teaching session.

6. After the part is reachable by the robot, the conveyor can be stopped and path positions can be taught. The conveyor can be moved in either direction to reposition the part anywhere within the robot workspace during the programming session.

Each time a position is recorded (or touched-up) the tracking system automatically determines the conveyor location and adjusts the recorded positions accordingly. You can also play back or single-step through the program to test for desired robot motion.

Note Boundary position checking is enforced during program execution or single-stepping, as determined by the value of $LNSCH[1].$SEL_BOUND within the line track schedule associated with the program. This might cause the robot to pause motionless if a position is not within the selected boundary window. However setting $LNSCH[1].$SEL_BOUND = -1 will disable the boundary checking and facilitate program editing.

Refer to Appendix A for more information on how to set $LNSCH[1].$SEL_BOUND.

Warning

Make sure you have set the value of SELECT BOUND properly before you run production. Otherwise, you could injure personnel or damage equipment.

You have completed planning a typical tracking program. For information on creating or modifying a program, refer to Section 4.3 . For details about specific tracking instructions, refer to Section 4.5 .

4.2.2 Program Examples

Three programming examples are listed in the following sections.

- The first section provides an example of a *job* or *main* program which monitors the conveyor for new parts, sets the tracking trigger value, and calls all robot motion programs. Refer to Job Program Example .
- The second is a process which moves the robot to a non-tracking rest position. Refer to Move to Rest Position .
- The third is a process which consists of a number of line tracking motions. Refer to Move and Track Conveyor .

Job Program Example

The following routine acts as a line tracking sequencing program, which monitors the conveyor and issues all robot motion routine calls.

Job Program Example

```
/PROG  MO250020
/MN
  1: ! Turn on Encoder ;
  2: LINE[1] ON ;
  3: !
  4: ! Move to rest pos (non-tracking)
  5: LBL[1] ;
```

```
  6: CALL MO250021 ;
  7: !
  8: !
  9: ! Wait for a part detect trigger
 10: LBL[2] ;
 11: WAIT DI[32]=ON ;
 12: !
 13: ! Read the line count and rate
 14: LINECOUNT[1] R[1] ;
 15: LINERATE[1] R[2] ;
 16: !
 17: ! Make sure the conveyor moves fwd
 18: IF R[2]<0, JMP LBL[2] ;
 19: !
 20: ! Store the trigger value
 21: LBL[4] ;
 22: SETTRIG LNSCH[1] R[1] ;
 23: !
 24: ! Select a boundary set
 25: SELBOUND LNSCH[1] BOUND[1] ;
 26: !
 27: ! Move to track the conveyor
 28: CALL MO250022 ;
 29: !
 30: ! Move to rest pos (non-tracking)
 31: CALL MO250021 ;
/POS
/END
```

Process Program Examples

Move to Rest Position and Move and Track Conveyor provide process program examples.

Move to Rest Position moves the robot to a rest or home position which does NOT track the conveyor.

Move to Rest Position

```
/PROG  MO250021
/MN
  1:J P[1] 100% FINE ;
/POS
/END
```

Move and Track Conveyor moves the robot to a number of locations while the robot also tracks the moving conveyor.

Move and Track Conveyor

```
/PROG  MO250022
/MN
  1:L P[1] 800mm/sec FINE ;
  2: WAIT .50(sec) ;
  3:L P[2] 800mm/sec FINE ;
  4:L P[3] 800mm/sec CNT100 ;
  5:L P[4] 800mm/sec CNT100 ;
  6:L P[5] 800mm/sec FINE ;
  7:C P[6]
   :  P[7] 800mm/sec FINE ;
  8:C P[8]
   :  P[9] 800mm/sec CNT100 ;
  9:C P[10]
   :  P[11] 800mm/sec FINE ;
/POS
/END
```

4.3 WRITING AND MODIFYING A PROGRAM

You can write new programs and modify existing programs to direct the robot to perform a task. Writing a program includes:

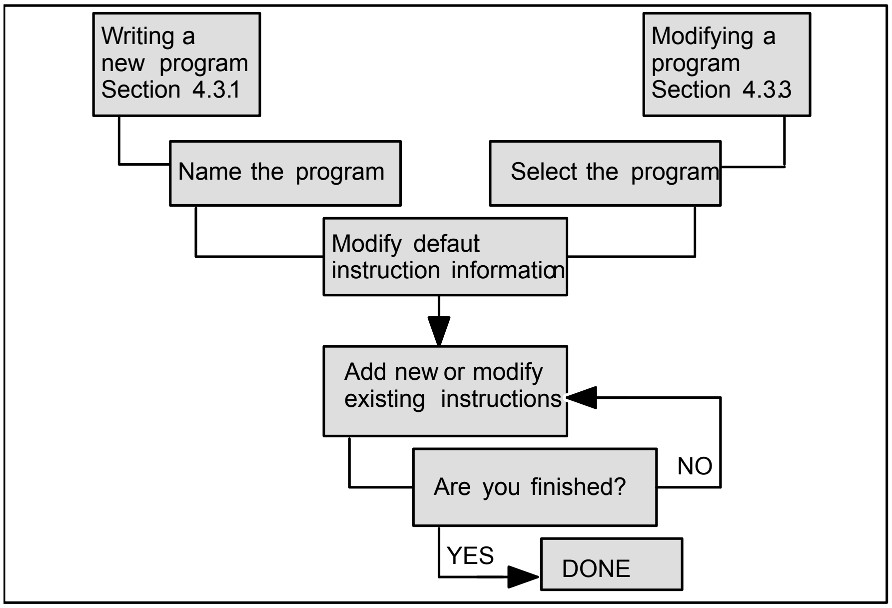
- Naming the program
- Defining detail information
- Defining default instructions
- Adding instructions to the program

Modifying a program includes:

- Selecting the program
- Modifying default instructions
- Inserting instructions
- Deleting instructions
- Copying and pasting instructions
- Searching for instructions
- Renumbering instructions

Figure 4–1 summarizes writing and modifying a program.

Figure 4–1. Writing and Modifying a Program



4.3.1 Writing a New Program

When you write a new program you must

- **Name** the program and set program header information.
- **Define** the detailed information for the program (refer to Section 4.5):
  - — Whether it is a job or process
  - — All paint related data
- **Modify** default instruction information. This includes modifying motion instructions and any application specific instructions.
- **Add** motion instructions to the program.
- **Add** arc welding, dispensing, material handling, painting, spot welding, and other instructions to the program.

Use Procedure 4-2 to create a program manually.

Naming the Program

You can name a program using three different methods:

- **Words -** This method provides a list of words you can use to build a program name. These words can vary depending on the kind of software you have.

- **Upper Case -** This method lets you use upper case letters and any numbers.
- **Lower Case -** This method lets you use lower case letters and any numbers. For the program name, lower case letters are automatically converted to upper case after you enter them.

Options allows you to change whether you are overwriting, inserting, or clearing the program name or comment information. The screen will display either Insert or Overwrite. Clear allows you to remove text from the current field.

The total length of the program name must be no more than eight characters. You can combine words, upper case letters, and lower case letters to form the program name. Give the program a unique name that indicates the purpose of the program.

Note Do not use the asterisk * symbol in program names.

Defining Detail Information

The detail of program header information includes:

- Creation date
- Modification date
- Copy Source
- Number of positions and program size
- Program Name
- Sub Type
- Comment
- Group mask
- Write protection
- Speed override
- Default user frame
- Default tool frame
- Start delay

Defining Default Instruction Information

Motion instructions tell the robot to move to an area in the workcell in a specific way. When you create a program you can define, in advance, the way you want the robot to move when you add a motion instruction. You do this by defining default motion instruction information.

After you have defined the default instructions you can add them to the program. You select one of the available default instructions to be the current default instruction by moving the cursor to that instruction. You can define and change default instructions any time while writing or modifying a program.

Adding Instructions

You can also add other instructions not included in the default motion instruction to your program. To add these instructions, select the kind of instruction you want to add to the program and use the information on the screen to enter specific instruction information.

You add all instructions using the same general procedure. Motion instructions, however, require some specific information. Refer to your application-specific *Setup and Operations Manual* for information on adding motion and other kinds of instructions.

Procedure 4-2 Creating and Writing a New Program

Conditions

- All personnel and unnecessary equipment are out of the workcell.
- All encoder and tracking parameters have been set up. (Refer to Section 3.1 and Section 3.3 .)

Naming the Program

1. Press SELECT.
2. Press F2, CREATE.

```
1 Words
  2 Upper Case
  3 Lower Case
  4 Options                -- Insert --
Select
  --- Create Teach Pendant Program ---
Program Name [        ]
                           -- More --
Press ENTER for next item
  JOB      PROC     TEST     MM_
```

3. Enter the program name:

   a. Move the cursor to select a method of naming the program: Words, Upper Case, or Lower Case.

   b. Press the function keys whose labels correspond to the name you want to give to the program. These labels vary depending on the naming method you chose in Step 3a . **To delete a character,** press BACK SPACE.

   For example, if you chose Upper Case or Lower Case, press a function key corresponding to the first letter. Press that key until the letter you want is displayed in the program name field. Press the right arrow key to move the cursor to the next space. Continue until the entire program name is displayed.

c. When you are finished, press ENTER. You will see a screen similar to the following.

```
1 Words
  2 Upper Case
  3 Lower Case
  4 Options                -- Insert --
Select
  --- Create Teach Pendant Program ---
Program Name [PROC742     ]
                            -- End --
Select function
```

4. **To display program header information** , press F2, DETAIL. You will see a screen similar to the following.

```
Program Detail
   Creation date:              ##-xxx-##
   Modification Date:          ##-xxx-##
   Copy source:        [              ]
   Positions:   10  Size          17 Byte
    1 Program Name     [       PROC742]
    2 Sub Type:        [PROCESS       ]
    3 Comment:         [              ]
    4 Group mask:      [1,*,*,*,*     ]
    5 Write protect:   [ON            ]
    6 Ignore Pause:    [OFF           ]
```

To skip setting program header information and begin editing the program, press F1, END, and skip to **Defining Default Motion Instructions** in this procedure.

Note You **must** set all DETAIL information when you create a PaintTool program.

5. **To set or rename the program,** move the cursor to the program name and press ENTER.

   a. Move the cursor to select a method of naming the program: Words, Upper Case, or Lower Case.

   b. Press the function keys whose labels correspond to the name you want to give to the program. These labels vary depending on the naming method you chose in Step 5a . **To delete a character,** press BACK SPACE.

6. **To select a sub type,** move the cursor to the sub type and press F4, [CHOICE]. You will see a screen similar to the following.

```
Sub Type
1 None
2 Job
3 Process
4 Macro
Program Detail
    1 Program Name        [        PROC742]
    2 Sub Type:           [PROCESS        ]
    3 Comment:            [               ]
    4 Group mask:         [1,*,*,*,*      ]
    5 Write protect:      [ON             ]
```

   a. Select whether the sub type is None, Job, Process, or Macro.
   b. Press ENTER.

7. **To type a comment,** move the cursor to the comment and press ENTER.
   a. Select a method of naming the comment.
   b. Press the appropriate function keys to add the comment.
   c. When you are finished, press ENTER.

   For example, if you chose Upper case, press a function key corresponding to the first letter. Press that key until the letter you want is displayed in the comment field. Press the right arrow key to move the cursor to the next space. Continue until the entire comment is displayed.

8. **To set the group mask (or motion group),** move the cursor to group you want to enable or disable. You can use multiple groups in a single program, but only two groups can perform Cartesian motion within a single program. The first position in the group mask corresponds to the first group. Only the group 1, 2, and 3 is currently available.
   a. To enable a group, press F4, 1.
   b. To disable a group, press F5, *.

   Note If your system is not set up for multiple groups, you will only be able to select a 1, for the first group, or a *, for no group.

   Note After the group mask has been set, and motion instructions have been added to the program, the group mask cannot be changed for that program.

9. **To set write protection,** move the cursor to write protection.
   a. To turn write protection on, press F4, ON.

b. To turn write protection off, press F4, OFF.

10. **To set ignore pause** , move the cursor to ignore pause.

 a. To turn on ignore pause, press F4, ON.

 b. To turn off ignore pause, press F4, OFF.

11. **To display the line tracking header information press F3, NEXT (or F2, PREV)** .

12. **Set the line tracking schedule number** to a value between 1 and 6. Schedule number 1 is the default frame.

 Note A line tracking schedule number of 0 indicates a non-tracking path.

13. **Set the continue track at program end** to TRUE or FALSE.

 - To have the robot continue to track after the program has finished, press F4, TRUE.
 - To have the robot stop tracking when the program finishes, press F5, FALSE.

14. **Selected Boundary** indicates the current boundary in the line tracking schedule. The current boundary value is automatically updated when you execute a program.

 - If Selected Boundary is set to zero (which is the default), the boundary value will not be updated when the program is executed.
 - If Selected Boundary is greater than zero, the boundary value will be automatically updated. When the program finishes, the current selected boundary value in the line tracking schedule will be reset.

 For example, if the current value of $LNSCH[1].$sel_bound = 1, and a line tracking program is executed with Selected Boundary = 6, $LNSCH[1].$sel_bound will be set to 6. When the program is finished, the current boundary will be set back to 1.

 Note If the system variable $LNCFG.$rstr_bnds = FALSE, the boundary will **not** be restored when the program finishes.

15. **To return to the detail screen or display more header information** , press F3, NEXT, (or F2, PREV) until F1, END is displayed.

16. Press F1, END. You must now synchronize the robot and conveyor before you begin adding motion instructions. Refer to Section 4.3.2 .

Background Program Editing

The teach pendant must remain ON to jog the robot. If you cannot turn the teach pendant on because another program is running, or if you turn the teach pendant off while writing or modifying the program, you must write or modify in the *background* .

Writing the program in the background means the motion instructions you add will be in the program but the positions will not be recorded.

If you write or modify a program in the background you can return to writing it when you are able to turn on the teach pendant. Then you can touch up the motion instructions to record the positional data. Refer to Section 4.3.3 , Modifying a Program, for more information.

17. To define default motion instructions

a. Continuously press the DEADMAN switch and turn the teach pendant ON/OFF switch to ON.

b. Press F1, POINT. You will see a list of default motion instructions.

Note If the instructions listed are the ones you want to use, do not modify them.

c. Press F1, ED_DEF.

d. Move the cursor to the default instruction you want to modify.

e. Move the cursor to the component you want to modify.

f. Use the appropriate keys and function keys to modify the component and press ENTER.

If the [CHOICE] function key is displayed, press F4 to display a list of values for the selected component.

For example, to change the speed value, move the cursor to 100. Type a new value and press ENTER. The new value will be displayed. Each time you add this instruction to the program the new value will be used.

g. Repeat Step 17d through Step 17f for each default instruction that you want to define.

h. **When you are finished defining default motion instructions,** move the cursor to the instruction you want to be the current default instruction and press F5, DONE.

i. **To save the modified default motion instructions,** refer to the section on titled "Backing Up Program System and Application Files," in your application-specific *Setup and Operations Manual.*

18. To record the position using the current default motion instruction

a. Jog the robot to the location in the workcell, that is between the selected boundaries, where you want to record the motion instruction. Refer to Section 3.3 for more information on setting up boundaries.

b. Press and hold in the SHIFT key and press F1, POINT. The instruction will be added to the program automatically.

Caution

Do not use UFRAMES when recording positions in a line tracking program. If you do, the point will remain uninitialized and an error will occur.

19. To record the position using one of the other three default motion positions

a. Jog the robot to the location in the workcell where you want to record the motion instruction.

b. Press F1, POINT.

c. Use the cursor to select new default positions.

d. Press ENTER. This records the position and selects the motion instruction as the default motion instruction.

20. **To add other instructions,** press F2, [INST]. Select the kind of instruction you want and use the appropriate selections on the screen to build the instruction.

Refer to the previous sections in this chapter for details about each instruction.

21. Turn the teach pendant ON/OFF switch to OFF and release the DEADMAN switch.

Note To test the program, refer to your application-specific *Setup and Operations Manual*.

4.3.2 Synchronizing the Robot and Conveyor

This section guides you through the robot/conveyor synchronization process. This process establishes the position of a part along the conveyor (relative to the nominal tracking frame), so that the robot tracking positions and motions can be recorded or executed.

Procedure 4-3 Synchronizing the Robot and Conveyor

Conditions

- Synchronization must be performed within the user programming environment, during product operation. This is typically accomplished using the line tracking TPE LINECOUNT and SETTRIG instructions.
- You are working with a newly created program.

Note The synchronization process is automatically entered any time you enter a teach pendant program for editing. You will see a screen similar to the following.

```
---------------------------------------
|                                     |
|  The current part detect            |
|  trigger value is                   |
|  NOT VALID!                         |
|  Must Resynchronize the             |
|  tracking system!                   |
|                                     |
|                OK                   |
|                                     |
---------------------------------------
```

1. Select OK. You will see a screen similar to the following.

```
 -------------------------------------
 |                                   |
 |  Stop the part when it is         |
 |  at a convenient location         |
 |  in the robot workspace.          |
 |                                   |
 |  Press <ENTER> when done.         |
 |                                   |
 |                                   |
 |                                   |
 -------------------------------------
```

2. Move the conveyor, with the part on it, to the desired position.
3. **When the conveyor is in the desired teach position** , press ENTER. You will see a screen similar to the following.

```
EDIT
 [END]

[INST]     STAT                 TOUCHUP     [EDCMD]
```

4. You can now begin adding motion instructions to the program. Refer to Section 4.3.3 .

4.3.3 Modifying a Program

You can modify an existing program any time you want to change the content of the program. Modifying a program includes

- **Selecting** a program
- **Modifying** motion instructions
- **Modifying** other instructions
- **Inserting** instructions
- **Deleting** instructions
- **Copying and pasting** an existing instruct ion or program element
- **Finding and replacing** an existing instruction or program element
- **Renumbering** positions after instructions have been added, removed, or moved

Selecting a Program

Selecting a program is choosing the program name from a list of existing programs in controller memory. Refer to your application-specific *Setup and Operations Manual* for more information on loading programs.

Touching Up and Modifying Motion Instructions

Touching up motion instructions changes any element of the motion instruction. The element you might modify most often is the position data.

Modifying Other Instructions

Modifying other instructions changes any element of the instruction.

Inserting Instructions

Inserting instructions places a specified number of new instructions between existing instructions. When you insert an instruction, the instructions that follow the new instruction are automatically renumbered.

Deleting Instructions

Deleting instructions removes them from the program permanently. When you remove an instruction the remaining instructions are automatically renumbered.

Copying and Pasting Instructions

Copying and pasting is selecting a group of instructions, making a copy of the group, and inserting the group at one or more locations in the program.

Finding and Replacing Instructions

Finding and replacing is finding specific instructions and, if desired, replacing those instructions with new instructions. This function is useful, for example, when setup information that affects the program is changed. It is also useful when you need to find a specific area of a long program quickly.

Renumbering Positions

Renumbering allows you to renumber positions in the program. When you add positions to a program, the first available position number is assigned to the position, regardless of its place in the program. When you delete motion instructions, all remaining positions keep their current numbers. Renumbering reassigns all position numbers in the program so that they are in sequential order.

Use Procedure 4-4 to modify a program.

Procedure 4-4 Modifying a Program

Conditions

- All personnel and unnecessary equipment are out of the workcell.
- The program has been created and all detail information has been set correctly. (Procedure 4-2)

Steps

1. Press SELECT.
2. Display the appropriate list of programs:
   a. Press F1, [TYPE].
   b. Select the list you want:
      - **All** displays all programs.
      - **Jobs** displays all job programs.
      - **Processes** displays all process programs.
      - **TP Programs** displays all teach pendant programs.
      - **KAREL Progs** displays all KAREL programs.
      - **Macro** displays all macro programs.
3. Move the cursor to the name of the program you want to modify.
4. Press ENTER.
5. Continuously press the DEADMAN switch and turn the teach pendant ON/OFF switch to ON.
6. You will be prompted to synchronize the conveyor. Refer to Section 4.3.5 for information on how to re-synchronize the conveyor and robot.
7. **To touch up and modify motion instructions**
   a. Move the cursor to the **line number** of the motion instruction you want to modify.
   b. **To change only the position component of the motion instruction,** jog the robot to the new position, press and hold in the SHIFT key and press F5, TOUCHUP.

> **Warning**
>
> **Changing the motion type of a positional instruction from linear to joint can cause the speed value to change from mm/sec to a default value as high as 100%. Be sure to check the speed value before you execute the instruction; otherwise, you could injure personnel or damage equipment.**

c. **To change other motion instruction components,** move the cursor to the component using the arrow keys, and press the appropriate function keys to modify the component:

- **If function key labels are available,** press the appropriate one.
- **If no function key labels are available,** press F4, [CHOICE], and select a value.
- **To change the position value,** move the cursor to the position number and press F5, POSITION. The position screen will be displayed showing the Cartesian coordinates or joint angles of the selected position. Move the cursor to the component you want to change and enter the new value using the number keys. To make other changes, use the function keys described in the list below.
  - — **To change the motion group number,** press F1, GROUP. This applies only to systems that have been set up for multiple groups.
  - — **To display components for extended axes,** press F2, PAGE. This only applies to systems that include extended axes.
  - — **To change the configuration** between flip (F) and no-flip or normal (N), press F3, CONFIG, and then use the up and down arrow keys to change F to N and N to F.

    Note Joint angles are useful for zero-positioning the robot or for non-kinematic motion control such as controlling the motion of a positioning table.
  - — **To change the format of the position from Cartesian coordinates to joint angles or from joint angles to Cartesian coordinates,** press F5, [REPRE] and select the coordinate system. The position is converted automatically.
  - — When you are finished, press F4, DONE.

d. Repeat Step 7a through Step 7c for each motion instruction you want to modify.

8. To modify other instructions

a. Move the cursor to the line number of the instruction you want to modify.

b. Move the cursor to the component you want to modify and press the appropriate key:

- **If function key labels are available,** press the appropriate one.
- **If no function key labels are available,** press F4, [CHOICE], and select a value.

c. Repeat Step 8a and Step 8b for each instruction you want to modify.

9. To insert instructions

a. Decide where you want to insert the instruction. Move the cursor to the line following that point. The cursor must be on the **line number** . For example, if you want to insert between lines 5 and 6 place the cursor on line 6.

b. Press NEXT, >, until F5, [EDCMD] is displayed. The function key labels for F1 through F4 might vary depending on your application.

c. Press F5, [EDCMD].

d. Select 1, Insert.

e. Type the number of lines to insert and press ENTER. A blank line will be inserted into the program for each line you want inserted. All lines in the program will be renumbered automatically.

f. Move the cursor to the line number of any inserted line and add any instruction.

10. To delete instructions

a. Move the cursor to the line number of the instruction you want to delete. If you want to delete several instructions in consecutive order, move the cursor to the first line to be deleted.

> **Caution**
>
> Deleting an instruction permanently removes the instruction from the program. Be sure you want to remove an instruction before you continue; otherwise, you could lose valuable information.

b. Press NEXT, >, until F5, [EDCMD] is displayed. The function key labels for F1 through F4 might vary depending on your application.

c. Press F5, [EDCMD].

d. Select 2, Delete.

e. **To delete a range of lines,** move the cursor to select the lines to be deleted. The line number of each line to be deleted will be highlighted as you move the cursor.

f. Delete the line or lines:

- **If you do not want to delete the selected line(s)** , press F5, NO.

- **To delete the selected line(s)** press F4, YES.

Note You can copy instructions from one program and paste them within that program or into another program.

11. To copy and paste instructions

a. Press NEXT, > until F5, [EDCMD] is displayed. The function key labels for F1 through F4 might vary depending on your application.

b. Press F5, [EDCMD].

c. Select 3, Copy.

d. Move the cursor to the first line to be copied.

e. Press F2, COPY.

f. Move the cursor to select the range of lines to be copied. The line number of each line to be copied will be highlighted as you move the cursor.

g. Press F2, COPY, again. Your information will be stored in a buffer.

h. Decide where you want to paste the lines. Move the cursor to the line following that point. The cursor must be on the **line number** .

i. Press F5, PASTE.

j. Press the function key that corresponds to the way you want to paste the copied lines:

- **LOGIC (F2) -** adds the lines exactly as they were, does not record positions, and leaves the position numbers blank.

- **POS_ID (F3) -** adds the lines exactly as they were and retains the current position numbers.

- **POSITION (F4) -** adds the lines exactly as they were and renumbers the copied positions with the next available position numbers. All positional data is transferred.

- **CANCEL (F5) -** cancels the paste, but the copied lines are retained so you can paste them elsewhere.

- **R-LOGIC (NEXT+F2)** - adds the lines in reverse order, does not record the positions, and leaves the position numbers blank.

- **R-POS-ID (NEXT+F3) -** adds the lines in reverse order and retains their original position numbers.

- **R-POSITION (NEXT+F4) -** adds the lines in reverse order and renumbers the copied positions with the next available position numbers.

k. Repeat Step 11a through Step 11j to paste the same set of instructions as many times as you want.

l. **When you are finished copying and pasting instructions,** press PREV.

12. To find instructions

a. Move the cursor to the line number of any instruction.

b. Press NEXT, >, until F5, [EDCMD], is displayed. The function key labels for F1 through F4 might vary depending on your application.

c. Press F5, [EDCMD].

d. Select 4, Find.

e. Select the kind of instruction to find.

f. When prompted, enter the necessary information. The system searches forward from the current cursor position for the item you want. If it finds an instance of the item, it highlights it on the screen.

g. **To find the next instance of the item,** press F4, NEXT.

h. **When you are finished finding items,** press F5, EXIT.

i. Press PREV.

13. To replace instructions

a. Move the cursor to the line number of any instruction.

b. Press NEXT, > until F5, [EDCMD], is displayed. The function key labels for F1 through F4 might vary depending on your application.

c. Press F5, [EDCMD].

d. Select 5, Replace.

e. Select the instruction you want to replace from the list of instructions. Follow the information on the screen to specify the instruction. The system finds the first instance of the existing instruction and highlights it.

f. Select the replacement item and enter the necessary information.

g. Decide how to replace the instruction:

- **To replace the existing instruction with the new instruction** press F3, YES. The system will prompt you to search for the next one.

- **To ignore this instance and find the next,** press F4, NEXT, and the system will find the next instance, if there is one.

- **To stop the cancel and replace operation,** press F5, EXIT.

h. Press PREV.

14. To renumber positions

a. Move the cursor to the line number of any instruction.

b. Press NEXT, >, until F5, [EDCMD], is displayed. The function key labels for F1 through F4 might vary depending on your application.

c. Press F5, [EDCMD].

d. Select 6, Renumber.

e. Renumber the positions:

- **If you do not want to renumber positions** press F5, NO.

- **To renumber positions** press F4, YES.

15. When you are finished, turn the teach pendant ON/OFF switch to OFF and release the DEADMAN switch.

Note To test the program, refer to your application-specific *Setup and Operations Manual*.

4.3.4 Modifying a Program in the Background (Background Editing)

Background editing is used to modify a program when the teach pendant is off. This can be used to edit a program while another program is running.

If the system variable $BACKGROUND is FALSE, the teach pendant must remain on during programming. If the variable is TRUE, you do not need to turn on the teach pendant during programming.

During background editing, you **can**

- Add new program instructions.
- Add new motion instructions.

  The position recorded will be the current position of the robot.

  — If the robot is currently executing a motion instruction in another program, the robot position at the time you add the motion instruction will be the recorded position.

  — If the robot is not executing a motion instruction in another program, the current robot position will be the recorded position.

- Modify existing program instructions.

During background editing, you **cannot** move the robot. You cannot move the robot unless the teach pendant is enabled.

If you add motion instructions during background program editing, you must remember to touch up the positions using TOUCHUP before you run the program.

For more information about the system variables related to background editing, refer to the *Software Reference Manual* . For more information on writing and modifying programs, refer to your application-specific *Setup and Operations Manual*.

Use Procedure 4-5 to modify a program in the background.

Procedure 4-5 Modifying a Program in the Background

Conditions

- All personnel and unnecessary equipment are out of the workcell.
- The program has been created and all detail information has been set correctly. (Procedure 4-2)

Steps

1. Set $BACKGROUND to TRUE, if necessary.
   a. Press MENU.
   b. Select SYSTEM.
   c. Press F1, [TYPE].
   d. Select Variables.
   e. Move the cursor to $BACKGROUND.
      - If the value is TRUE, go to Step 2 .
      - If the value is FALSE, go to Step 1f .
   f. Press F4, TRUE.
2. Press SELECT.
3. Display the appropriate list of programs:
   a. Press F1, [TYPE].
   b. Select the list you want:
      - **All** displays all programs.
      - **TP Programs** displays all teach pendant programs.

- **KAREL Progs** displays all KAREL programs.
- **Macro** displays all macro programs.

4. Move the cursor to the name of the program you want to modify.
5. Press ENTER.
6. Without enabling the teach pendant, add an instruction to the program.
7. You will see the first confirmation. Move the cursor to YES and press ENTER.
8. You will see the second confirmation. Press ENTER. "<<BACKGROUND>" will be displayed at the beginning of the program.
9. Modify the program. Refer to Procedure 4-4 .
10. **When you are finished writing in the background,** end the background editing session:
    a. Press NEXT, >.
    b. Press F5, [EDCMD].
    c. Select End_edit. "<<BACKGROUND>" will no longer be displayed at the beginning of the program.

4.3.5 Re-synchronizing the Robot and Conveyor

This section guides you through the robot/conveyor re-synchronization process. This process should be followed when you are prompted that the part trigger value is invalid. This process establishes the position of the part along the conveyor (relative to the nominal tracking frame), so that the robot tracking positions and motions can be touched-up or executed.

This section also contains a procedure to follow if you encounter errors while re-synchronizing the robot and conveyor.

Procedure 4-6 Re-synchronizing the Robot and Conveyor

Conditions

- Re-synchronization must be performed within the user programming environment, during product operation. This is typically accomplished using the line tracking TPE LINECOUNT and SETTRIG instructions.
- You are modifying a program where the part trigger value is invalid.

  Note The re-synchronization procedure is automatically entered any time you enter a teach pendant program for editing and your part trigger value is invalid. You will see a screen similar to the following.

```
[END]
     ------------------------------------
     |                                  |
     |  The current part detect         |
     |  trigger value MAY NOT           |
     |  be valid.                       |
     |                                  |
     |  Resynchronize the               |
     |  tracking system?                |
     |                                  |
     |        NO           YES          |
     |                                  |
     ------------------------------------
```

1. Follow the instructions on the screen above by selecting YES or NO

   - **Select NO** if you are sure that the trigger value currently stored in the system is valid.

   OR

   - **Select YES** if you are uncertain that the trigger value currently stored in the system is valid.

   If tracking simulation is currently enabled at the time you enter a program for edit, you will see a screen similar to the following.

```
[END]
     ------------------------------------
     |                                  |
     |  Move a new part past the        |
     |  part detect switch and          |
     |  into the robot workspace.       |
     |                                  |
     |                                  |
     |                                  |
     |                                  |
     |                OK                |
     |                                  |
     ------------------------------------
```

2. Follow the instructions on the above screen, by moving the conveyor with the part on it
3. Select OK. You will see a screen similar to the following.

```
[END]
     ------------------------------------
     |                                  |
     |  Stop the part when it is        |
     |  at a convenient location        |
     |  in the robot workspace.         |
     |                                  |
     |  Press <ENTER> when done.        |
     |                                  |
     |                                  |
     |                                  |
     |                                  |
     ------------------------------------
```

4. Follow the instructions on the above screen, by stopping the conveyor when the part is at a convenient location.
5. Press ENTER. You will see a screen similar to the following.

```
EDIT
 [END]
[INST]     STAT              TOUCHUP    [EDCMD]
```

Procedure 4-7 Possible Re-synchronization Errors

This procedure contains errors that could occur when you are re-synchronizing the robot and conveyor.

Note If re-synchronization fails for any reason, you will receive an error message. You will then be prompted to press ENTER. The program editing session will abort and you will be returned to the Select screen. This will prevent the recording of invalid positions. Any of the errors listed in this procedure will cause this.

```
[END]
     ------------------------------------
     |                                  |
     |  Tracking simulation is          |
     |  currently enabled.              |
     |                                  |
     |  Press <ENTER> to                |
     |  simulate a new part             |
     |  detect trigger value.           |
     |                                  |
     |              OK                  |
     |                                  |
     ------------------------------------
```

1. Follow the instructions on the screen above by pressing ENTER to simulate a new part detect trigger value, since the actual conveyor is not being used.

> **Warning**
>
> **Program positions should never be taught when using a simulated conveyor. Otherwise, you could injure personnel or damage equipment.**

- Several items are checked internally during the re-synchronization procedure.
  - — **If all of these checks are OK,** the teach pendant program will be displayed.
  - — **If any of these checks fail,** you will be prompted accordingly by one of the following steps.
- **If the conveyor was running in the reverse direction** when the trigger was detected, you will see a screen similar to the following.

```
[END]
     ------------------------------------
     |                                  |
     |  WARNING: A trigger was just     |
     |  generated while conveyor        |
     |  was moving BACKWARDS!           |
     |                                  |
     |  Exit the program and            |
     |  re-enter it to re-sync!         |
     |                                  |
     |              OK                  |
     |                                  |
     ------------------------------------
```

The above can occur during programming, if you had backed up the conveyor to perform the synchronization.

- **If the encoder associated with the specified line tracking schedule is not enabled** you will see a screen similar to the following.

```
[END]
     ------------------------------------
     |                                  |
     |  WARNING: The selected           |
     |  encoder is currently            |
     |  turned OFF!                     |
     |                                  |
     |  Enable the encoder then         |
     |  return and re-sync!             |
     |                                  |
     |                OK                |
     |                                  |
     ------------------------------------
```

The above is important because the encoder count values are used during position recording and playback.

- **If one or more of the parameters in the specified line tracking schedule are not set properly** (for example, Encoder Number, Scale Factor, Digital Input Number), you will see a screen similar to the following.

```
[END]
     ------------------------------------
     |                                  |
     |  WARNING: Selected tracking      |
     |  schedule has not been           |
     |  properly initialized.           |
     |                                  |
     |  Initialize the schedule         |
     |  then return and re-sync!        |
     |                                  |
     |                OK                |
     |                                  |
     ------------------------------------
```

Part Detect Not Found

- **If a part detect is not encountered within a certain time period** (approximately two minutes), you will see a screen similar to the following.

```
[END]
     -----------------------------------
     |                                 |
     |  PART DETECT TIMEOUT!!!         |
     |  A part detect was not          |
     |  observed within the time-      |
     |  out period (120 seconds).      |
     |  Exit the program and           |
     |  re-enter it to re-sync!        |
     |                                 |
     |               OK                |
     |                                 |
     -----------------------------------
```

The above error can occur due to a stopped or slow-moving conveyor, or because the digital input is not functioning properly. In such cases, this time-out prevents the system from waiting indefinitely for the part detect trigger to occur.

4.3.6 Set Up and Use Predefined Positions in a Program

Setting up predefined positions, such as home or repair involves writing a program that contains motion instructions that move to the predefined position. The program names for home and repair are predefined.

Using predefined positions in a program involves adding CALL PROGRAM instructions that call the program in which you set up the predefined position.

Procedure 4-8 describes how to set up predefined positions. Procedure 4-9 describes how to use predefined positions in a program.

Procedure 4-8 Setting Up Predefined Positions

Conditions

- All personnel and unnecessary equipment are out of the workcell.

Steps

1. Press SELECT.
2. Select the predefined program name you want to use. For example, if you want to build the program for the home position, select the program name HOME.PR.
3. **When you are finished selecting the program name,** press ENTER.

4. Press the appropriate function keys to enter the comment.
5. **When you are finished building the comment,** press ENTER.
6. **If you want to specify a motion group,** press F5, MORE. Press the function key that corresponds to the motion group you want. When you are finished, press ENTER.
7. **If you want to modify the program name or comment** press F2, NO. Select 4, Options to modify the program name or comment Press ENTER when you are finished modifying the program name or comment.

   - **OVRWRT** - Replaces existing characters with ones you enter.
   - **INSERT** - Adds new characters to existing characters, at the current cursor position.
   - **CLEAR** - Removes the entire program name or comment from the field.

   Note If, at any time, you want to return to the first SELECT menu, press PREV until this menu is displayed.

8. Add motion instructions to the program to move the robot to the desired position. Keep motion speed slow to ensure the safety of personnel and equipment any time the position is reached.

Procedure 4-9 Using Predefined Positions in a Program

Conditions

- The predefined position has been set up. (Refer to Procedure 4-8 .)

Steps

1. Press SELECT.
2. Select the program in which you want to use the predefined position and press ENTER.
3. Position the cursor on the line before which you want to move to the predefined position.
4. Insert one line for the instruction. Refer to Section 4.3.3 .
5. Press NEXT until F1, [INST] is displayed.
6. Press F1, [INST].
7. Select 6, CALL.
8. Select 1, CALL program.
9. Select the program for the predefined position you want to use:

   - **For the home position,** select HOME.
   - **For the bypass position,** select BYPASS.
   - **For the first special position,** select SPECIAL1.
   - **For the second special position,** select SPECIAL2.

4.4 RUNNING A TRACKING PROGRAM IN T1 MODE

When a tracking program is being written or modified, and the mode selector switch is set to T1 mode, the behavior of the robot is as follows:

- If the conveyor is not moving, and a tracking program is executed at 100% override, the robot’s speed will be the program speed if the program speed is below the T1 mode safe speed . T1 mode safe speed is defined as 250 mm/sec for the tool center point (TCP) and 10% of maximum joint speeds. If the program speed is above the T1 mode safe speed, robot motion will be executed at the T1 mode safe speed. With lower overrides, the robot speed is reduced proportionally according to the override setting.
- If the conveyor is moving, and you attempt to execute a tracking program, the robot will not move. Instead, the error LNTK-041 "Encoder is moved in T1 mode" will be displayed.
- If a tracking program is currently being executed at any override speed, and the conveyor begins to move, the robot motion will immediately stop; and the error LNTK-041 "Encoder is moved in T1 mode" will be displayed.

4.5 TRACKING INSTRUCTIONS

Tracking instructions are used to assist in running a tracking program.

Note Any values set by a tracking instruction in a running program will override all settings performed in tracking and encoder setup.

DEFENC

The DEFENC instruction defines the current tracking encoder number. It copies the contents of the specified $LNSNRSCH structure, into the specified $ENC_STAT structure, to configure the encoder parameters.

Figure 4–2. DEFENC

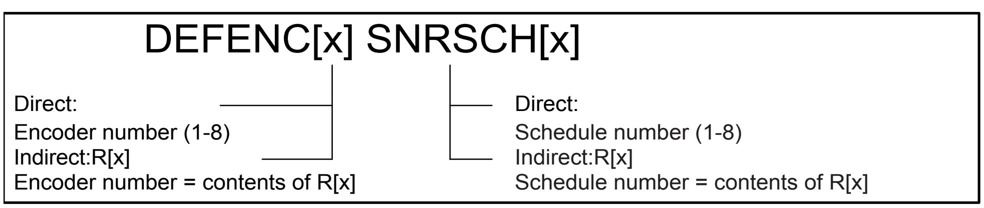
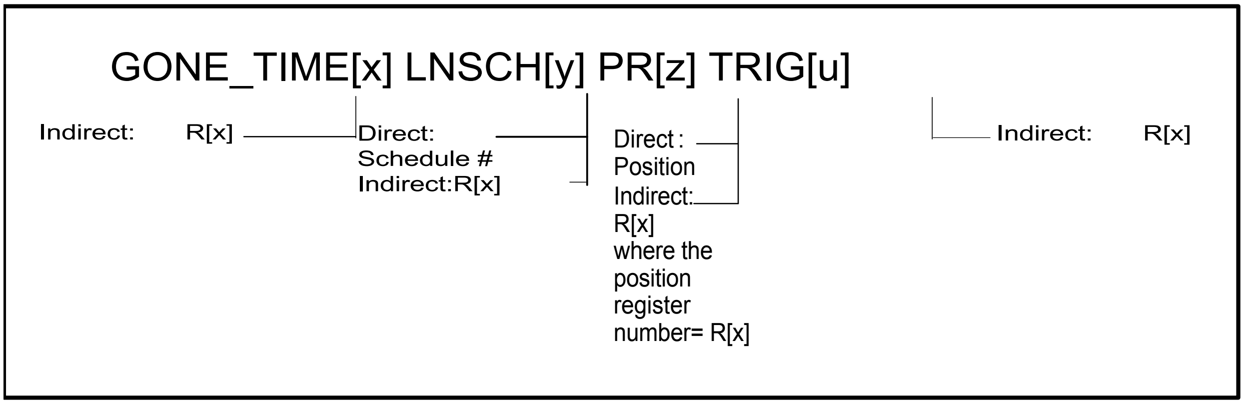


Figure 4–3. GONE_TIME



GONE_TIME

When the GONE_TIME instruction is executed, register x returns the number of seconds before which the position in position register z will be exiting out of the boundary specified in line tracking schedule y with the trigger value stored in register u. When this instruction is called, the conveyor speed at the time will be used for the calculation. If the conveyor speeds up afterward, GONE_TIME might not be accurately estimated. When the conveyor is stopped, GONE_TIME will return a large value instead of an infinite value.

This instruction can be used to monitor whether or not the part will be out of bounds when it is to be picked up. For example,

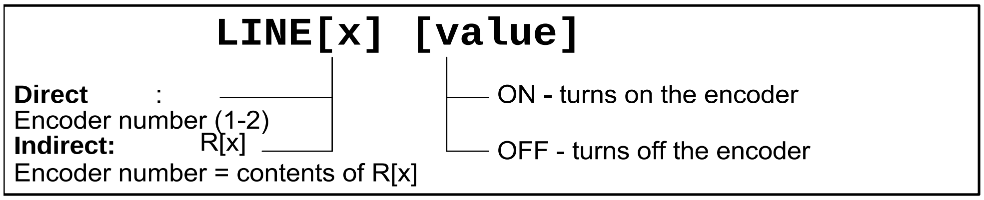
```
GONE_TIME[10] LNSCH[1] PR[1] TRIG[20]
IF R[10]<R[12] JUMP LBL[2]
CALL PICK_PART
LBL[2]:
```

When Register 12 stores the average time for the pick_part program. The trigger value for the part is stored in register 20. This can also be used to determine whether or not the main program has time to pre-rotate the robot tool while waiting for the part to be in the window (if this is a VISI-Track application). That way no time needs to be used to pre-rotate the tool.

LINE

The LINE enable instruction enables the encoder for tracking.

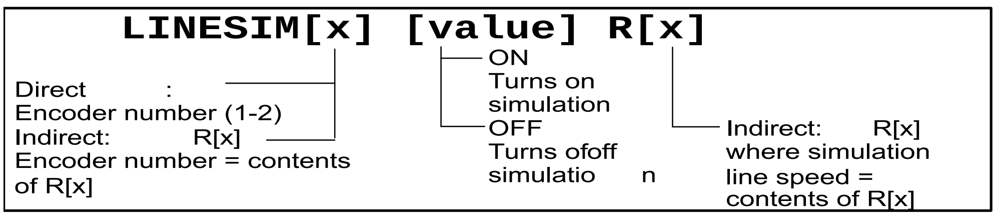
Figure 4–4. LINE Instruction



LINESIM

The LINESIM instruction sets up and enables encoder simulation. The line (encoder) must be enabled to simulate tracking using the LINESIM instruction. The simulation line speed used here is in units of encoder counts per encoder update.

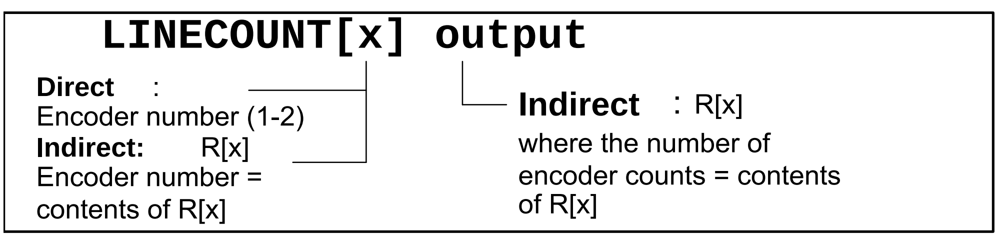
Figure 4–5. Linesim



LINECOUNT

The LINECOUNT instruction reports the current tracking encoder count. This instruction must be used immediately after detecting a part trigger, to record the position of the conveyor.

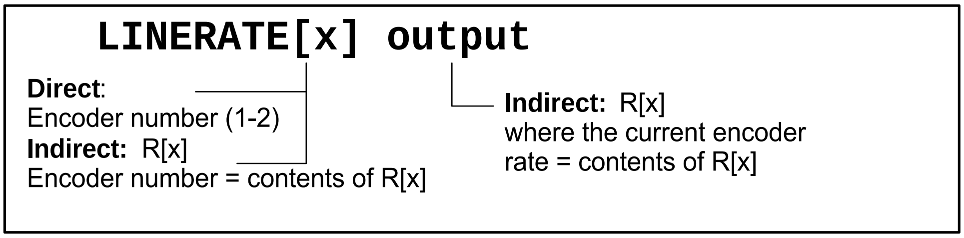
Figure 4–6. LINECOUNT



LINERATE

The LINERATE instruction reports the tracking encoder rate, in units of encoder counts per encoder update.

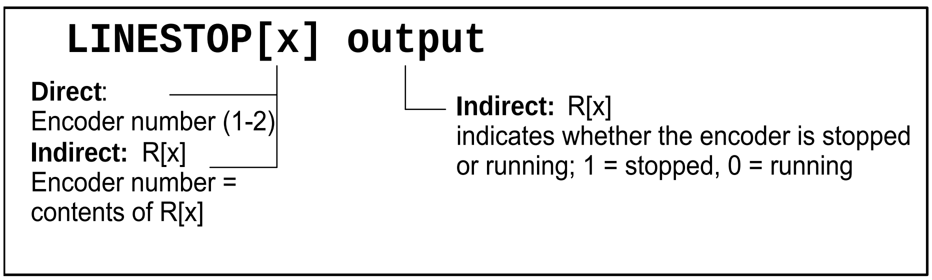
Figure 4–7. LINERATE



LINESTOP

The LINESTOP instruction reports the tracking encoder stopped status, based on the current line rate and encoder stop threshold.

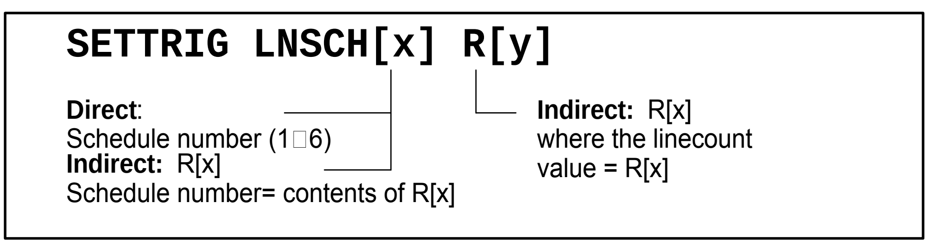
Figure 4–8. LINESTOP



SETTRIG

The SETTRIG instruction sets the tracking schedule trigger value. Refer to the “Line Tracking Setup” chapter in the *LineTracking Setup and Operations Manual* for more information on schedules. The LINECOUNT value is typically stored in the register through use of the LINECOUNT instruction defined above.

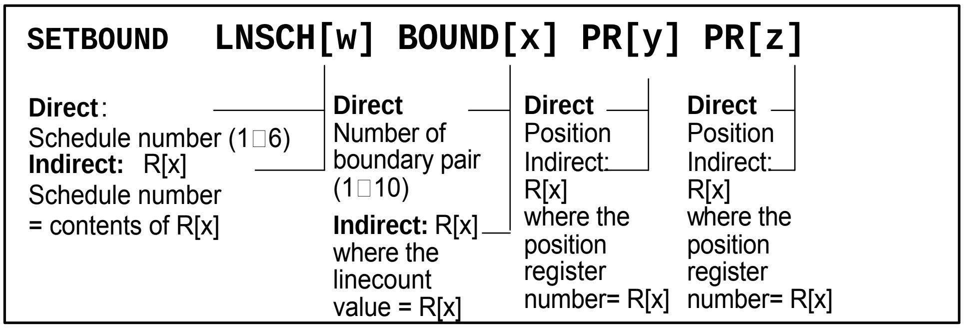
Figure 4–9. SETTRIG



SETBOUND

The SETBOUND instruction sets the tracking schedule boundary values, based on the WORLD frame positions stored in the two position registers. Refer to the “Line Tracking Setup” chapter in the *LineTracking Setup and Operations Manual* for more information on schedules.

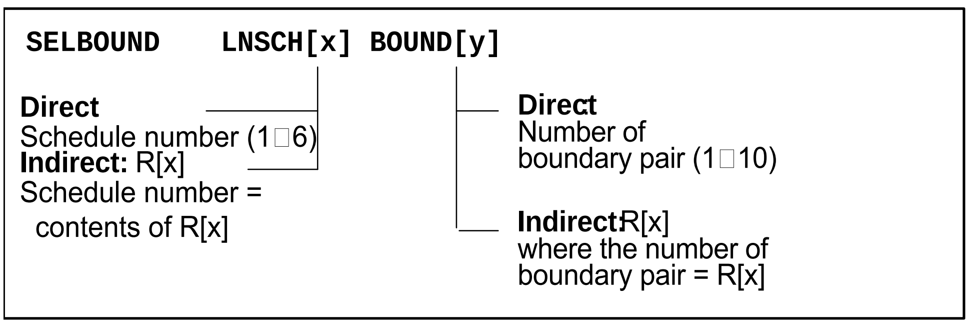
Figure 4–10. SETBOUND



SELBOUND

The SELBOUND instruction selects the tracking schedule boundary pair. Refer to the “Line Tracking Setup” chapter in the *LineTracking Setup and Operations Manual* for more information on schedules.

Figure 4–11. SELBOUND



After the tracking schedules have been set up, they can be used in a job or process program. Each process program uses a specific tracking schedule for the entire program. The schedule number is selected when the program is created or in the program DETAIL screen. The SELBOUND instruction is then used in the job program to determine the specific boundary used. See the following screen for an example.

```
JOB10
1:SELBOUND LNSCH[1] BOUND[1]
2:CALL PROC1001
3:SELBOUND LNSCH[2] BOUND[1]
4:CALL PROC1002
5:SELBOUND LNSCH[1] BOUND[2]
6:CALL PROC1003
```

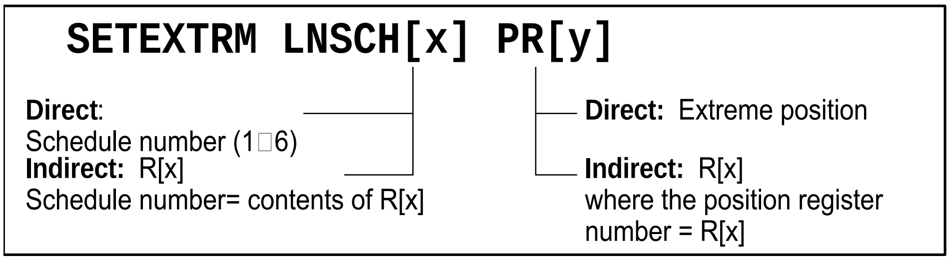
In this example, PROC1002 must have the line tracking schedule number set to 2 and the other programs have a schedule number set to 1.

SETEXTRM

The SETEXTRM instruction sets the tracking schedule extreme position, based on the nominal tracking frame position stored in the position register.

Note Extreme position checking is not fully supported for stand-alone tracking systems.

Figure 4–12. SETEXTRM



STOP_TRACKING

The STOP_TRACKING instruction is used inside a tracking program to end tracking motion temporarily. The robot will remain stopped until the program execution reaches the next tracking motion and the destination of that motion enters the boundary.

For example:

L P[1] 1000mm/sec CNT100

WAIT DI[10] = ON

L P[2] 1000mm/sec CNT100

In the above program, once the robot reaches P[1] it will continue to follow P[1] as it moves with the conveyor until the WAIT condition is satisfied and P[2] enters the boundary. After P[2] enters the boundary, the robot will begin moving to P[2]. If this program is modified as follows:

L P[1] 1000mm/sec CNT100

STOP_TRACKING

WAIT DI[10] = ON

L P[2] 1000mm/sec CNT100

In this version of the program, once the robot reaches P[1] it will stop and remained stopped. The robot will not begin moving again until after the WAIT condition is satisfied and P[2] enters the boundary. Only then will the robot begin moving towards P[2].

ACCUTRIG LNSCH

The ACCUTRIG LNSCH instruction activates an interrupt routine to set a system tick when I/O is triggered at plus edge (off to on). The LINECOUNT instruction uses this system tick to retrieve the encoder count at the system tick.

The ACCUTRIG LNSCH instruction should be used in the program before the program waits for the digital input. For example,

ACCUTRIG LNSCH

WAIT DI[1]=OFF

WAIT DI[1]=ON

LINECOUNT[1] R[1]

SETTRIG LNSCH[1] R[1]

In order to use the ACCUTRIG instruction, the following system variable must be set:

```
$LNCFG.$slc_pt_trig=TRUE
```

You must turn the controller off then on again for this variable to take effect. If ACCUTRIG is not being used, then this variable should be set to FALSE.

Figure 4–13. ACCUTRIG LNSCH

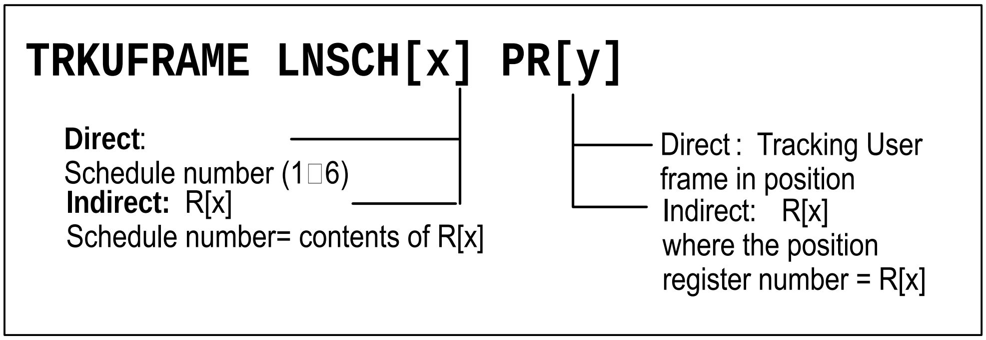
```
ACCUTRIG LNSCH[x]
```

Direct:
Schedule number (1–6)
Indirect: R[x]
Schedule number= contents of R[x]

TRKUFRAME

The TRKUFRAME instruction sets the tracking user frame in the schedule to the value in the position register.

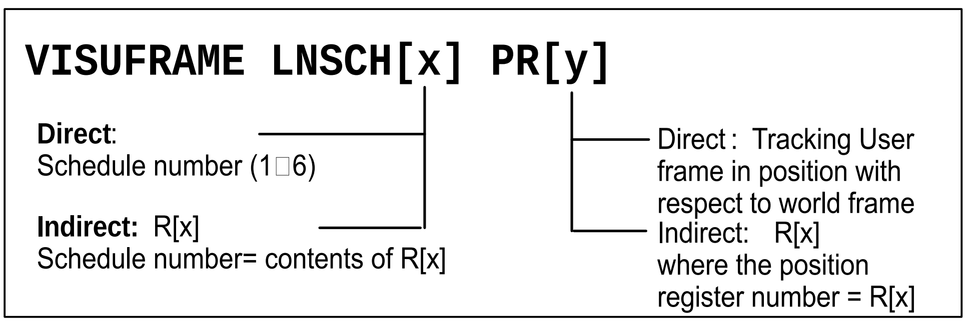
Figure 4–14. TRKUFRAME



VISUFRAME

The VISUFRAME instruction sets the tracking user frame according to the value in the position register. The position value in the position register is the tracking user frame with respect to the robot world frame.

Figure 4–15. VISUFRAME



Chapter 5

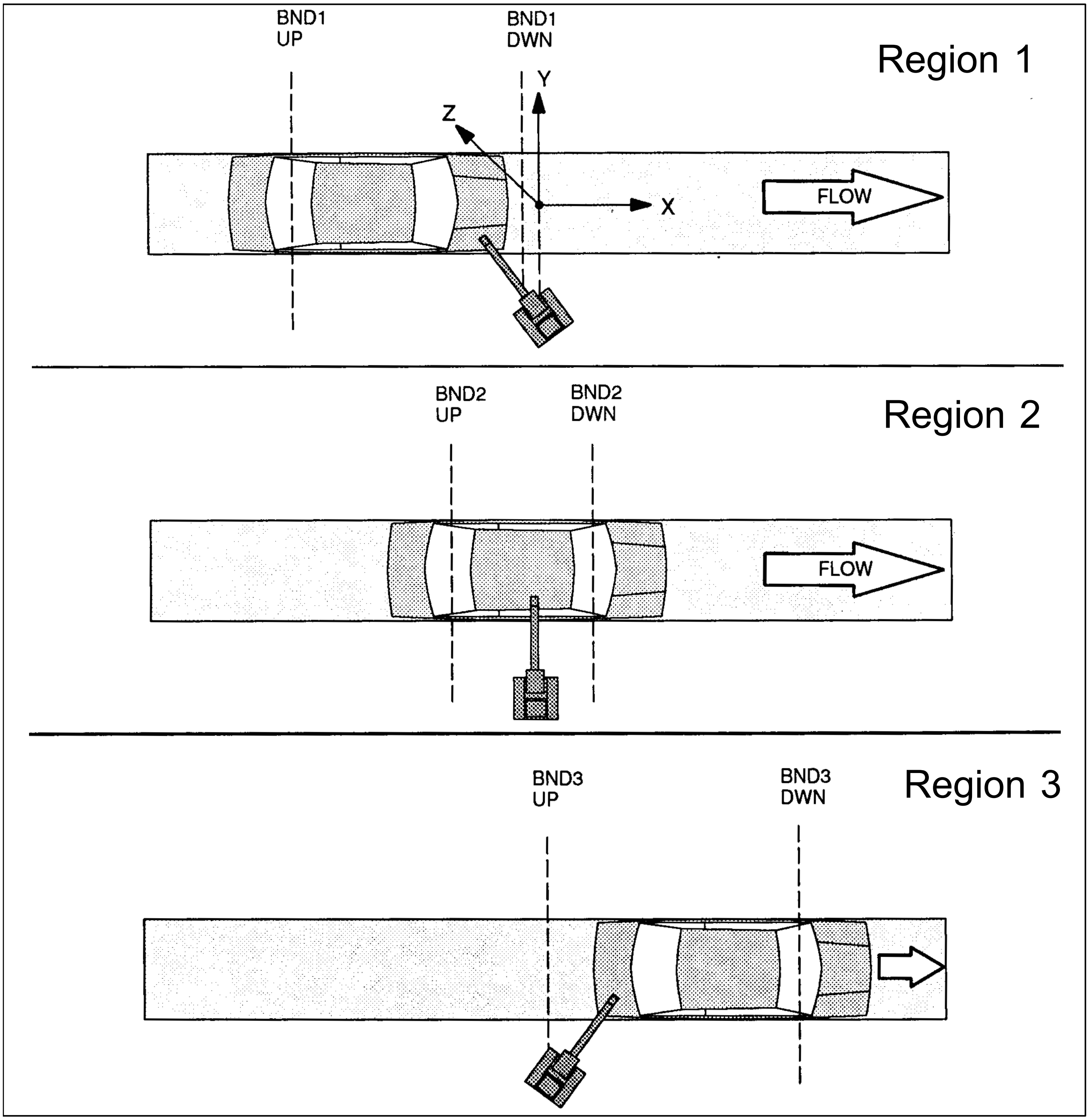
ADVANCED TECHNIQUES

Contents

5.1 MULTIPLE BOUNDARY POSITIONS EXAMPLE

This example examines the task of painting a car body. The task will be broken down into three *zones* or *windows* within the robot workspace. In this context a boundary *set* or *pair* describes the edges of each workspace zone. Refer to Figure 5–1 .

Figure 5–1. Tracking Boundary Positions



In this example the car will be painted in three regions:

- The front hood

- The top
- The back deck

In Figure 5–1 , the regions labeled 1, 2, and 3, are the boundaries of three work zones. Here UP is the up-stream boundary (corresponding to $LNSCH[ ].$boundl[n]) and DWN is the down-stream boundary ($LNSCH[ ].$bound2[n]) where up and down stream refer to the direction of conveyor motion in terms of conveyor flow.

The first region of the car is painted using boundary set 1, the second using boundary set 2, the third using boundary set 3. In each case the line track system will not issue robot motion until the current position is within the selected zone. This ensures that the robot will always be able to reach the program position, even though the position is moving to track the conveyor.

The teach pendant program in Job that Calls Programs to Paint a Car is an example of a line track job that calls the programs to paint the regions of the car shown in Figure 5–1 .

Job that Calls Programs to Paint a Car

```
/PROG  PAINT CAR
/MN
  1: LINE[1] ON ;
  2: LBL[1] ;
  3: ! call rest pos. (non-tracking)
  4: CALL HOME ;
  5: ! wait for part detect
  6: WAIT DI[32]=ON ;
  7: LINECOUNT[1] R[1] ;
  8: SETTRIG LNSCH[1] R[1] ;
  9: ! select zone 1
 10: SELBOUND LNSCH[1] BOUND[1] ;
 11: ! paint front (tracking)
 12: CALL CAR_FRNT ;
 13: ! select zone 2
 14: SELBOUND LNSCH[1] BOUND[2] ;
 15: ! paint top (tracking)
 16: CALL CAR_TOP ;
 17: ! select zone 3
 18: SELBOUND LNSCH[1] BOUND[3] ;
 19: ! paint back (tracking)
 20: CALL CAR_BACK ;
 21: JMP LBL[1] ;
/POS
/END
```

In this example, the issue of continuous tracking must also be taken into consideration. The tracking motion programs CAR_FRNT and CAR_TOP should have their associated program header data values set for CONTINUOUS TRACKING = TRUE, while CAR_BACK should have CONTINUOUS TRACKING = FALSE.

This allows the robot to continue to track the conveyor between the various tracking motion programs, but will stop the robot from tracking the conveyor upon completion of the last tracking program, before it returns to the rest position.

5.2 TRACKING PART QUEUES

A part queue routine or program task might be required to monitor the part detect switch, in order to set the part trigger value properly prior to issuing robot motion.

5.2.1 Using One Task

For example, a process line could include a conveyor which moves at an average speed of 50mm/sec and carries parts at a spacing of 2 meters. This would mean that parts pass through a particular station along that line at intervals of 40 seconds:

40 sec * 50mm/sec = 2000mm or 2 meters.

If, in this example, the part detect switch is located 1 meter up-stream of the IN-BOUND window boundary, it will take 20 seconds for the part to travel from the part detect switch to the point where the robot can begin to process the part. Also, the robot only requires 15 seconds to complete its process.

In this example the robot has enough time to complete processing and then return to monitor the part detect switch and wait for the next part to arrive. The time sequence is shown below in Table 5–1 .

Table 5–1. Line Tracking Conveyor Timing

Time	Action
0	Part triggers the part detect switch.
20	Part enters the robot work window and the robot begins processing.
35	The robot completes processing and returns to monitor the part detect switch.

Since the parts are spaced at 40 second intervals, a single task can be used to monitor the part detect switch, and then issue the robot motion associated with the processing. It will still have 5 seconds to spare while it waits for the next part to arrive. As each part passes the part detect switch, the task records the conveyor position using the LINECOUNT instruction and copies this directly into the appropriate line track schedule variable using the SETTRIG instruction.

5.2.2 Using Two Tasks

If you have twice as many parts on the conveyor (compared to the previous example), and they are spaced 1 meter apart, and arrive at 20 second intervals, using the previous time sequence, a single task would not have enough time to monitor each part and issue the robot motion commands for processing. This is because the entire sequence requires at least 35 seconds per part. However, since the robot processing only requires 15 seconds, it is still possible for the robot to process all of the parts.

In this case, a second task is required so that the tasks of monitoring the part detect switch and issuing robot motion commands for processing are separate. The task which monitors the part detect switch is now required to record the conveyor position (again using the LINECOUNT command), but can no longer directly copy this information into the appropriate line track schedule variable. Instead, it must temporarily store this data until the robot has completed processing the previous part. Then the trigger value associated with the next part can be safely copied into the line track schedule variable.

Handshaking to Monitor the Part Detect Switch

Since this example application requires two tasks to monitor the part detect switch separately and issue robot motion commands for the robot processing, *handshaking* is used to coordinate these two efforts. Handshaking ensures that the trigger value is always properly updated prior to beginning the robot processing.

There are several ways to accomplish handshaking. In this case, the robot processing task informs the part detect monitoring task when it is ready to receive the next part trigger value. The part detect monitoring task then informs the robot when the new part trigger value is available.

Monitoring the Part Detect Switch with Handshaking shows an example of a main program which initializes the system, runs a part detect task, and calls all motion programs.

Register Indexing to Monitor the Part Detect Switch

In another case, register indexing is used by both tasks to keep track of multiple part trigger values. In this case, the part trigger monitoring task updates the value of one register to be the index of the register into which it stores the next part's trigger value.

The robot process task then begins each process by first reading the value from the index register. It then copies the trigger value from the indicated register (again specified by the value within the index register) into the line track schedule number. In this case, no handshaking is required as long as the size of the part queue (the number of program registers needed to store consecutive part trigger values) exceeds the maximum number of parts allowed between the part detect switch and the robot OUT BOUND window at any given time.

Monitoring the Part Detect Switch with Index Registering shows an example of the part detect task run from the *main* program of Monitoring the Part Detect Switch with Handshaking .

Note The group mask of the part detect task is set=0, since it will never control robot motion, but must run concurrently with robot motion tasks.

Monitoring the Part Detect Switch with Handshaking

```
/MN
1: LBL[1] ;
 2: R[3]=0  ;
 3: R[4]=0 ;
 4: ;
 5: !INIT TRIG VALUE REGS ;
 6: R[1]=10 ;
 7: LBL[2] ;
 8: R[R[1]]=0 ;
 9: R[1]=R[1]+1 ;
10: IF R[1]<=20, JMP LBL[2] ;
11: ;
12: !INIT INDEX REG ;
13: R[1]=10 ;
14: R[2]=10 ;
15: ;
16: !SET UP TRACKING ;
17: LINE[1] ON ;
18: ;
19: !RUN TRIG MON PROG ;
20: RUN SUBPRG1 ;
21: ;
22: !MAIN LOOP ;
23: LBL[10];
24: ;
25: !MOVE TO STATIONARY POS ;
26: CALL HOME ;
27: ;
28: !WAIT FOR NEW PART TRIGGER ;
29: WAIT R[R[2]]<>0 ;
30: SETTRIG LNSCH[1] R[R[2]] ;
31: ;
32: !CALL TRACKING PROGS ;
33: CALL SUB_L1 ;
34: CALL SUB_C1 ;
35: ;
36: !INC INDEX REG ;
37: R[2]=R[2]+1 ;
38: IF R[2]<=20, JMP LBL[15] ;
39: R[2]=10 ;
40: ;
41: LBL[15] ;
42: !GO TO MAIN LOOP START ;
43: JMP LBL[10] ;
/POS
/END
```

Monitoring the Part Detect Switch with Index Registering

```
/MN
1: LBL[20] ;
 2: !INIT REGS ;
 3: R[1]=10 ;
 4: ;
 5: !MAIN PART DETECT LOOP ;
 6: LBL[30] ;
 7: R[R[1]]=0 ;
 8: !WAIT FOR PART DETECT ;
 9: WAIT DI[1]=ON ;
10: LINECOUNT[1] R[3] ;
11: R[R[1]]=R[3] ;
12: R[3]=0 ;
13: R[4]=0 ;
14: ;
15: !INC INDEX REG ;
16: R[1]=R[1]+1 ;
17: IF R[1]>20, JMP LBL[20] ;
18: ;
19: JMP LBL[30] ;
/POS
/END
```

5.3 MULTIPLE CONVEYORS (DUAL LINE TRACKING)

Line tracking supports up to two conveyor tracking encoders. The typical tracking environment involves manipulating a part moving, through the workspace, on a single conveyor. However, if you have a situation that requires moving a part from one conveyor to another, this can also be handled using line tracking, but requires the following special considerations.

- Both encoder schedules must be properly set up, each for a separate encoder.
- Since only one encoder can be specified within the parameter of any one tracking schedule, a multiple conveyor system requires the use of more than one schedule. You must be careful to set up each schedule properly. This includes setting up individual nominal tracking frames, and scale factors, and attaching them to the appropriate encoder number.
- Since only one tracking schedule can be associated with each tracking program (via the program header data), a multiple encoder system requires using separate programs for motions associated with each conveyor.

- The above mentioned situation also requires that two separate part detect switches be monitored and processed accordingly, to ensure that each corresponding trigger value is properly set prior to issuing any associated robot tracking motions.

5.3.1 Dual Line Tracking Setup

A typical Dual Line Tracking application includes:

- Monitoring signals from both part detect switches, Monitor Program .
- Looking for parts simultaneously on both conveyors, Part Processing Program .
- Verifying the robot tracks positions on each conveyor correctly, CONV_1 and CONV_2 Verify Robot Tracking Programs .
- Switching back and forth between conveyors to process parts, CLEAR_1 and CLEAR_2 Switch Part Processing Programs .

The following implementation process should be run to test any dual line tracking application:

- Test Line 1 tracking independently, using the example programs in this section.
- Test Line 2 tracking independently, using the example programs in this section.
- Use CONV_1 and CONV_2 respectively as your tracking program.
- Run the MONITOR program by itself and check if the pulse counts are displayed in appropriate registers as parts pass by the trigger sensor.
- Run the MAIN program and check if the application runs properly.

5.3.2 Example Programs

The following programs show an example of typical Dual Line Tracking Applications.

Monitor Program

This program continuously monitors the trigger signals of both conveyers. This program allows up to 9 parts to be queued, by storing each part detect signal received in a register. When a part detect signal from a conveyer is received, the MONITOR program reads the pulse counts of the signal and stores it in one of nine registers (one for each conveyor) in order. For example, if last part was stored in R[10], next part will be stored in R[11] for processing. The part processing program (refer to Part Processing Program) will use these pulse count values for processing the respective parts.

Note The following conditions apply when you run this program:

- This is a concurrent task run from the main program.

- This program attribute should be set to NOPAUSE.
- The group mask of this program should be set to ‘*’.

Monitor Program

```
1: R[1]=10;                       --Line 1 uses R[10] - R[19]
2: R[R[1]]=0;                     --initialize to 0
3: R[2]=20;                       --Line 2 uses R[20] - R[29]
4: R[R[2]]=0;                     --initialize to 0
5: JMP LBL[30];
6: LBL[10];                       --continuous loop
7: R[1]=10;                       --Line 1 uses R[10] - R[19]
8: R[R[1]]=0;                     --initialize to 0
9: F[1]=OFF
10: JMP LBL[30];
11: LBL[20];
12: R[2]=20;                      --Line 2 uses R[20] - R[29]
13: R[R[2]]=0;                     --initialize to 0
14: F[2]=OFF
15: LBL[30];                       --looping for trigger signal
16: IF DIN[1]=ON LBL[100];
17: LBL[40];
18: IF DIN[2]=ON LBL[200];
19: JMP LBL[30];
20: LBL[100];
21: LINECOUNT[1] R[3];             --Line 1 cnts put into R[3]
22: R[R[1]]=R[3];                 --Line 1 cnt is saved
23: R[3]=0;
24: R[4]=0;                       --R[3], R[4] are used for Line1
25: F[1]=ON
26: R[1]=R[1]+1;                   --increment register for next  part
27: IF R[1]>19, JMP LBL[10];
28: R[R[1]]=0;                     --initialize next register to  0
29: F[1]=OFF
30: JMP LBL[40];
31: LBL[200];
32: LINECOUNT[2] R[5];             --Line 2 cnts put into R[5]
33: R[R[2]]=R[5];                  --Line 2 cnt is saved
34: R[5]=0;
35: R[6]=0;                        --R[5], R[6] are used for Line2
36: F[2]=ON
37: R[2]=R[2]+1;                   --increment register for next  part
38: IF R[2]>29, JMP LBL[20];
39: R[R[2]]=0;                     --initialize next register to  0
40: F[2]=OFF
41: JMP LBL[30];
/POS
```

```
/END
```

Part Processing Program

The program in Part Processing Program looks for a part in either conveyor 1 or conveyor 2. Parts are processed on first come, first served basis. This program does not alternate processing between conveyors. However, it does alternate between conveyors to look for parts.

Part Processing Program

```
1: LINE[1] ON;
2: LINE[2] ON;
3: CALL_PROG HOME; --HOME is a non-tracking program
4: LBL[10];
5: WAIT (F[1]) TIMEOUT, LBL[11] --TIMEOUT = $WAITTMOUT = 10 (which is 100 ms)
6: LBL[11]
7: IF R[R[1]]<>0, JMP LBL[100];
8: LBL[20];
9: WAIT (F[2]) TIMEOUT, LBL[21]
10: LBL[21]
11: IF R[R[2]]<>0, JMP LBL[200];
12: JMP LBL[10]; --loop until a part passes through
13: LBL[100];
14: SETTRIG LNSCH[1] R[R[1]];
15: SELBOUND LNSCH[1] BOUND[1];
16: CALL_PROG CONV_1; --CONV_1 is tracking program
17: CALL_PROG CLEAR_1; --CLEAR_1 is non-tracking program
18: JMP LBL[10];
19: LBL[200];
20: SETTRIG LNSCH[2] R[2];
21: SELBOUND LNSCH[2] BOUND[1];
22: CALL_PROG CONV_2; --CONV_2 is tracking program
23: CALL_PROG CLEAR_2; --CLEAR_2 is non-tracking program
24: JMP LBL[20];
/POS
/END
```

Verify Robot Tracking Program

The program in CONV_1 and CONV_2 Verify Robot Tracking Programs verifies the robot tracks the target position properly for 2 seconds. CONV_1 and CONV_2 programs must be identical, except for the positions in each that are taught at the respective conveyers.

Note Tracking must be enabled for this program.

CONV_1 and CONV_2 Verify Robot Tracking Programs

```
1: L P[1:NEAR PART] CNT75;
    2: L P[2:TARGET POS] FINE;
    3: DELAY 2;                         --delay is to test tracking
    4: L P[1:NEAR PART] CNT100;
   /POS
   /END
```

Switch Conveyor Part Processing

The program in CLEAR_1 and CLEAR_2 Switch Part Processing Programs switches robot part processing back and forth, between conveyor 1 and conveyor 2. The CLEAR_1 and CLEAR_2 programs must be non-tracking programs. In addition, CLEAR_1 and CLEAR_2 programs must be identical, except for the positions in each that are taught at the respective conveyors.

CLEAR_1 and CLEAR_2 Switch Part Processing Programs

```
    1: L P[1:SAFE POS] CNT75;
    2: L P[2:CLEAR CONV] FINE;
   /POS
   /END
```

Main Program

The program in Main Program Runs Monitor and Part Programs runs the MONITOR program (Monitor Program) concurrently with the PART program (Part Processing Program).

Main Program Runs Monitor and Part Programs

```
    1: RUN_PROG MONITOR
    2: CALL_PROG PART
   /POS
   /END
```

5.4 FINE TUNING HIGH SPEED ACCURACY

If you want to *fine tune* or make the robot move closer to the specified position during high speed line tracking, you can use one or both of the following methods:

- Fine tuning the static tune variable
- Fine tuning the dynamic tune variable

5.4.1 Static Tune Variable

The first method used to fine tune high speed line tracking accuracy is to adjust the static tune variable. This is done using the following method:

1. Write a tracking program. For example,

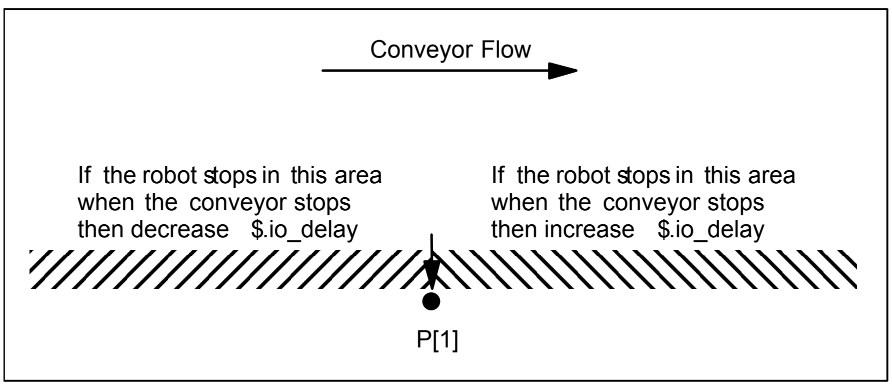
   L P[1] 500mm/sec FINE

   WAIT FOR 30.00(sec)

   Note Record P[1] past the part detect at a slow speed.

2. Run the tracking program at maximum production speed.
3. After the robot moves to the position (P[1]), stop the conveyor but keep the program running.
4. Verify that the robot is lined up at the correct position (P[1]). If the robot is not lined up with P[1], then modify $LNCFG_GRP.$io_delay as shown in Figure 5–2 .

Figure 5–2. Modifying $LNCFG_GRP.$io_delay



For example, if the robot is off by 3 mm and the conveyor speed is 500 mm/sec then,

$$\frac{3}{500} \sim 6\ \text{ms}$$

Therefore, adjust $io_delay by 6 ms.

5. When DI input type is different for different encoder, the io_delay for different encoder may be different. In this case use $ENC_IOD_ENB[encoder] and $ENC_IODELAY[encoder] instead of $LNCFG_GRP.$io_delay. Set $ENC_IOD_ENB[encoder] to TRUE and adjust $ENC_IODELAY[encoder].

6. Repeat Steps 2 through 4 until the robot lines up with P[1].

Note Although the variable $io_delay is part of $LNCFG_GRP[], it is only group based when the ACCUTRIG instruction is used. When the ACCUTRIG instruction is used, the $io_delay value will be taken from $LNCFG_GRP[g], where g is the group number from the tracking schedule. If ACCUTRIG is not used, then $LNCFG_GRP[1].$io_delay is the value that will be used.

Also, when the ACCUTRIG instruction is used, the amount of adjustment in $LNCFG_GRP[g].$io_delay is limited to 1 ITP in the negative direction, and 5 ITP in the positive direction, where ITP is the controller ITP time, and it is stored in the read-only system variable $SCR.$ITP_TIME. Larger adjustments will not be performed. If the ACCUTRIG instruction is not used, then there is no limitation on $LNCFG_GRP[1].$io_delay.

When DI input type is different for different encoder, the io_delay for different encoder may be different. In this case use $ENC_IOD_ENB[encoder] and $ENC_IODELAY[encoder] instead of $LNCFG_GRP.$io_delay.

5.4.2 Dynamic Tune Variable

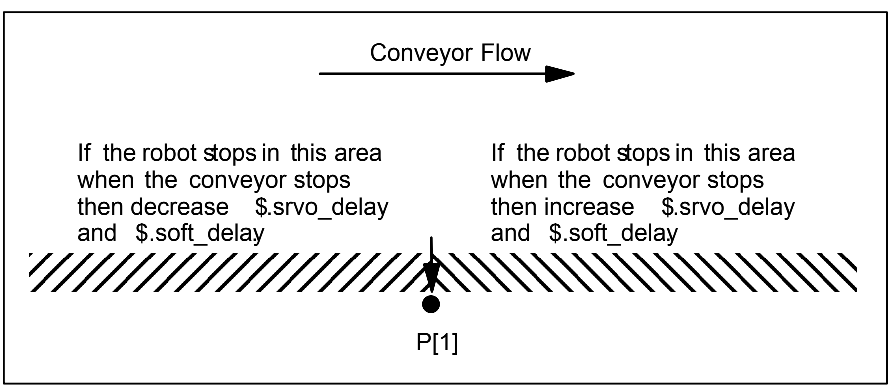
To modify the dynamic tune variables, you must first set up the static tune variable (Section 5.4.1). After you have set up the static tune variable, then you must modify the following two variables:

- $LNCFG_GRP. **$** srvo_delay
- $LNCFG_GRP. **$** soft_delay

These two variables work together. This is done using the following method:

1. Run the job at maximum production speed and **do not stop** the conveyor.
2. Verify that the robot is lined up at the correct position while the conveyor is moving and the program is running. If the robot is not lined up perfectly with P[1] then modify $LNCFG_GRP.$srvo_delay and $LNCFG_GRP.$soft_delay as shown in Figure 5–3 .

Figure 5–3. Modifying $LNCFG_GRP.$srvo_delay and $LNCFG_GRP.$soft_ delay



3. Observe the movement to the position while the WAIT is executed. If it is not accurate enough, repeat Steps 1 and 2.

5.5 TRACKING USER FRAME

5.5.1 Overview

A Tracking User frame is used to compensate for part location or orientation changes dynamically during production. To use this feature, you can use one of the following two instructions in your program:

- TRKUFRAME
- VISUFRAME

To use the Tracking User frame functions, in addition to setting up Line Tracking, you must also set up two other items.

- Set Use Tracking Uframe to YES (default is NO) in order for both TRAKUFRAME and VISUFRAME to be used.
- Set Vision Uframe Dist to a proper value (default is zero) in order to use the VISUFRAME instruction in Tracking Schedule Setup.

For more information about Tracking Setup, refer to the *Tracking Setup* section of this manual.

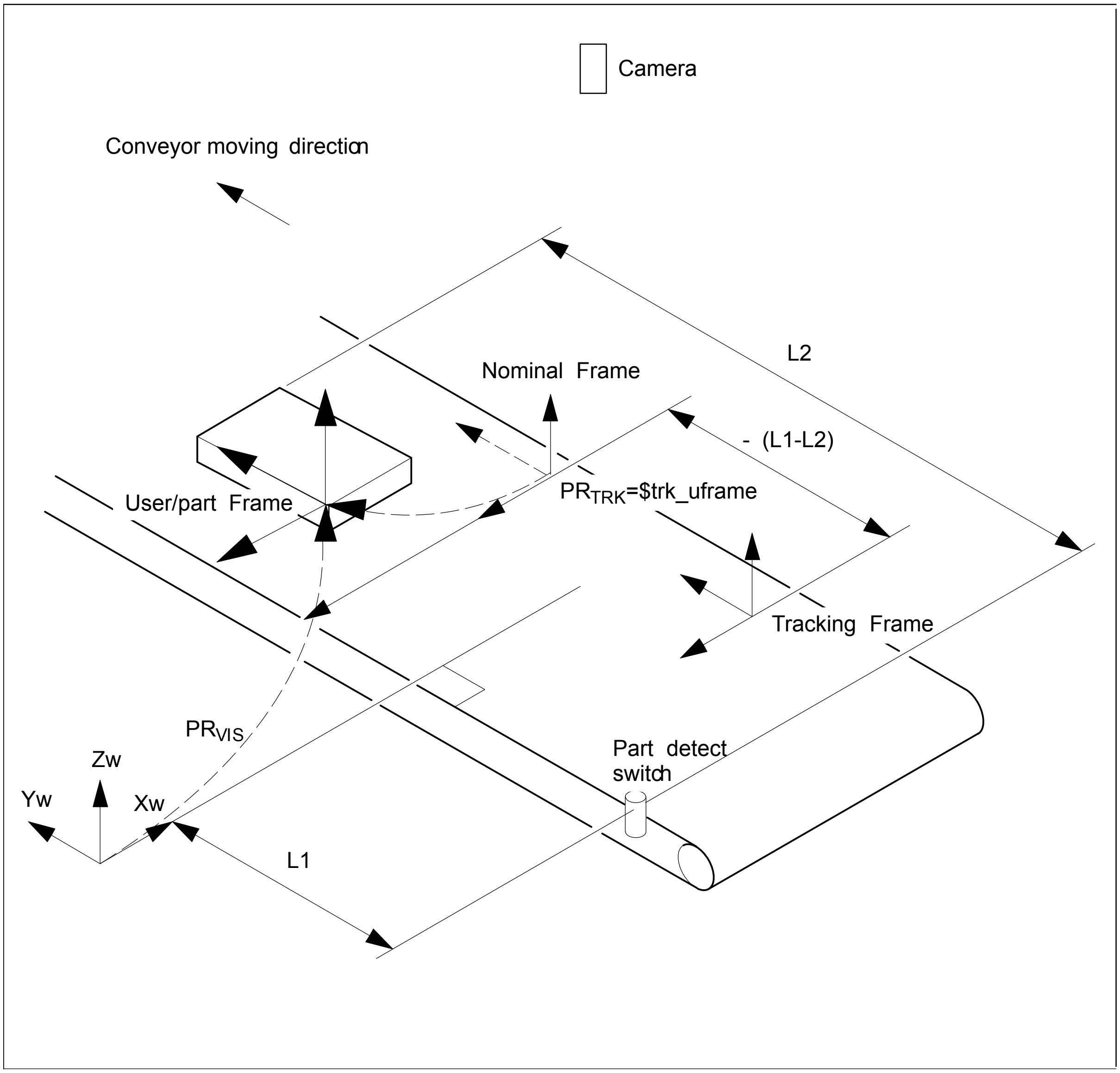
Note Only one of the two Tracking User frame instructions, TRKUFRAME or VISUFRAME, can be used in a single tracking schedule.

Note Circular Tracking does not support tracking uframe

5.5.2 Tracking Frame Terminology

In order to use the Tracking User frame functionality, you must understand Line Tracking frames and their relationship. See Figure 5–4 for a typical robot-conveyor setup and frame relationship. Line Tracking frames and their relationship. See a typical robot-conveyor setup and frame relationship.

Figure 5–4. A typical Tracking Frame Setup and Frame Relationship



The following terminology must be understood:

Tracking Frame - A nominal tracking frame defined in Tracking Schedule Setup. It is stationary during the line tracking motion. Refer to the *Line Tracking Setup and Operations Manual* for more information.

Nominal Frame - A runtime nominal tracking frame. It is parallel to the tracking frame and moves with the conveyor while tracking.

User/Part Frame - A frame on the tracked object (part) defined by the user.

World Frame - The standard robot world frame.

PR $_{TRK}$ [x]- A user/part frame with respect to the Nominal frame in position form, which is equal to $trk_uframe in the $LNSCH[i] system variable.

PR $_{VIS}$ [x] - A user/part frame with respect to the robot world frame in position form. It is used when a vision system is used.

L $_1$ - Part DetectDist.(mm) in the Line Tracking Schedule Setup.

L $_2$ - Travel distance in which the part is past the part detect switch. It can be calculated using the following formula:

$$L_2 = \frac{\text{Encoder Count - Trigger Value}}{\text{Enc Scale Factor}}$$

5.5.3 Calculate PR[] for TRKUFRAME

In general, you can set up the Line Tracking system any way you want, and PR[x] can be determined by analyzing the frame relationships. However, if you set up the conveyor parallel to the y axis of the robot world frame (see Figure 5–4), the calculation of PR[x] will be simple. Procedure 5-1 can be used to calculate the PR[x] value for TRKUFRAME for the tracking setup in which the conveyor is parallel to the y axis of the robot world frame.

Procedure 5-1 Calculate PR[x] for TRKUFRAME

Steps

1. Pass the object (part) through the part detect switch, then stop the conveyor at a position where the robot can reach the part easily.
2. Look at the Tracking Schedule Setup screen to observe the Encoder Count, Trigger Value, Enc Scale Factor, Part Detect Dist., and Tracking frame values.
3. Calculate*L $_2$ using the equation at the end of* Section 5.5.2 .
4. Calculate*L $_1$ - $_2$*
5. Add-*(L $_1$ -L $_2$* to the Y component of the origin of the Tracking frame to shift the Tracking frame by - *(L $_1$ - L $_2$)* along the Tracking frame x direction. The resulting frame is the Nominal frame at the current part location (note that conveyor is stopped now).
6. Choose a User frame on the object (part) and the move robot tool center point (TCP) to the origin of the frame to define the User frame's origin in robot world coordinates. Now both the Nominal frame and the User frame are in the world coordinates.
7. Subtract the nominal frame's corresponding component from the User frame's to get PR[x] for TRKUFRAME. User frame= $_u$ *and nominal frame=* $_n$.

$\{Y_u - Y_n, -(X_u - X_n), Z_{u-n}, P_u - P_n, -(W_u - W_n), R_{u-n}\}$

Note The orientation of the tracking frame is the same as rotating the world frame by 90 degrees around the z axis. In most cases, the differences in w, p, r between the tracking Uframe and the nominal frame are set to zero.

5.5.4 Calculate PR[] for VISUFRAME

When a vision system is used in a line tracking system, the VISUFRAME instruction can be used to perform the path compensation. For the VISUFRAME instruction, L_2 represents the part travel distance between the part detect switch and the part location where the snapshot is taken by the vision system.

The value is measured by the user and put into the Vision Uframe Dist. in the Tracking Schedule Setup. PR[x] represents the User/Part frame with respect to the robot world frame. The Vision system will obtain PR[x] automatically and pass it to the proper position register. See $PR_{VIS}[x]$ in Figure 5–4 . The line tracking softpart then converts PR_{VIS} into PR_{TRK} and stores it in the line tracking system variable $LNSCH[i].$trk_uframe.

5.5.5 Sample Tracking Uframe Program and Execution

A sample Tracking Uframe teach pendant program is shown in Sample Tracking Uframe Program .

Sample Tracking Uframe Program

```
1:Line [1] ON,
--turn on encoder

2:LBL[1];
3:CALL HOME1;
--home the robot

4:WAIT DI[27]=ON;
--wait for part detect switch is triggered

5:LINECOUNT[1] R[1];
--put the encoder count into the register

6:SETTRIG LNSCH[1] R[1];
--set the trigger count

7:SELBOUND LNSCH[1] COUNT[1]
--select a boundry set
```

```
8:TRKUFRAME LNSCH[1] PR[1];
--set the tracking uframe

9:CALL LNTK1;
--call a tracking program

10:JMP LBL[1]
--restart the process
```

Note The TRKUFRAME instruction in line 8 can be replaced by VISUFRAME. HOME1 and HOME2 are non-tracking programs. LNTK1 is a tracking program and is shown in Sample Tracking Program (LNTK1) .

Sample Tracking Program (LNTK1)

```
1:L P[1] 1000mm/sec FINE;          --move to above of the part
2:L P[2] 1000mm/sec FINE;          --move to P2
3:  WAIT 1.00(sec);                --wait for one second
4:L P[3] 1000mm/sec FINE;          --move to P3
5:L P[4] 1000mm/sec FINE;          --move to P4
6:L P[5] 1000mm/sec FINE;          --move to P5
7:L P[2] 1000mm/sec FINE;          --move to P2
```

5.5.6 Teaching and Executing the Tracking Uframe Program

After PR[x] is determined (Section 5.5.3 and Section 5.5.4), you can teach and execute the Tracking Uframe program as shown in Procedure 5-2 .

Procedure 5-2 Teach and Execute the Tracking Uframe Program

Steps

1. Type the PR[x] value into the proper position register. Run the main teach pendant program to execute TRKUFRAME or VISUFRAME with the corresponding LNSCH[i] and PR[x]. Abort the main program. Start the conveyor. Let the part pass the part detect switch, then stop the conveyor at the proper position so that the robot can reach the part easily.
2. Record the tracking positions in the line tracking program. Test run the line tracking program as in regular line tracking to make sure the taught path is followed. Refer to the "Testing a Program and Running Production" chapter in your application-specific *Setup and Operations Manual* . Obtain the part location offset by using a position sensor or vision system, and pass the offset to PR[x] before the play back tracking motion is started.

For example, when a VISUFRAME instruction is used, the vision system used in determining PR[x] before teaching will be used to find PR[x] for the play back as well. When a TRKUFRAME instruction is used, a one-dimensional position sensor, for example, can be used to detect the y direction shift of the part. The value can be passed to PR[x] and used as an offset.

3. Test run the program again to see the path compensation.

Example path compensations are illustrated in Figure 5–5 and Figure 5–6 . The dashed line represents the original path, and the solid line represents the path with compensation in the y and r directions respectively.

Note Only one of the two Tracking Uframe instructions, TRKUFRAME and VISUFRAM, can be used in one tracking schedule.

Figure 5–5. Path Compensation in the y Direction

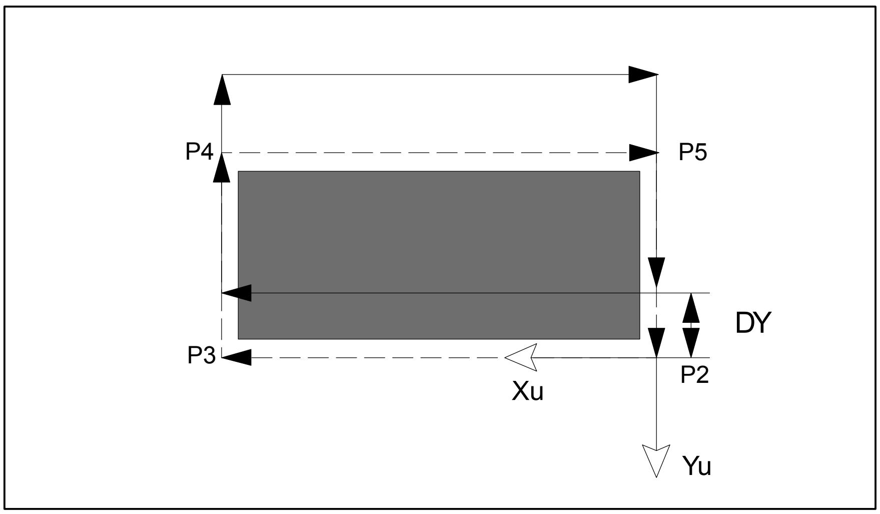
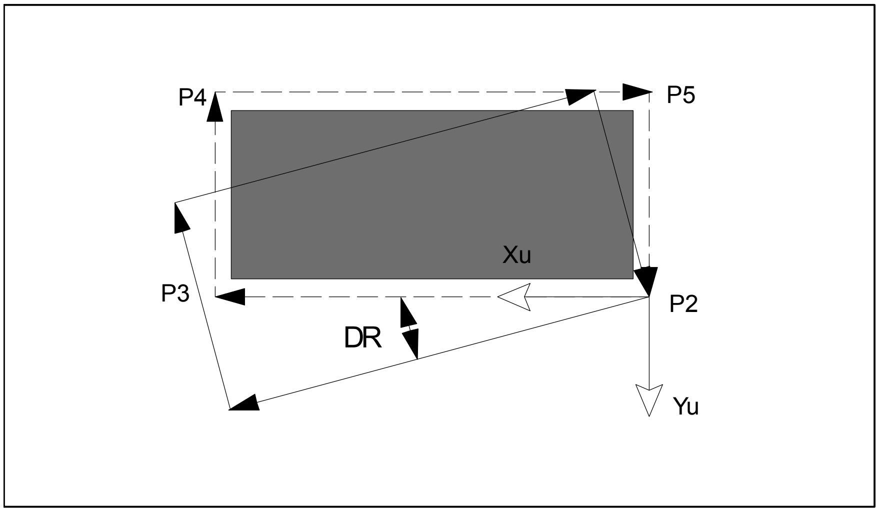


Figure 5–6. Path Compensation in the r rotation

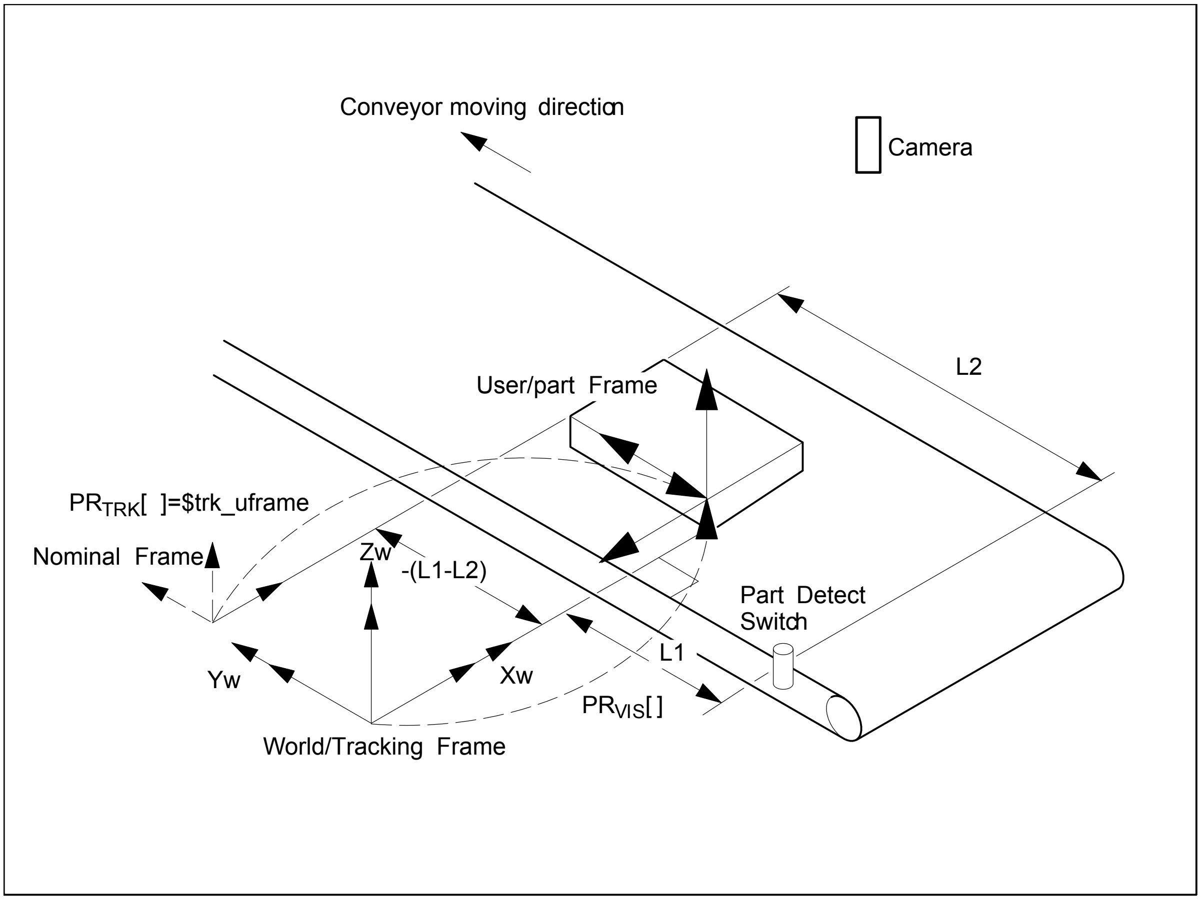


5.5.7 Using TRKUFRAME and VISUFRAME in Rail Tracking

Rail tracking is the same as line tracking except that the robot is on an integrated rail (normally in the y direction) and the tracking frame is the same as the robot World frame. See Figure 5–7 .

Use the same procedure described in Section 5.5.3 . PR[x] can be determined for TRKUFRAME and VISUFRAME. See Figure 5–7 . The tracking uframe can be used to compensate for the tracking path in rail tracking in the same way as in line tracking.

Figure 5–7. Rail Tracking Setup and Frame Relationships



5.6 HIGH SPEED SCANNING

5.6.1 Overview

The High Speed Scanning feature ensures an accurate part detection process when the conveyor operates at very fast speeds. R30*i*B supports up to 5 High Speed Digital Input (HSDI) #1–#5, located on the JRL5 connector of the controller, in place of the standard digital input normally used for part detection. Standard Cable support two HSDI port. For more than two HSDI, you need a special interface cable. Contact your FANUC America Corporation technical representative if you need more than two ports.

To use High Speed Scanning, you must

- Enable the High Speed Scanning system variable
- Modify your line tracking program

Refer to Appendix C for more information on the HDI cable.

5.6.2 Enabling High Speed Scanning

When the high speed scanning option is loaded, a new system variable, $HSLTENBL, is created. To enable this feature, you must set this variable to TRUE. Additional system variable $HDI_FLAG[port] are used to enable and disable the specific port of HSDI trigger dynamically. For example, when this flag variable is set to TRUE, HSDI triggers will be accepted and processed; when FALSE, HSDI triggers will be ignored.

Use Procedure 5-3 to enable the High Speed Scanning feature.

Procedure 5-3 Enabling High Speed Scanning

Conditions

- The High Speed Scanning option has been loaded.
- The System Trigger option has been loaded.
- The Trigger INPUT Number in the tracking schedule is set to 1.
- The part detect hardware is wired to HSDI. Contact FANUC representative for the cable part number.
- Encoder #1 is set up as the tracking encoder.

Steps

1. Press MENU.
2. Press NEXT.
3. Select SYSTEM.
4. Press F1, [TYPE].
5. Select Variables.
6. Move the cursor to the following variables and set their values accordingly,
   - $HSLTENBL = TRUE
   - $LNCFG.$HSDI_ENABLE = TRUE
7. Turn off the controller, and then turn it on again to accept the new setting.
8. Use the Encoder Setup Menu to select HDI port for individual encoder.

```
SETUP Encoders
       Encoder Number:  1
  1 Encoder Axis:                     0
  2 Encoder Type:             INCREMENTAL
  3 Encoder Enable:                  ON
    Current Count (cnts):             1
  4 Multiplier (ITP/update):          1
  5 Average (updates):                1
  6 Stop Threshold (cnt/updt):        0
  7 Simulate:          Enable:      OFF
  8           Rate (cnt/updt):        0
  9 HDI port Id:                      1
[ TYPE ]        ENCODER
```

5.6.3 Modifying Your Line Tracking Program to Use High Speed Scanning

After you have set the high speed scanning variables to TRUE, you can change your line tracking programs to make use of the high speed scanning option. Main Program (Job) without High Speed Scanning Instructions shows a standard line tracking program that **does not** use the high speed scanning feature.

Main Program (Job) without High Speed Scanning Instructions

```
! MOVE TO HOME
J P[1] 50% FINE
! ENABLE THE ENCODER
LINE[1] ON
! WAIT FOR PART DETECT
WAIT DI[1] ON
! GET TRIGGER VALUE
LINECOUNT[1] R[1]
! SET TRIGGER VALUE
SETTRIG LNSCH[1] R[1]
!SELECT A BOUNDARY
SELBOUND LNSCH[1] BOUND[1]
! CALL TRACKING PROGRAM
CALL TRACK
!MOVE TO HOME
J P[1] 50% FINE
```

Main Program (Job) with High Speed Scanning Instructions shows the same program but includes instructions for using the high speed scanning feature. The part of the program that has changed is shown between the dashed lines .

Main Program (Job) with High Speed Scanning Instructions

```
! MOVE TO HOME
J P[1] 50% FINE
! ENABLE THE ENCODER
LINE[1] ON
! ------------------------------
! ENABLE THE HSDI port 1
$HDI_FLAG[1] = 1
! WAIT FOR PART DETECT
WAIT $ENC_STAT[1].$ENC_HSDI = 1
! ------------------------------
! GET TRIGGER VALUE
LINECOUNT[1] R[1]
! SET TRIGGER VALUE
SETTRIG LNSCH[1] R[1]
! SELECT A BOUNDARY
SELBOUND LNSCH[1] BOUND[1]
! CALL TRACKING PROGRAM
CALL TRACK
! MOVE TO HOME
J P[1] 50% FINE
```

Differences between the Example Programs

The WAIT statement in Main Program (Job) with High Speed Scanning Instructions , which is used for part detection, is waiting for the value of the system variable $ENC_STAT[1].$ENC_HSDI to become 1 (TRUE), rather than waiting for a digital input to turn on. Though the tracking schedule indicates DI[1] as the trigger input, DI[1] (or any other DI input) will be ignored.

In addition $HDI_FLAG[1] = 1 statements were added in Main Program (Job) with High Speed Scanning Instructions . Setting $HDI_FLAG[1] to 1 activates the HSDI port 1 input and allows triggers on this input to be processed. When this variable is set to 0, triggers on the HSDI hardware input are ignored. This variable is set to 0 immediately after the part is detected so that no further triggers will be processed until the program has finished processing and has returned to the WAIT statement.

Note Every time the trigger is done, the $HDI_FLAG[] will be set to 0.

Note When you are editing tracking programs, conveyor resynchronization automatically uses the corresponding HSDI port hardware input for part detection. Therefore, you do not need to modify $HDI_FLAG[].

Limitations

The High Speed Scanning option has the following limitations:

- The ACCUTRIG instruction can not be used simultaneously with High Speed Scanning.
- The static accuracy tuning adjustment $LNCFG_GRP[1].$io_delay is available with High Speed Scanning (refer to Section 5.4.1); however, only the group 1 value is used, i.e. $LNCFG_GRP[2].$io_delay, $LNCFG_GRP[3].$io_delay, and so forth, will be ignored. There is no limitation on the amount of adjustment available through $LNCFG_GRP[1].$io_delay. For R30iB, you can use $ENC_IOD_ENB[] and $ENC_IODELAY[] to adjust the trigger value for individual encoder. When $ENC_IOD_ENB[1] is TRUE, the $ENC_IODELAY[1] is used for static accuracy tuning adjustment.

5.7 SKIP OUTBOUND MOVE

Skip Outbound Move allows a part to travel out of the boundary window without stopping production. This speeds up production and eliminates the need for you to manage an error condition when this occurs. This feature is enabled using the following system variables:

- $LNCFG_GRP[].$SKIP_OBNDMV : Enables the skip outbound feature.
- $LNCFG_GRP[].$SKP_ADJ_MS : Skip adjust time in milliseconds.
- $LNCFG_GRP[].$SKP_FLG_NO : Flag number to turn on when the skip condition occurs.

When the feature is enabled, the system skips the motion instruction that causes the robot to go out of the down stream boundary. Typically a tracking program that picks up a part on the conveyor would have three tracking motion instructions: above pick (P1), pick (P2), and above pick (P3). Depending on the timing there are four possible conditions that could occur if the Skip Outbound Move feature is enabled:

- When the program starts, P1 might already be out of bounds. In this case the system will skip all three positions.
- When the program starts, P1 is still inbounds, but P2 and P3 will be out of bounds when robot starts to move P2 and P3. In this case the system will reach P1 and skip P2 and P3.
- When the program starts, P1 and P2 are inbounds. But when the robot reaches P1 and P2 but before the robot starts to move to P3, P3 becomes out of bounds. In this case the system will skip P3.
- The system can reach all three positions while they are inside the boundary.

For a single pick program, the system will drop the part after picking up the part, so when the system skips the outbound move the robot will directly move to the non-tracking drop position. In this case there is no problem.

For a multiple pick program, the system will wait or execute the pick up for next part when current part is done. With the skip outbound move feature, the robot could be at P1 (condition 2) or P2 (condition 3) location when the skip condition was satisfied.

This feature only skips the outbound move. It does not guarantee the “destination gone” error would never occur. If the previous motion is a tracking move, the robot might still track out of bounds while waiting for next part to be inbound when user did not specify to stop tracking.

The robot should not stay at the P2 position because it will hold the part at the conveyor position too long and cause the robot to block the part flow on the conveyor.

To overcome this problem, you must set up a system variable $LNCFG_GRP[].$SKP_ADJ_MS to specify the time margin that would prevent this condition. This should be derived from the user program. The value should be the distance between P2 and P3 divided by the program speed of P3.

When the system determines whether or not P2 is out of bounds, the system uses this value to determine whether or not it has time to reach P3. If it does not have time to reach P3 then the system will skip P2 also. The system will adjust the time internally for a low override condition.

When you specify $LNCFG_GRP[].$SKP_FLG_NO to a valid flag port the system will turn on the flag you specified when the skip condition occurred. Your application program can set this flag to determine whether or not to use the same tool to pick up the next part. Also, your program can request that the next robot picks up the skipped part. Because the system only sets the flag, you need to reset the flag before using it.

Because the motion is skipped, all the local conditions associated with the motion will be skipped.

5.8 LIMIT CHECKING

Before the line tracking motion is executed, the system will check to determine if the last axis will reach the limit or not. If it will reach the limit, then the system will change the direction of the last axis movement. This function works only when the all following conditions are satisfied:

- $LNCFG_GRP[gnum].$LMT_CHK_ENB = TRUE (this is FALSE by default).
- The motion is:
  - Line tracking
  - Linear
  - RS_WORLD

The following system variables are used to support this function:

- $LNCFG_GRP[gnum].$LMT_CHK_ENB Enable/Disable function (default is FALSE)
- $LNCFG_GRP[gnum].$LMT_CHK_UL Upper soft limit margin (default is 20deg)
- $LNCFG_GRP[gnum].$LMT_CHK_LL Lower soft limit margin (default is 20deg)

For example, if the J6 axis stroke range is -360° to 360°, and both $LMT_CHK_UL and $LMT_CHK_LL are 20[deg], when the expected next destination exceeds the range from -340° to 340°, the robot will take another direction.

5.9 ETHERNET ENCODER

5.9.1 Overview

A typical line tracking system uses conveyor/conveyors to transfer a work piece for robot/robots to process. The application can be a material handling or painting application. When there are multiple robots working on the same conveyor, each robot needs to know the part's location on the conveyor. The Encoder Multiplexer is used to supply the encoder information to each robot on the same conveyor.

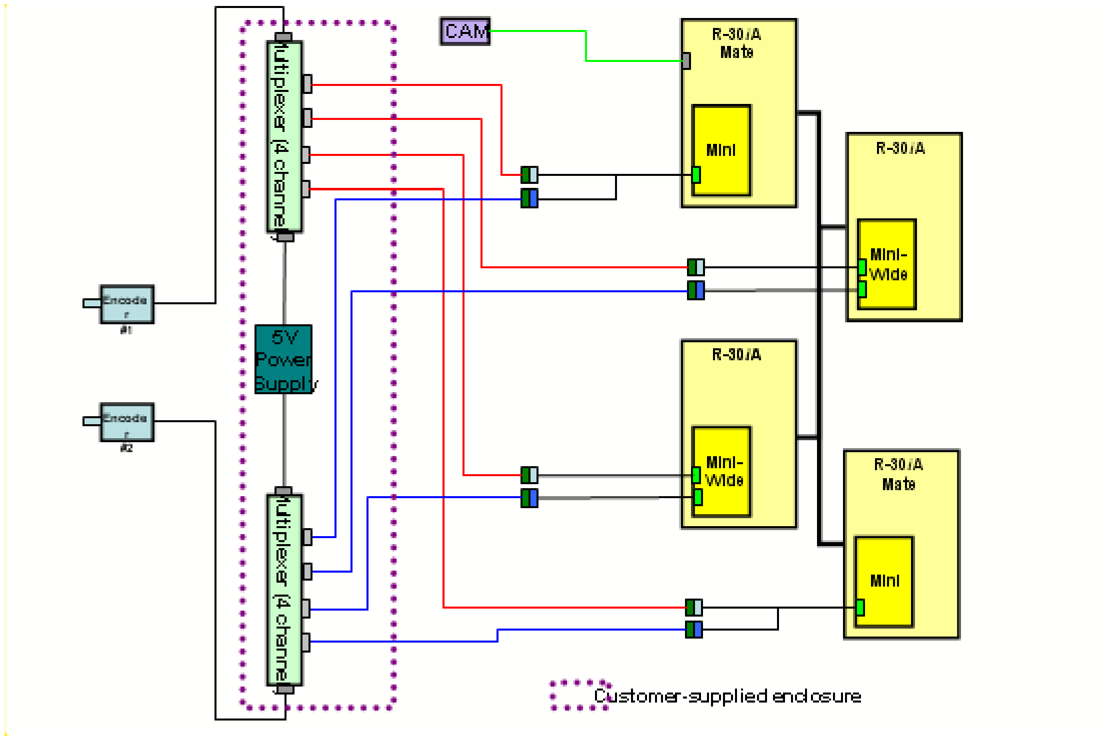
The Ethernet Encoder software option uses the Ethernet connection between robots instead of Encoder Multiplexer to supply the encoder information to each robot on the same conveyor.

The Ethernet Encoder provide the following benefits:

- No Encoder Multiplexer box is needed
- No Encoder Cable required for Slave controller.
- No Line tracking interface board required for slave controller.
- Much easier to set up (reduce labor cost)
- Use existing Ethernet network in the conveyor tracking Pick or Paint cell.
- Cost saving is significant!

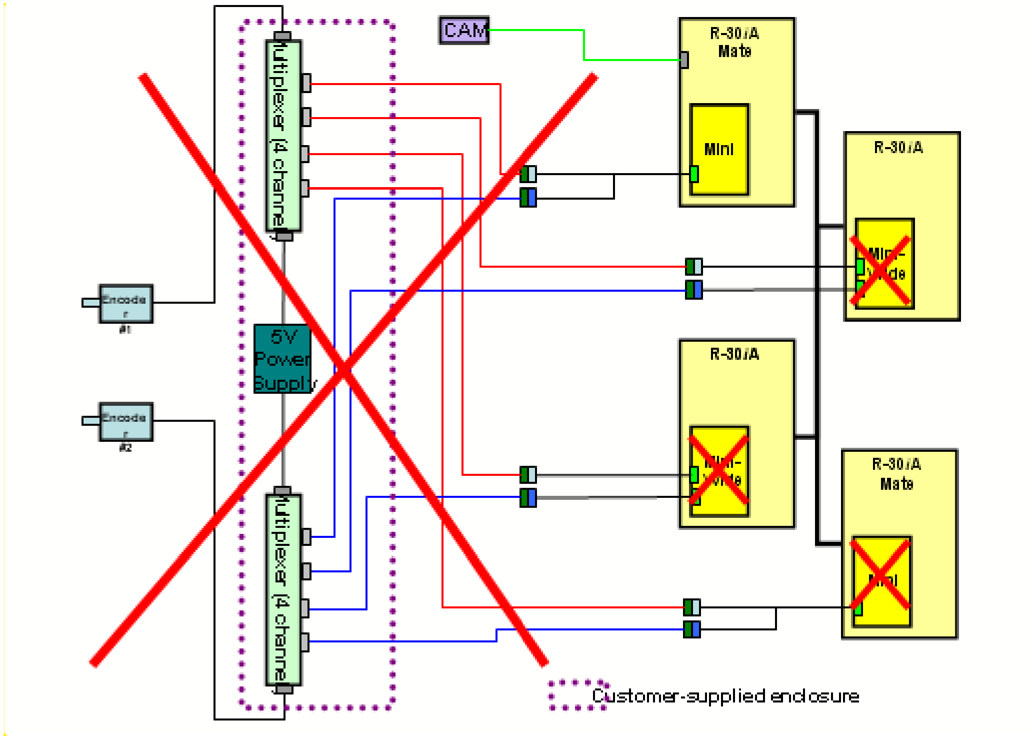
The following figure shows the existing system configuration for a four robots workcell with an infeed conveyor and an outfeed conveyor.

Figure 5–8. System with Encoder Multiplexer



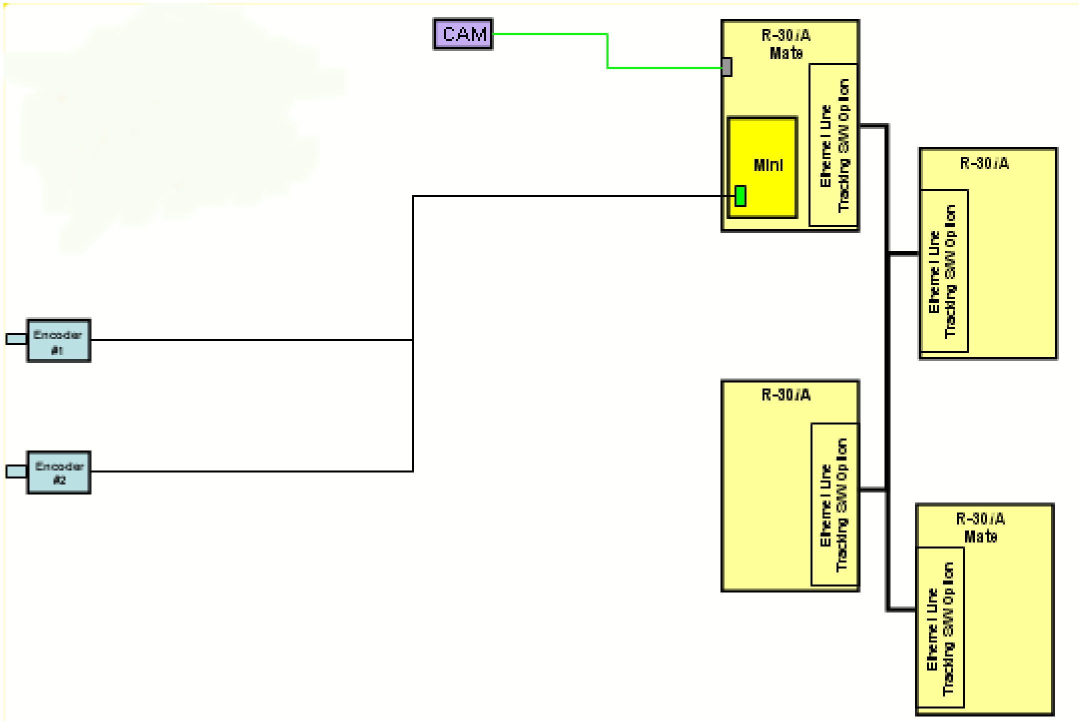
With the Ethernet Encoder option the following hardware (multiplexer, cable, line tracking interface board on slave) are not required.

Figure 5–9. System Changes



With the Ethernet Encoder option, the hardware Setup is as follows

Figure 5–10. System with Ethernet Encoder



The Ethernet Encoder option consists of both master and slave controllers where the master is the controller with the encoder(s) connect to it. The master controller transmits the encoder information to other slave controller/controllers over the **local** Ethernet Network connection.

Note This standalone option requires the Line tracking option (J512 or R 569) and ROS IP options (This will be automatically included when you load the Ethernet encoder option) . The Ethernet Encoder RIPE communications share same pipe with other protocols (EIP, PCDK, FTP etc) running on the same network. However, in normal installation condition they should not interfere with other protocols (EIP, PCDK, FTP etc).

5.9.2 Limitations

- Each controller can support up to four master Ethernet Encoders.
- Supports four controllers with Ethernet Encoder in one Ripe network. An additional controller would need to be evaluated based on the complexity of the application.
- The $scr.$itp_time for all the controllers should be set the same value.
- Ethernet Encoder does not support indexer conveyor.

5.9.3 Setup

5.9.3.1 Hardware setup

The Ethernet Encoder uses Ethernet network. This Ethernet network must be an isolated network so that it will not be affected by building/plant network. You would need an industrial grade switch when your system has more than two controllers. The suggested switch is N-Tron 306tx Industrial Ethernet Switch (Part #: MODLO000000113O).

- A system with only two controllers: You have a choice to use switch or not. Without a switch, Plug one end of an Ethernet cable into the Ethernet port of one controller and the other end to the other controller. If your system is Mate controller you needs to use crossover Ethernet Cable instead of regular Ethernet Cable. With a switch, do as below method.
- A system with more than two controllers: Each controller needs an Ethernet Cable. Connect one end of the Ethernet Cable to the Ethernet Port of controller and the other end of Ethernet Cable to an Ethernet Port of the switch.

5.9.3.2 Software Setup

- Set RIPE network:

  Refer to the “ROS itnerface Packet over Ethernet” in the *Internet Options Setup and Operations Manual* for more information.

- Setup the Master Encoder and Slave Encoder/Encoders.

  Display the Encoder Setup menu to setup Ethernet Encoder as follows:

```
SETUP Encoders
       Encoder Number:  1
  1 Encoder Axis:                      0
  2 Encoder Type:             INCREMENTAL
  3 Encoder Enable:                   ON
    Current Count (cnts):               1
  4 Multiplier (ITP/update):            1
  5 Average (updates):                  1
  6 Stop Threshold (cnt/updt):          0
  7 Simulate:          Enable:       OFF
  8           Rate (cnt/updt):          0
  9 Ethernet Master RIPE Id:            1
 10 Ethernet Master Encoder:            1
[ TYPE ]        ENCODER
```

— Set up the Master Ethernet Encoder

On a RIPE Ethernet Network, each controller has a unique id. The system variable $PH_ROSIP.$my_index is this id. To define a encoder as Master Encoder, item 9 “Ethernet Master RIPE Id” in above screen need to set to this controller’s RIPE id. The item 10 “Ethernet Master Encoder” need to set to the encoder number.

The controller with RIPE_id ($PH_ROSIP.$my_index) equal to 1 is the RIPE network master. The Master Encoder Controller should be the RIPE Master to reduce network traffic.

For example, In a Ripe network with 4 member, you want to set the encoder 1 of RIPE 1 controller (with $PH_ROSIP.$my_index = 1) as Master Encoder, you set item 9 “Ethernet Master RIPE Id” to 1 and item 10 “Ethernet Master Encoder” to 1.

The Master controller screen is displayed.

```
SETUP Encoders
       Encoder Number:  1
  1 Encoder Axis:                     1
  2 Encoder Type:            INCREMENTAL
  3 Encoder Enable:                  ON
    Current Count (cnts):             1
  4 Multiplier (ITP/update):          1
  5 Average (updates):                1
  6 Stop Threshold (cnt/updt):        0
  7 Simulate:          Enable:      OFF
  8          Rate (cnt/updt):         0
  9 Ethernet Master RIPE Id:          1
 10 Ethernet Master Encoder:          1
[ TYPE ]        ENCODER
```

You will be prompted to cycle power once you finish enter item 9 and 10. You should cycle power right away. However, if you have more than one encoder need to setup as Ethernet Encoder (for standard infeed and outfeed conveyors), you can wait until all of them was setup then cycle power.

— Set up the Slave Ethernet Encoder

You can set any encoder in any other controllers as Slave Encoder to that Master Encoder. To do that you need to duplicate the same information of item 9 and item 10 of the Master Encoder.

For example, in the same RIPE network, if you want to set Encoder 3 of RIPE 2 controller (with $PH_ROSIP.$my_index = 2) as a Slave Encoder of the Master Encoder described above, you get into Encoder setup menu and set item 9 “Ethernet Master RIPE Id” to 1 and item 10 “Ethernet Master Encoder” to 1.

```
SETUP Encoders
       Encoder Number:  3
  1 Encoder Axis:                     0
  2 Encoder Type:            INCREMENTAL
  3 Encoder Enable:                  ON
    Current Count (cnts):             1
  4 Multiplier (ITP/update):          1
  5 Average (updates):                1
  6 Stop Threshold (cnt/updt):        0
  7 Simulate:          Enable:      OFF
  8           Rate (cnt/updt):        0
  9 Ethernet Master RIPE Id:          1
 10 Ethernet Master Encoder:          1
[ TYPE ]        ENCODER
```

You will be prompted to cycle power once you finish enter item 9 and 10. You should cycle power right away. However, if you have more than one encoder need to setup as Ethernet Encoder (for standard infeed and outfeed conveyors), you can wait until all of them was setup then cycle power.

5.9.4 Verify Setup

The following steps can be used to verify whether setup is done correctly

- Cycle power for all the controllers that has done the setup procedure described in section 1.1.3.
- Go to the Encoder set up menu of Master Encoder controller, set item 7 “Simulate: Enable:” to on. Set item 8 “Rate” to 10. The “current count” item should start to change. You can also run the conveyor or turn the pulse coder to see “current count” change. Physically move the conveyor allows you to verify your encoder connection.

```
SETUP Encoders
       Encoder Number:  1
  1 Encoder Axis:                     1
  2 Encoder Type:            INCREMENTAL
  3 Encoder Enable:                  ON
    Current Count (cnts):          4565
  4 Multiplier (ITP/update):          1
  5 Average (updates):                1
  6 Stop Threshold (cnt/updt):        0
  7 Simulate:          Enable:       ON
  8           Rate (cnt/updt):       10
  9 Ethernet Master RIPE Id:          1
 10 Ethernet Master Encoder:          1
[ TYPE ]        ENCODER
```

- Go to the Encoder set up menu of Slave Encoder controller. The value of “current count” should be continuously changing.

```
SETUP Encoders
       Encoder Number:  3
  1 Encoder Axis:                     0
  2 Encoder Type:            INCREMENTAL
  3 Encoder Enable:                  ON
    Current Count (cnts):          4883
  4 Multiplier (ITP/update):          1
  5 Average (updates):                1
  6 Stop Threshold (cnt/updt):        0
  7 Simulate:          Enable:      OFF
  8           Rate (cnt/updt):        0
  9 Ethernet Master RIPE Id:          1
 10 Ethernet Master Encoder:          1
[ TYPE ]        ENCODER
```

- Go to the Encoder set up menu of Master Encoder controller. Set item 7 “Simulate: Enable:” to OFF. The value of “current count” item should stop changing. Stop conveyor can also stop the change of “current count”

```
SETUP Encoders
       Encoder Number:  1
  1 Encoder Axis:                     1
  2 Encoder Type:            INCREMENTAL
  3 Encoder Enable:                  ON
    Current Count (cnts):          7565
  4 Multiplier (ITP/update):          1
  5 Average (updates):                1
  6 Stop Threshold (cnt/updt):        0
  7 Simulate:          Enable:      OFF
  8           Rate (cnt/updt):       10
  9 Ethernet Master RIPE Id:          1
 10 Ethernet Master Encoder:          1
[ TYPE ]        ENCODER
```

- Go to the Encoder set up menu of Slave Encoder controller. The value of “current count” item should stop changing. If master encoder and slave encoder are the same then the setup is correct.

```
SETUP Encoders
       Encoder Number:  3
  1 Encoder Axis:                     0
  2 Encoder Type:            INCREMENTAL
  3 Encoder Enable:                  ON
    Current Count (cnts):          7565
  4 Multiplier (ITP/update):          1
  5 Average (updates):                1
  6 Stop Threshold (cnt/updt):        0
  7 Simulate:          Enable:      OFF
  8           Rate (cnt/updt):        0
  9 Ethernet Master RIPE Id:          1
 10 Ethernet Master Encoder:          1
```

```
[ TYPE ]          ENCODER
```

5.9.5 System Variables

The following system variables are related to the tracking application process. You might need to set the variables according to your error handling specification.

- $ENC_ETHCFG[].$TIMEOUT : Timeout value in milliseconds. When the slave controller did not receive a packet for more than this timeout value. The slave controller will post an error according to the setting of $ENC_ETHCFG[].$POST_TIMEOUT.
- $ENC_ETHCFG[].$POST_TIMEOUT : When this value of slave controller is not zero, slave controller will post error “LNTK-56 Ethernet Enc Timeout (Enc: 1” when no data packet is received after the timeout. When this value is zero, the slave will post error “LNTK-53 Ethernet Encoder NOT ready” when the tracking program is run.
- $ENC_ETHCFG[].$POST_M_DAL : When this value of slave controller is not zero, slave controller will post the following alarm, “LNTK-55 Ethernet Master DAL (Enc:enc_num)” when the master controller encounters DAL alarm for pulsecoder (A860-0301-T001) or “LNTK-59 Ethernet Master DTER (Enc:enc_num)” when master controller encounters encounters Serial pulse coder error such as DTER or CRCERR alarm for (αA1000S pulsecoder). This error will not be cleared by pressing the RESET key. When this value is 0, no error message will be posted on the slave controller right away. The system will set the internal variable master_not_ready to TRUE. When you try to run a tracking program, the error “LNTK-53 Ethernet Encoder NOT ready” will be posted.

5.9.6 Error Handling

This section describes how the application should handle the error caused by hardware problem.

There are five possible hardware errors:

- The Master Encoder DAL alarm “SRVO-96 DAL alarm (Track encoder 1)”
- The Master Ethernet Cable disconnected
- The Master Controller Power off
- The Slave Ethernet cable disconnected
- The Slave controller power off

The system that used Ethernet Encoder has a long conveyor. The parts flow from master to slave/slaves. In a Pick application, parts will be set into part queue and send down from the master controller to the slave controllers. In a Painting application system, the car/part may be detected in the slave controller and processed in the slave controller. When a hardware error occurs on the master controller, the conveyor may not stop moving.

Because of a hardware problem, the encoder count of slave controller will not be updated. So the encoder count will no longer match physical location of the part. The parts in the queue (sent down from the master controller or just created by the slave controller) will no longer be valid. When hardware problem is corrected, there is no guarantee that encoder count and physical location of the part will be match again. In this case, the best approach is discard all the remaining parts in the system (remove them from part queue) when an hardware error occur on the master controller. When the hardware error is fixed, Use the new encoder count with the new parts.

You can use system variables $enc_ethnet[enc_num].$M_ESTATE with following steps to accomplish above approach .

- When master encounter DAL alarm (such as encoder cable been disconnected):
  - The slave will post “Ethernet Master DAL (Enc:%d)” stop error alarm when $ENC_ETHCFG[].$post_m_dal is true.
  - Master Controller will set $ENC_ETHNET[].$M_ESTATE system variable to 0.
  - All the slaves corresponding encoder $ENC_ETHNET[corresponding_enc].$M_ESTATE will be updated to reflect master status because master can still send Ethernet Packet to all the slave in this condition.
  - Application software in the slave controller should discard all the parts in the queue and inform master that it has done so by requesting master to set $ENC_ETHNET[].$M_ESTATE with its ripe_id bit on. Application need to write software to do this.
  - When master recover from its DAL alarm, it will send Ethernet packet with new value (with slave’s ripe_id bit turn on). When slave see its own bit in $ENC_ETHNET[].$M_ESTATE is on, slave will know the encoder count can be used again.
- When the master Ethernet cable is pulled / master controller is power off:
  - All its slaves will post “Ethernet Master Encoder timeout” error when $ENC_ETHCFG[].$POST_TIMEOUT is true. When $ENC_ETHCFG[].$POST_TIMEOUT is false, the error will be posted only when tracking program is running.
  - All the slave’s system variables $ENC_ETHNET[].$ENABLE will be set to FALSE.
  - Slave set $ENC_ETHNET[].$M_ESTATE to 0.
  - The slave Application needs to clean up its application such as discard all it part queue
  - When Master power up, it should set $ENC_ETHNET[].$M_ESTATE to 0.
  - When master found error in sending Ethernet packet it should set $ENC_ETHNET[].$M_ESTATE to 0.
  - Application software in the slave controller should discard all the parts in the queue.
  - When master power up or Ethernet Cable is reconnected, master will set Ethernet packet to all the slaves with $ENC_ETHNET[].$M_ESTATE equal to 0
  - When Salve received new Packet and found it’s own ripe_id bit was not turn on in $ENC_ETHNET[].$M_ESTATE, it should request master to turn on slave’s ripe_id bit in $ENC_ETHNET[].$M_ESTATE,

- After master set slave's ripe_id bit on $ENC_ETHNET[].$M_ESTATE, The Ethernet packet with new value (with slave's ripe_id bit turn on) will be sent. When slave see its own bit is been turn on, it will know it can use the encoder count / parts again.

- When Slave Ethernet cable is pulled:
  - Slave will post “Ethernet Master Encoder timeout” error when $ENC_ETHCFG[].$POST_TIMEOUT is true. When $ENC_ETHCFG[].$POST_TIMEOUT is false, the error will be posted only when tracking program is running.
  - All the slave's system variables $ENC_ETHNET[].$ENABLE will be set to FALSE.
  - Slave set $ENC_ETHNET[].$M_ESTATE to 0.
  - The slave Application needs to clean up its application such as discard all it part queue
  - When slave's Ethernet cable is reconnected
  - Because master never change slave's ripe_id bit, So bit should remain on. When slave see its own bit is been turn on, it will know it can use the encoder count / parts again.
- When Slave power off and on
  - Because master never change slave's ripe_id bit, So bit should remain on. When slave see its own bit is been turn on, it will know it can use the encoder count / parts again.

5.10 INDEXER TRACKING WITH SERVO CONVEYOR LINE TRACKING (V8.10 and V8.20)

5.10.1 Overview

Servo Conveyor Line Tracking function is the function for using an extended axis as a conveyor. Therefore, the robot can track the conveyor that is indexed with high accuracy. This function requires the Line Tracking option (J512) and Servo conveyor Line Tracking option (J589). The Servo conveyor Line Tracking option includes Multi Motion Group option (J601) and Continuous Turn option (J613). These are used for keeping on moving an extended axis as a conveyor. As a limitation of Line Tracking, it is necessary to separate a robot and an extended axis as a conveyor into different groups.

5.10.2 Installation

The following options are needed for an indexer installation:

- J589: ServoConv. Ln Tk
- J512: Line Tracking
- R663: Constant Path

5.10.3 Setup

The following items need to be set up to use the Servo Conveyor Line Tracking System:

- Independent Extended Axis Setup
- Servo Conveyer Setup
- TP Program for Servo Conveyer
- Tracking Schedule Setup
- Example of TP Program

5.10.3.1 Independent Extended Axis Setup

In the Servo Conveyer Line Tracking function, you can use Independent Extended Axis as a conveyer. Set up the Independent Extended Axis on the ROBOT MAINTENANCE MENU at Controlled Start.

Procedure 5-4 Independent Axis Setup

1. While holding the PREV key and the NEXT key, turn on the power. After a while, You will see a screen similar to the following.

```
------- CONFIGURATION MENU -------
1. Hot start
2. Cold start
3. Controlled Start
4. Maintenance
Select >3
```

2. Select 3. Controlled Start and press ENTER. After a while a Controlled Start Menu will be displayed.
3. Press MENU to display the screen menu. Select 9 MAINTENANCE and press ENTER. The ROBOT MAINTENANCE screen will be displayed.

```
ROBOT MAINTENANCE
                             1/10
Setup Robot System Variables

 Group Robot Library/Option Ext Axes
   1   M-3iA/6S            0
   2   Independent Axes       0
[ TYPE ] ORD NO  AUTO  MANUAL
```

4. Select Independent Axes and press F4, MANUAL. The Independent Axes Setup Menu will be displayed.

```
*** Group 2 Total Axes Installed =  0
 1. Display/Modify Nobot Axis 1~4
 2. Add Axis
 3. Delete Axis
 4. EXIT
Select Item? 2
```

5. If you add axis, select 2: Add Axis. Press 2 key and ENTER and then the MOTOR SELECTION will be displayed. When axis setup is finished, the Independent Axes Setup Menu is displayed. If you finish the axis setup, select 4 EXIT.

Note Refer to the Mechanical Specification for the following procedure.

6. Select a servo motor which is used as Independent Axis. Select 1 Standard Method.

```
-- MOTOR SELECTION
1: Standard Method
2: Enhanced Method
3: Direct Entry Method
Select ==> 1
```

Then, select MOTOR SIZE.

```
MOTOR SIZE (Beta standard, Beta is)
80. biS0.2    84. biS1    88. biS12
81. biS0.3    85. biS2    89. biS22
82. biS0.4    86. biS3
83. biS0.5    87. biS6
  0. Next page
Select ==> 85
```

Finally, select MOTOR TYPE.

```
MOTOR TYPE
1. /2000      11. /4000
2. /3000      12. /5000
    13. /6000
Select ==> 11
```

The motor selection should consider motor speed and motor torque. Motor speed is dependent on the desired indexer speed and gear box used. Motor torque is dependent on what kind of parts (mass), how many parts on the whole conveyor, the mass and inertia of conveyor (from above information) and gear box used. When motor selected is under size, the OVC or overheat might occur during the continue operation.

```
CURRENT LIMIT FOR AMPLIFIER
2.   4A     10.  20A
5.  40A     12. 160A
80A
Select==> 10
```

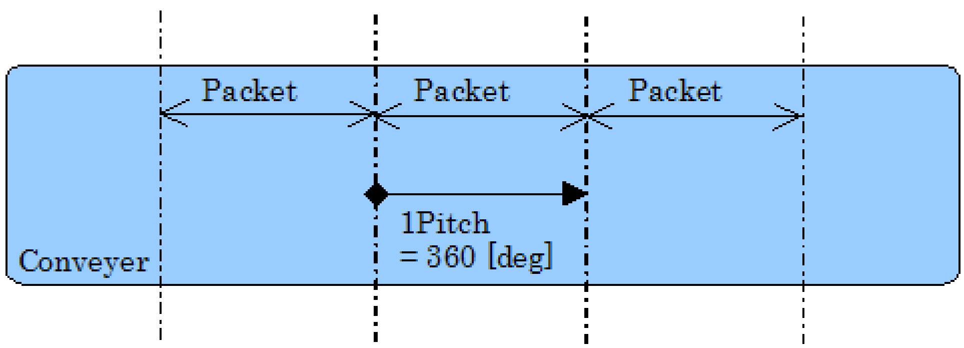
7. Select a current limit for the amplifier. You will see a screen similar to the following.

```
-- INDEPENDENT AXES TYPE -
1. Linear Axis
2. Rotary Axis
Select? 2
```

8. Select 2 Rotary Axis as the Independent axis type.
9. Enter the number of motor revolutions which correspond to one pitch of the conveyer. By this setting, a conveyer move one pitch when you move to 360 degree with controller.

   Pitch: The distance of conveyer when move conveyer to one packet of conveyer Packet: A part of a conveyer is divided by a constant distance.

Figure 5–11. Packet on Conveyor and Pitch




```
 — — GEAR RATIO — —
Enter Gear Ratio?
```

Because this axis will be used as a Servo Conveyor, the actual value will be determined later during the Servo Conveyor setup stage. The servo conveyor setup menu will change this value after you finish the setup. Therefore, you can enter any value such as 3.

10. Select 2:NO Change for setting a suggested speed as a max joint speed.

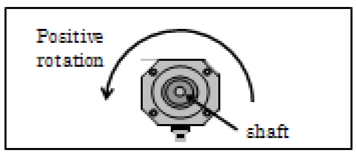
```
--MAX JOINT SPEED SETTING --
 Suggested Speed = 800.000(deg/s)
 (Calculated with Max Motor Speed)
Enter (1:Change, 2:No Change)? 2
```

11. Select Motor Direction. Choose 1:TRUE if the joint coordinate position of the conveyer increases when the motor rotates in the plus direction. Choose 2:FALSE if the joint coordinate position of the conveyer decreases when motor rotates in the plus direction.

```
-- MOTOR DIRECTION
INDEPENDENT AXES 1 Motion Sign = TRUE
Enter (1:TRUE, 2:FALSE)?
```

If you look at a motor from the front of the flange, a counter clockwise rotation is plus direction of a motor.

Figure 5–12. Positive Rotation



12. Enter the limit of the axis. Enter 180[deg] as an upper limit and –180[deg] as a lower limit.

```
-- UPPER LIMIT --
Enter Upper Limit ( deg)? 180
```

```
--LOWER LIMIIT--
Enter Lower Limit ( deg)? -180
```

13. Enter the mastering position of the axis. Enter the position where it is possible to carry out mastering within the motion range. Normally, the position is 0.

```
--MASTER POSITION –
Enter Master Position (deg)?
```

14. Enter the acceleration/deceleration time constants (ACC/DEC time). First, enter the 1st ACC/DEC time (acc_time1). Default value of acc_time1 is the default value.

Caution

It is very important that axis capability is set up so that desired maximum production indexing speed can be achieved. One indexing time = motion time + dwell time. Motion time = acceleration time + deceleration time. A typical index motion, acceleration time is equal to deceleration time. Acceleration time = acc_time 1 + acc_time 2 Acceleration time = min acc time So it is important that when set up motor motion, set the acc_time 1, acc_time 2 and min acc time.

For example, the application required maximum product is 300 ppm with dwell of 80 ms. The motion time for motor per indexing will be 120 ms. Min acc time, acc time 1 and acc time 2 should be multiple of itp_time (8 ms). So you should set min acc time = 56ms and acc time 1 = 32 , acc time 2 = 24.

15.

```
-- ACC/DEC TIME--
Default Value of acc_time1 = 384(ms)
Enter (1:Change, 2:No Change)?
```

If you want to change the value, choose 1:Change and enter the new value. If you don't want to change the value, choose 2:No Change.

```
Enter Accel Time 1 (ms)?
```

16. Next, enter the 2nd ACC/DEC time (acc_time2). Default value of acc_time2 is the default value.

```
Default value of acc_time2 = 192 (ms) Enter (1:Change, 2:No Change)?
```

17. If you want to change the value, choose 1:Change and enter the new value. If you don't want to change the value, choose 2:No Change. You will see a screen similar to the following.

```
Enter Accel Time 2 (ms)?
```

18. Enter Minimum Accel Time. When doing motion, if the calculated acceleration/deceleration time is smaller than the specified time, the acceleration/deceleration time will be clamped to the specified time.

```
-- MIN_ACCEL TIME --
Default Value of min_acctime = 384(ms)
Enter (1:Change, 2:No Change)?
```

If you want to change the value, choose 1:Change and enter the new value. If you don't want to change the value, choose 2:No Change. Normally, min_acctime is the sum of acc_time1 and acc_time2.

```
Enter Minimum Accel Time (ms)?
```

19. Enter Load Ratio. This value is the ratio of all load inertia to the rotor inertia. The valid range of Load Ratio is from 1.0 to 5.0. If you don't set this value, enter 0.

```
-- LOAD RATIO --
             LoadInertia + MotorInertia
 Load Ratio = -----------------------------------
                   MotorInertia
```

20. Enter Amplifier Number.

–SELECT AMP NUMBER– Enter amplifier number (1-56)?

21. Select Amplifier Type

```
-- SELECT AMP TYPE --
1. A06B-6107 series 6 axes amplifier
2. A06B-6117 series Alpha i amp. or
A06B-6130 series Beta i amp.
Select?
```

22. Enter Brake Number.

```
--BRAKE SETTING --
Enter Brake Number (0~16)?
```

23. Select the type of brake control (Servo Timeout). The brake control function put on a brake automatically when an axis does not move for a given length of time.

```
--SERVO TIMEOUT --
Servo off is Disable
Enter (1:Enable, 2:Disable)?
```

If you choose 1:Enable, then enter the delay time of brake control (Servo Off Time). The valid range of Servo Off Time is from 0 to 30(sec).

```
-- SERVO TIMEOUT VALUE --
Enter Servo Off Time? (0.0~30.0)
```

5.10.3.2 Servo Conveyor Setup

Set up Servo Conveyer using Procedure 5-5 . The setting of encoder and the setting of Continuous Turn are updated automatically by setting Servo Conveyer.

1. Press MENU.
2. Select SETUP.
3. Press F1, TYPE.
4. Select Indexers. You will see a screen similar to the following.

```
SETUP Indexers
                                   1/7
       Indexer Number: 1
  1 Encoder Number:                   0
  2 Indexer Type                 FANUC
  3 Index Advance Trigger DI:            1
  4 Delay move after trig (ms):            0
  5 Indexer Ready DO:                   1
  6 Tracking Schedule:                   1
  7 Create Index program:           Detail

[ TYPE ] DETAIL  INDEX  [CHOICE]
```

5. Move the cursor to Encoder Number and enter the number of the encoder that is used as servo conveyer.
6. Move the cursor to Indexer Type and select FANUC.
7. Move the cursor to Index Advance Trigger DI and enter the index of DI is used for moving the servo conveyer.
8. Move the cursor to Delay move after trig (ms) and enter the value of delay time until the servo conveyer starts after trigger.
9. Move the cursor to Indexer Ready DO. If you want to output DO at start of TP program for the servo conveyer, enter the index of DO. Refer to Section 5.10.3.3 about TP program for the servo conveyer.
10. Move the cursor to Tracking Schedule and enter the value of the tracking schedule that is used for servo conveyer.
11. Move the cursor to Indexer Type and press F2, DETAIL key or Enter Key. You will see a screen similar to the following. Because there is no [TYPE] in this menu, you can not select any other setup menu. To return to previous menu with [TYPE] press PREV key.

```
SETUP Indexer axis
                                  1/7
     Indexer  1: FANUC motor UNINIT
    Encoder Number:    0
  1 Robot Group:                   2
          Axis:                   1
  2 Motor Gear teeth                1
  3 Rotor input Gear teeth            1
  4 Rotor output Gear teeth           1
  5 Conveyor belt teeth              1
  6 Number of Flight                1
  7 Index Distance (mm)          10.000

          EXEC
```

12. Move the cursor to Robot Group and enter the group number of the extended axis that is used as the servo conveyer.

13. Move the cursor to Motor Gear teeth, Rotor input Gear teeth, Rotor output Gear teeth and Conveyer belt teeth and enter each value. The following figure shows the relationship between these values.

Figure 5–13. Motor, Rotor, and Conveyor

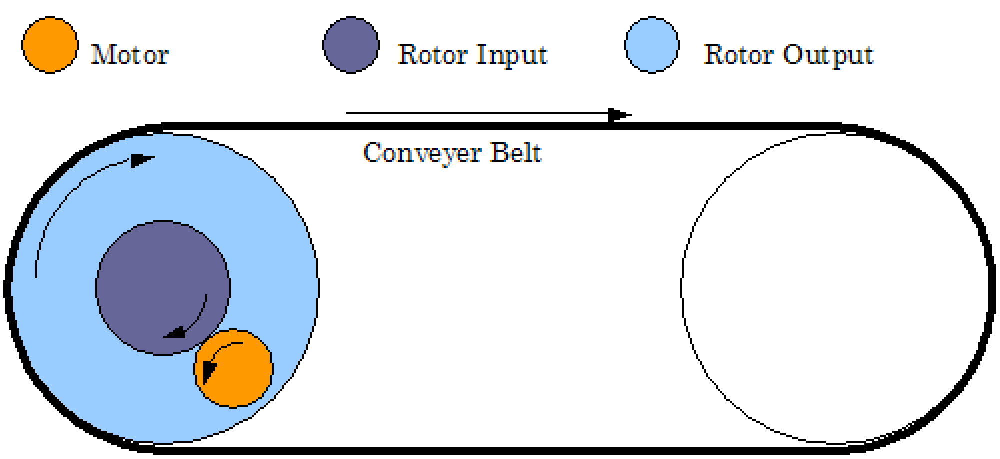
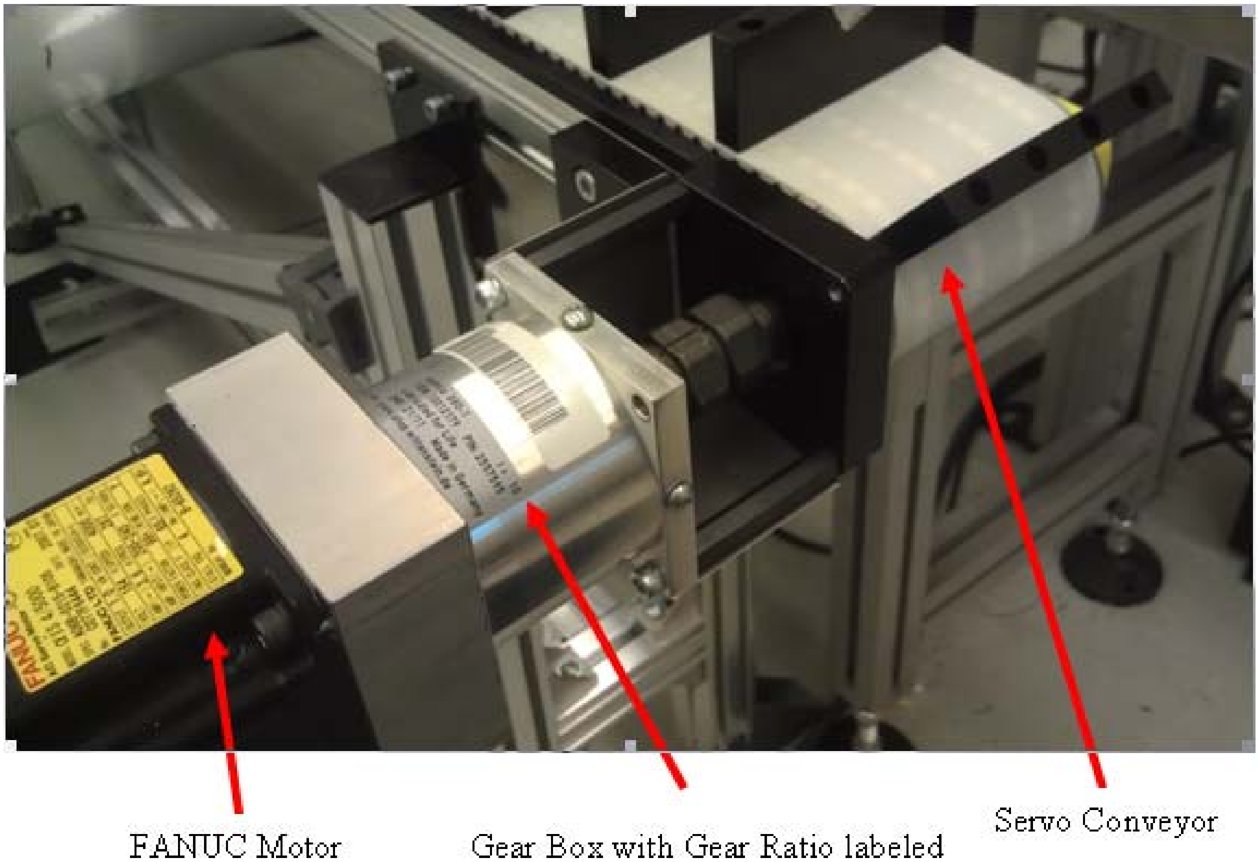
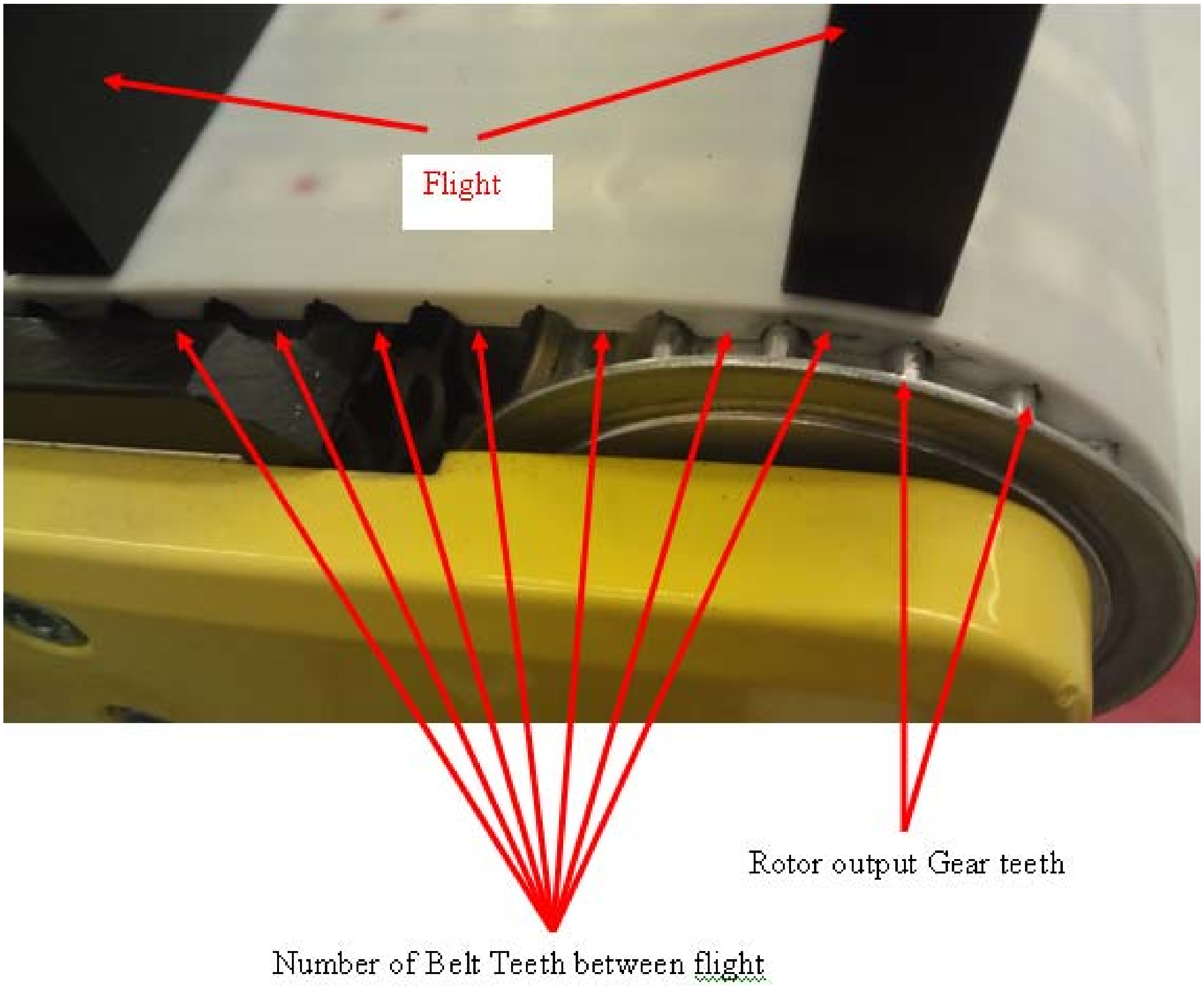


Figure 5–14. Indexer_gearbox



Most indexers have a gear box as show in Figure 5–23 . The gear box will have gear ratio stated in the label. If gear ratio is an integer value then enter 1 for “Motor Gear teeth” and enter gear ratio in “Rotor input Gear teeth”. If gear ratio is not an integer, find a value such that multiply this value the gear ratio become an integer. Enter the value found in “Motor Gear teeth” and enter final gear ratio in “Rotor input Gear teeth”. For example, if the labeled gear ratio is 3.5, we can use 2 as value to multiply and the final gear ratio will be 7. Therefore, enter 2 in “Motor Gear teeth” and enter 7 in “Rotor input Gear teeth”

Figure 5–15. indexer_belt_teeth



Enter “Rotor output Gear teeth” with the number of teeth of conveyor shaft as shown in the above figure. The distance between flights for the indexer conveyor are normally multiple of teeth as show in above figure. The total number of teeth in conveyor belt is belt teeth between flight multiply with total number of flights in the conveyor. In above picture’s example. If there are 100 flights, because there are 7 belt teeth between flight, so enter 700 for “Conveyor belt teeth”.

14. Move the cursor to Number of Flight and enter the number of packet on the servo conveyer.
15. Move the cursor to Index Distance and the distance a one pitch on the servo conveyer.
16. After the above setup, press F2, EXEC and be sure to Power off/on.

Note By the above setting, Continuous Rotation setup of the conveyer and Encoder setup of the specified encoder are also done.

5.10.3.3 TP Program for Servo Conveyor

It is necessary to prepare the TP program for moving the servo conveyer on tracking because the servo conveyer is set up as extended axis. By following the steps, the standard TP program for moving the servo conveyer according to DI is created. This TP program is called INDXG*.TP (* is group number of the servo conveyer).

Procedure 5-6 How to Create aTeach Pendant Program for the Servo Conveyor

1. Press MENU.
2. Select SETUP.
3. Press F1, TYPE.
4. Select Indexers. You will see a screen similar to the following.

```
SETUP Indexers
                                   1/7
       Indexer Number: 1
  1 Encoder Number:                   0
  2 Indexer Type                 FANUC
  3 Index Advance Trigger DI:            1
  4 Delay move after trig (ms):            0
  5 Indexer Ready DO:                  1
  6 Tracking Schedule:                  1
  7 Create Index program:            Detail

[ TYPE ] DETAIL  INDEX  [CHOICE]
```

5. Move the cursor to Create Index program in SETUP Indexers and press F2, DETAIL or ENTER. You will see a screen similar to the following. Because there is no [TYPE] in this menu, you can not select any other setup menu. To return to the previous menu, press PREV.

```
SETUP Indexer Motn
                                   4/4
            FANUC Indexer:  1
  1 Index Speed (part/min)            100
  2 Index Dwell (ms):                  0
  3 Indexer Register start              60
  4 Generate Index program
```

6. Move the cursor to Index Speed (part/min) and enter the value of Index Speed of the servo conveyer.
7. Move the cursor to Index Dwell and enter the value of the time to stop the servo conveyer. The speed pattern of the servo conveyer is different according to the value. Refer to following figures.

Figure 5–16. Speed Pattern if Index Dwell > 0

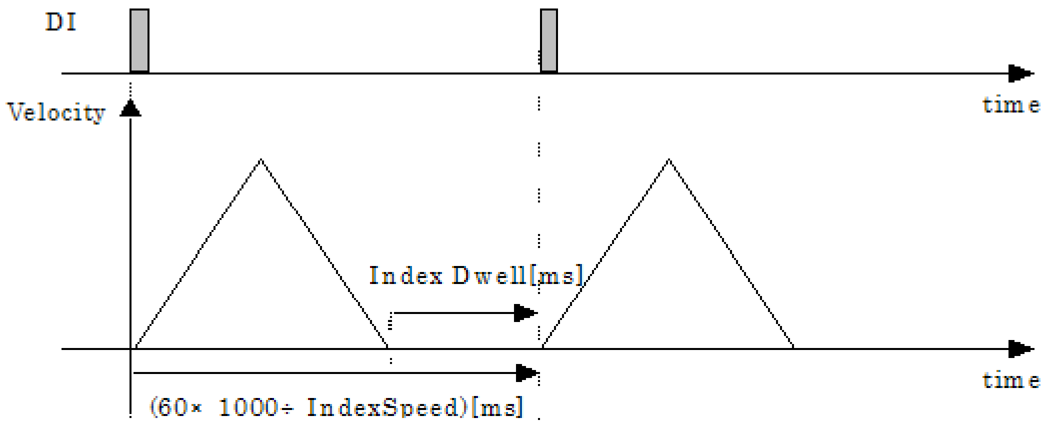


Figure 5–17. Speed Pattern if Index Dwell = 0

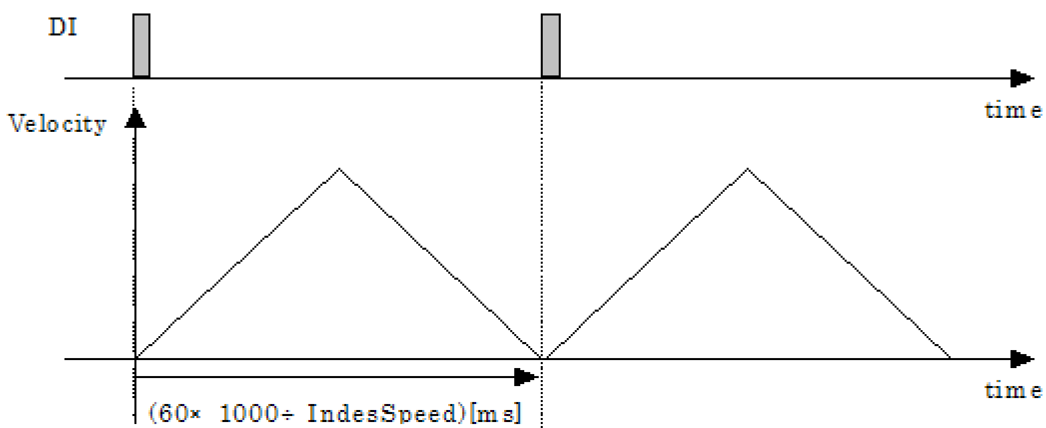
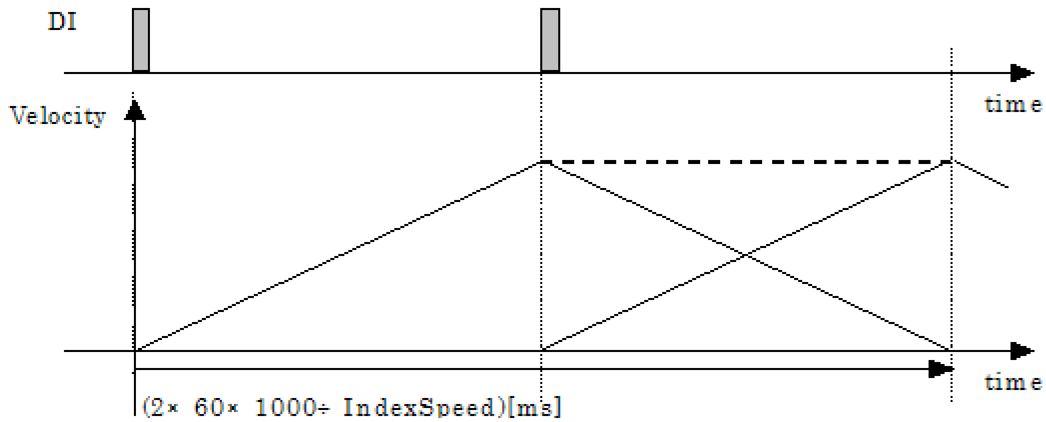


Figure 5–18. Speed Pattern if Index Dwell = -1



8. Move the cursor to Indexer Resister start and enter the start index of Register is used in TP program for the servo conveyer. Two registers are used in the TP program. For example, R[60] and R[61] are used when Indexer Resister start is 60.

9. After the above setting move the cursor to Generate Index program and press F2, CREATE. If there is no problem in the setting, Done is displayed and INDXG*.TP (* is group number of the servo conveyer) is created.

```
SETUP Indexer Motn
                                   4/4
            FANUC Indexer:  1
  1 Index Speed (part/min)            100
  2 Index Dwell (ms):                  0
  3 Indexer Register start              60
  4 Generate Index program
        Create Replace
```

Note

- Do not change INDXG*.TP directly because it is default program in the system. Rename INDXG*.TP and use it.
- Part Rate exceed allowable value might be displayed by the setting. This message shows that the calculated conveyer acceleration from the setting for TP exceeds allowable acceleration of the servo conveyer or robot. In this case, INDX*.TP is not created. adjust the setting so that the acceleration decreases.
- If there is INDXG*.TP and you press F2, CREATE, the Index Program that already exists is displayed. In this case, INDXG*.TP is not created. If you want to create TP program, press SHIFT key and F3, Replace key. If there is no problem in the setting, existing INDXG*.TP is overwritten by the created TP program from the current setting.

— If you press only F3, REPLACE, Hold Shift & Replace to replace program is displayed. In this case, INDXG*.TP is not overwritten.

— If INDXG*.TP is currently open in editor, the system will not be able to update the program. In this case the error message will displayed and done will not be displayed.

— Part Rate exceed allowable value might be displayed by the setting when you press SHIFT key and F3, REPLACE. Adjust the setting so that the acceleration decreases.

The following is an example of INDXG*.TP.

```
/MN
1:J P[1] 50% CNT0    ;   Movement
 to P[1]
2:  IF R[60:G2 Ready DO]=0,JMP LBL[3] ;
3:  DO[R[60]]=ON ;    Specified DO by R[60] become on
4:  LBL[3] ;
5:  R[61:G2 cur slot ID]=0    ;   Reset R[61]
6:  LBL[1] ;
7:  $INDEXER[1].$INDEX_MV=1 ;  Setting for waiting for DI trigger
8:J P[2] 72msec CNT100 INC ACC100    ; One pitch movement
9:  $INDEXER[1].$INDEX_MV=0 ;  Setting for waiting for DI trigger
10:  R[61:G2 cur slot ID]=R[61:G2 cur slot ID]+1; Keep count of movement
11:  IF R[61:G2 cur slot ID]=88,JMP LBL[2] ; If conveyer is turned once,
        Jump to LBL[2].
12:  JMP LBL[1] ;
13:  LBL[2] ;
14:J P[1] 100% CNT100    ;   Go to P[1] to correct iteration error
15:  R[61:G2 cur slot ID]=0    ;   Reset R[61]
16:  JMP LBL[1] ;
/POS
P[1:""]{
   GP2:
 UF : 0, UT : 1,
 J1=     0.000 deg
};
P[2:""]{
   GP2:
 UF : 0, UT : 1,
 J1=   360.000 deg
};
```

5.10.3.4 Tracking Schedule Setup

Tracking Schedule Setup for servo conveyer is the same as ordinary Line Tracking function. However, Nominal Tracking Frame Setup and Scale Factor Setup are different. Refer to the following procedures when the Servo Conveyer Line Tracking Function is installed.

Procedure 5-7 Nominal Tracking Frame Setup

1. Move the cursor to Nominal Track Frame in SETUP tracking.
2. Press F2, DETAIL. You will see a screen similar to the following.

```
SETUP Indexer Tkfr
                                   3/5
 Track Frame of Schedule:      1
  X:     0.00  Y:     0.00  Z:     0.00
  W:     0.00  P:     0.00  R:     0.00
 Teach Data:
  Origin:UNINIT      Enc_cnt           0
  X:     0.00  Y:     0.00  Z:     0.00
  +X dir:UNINIT      Enc_cnt           0
  X:     0.00  Y:     0.00  Z:     0.00
  +Y dir:UNINIT
  X:     0.00  Y:     0.00  Z:     0.00
 Scale (cnt/mm):      500.00

         TEACH  COMPUTE SCALE
```

3. Move TCP of the robot to a convenient position on the conveyer by jog. (This position should be an easily distinguish position on the conveyer).
4. Move the cursor to Origin of Teach Data and record this position by pressing SHIFT key and F2 Teach key.
5. Move the robot away from the conveyer so that the conveyer can be moved without running into the robot. Move the conveyer in the direction of normal flow for a distance of at least several hundred millimeters.
6. Move TCP of the robot to the same location relative to the conveyor that was used for the Origin position.
7. Move the cursor to +X dir of Teach Data and record this position by pressing SHIFT key and F2 Teach key.
8. Without moving the conveyer, move the robot at least 50mm in the direction perpendicular to the conveyor. Typically this is the left side of the conveyer.
9. Move the cursor to +Y dir of Teach Data and record this position by pressing SHIFT key and F2 Teach key.
10. Press F3, COMPUTE key for calculating the nominal tracking frame from recorded data. The value of the nominal tracking frame is update when the process is completed.

11. If you want to compute the scale factor for the servo conveyer here, move the cursor to Origin or +X dir and press F4, SCALE. The value of Scale (cnt/mm)is updated when the process is completed.

Procedure 5-8 Scale Factor Setup

1. Move the cursor to Enc Scale Factor in SETUP tracking,.
2. Press F2, TEACH key. You will see a screen similar to the following.

```
SETUP Indexer Scal
                                   1/3
 Track Schedule:              1
  Track Scale  (cntmm):         500.000

   Start Point:    UNINIT
    TCP X:    0.00 Y:   0.00 Z:    0.00
    Encoder Count:                    0
   End Point:      UNINIT
    TCP X:    0.00 Y:   0.00 Z:    0.00
    Encoder Count:                    0

        TEACH   COMPUTE
```

3. Move the conveyer until a convenient position on the conveyer reaches the upstream end of the robot workspace. This convenient position on the conveyer is used as the reference point.
4. Move TCP of the robot the reference point by jog.
5. Move the cursor to Start point and record this position and encoder count by pressing SHIFT key and F2, TEACH.
6. Move the robot away from the conveyer so that the conveyer can be moved without running into the robot.
7. Move the conveyer until the reference point on the conveyer reaches the downstream end of the robot workspace.
8. Move TCP of the robot the reference point by jog.
9. Move the cursor to End point and record this position and encoder count by pressing SHIFT key and F2, Teach.
10. Press F3, COMPUTE key for calculating the scale factor from recorded data. The value of the scale factor is update when the process is completed.

5.10.3.5 Example of Main TP Program

Teaching the TP program is the same as ordinary Line tracking function. However, it is necessary to run TP program for working a servo conveyer by multitasking. INDXG2.TP for working a servo conveyer is run at line 3 in the following Sample Program.

```
/MN
1:  J P[1] 100% FINE ;
2:  LINE[1] ON ;
3:  RUN INDXG2;
4:  LBL[1] ;
5:  WAIT DI[2]=ON    ;
6:  LINECOUNT[1] R[1] ;
7:  SETTRIG LNSCH[1] R[1] ;
8:  CALL TRACK    ;
9:  J P[1] 100% FINE    ;
10:  JMP LBL[1] ;
```

When Picktool is used, PTMAIN1.TP needs to updated to add RUN INDXG2 in the beginning of the program (when one indexer conveyor is used).

When more than one indexer is used in the system, PTMAIN1 should add all the indexer program (INDXG2, INDXG3, INDXG4 or INDXG5) in the beginning of the PTMAIN1.TP.

5.10.4 Servo Conveyer Line Tracking Function

5.10.4.1 Wait Indexer Stop Function

This function is enabled by default. When this function is enabled, a robot starts/ends a tracking while a servo conveyer stop. If you want to start a tracking motion while a servo conveyer moves, disable this function by the following procedure. The flag for switching the setting of this function is bit 30 of $LNCFG.$COMP_SW.

Note When Wait Indexer Stop function is disabled, a robot starts a tracking motion even if servo conveyer moves. If a servo conveyer moves at start of the tracking motion, the acceleration of the robot increases at that time. If the motion of robot is very aggressive, enable this function.

If the setting of this function is changed, create TP program for the servo conveyer again.

Procedure 5-9 Wait Indexer Stop Function Setup

1. Divide the value of $LNCFG.$COMP_SW by 1073741824 and then check current setting by following step. -

   - If the integer part of the calculated value is Odd number, this function is Enabled.

- If the integer part of the calculated value is Even number, this function is Disabled.

2. Change the value of $LNCFG.$COMP_SW according to the current setting.

   - If this function is enabled and you want to disable this function, subtract 1073741824 from the value of $LNCFG.$COMP_SW.
   - If this function is disabled and you want to enable this function, add 1073741824 to the value of $LNCFG.$COMP_SW.

3. Power Off/ON.

5.10.5 Limitations

Servo Conveyer Line Tracking function requires Line tracking function.

- It is possible to add up to four servo conveyer to a controller.(G1:Robot, G2-G5:Servo Conveyer)
- Servo Conveyer Line Tracking function can be used together with a traditional encoder conveyer.
  — Servo Conveyer Line Tracking function supports HSDI and ACCUTRIG instruction.
- It is not possible to use Visual Tracking on Servo conveyor. (It is possible to use Visual Tracking on traditional encoder conveyor.)
- The tracking robot can not use Continuous Turn function. - Original Path Resume feature is disabled.
- It is not possible to execute TP program of which a motion mask has the tracking robot group and Servo conveyor group.
- When conveyor speed is high, the robot motion will be aggressive and cause some noise when tracking. For any part rate greater than 200 ppm, the wait mechanism should be used to prevent robot jerking during transition of tracking. User need to set $indx_track[tracking_schedule] to 6. This will cause robot to wait until conveyor speed is 0 before start tracking and wait until conveyor speed to 0 before get off the tracking.

5.10.6 Alarms

Refer to the *Error Code Manual* for more information.

5.11 INDEXER TRACKING WITH SERVO CONVEYOR LINE TRACKING (V8.30 and later)

5.11.1 Overview

Indexer Tracking with Servo Conveyor Line Tracking uses an extended axis as a conveyor. The robot can track the conveyor that is indexed with high accuracy, or it can continue moving. This function requires the Line Tracking option (J512) and Servo conveyor Line Tracking option (J589). The Servo conveyor Line Tracking option includes Multi Motion Group (J601) and Continuous Turn (J613). These are used for keeping an extended axis as a conveyor. As a limitation of the Line Tracking function, it is necessary to separate a robot and an extended axis as a conveyor into different groups.

This function supports both indexing and continuous conveyor types.

5.11.2 Installation

The following options are needed for an indexer with Servo Conveyor Line tracking installation:

- J589: ServoConv. Ln Tk
- J512: Line Tracking
- R663: Constant Path

5.11.3 Setup

The following items need to be set up to use the Servo Conveyor Line Tracking System:

- Independent Extended Axis Setup
- Servo Conveyer Setup
- TP Program for Servo Conveyer
- Tracking Schedule Setup
- Example of TP Program

5.11.3.1 Independent Extended Axis Setup

In the Servo Conveyer Line Tracking function, you can use Independent Extended Axis as a conveyer. Set up the Independent Extended Axis on the ROBOT MAINTENANCE MENU at Controlled Start.

Procedure 5-10 Independent Axis Setup

1. While holding the PREV key and the NEXT key, turn on the power. After a while, you will see a screen as following.

```
------- CONFIGURATION MENU -------
1. Hot start
2. Cold start
3. Controlled Start
4. Maintenance
Select >3
```

2. Select 3, Controlled Start and press ENTER. After a while a Controlled Start Menu is displayed.
3. Press MENU to display the screen menu. Select 9 MAINTENANCE and press ENTER. The ROBOT MAINTENANCE screen will be displayed.

```
ROBOT MAINTENANCE
                             1/10
Setup Robot System Variables

 Group Robot Library/Option Ext Axes
   1   M-3iA/6S             0
   2   Independent Axes       0
[ TYPE ] ORD NO  AUTO  MANUAL
```

4. Select Independent Axes and press F4, MANUAL. The Independent Axes Setup Menu will be displayed.

```
*** Group 2 Total Axes Installed =  0
 1. Display/Modify Nobot Axis 1~4
 2. Add Axis
 3. Delete Axis
 4. EXIT
Select Item? 2
```

5. If you add axis, select 2: Add Axis. Press 2 and ENTER and then the MOTOR SELECTION will be displayed. When axis setup is finished, Independent Axes Setup Menu is displayed. If you finish the axis setup, select 4 EXIT.

   Note Refer to the Mechanical Specification for the following procedure.

6. Select a servo motor which is used as Independent Axis. Select 1 Standard Method

```
-- MOTOR SELECTION
1: Standard Method
2: Enhanced Method
3: Direct Entry Method
Select ==> 1
```

Then, select MOTOR SIZE.

```
MOTOR SIZE (Beta standard, Beta is)
80. biS0.2    84. biS1    88. biS12
81. biS0.3    85. biS2    89. biS22
82. biS0.4    86. biS3
83. biS0.5    87. biS6
  0. Next page
Select ==> 85
```

Finally, select MOTOR TYPE.

```
MOTOR TYPE
1. /2000      11. /4000
2. /3000      12. /5000
    13. /6000
Select ==> 11
```

The motor selection should consider motor speed and motor torque. Motor speed is dependent on the desired indexer speed and gear box used. Motor torque is dependent on what kind of parts (mass), how many parts on the whole conveyor, the mass and inertia of conveyor (from above information) and gear box used. When motor selected is under size, the OVC or overheat might occur during the continue operation.

```
CURRENT LIMIT FOR AMPLIFIER
2.   4A     10.  20A
5.  40A     12. 160A
80A
Select==> 10
```

7. Select a current limit for the amplifier. You will see a screen similar to the following.

```
-- INDEPENDENT AXES TYPE –
1. Linear Axis
2. Rotary Axis
Select? 2
```

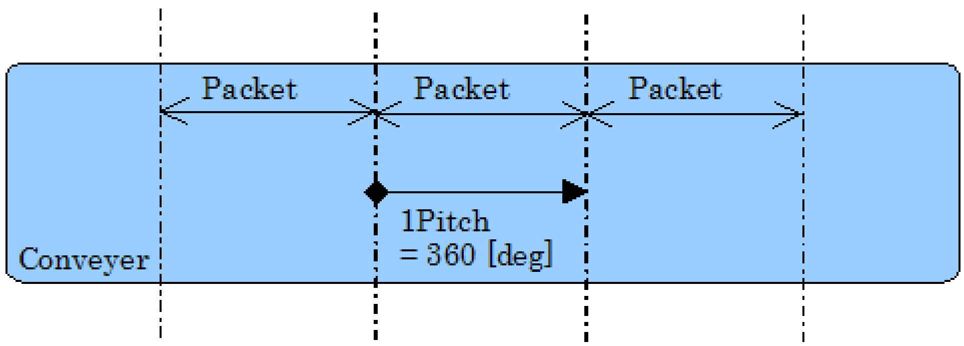
8. Select 2 Rotary Axis as the Independent axis type.

9. For the **index** conveyor type, enter the number of revolutions of the motor which corresponds to one pitch of the conveyer. By this setting, a conveyer moves one pitch when you move to 360 degrees with the controller.

   For a **continuously** moving conveyor, there is no Pitch. The maximum moving speed is determined by gear train.

   Pitch: The distance of conveyer when move conveyer to one packet of conveyer Packet: A part of a conveyer is divided by a constant distance.

Figure 5–19. Packet on Conveyor and Pitch




```
 — — GEAR RATIO — —
Enter Gear Ratio?
```

   Because this axis will be used as Servo Conveyor, the actual value will be determined later during the Servo Conveyor setup stage. The servo conveyor setup menu will change this value after you finish the setup. Therefore, you can enter any value such as 3.

10. Select 2:NO Change for setting a suggested speed as a max joint speed.

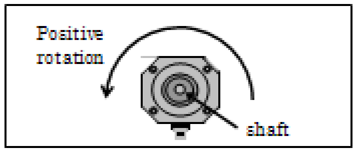
```
--MAX JOINT SPEED SETTING --
 Suggested Speed = 800.000(deg/s)
 (Calculated with Max Motor Speed)
Enter (1:Change, 2:No Change)? 2
```

11. Select Motor Direction. Choose 1:TRUE if the joint coordinate position of the conveyer increases when the motor rotates in the plus direction. Choose 2:FALSE if the joint coordinate position of the conveyer decreases when motor rotates in the plus direction.

```
-- MOTOR DIRECTION
INDEPENDENT AXES 1 Motion Sign = TRUE
Enter (1:TRUE, 2:FALSE)?
```

If you look at a motor from the front of the flange, a counter clockwise rotation is plus direction of a motor.

Figure 5–20. Positive Rotation



12. Enter the limit of the axis. Enter 180[deg] as an upper limit and –180[deg] as a lower limit.

```
-- UPPER LIMIT --
Enter Upper Limit ( deg)? 180
```

```
--LOWER LIMIIT--
Enter Lower Limit ( deg)? -180
```

13. Enter the mastering position of the axis. Enter the position where it is possible to carry out mastering within the motion range. Normally, the position is 0.

```
--MASTER POSITION -
Enter Master Position (deg)?
```

14. Enter the acceleration/deceleration time constants (ACC/DEC time). First, enter the 1st ACC/DEC time (acc_time1). Default value of acc_time1 is the default value.

Caution

For the **index** conveyor type, It is very important that axis capability is set up so that desire maximum production indexing speed can be achieved. One indexing time = motion time + dwell time. Motion time = acceleration time + deceleration time. A typical index motion, acceleration time is equal to deceleration time. Acceleration time = acc_time 1 + acc_time 2 - itp_time. Acceleration time = min acc time So it is important that when set up motor motion, set the acc_time 1, acc_time 2 and min acc time.

For example, the application required maximum product is 300 ppm (200 ms per index) with dwell of 80 ms. The motion time for motor per indexing will be 120 ms. So the acceleration time should be less than 60 ms. Min acc time, acc time 1 and acc time 2 should be multiple of itp_time (8 ms). So user should set min acc time = 64 ms and acc time 1 = 32 , acc time 2= 32.

For a **continuously** moving conveyor, It is very important that axis capability is set up so that desire stop distance can be achieved. User should consider the load of conveyor to see if the desired stop distance can be achieved. The stop distance is conveyor speed multiply acceleration time and divided by 2.

15.

```
-- ACC/DEC TIME--
Default Value of acc_time1 = 384(ms)
Enter (1:Change, 2:No Change)?
```

If you want to change the value, choose 1:Change and enter the new value. If you don't want to change the value, choose 2:No Change.

```
Enter Accel Time 1 (ms)?
```

16. Next, enter the 2nd ACC/DEC time (acc_time2). Default value of acc_time2 is the default value.

```
Default value of acc_time2 = 192 (ms) Enter (1:Change, 2:No Change)?
```

17. If you want to change the value, choose 1:Change and enter the new value. If you don't want to change the value, choose 2:No Change. You will see a screen similar to the following.

```
Enter Accel Time 2 (ms)?
```

18. Enter Minimum Accel Time. When doing motion, if the calculated acceleration/deceleration time is smaller than the specified time, the acceleration/deceleration time will be clamped to the specified time.

```
-- MIN_ACCEL TIME --
Default Value of min_acctime = 384(ms)
Enter (1:Change, 2:No Change)?
```

If you want to change the value, choose 1:Change and enter the new value. If you don't want to change the value, choose 2:No Change. Normally, min_acctime is the sum of acc_time1 and acc_time2.

```
Enter Minimum Accel Time (ms)?
```

19. Enter Load Ratio. This value is the ratio of all load inertia to the rotor inertia. The valid range of Load Ratio is from 1.0 to 5.0. If you don't set this value, enter 0.

```
-- LOAD RATIO --
             LoadInertia + MotorInertia
 Load Ratio = -----------------------------------
                   MotorInertia
```

20. Enter Amplifier Number.

    –SELECT AMP NUMBER– Enter amplifier number (1-56)?

21. Select Amplifier Type

```
-- SELECT AMP TYPE --
1. A06B-6107 series 6 axes amplifier
2. A06B-6117 series Alpha i amp. or
A06B-6130 series Beta i amp.
Select?
```

22. Enter Brake Number.

```
--BRAKE SETTING --
Enter Brake Number (0~16)?
```

23. Select the type of brake control (Servo Timeout). The brake control function put on a brake automatically when an axis does not move for a given length of time.

```
--SERVO TIMEOUT --
Servo off is Disable
Enter (1:Enable, 2:Disable)?
```

    If you choose 1:Enable, then enter the delay time of brake control (Servo Off Time). The valid range of Servo Off Time is from 0 to 30(sec).

```
-- SERVO TIMEOUT VALUE --
Enter Servo Off Time? (0.0~30.0)
```

5.11.3.2 Servo Conveyor Setup

Set up Servo Conveyer using Procedure 5-5 . The setting of encoder and the setting of Continuous Turn are updated automatically by setting Servo Conveyer.

1. Press MENU.
2. Select SETUP.
3. Press F1, TYPE.
4. Select Indexers. You will see a screen as following.

```
SETUP Indexers
                                   1/4
       FANUC conveyor: 1
  1 Encoder Number:                   0
  2 Conveyor Type                 Index
  3 Tracking Schedule:                1
    Encoder Scale (cnt/mm):    17745.13
  4 Program and Speed Control:   Detail

[ TYPE ] DETAIL  CNVY      [CHOICE]
```

5. Move the cursor to Encoder Number and enter the number of the encoder that is used as servo conveyer.
6. Move the cursor to Indexer Type and select Index or Continuous.
7. Move the cursor to Tracking Schedule and enter the value of the tracking schedule that is used for servo conveyer.
8. Move the cursor to Conveyor Type and press F2, DETAIL or ENTER. If you select **Index** as conveyor type you will see a screen similar to the following. Because there is no [TYPE] in this menu, you can not select any other setup menu. To return to the previous menu, press PREV.

```
SETUP Indexer axis
                                   1/12
     Indexer  1: FANUC motor UNINIT
    Encoder Number:    0
  1 Robot Group:                      2
          Axis:                       1
  2 Motor Gear teeth                  1
  3 Rotor input Gear teeth            1
  4 Rotor output Gear teeth           1
  5 Conveyor belt teeth               1
  6 Number of Flight                  1
  7 Index Distance (mm)          10.000
  8 Index advance Trigger DI          1
  9 Delay move after trigger (ms):   16
 10 Indexer Ready DO:                 2
 11 Flag for internal use:          131
 12 Flag 2 for internal use:        132

          EXEC
```

9. Move the cursor to Conveyor Type and press F2, DETAIL or ENTER. If you select **Continuous** as conveyor type you will see a screen similar to the following. Because there is no [TYPE] in this menu, you can not select any other setup menu. To return to the previous menu, press PREV.

```
SETUP Indexer axis
                                    1/8
     Conveyor  1: motor UNINIT
    Encoder Number:    0
  1 Robot Group:                       2
          Axis:                        1
  2 Motor Gear teeth                   1
  3 Rotor input Gear teeth             1
  4 Rotor output Gear teeth            1
  5 Belt Section teeth                 1
  6 Belt Section length (mm)      10.000
  7 Conveyor on DI                     1
  8 Flag for internal use:           131

          EXEC
```

10. Move the cursor to Robot Group and enter the group number of the extended axis that is used as the servo conveyer.
11. Move the cursor to Motor Gear teeth, Rotor input Gear teeth, Rotor output Gear teeth and Conveyer belt teeth and enter each value. The following figure shows the relationship between these values.

Figure 5–21. Motor, Rotor, and Conveyor

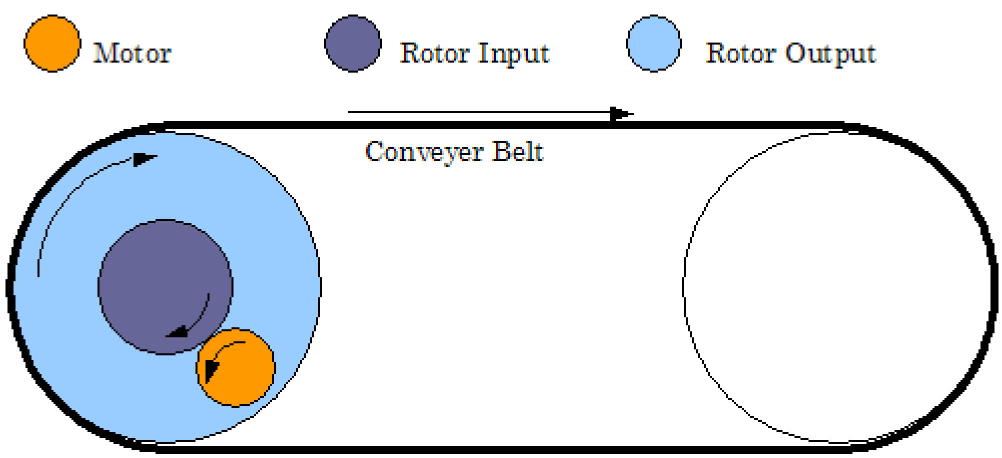
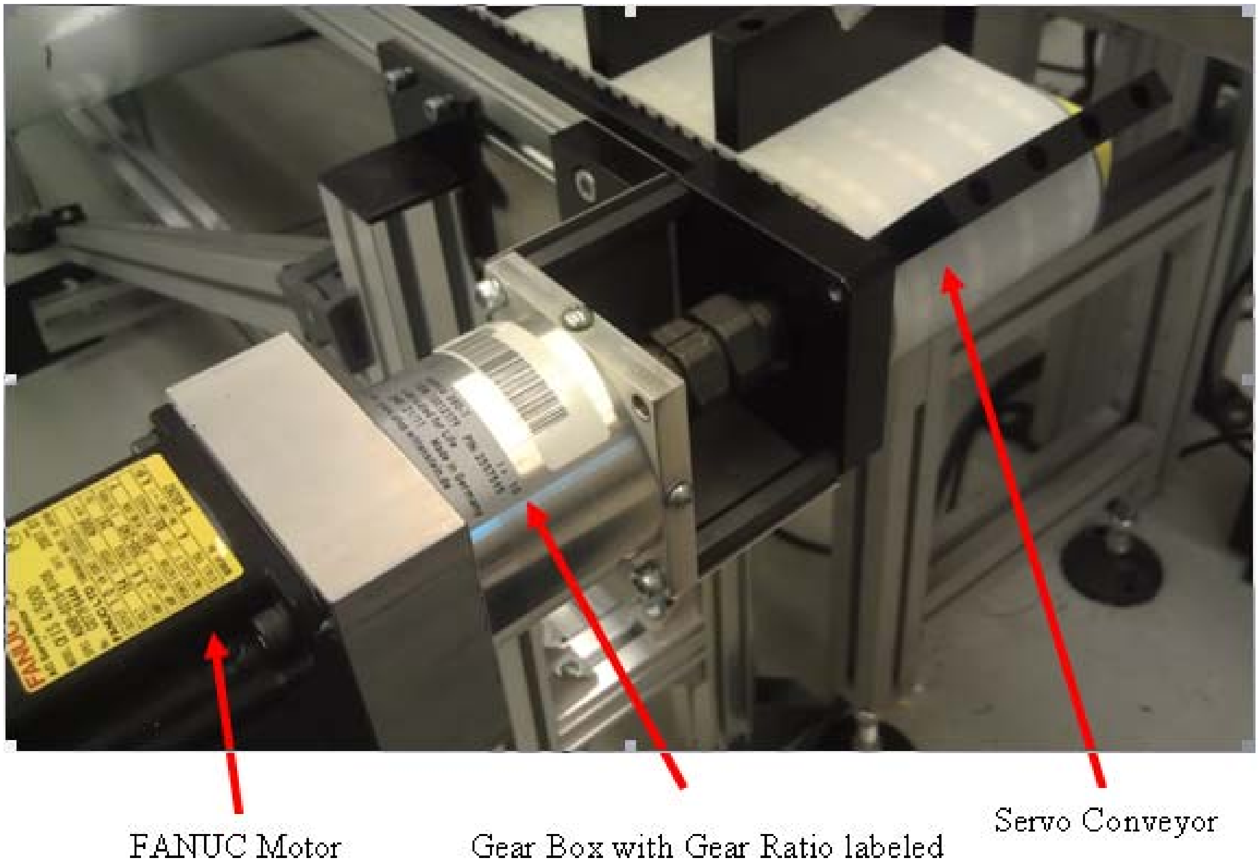
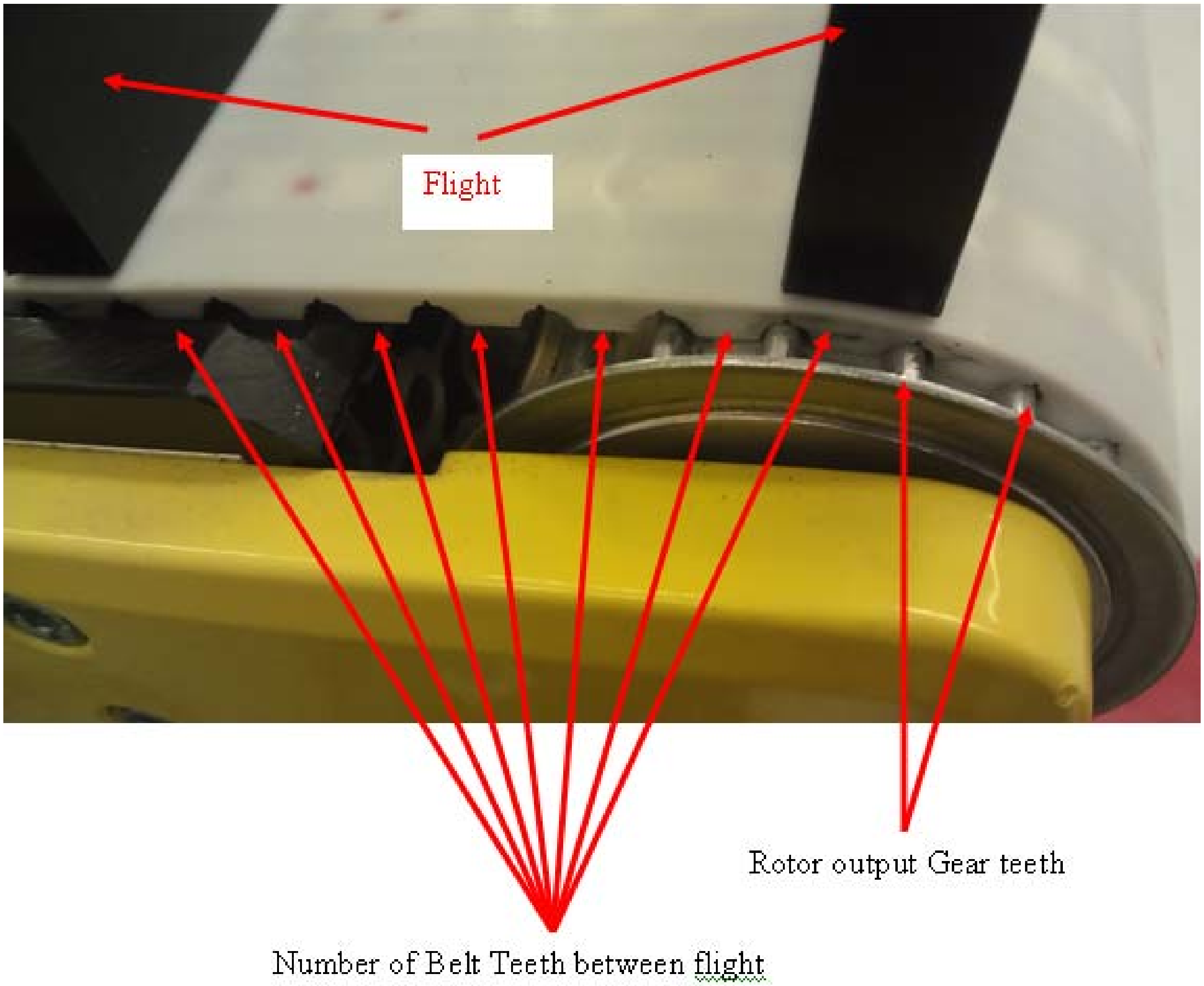


Figure 5–22. Indexer_gearbox



Most indexers have a gear box as show in the picture. The gear box will have gear ratio stated in the label. If gear ratio is an integer value then enter 1 for “Motor Gear teeth” and enter gear ratio in “Rotor input Gear teeth”. If gear ratio is not an integer, find a value such that multiply this value the gear ratio become an integer. Enter the value found in “Motor Gear teeth” and enter final gear ratio in “Rotor input Gear teeth”. For example, if the labeled gear ratio is 3.5, we can use 2 as value to multiply and the final gear ratio will be 7. Therefore, enter 2 in “Motor Gear teeth” and enter 7 in “Rotor input Gear teeth”

Figure 5–23. indexer_belt_teeth



Enter “Rotor output Gear teeth” with the number of teeth of conveyor shaft as shown in the above figure. The distance between flights for the indexer conveyor are normally multiple of teeth as show in above figure. The total number of teeth in conveyor belt is belt teeth between flight multiply with total number of flights in the conveyor. In above picture’s example. If there are 100 flights, because there are 7 belt teeth between flight, enter 700 for “Conveyor belt teeth”.

12. For the **index** conveyor type, move the cursor to Number of Flight and enter the number of packet on the servo conveyer.

For a **continuously** moving conveyor, move the cursor to Belt section teeth and enter the number. If the conveyor is assemble of several sections, enter the belt section teeth number. If conveyor is a continuous belt, enter any number between 4 to 7.

13. For the **index** conveyor type, move the cursor to Index Distance and the distance a one pitch on the servo conveyer.

For a **Continuous** conveyor type, move the cursor to Belt section length. Enter the length of belt section for a belt section conveyor. For a continuous conveyor enter the length between teeth distance (as shown in above picture) times number of belt section teeth.

14. For the **index** conveyor type, move the cursor to Index Advance Trigger DI and enter the index of DI is used for moving the servo conveyer.

    For a **Continuous** conveyor type, move the cursor to Conveyor ON DI and enter the index of DI to Start/Stop servo conveyer.

15. For the **index** conveyor type, move the cursor to Delay move after trig (ms) and enter the value of delay time until the servo conveyer starts after trigger.

    For a **Continuous** conveyor type, there are flag used internally by system. The default value will be shown in value. If there is any conflict with your system usage of flag, you can move this internal used flag to other none-conflict flag.

16. For the **index** conveyor type, move the cursor to Indexer Ready DO. If you want to output DO at start of TP program for the servo conveyer, enter the index of DO. Refer to Section 5.10.3.3 about TP program for the servo conveyer.
17. For the **index** conveyor type, move the cursor to Flag for internal use. There are flag used internally by system. The default value will be shown in value. If there is any conflict with your system usage of flag, you can move this internal used flag to other none-conflict flag by change the value and cold start.
18. For the **index** conveyor type, move the cursor to Flag 2 for internal use. There are flag used internally by system. The default value will be shown in value. If there is any conflict with your system usage of flag, you can move this internal used flag to other none-conflict flag by change the value and cold start.
19. After the above setup, press F2, EXEC and be sure to Power off/on.

    Note By the above setting, Continuous Rotation setup of the conveyer and Encoder setup of the specified encoder are also done.

5.11.3.3 TP Program for Servo Conveyor

It is necessary to prepare the TP program for moving the servo conveyer on tracking because the servo conveyer is set up as an extended axis. By following the steps, the standard TP program for moving the servo conveyer according to DI is created. For the index conveyor type, this TP program is called INDXG*.TP (* is group number of the servo conveyer). For a continuous conveyor type, this TP program is called CNVYG*.TP (* is group number of the servo conveyer).

Procedure 5-12 How to Create at Teach Pendant Program for Servo Conveyor

1. Press MENU.
2. Select SETUP.

3. Press F1, TYPE.
4. Select Indexers. You will see a screen as following.

```
SETUP Indexers
                                   1/4
       FANUC conveyor: 1
  1 Encoder Number:                  0
  2 Conveyor Type                Index
  3 Tracking Schedule:               1
    Encoder Scale (cnt/mm):    17745.13
  4 Program and Speed Control:   Detail

[ TYPE ] DETAIL  CNVY      [CHOICE]
```

5. Move the cursor to Program and Speed Control in SETUP Indexers and press F2, DETAIL key or ENTER key. If you select the **index** conveyor type, You will see a screen as following. Because there is no [TYPE] in this menu, you can not select any other setup menu. To return back to previous menu with [TYPE] press PREV key.

```
SETUP Indexer Motn
                                   4/4
            FANUC conveyor:  1
  1 Index Speed (part/min)          100
  2 Index Dwell (ms):                 0
  3 Indexer Register start           60
  4 Generate Index program
```

If you select the **Continuous** conveyor type, you will see a screen similar to the following. Because there is no [TYPE] in this menu, you can not select any other setup menu. To return to the previous menu, press PREV.

```
SETUP Indexer Motn
                                   2/2
            FANUC Conveyor:  1
  1 Conveyor Speed (mm/sec)          300
  2 Generate Conveyor program
```

6. For the **index** conveyor type, move the cursor to Index Speed (part/min) and enter the value of Index Speed of the servo conveyer. 3.

   For a **continuous** conveyor type, move the cursor to Conveyor Speed (mm/sec) and enter the desired value. If the conveyor already run, the conveyor will change the speed to new speed

7. For the **index** conveyor type, move the cursor to Index Dwell and enter the value of the time to stop the servo conveyer. The speed pattern of the servo conveyer is different according to the value. Refer to following figures.

Figure 5–24. Speed Pattern if Index Dwell > 0

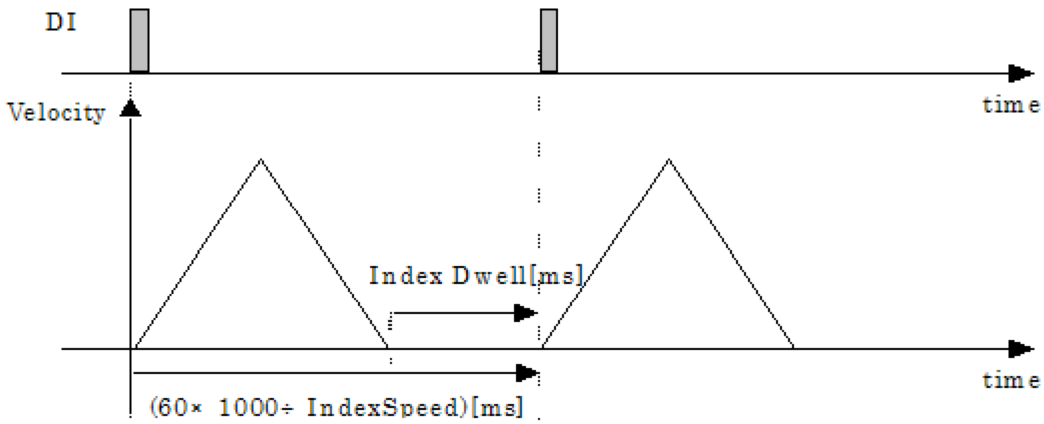


Figure 5–25. Speed Pattern if Index Dwell = 0

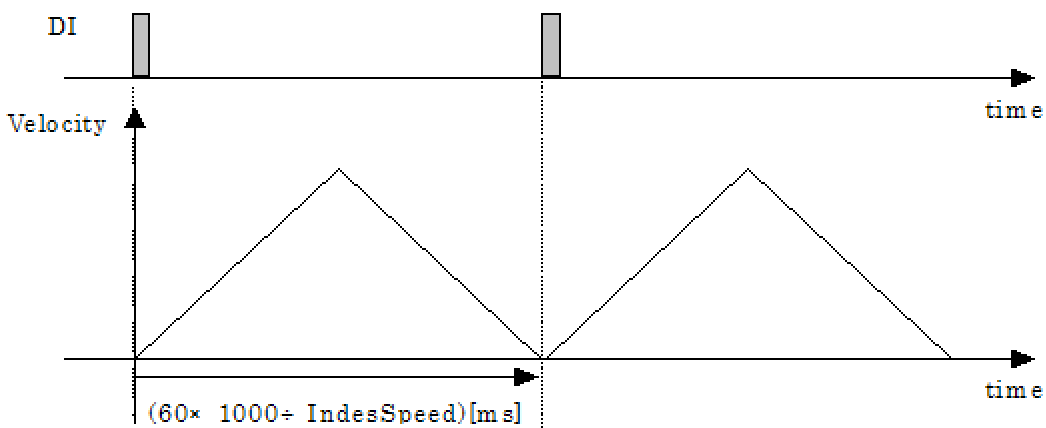
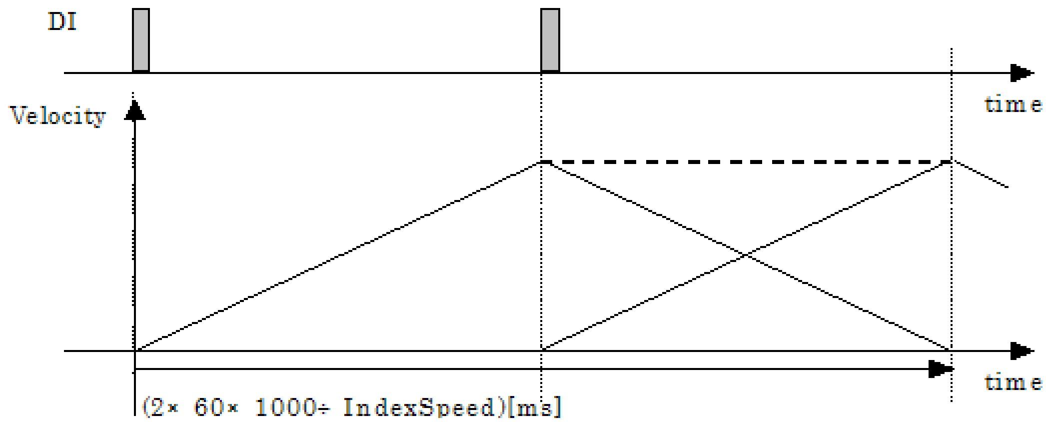


Figure 5–26. Speed Pattern if Index Dwell = -1



For a **continuous** conveyor type, move the cursor to Generate Conveyor program. If CNVYG*.TP does not exist in the system, F2 will be labeled CREATE. If CNVYG*.TP already exists in the system, F3, REPLACE will be displayed. Press F3, CREATE to generate the CNVYG*.TP program. Press F3, REPLACE to replace current program.

```
SETUP Indexer Motn
                                   2/2
            FANUC Conveyor:  1
  1 Conveyor Speed (mm/sec)           300
  2 Generate Conveyor program
```

Do not change CNVYG*.TP directly because it is the default program in the system. There should be no need for you to change CNVYG*.TP. The conveyor speed will not change with CNVYG*.TP. The conveyor speed is controlled by the value of item 1.

8. For the **index** conveyor type, move the cursor to Indexer Register start and enter the start index of Register is used in TP program for the servo conveyer. Two registers are used in TP program. For example, R[60] and R[61] are used when Indexer Resister start is 60.

9. For the **index** conveyor type, after the above setting move the cursor to Generate Index program. If INDXG*.TP does not exist in the system, F2 will be labeled CREATE. If INDXG*.TP already exists in the system, F3, REPLACE will be displayed. Press F2, CREATE. If there is no problem in the setting, Done is displayed and INDXG*.TP (* is group number of the servo conveyer) is created.

```
SETUP Indexer Motn
                                    4/4
            FANUC Indexer:  1
  1 Index Speed (part/min)             100
  2 Index Dwell (ms):                    0
  3 Indexer Register start              60
  4 Generate Index program
        Create
```

```
SETUP Indexer Motn
                                    4/4
            FANUC Indexer:  1
  1 Index Speed (part/min)             100
  2 Index Dwell (ms):                    0
  3 Indexer Register start              60
  4 Generate Index program
              Replace
```

Note

- Do not change INDXG*.TP directly because it is default program in the system. Rename INDXG*.TP and use it.
- Part Rate exceed allowable value might be displayed by the setting. This message shows that the calculated conveyer acceleration from the setting for TP exceeds allowable acceleration of the servo conveyer or robot. In this case, INDX*.TP is not created. adjust the setting so that the acceleration decreases.
- If there is INDXG*.TP and you press F2, CREATE, the Index Program that already exists is displayed. In this case, INDXG*.TP is not created. If you want to create the TP program, press SHIFT and F3, REPLACE. If there is no problem in the setting, the existing INDXG*.TP is overwritten by the created TP program from the current setting.
  - If you press only F3, REPLACE, press SHIFT & REPLACE to replace the program that is displayed. In this case, INDXG*.TP is not overwritten.
  - If INDXG*.TP is currently open in editor, the system will not be able to update the program. In this case the error message will displayed and done will not be displayed.
  - Part Rate exceed allowable value might be displayed by the setting when you press SHIFT and F3, REPLACE. Adjust the setting so that the acceleration decreases.

The following is an example of INDXG*.TP.

```
/MN
1:J P[1] 50% CNT0    ;   Movement
 to P[1]
2:  IF R[60:G2 Ready DO]=0,JMP LBL[3] ;
3:  DO[R[60]]=ON ;    Specified DO by R[60] become on
4:  LBL[3] ;
```

```
5:  R[61:G2 cur slot ID]=0    ;   Reset R[61]
6:  LBL[1] ;
7:  $INDEXER[1].$INDEX_MV=1 ;  Setting for waiting for DI trigger
8:J P[2] 72msec CNT100 INC ACC100    ; One pitch movement
9:  $INDEXER[1].$INDEX_MV=0 ;  Setting for waiting for DI trigger
10:  R[61:G2 cur slot ID]=R[61:G2 cur slot ID]+1; Keep count of movement
11:  IF R[61:G2 cur slot ID]=88,JMP LBL[2] ; If conveyer is turned once,
        Jump to LBL[2].
12:  JMP LBL[1] ;
13:  LBL[2] ;
14:J P[1] 100% CNT100    ;   Go to P[1] to correct iteration error
15:  R[61:G2 cur slot ID]=0    ;   Reset R[61]
16:  JMP LBL[1] ;
/POS
P[1:""]{
   GP2:
 UF : 0, UT : 1,
 J1=     0.000 deg
};
P[2:""]{
   GP2:
 UF : 0, UT : 1,
 J1=   360.000 deg
};
```

The following is an example of CNVYG*.TP.

```
/MN
1:  LBL[1] ;
2:  $INDEXER[1].$INDEX_MV=1 ;
3:J P[2] 32msec CNT100 INC ACC100    ;
4:  $INDEXER[1].$INDEX_MV=0 ;
16:  JMP LBL[1] ;
/POS
P[1:""]{
   GP2:
 UF : 0, UT : 1,
 J1=     0.000 deg
};
P[2:""]{
   GP2:
 UF : 0, UT : 1,
 J1=   360.000 deg
};
```

5.11.3.4 Tracking Schedule Setup

Tracking Schedule Setup for servo conveyer is the same as ordinary Line Tracking function. However, Nominal Tracking Frame Setup and Scale Factor Setup are different. Refer to the following procedures when the Servo Conveyer Line Tracking Function is installed.

Procedure 5-13 Nominal Tracking Frame Setup

1. Move the cursor to Nominal Track Frame in SETUP tracking.
2. Press F2, DETAIL. You will see a screen similar to the following.

```
SETUP Indexer Tkfr
                                    3/5
 Track Frame of Schedule:      1
  X:    0.00  Y:    0.00  Z:    0.00
  W:    0.00  P:    0.00  R:    0.00
 Teach Data:
  Origin:UNINIT      Enc_cnt          0
  X:    0.00  Y:    0.00  Z:    0.00
  +X dir:UNINIT      Enc_cnt          0
  X:    0.00  Y:    0.00  Z:    0.00
  +Y dir:UNINIT
  X:    0.00  Y:    0.00  Z:    0.00
 Scale (cnt/mm):     500.00

         TEACH  COMPUTE SCALE
```

3. Jog the TCP to a convenient position on the conveyer. (This position should be an easily repeatable position on the conveyer).
4. Move the cursor to Origin of Teach Data and record this position by pressing SHIFT and F2, TEACH.
5. Move the robot away from the conveyer so that the conveyer can be moved without running into the robot. Move the conveyer in the direction of normal flow for a distance of at least several hundred millimeters.
6. Move TCP of the robot to the same location relative to the conveyor that was used for the Origin position.
7. Move the cursor to +X dir of Teach Data and record this position by pressing SHIFT and F2, TEACH.
8. Without moving the conveyer, move the robot at least 50mm in the direction perpendicular to the conveyor. Typically this is the left side of the conveyer.
9. Move the cursor to +Y dir of Teach Data and record this position by pressing SHIFT ke and F2, TEACH.
10. Press F3, COMPUTE, to calculate the nominal tracking frame from recorded data. The value of the nominal tracking frame is update when the process is completed.

11. If you want to compute the scale factor for the servo conveyer here, move the cursor to Origin or +X dir and press F4, SCALE. The value of Scale (cnt/mm)is updated when the process is completed.

Procedure 5-14 Scale Factor Setup

1. Move the cursor to Enc Scale Factor in SETUP tracking,
2. Press F2, TEACH. You will see a screen similar to the following.

```
SETUP Indexer Scal
                                   1/3
 Track Schedule:              1
  Track Scale  (cntmm):         500.000

   Start Point:    UNINIT
    TCP X:    0.00 Y:   0.00 Z:    0.00
    Encoder Count:                    0
   End Point:      UNINIT
    TCP X:    0.00 Y:   0.00 Z:    0.00
    Encoder Count:                    0

        TEACH   COMPUTE
```

3. Move the conveyer until a convenient position on the conveyer reaches the upstream end of the robot workspace. This convenient position on the conveyer is used as the reference point.
4. Move TCP of the robot the reference point by jog.
5. Move the cursor to Start point and record this position and encoder count by pressing SHIFT key and F2, TEACH.
6. Move the robot away from the conveyer so that the conveyer can be moved without running into the robot.
7. Move the conveyer until the reference point on the conveyer reaches the downstream end of the robot workspace.
8. Jog the TCP to the reference point.
9. Move the cursor to End point, and record this position and encoder count by pressing SHIFT and F2, TEACH.
10. Press F3, COMPUTE to calculate the scale factor from recorded data. The value of the scale factor is update when the process is completed.

5.11.3.5 Example of Main TP Program

Teaching the TP program is the same as ordinary Line tracking function. However, it is necessary to run the TP program when using a servo conveyer using multitasking. INDXG2.TP/CNVYG2.TP for working a servo conveyer is run at line 3 in the following Sample Program.

```
/MN
1:  J P[1] 100% FINE ;
2:  LINE[1] ON ;
3:  RUN INDXG2;   or  RUN CNVYG2;
4:  LBL[1] ;
5:  WAIT DI[2]=ON    ;
6:  LINECOUNT[1] R[1] ;
7:  SETTRIG LNSCH[1] R[1] ;
8:  CALL TRACK    ;
9:  J P[1] 100% FINE    ;
10:  JMP LBL[1] ;
```

For the **index** conveyor type, when *i*RPicktool is used, PTMAIN1.TP needs to updated to add RUN INDXG2 in the beginning of the program (when one indexer conveyor is used).

When more than one indexer is used in the system, PTMAIN1 should add all the indexer program (INDXG2, INDXG3, INDXG4 or INDXG5) in the beginning of the PTMAIN1.TP.

For a **continuous** conveyor type,when *i*RPicktool is used, PTMAIN1.TP needs to updated to add RUN CNVYG2 in the beginning of the program (when one indexer conveyor is used).

When more than one conveyor is used in the system, PTMAIN1 should add all the converyor program (CNVYG2, CNVYG3, CNVYG4 or CNVYG5) in the beginning of the PTMAIN1.TP.

5.11.4 Servo Conveyer Line Tracking Function

5.11.4.1 Wait Indexer Stop Function

This function is enabled by default. When this function is enabled, a robot starts/ends a tracking while a servo conveyer stop. If you want to start a tracking motion while a servo conveyer moves, disable this function by the following procedure. The flag for switching the setting of this function is bit 30 of $LNCFG.$COMP_SW.

Note When Wait Indexer Stop function is disabled, a robot starts a tracking motion even if servo conveyer moves. If a servo conveyer moves at start of the tracking motion, the acceleration of the robot increases at that time. If the motion of robot is very aggressive, enable this function.

If the setting of this function is changed, create TP program for the servo conveyer again.

Procedure 5-15 Wait Indexer Stop Function Setup

1. Divide the value of $LNCFG.$COMP_SW by 1073741824 and then check current setting by following step. -
   - If the integer part of the calculated value is Odd number, this function is Enabled.
   - If the integer part of the calculated value is Even number, this function is Disabled.
2. Change the value of $LNCFG.$COMP_SW according to the current setting.
   - If this function is enabled and you want to disable this function, subtract 1073741824 from the value of $LNCFG.$COMP_SW.
   - If this function is disabled and you want to enable this function, add 1073741824 to the value of $LNCFG.$COMP_SW.
3. Power Off/ON.

5.11.5 Limitations

Servo Conveyer Line Tracking function requires Line tracking function.

- It is possible to add up to four servo conveyer to a controller.(G1:Robot, G2-G5:Servo Conveyer)
- Servo Conveyer Line Tracking function can be used together with a traditional encoder conveyer.
  — Servo Conveyer Line Tracking function supports HSDI and ACCUTRIG instruction.
- It is not possible to use Visual Tracking on Servo conveyor. (It is possible to use Visual Tracking on traditional encoder conveyor.)
- The tracking robot can not use Continuous Turn function. - Original Path Resume feature is disabled.
- It is not possible to execute TP program of which a motion mask has the tracking robot group and Servo conveyor group.
- When conveyor speed is high, the robot motion will be aggressive and cause some noise when tracking. For any part rate greater than 200 ppm, the wait mechanism should be used to prevent robot jerking during transition of tracking. You need to set $indx_track[tracking_schedule] to 6. This will cause robot to wait until conveyor speed is 0 before start tracking and wait until conveyor speed to 0 before get off the tracking.

5.11.6 Alarms

Refer to the *Error Code Manual* for more information.

Chapter 6

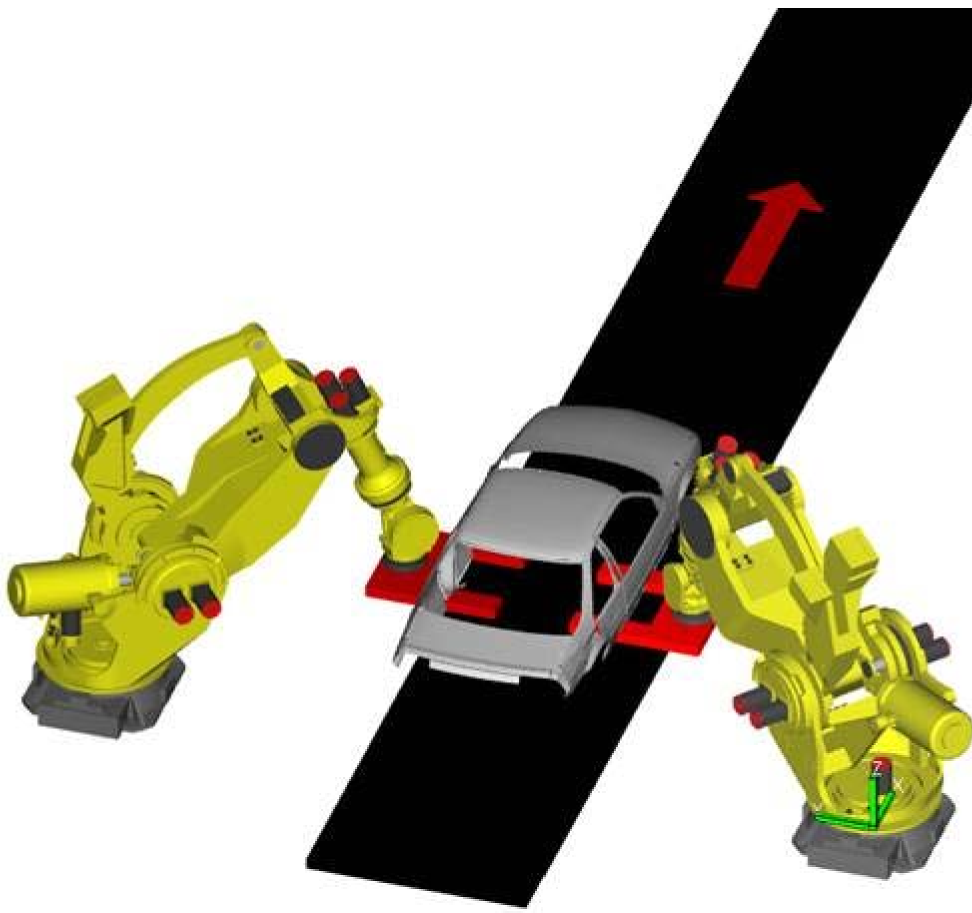
LINE TRACK COORD

Contents

6.1 Overview

Line Track Coord (R853), available in V8.30P/08 and later, allows motion group pairs to be programmed to line track together in a coordinated relationship. This is the only situation where multiple group motion is allowed within a line-tracking program. One example of an application is picking and placing parts on a moving conveyor which are too heavy to be lifted by one robot. In this case, a coordinated relationship must be maintained while in motion in order to avoid dropping or damaging the part. The example application is shown in Figure 6–1.

Figure 6–1. Example Line Track Coord Application



6.2 Requirements

The following hardware and software is required to use Dual Arm Coordinated Line Tracking (Line Track Coord):

- HandlingTool v8.30/P08 and Later
- 128MB DRAM

6.3 Included Software

All of the following features are included as part of Line Track Coord:

- Multi-Arm Package (R819) which includes Tracking Jog (R624)
- Coordinated Motion Plus Package(R669) which includes iRCalibration Vision Multi-Cal(J993)
- Dynamic UFrame(R700)
- Constant Path (R663)
- Line Tracking (J512)
- Cntrl stop by E-Stop (J570)

Caution

Cntrl stop by E-Stop (J570) replaces certain types of immediate stops with controlled stops. This increases stopping time in those scenarios but maintains the desired path and robot to robot coordination during these stops. For more information, refer to the Safety section of your *Mechanical Unit Operator ' s Manual* for more information.

6.4 Capabilities

With Line Track Coord you will be able to:

- Program two robots to track a conveyor while maintaining a coordinated relationship
- Program regular single-arm line tracking applications
- Perform regular coordinated motion applications

6.5 Limitations

Line Track Coord is not supported with any other motion features except:

- DCS (J567)
- Interference Check (R759, R761)
- Constant Path (R663)

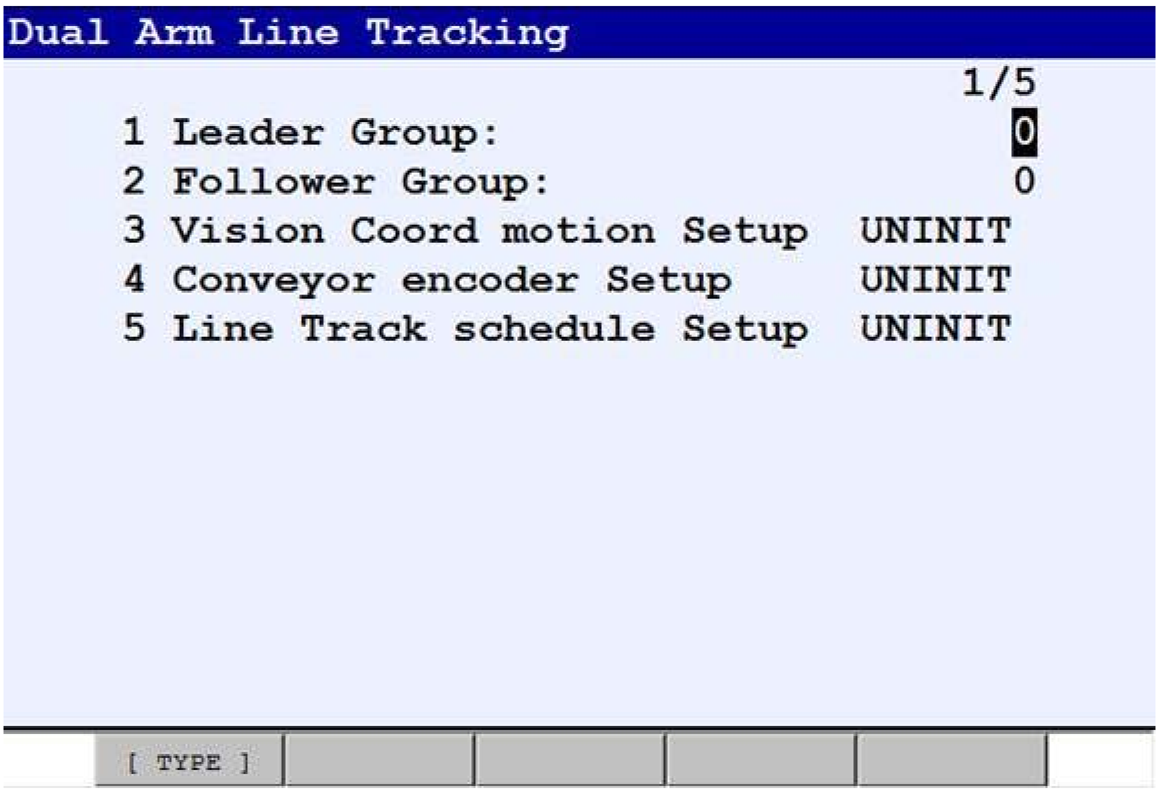
6.6 Setup

6.6.1 Overview

In order to set up Line Track Coord, follow the steps defined in the Line Track Coord Setup Screen.

The screen can be reached by Menu->Setup->[Type]->lntk Coord and should initially look as shown in Figure 6–2.

Figure 6–2. Line Track Coord Main Setup Screen



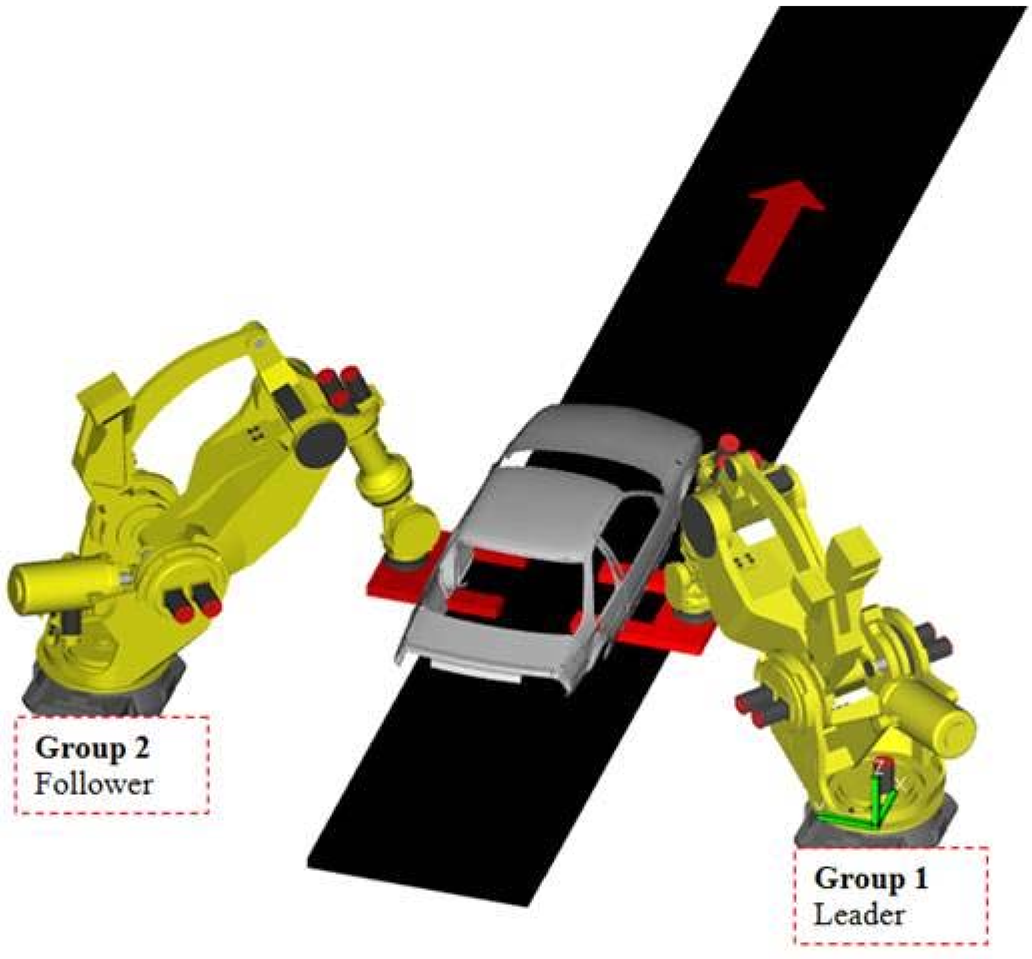
6.6.2 Leader Group

For item 1, enter the leader group for the coordinate pair which is to be used for line tracking. The selected leader group will be used to set up Coordinated Pair 1 automatically. In the example application, the leader group will be Group 1. Refer to Figure 6–3 for more detail.

6.6.3 Follower Group

For item 2, enter the follower group for the coordinate pair which is to be used for line tracking. The selected follower group will be used to set up Coordinated Pair 1 automatically. In the example application, the follower group will be Group 2. Refer to Figure 6–3 for more detail.

Figure 6–3. Leader and follower groups for Line Track Coord Application



6.6.4 *i*RCalibration MultiCal Vision Coord Motion Setup

In order to calibrate the coordinated pair accurately, it is highly recommended that *i*RCalibration Vision MultiCal be used. This method requires a camera to be set up on one of the robots in the coordinated pair and a target on the other. The option comes standard with the R853 option.

Please select ’3 Vision Coord motion Setup’ from the Line Track Coord setup screen and hit enter or the ’Detail’ softkey. Follow the instructions detailed in Vision Multi-Cal to calibrate the coordinated pair to be used for line tracking.

After calibrating the coordinated pair, a cold start message will be posted. This can be ignored until the end of Line Track Coord Setup.

Figure 6–4. Line Track Coord Main Setup Screen

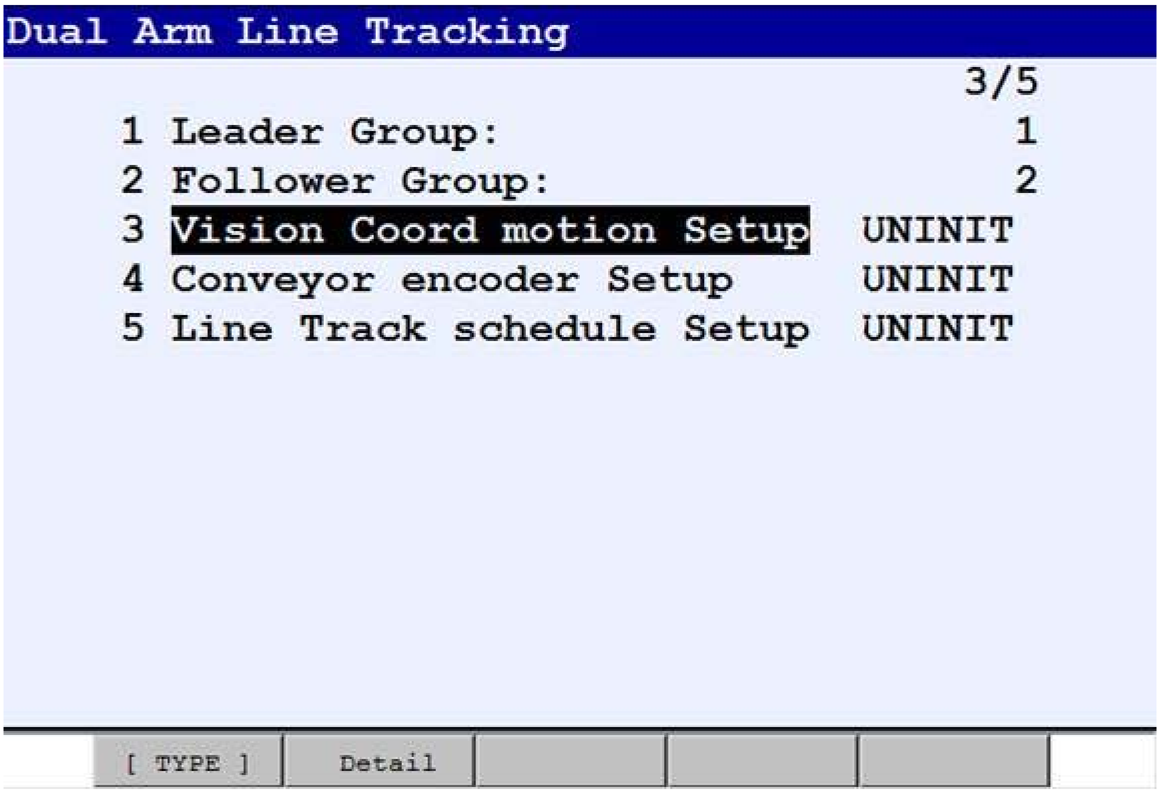
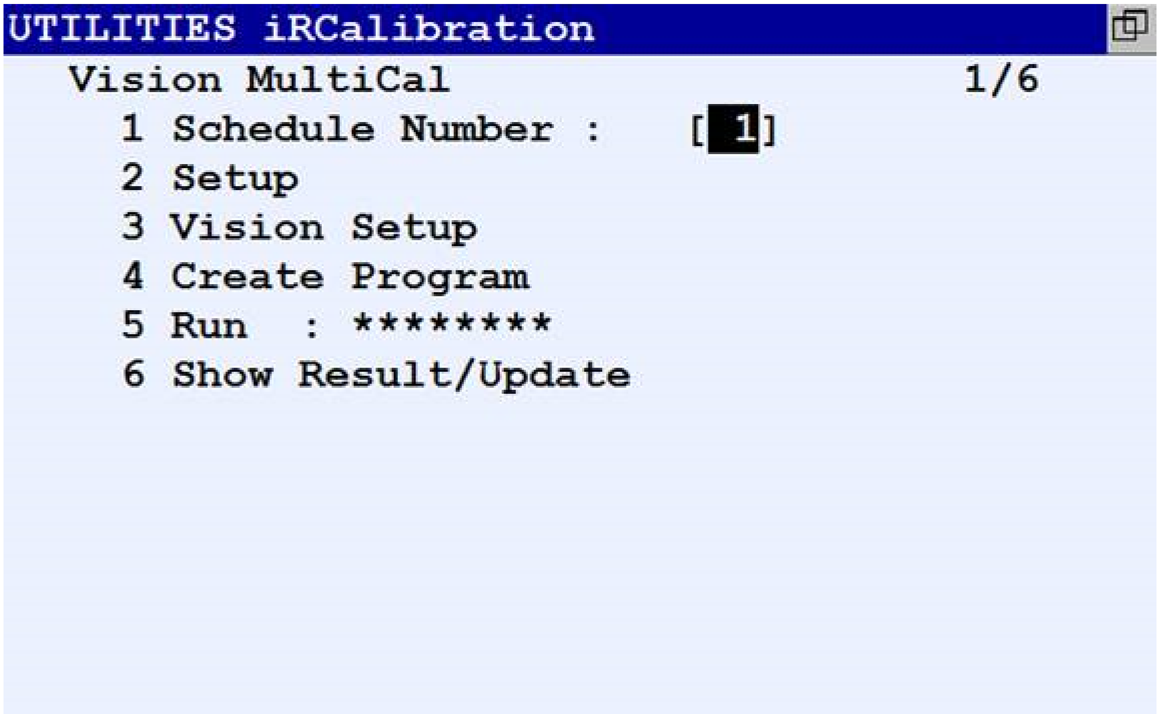


Figure 6–5. iRCalibration Vision MultiCal Setup Screen



6.6.5 Other Calibration Methods

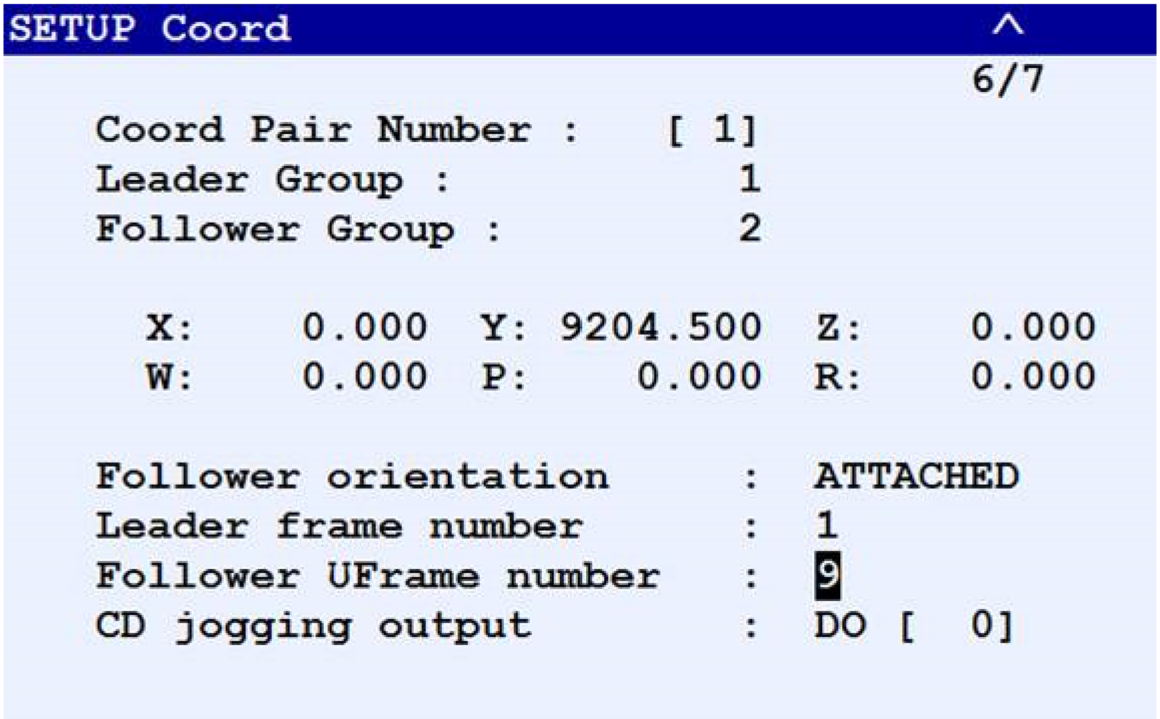
If you do not wish to use *i*RCalibration Vision MultiCal to calibrate the coordinated pair for Line Track Coord, other methods to do the calibration can also be used. However, accuracy is very important and the calibration should be done very carefully. For more information on the other methods, refer to the *Coordinated Motion Setup and Operations Manual*.

Caution

Inaccurate calibration may cause the real physical relationship between the two robots to vary even though it is programmed to stay the same, especially on moves with orientation changes. Therefore, inaccurate calibration can lead to dropping or damaging the part as well as other damage to equipment or personnel.

A properly set up Coord Pair is shown in Figure 6–6.

Figure 6–6. Example of a Properly Formatted Coord Pair



For more information, refer to “Coordinated Pair Setup” in the *Coordinated Motion Setup and Operations Manual*.

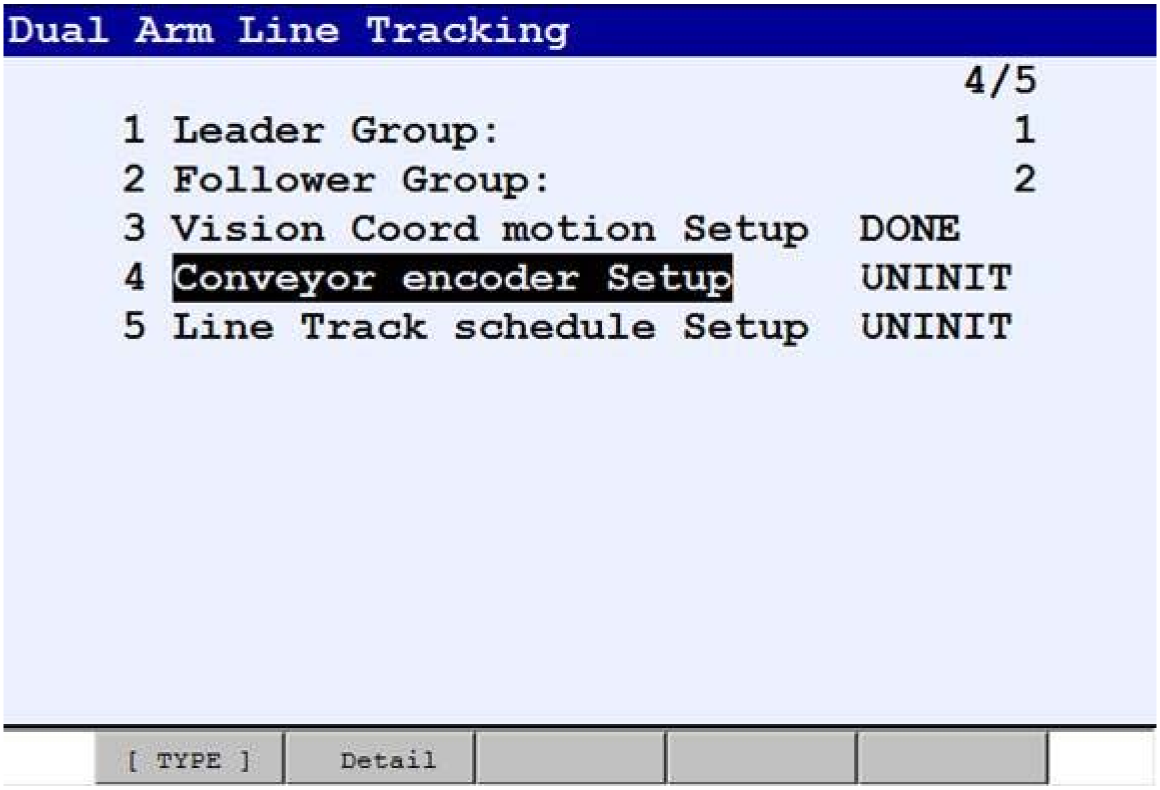
6.6.6 Conveyor encoder Setup

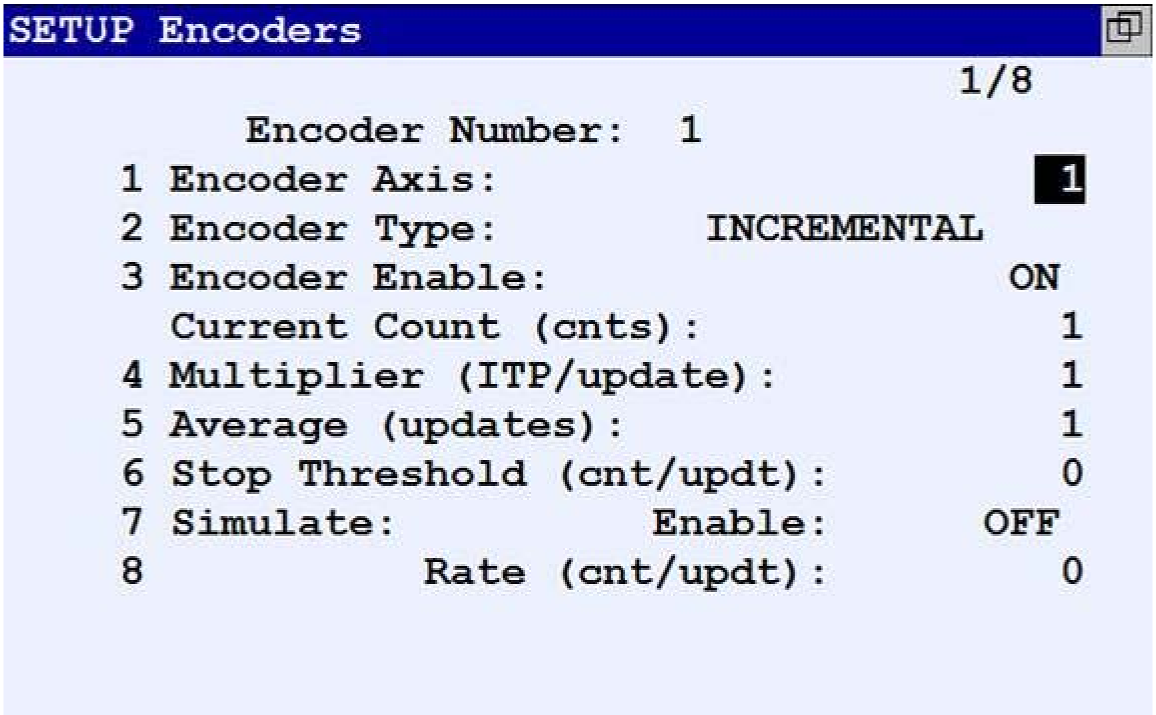
In order to set up the line tracking conveyor encoder, select ’4 Conveyor encoder Setup’ and hit Enter or the Detail softkey. Follow the instructions detailed in Section 3.1 to complete the encoder setup process.

The status of ’Conveyor Encoder Setup’ becomes DONE when (by default) Encoder 1 has a nonzero Encoder Axis and Encoder Enable is set to ON. For more information, refer to Section 3.1.

After the encoder is calibrated, a cold start message will be posted. Please cold start the controller at this time.

Figure 6–7. Line Track Coord Main Setup Screen to Select the Encoder Setup Screen



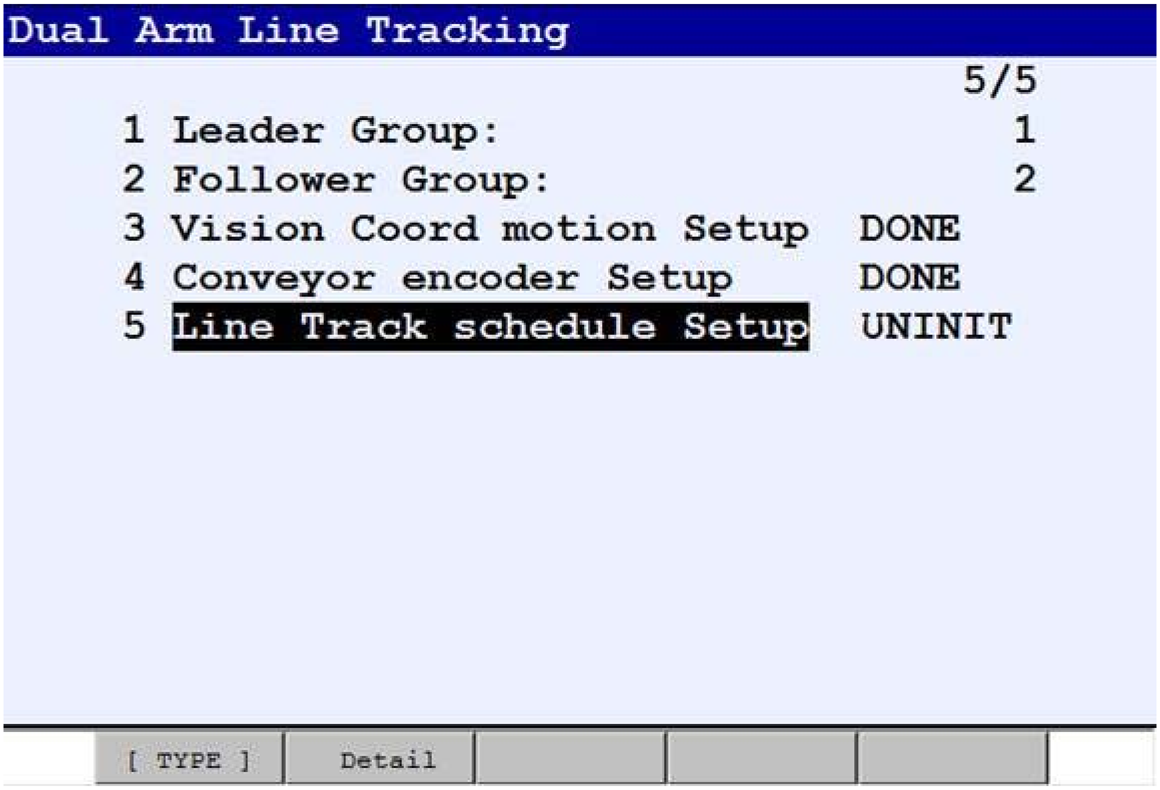


6.6.7 Line Track schedule Setup

In order to set up the line tracking schedule, select ’5 Line Track schedule Setup’ from the Line Track Coord setup screen and hit enter or the ’Detail’ softkey.

Follow the instructions in Section 3.3 to complete the schedule setup for your system.

Figure 6–8. Line Track Coord Main Setup Screen



A properly formatted Line Tracking Schedule is shown below:

Figure 6–9. Properly formatted Line Track Schedule Setup Screen for an Example Application

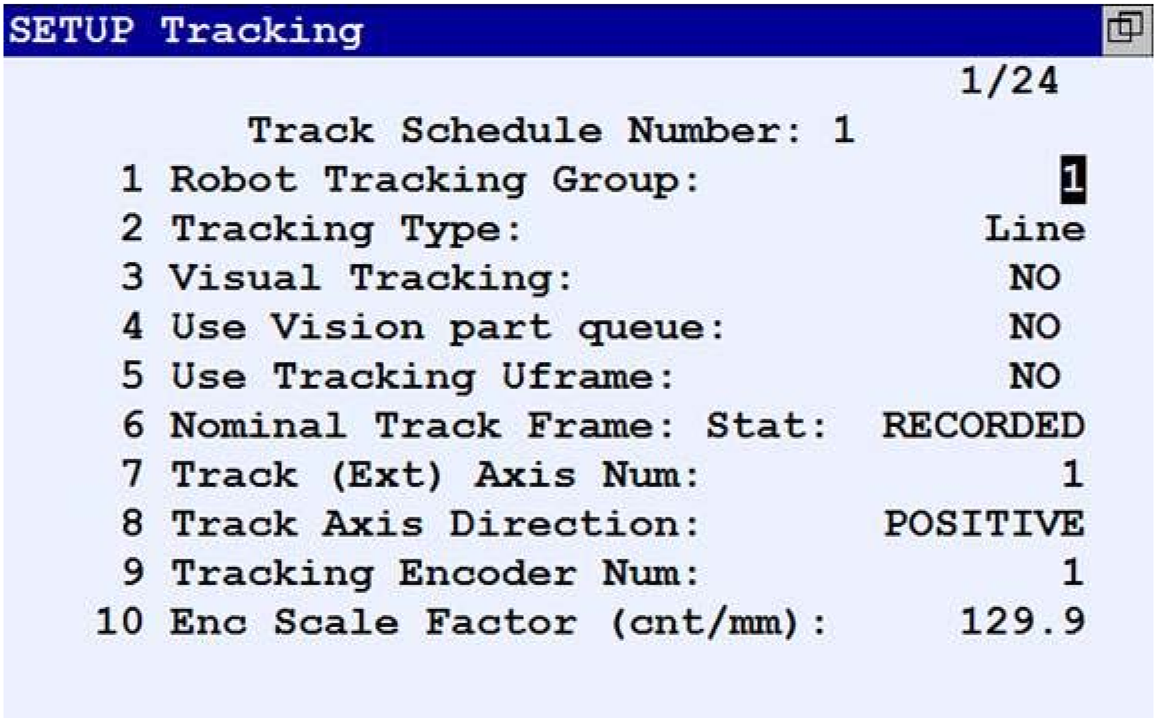
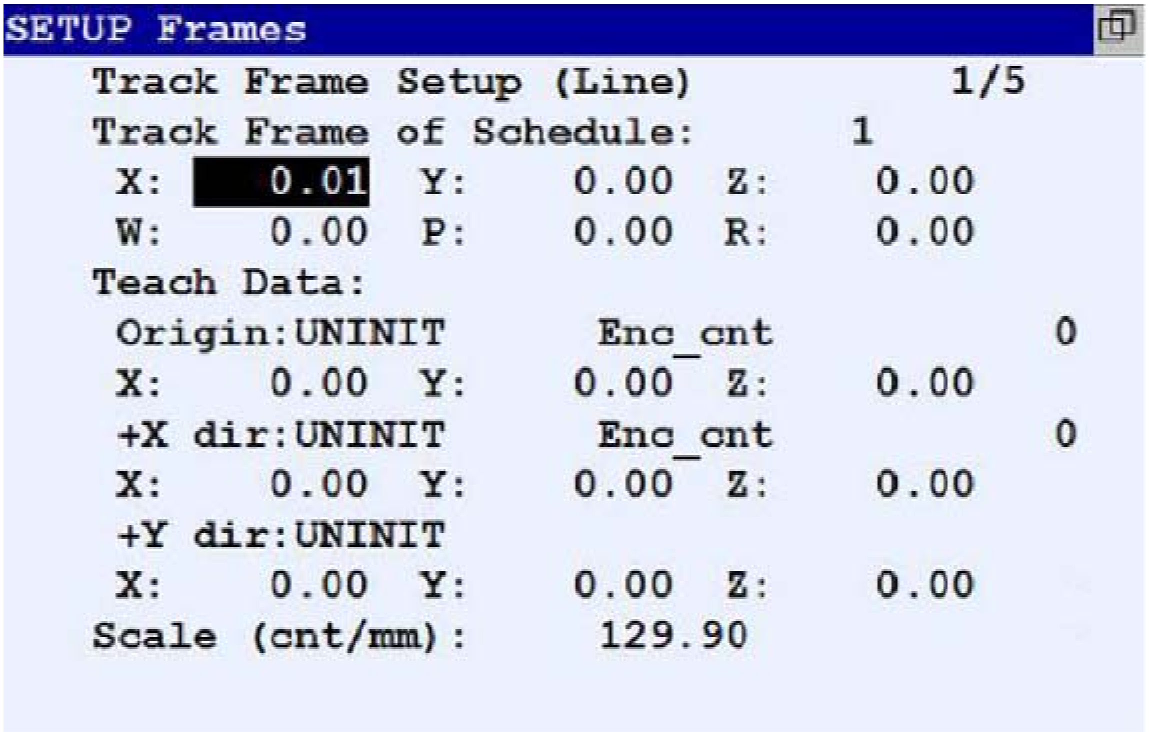


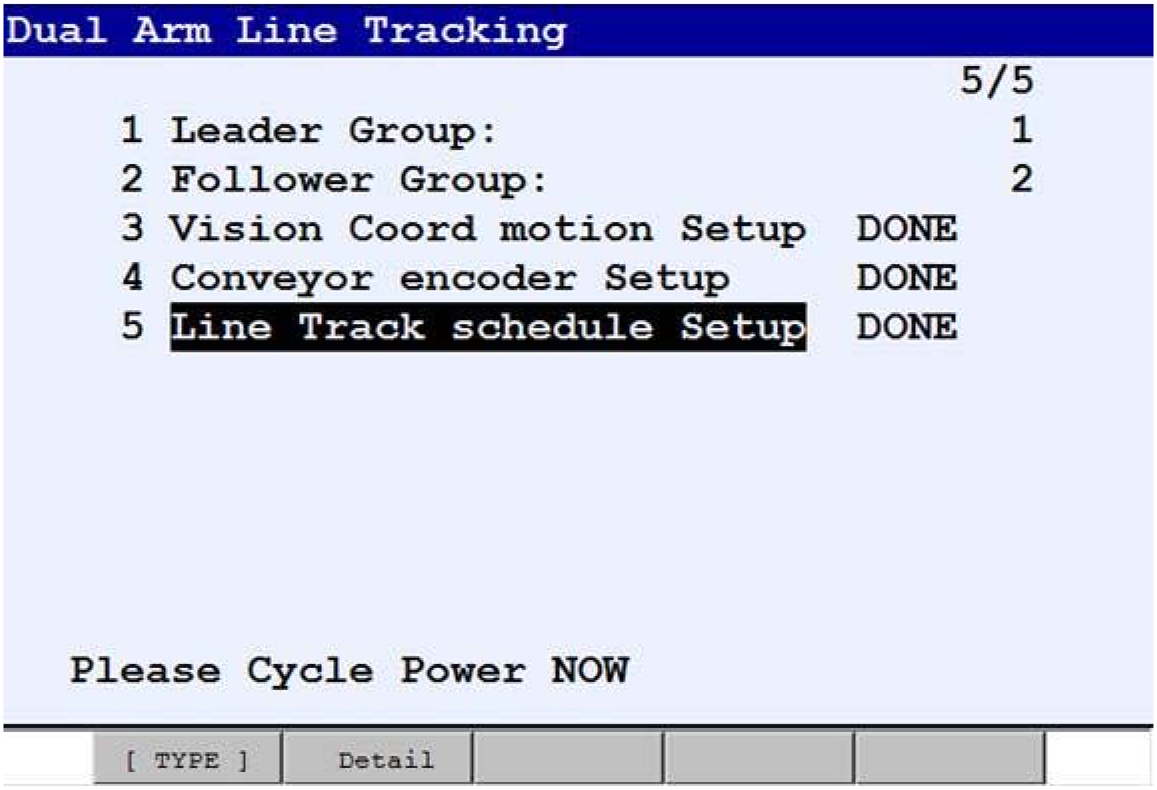
Figure 6–10. Correct Setup for when the Desired Tracking Frame is Coincident with the Leader's World Frame



Please note the following regarding setting up Line Tracking for Dual Arm Coordinated Line Tracking:

- '1 Robot Tracking Group' should be set to the Leader group of the coordinated pair, group 1 in our example application. Refer to Figure 6–3 for more detail.
- '6 Nominal Track Frame' should be set to some non-zero relationship. In the case that the desired tracking frame is coincident with the leader's world frame, please enter some small value for X, Y, or Z. See Figure 6–10 for a properly set up tracking frame for this case.
- '10 Enc Scale Factor (cnt/mm)' should be some value other than 500.0.
- The status for Line Track schedule Setup on the Line Track Coord setup screen will become 'Done' when the tracking frame is nonzero and when the Enc Scale Factor is anything but 500.0.

Figure 6–11. Line Track Coord Main Setup Screen



After completing Line Track schedule Setup restart the controller to allow the changes to take effect.

6.7 Jogging

There are four options for jogging the coordinated pair after it has been set up in the previous steps: Independent Jog, Coordinated Jog, Tracking Jog, and Mirror Jog. These can make teaching coordinated programs much easier to teach but care should be taken to understand each mode so that no unexpected motion occurs while holding a part.

6.7.1 Independent Jog

Each robot of the coordinated pair can be jogged independently.. For more information, refer to the *Handling Tool and MATE HandlingTool Setup and Operations Manual* .

Caution

Independently jogging robots does not maintain a coordinated relationship between the robots, and can therefore lead to dropping or damaging the part as well as other damage to equipment or personnel. Independent jogging while holding a part is not recommended.

6.7.2 Coordinated Jog

The coordinated pair can be jogged in a coordinated manner. In this mode, the relationship between the leader and follower is maintained throughout jogging. Refer to the *Coordinated Motion Setup and Operations Manual* for more information. This is the recommended jog mode while holding a part.

6.7.3 Tracking Jog

Tracking jog allows both of the coordinated robots to be jogged together simultaneously, applying the same command to each. However, this jog mode does not maintain the coordinated relationship between the robots. Please refer to the *R-30iB Controller MULTI ARM Controller Option Manual* for more information on this jogging mode.

Caution

Using tracking jog on the robots does not maintain a coordinated relationship between the robots and can therefore lead to dropping or damaging the part as well as other damage to equipment or personnel. Tracking jog while holding a part is not recommended.

6.7.4 Mirror Jog

Mirror jog is similar to tracking jog but a mirror plane can be defined such that the leader and follower movements are mirrored when moving in a direction orthogonal to the mirror plane. Mirror jog does not maintain a coordinated relationship between the robots. Please refer to the *Handling Tool and MATE HandlingTool Setup and Operations Manual* for more information on this jogging mode.

Caution

Using mirror jog on the robots does not maintain a coordinated relationship between the robots and can therefore lead to dropping or damaging the part as well as other damage to equipment or personnel. Mirror jog while holding a part is not recommended.

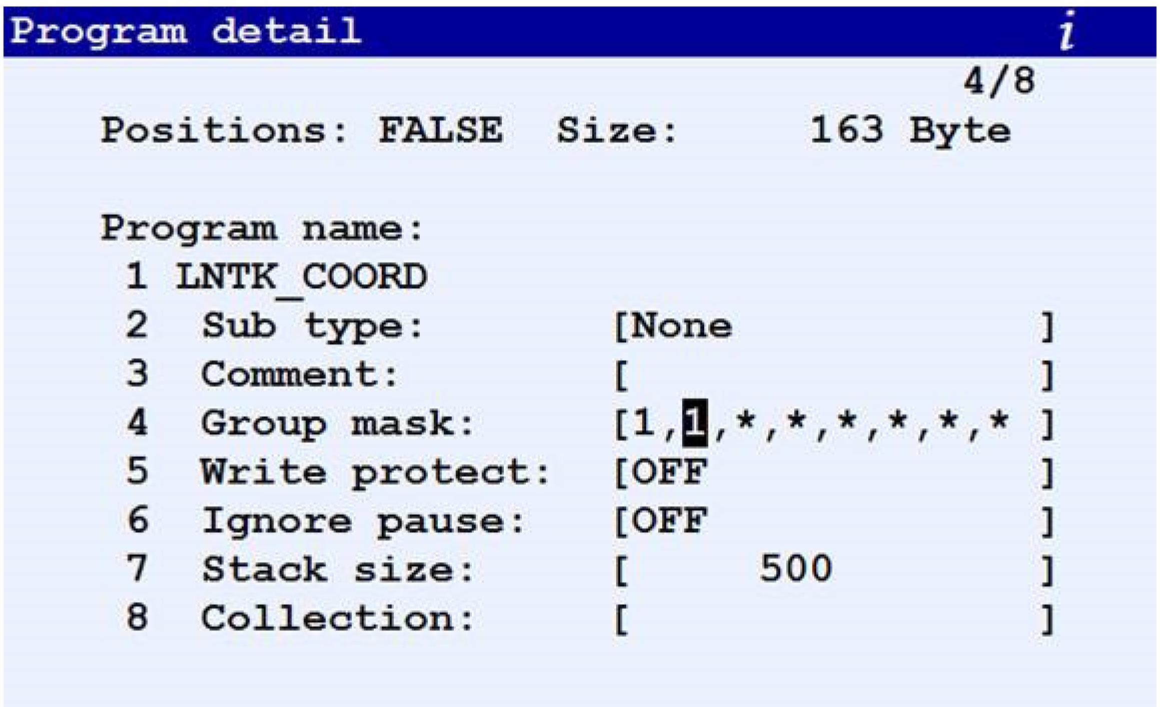
6.8 Programming

6.8.1 TP Program Setup

When creating the Line Tracking Coordinated Motion TP program, follow these guidelines when modifying the program details upon creation:

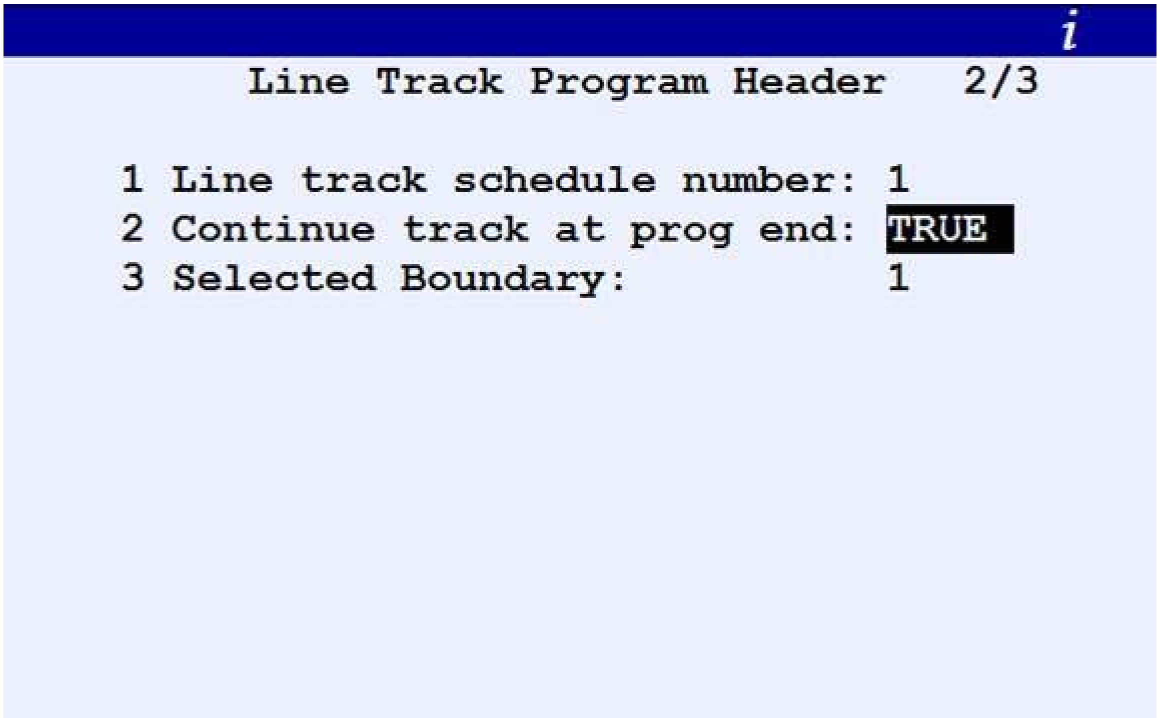
1) The group mask should include the two groups which constitute the desired coordinated pair to be used for line tracking. In our example application, the two desired groups are 1 and 2, the leader and follower respectively. Refer to Figure 6–3 for more detail.

Figure 6–12. Properly Formatted Program DETAIL screen for an Example Application



2) 'Continue track at prog end:' should be set to TRUE. See Figure 6–13.

Figure 6–13. Properly Formatted Line Track Program Header for an Example Application



6.8.2 Programming

While programming a coordinated motion line tracking program, observe the following guidelines:

- The last motion line preceding the Line Tracking Coordinated Motion program must be a non line-tracking COORD move.
- The first motion line succeeding the Line Tracking Coordinated Motion program must be a non line-tracking COORD move. Otherwise, the error 'LNTK-65 COORD needed in TRKCD trans' will be posted.
- Each motion move inside the Line Tracking Coordinated Motion program must have the COORD motion option at the end. Otherwise, the error 'LNTK-63 COORD is needed in TRKCD prog' will be posted.
- Each position recorded in the coordinated motion line tracking program must have the follower group's position recorded in the Dynamic UFrame. By default, this is UFrame 9 and it is selected automatically upon entering the coordinated motion line tracking program. If you're using a custom Coordinated Pair setup, use the value for 'Follower UFrame number' instead of 9. For more information, refer to the *Handling Tool and MATE HandlingTool Setup and Operations Manual*.

- Once the coordinated pair is in the relationship which is to remain fixed, CALL the included TP program ARMLOCK. This prevents operators from jogging either of the robots without Coordinated Jog.

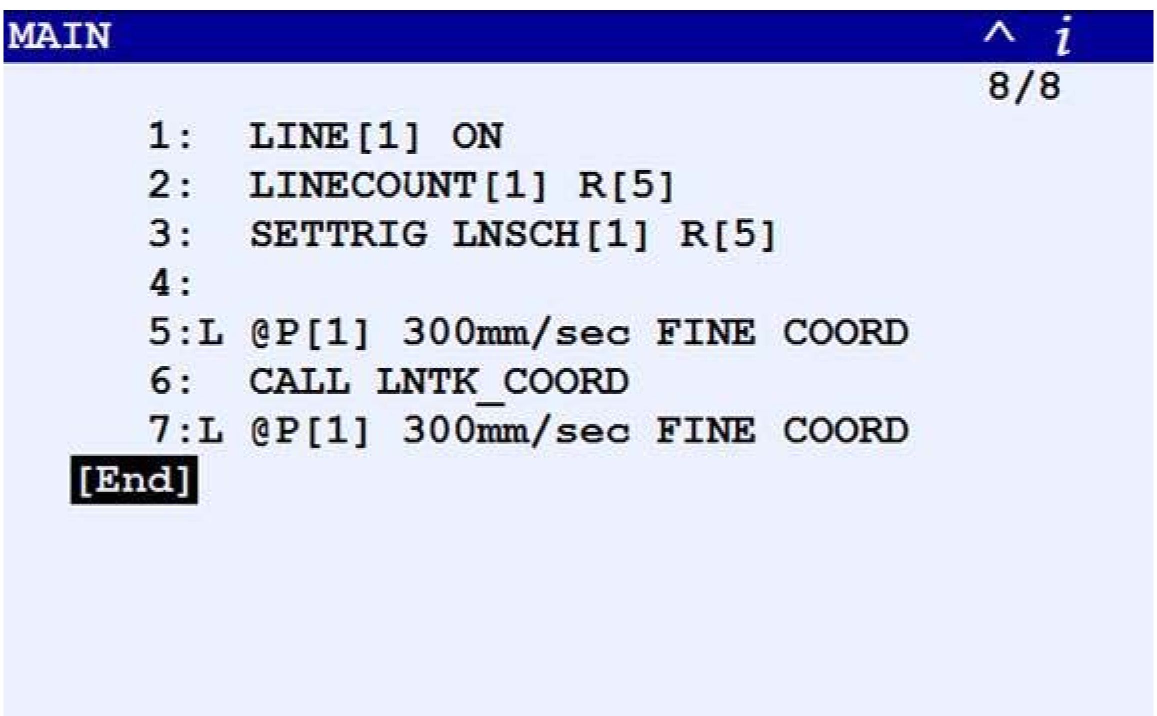
> **Caution**
>
> If the ARMLOCK program is not called at the proper time, the robots will be able to jog in a non-coordinated manner, leaving them susceptible to dropping or damaging the part as well as causing damage to other equipment or personnel.

- • Once the coordinated pair is no longer required to be in a fixed relationship, CALL the included TP program ARMUNLOCK. This allows all jogging modes once again.

Note Do not modify or tamper with ARMLOCK or ARMUNLOCK in any way.

The coordinated motion line tracking program should always be called from another, non line tracking program rather than being called directly. A sample main (not line tracking) program is displayed in Figure 6–14 .

Figure 6–14. Example Main Program (Not Line Tracking)



Lines 1-3 simulate a part detect.

Line 5 is the required COORD move before the first Line Tracking Coordinate Motion move.

Line 7 is the required COORD move after the last Line Tracking Coordinate Motion move.

Line 6 calls LNTK_COORD, the Line Tracking Coordinated Motion Program.

LNTK_COORD is now shown in Figure 6–15.

Figure 6–15. Example Line Track Coord Program (Line Tracking)

```
LNTK_COORD                                              ^ i
                                                        1/9
      1:   UFRAME_NUM[GP1]=0
      2:   UFRAME_NUM[GP2]=9
      3:L  P[1] 2000mm/sec FINE COORD
      4:   CALL ARMLOCK
      5:L  P[2] 2000mm/sec FINE COORD
      6:L  P[3] 2000mm/sec FINE COORD
      7:   CALL ARMUNLOCK
      8:L  P[4] 2000mm/sec FINE COORD
 [End]
```

Lines 1-2 set the UFrame correctly for each group. The follower group positions should be recorded and executed using the Dynamic UFrame. By default, this is UFrame 9. If you're using a custom Coordinated Pair setup, use the value for 'Follower UFrame number' instead of 9. For more information, refer to the *Handling Tool and MATE HandlingTool Setup and Operations Manual*.

Line 3 performs a move for which the relationship between the two robots is not required to stay fixed. In our example application, this might be a move to get under the car while it is on the moving conveyor. It might be a position similar to the one shown in Figure 6–16.

Figure 6–16. Lines 3 Commands the Robots to Pick Up the Part

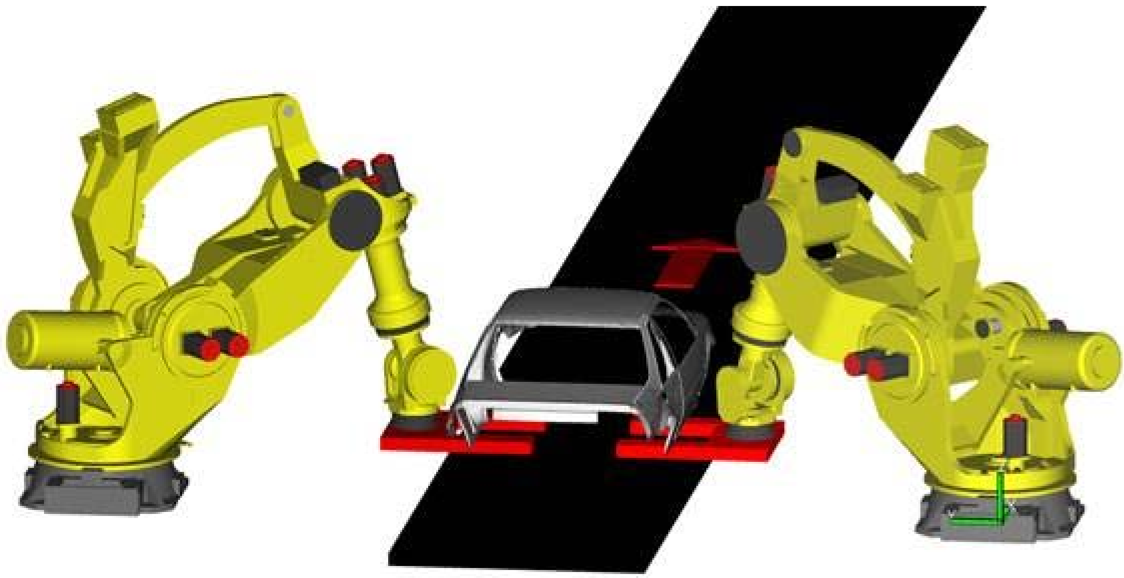
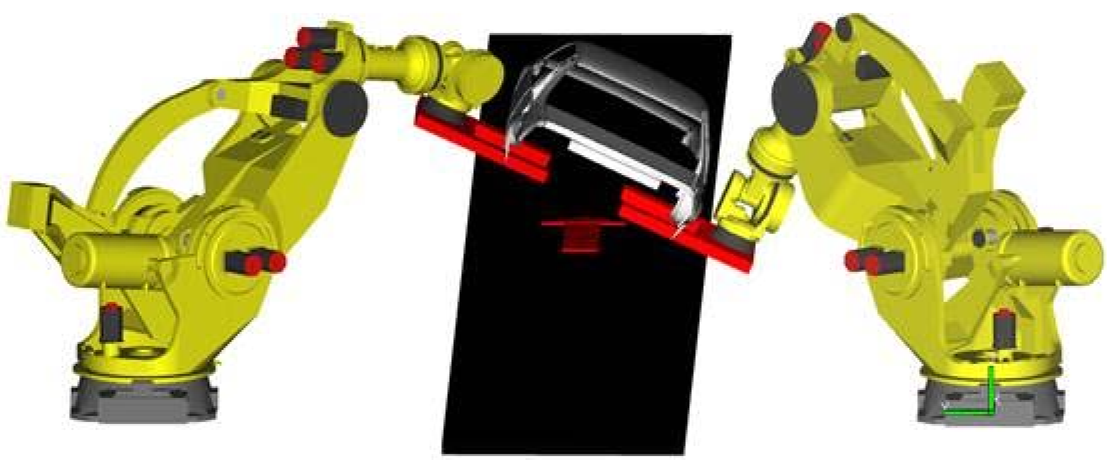


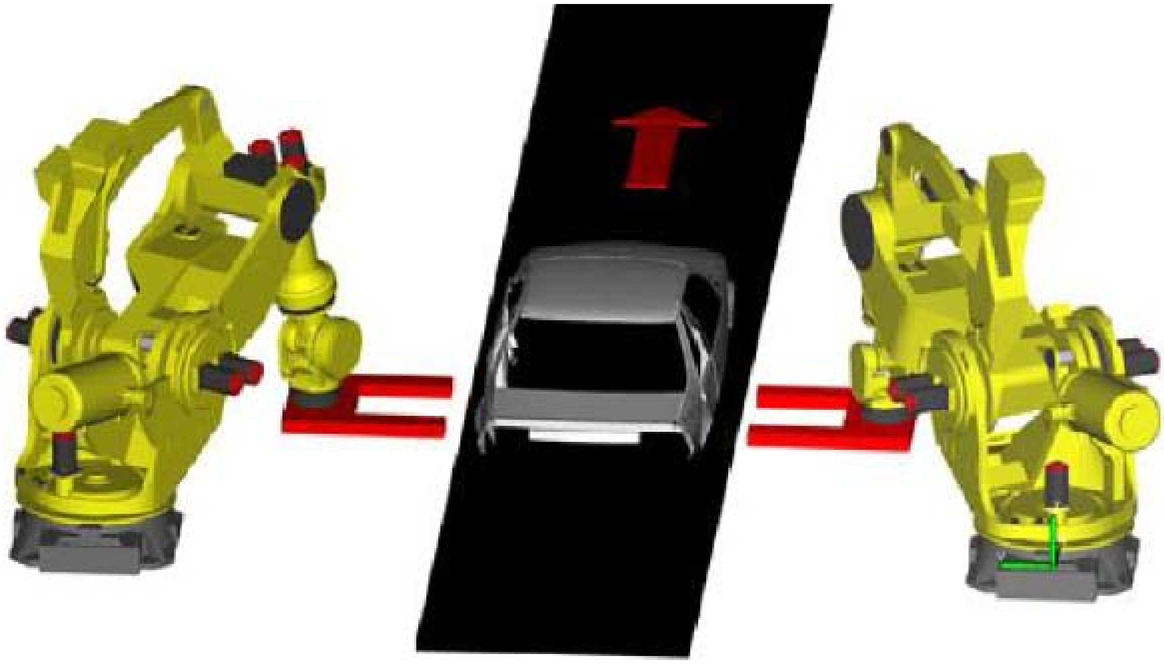
Figure 6–17. Line 5–6 Manipulate the Part but Maintain the Relationship Between the Leader and the Follower While Line Tracking



Line 4 is called once the robots have moved into the position where their relationship is to stay fixed. In the example application, this is when the robots are holding the car body.

Lines 5 - 6 are motion lines which are required to maintain the same relationship. In the example application, this might be picking the car up off the conveyor, manipulating it, and then putting it down. If the program is paused during these lines, the operator will only be able to jog in the Coordinated Jog mode. An example position is shown in Figure 6–17 .

Figure 6–18. Line 8 Commands the Robots to Move Away from the Vehicle, Not Requiring the Relationship Between the Leader and Follower to Stay Fixed



Line 7 is called once the robots are no longer required to maintain a fixed relationship. In the example application, this is when the robots are no longer holding the car.

Line 8 is a move where the relationship between the robots is not required to stay fixed. This might be a move away from the car while line tracking to avoid collisions with the conveyor stanchions. The position might look like the one shown in Figure 6–18.

Each line is a COORD move.

6.9 Error Codes

Refer to the *Error Code Manual* for more information.

6.9.1 Line Track Coord Error Codes

6.9.1.1 LNTK-061 Not a valid trkcd group mask

Cause: This tracking program group mask does not match $cd_param.$mh_rob_gmsk

Remedy: Change program group mask

6.9.1.2 LNTK-063 COORD is needed in TRKCD prog

Cause: All the motion line inside a trkcd program needs to be COORD motion option

Remedy: Add Coord motion option to every motion line

6.9.1.3 LNTK-065 COORD needed in TRKCD trans

Cause: Transition from TRKCD needs to be a COORD motion

Remedy: Add Coord motion option to TRKCD Transition motion

6.9.1.4 LNTK-066 TRKCD Trans nxtang error

Cause: Transition of TRKCD program has problem get leader joint angle

Remedy: Contact FANUC with cpdebug information

Appendix A

ERROR CODES

Contents

When an error occurs while the robot is running a tracking program, that error code is displayed on the teach pendant. Refer to the *Error Code Manual* for a listing of error codes, their causes and suggested recovery methods.

Appendix B

CONVEYOR LIMITS AND TRACKING ACCURACY

Contents

B.1 CONVEYOR SPEED LIMIT

The *conveyor speed limit* determines how fast parts on the conveyor can travel through the robot workspace in a tracking application.

Conveyor speed can be limited by the robot speed and the cycle time to complete the desired task. Conveyor speed can also be limited by the tracking accuracy error tolerance, (imposed by the process) which limits errors due to tracking inaccuracies. The maximum conveyor speed can be computed using the following equation.

$$MaxConvSpeed_{mm/sec} = \frac{MaxError_{mm}}{ITPtime_{sec}}$$

For example, to obtain a tracking error of less than ±4 mm on a robot with an ITP time of 20 msec, the maximum conveyor speed would be:

$$200_{mm/sec} = \frac{4_{mm}}{20_{msec}}$$

A conveyor speed of 12 m/min or 200 mm/sec is approaching the maximum speed for many processes. However, there are some processes which can successfully use faster conveyor speeds.

When you use ACCUTRIG, the tracking error is ± one system tick (4 ms) so the conveyor can be much faster. However, this is bound by the process speed. Refer to Section 4.5 .

B.2 CONVEYOR ACCELERATION LIMIT

The *conveyor acceleration/deceleration limit* determines how fast the speed of the conveyor can change, during robot tracking operations. Typically the robot should be able to accelerate at least twice as fast as the conveyor. The maximum conveyor acceleration can be computed using the following equation.

$$MaxConvAccel_{mm/sec}{}^{2} = \frac{MinRobotAccel}{2}$$

For example:

$$200_{mm/sec}{}^{2} = \frac{400_{mm/sec}{}^{2}}{2}$$

The value of 200 $_{mm/sec}{}^{2}$ is a general limit.

B.3 TRACKING ACCURACY

Tracking accuracy is the maximum tracking error offset you can expect in an application. The tracking accuracy is a function of synchronizing the part detect switch and the encoder-read/set-trigger operation.

The part might trip the part detect switch (trigger) at any time, and not be synchronized with the controller interpolation cycle in any way. The controller will detect this within one ITP time. If ACCUTRIG is used, the controller will detect this within one system tick (4 ms). Refer to Section 4.5 .

However, even under the best of conditions it might take up to one additional ITP time to read and store the value of the encoder (to perform the set trigger operation). The value that is read is the value stored during the last encoder data update.

Therefore, there can be synchronization delays of up to 2 ITP times for this operation; and more if the encoder update time multiplier is set larger than 1. The maximum error can be computed using the following equation.

$$ErrorDist_{mm} = Conv.Speed_{mm/sec} * 2 * ITPtime_{sec}$$

For example:

$$11.2_{mm} = 200_{mm/sec} * 2 * 0.028_{sec}$$

When ACCUTRIG is used, the system tick would be recorded when the part detect switch is triggered. The system then finds the encoder value at that system tick. Therefore, the formula is

$$ErrorDist_{mm} = Conv.Speed_{mm/sec} * 2 * System\ Tick_{ms}$$

For example:

$$1.6_{mm} = 200_{mm/sec} * 2 * 4_{ms}$$

However, since the relative tracking position can be changed by adjusting the prediction times, the accuracy can be expressed as ±1 ms. The tracking accuracy can be computed using the following equation.

$$+/-Accuracy_{mm} = Conv.Speed_{mm/sec} * 1\ ms$$

For example:

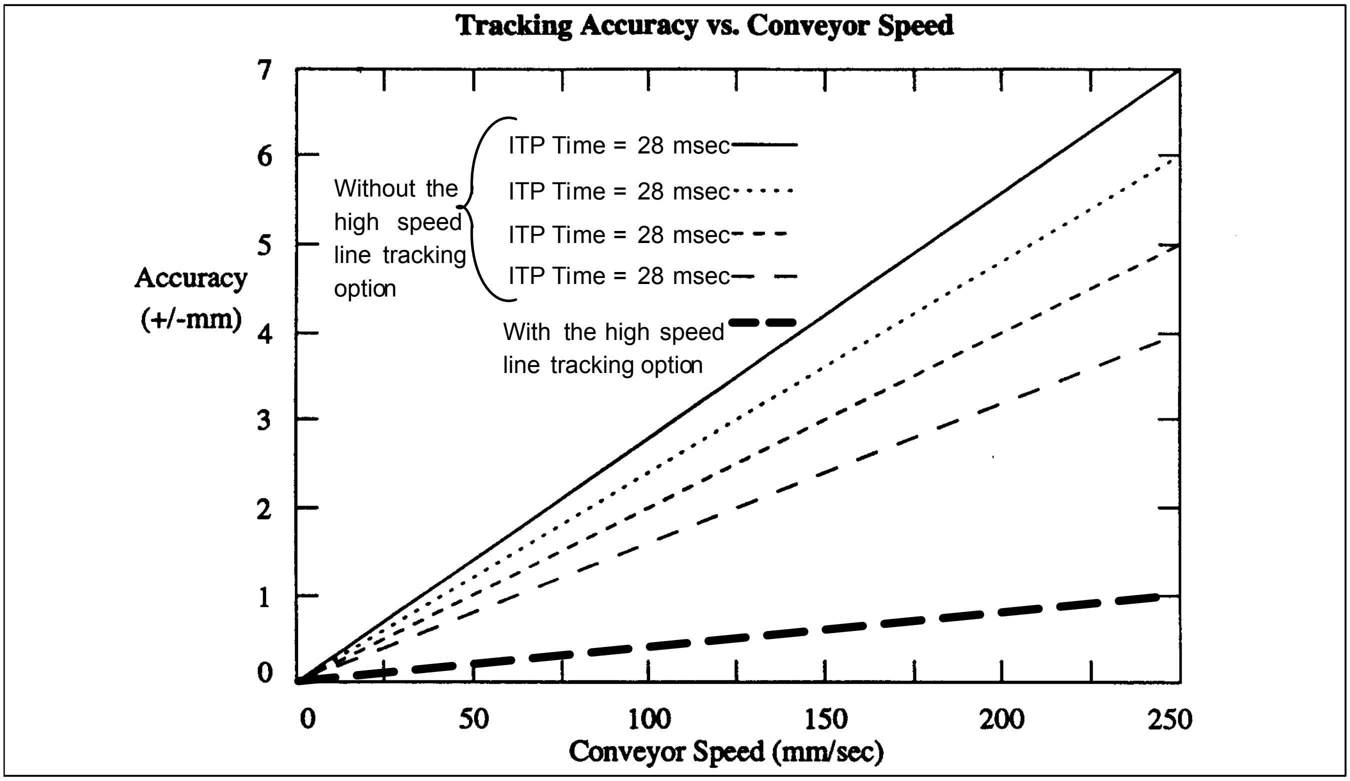
$$\pm 0.2_{mm} = 200_{mm/sec} * 1_{ms}$$

Note The larger ErrorDist value will always be used.

Note PaintTool disables the ACCUTRIG tracking accuracy feature.

Figure B–1 contains a graph which displays conveyor speed and level of accuracy, for various robot ITP times.

Figure B–1. Tracking Accuracy -vs- Conveyor Speed



The tracking accuracy can also be limited by the resolution of the encoder being used, and any gear ratio associated with it. The resolution is a combination of the encoder scale factor and the conveyor speed. The resolution can be computed using the following equation.

$$Resolution_{\ pulses/update} = ScaleFactor_{\ pulses/mm} * Conv.Speed_{\ mm/sec} * UpdateTime_{\ sec/update}$$

For example, in a system with an encoder scale factor of 10 $_{pulses/mm}$ and an ITP time (update time) of 20 $_{msecs}$, the resolution for a conveyor speed of 200 $_{sec/update}$ is:

$$40_{pulses/update} = 10_{pulses/mm} * 200_{mm/sec} * 0.020_{sec/update}$$

In a system that uses high speed line tracking, an encoder scale factor of 10 $_{pulses/mm}$ and an update time of 4 $_{msecs}$, the resolution for a conveyor speed of 200 $_{sec/update}$ is:

$$8_{\ pulses/update} = 10_{\ pulses/mm} * 200_{\ mm/sec} * 0.004_{\ sec/update}$$

Note Keep the encoder resolution above 10 $_{pulses/update}$. Values around 40 or 50 are more desirable.

B.4 RESOLVING AGGRESSIVE MOTION DURING TRACKING ON AN R-30*i*A CONTROLLER

When the robot seems to move too aggressively during tracking on an R-30*i*A controller, one possible reason is low encoder resolution. The other possible reason is short tracking filter length.

For software versions before V7.30P07

- Use a hardware solution: add a gear between the motor and the encoder to boost encoder resolution to a desirable range. A higher encoder resolution would be 40-50 pulses/update.
- Use a software solution: If you do not want to use hardware method to boost encoder resolution, and have used this same application setup (with low encoder resolution) on an R-J3*i*B controller and were satisfied with the accuracy result, you can try setting $CPCFG.$CP_ENABLE to FALSE, and then cycle power. This would make the controller fall back to R-J3*i*B functionality.

Note By changing $CPCFG.$CP_ENABLE to FALSE, several new line tracking functions such as stop tracking and the skip outbound function will not be available.

For software version V7.30P07 or later

- Use a hardware solution: add a gear between the motor and the encoder to boost encoder resolution to a desirable range. A higher encoder resolution would be 40-50 pulses/update.
- Use a software solution1: If you do not want to use the hardware method to boost encoder resolution, and have used this same application setup (with low encoder resolution) on an R-J3*i*B controller and were satisfied with the accuracy result, you can try setting $CPCFG.$CP_ENABLE to FALSE, and then cycle power. This would make the controller fall back to R-J3*i*B functionality.

Note By changing $CPCFG.$CP_ENABLE to FALSE, several new line tracking functions such as stop tracking and the skip outbound function will not be available.

- Use software solution2: Enable the software gear by turning on bit 3 of $LNCFG.$COMP_SW (Add value 8 if it was not turned on before). Then cycle the power. After that because the gear ratio would not be the same any more, you need to re-teach the scale of the tracking schedule using F2, TEACH on the SETUP Tracking menu. Refer to for more information. Or re-enter the scale (cnt/mm or cnt/deg) in the tracking schedule that uses the encoder by multiplying the previous value of the teach scale by the value of $ENC_SCALE[x] where x is the encoder number in the tracking schedule.

The Solution for Short Tracking Filter

This occurs when the robot moves from a stationary position and moves to the part on the conveyor the tracking filter is used. The filter length is the time robot has to catch up with the parts on the conveyor. When the filter length is short, the robot is heavy and the conveyor speed is high, the robot will be commanded to accelerate too fast. The motion would look aggressive if the robot can accelerate. A collision alarm would occur if it is beyond the robot's capability.

Therefore, for a large robot and high conveyor speed you should adjust $LNSCH[].$TRK_FLTR_LN (default 7, maximum 40) accordingly to a higher value such as 15 or 20.

Appendix C

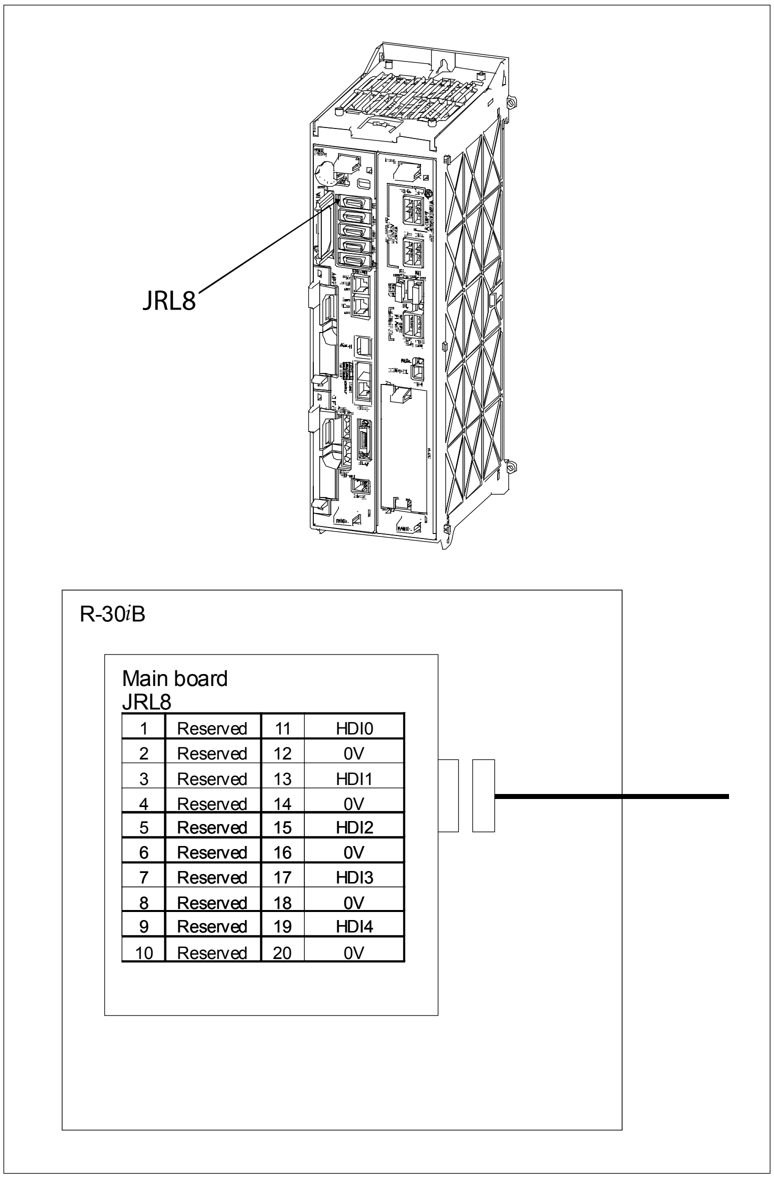
SCHEMATICS

Contents

C.1 OVERVIEW

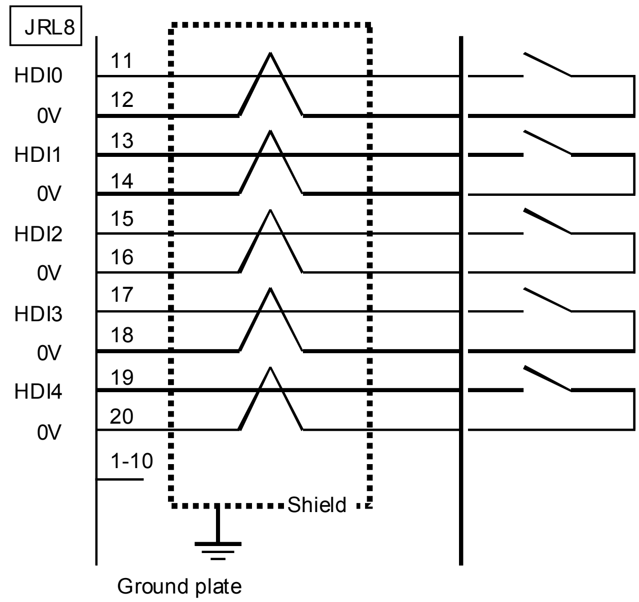
This section contains the schematic drawings of cables used for the HDI interface and line tracking encoders. The HDI signals are used in combination with special application software. The HDI signals cannot be used as general-purpose DIs.

Figure C–1. HDI Interface



1	Reserved	11	HDI0
2	Reserved	12	0V
3	Reserved	13	HDI1
4	Reserved	14	0V
5	Reserved	15	HDI2
6	Reserved	16	0V
7	Reserved	17	HDI3
8	Reserved	18	0V
9	Reserved	19	HDI4
10	Reserved	20	0V

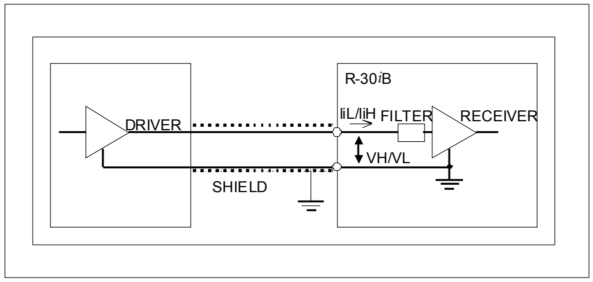
Figure C–2. HDI Cable Connections



Recommended cable connector:
PCR-E20FA (Honda Tsushin Kogyo)
FI30-20S (Hirose Electric)
FCN-247J020-G/E (Fujitsu)
52622-2011 (Molex Japan)

Figure C–3. Input Signal Rules for the High-Speed Skip (HDI)

Absolute maximum rating
Input voltage range Vin: -3.6 to +10 V
Input characteristics



Unit	Symbol	Specification	Unit	Remark
High level input voltage	VH	3.6 to 11.6	V	
Low level input voltage	VL	0 to 1.0	V	
High level input current	IiH	2 max	mA	Vin=5 V
		11 max	mA	Vin = 10 V
Low level input current	IiL	-8.0 max	mA	Vin = 0 V
Input signal pulse duration		20 min	*m* s	
Input signal delay or variations		0.02(max)	ms	

NOTE

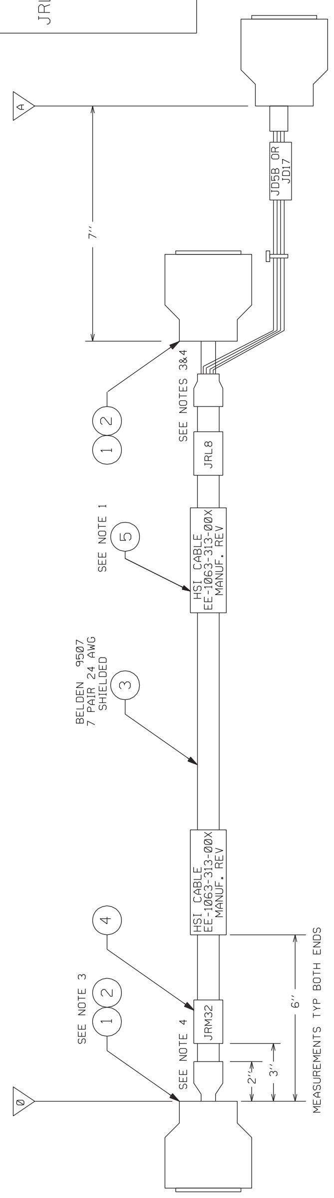
1 The plus (+) sign of IiH/IiL represents the direction of flow into the receiver. The minus (-) sign of IiH/IiL represents the direction of flow out of the receiver.

2 The high-speed skip signal is assumed to be 1 when the input voltage is at the low level and 0 when it is at the high level.

Figure C–4. R-30iB HSI Cable (EE-1063-313-001 & 002)

ECN NO.	SYM.	CHANGE	DATE	BY	CHK.
E120610	A	PRODUCTION RELEASE	9/12	JEA	CH

REV LEVEL	CABLE VERSION	DIM. A	PG
A	EE-1063-313-001	48 IN 1219 mm	NO
A	EE-1063-313-002	97 IN 2464 mm	NO



JRL8 CONNECTOR CHART

LABEL	PIN NUMBER	WIRE COLOR	WIRE PAIR	SIGNAL NAME
JRL8	1			
	2			
	3			
	4			
	5			
	6			
	7			
	8			
	9			
	10			
	11	RED	1	HDI-0
	12	BLACK		0V
	13	BLUE	4	HDI-1
	14	BLACK		0V
	15	WHITE	2	HDI-2
	16	BLACK		0V
	17	YELLOW	5	HDI-3
	18	BLACK		0V
	19			
	20			

JD5B/JD17 CONNECTOR CHART

LABEL	PIN NUMBER	WIRE COLOR	WIRE PAIR	SIGNAL NAME
JD5B JD17	1			
	2			
	3			
	4			
	5			
	6	BLACK	3	0V
	7			
	8			
	9			
	10	GREEN	3	24VE
	11			
	12			
	13			
	14			
	15			
	16	BLACK	6	0V
	17			
	18			
	19	BROWN	6	24VE
	20			

STRAIGHT THROUGH CABLE CHART

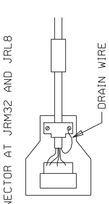
LABEL	PIN NUMBER	WIRE COLOR	WIRE PAIR	SIGNAL NAME
JRM32	1	RED	1	HDI-0
	2	BLACK		0V
	3	WHITE	2	HDI-2
	4	BLACK		0V
	5			
	6			
	7			
	8			
	9	BLACK	3	0V
	10	GREEN		24VE
	11	BLUE	4	HDI-1
	12	BLACK		0V
	13	YELLOW	5	HDI-3
	14	BLACK		0V
	15			
	16			
	17			
	18			
	19	BROWN	6	24VE
	20	BLACK		0V

ASSEMBLY NOTES:

1. PART NUMBER SHOWN ON CABLE LABEL MUST CORRESPOND TO THE LENGTH AND VERSION NUMBER IN TABLE. VENDOR NAME AND REVISION LEVEL MUST ALSO BE INCLUDED ON CABLE LABEL.

2. ▽ DENOTES LENGTH MEASUREMENT POINTS. SEE TABLE FOR DIMENSION. ALL LENGTH DIMENSIONS ±½IN.

3. CONNECT SHIELD DRAIN WIRE TO SHELL OF CONNECTOR AT JRM32 AND JRL8



4. CUT OFF SPARES BOTH ENDS PAIR 7 ORANGE / BLACK

Figure C–5. R-30iB Mate HSI Cable (EE-1063-314-001)

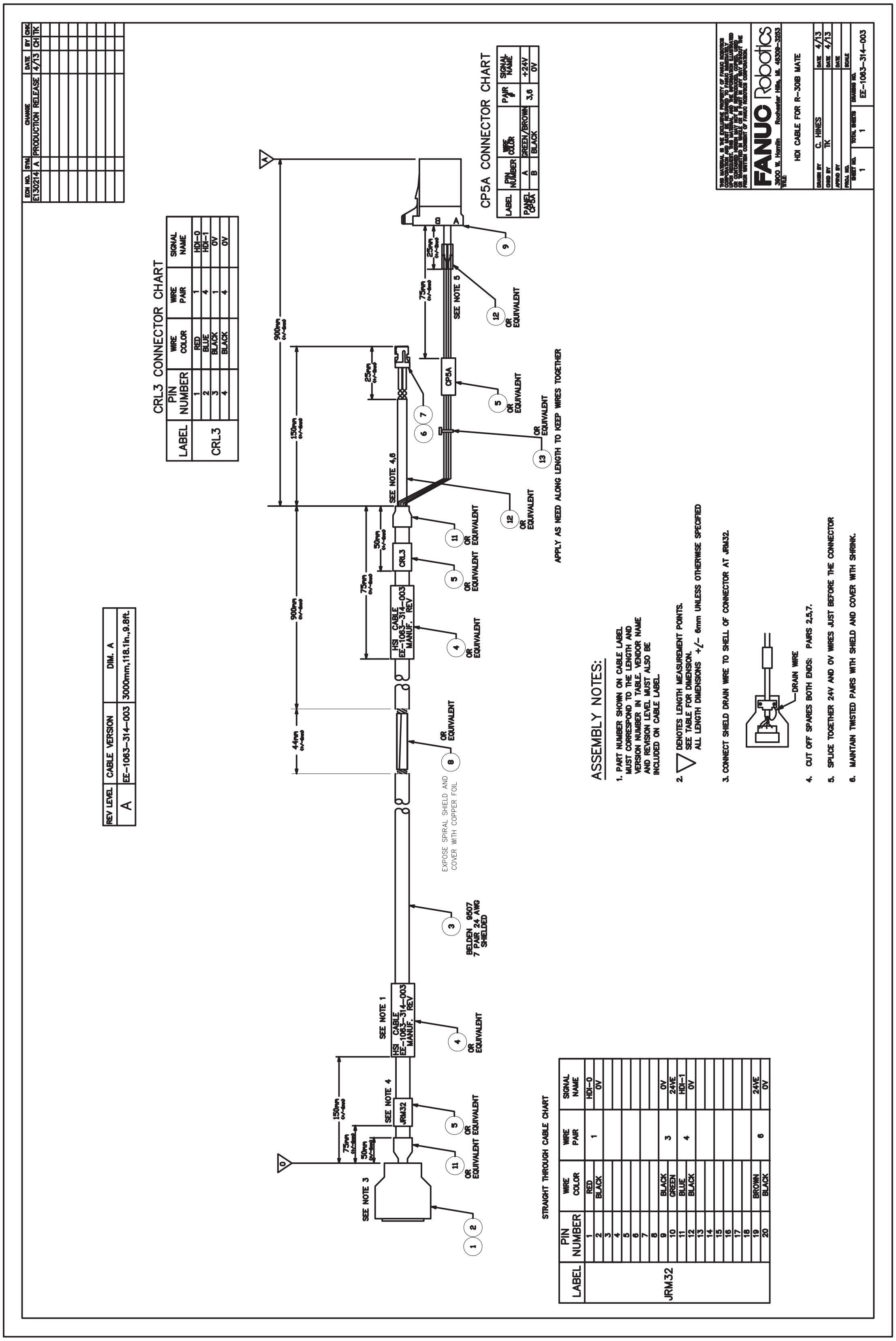
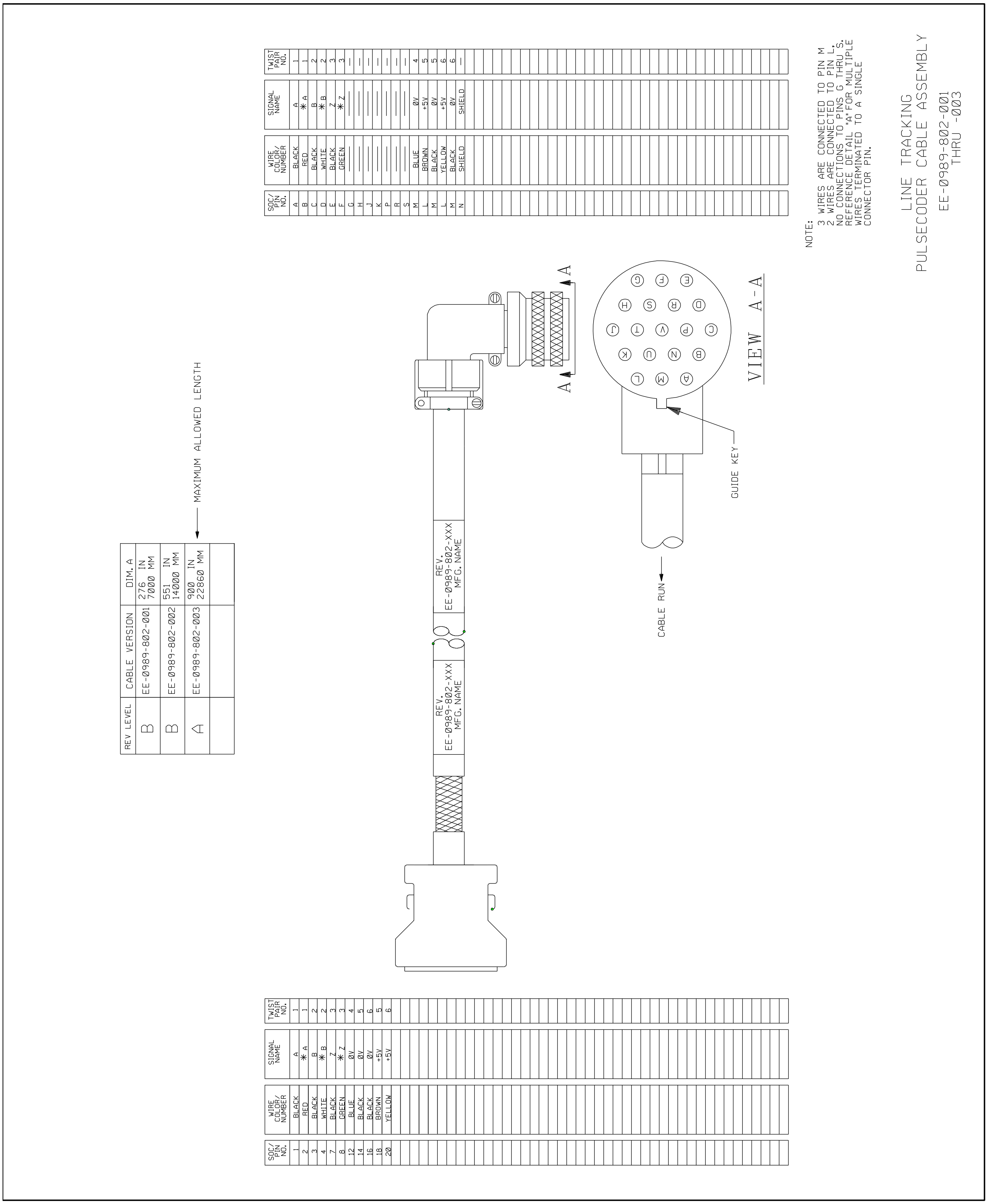


Figure C–6. Line Tracking Pulse Coder Cable Assembly EE-0989-802-00X (for use with Absolute Pulse Coder A860–0324–T101)



Glossary

A

abort

Abnormal termination of a computer program caused by hardware or software malfunction or operator cancellation.

absolute pulse code system

A positional information system for servomotors that relies on battery-backed RAM to store encoder pulse counts when the robot is turned off. This system is calibrated when it is turned on.

A/D value

An analog to digital-value. Converts a multilevel analog electrical system pattern into a digital bit.

AI

Analog input.

AO

Analog output.

alarm

The difference in value between actual response and desired response in the performance of a controlled machine, system or process. Alarm=Error.

algorithm

A fixed step-by-step procedure for accomplishing a given result.

alphanumeric

Data that are both alphabetical and numeric.

AMPS
Amperage amount.

analog
The representation of numerical quantities by measurable quantities such as length, voltage or resistance. Also refers to analog type I/O blocks and distinguishes them from discrete I/O blocks. Numerical data that can vary continuously, for example, voltage levels that can vary within the range of -10 to +10 volts.

AND
An operation that places two contacts or groups of contacts in series. All contacts in series control the resulting status and also mathematical operator.

ANSI
American National Standard Institute, the U.S. government organization with responsibility for the development and announcement of technical data standards.

APC
See absolute pulse code system.

APC motor
See servomotor.

application program
The set of instructions that defines the specific intended tasks of robots and robot systems to make them reprogrammable and multifunctional. You can initiate and change these programs.

arm
A robot component consisting of an interconnecting set of links and powered joints that move and support the wrist socket and end effector.

articulated arm
A robot arm constructed to simulate the human arm, consisting of a series of rotary motions and joints, each powered by a motor.

ASCII
Abbreviation for American Standard Code for Information Interchange. An 8-level code (7 bits plus 1 parity bit) commonly used for the exchange of data.

automatic mode
The robot state in which automatic operation can be initiated.

automatic operation
The time during which robots are performing programmed tasks through unattended program execution.

axis

1. A straight line about which a robot joint rotates or moves. 2. One of the reference lines or a coordinate system. 3. A single joint on the robot arm.

B

backplane

A group of connectors mounted at the back of a controller rack to which printed circuit boards are mated.

BAR

A unit of pressure equal to 100,000 pascals.

barrier

A means of physically separating persons from the restricted work envelope; any physical boundary to a hazard or electrical device/component.

battery low alarm

A programmable value (in engineering units) against which the analog input signal automatically is compared on Genius I/O blocks. A fault is indicated if the input value is equal to or less than the low alarm value.

baud

A unit of transmission speed equal to the number of code elements (bits) per second.

big-endian

The adjectives big-endian and little-endian refer to which bytes are most significant in multi-byte data types and describe the order in which a sequence of bytes is stored in a computer's memory. In a big-endian system, the most significant value in the sequence is stored at the lowest storage address (i.e., first). In a little-endian system, the least significant value in the sequence is stored first.

binary

A numbering system that uses only 0 and 1.

bit

Contraction of binary digit. 1. The smallest unit of information in the binary numbering system, represented by a 0 or 1. 2. The smallest division of a programmable controller word.

bps

Bits per second.

buffer

A storage area in the computer where data is held temporarily until the computer can process it.

bus

A channel along which data can be sent.

bus controller

A Genius bus interface board for a programmable controller.

bus scan

One complete communications cycle on the serial bus.

Bus Switching Module

A device that switches a block cluster to one bus or the other of a dual bus.

byte

A sequence of binary digits that can be used to store a value from 0 to 255 and usually operated upon as a unit. Consists of eight bits used to store two numeric or one alpha character.

C

calibration

The process whereby the joint angle of each axis is calculated from a known reference point.

Cartesian coordinate system

A coordinate system whose axes (x, y, and z) are three intersecting perpendicular straight lines. The origin is the intersection of the axes.

Cartesian coordinates

A set of three numbers that defines the location of a point within a rectilinear coordinate system and consisting of three perpendicular axes (x, y, z).

cathode ray tube

A device, like a television set, for displaying information.

central processing unit

The main computer component that is made up of a control section and an arithmetic-logic section. The other basic units of a computer system are input/output units and primary storage.

channel

The device along which data flow between the input/output units of a computer and primary storage.

character

One of a set of elements that can be arranged in ordered groups to express information. Each character has two forms: 1. a man-intelligible form, the graphic, including the decimal digits 0-9, the letters A-Z, punctuation marks, and other formatting and control symbols; 2. a computer intelligible form, the code, consisting of a group of binary digits (bits).

circular

A MOTYPE option in which the robot tool center point moves in an arc defined by three points. These points can be positions or path nodes.

clear

To replace information in a storage unit by zero (or blank, in some machines).

closed loop

A control system that uses feedback. An open loop control system does not use feedback.

C-MOS RAM

Complementary metal-oxide semiconductor random-access memory. A read/write memory in which the basic memory cell is a pair of MOS (metal-oxide semiconductor) transistors. It is an implementation of S-RAM that has very low power consumption, but might be less dense than other S-RAM implementations.

coaxial cable

A transmission line in which one conductor is centered inside and insulated from an outer metal tube that serves as the second conductor. Also known as coax, coaxial line, coaxial transmission line, concentric cable, concentric line, concentric transmission line.

component

An inclusive term used to identify a raw material, ingredient, part or subassembly that goes into a higher level of assembly, compound or other item.

computer

A device capable of accepting information, applying prescribed processes to the information, and supplying the results of these processes.

configuration

The joint positions of a robot and turn number of wrist that describe the robot at a specified position. Configuration is designated by a STRING value and is included in positional data.

continuous path

A trajectory control system that enables the robot arm to move at a constant tip velocity through a series of predefined locations. A rounding effect of the path is required as the tip tries to pass through these locations.

continuous process control

The use of transducers (sensors) to monitor a process and make automatic changes in operations through the design of appropriate feedback control loops. While such devices historically have been mechanical or electromechanical, microcomputers and centralized control is now used, as well.

continuous production

A production system in which the productive equipment is organized and sequenced according to the steps involved to produce the product. Denotes that material flow is continuous during the production process. The routing of the jobs is fixed and set-ups are seldom changed.

controlled stop

A controlled stop controls robot deceleration until it stops. When a safety stop input such as a safety fence signal is opened, the robot decelerates in a controlled manner and then stops. After the robot stops, the Motor Control Contactor opens and drive power is removed.

controller

A hardware unit that contains the power supply, operator controls, control circuitry, and memory that directs the operation and motion of the robot and communications with external devices. See control unit.

controller memory

A medium in which data are retained. Primary storage refers to the internal area where the data and program instructions are stored for active use, as opposed to auxiliary or external storage (magnetic tape, disk, diskette, and so forth.)

control, open-loop

An operation where the computer applies control directly to the process without manual intervention.

control unit

The portion of a computer that directs the automatic operation of the computer, interprets computer instructions, and initiates the proper signals to the other computer circuits to execute instructions.

coordinate system

See Cartesian coordinate system.

CPU

See central processing unit.

CRT

See cathode ray tube.

cps (viscosity)

Centipoises per second.

CRT/KB

Cathode ray tube/keyboard. An optional interface device for the robot system. The CRT/KB is used for some robot operations and for entering programs. It can be a remote device that attaches to the robot via a cable.

cycle

1. A sequence of operations that is repeated regularly. The time it takes for one such sequence to occur. 2. The interval of time during which a system or process, such as seasonal demand or a manufacturing operation, periodically returns to similar initial conditions. 3. The interval of time during which an event or set of events is completed. In production control, a cycle is the length of time between the release of a manufacturing order and shipment to the customer or inventory.

cycle time

1. In industrial engineering, the time between completion of two discrete units of production. 2. In materials management, the length of time from when material enters a production facility until it exits. See throughput.

cursor

An indicator on a teach pendant or CRT display screen at which command entry or editing occurs. The indicator can be a highlighted field or an arrow (> or ^).

cylindrical

Type of work envelope that has two linear major axes and one rotational major axis. Robotic device that has a predominantly cylindrical work envelope due to its design. Typically has fewer than 6 joints and typically has only 1 linear axis.

D

D/A converter

A digital-to-analog converter. A device that transforms digital data into analog data.

D/A value

A digital-to-analog value. Converts a digital bit pattern into a multilevel analog electrical system.

daisy chain

A means of connecting devices (readers, printers, etc.) to a central processor by party-line input/output buses that join these devices by male and female connectors. The last female connector is shorted by a suitable line termination.

daisy chain configuration

A communications link formed by daisy chain connection of twisted pair wire.

data

A collection of facts, numeric and alphabetical characters, or any representation of information that is suitable for communication and processing.

data base

A data file philosophy designed to establish the independence of computer program from data files. Redundancy is minimized and data elements can be added to, or deleted from, the file designs without changing the existing computer programs.

DC

Abbreviation for direct current.

DEADMAN switch

A control switch on the teach pendant that is used to enable servo power. Pressing the DEADMAN switch while the teach pendant is on activates servo power and releases the robot brakes; releasing the switch deactivates servo power and applies the robot brakes.

debugging

The process of detecting, locating and removing mistakes from a computer program, or manufacturing control system. See diagnostic routine.

deceleration tolerance

The specification of the percentage of deceleration that must be completed before a motion is considered finished and another motion can begin.

default

The value, display, function or program automatically selected if you have not specified a choice.

deviation

Usually, the absolute difference between a number and the mean of a set of numbers, or between a forecast value and the actual data.

device

Any type of control hardware, such as an emergency-stop button, selector switch, control pendant, relay, solenoid valve, or sensor.

diagnostic routine

A test program used to detect and identify hardware/software malfunctions in the controller and its associated I/O equipment. See debugging.

diagnostics

Information that permits the identification and evaluation of robot and peripheral device conditions.

digital

A description of any data that is expressed in numerical format. Also, having the states On and Off only.

digital control

The use of a digital computer to perform processing and control tasks in a manner that is more accurate and less expensive than an analog control system.

digital signal

A single point control signal sent to or from the controller. The signal represents one of two states: ON (TRUE, 1. or OFF (FALSE, 0).

directory

A listing of the files stored on a device.

discrete

Consisting of individual, distinct entities such as bits, characters, circuits, or circuit components. Also refers to ON/OFF type I/O blocks.

disk

A secondary memory device in which information is stored on a magnetically sensitive, rotating disk.

disk memory

A non-programmable, bulk-storage, random-access memory consisting of a magnetized coating on one or both sides of a rotating thin circular plate.

drive power

The energy source or sources for the robot servomotors that produce motion.

DRAM

Dynamic Random Access Memory. A read/write memory in which the basic memory cell is a capacitor. DRAM (or D-RAM) tends to have a higher density than SRAM (or S-RAM). Due to the support circuitry required, and power consumption needs, it is generally impractical to use. A battery can be used to retain the content upon loss of power.

E

edit

1. A software mode that allows creation or alteration of a program. 2. To modify the form or format of data, for example, to insert or delete characters.

emergency stop

The operation of a circuit using hardware-based components that overrides all other robot controls, removes drive power from the actuators, and causes all moving parts of to stop. The operator panel and teach pendant are each equipped with EMERGENCY STOP buttons.

enabling device

A manually operated device that, when continuously activated, permits motion. Releasing the device stops the motion of the robot and associated equipment that might present a hazard.

encoder

1. A device within the robot that sends the controller information about where the robot is. 2. A transducer used to convert position data into electrical signals. The robot system uses an incremental optical encoder to provide position feedback for each joint. Velocity data is computed from the encoder signals and used as an additional feedback signal to assure servo stability.

end effector

An accessory device or tool specifically designed for attachment to the robot wrist or tool mounting plate to enable the robot to perform its intended tasks. Examples include gripper, spot weld gun, arc weld gun, spray paint gun, etc.

end-of-arm tooling

Any of a number of tools, such as welding guns, torches, bells, paint spraying devices, attached to the faceplate of the robot wrist. Also called end effector or EOAT.

engineering units

Units of measure as applied to a process variable, for example, psi, Degrees F., etc.

envelope, maximum

The volume of space encompassing the maximum designed movements of all robot parts including the end effector, workpiece, and attachments.

EOAT

See end of arm tooling, tool.

EPROM

Erasable Programmable Read Only Memory. Semiconductor memory that can be erased and reprogrammed. A non-volatile storage memory.

error

The difference in value between actual response and desired response in the performance of a controlled machine, system or process. Alarm=Error.

error message

A numbered message, displayed on the CRT/KB and teach pendant, that indicates a system problem or warns of a potential problem.

Ethernet

A Local Area Network (LAN) bus-oriented, hardware technology that is used to connect computers, printers, terminal concentrators (servers), and many other devices together. It consists of a master cable and connection devices at each machine on the cable that allow the various devices to "talk" to each other. Software that can access the Ethernet and cooperate with machines connected to the cable is necessary. Ethernets come in varieties such as baseband and broadband and can run on different media, such as coax, twisted pair and fiber. Ethernet is a trademark of Xerox Corporation.

execute

To perform a specific operation, such as one that would be accomplished through processing one statement or command, a series of statements or commands, or a complete program or command procedure.

extended axis

An optional, servo-controlled axis that provides extended reach capability for a robot, including in-booth rail, single- or double-link arm, also used to control motion of positioning devices.

F

faceplate

The tool mounting plate of the robot.

feedback

1. The signal or data fed back to a commanding unit from a controlled machine or process to denote its response to the command signal. The signal representing the difference between actual response and desired response that is used by the commanding unit to improve performance of the controlled machine or process. 2. The flow of information back into the control system so that actual performance can be compared with planned performance, for instance in a servo system.

field

A specified area of a record used for a particular category of data. 2. A group of related items that occupy the same space on a CRT/KB screen or teach pendant LCD screen. Field name is the name of the field; field items are the members of the group.

field devices

User-supplied devices that provide information to the PLC (inputs: push buttons, limit switches, relay contacts, an so forth) or perform PLC tasks (outputs: motor starters, solenoids, indicator lights, and so forth.)

file

1. An organized collection of records that can be stored or retrieved by name. 2. The storage device on which these records are kept, such as bubble memory or disk.

filter

A device to suppress interference that would appear as noise.

Flash File Storage

A portion of FROM memory that functions as a separate storage device. Any file can be stored on the FROM disk.

Flash ROM

Flash Read Only Memory. Flash ROM is not battery-backed memory but it is non-volatile. All data in Flash ROM is saved even after you turn off and turn on the robot.

flow chart

A systems analysis tool to graphically show a procedure in which symbols are used to represent operations, data, flow, and equipment. See block diagram, process chart.

flow control

A specific production control system that is based primarily on setting production rates and feeding work into production to meet the planned rates, then following it through production to make sure that it is moving. This concept is most successful in repetitive production.

format

To set up or prepare a memory card or floppy disk (not supported with version 7.20 and later) so it can be used to store data in a specific system.

FR

See Flash ROM.

F-ROM

See Flash ROM.

FROM disk

See Flash ROM.

G

general override stat

A percentage value that governs the maximum robot jog speed and program run speed.

Genius I/O bus

The serial bus that provides communications between blocks, controllers, and other devices in the system especially with respect to GE FANUC Genius I/O.

gripper

The "hand" of a robot that picks up, holds and releases the part or object being handled. Sometimes referred to as a manipulator. See EOAT, tool.

group signal

An input/output signal that has a variable number of digital signals, recognized and taken as a group.

gun

See applicator.

H

Hand Model.

Used in Interference Checking, the Hand Model is the set of virtual model elements (spheres and cylinders) that are used to represent the location and shape of the end of arm tooling with respect to the robot’s faceplate.

hardware

1. In data processing, the mechanical, magnetic, electrical and electronic devices of which a computer, controller, robot, or panel is built. 2. In manufacturing, relatively standard items such as nuts, bolts, washers, clips, and so forth.

hard-wire

To connect electric components with solid metallic wires.

hard-wired

1. Having a fixed wired program or control system built in by the manufacturer and not subject to change by programming. 2. Interconnection of electrical and electronic devices directly through physical wiring.

hazardous motion

Unintended or unexpected robot motion that can cause injury.

hexadecimal

A numbering system having 16 as the base and represented by the digits 0 through 9, and A through F.

hold

A smoothly decelerated stopping of all robot movement and a pause of program execution. Power is maintained on the robot and program execution generally can be resumed from a hold.

HTML.

Hypertext Markup Language. A markup language that is used to create hypertext and hypermedia documents incorporating text, graphics, sound, video, and hyperlinks.

http.

Hypertext transfer protocol. The protocol used to transfer HTML files between web servers.

I

impedance

A measure of the total opposition to current flow in an electrical circuit.

incremental encoder system

A positional information system for servomotors that requires calibrating the robot by moving it to a known reference position (indicated by limit switches) each time the robot is turned on or calibration is lost due to an error condition.

index

An integer used to specify the location of information within a table or program.

index register

A memory device containing an index.

industrial robot

A reprogrammable multifunctional manipulator designed to move material, parts, tools, or specialized devices through variable programmed motions in order to perform a variety of tasks.

industrial robot system

A system that includes industrial robots, end effectors, any equipment devices and sensors required for the robot to perform its tasks, as well as communication interfaces for interlocking, sequencing, or monitoring the robot.

information

The meaning derived from data that have been arranged and displayed in a way that they relate to that which is already known. See data.

initialize

1. Setting all variable areas of a computer program or routine to their desired initial status, generally done the first time the code is executed during each run. 2. A program or hardware circuit that returns a program a system, or hardware device to an original state. See startup, initial.

input

The data supplied from an external device to a computer for processing. The device used to accomplish this transfer of data.

input device

A device such as a terminal keyboard that, through mechanical or electrical action, converts data from the form in which it has been received into electronic signals that can be interpreted by the CPU or programmable controller. Examples are limit switches, push buttons, pressure switches, digital encoders, and analog devices.

input processing time

The time required for input data to reach the microprocessor.

input/output

Information or signals transferred between devices, discreet electrical signals for external control.

input/output control

A technique for controlling capacity where the actual output from a work center is compared with the planned output developed by CRP. The input is also monitored to see if it corresponds with plans so that work centers will not be expected to generate output when jobs are not available to work on.

integrated circuit

A solid-state micro-circuit contained entirely within a chip of semiconductor material, generally silicon. Also called chip.

interactive

Refers to applications where you communicate with a computer program via a terminal by entering data and receiving responses from the computer.

interface

1. A concept that involves the specifications of the inter-connection between two equipments having different functions. 2. To connects a PLC with the application device, communications channel, and peripherals through various modules and cables. 3. The method or equipment used to communicate between devices.

interference zone

An area that falls within the work envelope of a robot, in which there is the potential for the robot motion to coincide with the motion of another robot or machine, and for a collision to occur.

interlock

An arrangement whereby the operation of one control or mechanism brings about, or prevents, the operations of another.

interrupt

A break in the normal flow of a system or program that occurs in a way that the flow can be resumed from that point at a later time. Interrupts are initiated by two types of signals: 1. signals originating within the computer system to synchronize the operation of the computer system with the outside world; 2. signals originating exterior to the computer system to synchronize the operation of the computer system with the outside world.

I/O

Abbreviation for input/output or input/output control.

I/O block

A microprocessor-based, configurable, rugged solid state device to which field I/O devices are attached.

I/O electrical isolation

A method of separating field wiring from logic level circuitry. This is typically done through optical isolation devices.

I/O module

A printed circuit assembly that is the interface between user devices and the Series Six PLC.

I/O scan

A method by which the CPU monitors all inputs and controls all outputs within a prescribed time. A period during which each device on the bus is given a turn to send information and listen to all of the broadcast data on the bus.

ISO

The International Standards Organization that establishes the ISO interface standards.

isolation

1. The ability of a logic circuit having more than one inputs to ensure that each input signal is not affected by any of the others. 2. A method of separating field wiring circuitry from logic level circuitry, typically done optically.

item

1. A category displayed on the teach pendant on a menu. 2. A set of adjacent digits, bits, or characters that is treated as a unit and conveys a single unit of information. 3. Any unique manufactured or purchased part or assembly: end product, assembly, subassembly, component, or raw material.

J

jog coordinate systems

Coordinate systems that help you to move the robot more effectively for a specific application. These systems include JOINT, WORLD, TOOL, and USER.

JOG FRAME

A jog coordinate system you define to make the robot jog the best way possible for a specific application. This can be different from world coordinate frame.

jogging

Pressing special keys on the teach pendant to move the robot.

jog speed

Is a percentage of the maximum speed at which you can jog the robot.

joint

1. A single axis of rotation. There are up to six joints in a robot arm (P-155 swing arm has 8). 2. A jog coordinate system in which one axis is moved at a time.

JOINT

A motion type in which the robot moves the appropriate combination of axes independently to reach a point most efficiently. (Point to point, non-linear motion).

joint interpolated motion

A method of coordinating the movement of the joints so all joints arrive at the desired location at the same time. This method of servo control produces a predictable path regardless of speed and results in the fastest cycle time for a particular move. Also called joint motion.

K

K

Abbreviation for kilo, or exactly 1024 in computer jargon. Related to 1024 words of memory.

KAREL

The programming language developed for robots by the FANUC America Corporation.

L

label

An ordered set of characters used to symbolically identify an instruction, a program, a quantity, or a data area.

LCD

See liquid crystal display.

lead time

The span of time needed to perform an activity. In the production and inventory control context, this activity is normally the procurement of materials and/or products either from an outside supplier or from one's own manufacturing facility. Components of lead time can include order preparation time, queue time, move or transportation time, receiving and inspection time.

LED

See Light Emitting Diode.

LED display

An alphanumeric display that consists of an array of LEDs.

Light Emitting Diode

A solid-state device that lights to indicate a signal on electronic equipment.

limiting device

A device that restricts the work envelope by stopping or causing to stop all robot motion and that is independent of the control program and the application programs.

limit switch

A switch that is actuated by some part or motion of a machine or equipment to alter the electrical circuit associated with it. It can be used for position detection.

linear

A motion type in which the appropriate combination of axes move in order to move the robot TCP in a straight line while maintaining tool center point orientation.

liquid crystal display

A digital display on the teach pendant that consists of two sheets of glass separated by a sealed-in, normally transparent, liquid crystal material. Abbreviated LCD.

little-endian

The adjectives big-endian and little-endian refer to which bytes are most significant in multi-byte data types and describe the order in which a sequence of bytes is stored in a computer's memory. In a big-endian system, the most significant value in the sequence is stored at the lowest storage address (i.e., first). In a little-endian system, the least significant value in the sequence is stored first.

load

1. The weight (force) applied to the end of the robot arm. 2. A device intentionally placed in a circuit or connected to a machine or apparatus to absorb power and convert it into the desired useful form. 3. To copy programs or data into memory storage.

location

1. A storage position in memory uniquely specified by an address. 2. The coordinates of an object used in describing its x, y, and z position in a Cartesian coordinate system.

lockout/tagout

The placement of a lock and/or tag on the energy isolating device (power disconnecting device) in the off or open position. This indicates that the energy isolating device or the equipment being controlled will not be operated until the lock/tag is removed.

log

A record of values and/or action for a given function.

logic

A fixed set of responses (outputs) to various external conditions (inputs). Also referred to as the program.

loop

The repeated execution of a series of instructions for a fixed number of times, or until interrupted by the operator.

M

mA

See milliampere.

machine language

A language written in a series of bits that are understandable by, and therefore instruct, a computer. This is a "first level" computer language, as compared to a "second level" assembly language, or a "third level" compiler language.

machine lock

A test run option that allows the operator to run a program without having the robot move.

macro

A source language instruction from which many machine-language instructions can be generated.

magnetic disk

A metal or plastic floppy disk (not supported on version 7.10 and later) that looks like a phonograph record whose surface can store data in the form of magnetized spots.

magnetic disk storage

A storage device or system consisting of magnetically coated metal disks.

magnetic tape

Plastic tape, like that used in tape recorder, on which data is stored in the form of magnetized spots.

maintenance

Keeping the robots and system in their proper operating condition.

MC

See memory card.

mechanical unit

The robot arm, including auxiliary axis, and hood/deck and door openers.

medium

plural **media** . The physical substance upon which data is recorded, such as a memory card (or floppy disk which is not supported on version 7.10 and later).

memory

A device or media used to store information in a form that can be retrieved and is understood by the computer or controller hardware. Memory on the controller includes C-MOS RAM, Flash ROM and D-RAM.

memory card

A C-MOS RAM memory card or a flash disk-based PC card.

menu
A list of options displayed on the teach pendant screen.

message
A group of words, variable in length, transporting an item of information.

microprocessor
A single integrated circuit that contains the arithmetic, logic, register, control and memory elements of a computer.

microsecond
One millionth (0.000001) of a second

milliampere
One one-thousandth of an ampere. Abbreviated mA.

millisecond
One thousandth of a second. Abbreviated msec.

module
A distinct and identifiable unit of computer program for such purposes as compiling, loading, and linkage editing. It is eventually combined with other units to form a complete program.

motion type
A feature that allows you to select how you want the robot to move from one point to the next. MOTYPES include joint, linear, and circular.

mode
1. One of several alternative conditions or methods of operation of a device. 2. The most common or frequent value in a group of values.

N

network
1. The interconnection of a number of devices by data communication facilities. "Local networking" is the communications network internal to a robot."Global networking" is the ability to provide communications connections outside of the robot’s internal system. 2. Connection of geographically separated computers and/or terminals over communications lines. The control of transmission is managed by a standard protocol.

non-volatile memory
Memory capable of retaining its stored information when power is turned off.

O

Obstacle Model.

Used in Interference Checking, the Obstacle Model is the set of virtual model elements (spheres, cylinders, and planes) that are used to represent the shape and the location of a given obstacle in space.

off-line

Equipment or devices that are not directly connected to a communications line.

off-line operations

Data processing operations that are handled outside of the regular computer program. For example, the computer might generate a report off-line while the computer was doing another job.

off-line programming

The development of programs on a computer system that is independent of the "on-board" control of the robot. The resulting programs can be copied into the robot controller memory.

offset

The count value output from a A/D converter resulting from a zero input analog voltage. Used to correct subsequent non-zero measurements also incremental position or frame adjustment value.

on-line

A term to describe equipment or devices that are connected to the communications line.

on-line processing

A data processing approach where transactions are entered into the computer directly, as they occur.

operating system

Lowest level system monitor program.

operating work envelope

The portion of the restricted work envelope that is actually used by the robot while it is performing its programmed motion. This includes the maximum the end-effector, the workpiece, and the robot itself.

operator

A person designated to start, monitor, and stop the intended productive operation of a robot or robot system.

operator box

A control panel that is separate from the robot and is designed as part of the robot system. It consists of the buttons, switches, and indicator lights needed to operate the system.

operator panel

A control panel designed as part of the robot system and consisting of the buttons, switches, and indicator lights needed to operate the system.

optional features

Additional capabilities available at a cost above the base price.

OR

An operation that places two contacts or groups of contacts in parallel. Any of the contacts can control the resultant status, also a mathematical operation.

orientation

The attitude of an object in space. Commonly described by three angles: rotation about x (w), rotation about y (p), and rotation about z (r).

origin

The point in a Cartesian coordinate system where axes intersect; the reference point that defines the location of a frame.

OT

See overtravel.

output

Information that is transferred from the CPU for control of external devices or processes.

output device

A device, such as starter motors, solenoids, that receive data from the programmable controller.

output module

An I/O module that converts logic levels within the CPU to a usable output signal for controlling a machine or process .

outputs

Signals, typically on or off, that controls external devices based upon commands from the CPU.

override

See general override.

overtravel

A condition that occurs when the motion of a robot axis exceeds its prescribed limits.

overwrite

To replace the contents of one file with the contents of another file when copying.

P

parity

The anticipated state, odd or even, of a set of binary digits.

parity bit

A binary digit added to an array of bits to make the sum of all bits always odd or always even.

parity check

A check that tests whether the number of ones (or zeros) in an array of binary digits is odd or even.

parity error
A condition that occurs when a computed parity check does not agree with the parity bit.

part
A material item that is used as a component and is not an assembly or subassembly.

pascal
A unit of pressure in the meter-kilogram-second system equivalent to one newton per square meter.

path
1. A variable type available in the KAREL system that consists of a list of positions. Each node includes positional information and associated data. 2. The trajectory followed by the TCP in a move.

PCB
See printed circuit board.

PC Interface
The PC Interface option provides the RPC functions and PC send macros required by applications created using PC Developer's Kit.

pendant
See teach pendant.

PLC
See programmable logic controller or cell controller.

PMC
The programmable machine controller (PMC) functions provide a ladder logic programming environment to create PMC functions. This provides the capability to use the robot I/O system to run PLC programs in the background of normal robot operations. This function can be used to control bulk supply systems, fixed automation that is part of the robot workcell, or other devices that would normally require basic PLC controls.

printed circuit board
A flat board whose front contains slots for integrated circuit chips and connections for a variety of electronic components, and whose back is printed with electrically conductive pathways between the components.

production mode
See automatic mode.

program
1. A plan for the solution of a problem. A complete program includes plans for the transcription of data, coding for the computer, and plans for the absorption of the results into the system. 2. A sequence of instructions to be executed by the computer or controller to control a robot/robot system. 3. To furnish a computer with a code of instructions. 4. To teach a robot system a specific set of movements and instructions to do a task.

programmable controller

See programmable logic controller or cell controller.

programmable logic controller

A solid-state industrial control device that receives inputs from user-supplied control devices, such as switches and sensors, implements them in a precise pattern determined by ladder diagram-based programs stored in the user memory, and provides outputs for control of processes or user-supplied devices such as relays and motor starters.

Program ToolBox

The Program ToolBox software provides programming utilities such as mirror image and flip wrist editing capabilities.

protocol

A set of hardware and software interfaces in a terminal or computer that allows it to transmit over a communications network, and that collectively forms a communications language.

psi

Pounds per square inch.

Q

queue.

1. Waiting lines resulting from temporary delays in providing service. 2. The amount of time a job waits at a work center before set-up or work is performed on the job. See also job queue.

R

RAM

See Random Access Memory.

random access

A term that describes files that do not have to be searched sequentially to find a particular record but can be addressed directly.

Random Access Memory

1. Volatile, solid-state memory used for storage of programs and locations; battery backup is required. 2. The working memory of the controller. Programs and variable data must be loaded into RAM before the program can execute or the data can be accessed by the program.

range

1. A characterization of a variable or function. All the values that a function can possess. 2. In statistics, the spread in a series of observations. 3. A programmable voltage or current spectrum of values to which input or output analog signals can be limited.

RI

Robot input.

RO

Robot output.

read

To copy, usually from one form of storage to another, particularly from external or secondary storage to internal storage. To sense the meaning of arrangements of hardware. To sense the presence of information on a recording medium.

Read Only Memory

A digital memory containing a fixed pattern of bits that you cannot alter.

record

To store the current set or sets of information on a storage device.

recovery

The restoration of normal processing after a hardware or software malfunction through detailed procedures for file backup, file restoration, and transaction logging.

register

1. A special section of primary storage in a computer where data is held while it is being worked on.
2. A memory device capable of containing one or more computer bits or words.

remote/local

A device connection to a given computer, with remote devices being attached over communications lines and local devices attached directly to a computer channel; in a network, the computer can be a remote device to the CPU controlling the network.

repair

To restore robots and robot systems to operating condition after damage, malfunction, or wear.

repeatability

The closeness of agreement among the number of consecutive movements made by the robot arm to a specific point.

reset

To return a register or storage location to zero or to a specified initial condition.

restricted work envelope

That portion of the work envelope to which a robot is restricted by limiting devices that establish limits that will not be exceeded in the event of any reasonably foreseeable failure of the robot or its controls. The maximum distance the robot can travel after the limited device is actuated defines the restricted work envelope of the robot.

RIA

Robotic Industries Association Subcommittee of the American National Standards Institute, Inc.

robot

A reprogrammable multifunctional manipulator designed to move material, parts, tools, or specialized devices, through variable programmed motions for the performance of a variety of tasks.

Robot Model.

Used in Interference Checking, the Robot Model is the set of virtual model elements (sphere and cylinders) that are used to represent the location and shape of the robot arm with respect to the robot's base. Generally, the structure of a six axes robot can be accurately modeled as a series of cylinders and spheres. Each model element represents a link or part of the robot arm.

ROM

See Read Only Memory.

routine

1. A list of coded instructions in a program. 2. A series of computer instructions that performs a specific task and can be executed as often as needed during program execution.

S

saving data.

Storing program data in Flash ROM, to a floppy disk (not supported on version 7.10 and later), or memory card.

scfm

Standard cubic feet per minute.

scratch start

Allows you to enable and disable the automatic recovery function.

sensor

A device that responds to physical stimuli, such as heat, light, sound pressure, magnetism, or motion, and transmits the resulting signal or data for providing a measurement, operating a control or both. Also a device that is used to measure or adjust differences in voltage in order to control sophisticated machinery dynamically.

serial communication

A method of data transfer within a PLC whereby the bits are handled sequentially rather than simultaneously as in parallel transmission.

serial interface

A method of data transmission that permits transmitting a single bit at a time through a single line. Used where high speed input is not necessary.

Server Side Include (SSI)

A method of calling or "including" code into a web page.

servomotor
An electric motor that is controlled to produce precision motion. Also called a "smart" motor.

SI
System input.

signal
The event, phenomenon, or electrical quantity that conveys information from one point to another.

significant bit
A bit that contributes to the precision of a number. These are counted starting with the bit that contributes the most value, of "most significant bit", and ending with the bit that contributes the least value, or "least significant bit".

singulating
Separating parts into a single layer.

slip sheet
A sheet of material placed between certain layers of a unit load. Also known as tier sheet.

SO
System output.

specific gravity
The ratio of a mass of solid or liquid to the mass of an equal volume of water at 45C. You must know the specific gravity of the dispensing material to perform volume signal calibration. The specific gravity of a dispensing material is listed on the MSDS for that material.

SRAM
A read/write memory in which the basic memory cell is a transistor. SRAM (or S-RAM) tends to have a lower density than DRAM. A battery can be used to retain the content upon loss of power.

slpm
Standard liters per minute.

Standard Operator Panel (SOP).
A panel that is made up of buttons, keyswitches, and connector ports.

state
The on or off condition of current to and from and input or output device.

statement
See instruction.

storage device
Any device that can accept, retain, and read back one or more times. The available storage devices are SRAM, Flash ROM (FROM or F-ROM), floppy disks (not available on version 7.10 and later), memory cards, or a USB memory stick.

system variable

An element that stores data used by the controller to indicate such things as robot specifications, application requirements, and the current status of the system.

T

Tare

The difference between the gross weight of an object and its contents, and the object itself. The weight of an object without its contents.

TCP

See tool center point.

teaching

Generating and storing a series of positional data points effected by moving the robot arm through a path of intended motions.

teach mode

1. The mode of operation in which a robot is instructed in its motions, usually by guiding it through these motions using a teach pendant. 2. The generation and storage of positional data. Positional data can be taught using the teach pendant to move the robot through a series of positions and recording those positions for use by an application program.

teach pendant

1. A hand-held device used to instruct a robot, specifying the character and types of motions it is to undertake. Also known as teach box, teach gun. 2. A portable device, consisting of an LCD display and a keypad, that serves as a user interface to the KAREL system and attaches to the operator box or operator panel via a cable. The teach pendant is used for robot operations such as jogging the robot, teaching and recording positions, and testing and debugging programs.

telemetry

The method of transmission of measurements made by an instrument or a sensor to a remote location.

termination type

Feature that controls the blending of robot motion between segments.

tool

A term used loosely to define something mounted on the end of the robot arm, for example, a hand, gripper, or an arc welding torch.

tool center point

1. The location on the end-effector or tool of a robot hand whose position and orientation define the coordinates of the controlled object. 2. Reference point for position control, that is, the point on the tool that is used to teach positions. Abbreviated TCP.

TOOL Frame

The Cartesian coordinate system that has the position of the TCP as its origin to stet. The z-axis of the tool frame indicates the approach vector for the tool.

TP.

See teach pendant.

transducer

A device for converting energy from one form to another.

U

UOP

See user operator panel.

URL

Universal Resource Locator. A standard addressing scheme used to locate or reference files on web servers.

USB memory stick

The controller USB memory stick interface supports a USB 1.1 interface. The USB Organization specifies standards for USB 1.1 and 2.0. Most memory stick devices conform to the USB 2.0 specification for operation and electrical standards. USB 2.0 devices as defined by the USB Specification must be backward compatible with USB 1.1 devices. However, FANUC America Corporation does not support any security or encryption features on USB memory sticks. The controller supports most widely-available USB Flash memory sticks from 32MB up to 1GB in size.

USER Frame

The Cartesian coordinate system that you can define for a specific application. The default value of the User Frame is the World Frame. All positional data is recorded relative to User Frame.

User Operator Panel

User-supplied control device used in place of or in parallel with the operator panel or operator box supplied with the controller. Abbreviated UOP .

V

variable

A quantity that can assume any of a given set of values.

variance

The difference between the expected (or planned) and the actual, also statistics definitions.

vision system

A device that collects data and forms an image that can be interpreted by a robot computer to determine the position or to “see” an object.

volatile memory
Memory that will lose the information stored in it if power is removed from the memory circuit device.

W

web server
An application that allows you to access files on the robot using a standard web browser.

warning device
An audible or visible device used to alert personnel to potential safety hazards.

work envelope
The volume of space that encloses the maximum designed reach of the robot manipulator including the end effector, the workpiece, and the robot itself. The work envelope can be reduced or restricted by limiting devices. The maximum distance the robot can travel after the limit device is actuated is considered the basis for defining the restricted work envelope.

write
To deliver data to a medium such as storage.

Index