FANUC Robot series

R-30*i*B CONTROLLER

Line Tracking

OPERATOR'S MANUAL

B-83474EN/01

Before using the Robot, be sure to read the "FANUC Robot Safety Manual (B-80687EN)" and understand the content.

The products in this manual are controlled based on Japan's "Foreign Exchange and Foreign Trade Law". The export from Japan may be subject to an export license by the government of Japan.
Further, re-export to another country may be subject to the license of the government of the country from where the product is re-exported. Furthermore, the product may also be controlled by re-export regulations of the United States government.
Should you wish to export or re-export these products, please contact FANUC for advice.

In this manual we have tried as much as possible to describe all the various matters.
However, we cannot describe all the matters which must not be done, or which cannot be done, because there are so many possibilities.
Therefore, matters which are not especially described as possible in this manual should be regarded as "impossible".

SAFETY PRECAUTIONS

Thank you for purchasing FANUC Robot.
This chapter describes the precautions which must be observed to ensure the safe use of the robot.
Before attempting to use the robot, be sure to read this chapter thoroughly.

Before using the functions related to robot operation, read the relevant operator's manual to become familiar with those functions.

If any description in this chapter differs from that in the other part of this manual, the description given in this chapter shall take precedence.

For the safety of the operator and the system, follow all safety precautions when operating a robot and its peripheral devices installed in a work cell.
In addition, refer to the “FANUC Robot SAFETY HANDBOOK (B-80687EN)”.

1 WORKING PERSON

The personnel can be classified as follows.

Operator:
- Turns robot controller power ON/OFF
- Starts robot program from operator’s panel

Programmer or teaching operator:
- Operates the robot
- Teaches robot inside the safety fence

Maintenance engineer:
- Operates the robot
- Teaches robot inside the safety fence
- Maintenance (adjustment, replacement)

- An operator cannot work inside the safety fence.
- A programmer, teaching operator, and maintenance engineer can work inside the safety fence. The working activities inside the safety fence include lifting, setting, teaching, adjusting, maintenance, etc.
- To work inside the fence, the person must be trained on proper robot operation.

During the operation, programming, and maintenance of your robotic system, the programmer, teaching operator, and maintenance engineer should take additional care of their safety by using the following safety precautions.

- Use adequate clothing or uniforms during system operation
- Wear safety shoes
- Use helmet

2 DEFINITION OF WARNING, CAUTION AND NOTE

To ensure the safety of user and prevent damage to the machine, this manual indicates each precaution on safety with "Warning" or "Caution" according to its severity. Supplementary information is indicated by "Note". Read the contents of each "Warning", "Caution" and "Note" before attempting to use the oscillator.

> ⚠**WARNING**
> Applied when there is a danger of the user being injured or when there is a danger of both the user being injured and the equipment being damaged if the approved procedure is not observed.

> ⚠**CAUTION**
> Applied when there is a danger of the equipment being damaged, if the approved procedure is not observed.

> **NOTE**
> Notes are used to indicate supplementary information other than Warnings and Cautions.

- Read this manual carefully, and store it in a sales place.

3 WORKING PERSON SAFETY

Working person safety is the primary safety consideration. Because it is very dangerous to enter the operating space of the robot during automatic operation, adequate safety precautions must be observed. The following lists the general safety precautions. Careful consideration must be made to ensure working person safety.

(1) Have the robot system working persons attend the training courses held by FANUC.

> FANUC provides various training courses. Contact our sales office for details.

(2) Even when the robot is stationary, it is possible that the robot is still in a ready to move state, and is waiting for a signal. In this state, the robot is regarded as still in motion. To ensure working person safety, provide the system with an alarm to indicate visually or aurally that the robot is in motion.
(3) Install a safety fence with a gate so that no working person can enter the work area without passing through the gate. Install an interlocking device, a safety plug, and so forth in the safety gate so that the robot is stopped as the safety gate is opened.

> The controller is designed to receive this interlocking signal of the door switch. When the gate is opened and this signal received, the controller stops the robot (Please refer to "STOP TYPE OF ROBOT" in SAFETY PRECAUTIONS for detail of stop type). For connection, see Fig.3 (a) and Fig.3 (b).

(4) Provide the peripheral devices with appropriate grounding (Class A, Class B, Class C, and Class D).
(5) Try to install the peripheral devices outside the work area.
(6) Draw an outline on the floor, clearly indicating the range of the robot motion, including the tools such as a hand.
(7) Install a mat switch or photoelectric switch on the floor with an interlock to a visual or aural alarm that stops the robot when a working person enters the work area.
(8) If necessary, install a safety lock so that no one except the working person in charge can turn on the power of the robot.

The circuit breaker installed in the controller is designed to disable anyone from turning it on when it is locked with a padlock.

(9) When adjusting each peripheral device independently, be sure to turn off the power of the robot
(10) Operators should be ungloved while manipulating the operator’s panel or teach pendant. Operation with gloved fingers could cause an operation error.
(11) Programs, system variables, and other information can be saved on memory card or USB memories. Be sure to save the data periodically in case the data is lost in an accident.
(12) The robot should be transported and installed by accurately following the procedures recommended by FANUC. Wrong transportation or installation may cause the robot to fall, resulting in severe injury to workers.
(13) In the first operation of the robot after installation, the operation should be restricted to low speeds. Then, the speed should be gradually increased to check the operation of the robot.
(14) Before the robot is started, it should be checked that no one is in the area of the safety fence. At the same time, a check must be made to ensure that there is no risk of hazardous situations. If detected, such a situation should be eliminated before the operation.
(15) When the robot is used, the following precautions should be taken. Otherwise, the robot and peripheral equipment can be adversely affected, or workers can be severely injured.
 - Avoid using the robot in a flammable environment.
 - Avoid using the robot in an explosive environment.
 - Avoid using the robot in an environment full of radiation.
 - Avoid using the robot under water or at high humidity.
 - Avoid using the robot to carry a person or animal.
 - Avoid using the robot as a stepladder. (Never climb up on or hang from the robot.)
(16) When connecting the peripheral devices related to stop(safety fence etc.) and each signal (external emergency , fence etc.) of robot. be sure to confirm the stop movement and do not take the wrong connection.
(17) When preparing trestle, please consider security for installation and maintenance work in high place according to Fig.3 (c). Please consider footstep and safety bolt mounting position.

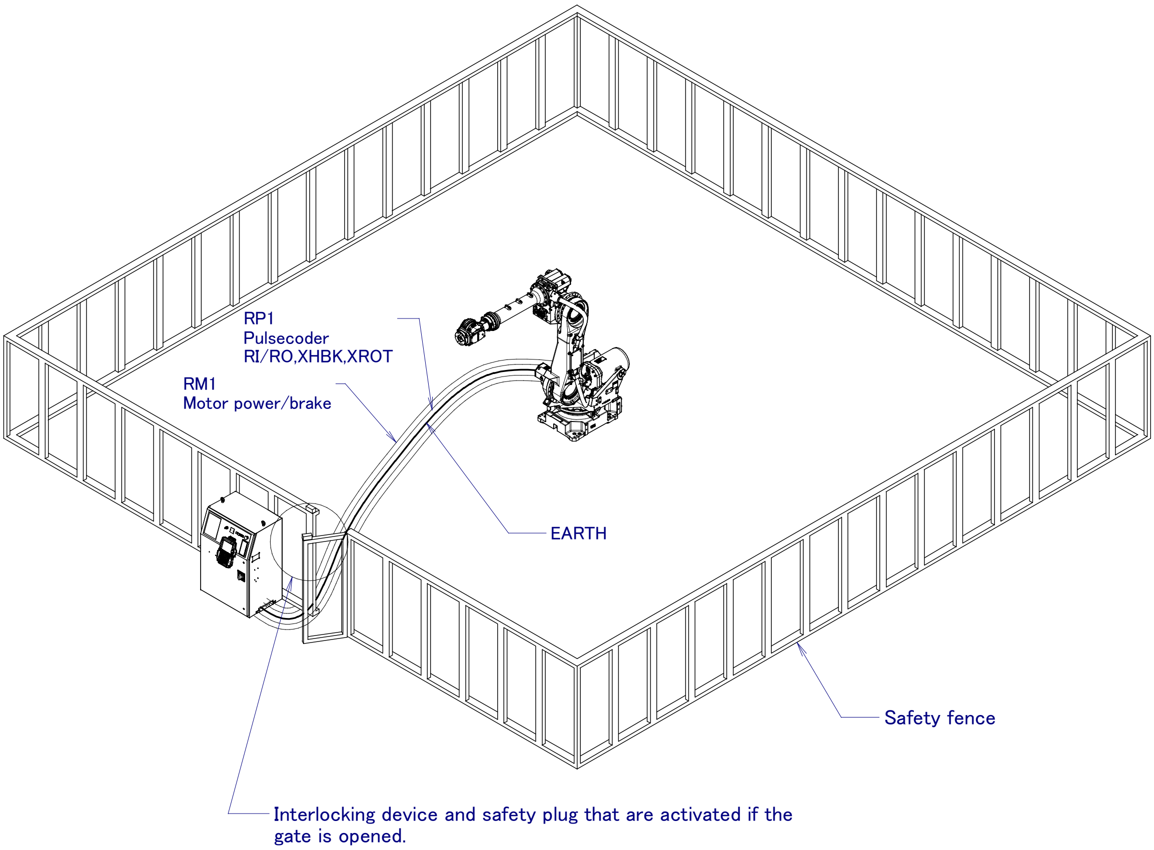


Fig. 3 (a) Safety fence and safety gate

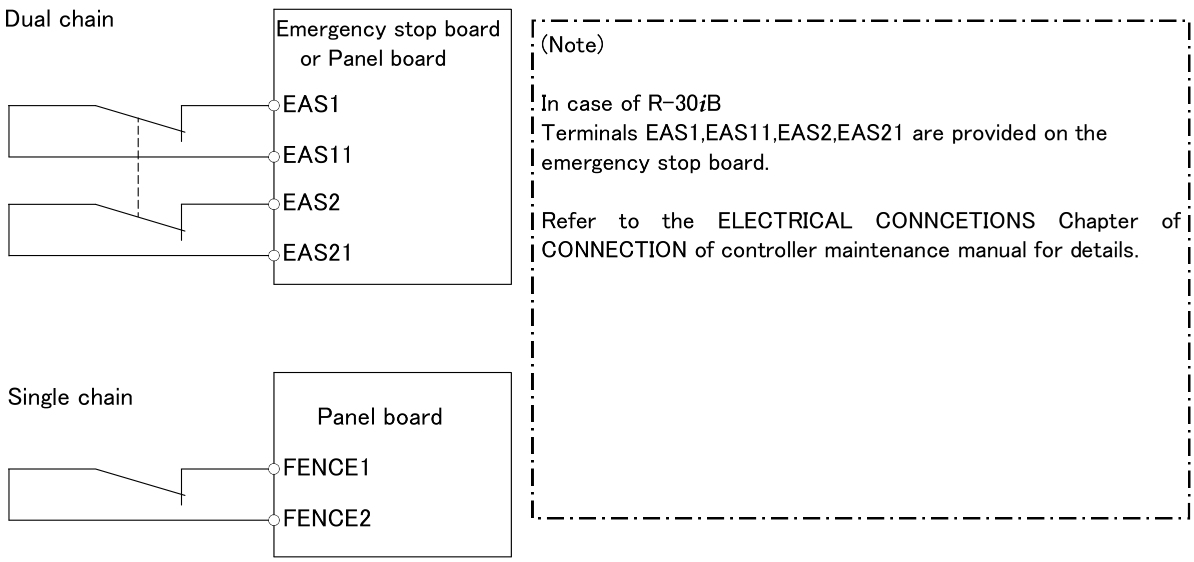


Fig. 3 (b) Limit switch circuit diagram of the safety fence

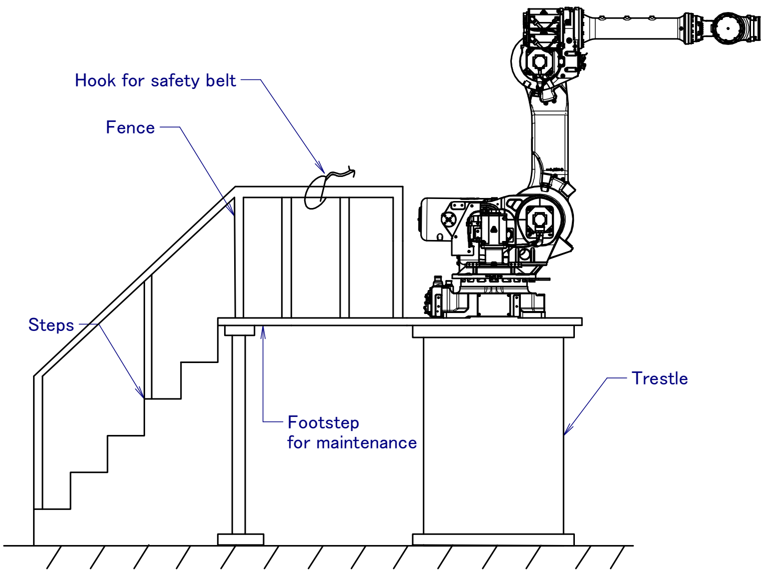


Fig.3 (c) Footstep for maintenance

3.1 OPERATOR SAFETY

The operator is a person who operates the robot system. In this sense, a worker who operates the teach pendant is also an operator. However, this section does not apply to teach pendant operators.

(1) If you do not have to operate the robot, turn off the power of the robot controller or press the EMERGENCY STOP button, and then proceed with necessary work.
(2) Operate the robot system at a location outside of the safety fence
(3) Install a safety fence with a safety gate to prevent any worker other than the operator from entering the work area unexpectedly and to prevent the worker from entering a dangerous area.
(4) Install an EMERGENCY STOP button within the operator's reach.

The robot controller is designed to be connected to an external EMERGENCY STOP button. With this connection, the controller stops the robot operation (Please refer to "STOP TYPE OF ROBOT" in SAFETY PRECAUTIONS for detail of stop type), when the external EMERGENCY STOP button is pressed. See the diagram below for connection.

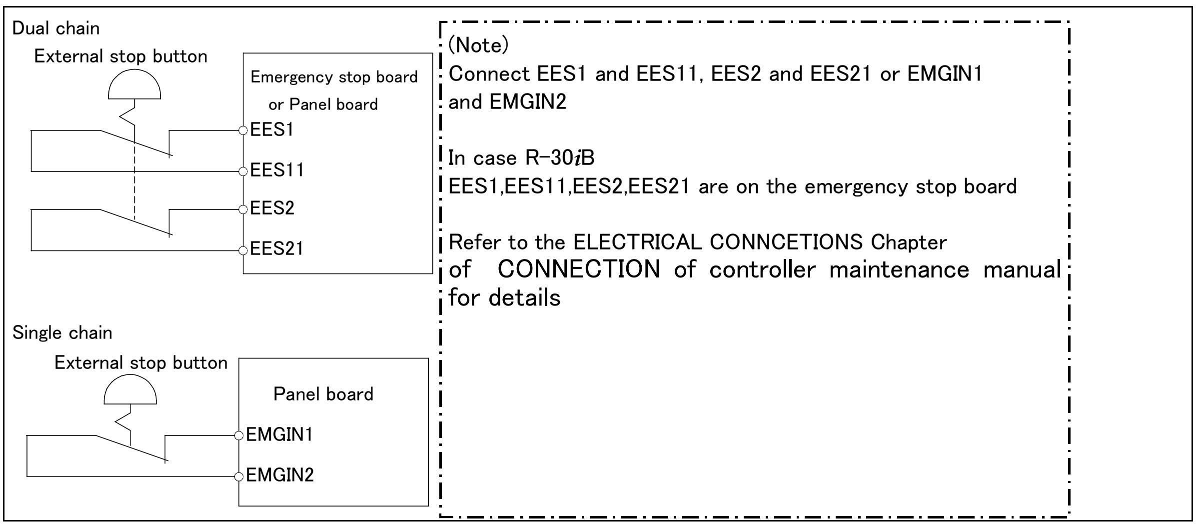


Fig.3.1 Connection diagram for external emergency stop button

3.2 SAFETY OF THE PROGRAMMER

While teaching the robot, the operator must enter the work area of the robot. The operator must ensure the safety of the teach pendant operator especially.

(1) Unless it is specifically necessary to enter the robot work area, carry out all tasks outside the area.
(2) Before teaching the robot, check that the robot and its peripheral devices are all in the normal operating condition.
(3) If it is inevitable to enter the robot work area to teach the robot, check the locations, settings, and other conditions of the safety devices (such as the EMERGENCY STOP button, the DEADMAN switch on the teach pendant) before entering the area.
(4) The programmer must be extremely careful not to let anyone else enter the robot work area.
(5) Programming should be done outside the area of the safety fence as far as possible. If programming needs to be done in the area of the safety fence, the programmer should take the following precautions:
- Before entering the area of the safety fence, ensure that there is no risk of dangerous situations in the area.
- Be prepared to press the emergency stop button whenever necessary.
- Robot motions should be made at low speeds.
- Before starting programming, check the entire system status to ensure that no remote instruction to the peripheral equipment or motion would be dangerous to the user.

> Our operator panel is provided with an emergency stop button and a key switch (mode switch) for selecting the automatic operation mode (AUTO) and the teach modes (T1 and T2). Before entering the inside of the safety fence for the purpose of teaching, set the switch to a teach mode, remove the key from the mode switch to prevent other people from changing the operation mode carelessly, then open the safety gate. If the safety gate is opened with the automatic operation mode set, the robot stops (Please refer to "STOP TYPE OF ROBOT" in SAFETY PRECAUTIONS for detail of stop type). After the switch is set to a teach mode, the safety gate is disabled. The programmer should understand that the safety gate is disabled and is responsible for keeping other people from entering the inside of the safety fence.

> Our teach pendant is provided with a DEADMAN switch as well as an emergency stop button. These button and switch function as follows:
> (1) Emergency stop button: Causes an emergency stop (Please refer to "STOP TYPE OF ROBOT" in SAFETY PRECAUTIONS for detail of stop type) when pressed.
> (2) DEADMAN switch: Functions differently depending on the teach pendant enable/disable switch setting status.
> (a) Disable: The DEADMAN switch is disabled.
> (b) Enable: Servo power is turned off when the operator releases the DEADMAN switch or when the operator presses the switch strongly.
> Note) The DEADMAN switch is provided to stop the robot when the operator releases the teach pendant or presses the pendant strongly in case of emergency. The R-30*i*B employs a 3-position DEADMAN switch, which allows the robot to operate when the 3-position DEADMAN switch is pressed to its intermediate point. When the operator releases the DEADMAN switch or presses the switch strongly, the robot stops immediately.

> The operator's intention of starting teaching is determined by the controller through the dual operation of setting the teach pendant enable/disable switch to the enable position and pressing the DEADMAN switch. The operator should make sure that the robot could operate in such conditions and be responsible in carrying out tasks safely.

> Based on the risk assessment by FANUC, number of operation of DEADMAN SW should not exceed about 10000 times per year.

The teach pendant, operator panel, and peripheral device interface send each robot start signal. However the validity of each signal changes as follows depending on the mode switch and the DEADMAN switch of the operator panel, the teach pendant enable switch and the remote condition on the software.

In case of R-30*i*B controller

Mode	Teach pendant enable switch	Software remote condition	Teach pendant	Operator panel	Peripheral device
AUTO mode	On	Local	Not allowed	Not allowed	Not allowed
		Remote	Not allowed	Not allowed	Not allowed
	Off	Local	Not allowed	Allowed to start	Not allowed
		Remote	Not allowed	Not allowed	Allowed to start
T1, T2 mode	On	Local	Allowed to start	Not allowed	Not allowed
		Remote	Allowed to start	Not allowed	Not allowed
	Off	Local	Not allowed	Not allowed	Not allowed
		Remote	Not allowed	Not allowed	Not allowed

T1,T2 mode: DEADMAN switch is effective.

(6) To start the system using the operator’s panel, make certain that nobody is the robot work area and that there are no abnormal conditions in the robot work area.
(7) When a program is completed, be sure to carry out a test operation according to the procedure below.
 (a) Run the program for at least one operation cycle in the single step mode at low speed.
 (b) Run the program for at least one operation cycle in the continuous operation mode at low speed.
 (c) Run the program for one operation cycle in the continuous operation mode at the intermediate speed and check that no abnormalities occur due to a delay in timing.
 (d) Run the program for one operation cycle in the continuous operation mode at the normal operating speed and check that the system operates automatically without trouble.
 (e) After checking the completeness of the program through the test operation above, execute it in the automatic operation mode.
(8) While operating the system in the automatic operation mode, the teach pendant operator should leave the robot work area.

3.3 SAFETY OF THE MAINTENANCE ENGINEER

For the safety of maintenance engineer personnel, pay utmost attention to the following.

(1) During operation, never enter the robot work area.
(2) A hazardous situation may arise when the robot or the system, are kept with their power-on during maintenance operations. Therefore, for any maintenance operation, the robot and the system should be put into the power-off state. If necessary, a lock should be in place in order to prevent any other person from turning on the robot and/or the system. In case maintenance needs to be executed in the power-on state, the emergency stop button must be pressed.
(3) If it becomes necessary to enter the robot operation range while the power is on, press the emergency stop button on the operator panel, or the teach pendant before entering the range. The maintenance personnel must indicate that maintenance work is in progress and be careful not to allow other people to operate the robot carelessly.
(4) When entering the area enclosed by the safety fence, the maintenance worker must check the entire system in order to make sure no dangerous situations exist. In case the worker needs to enter the safety area whilst a dangerous situation exists, extreme care must be taken, and entire system status must be carefully monitored.
(5) Before the maintenance of the pneumatic system is started, the supply pressure should be shut off and the pressure in the piping should be reduced to zero.

(6) Before the start of teaching, check that the robot and its peripheral devices are all in the normal operating condition.
(7) Do not operate the robot in the automatic mode while anybody is in the robot work area.
(8) When you maintain the robot alongside a wall or instrument, or when multiple workers are working nearby, make certain that their escape path is not obstructed.
(9) When a tool is mounted on the robot, or when any moving device other than the robot is installed, such as belt conveyor, pay careful attention to its motion.
(10) If necessary, have a worker who is familiar with the robot system stand beside the operator panel and observe the work being performed. If any danger arises, the worker should be ready to press the EMERGENCY STOP button at any time.
(11) When replacing a part, please contact FANUC service center. If a wrong procedure is followed, an accident may occur, causing damage to the robot and injury to the worker.
(12) When replacing or reinstalling components, take care to prevent foreign material from entering the system.
(13) When handling each unit or printed circuit board in the controller during inspection, turn off the circuit breaker to protect against electric shock.
If there are two cabinets, turn off the both circuit breaker.
(14) A part should be replaced with a part recommended by FANUC. If other parts are used, malfunction or damage would occur. Especially, a fuse that is not recommended by FANUC should not be used. Such a fuse may cause a fire.
(15) When restarting the robot system after completing maintenance work, make sure in advance that there is no person in the work area and that the robot and the peripheral devices are not abnormal.
(16) When a motor or brake is removed, the robot arm should be supported with a crane or other equipment beforehand so that the arm would not fall during the removal.
(17) Whenever grease is spilled on the floor, it should be removed as quickly as possible to prevent dangerous falls.
(18) The following parts are heated. If a maintenance worker needs to touch such a part in the heated state, the worker should wear heat-resistant gloves or use other protective tools.
- Servo motor
- Inside the controller
- Reducer
- Gearbox
- Wrist unit

(19) Maintenance should be done under suitable light. Care must be taken that the light would not cause any danger.
(20) When a motor, reducer, or other heavy load is handled, a crane or other equipment should be used to protect maintenance workers from excessive load. Otherwise, the maintenance workers would be severely injured.
(21) The robot should not be stepped on or climbed up during maintenance. If it is attempted, the robot would be adversely affected. In addition, a misstep can cause injury to the worker.
(22)When performing maintenance work in high place, secure a footstep and wear safety belt.
(23) After the maintenance is completed, spilled oil or water and metal chips should be removed from the floor around the robot and within the safety fence.
(24) When a part is replaced, all bolts and other related components should put back into their original places. A careful check must be given to ensure that no components are missing or left not mounted.
(25) In case robot motion is required during maintenance, the following precautions should be taken :
- Foresee an escape route. And during the maintenance motion itself, monitor continuously the whole system so that your escape route will not become blocked by the robot, or by peripheral equipment.
- Always pay attention to potentially dangerous situations, and be prepared to press the emergency stop button whenever necessary.

(26) The robot should be periodically inspected. (Refer to the robot mechanical manual and controller maintenance manual.) A failure to do the periodical inspection can adversely affect the performance or service life of the robot and may cause an accident

(27) After a part is replaced, a test operation should be given for the robot according to a predetermined method. (See TESTING section of "Controller operator's manual".) During the test operation, the maintenance staff should work outside the safety fence.

4 SAFETY OF THE TOOLS AND PERIPHERAL DEVICES

4.1 PRECAUTIONS IN PROGRAMMING

(1) Use a limit switch or other sensor to detect a dangerous condition and, if necessary, design the program to stop the robot when the sensor signal is received.
(2) Design the program to stop the robot when an abnormal condition occurs in any other robots or peripheral devices, even though the robot itself is normal.
(3) For a system in which the robot and its peripheral devices are in synchronous motion, particular care must be taken in programming so that they do not interfere with each other.
(4) Provide a suitable interface between the robot and its peripheral devices so that the robot can detect the states of all devices in the system and can be stopped according to the states.

4.2 PRECAUTIONS FOR MECHANISM

(1) Keep the component cells of the robot system clean, and operate the robot in an environment free of grease, water, and dust.
(2) Don't use unconfirmed liquid for cutting fluid and cleaning fluid.
(3) Employ a limit switch or mechanical stopper to limit the robot motion so that the robot or cable does not strike against its peripheral devices or tools.
(4) Observe the following precautions about the mechanical unit cables. When theses attentions are not kept, unexpected troubles might occur.
   - Use mechanical unit cable that have required user interface.
   - Don't add user cable or hose to inside of mechanical unit.
   - Please do not obstruct the movement of the mechanical unit cable when cables are added to outside of mechanical unit.
   - In the case of the model that a cable is exposed, Please do not perform remodeling (Adding a protective cover and fix an outside cable more) obstructing the behavior of the outcrop of the cable.
   - Please do not interfere with the other parts of mechanical unit when install equipments in the robot.
(5) The frequent power-off stop for the robot during operation causes the trouble of the robot. Please avoid the system construction that power-off stop would be operated routinely. (Refer to bad case example.) Please execute power-off stop after reducing the speed of the robot and stopping it by hold stop or cycle stop when it is not urgent. (Please refer to "STOP TYPE OF ROBOT" in SAFETY PRECAUTIONS for detail of stop type.)
   (Bad case example)
   - Whenever poor product is generated, a line stops by emergency stop.
   - When alteration was necessary, safety switch is operated by opening safety fence and power-off stop is executed for the robot during operation.
   - An operator pushes the emergency stop button frequently, and a line stops.
   - An area sensor or a mat switch connected to safety signal operate routinely and power-off stop is executed for the robot.
(6) Robot stops urgently when collision detection alarm (SRVO-050) etc. occurs. The frequent urgent stop by alarm causes the trouble of the robot, too. So remove the causes of the alarm.

5 SAFETY OF THE ROBOT MECHANISM

5.1 PRECAUTIONS IN OPERATION

(1) When operating the robot in the jog mode, set it at an appropriate speed so that the operator can manage the robot in any eventuality.
(2) Before pressing the jog key, be sure you know in advance what motion the robot will perform in the jog mode.

5.2 PRECAUTIONS IN PROGRAMMING

(1) When the work areas of robots overlap, make certain that the motions of the robots do not interfere with each other.
(2) Be sure to specify the predetermined work origin in a motion program for the robot and program the motion so that it starts from the origin and terminates at the origin.
Make it possible for the operator to easily distinguish at a glance that the robot motion has terminated.

5.3 PRECAUTIONS FOR MECHANISMS

(1) Keep the work areas of the robot clean, and operate the robot in an environment free of grease, water, and dust.

5.4 PROCEDURE TO MOVE ARM WITHOUT DRIVE POWER IN EMERGENCY OR ABNORMAL SITUATIONS

For emergency or abnormal situations (e.g. persons trapped in or by the robot), brake release unit can be used to move the robot axes without drive power.
Please refer to controller maintenance manual and mechanical unit operator’s manual for using method of brake release unit and method of supporting robot.

6 SAFETY OF THE END EFFECTOR

6.1 PRECAUTIONS IN PROGRAMMING

(1) To control the pneumatic, hydraulic and electric actuators, carefully consider the necessary time delay after issuing each control command up to actual motion and ensure safe control.
(2) Provide the end effector with a limit switch, and control the robot system by monitoring the state of the end effector.

7 STOP TYPE OF ROBOT

The following three robot stop types exist:

Power-Off Stop (Category 0 following IEC 60204-1)

Servo power is turned off and the robot stops immediately. Servo power is turned off when the robot is moving, and the motion path of the deceleration is uncontrolled.

The following processing is performed at Power-Off stop.

- An alarm is generated and servo power is turned off.
- The robot operation is stopped immediately. Execution of the program is paused.

Controlled stop (Category 1 following IEC 60204-1)

The robot is decelerated until it stops, and servo power is turned off.

The following processing is performed at Controlled stop.

- The alarm "SRVO-199 Controlled stop" occurs along with a decelerated stop. Execution of the program is paused.
- An alarm is generated and servo power is turned off.

Hold (Category 2 following IEC 60204-1)

The robot is decelerated until it stops, and servo power remains on.

The following processing is performed at Hold.

- The robot operation is decelerated until it stops. Execution of the program is paused.

> ⚠ **WARNING**
> The stopping distance and stopping time of Controlled stop are longer than the stopping distance and stopping time of Power-Off stop. A risk assessment for the whole robot system, which takes into consideration the increased stopping distance and stopping time, is necessary when Controlled stop is used.

When the emergency stop button is pressed or the FENCE is open, the stop type of robot is Power-Off stop or Controlled stop. The configuration of stop type for each situation is called *stop pattern*. The stop pattern is different according to the controller type or option configuration.

There are the following 3 Stop patterns.

Stop pattern	Mode	Emergency stop button	External Emergency stop	FENCE open	SVOFF input	Servo disconnect
A	AUTO	P-Stop	P-Stop	C-Stop	C-Stop	P-Stop
	T1	P-Stop	P-Stop	-	C-Stop	P-Stop
	T2	P-Stop	P-Stop	-	C-Stop	P-Stop
B	AUTO	P-Stop	P-Stop	P-Stop	P-Stop	P-Stop
	T1	P-Stop	P-Stop	-	P-Stop	P-Stop
	T2	P-Stop	P-Stop	-	P-Stop	P-Stop
C	AUTO	C-Stop	C-Stop	C-Stop	C-Stop	C-Stop
	T1	P-Stop	P-Stop	-	C-Stop	P-Stop
	T2	P-Stop	P-Stop	-	C-Stop	P-Stop

P-Stop: Power-Off stop
C-Stop: Controlled stop
-: Disable

The following table indicates the Stop pattern according to the controller type or option configuration.

Option	R-30*i*B
Standard	A (*)
Controlled stop by E-Stop (A05B-2600-J570)	C (*)

(*) R-30*i*B does not have servo disconnect.

The stop pattern of the controller is displayed in "Stop pattern" line in software version screen. Please refer to "Software version" in operator's manual of controller for the detail of software version screen.

"Controlled stop by E-Stop" option

When "Controlled stop by E-Stop" (A05B-2600-J570) option is specified, the stop type of the following alarms becomes
Controlled stop but only in AUTO mode. In T1 or T2 mode, the stop type is Power-Off stop which is the normal operation of the system.

Alarm	Condition
SRVO-001 Operator panel E-stop	Operator panel emergency stop is pressed.
SRVO-002 Teach pendant E-stop	Teach pendant emergency stop is pressed.
SRVO-007 External emergency stops	External emergency stop input (EES1-EES11, EES2-EES21) is open. (R-30*i*B controller)
SRVO-218 Ext.E-stop/Servo Disconnect	External emergency stop input (EES1-EES11, EES2-EES21) is open. (R-30*i*B controller)
SRVO-408 DCS SSO Ext Emergency Stop	In DCS Safe I/O connect function, SSO[3] is OFF.
SRVO-409 DCS SSO Servo Disconnect	In DCS Safe I/O connect function, SSO[4] is OFF.

Controlled stop is different from Power-Off stop as follows:

- In Controlled stop, the robot is stopped on the program path. This function is effective for a system where the robot can interfere with other devices if it deviates from the program path.
- In Controlled stop, physical impact is less than Power-Off stop. This function is effective for systems where the physical impact to the mechanical unit or EOAT (End Of Arm Tool) should be minimized.
- The stopping distance and stopping time of Controlled stop is longer than the stopping distance and stopping time of Power-Off stop, depending on the robot model and axis. Please refer to the operator's manual of a particular robot model for the data of stopping distance and stopping time.

When this option is loaded, this function cannot be disabled.

The stop type of DCS Position and Speed Check functions is not affected by the loading of this option.

> ⚠ **WARNING**
> The stopping distance and stopping time of Controlled stop are longer than the stopping distance and stopping time of Power-Off stop. A risk assessment for the whole robot system, which takes into consideration the increased stopping distance and stopping time, is necessary when this option is loaded.

120919

TABLE OF CONTENTS

APPENDIX

1 OVERVIEW

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 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.

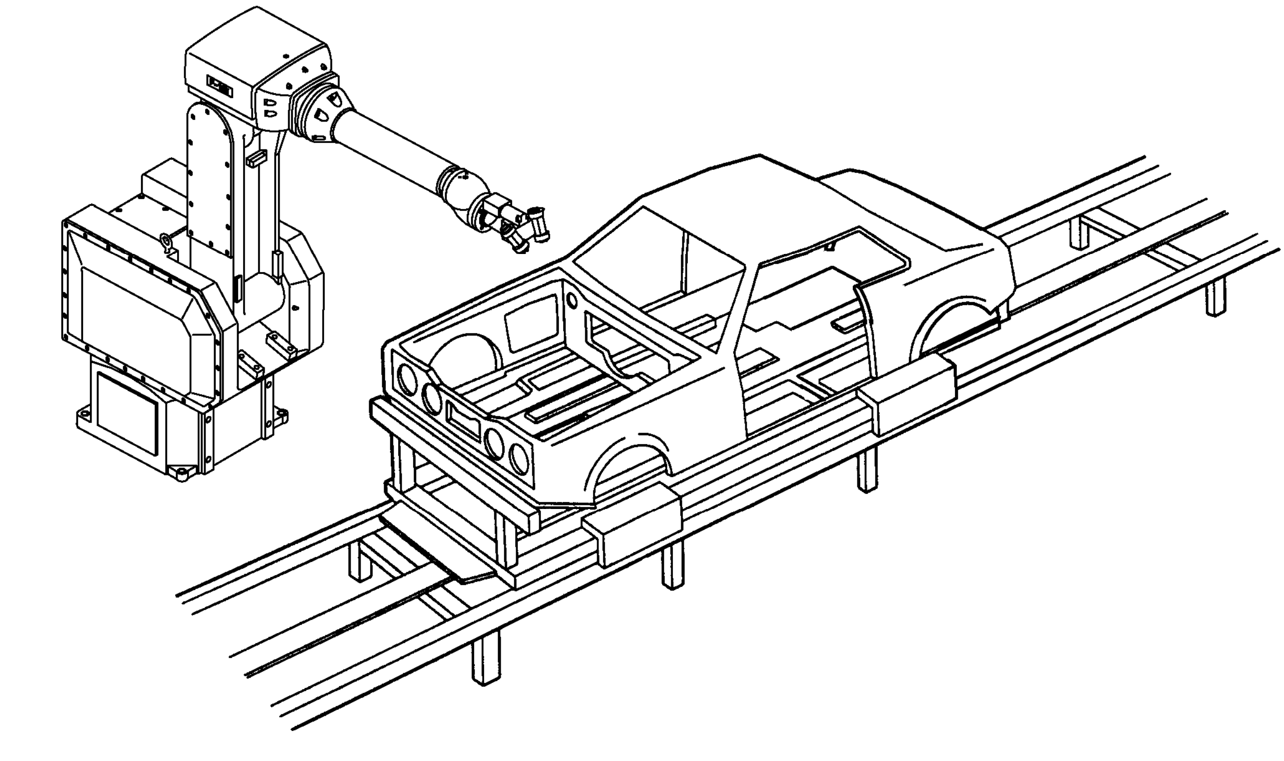


Fig. 1.1 Typical tracking workcell

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 Subsection 1.4.1 and Subsection 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 and Chapter 5 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
Circular tracking does not use tracking boundaries.

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

2 HARDWARE AND SOFTWARE

2.1 REQUIREMENTS

The line tracking system requires a line tracking interface board (option) within the controller, or uses the interface on main CPU board to connect an encoder. When a line tracking interface board is used, a fiber optic FSSB connection cable is required too.
Additionally, external hardware (a tracking encoder) and the associated interconnections (an encoder cable) are required to track the line (conveyor, platform, table, and so forth). See Fig. 2.2.1(a) - 2.2.1 (b) in using αA1000S Pulsecoder, A860-0372-T001 (which is adaptable to both incremental and absolute.). See Appendix C in using incremental Pulsecoder, A860-0301-T001 to T004.
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

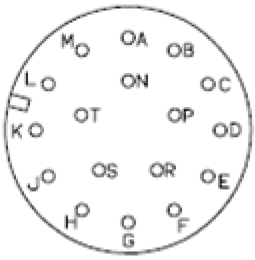
R-30*i*B line tracking system requires the items shown below. Table 2.1(a) shows the items for αA1000S Pulsecoder, A860-0372-T001. Appendix C shows the items for incremental Pulsecoder, from A860-0301-T001 to T004.

Table 2.1(a) Requirements (for αA1000S Pulsecoder A860-0372-T001)

Required Components	R-30*i*B Controller A-Cabinet	R-30*i*B Controller B-Cabinet	Comments
Hardware			
Line Tracking Interface Board	A20B-8101-0421 (wide-mini slot) or A20B-8101-0601 (mini slot)	A20B-8101-0421 (wide-mini slot) or A20B-8101-0601 (mini slot)	Separate Detector Unit (SDU) - A02B-0323-C205 can be used in place of Line tracking Interface board. NOTE: SDU requires retrofit work to mount in the container (see Fig. 2.1(i) through Fig. 2.1(m)). Main CPU board can also connect to αA1000S Pulsecoder. NOTE: In this case, learning robot option can not be used.
Fiber Optic (FSSB) Cable	A66L-6001-0023	A66L-6001-0023	
Pulsecoder Cable (In case of an αA1000S Pulsecoder as an incremental Pulsecoder)	for A20B-8101-0421 (wide-mini slot): A05B-2601-J220 (7M) A05B-2601-J221 (14M) A05B-2601-J222 (20M) A05B-2601-J223 (30M)	for A20B-8101-0421 (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.
	for A20B-8101-0601 (mini slot) (one Pulsecoder): A05B-2601-J210 (7M) A05B-2601-J211 (14M) A05B-2601-J212 (20M) A05B-2601-J213 (30M)	for A20B-8101-0601 (mini slot) (one Pulsecoder): A05B-2603-J210 (7M) A05B-2603-J211 (14M) A05B-2603-J212 (20M) A05B-2603-J213 (30M)	
	for A20B-8101-0601 (mini slot) (two Pulsecoder): A05B-2601-J260 (7M) A05B-2601-J261 (14M) A05B-2601-J262 (20M) A05B-2601-J263 (30M)	A20B-8101-0601 (mini slot) (two Pulsecoder) : A05B-2603-J260 (7M) A05B-2603-J261 (14M) A05B-2603-J262 (20M) A05B-2603-J263 (30M)	
for main CPU board	A05B-2601-J270(7M) A05B-2601-J271(14M) A05B-2601-J272(20M) A05B-2601-J273(30M)	A05B-2603-J270(7M) A05B-2603-J271(14M) A05B-2603-J272(20M) A05B-2603-J273(30M)	[PSG corrected numbers for the 20m and 30m]
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.)			

Refer to Fig. 2.1(a) and (c) for Pulsecoder signal information, and images containing the dimensions of the encoders.
Refer to Fig. 2.1(d) through Fig. 2.1(m) for information on dimensions, connections, and installation of the Detector Interface Units.

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

Fig. 2.1 (a) αA1000S pulsecoder (A860-0372-T001) connection signal information

Item		Specification
Power voltage		5[V]±5%
Current consumption		Up to 0.3[A]
Working temperature range		0 to 60[℃]
Resolution		1 000 000[/rev]
Maximum speed of revolution		4000[min^{-1}]
Input shaft inertia		Up to 1×10^{-4}[kg m^{2}]
Input shaft startup torque		Up to 0.1[N m]
Ratio loads	Radial	2.0[N]
	Axial	1.0[N]
Shaft diameter runout		0.02×10^{-3}[m]
Configuration		dust proof and drip-proof (equivalent to IP55: when using waterproof connector)
Vibration proof acceleration		5[G] (50 to 2000[Hz])
Weight		Approx. 0.75[kg]

Fig. 2.1 (b) αA1000S pulsecoder (A860-0372-T001) specifications

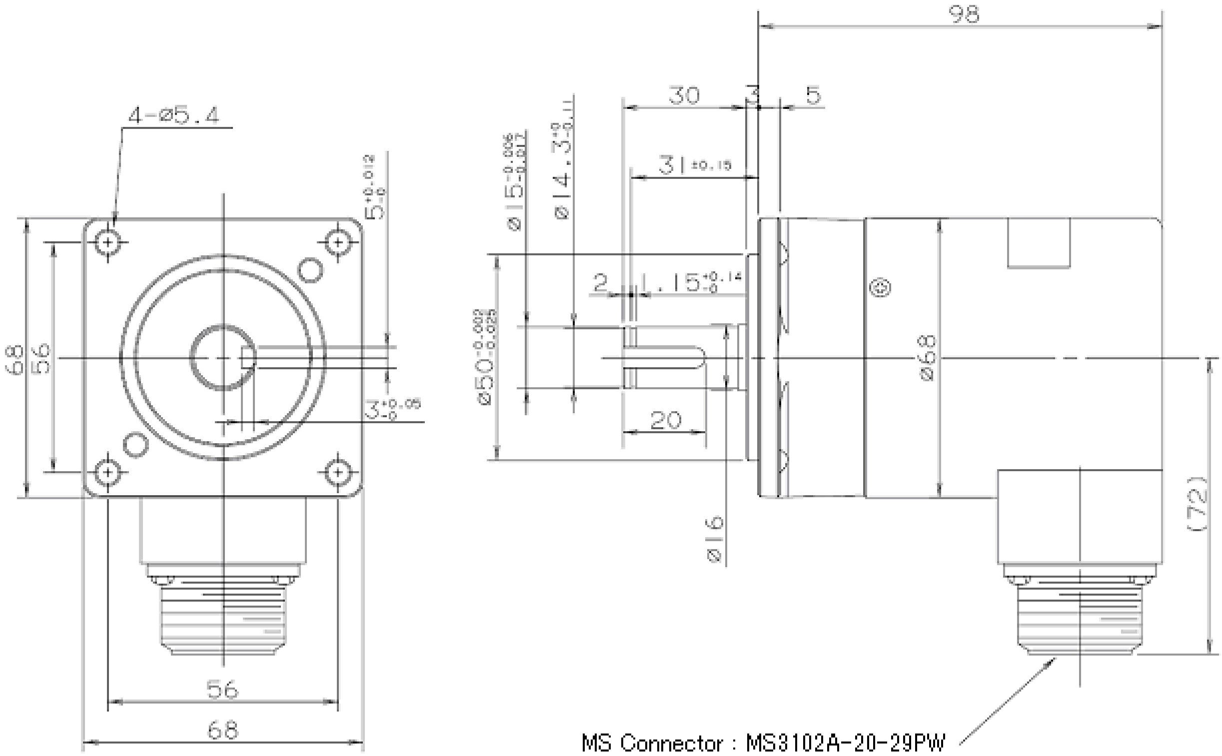


Fig. 2.1 (c) αA1000S pulsecoder dimensions (A860-0372-T001)

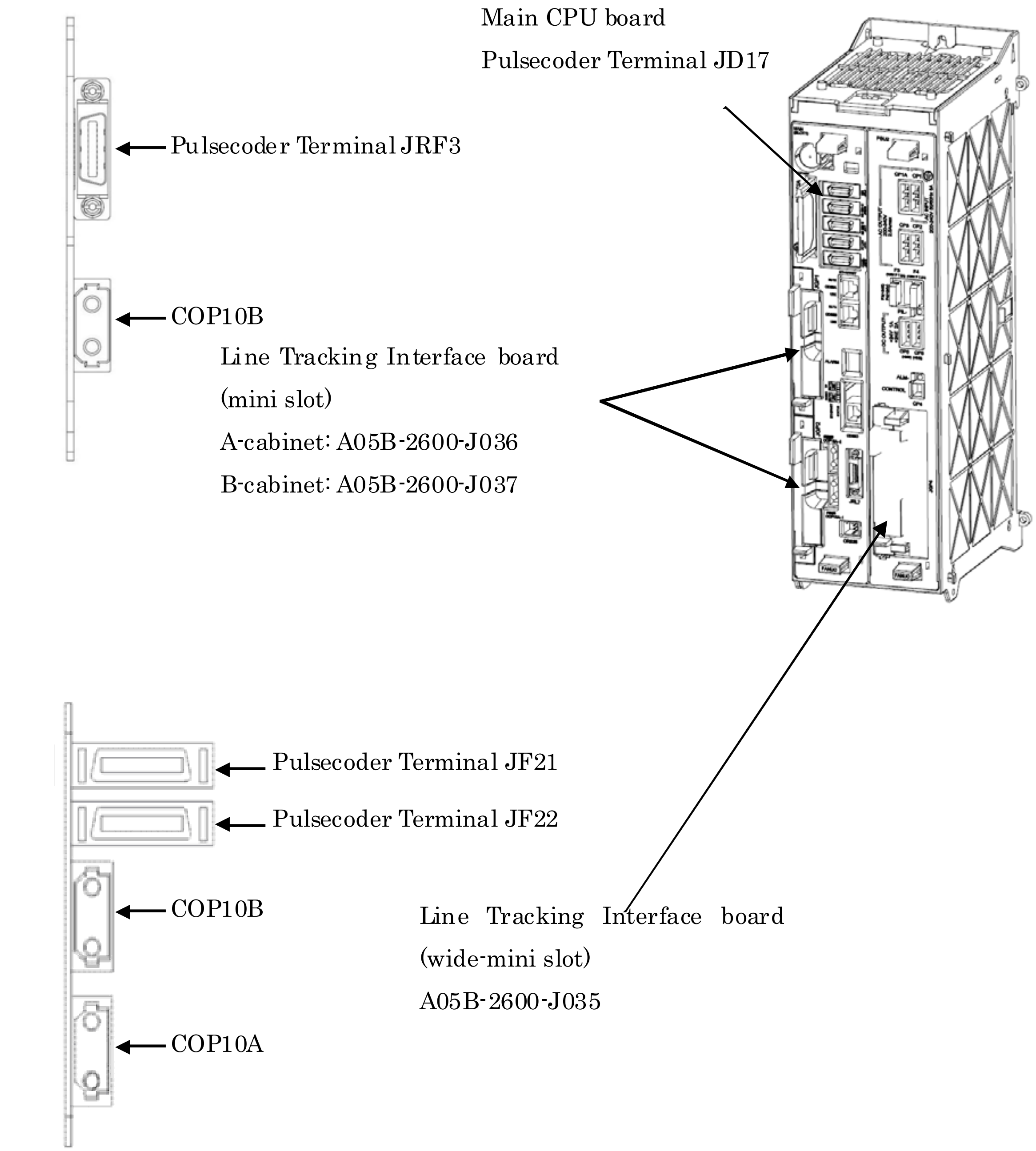


Fig. 2.1(d) Controller with line tracking connections

> **NOTE**
> If the line tracking interface board and main CPU board cannot be used or is not available, you can use the SDU shown in Fig. 2.1(e).

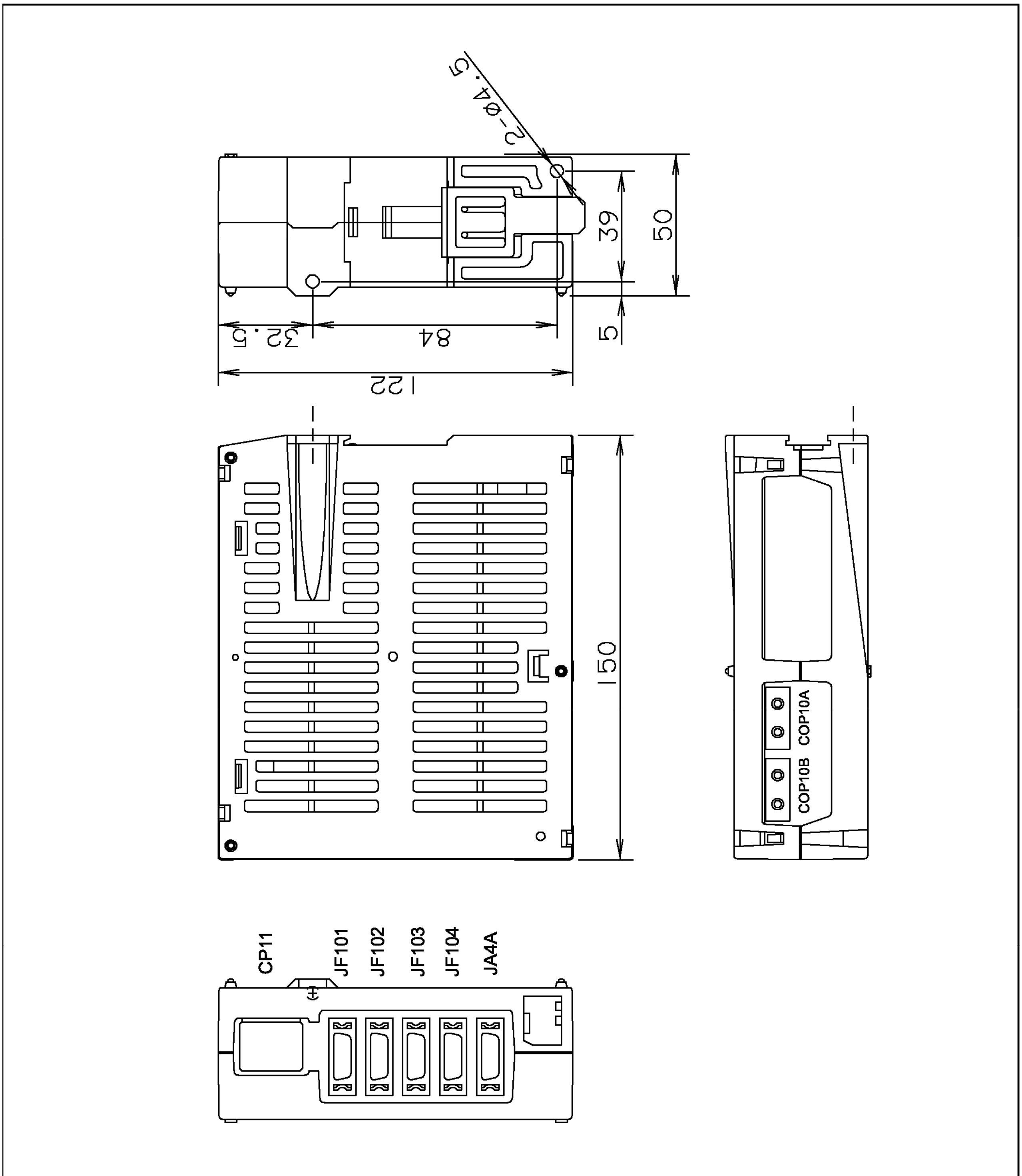


Fig. 2.1(e) External dimensions of separate detector interface unit

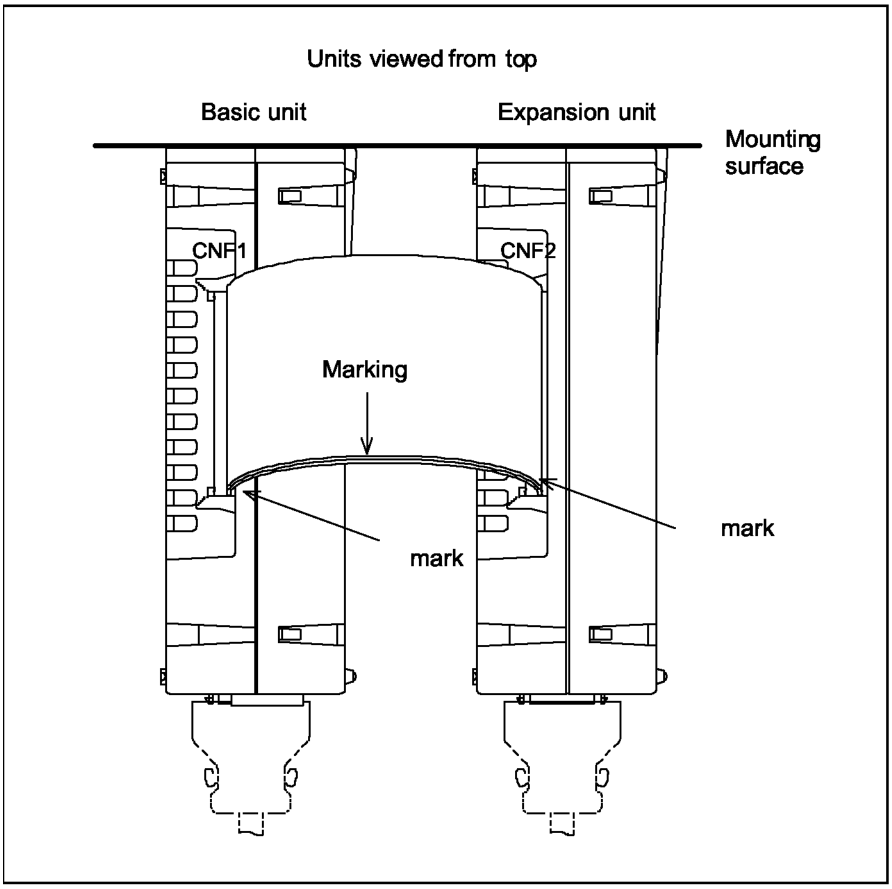


Fig. 2.1(f) Cable connection between basic unit and expansion unit

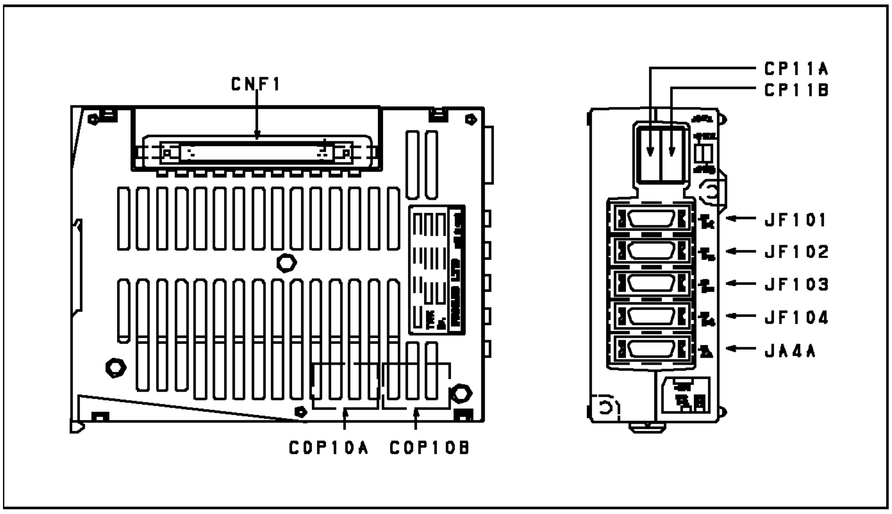


Fig. 2.1(g) Connector locations on the basic unit

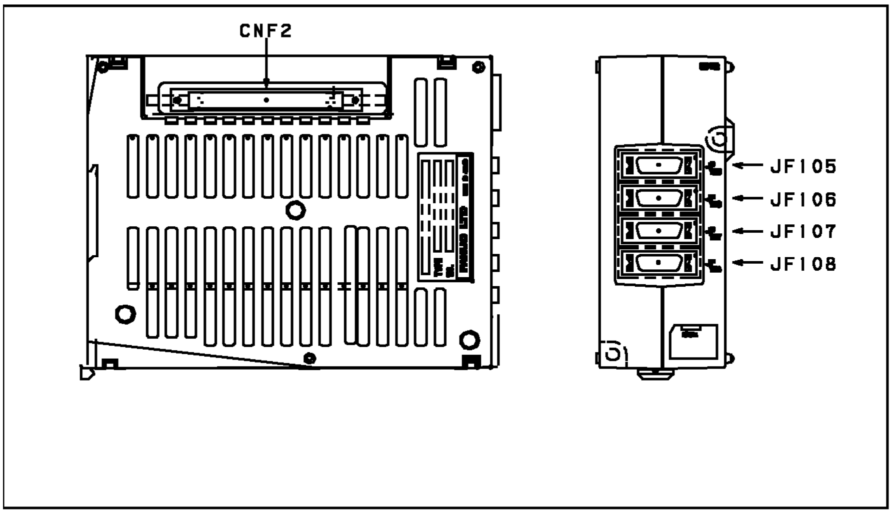


Fig. 2.1(h) Connector locations on the expansion unit

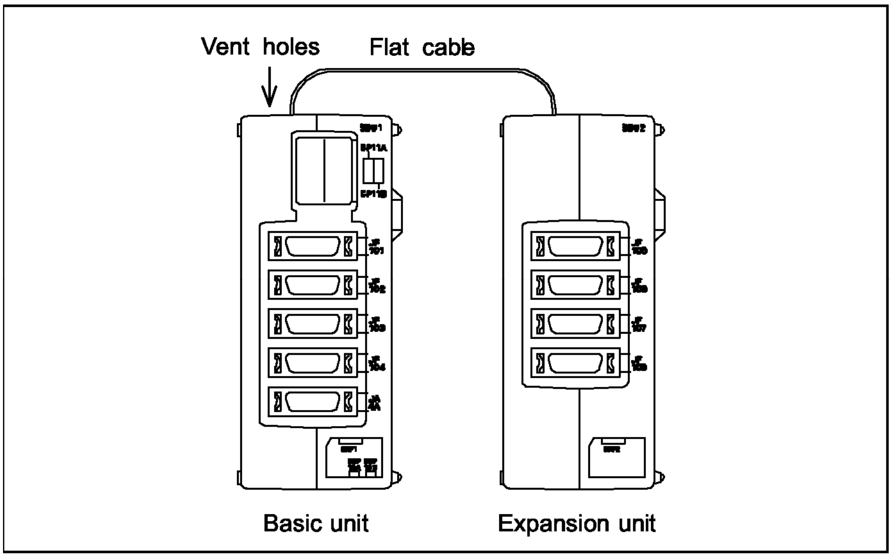


Fig. 2.1(i) Flat cable placement during installation

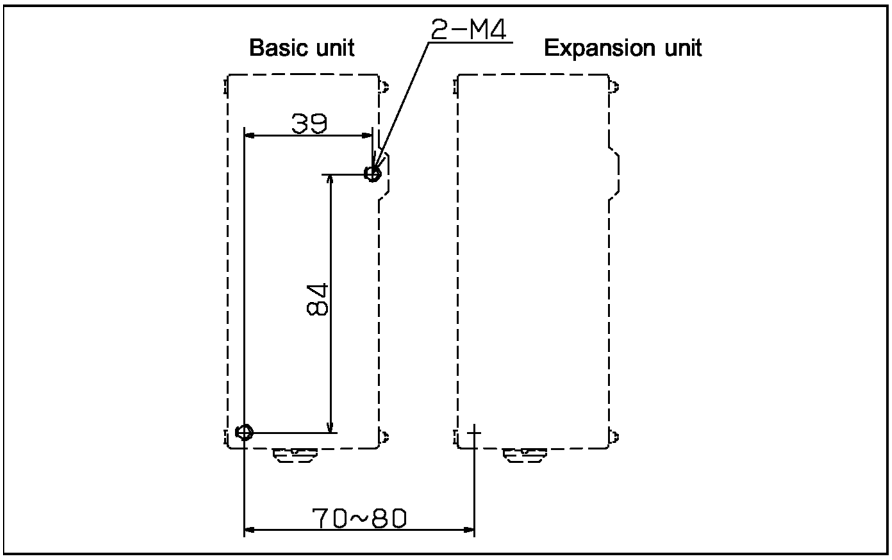
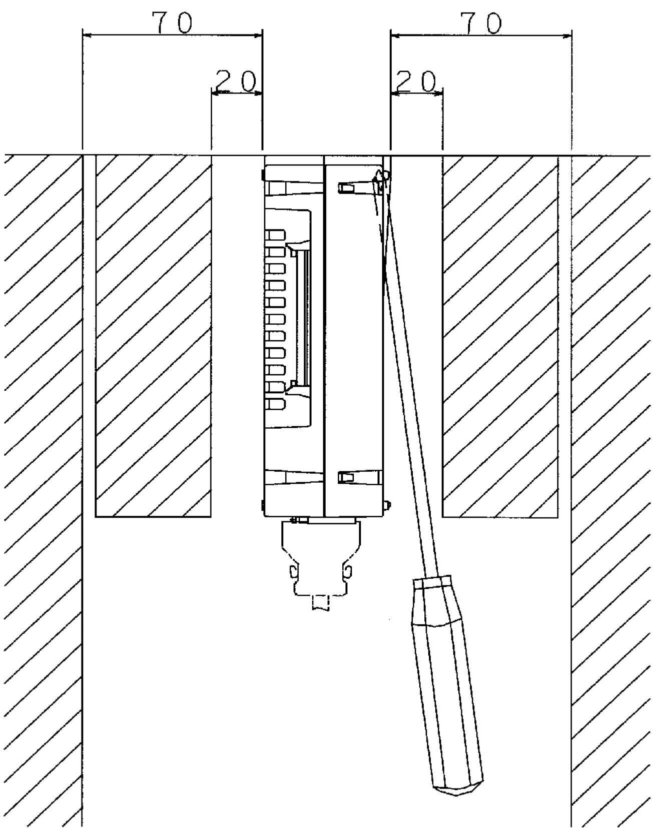


Fig. 2.1(j) Horizontal separation of mounting holes during installation

⚠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.



Access clearance near a separate detector interface unit

Fig. 2.1(k) Accessing the 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).

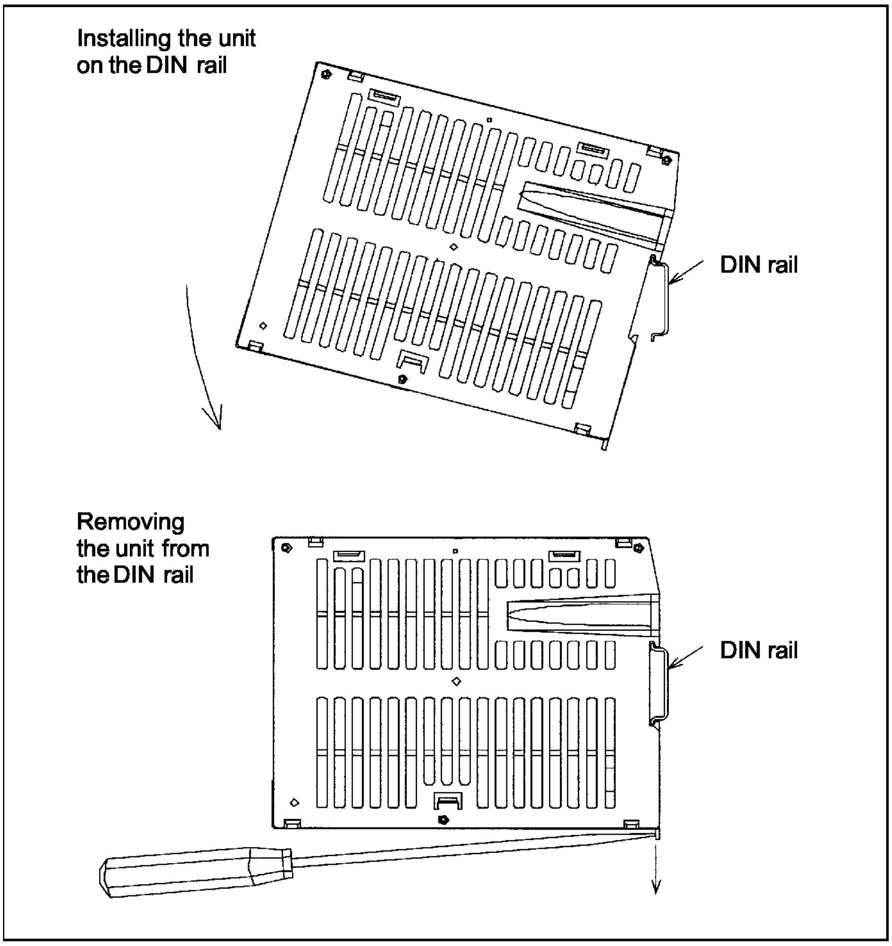


Fig. 2.1(l) Installing and removing the unit

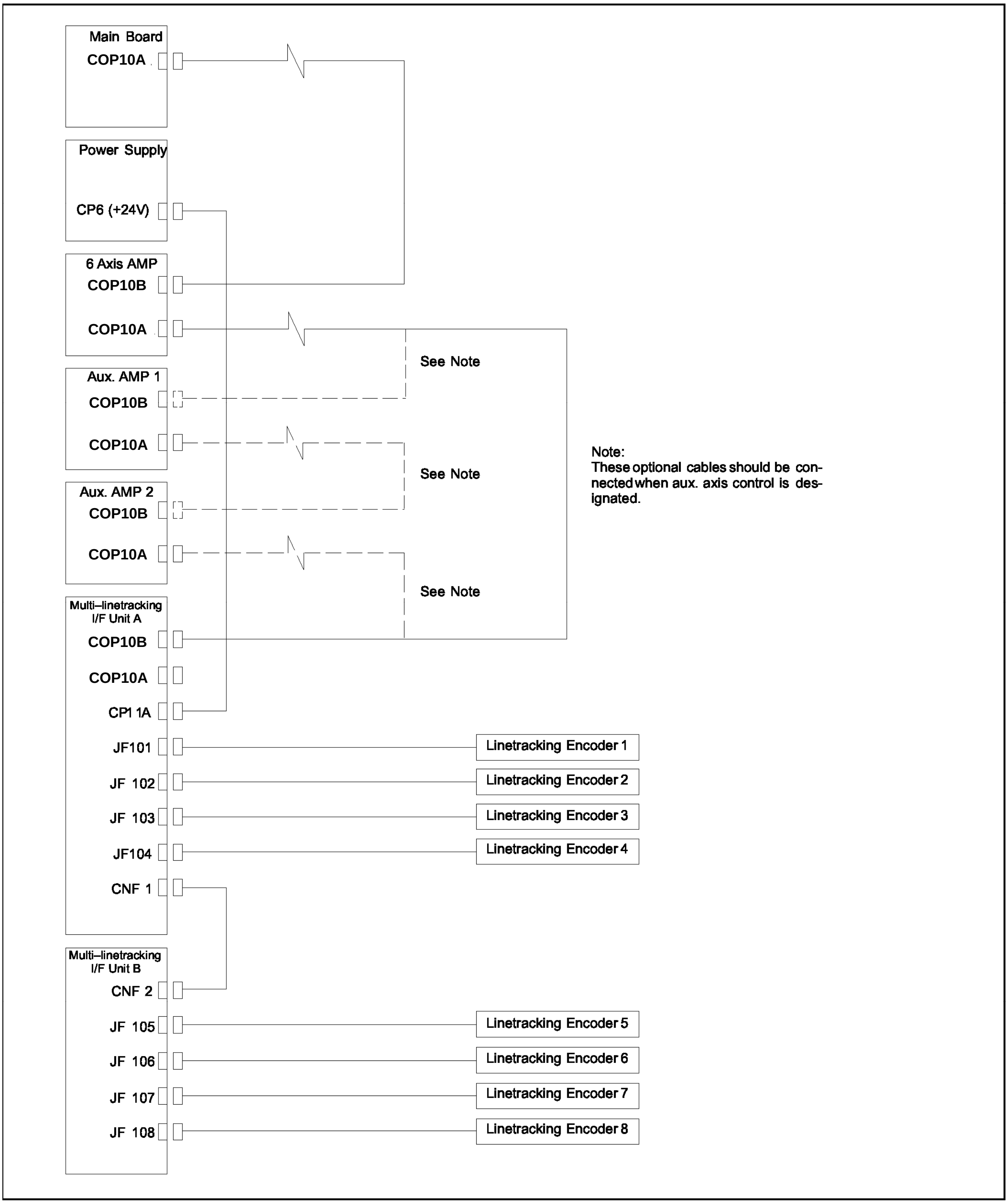


Fig. 2.1(m) Connection diagram

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 Fig. 2.1(d). If Separate Detector Interface units are used, they can be mounted in the cabinet separately from the main CPU board. See Fig. 2.1(e) through Fig. 2.1(m).

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 Fig. 2.1(d).)
- When the main CPU board is connected to encoder, there are no additional fiber optic FSSB cables.
- If Separate Detector Interface units are used, see Fig. 2.1(m) for a connection diagram.

Tracking Encoder

3 kinds of Encoders (Pulsecoders) that can be used on R-30*i*B robots are showed below.

- αA1000S Pulsecoder A860-0372-T001 (available as both incremental and absolute)
- Incremental Pulsecoder A860-0301-T001 to T004

Normally, use αA1000S Pulsecoder A860-0372-T001.
Make sure you use appropriate gear or reducer to get desirable resolution (typically 30-80 pulses per mm for Line Tracking).

Part Detect Switch

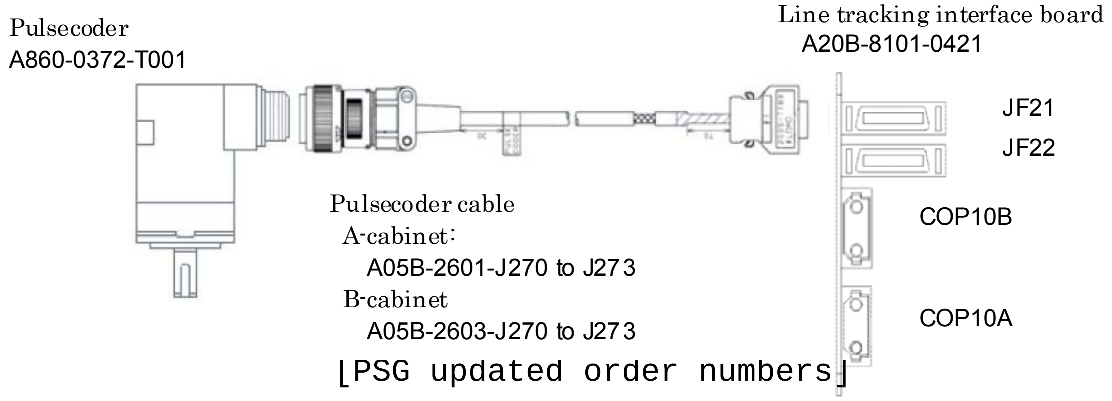
A part detect switch must be installed, as a digital input, to monitor when a part on the conveyor is approaching the robot workspace. Refer to "Digital I/O" in R-30*i*B CONTROLLER OPERATOR'S MANUAL (Basic Operation) B-83284EN for more information on setting up a digital input.
This switch might be one 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.

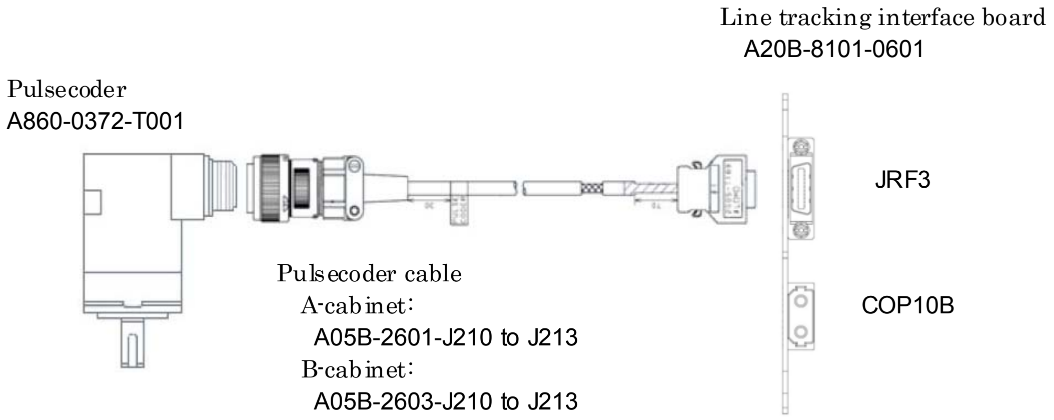
> **NOTE**
> Tracking accuracy depends on the precision of the trigger switch. A faster part detect switch gives a more precise trigger value.

Pulse Multiplexer

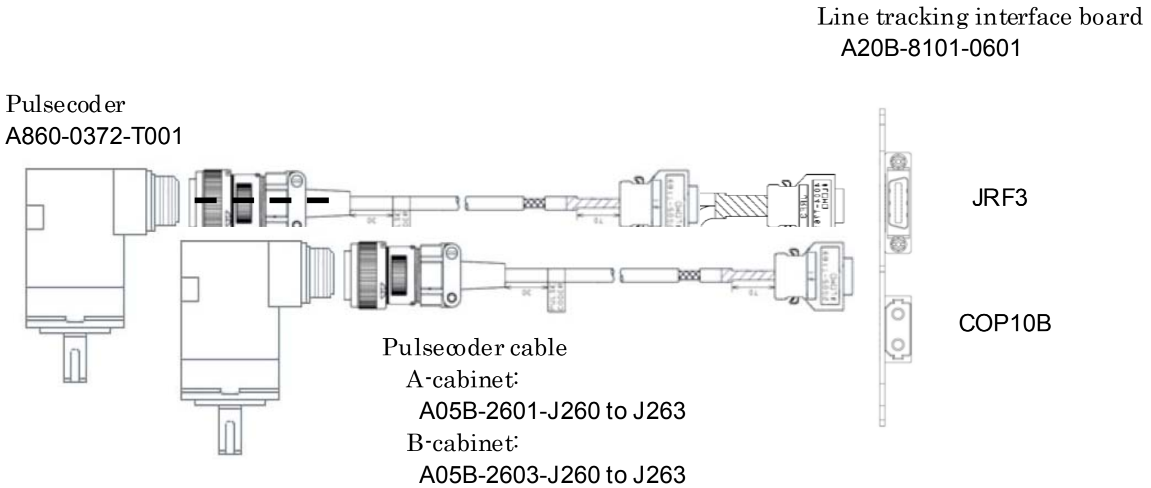
When using multiple robots, input the value of the encoder to each robot controller via the pulse multiplexer. Connect the line tracking cables and power cable to the pulse multiplexer as shown in Appendix C. When the pulse multiplexer is used, αA1000S Pulsecoder A860-0372-T001 can not be used. If you want to use multiple robots with αA1000S Pulsecoder, use Ethernet encoder function (option).



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

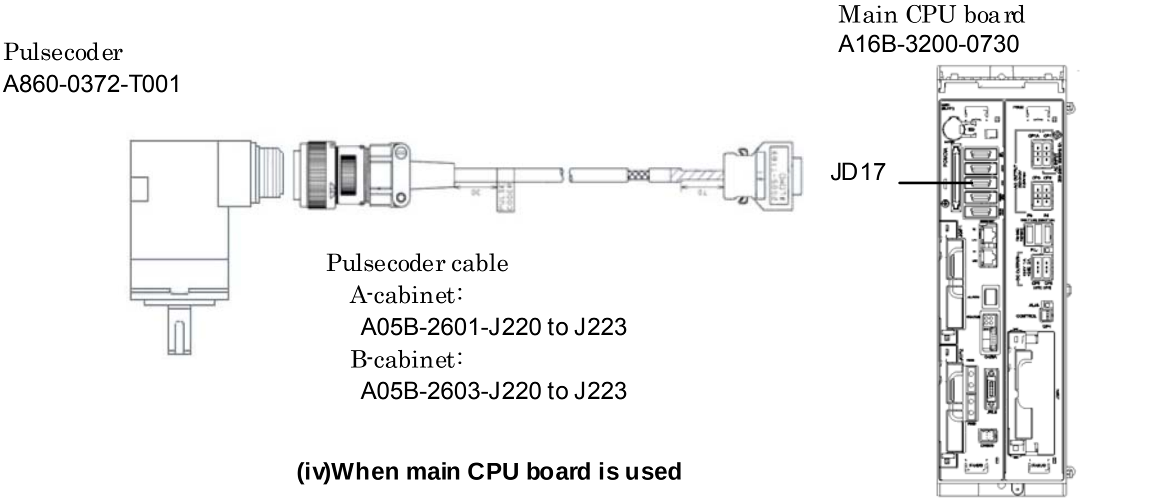


(ii) When A20B-8101-0601 is used with one Pulsecoder.



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

Fig. 2.2.1(a) Connecting cables 1 (Pulsecoder, A860-0372-T001)



(iv)When main CPU board is used

Fig. 2.2.1(b) Connecting cables 2 (Pulsecoder, A860-0372-T001)

2.2.2 Software

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

2.2.3 Restriction

Line tracking can not be used with below 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

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

3 LINE TRACKING SETUP

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.

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

Table 3.1.1 Encoder setup items

ENCODER SETUP SCREEN PARAMETERS	DESCRIPTION	RELATED SYSTEM VARIABLE
Encoder Number Value: 1 - 8 Default: 1	This item is the schedule selection number of the encoder you are setting up.	N/A
Encoder Axis Value: 0 – 32 Default: 0	This item allows you to select the encoder axis to set up.	$SCR.$ENC_AXIS[ ]
Encoder Type Value: 0 = incremental 1 = absolute Default: 0	This item specifies the type of tracking to be used.	$SCR.$ENC_TYPE[ ]
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: 1	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: 1	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: 0	This item is the number of encoder counts per encoder update. If the encoder counts per update go 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

ENCODER SETUP SCREEN PARAMETERS	DESCRIPTION	RELATED SYSTEM VARIABLE
Simulate Rate(cnt/updt) Value: Integer Default: 0	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

Procedure 3-1 Encoder Setup

Steps

1. Press MENUS.
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
[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-0323-C205) and SDU2 (part number A02B-0323-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.
   - For ABSOLUTE, press F4 [CHOICE]. Select ABSOLUTE.
   - For INCREMENTAL, press F4 [CHOICE]. Select INCREMENTAL.

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 time increment (ITP_TIME:8msec).

$$\text{Multiplier} \times \text{ITP\_TIME(ms)} = \text{encoder 1update(ms)}$$

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 count 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. 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).
15. Verify that you have set up the encoder correctly. Refer to Subsection 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

Fig. 3.1.2 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.

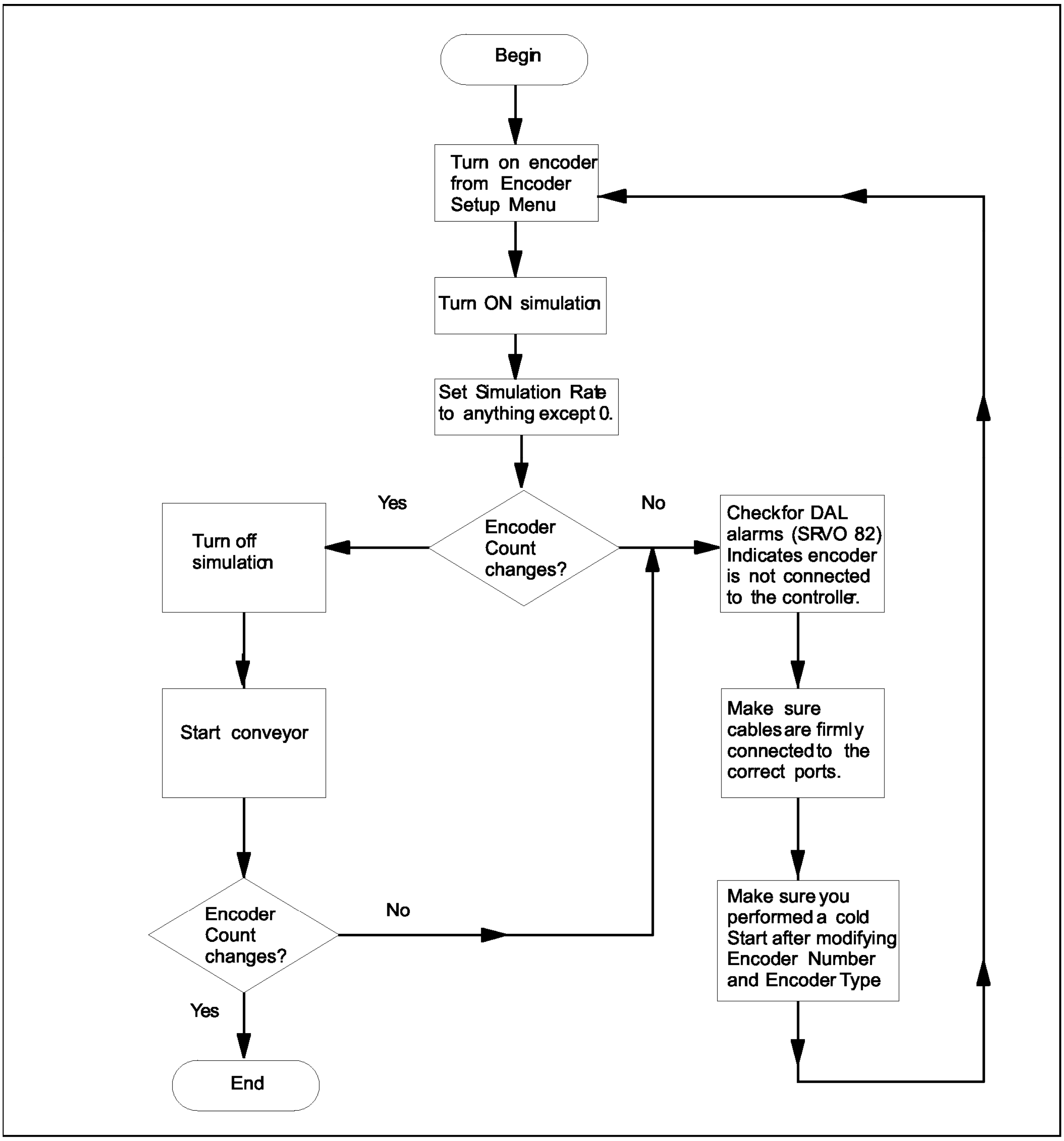


Fig. 3.1.2 Verify encoder setup

3.2 COLD START

If you have modified Encoder Axis or Encoder Type on the Encoder Setup Screen (Procedure 3-2), you must perform a Cold start.
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

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 work cell.

> ⚠**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              XXXX/XX

     Copyright xxxx, All Rights Reserved
            FANUC CORPORATION
     FANUC Robotics 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 precede 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              XXXX/XX

     Copyright xxxx, All Rights Reserved
           FANUC CORPORATION
     FANUC Robotics 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 precede 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 Subsection 3.3.2
- Verify that you have set up tracking correctly, Subsection 3.3.3

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

The Tracking Setup Screen parameters are contained in the system variable structure $LNSCH. Refer to Table 3.3 for an overview of each tracking setup item and its related system variable.

Table 3.3 Tracking setup items

TRACKING SETUP PARAMETERS	DESCRIPTION	RELATED SYSTEM VARIABLES
Schedule Number Value: 1 - 8 Default: 1	This item is the schedule number for a tracking program.	N/A
Robot Tracking GroupValue: 1 - 5 Default: 1	This item specifies the robot motion group associated with the current tracking schedule.	$LNSCH.$TRK_GRP_NUM

TRACKING SETUP PARAMETERS	DESCRIPTION	RELATED SYSTEM VARIABLES
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 - 8 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
Encoder Scale Factor(cnt/mm) or (cnt/deg) Value: -99999.0 to 99999.0 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

TRACKING SETUP PARAMETERS	DESCRIPTION	RELATED SYSTEM VARIABLES
Trigger INPUT Number Value: 0 - 4096 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
Selected B0oundary 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: -99999.0 to 99999.0 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 MENUS.
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)                    1/5
 Track Schedule Number: 1
   Frame Components:
       X:   0.00    Y:   0.00   Z:   0.00
       W:   0.00    P:   0.00   R:   0.00
Teach  Data:
 Origin:        UNINIT          Enc_cnt:        0.0
 X:       0.00     Y:       0.00     Z:       0.00
 +X  dir:       UNINIT          Enc_cnt:        0.0
 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 (Cir)                 1/5
 Track Schedule Number: 1
   Frame Components:
       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:       0.0
 X:      0.00    Y:      0.00     Z:      0.00
 +Y  dir:        UNINIT          Enc_cnt:       0.0
 X:      0.00    Y:      0.00     Z:      0.00
 Assist:         UNINIT
 X:      0.00    Y:      0.00     Z:      0.00
Scale  (cnt/deg): 8.73

         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 Subsection 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 (Subsection 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.
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 Subsection 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 Fig. 3.3(a).

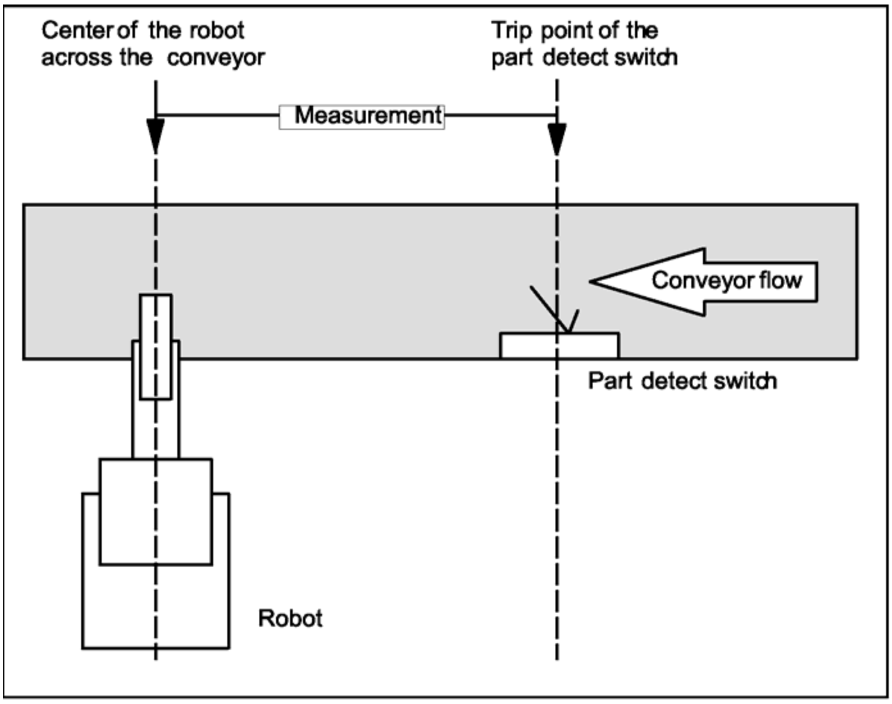


Fig. 3.3(a) Part detect switch (line and rail tracking)

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 0-4096.
    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 Fig. 3.3(b).
    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 should not be set.

23. Move the cursor to Boundary Set 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 Fig. 3.3(b).
    - Conveyor positions further up-stream of this position is 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.
    - 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 Fig. 3.3(b).
24. Move the cursor to Boundary Set Down. 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 Fig. 3.3(b).
    - Conveyor positions further up-stream of this position is considered either IN-WINDOW or IN-BOUND.
    - Conveyor positions further down-stream of this position is 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.
    - 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 Fig. 3.3(b).
25. Verify that you have set up tracking correctly. Refer to Subsection 3.3.3. This should be done before you run 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.

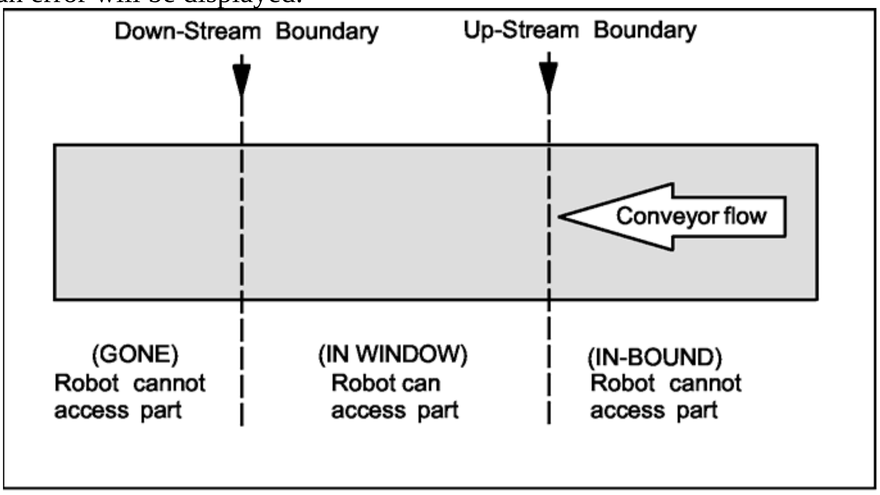


Fig. 3.3(b) Boundary window

The following lists some example values for a line tracking system. All Down bounds values are greater than their corresponding Up bounds 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 three values:

- UNINIT - indicates that the position is un-initialized
- RECORDED - indicates that the position has been recorded but not yet used during processing
- PROCESSED - indicates that the position has been recorded and already used to compute a new nominal tracking frame

When any of these positions is selected, the word RECORD appears above the F2 function key. Pressing SHIFT and RECORD 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 “Setting a Tool Coordinate System” in R-30*i*B CONTROLLER OPERATOR’S MANUAL (Basic Operation) B-83284EN 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 Fig. 3.3.1(a).
- 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.

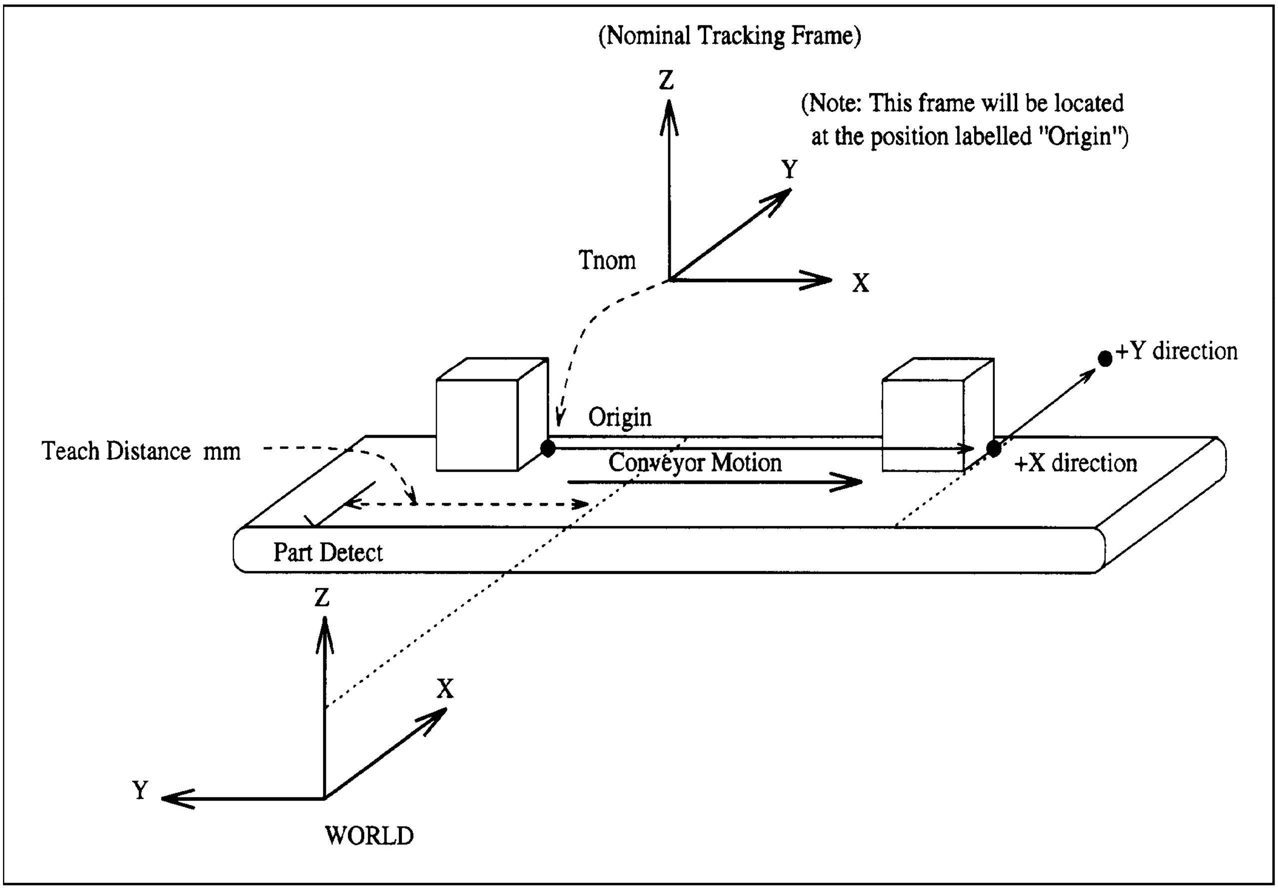


Fig. 3.3.1(a) Nominal track frame - line conveyor motion

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 Schedule Number: 1
   Frame Components:
       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:        2356
 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, 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 Schedule Number: 1
   Frame Components:
X:  2241.80     Y:    754.09     Z:         30.00
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:       2356
 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. If you do not want to setup the encoder scales at this time, refer to Subsection 3.3.2, and set it at next time. If you want to setup it at this time, follow the steps here: 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 Fig. 3.3.1(b) or Fig. 3.3.1(c).
- For counter clockwise conveyor motion, the z-axis of the Nominal Tracking frame must point up. Refer to Fig. 3.3.1(b).
- For clockwise conveyor motion, the z-axis must point down. Refer to Fig. 3.3.1(c).
- 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.
- Circular tracking boundary values should not be used.

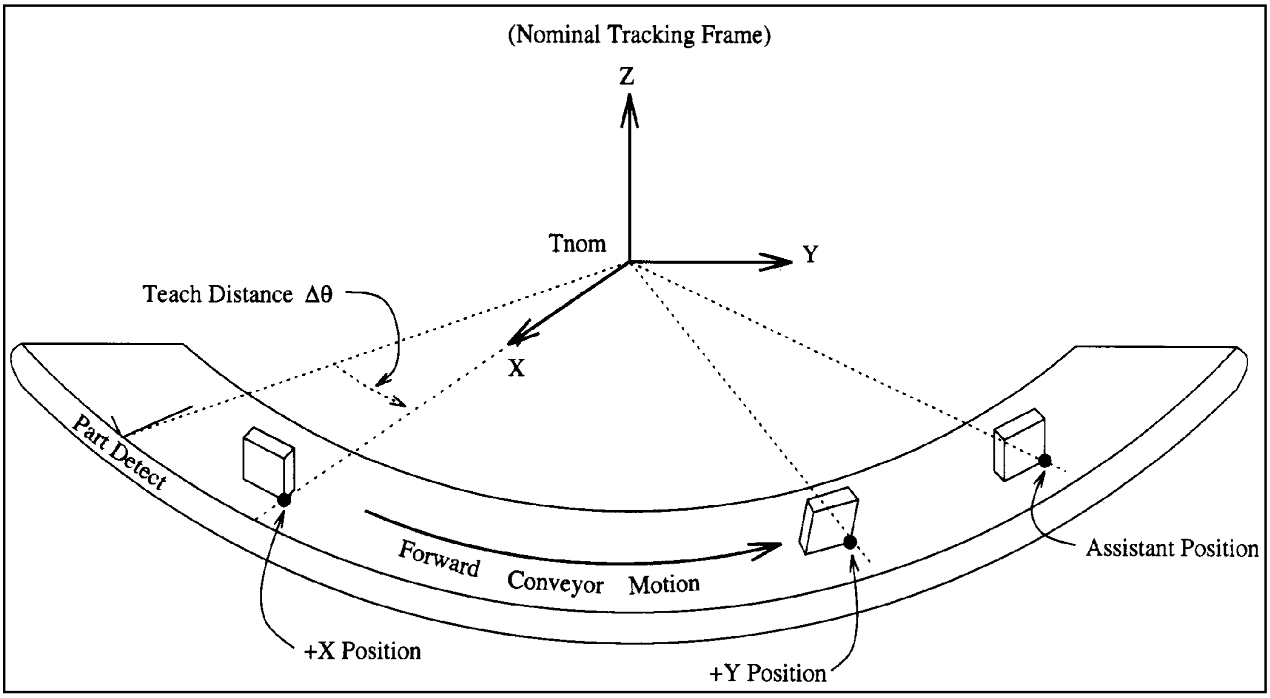


Fig. 3.3.1(b) Nominal tracking frame - counter clockwise circular tracking

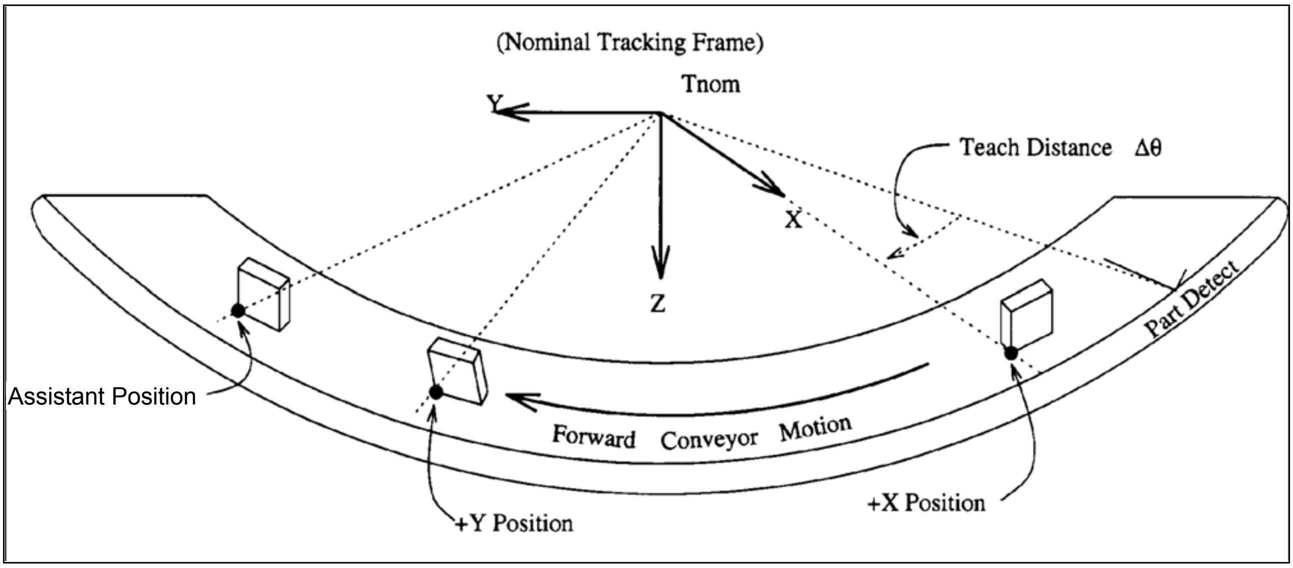


Fig. 3.3.1(c) Nominal tracking frame - clockwise circular tracking

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 Fig. 3.3.1(b) or Fig. 3.3.1(c) for an illustration showing how to teach the nominal tracking frame for circular tracking.

```
SETUP Frames
 Track Frame Setup (Cir)                    1/5
 Track Schedule Number: 1
   Frame Components:
       X:   0.00    Y:   0.00   Z:   0.00
       W:   0.00    P:   0.00   R:   0.00
Teach   Data:
 +X   dir:        RECORDED    Enc_cnt:      2356
 X:  2241.80     Y:     754.09     Z:       30.00
 +Y   dir:        RECORDED    Enc_cnt:      2356
 X:  2241.80     Y:     954.09     Z:       30.00
 Assist:          UNINIT
 X:       0.00     Y:       0.00       Z:       0.00
Scale   (cnt/deg): 8.73

           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 (Cir)                    1/5
 Track Schedule Number: 1
   Frame Components:
  X:   1541.80      Y:      564.09      Z:          30.00
  W:        0.00    P:         0.00     R:           0.00
Teach   Data:
 +X   dir:          RECORDED    Enc_cnt:        2356
 X:   2241.80      Y:      754.09      Z:        30.00
 +Y   dir:          RECORDED    Enc_cnt:        2356
 X:   2241.80      Y:      954.09      Z:        30.00
 Assist:           RECORDED
 X:   2341.80      Y:     1035.09     Z:        30.00
Scale   (cnt/deg): 8.73

          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. 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 "Setting a Tool Coordinate System" in R-30*i*B CONTROLLER OPERATOR'S MANUAL (Basic Operation) B-83284EN 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 or the robot so as not to interfere with each other, when the conveyor or the robot is moved. 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 has 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 at Procedure 3-4, Step 19.

3.3.3 Verify Tracking Setup is Correct

Fig. 3.3.3 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.

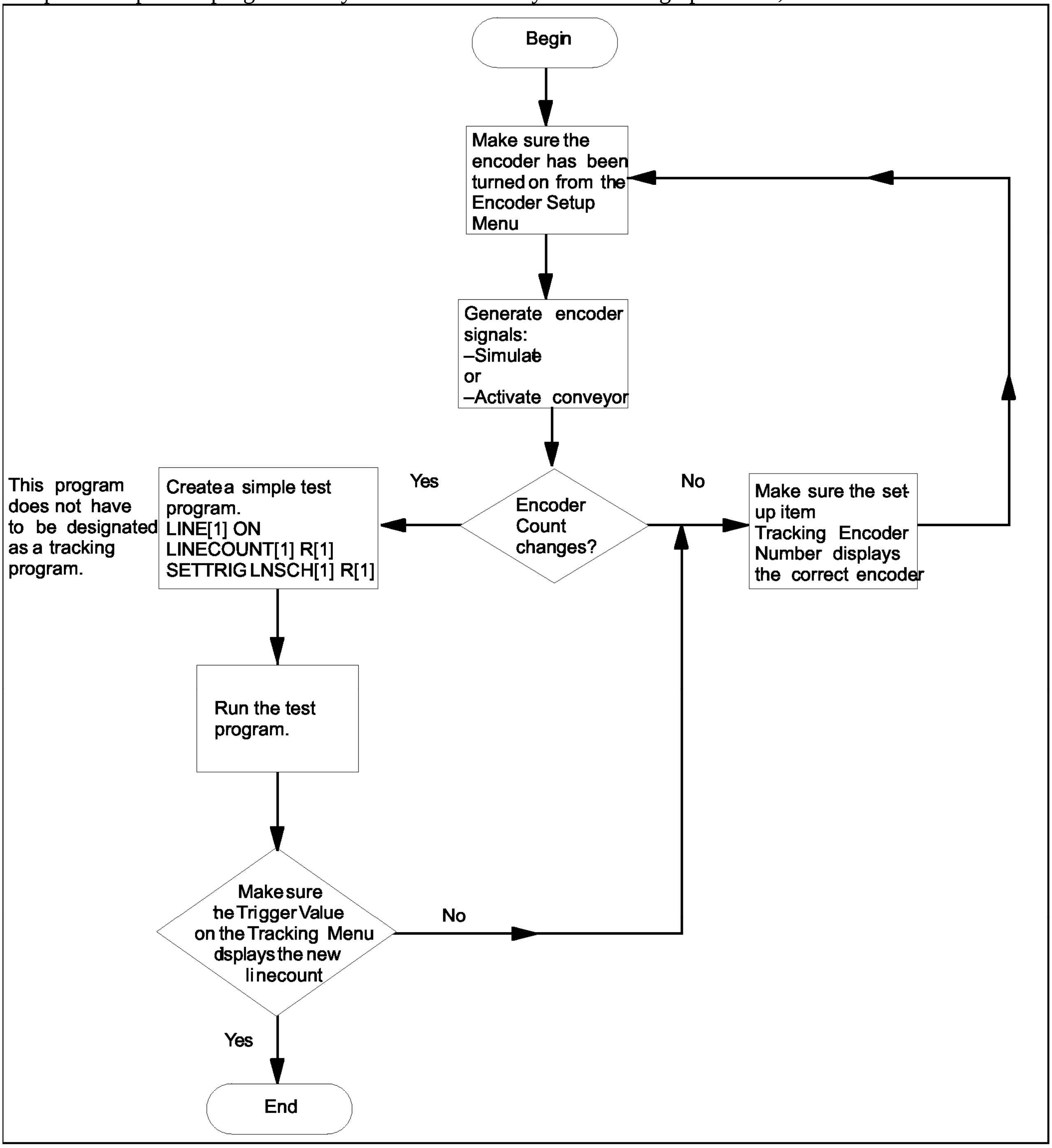


Fig. 3.3.3 Verify tracking setup

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 (Subsection 3.1.2) and Tracking Setup (Subsection 3.3.3).

> **NOTE**
> For more detailed information on planning, writing, or modifying a program, refer to Chapter 4.

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 Example3.4.1:

- Header data: Schedule 0 (Non-tracking)
- DI [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.

Example3.4.1. 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 Example3.4.2:

- Header data: Schedule 1, Cont. Track = FALSE, SELBND = 0.
- P [2] is a safe approach point, typically above the part (see Fig. 3.4.2).
- P [3] is a known location on the part (see Fig. 3.4.2).

> ⚠**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.

Example3.4.2. 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
```

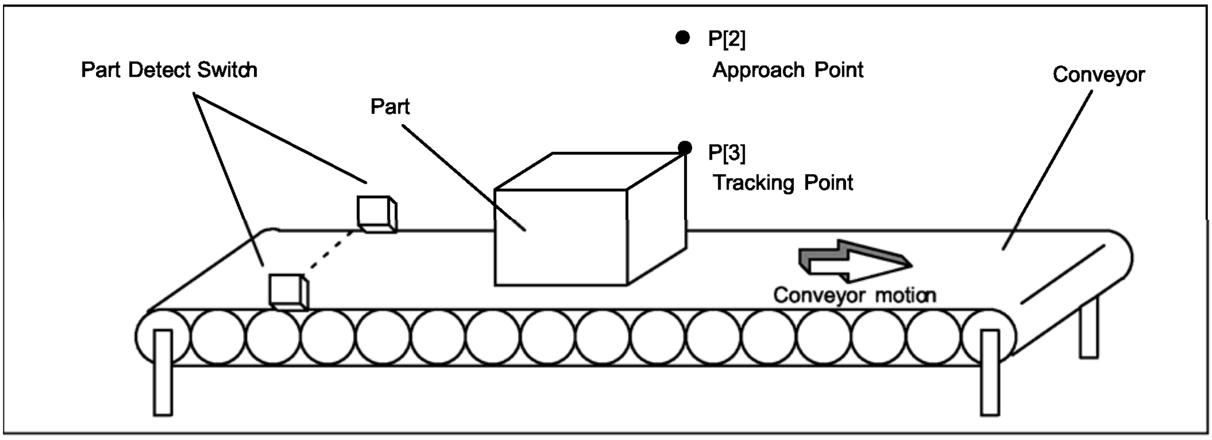


Fig. 3.4.2 Approach and tracking points

3.4.3 Verify Sample Program Execution is Correct

When the program MAIN (Example 3.4.1) and TRACK (Example3.4.2) 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 Subsection 3.3.1)
2. Ensure the encoder scale factor for schedule 1 has been taught correctly. (See Subsection 3.3.2)
3. Ensure the part detect distance has been measured and set correctly. (See Fig. 3.3(a))
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 Subsection 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 Fig. 3.3(b))
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
```

4 PLANNING AND CREATING A PROGRAM

4.1 OVERVIEW

TP program includes a series of commands, called instructions, which 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, Subsection 4.3.2
- Re-synchronize the conveyor, Subsection 4.3.5
- Set up and use predefined positions in a program, Subsection 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 detects.

   > **NOTE**
   > Step 1.d 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 lines 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 Subsection 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.

> ⚠**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 Example 4.2.2 (a).
- The second is a process which moves the robot to a non-tracking rest position. Refer to Example 4.2.2 (b).
- The third is a process which consists of a number of line tracking motions. Refer to Example 4.2.2 (c).

Main (Job) Program Example

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

Example 4.2.2 (a) Main (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

Example4.2.2 (b) and Example4.2.2 (c) provide process program examples.
Example4.2.2 (b) moves the robot to a rest or home position which does NOT track the conveyor.

Example 4.2.2 (b) Move to Rest Position

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

Example 4.2.2 (c) moves the robot to a number of locations while the robot also tracks the moving conveyor.

Example 4.2.2 (c) 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

Fig. 4.3 summarizes writing and modifying a program.

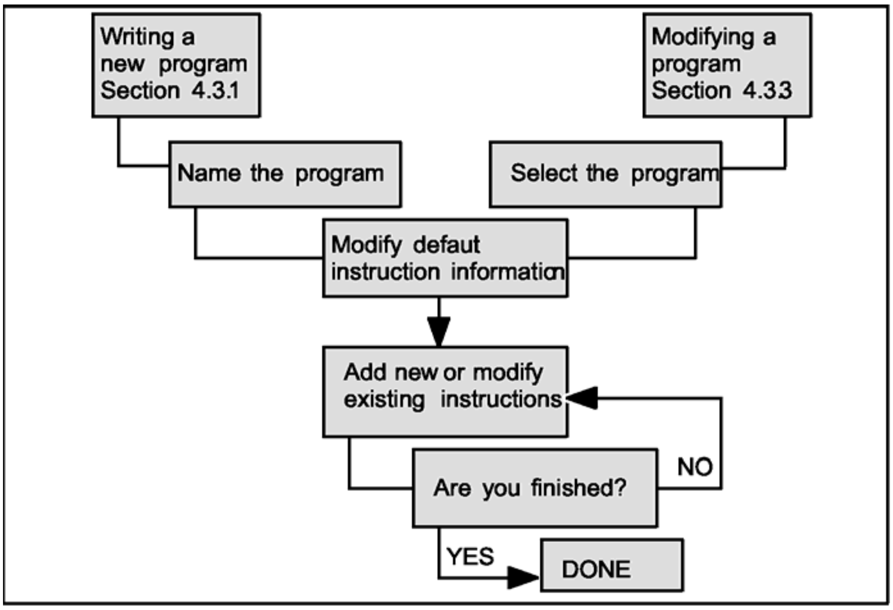


Fig. 4.3 Writing and modifying a program

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 allow 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 less than thirty-six 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 only under line _ symbol and do not use at mark @ symbol and 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

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 “MOTION INSTRUCTION” in R-30*i*B CONTROLLER OPERATOR’S MANUAL (Basic Operation) B-83284EN 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 3.a. 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 5.a. 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 Subsection 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 Subsection 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 17.d through Step 17.f 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 “SAVING FILES" in R-30*i*B CONTROLLER OPERATOR’S MANUAL (Basic Operation) B-83284EN.
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.

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 Subsection 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 “Selecting a Program” in R-30*i*B CONTROLLER OPERATOR’S MANUAL (Basic Operation) B-83284EN 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 display all programs.

- Jobs display all job programs.
- Processes display all process programs.
- TP Programs display all teach pendant programs.
- KAREL Progs display 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 Subsection 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.
      - 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 7.a through Step 7.c 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 8.a and Step 8.b 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 11.a through Step 11.j 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.

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 “BACKGROUND EDITING” in R-30*i*B CONTROLLER OPERATOR’S MANUAL (Basic Operation) B-83284EN.

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 MENUS.
   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 1.f.
   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?

       YES          NO
```

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.
   - 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 - Add 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 Subsection 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.

```
        DEFENC[x]  SNRSCH[x]

Direct:                               Direct:
Encoder number (1-8)                  Schedule number (1-8)
Indirect:R[x]                         Indirect:R[x]
Encoder number = contents of R[x]     Schedule number = contents of R[x]
```

Fig. 4.5(a) DEFENC

GONE_TIME

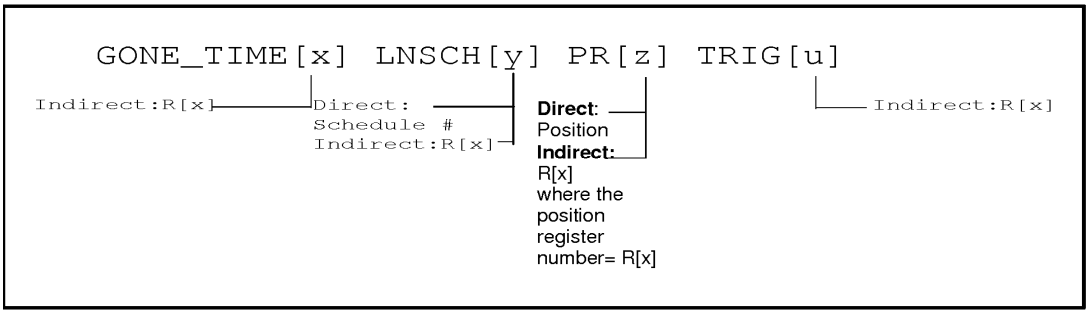


Fig. 4.5(b) GONE_TIME

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,

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

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.

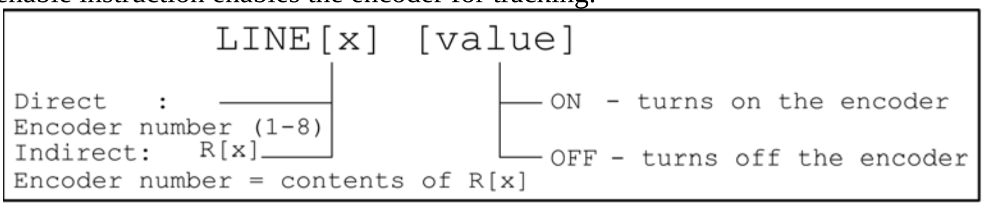


Fig. 4.5(c) LINE

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.

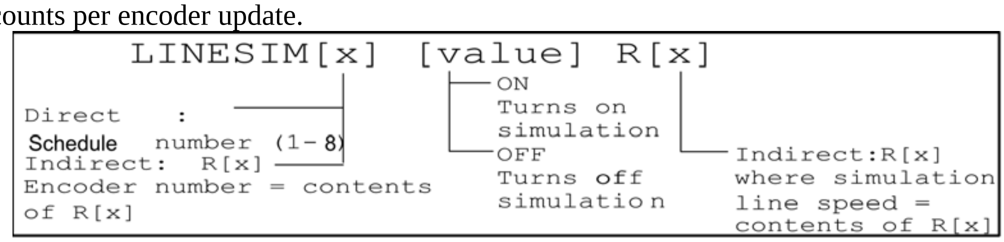


Fig. 4.5(d) LINESIM

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.

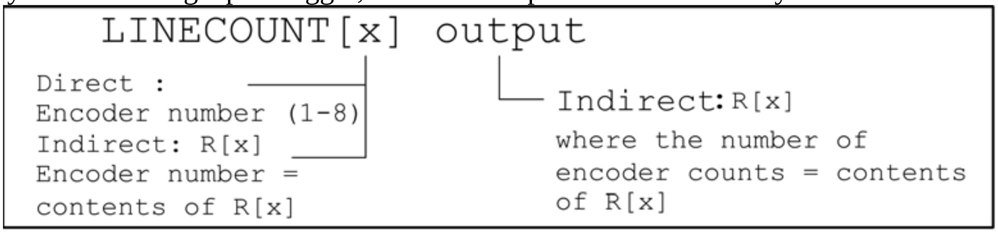


Fig. 4.5(e) LINECOUNT

LINERATE

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

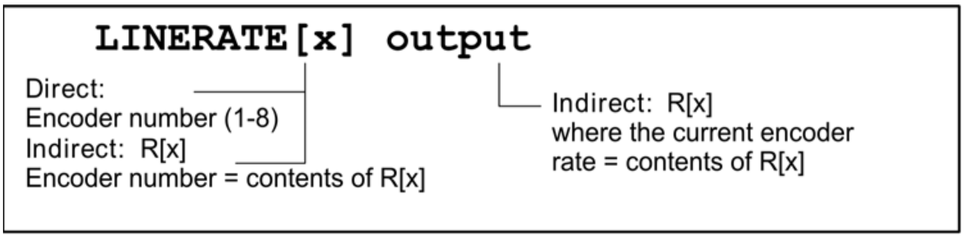


Fig. 4.5(f) LINERATE

LINESTOP

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

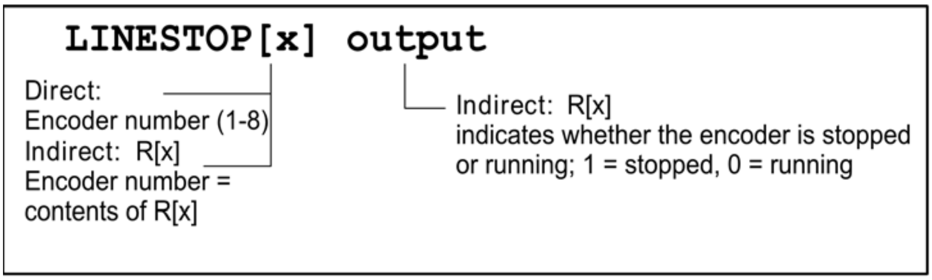


Fig. 4.5(g) LINESTOP

SETTRIG

The SETTRIG instruction sets the tracking schedule trigger value. Refer to the “Line Tracking Setup” chapter in the FANUC Robotics Line Tracking 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.

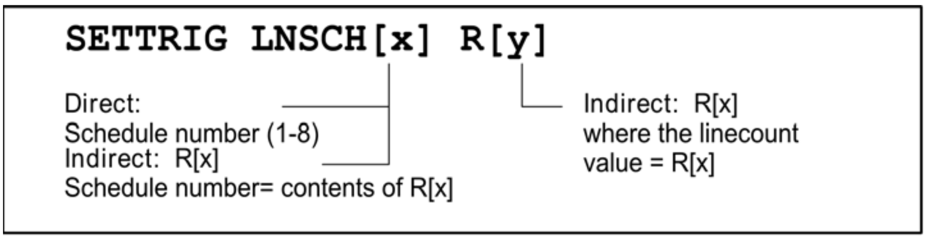


Fig. 4.5(h) SETTRIG

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 FANUC Robotics Line Tracking Setup and Operations Manual for more information on schedules.

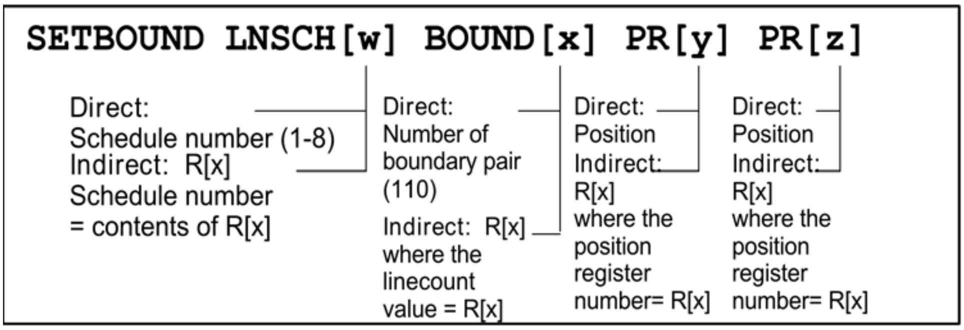


Fig. 4.5(i) SETBOUND

SELBOUND

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

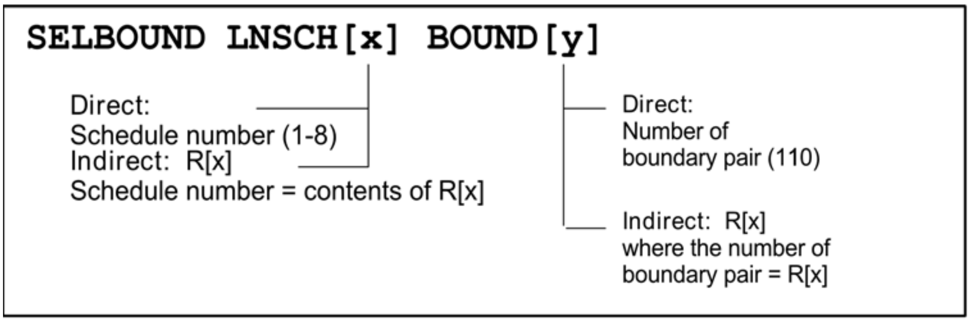


Fig. 4.5(j) SELBOUND

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.

```
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.

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.

> **NOTE**
> The STOP_TRACKING feature is not available for use with Circular tracking. When the STOP_TRACKING instruction is specified in a Circular tracking program, the instruction will do nothing as if the instruction is not there.

For example:

```
1:L P[1] 1000mm/sec CNT100
2:WAIT DI[10] = ON
3: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:

```
1:L P[1] 1000mm/sec CNT100
2:STOP_TRACKING
3:WAIT DI[10] = ON
4: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. 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,

```
1:ACCUTRIG LNSCH[1]
2:WAIT DI[1]=OFF
3:WAIT DI[1]=ON
4:LINECOUNT[1] R[1]
5: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.

ACCUTRIG LNSCH[x]

Direct:
Schedule number (1-8)
Indirect: R[x]
Schedule number= contents of R[x]

Fig. 4.5(k) ACCUTRIG LNSCH

TRKUFRAME

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

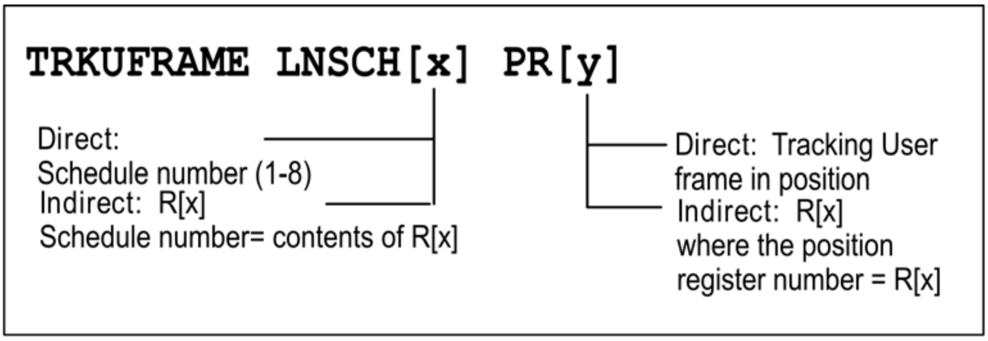


Fig. 4.5(l) TRKUFRAME

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.

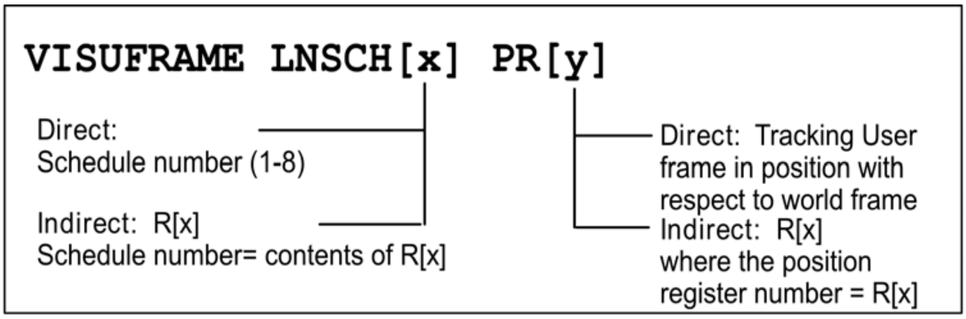


Fig. 4.5(m) VISUFRAME

5 ADVANCED TECHNIQUES

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 Fig. 5.1.

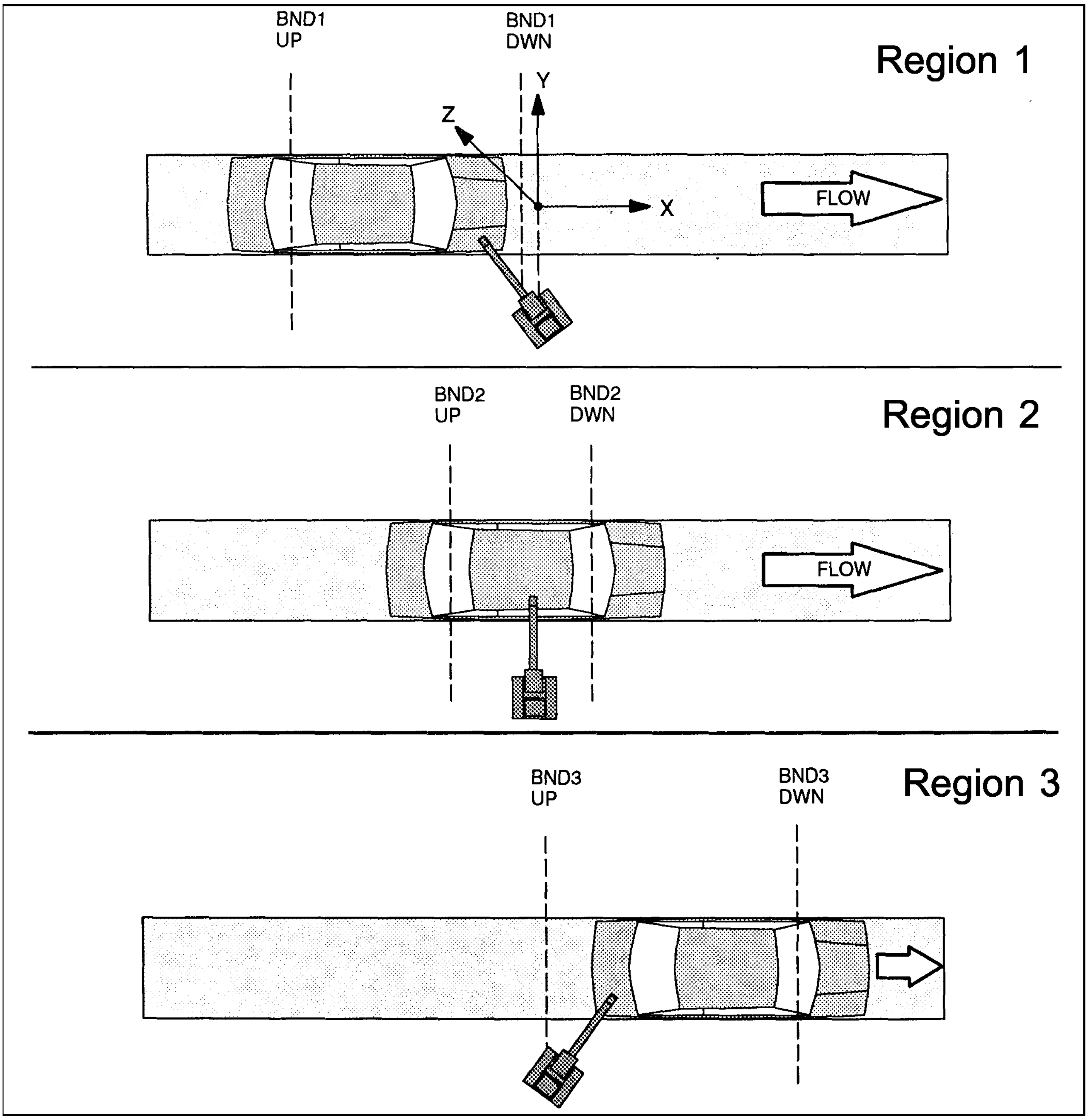


Fig. 5.1 Tracking boundary positions

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

- The front hood
- The top

- The back deck

In Fig. 5.1, the regions labeled 1, 2, and 3, are the boundaries of three work zones. Here UP is the up-stream boundary and DWN is the down-stream boundary where up and down stream refers 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 Example 5.1 is an example of a line track job that calls the programs to paint the regions of the car shown in Fig. 5.1.

Example 5.1. Job that calls programs to paint a car

```
/PROG  PAINT CAR
/MN
  1: LINE[1] ON ;
  2: LBL[1] ;
  3: CALL HOME ;                          -- call rest pos. (non-tracking)
  4: WAIT DI[32]=ON ;                     -- wait for part detect
  5: LINECOUNT[1] R[1] ;
  6: SETTRIG LNSCH[1] R[1] ;
  7: SELBOUND LNSCH[1] BOUND[1] ;         -- select zone 1
  8: CALL CAR_FRNT ;                      -- paint front (tracking)
  9: SELBOUND LNSCH[1] BOUND[2] ;         -- select zone 2
 10: CALL CAR_TOP ;                       -- paint top (tracking)
 11: SELBOUND LNSCH[1] BOUND[3] ;         -- select zone 3
 12: CALL CAR_BACK ;                      -- paint back (tracking)
 13: 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 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 returns to monitor the part detect switch and wait for the next part to arrive. The time sequence is shown below in Table 5.2.1.

Table 5.2.1 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, Main Program (Process Task) informs Monitor Program (the part detect monitoring task) when it is ready to receive the next part trigger value. Monitor Program then informs the robot when the new part trigger value is available.

Example 5.2.2 (b) shows an example of Main Program which initializes the system, runs Monitor Program, 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, Monitor Program 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.

Example5.2.2 (a) shows an example of Monitor Program run from Main Program of Example 5.2.2 (b).

In this case, this program allows up to 10 parts to be queued, by storing each part detect signal received in a register. If the maximum number of parts allowed between the parts detect switch and the robot OUT BOUND window at any given time exceeds the size of the part queue (the number of program registers needed to store consecutive part trigger values), increase the number of registers where each part detect signal is stored.

> **NOTE**
> The group mask of Monitor Program is set=0, since it will never control robot motion, but must run concurrently with robot motion tasks.

Example 5.2.2 (a) Monitor Program (SUBPRG1)

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

Example 5.2.2 (b) Main Program (Process Task)

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

5.3 MULTIPLE CONVEYORS (DUAL LINE TRACKING)

Line tracking supports up to eight 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.

- Each 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.
- When tracking motion of different schedule is executed continuously, execute STOP_TRACKING or normal motion once before next tracking motion is started.

5.3.1 Dual Line Tracking Setup

A typical Dual Line Tracking application includes:

- Monitoring signals from both parts detect switches, MONITOR1, MONITOR2.
- Looking for parts simultaneously on both conveyors.
- Verifying the robot tracks positions on each conveyor correctly, CONV_1, CONV_2.

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 Monitor Program by itself and check if the pulse counts are displayed in appropriate registers as parts pass by the trigger sensor.
- Run 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.
This 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. Therefore, take care of the cases as below.

- When the number of works becomes bigger,
  Because this program does not alternate processing between conveyors, it is possible that it does not finish processing of works on the conveyor detected now, and so it does not process the other conveyor. In this case, decrease the rate of works of the conveyor.

- When the speed of works becomes faster,
  Because works move quickly, it is possible that the tracking path is severe for the robot, and so the limit error of the servo occurs. In this case, adjust the position relation between the robot and the conveyor, or decrease the speed of works.

Monitor Program

This program continuously monitors the trigger signals of both conveyors. This program allows up to 10 parts to be queued, by storing each part detect signal received in a register. When a part detect signal from a conveyor is received, Monitor Program reads the pulse counts of the signal and stores it in one of ten 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. Process Program (refer to Example 5.3.2 (c)) 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 `*'.

Example 5.3.2 (a) Monitor Program (MONITOR 1)

```
/MN
 1: LBL[20] ;
 2: R[1]=10 ;                             -- REGS[10]-[19] used for encoder 1
 3: LBL[30] ;                             -- MAIN LOOP
 4: R[R[1]]=0 ;
 5: WAIT DI[1]=ON+ ;                      -- WAIT FOR PART DETECT
 6: LINECOUNT[1] R[3] ;
 7: R[R[1]]=R[3] ;
 8: R[3]=0 ;
 9: R[1]=R[1]+1 ;                         -- INC INDEX REG
10: IF R[1]>19, JMP LBL[20] ;
11: JMP LBL[30] ;                         -- GO TO MAIN LOOP
/POS
/END
```

Example 5.3.2 (b) Monitor Program (MONITOR 2)

```
/MN
 1: LBL[20] ;
 2: R[2]=20 ;                             -- REGS[20]-[29] used for encoder 2
 3: LBL[30] ;                             -- MAIN LOOP
 4: R[R[2]]=0 ;
 5: WAIT DI[2]=ON+ ;                      -- WAIT FOR PART DETECT
 6: LINECOUNT[2] R[5] ;
 7: R[R[2]]=R[5] ;
 8: R[5]=0 ;
 9: R[2]=R[2]+1 ;                         -- INC INDEX REG
10: IF R[2]>29, JMP LBL[20] ;
11: JMP LBL[30] ;                         -- GO TO MAIN LOOP
/POS
/END
```

Process Program

The program in Example 5.3.2 (c) 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.

Example 5.3.2 (c) Process Program (PART)

```
 1: R[4]=10;          -- Register for comparing with detect register of Montor1
 2: R[6]=20;          -- Register for comparing with detect register of Montor2
 3: LINE[1] ON;
 4: LINE[2] ON;
 5: CALL_PROG HOME;                  --HOME is a non-tracking program
 6: LBL[10];
 7: WAIT R[1]<>R[4] TIMEOUT, LBL[20];--TIMEOUT = $WAITTMOUT = 10 (which is 100 ms)
```

```
 8: JMP LBL[100];
 9: LBL[20];
10: WAIT R[2]<>R[6] TIMEOUT, LBL[10];
11: JMP LBL[200];
12: LBL[100];
13: SETTRIG LNSCH[1] R[R[4]];
14: SELBOUND LNSCH[1] BOUND[1];
15: CALL_PROG CONV_1;              --CONV_1 is tracking program
16: CALL_PROG CLEAR_1;             --CLEAR_1 is not for tracking
17: R[4]=R[4]+1;
18: IF R[4]<=19, JMP LBL[10];
19: R[4]=10;
20: JMP LBL[10];
21: LBL[200];
22: SETTRIG LNSCH[2] R[R[6]];
23: SELBOUND LNSCH[2] BOUND[1];
24: CALL_PROG CONV_2;              --CONV_2 is tracking program
25: CALL_PROG CLEAR_2;             --CLEAR_2 is not for tracking
26: R[6]=R[6]+1;
27: IF R[6]<=29, JMP LBL[20];
28: R[6]=20;
29: JMP LBL[20];
/POS
/END
```

Verify Robot Tracking Program

The program in Example 5.3.2 (d) 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 conveyors.

> **NOTE**
> Tracking must be enabled for this program.

Example 5.3.2 (d) CONV_1 and CONV_2 Verify Robot Tracking Programs

```
1: L P[1:NEAR PART] CNT75;
2: L P[2:TARGET POS] FINE;
3: WAIT 2.00(sec);                   --delay is to test tracking
4: L P[1:NEAR PART] CNT100;
/POS
/END
```

Switch Conveyor Part Processing

The program in Example 5.3.2 (d) 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.

Example 5.3.2 (e) 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
```

Line 17 (CALL_PROG CLEAR_1) and line 23 (CALL_PROG CLEAR_2) of example 5.3.2(b) Part Processing Program can be exchanged for STOP_TRACKING.

```
STOP_TRACKING;
```

Main Program

The program in Example 5.3.2 (f) runs Monitor Program (Example 5.3.2 (a), (b)) concurrently with Process Program (Example 5.3.2 (c)).

Example 5.3.2 (f) Main Program Runs Monitor and Process Programs

```
1: RUN_PROG MONITOR1
2: RUN_PROG MONITOR2
3: 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 Fig. 5.4.1 .

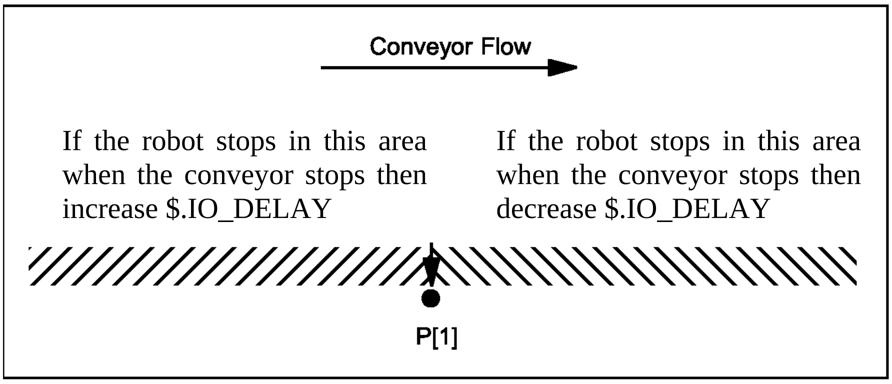


Fig. 5.4.1 Modifying $LNCFG_GRP.$IO_DELAY

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

$$\frac{3}{500} \cong 6(\text{ms})$$

   Therefore, adjust $IO_DELAY by 6 ms.

5. 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.

NOTE

Adjust static tune variable by $IO_DELAY in case of line tracking, then adjust static tune variable by reference points in case of visual tracking. Refer to 4.6.1 in *i*RVision visual tracking manual B-83304EN-4 for details.

You can adjust static tune variable by $ENC_IODELAY[encoder number] instead of $LNCFG_GRP[g].$IO_DELAY. Before adjusting, make the following settings.
- Set $ENC_IOD_ENB[encoder number] to TRUE.
- Set $ENC_IODELAY[encoder number] to 0.

Adjust it referring to 5.4.1 Static Tune Variable.

5.4.2 Dynamic Tune Variable

To modify the dynamic tune variables, you must first set up the static tune variable (Subsection 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 Fig. 5.4.2 .

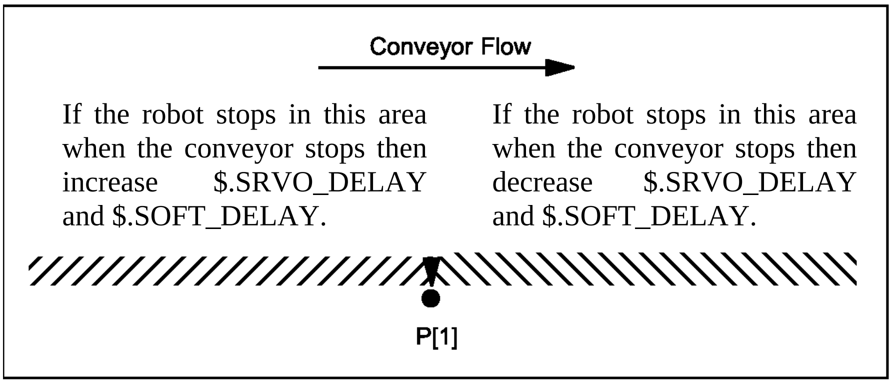


Fig. 5.4.2 Modifying $LNCFG_GRP.$SRVO_DELAY and $LNCFG_GRP.$SOFT_ DELAY

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.

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 Fig. 5.5.2 for a typical robot-conveyor setup and frame relationship.

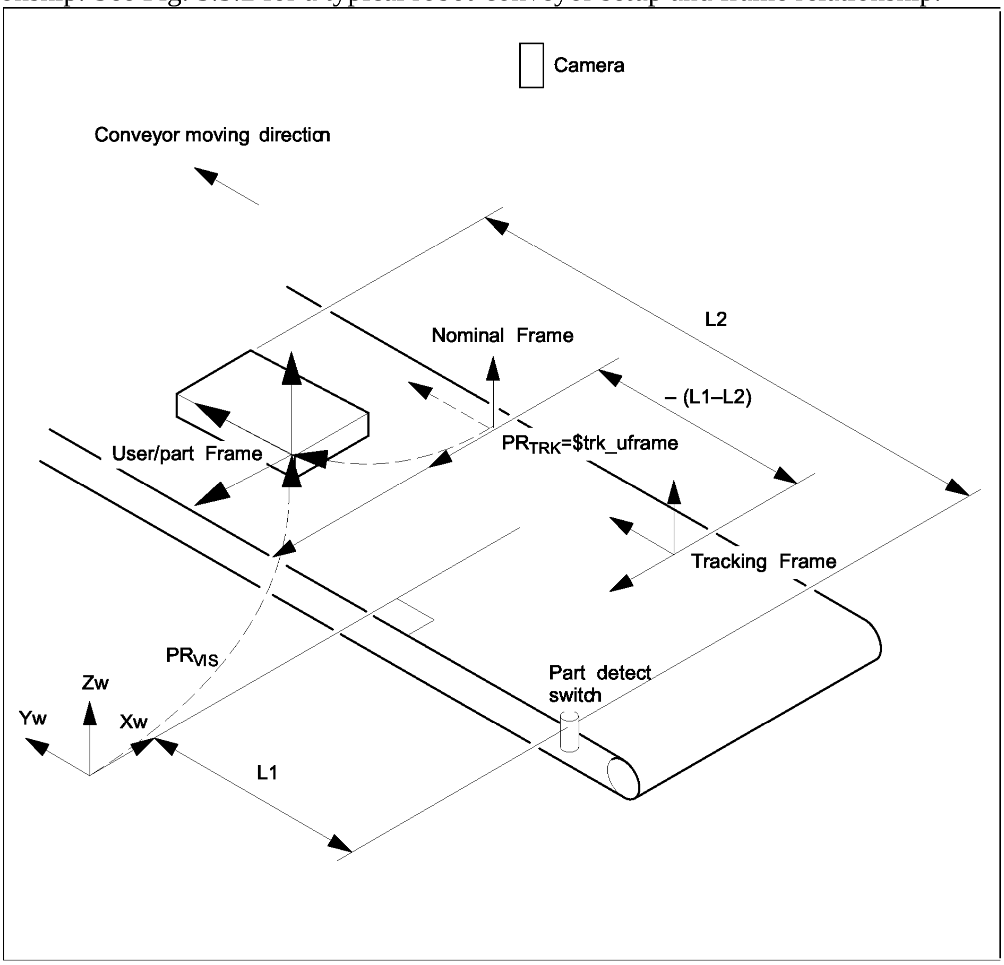


Fig. 5.5.2 A typical tracking frame setup and frame relationship

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 Position Resister 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 Fig. 5.5.2), 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 Position Resister for TRKUFRAME

Steps

1. Pass the object (part) through the part detects 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 Subsection 5.5.2.
4. Calculate L_1 - L_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 Position Resister 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 Fig. 5.5.2. 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 Example 5.5.5 (a).

Example 5.5.5 (a) Sample Tracking Uframe Program

```
1:Line [1] ON,                    --turn on encoder2:LBL[1];
2:CALL HOME1;                     --home the robot
3:WAIT DI[27]=ON;                 --wait for part detect switch is triggered
4:LINECOUNT[1] R[1];              --put the encoder count into the register
5:SETTRIG LNSCH[1] R[1];          --set the trigger count
6:SELBOUND LNSCH[1] COUNT[1]      --select a boundary set
7:TRKUFRAME LNSCH[1] PR[1];       --set the tracking uframe
8:CALL LNTK1;                     --call a tracking program
9: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 Example 5.5.5 (b).

Example 5.5.5 (b) 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 (Subsection 5.5.3 and Subsection 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, and 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 "PROGRAMMING" and "EXECUTING A PROGRAM" chapter in R-30*i*B CONTROLLER OPERATOR'S MANUAL

(Basic Operation) B-83284EN. 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 runs the program again to see the path compensation.
Example path compensations are illustrated in Fig. 5.5.6(a) and Fig. 5.5.6(b). 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.

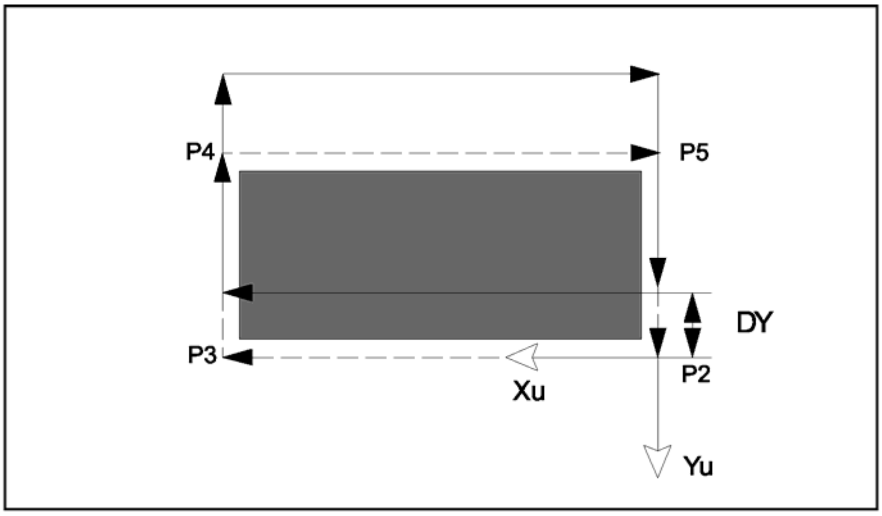


Fig. 5.5.6(a) Path compensation in the y direction

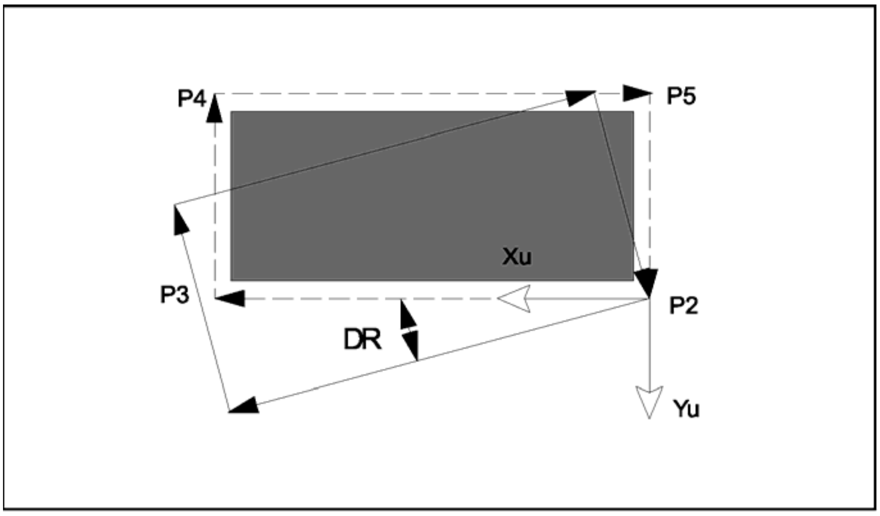


Fig. 5.5.6(b) Path compensation in the r rotation

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 Fig. 5.5.7.
Use the same procedure described in Subsection 5.5.3. PR[x] can be determined for TRKUFRAME and VISUFRAME. See Fig. 5.5.7. The tracking uframe can be used to compensate for the tracking path in rail tracking in the same way as in line tracking.

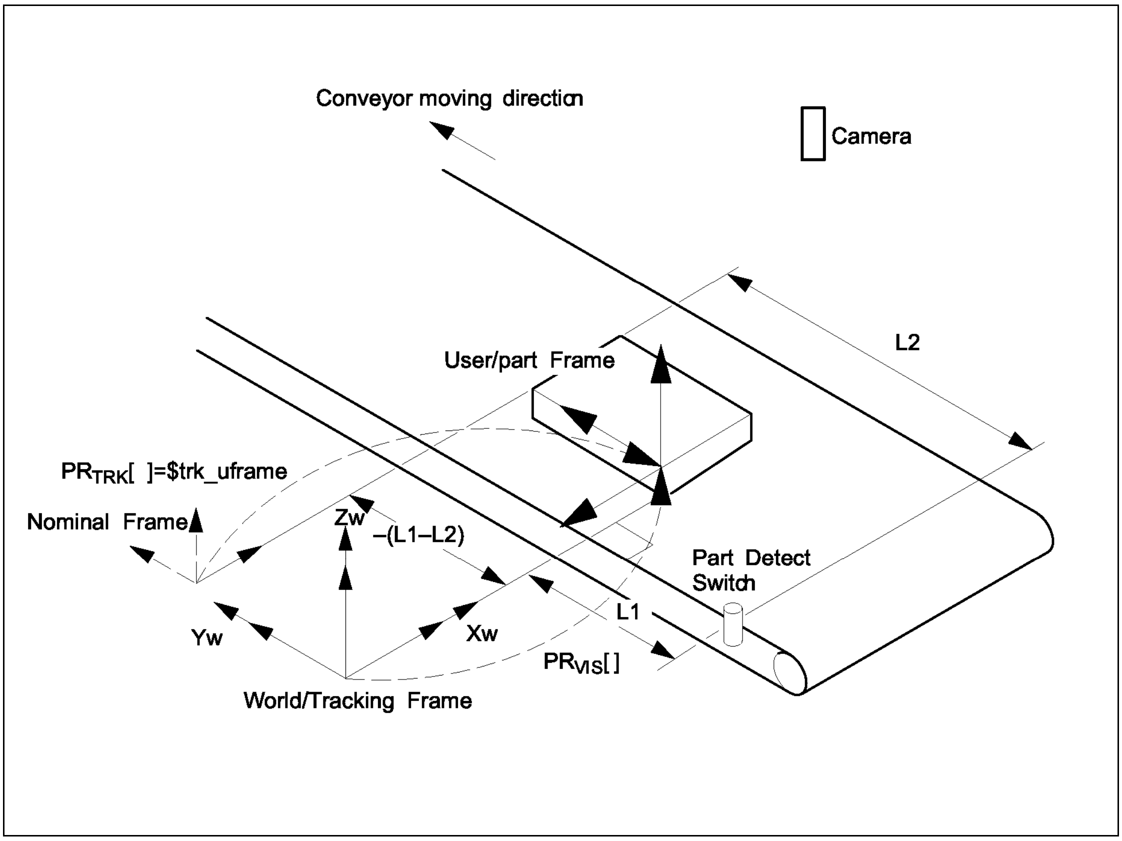


Fig. 5.5.7 Rail tracking setup and frame relationships

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 JRL8 connector of the controller, in place of the standard digital input normally used for part detection.
To use High Speed Scanning, you must

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

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 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#1. (Refer to Appendix B)
- Encoder #1 is set up as the tracking encoder.

> **NOTE**
> HSDI for Line Tracking is supported in R-30*i*B Controller, but is not supported in R-30*i*B Mate Controller.

Steps

1. Press MENUS.
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 input used HDI port id to HDI port id.
9. Turn off the controller, and then turn it on again to accept the new setting.

```
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 HDI port Id:                            1
[TYPE]                 ENCODER
```

> **NOTE**
> When HSDI (J831) is ordered and tracking sub program is taught by using normal DI input, set $LNCFG.$HSDI_ENABLE to FALSE. Otherwise, it is possible that tracking program can be not edited.

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. Example 5.6.3 (a) shows a standard line tracking program that does not use the high speed scanning feature.

Example 5.6.3 (a) Main program (job) without high speed scanning instructions

```
1: J P[1] 50% FINE                        -- MOVE TO HOME
2: LINE[1] ON                             -- ENABLE THE ENCODER
3: WAIT DI[1] ON                          -- WAIT FOR PART DETECT
4: LINECOUNT[1] R[1]                      -- GET TRIGGER VALUE
5: SETTRIG LNSCH[1] R[1]                  -- SET TRIGGER VALUE
6: SELBOUND LNSCH[1] BOUND[1]             -- SELECT A BOUNDARY
7: CALL TRACK                             -- CALL TRACKING PROGRAM
8: J P[1] 50% FINE                        -- MOVE TO HOME
```

Example 5.6.3 (b) 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.

Example 5.6.3 (b) Main Program (Job) with High Speed Scanning Instructions

```
 1: J P[1] 50% FINE                      -- MOVE TO HOME
 2: LINE[1] ON                           -- ENABLE THE ENCODER
 3: ! -----------------------------
 4: $HDI_FLAG[1] = 1                     -- ENABLE THE HSDI port 1
 5: WAIT $ENC_STAT[1].$ENC_HSDI = 1      -- WAIT FOR PART DETECT
 6: ! -----------------------------
 7: LINECOUNT[1] R[1]                    -- GET TRIGGER VALUE
 8: SETTRIG LNSCH[1] R[1]                -- SET TRIGGER VALUE
 9: SELBOUND LNSCH[1] BOUND[1]           -- SELECT A BOUNDARY
10: CALL TRACK                           -- CALL TRACKING PROGRAM
11: J P[1] 50% FINE                      -- MOVE TO HOME
```

Differences between the Example Programs

The WAIT statement in Example 5.6.3 (b), 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 Example 5.6.3 (b) . Setting this variable to 1 activates the HSDI 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**
> When you are editing tracking programs, conveyor resynchronization automatically uses the HSDI #1 hardware input for part detection. Therefore, you do not need to modify $HDI_FLAG[port].

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 Subsection 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.
- You can adjust static tune variable by $ENC_IODELAY[encoder number] instead of $LNCFG_GRP[g].$IO_DELAY. Before adjusting, make the following settings.
  - Set $ENC_IOD_ENB[encoder number] to TRUE.
  - Set $ENC_IODELAY[encoder number] to 0.

  Adjust it referring to 5.4.1 Static Tune Variable.

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.

NOTE

Limit checking is worked in case of following tracking motion.

- Motion from reference point of tracking motion program to reference point of tracking motion program
- Motion from reference point of normal motion program to reference point of tracking motion program

Limit checking is not worked in case of following normal motion.

- Motion from reference point of tracking motion program to reference point of normal motion program

However, limit checking is avoided when the additional order 'Wjnt' is added on first motion order after tracking motion or when turn on bit18 (262144) of $LNCFG.$COMP_SW.

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. When there are multiple robots working on the same conveyor, each robot needs to know the part's location on the conveyor. The Pulse 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 Pulse Multiplexer to supply the encoder information to each robot on the same conveyor. The Ethernet Encoder 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 Ethernet Network connection.

The Ethernet Encoder provides the following benefits:

- No Pulse Multiplexer is needed.
- No Encoder Cable required for Slave controller.
- No Line tracking interface board required for slave controller.
- Much easier to set up.

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

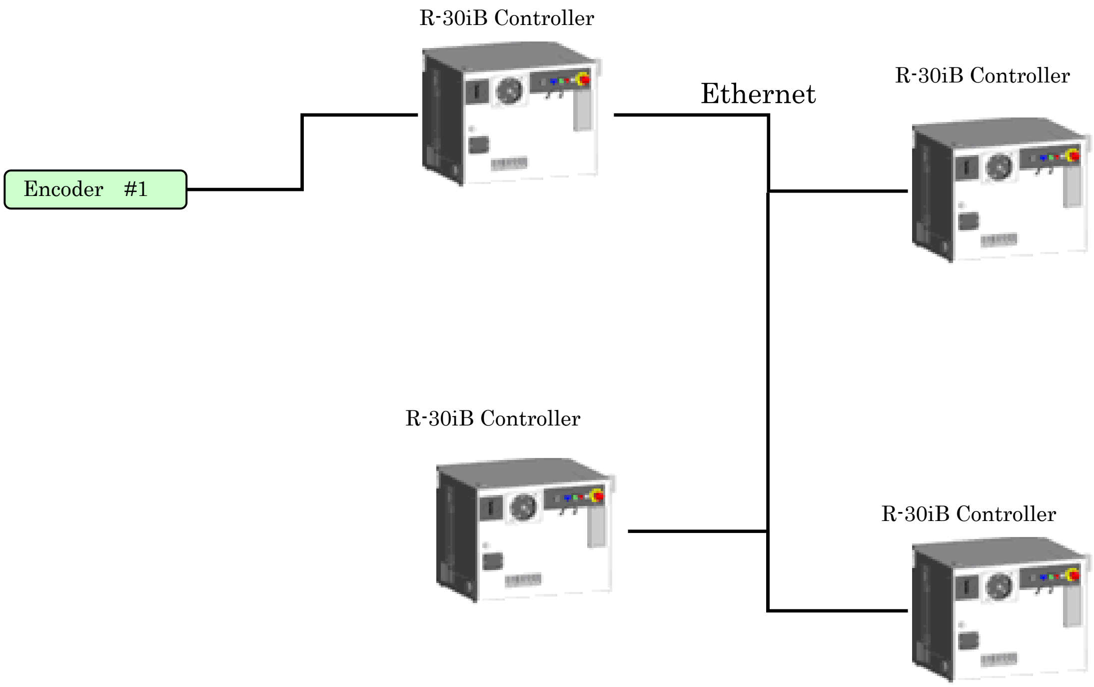


Fig. 5.9.1 Hardware setup with the ethernet encoder option

5.9.2 Explanation of Terms

There are terms as “Master” and “Slave” in using Ethernet Encoder function. These are “Master Controller”, “Slave Controller”, “Master Encoder”, and “Slave Encoder”. The following are described so that they are not mistaken easily.

Master Controller and Slave Controller

Robot ring internally uses the communication function, “ROS interface packet over Ethernet (RIPE)”. The master controller is the master of RIPE. RIPE consists of one master controller and the other slave controllers.

Master Encoder and Slave Encoder

Master encoders are encoders connected with the controller. The information of these Encoders is transmitted to the other controllers over the Ethernet. The controller receiving information can refer to information of master encoders as it is connected with them. The encoder referred to over the Ethernet option is called as the slave encoder.

When the Ethernet Encoder option is used, the master encoder needs not to be connected with the master controller. However, it is recommended that the master encoder is connected directly with the master controller for decreasing the communication traffic.

5.9.3 Limitations

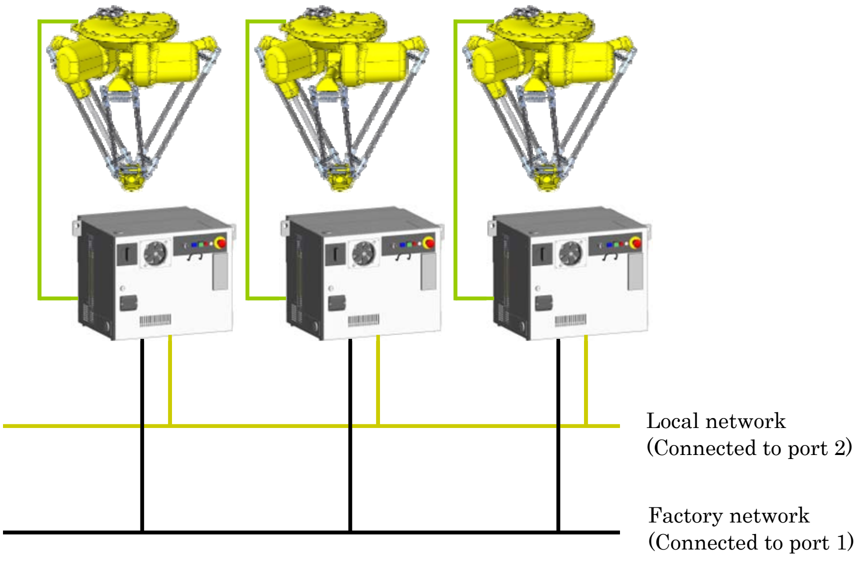
- Each controller can support up to four master encoders.
- The Ethernet Encoder in one Ripe network can Support up to four controllers.
- The $SCR.$ITP_TIME for all the controllers should be set to the same value.
- Encoders using servo conveyor line tracking function should not be used as the Ethernet Encoder.

- The communication causes the difference between the master encoder's pulse count and the slave encoder's pulse count. The difference is converted into time is about plus or minus 2ms as compared with using Pulse Multiplexer.

5.9.4 Create a Network

5.9.4.1 Connecting a network cable to the robot

The Ethernet Encoder uses Ethernet network. This Ethernet network should be an isolated network so that it is not affected by building/plant network.
R-30*i*B Controller has two Ethernet ports. Accordingly, the robot controllers can belong to two different networks. If a robot controller needs to connect to an external network, for example which is a factory network, port 1 is used for the external network and port 2 is used for the robot controller network as shown in the figure below.



When port 1 is used, connect the Ethernet cable to the CD38A connecter on the MAIN board. When port 2 is used, connect the Ethernet cable to the CD38B connecter on the MAIN board.

> **NOTE**
> If a robot controller is not connected to an external network, you can use whichever port you prefer.

5.9.4.2 Caution on create a network

When a network is created, there are following cautions. Conform to them and setup steps below.

- According to “5.9.4.1 Connecting a Network Cable to the robot”, there are two ports in one controller. Set the IP address and the subnet mask of the port used in “5.9.4.3 Setting IP Address”.
- When the IP address is set, each robot controller needs to be named differently.
- When the IP address is set, the IP address of the port used in the communication of robots needs to be continuous number. For example, if there are four robot controllers and port 2 is used, the IP address of port 2 of each robot controller is set as 172.16.0.1, 172.16.0.2, 172.16.0.3, and 172.16.0.4.
- If Ethernet Encoder is used, the IP address of the robot controller connected with the encoder is set as first number. For the previous example, the IP address of port 2 of the robot controller connected with the encoder is set as 172.16.0.1.

- A system with only two controllers: You have a choice to use switch or not. Without a switch, you need to use a crossover Ethernet Cable instead of a regular Ethernet Cable. In this case, plug one end of the Ethernet cable into the Ethernet port of one controller and the other end to the other controller. With a switch, do as method below.
- A system m 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 the Ethernet Cable to the Ethernet Port of the switch.

5.9.4.3 Setting IP address

Set up the IP address for each robot controller by following this procedure:

1. Press [MENUS] on the teach pendant.
2. Select [6 SETUP].
3. Select [Host Comm].
4. Select [TCP/IP], and press [ENTER] button.
5. Input this robot controller name to [Robot name].

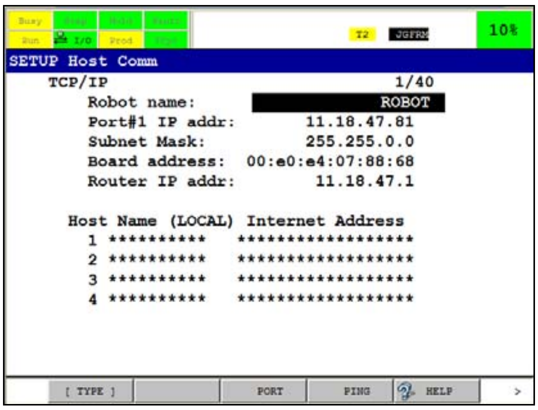
> ⚠**CAUTION**
> Only alphabet, number, and minus sign are used for robot name. Only alphabet is used for first character. Minus sign is not used for last character. You should not insert wasted space.

1. Input the IP address for this robot controller to [Port#1 IP addr].

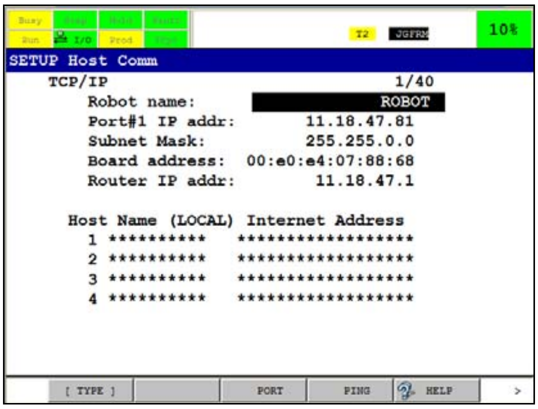
> ⚠**CAUTION**
> You should not insert wasted space or 0 to IP address. If there are wasted spaces or 0 in IP address, the communication becomes wrong.

2. Input the subnet mask to [Subnet Mask].
3. Input the IP address of the router to [Router IP addr].
4. Press F3, [PORT], and select the port.
5. Set IP address and subnet mask by the same way as the procedure from 7 to 8.
6. Cycle power to the robot controller.

> ⚠**CAUTION**
> If you don't cycle power of the robot controller, following setup is not performed correctly. Cycle power of the robot controller necessarily.



Setting example of Port 1



Setting example of Port 2

> ⚠**CAUTION**
> If you use both port 1 and port 2, they need to be set to the different network address. For example, when subnet mask is 255.255.0.0 and IP address is 172.16.0.1, network address is 172.16 and host address is 0.1. If port 1 and port 2 are set to the IP address whose network address is same, the alarm “HOST-179 IP Address mis-configuration” is posted when the controller is turned on. Then, only port 1 becomes valid and port 2 becomes invalid.

5.9.4.4 Setting of the robot ring

Robot ring internally uses the communication function, “ROS interface packet over Ethernet (RIPE)”. In detail of RIPE, refer to “ROS interface packet over Ethernet” in Ethernet Function OPERATOR’S MANUAL B-82974EN.

Master and Slave

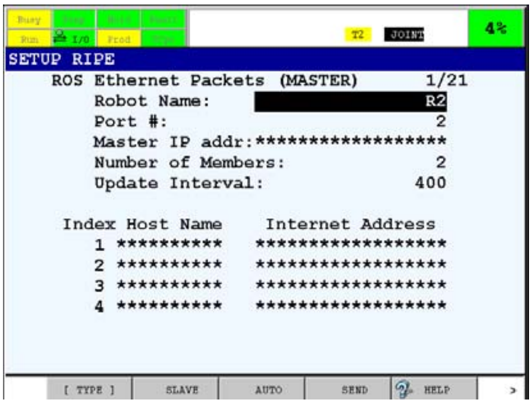
Robot ring consists of one master controller and the other slave controllers. When the Ethernet Encoder option is used, the master encoder needs not to be connected with the master controller. However, it is recommended that the master encoder is connected directly with the master controller for decreasing the communication traffic.

At first, set the slave controller when robot ring is set. Set the master controller after all the slave controllers are set. After you finish setting the master controller, the power of all the controllers is cycled automatically, and the setup is finished. The following is the detail procedure.

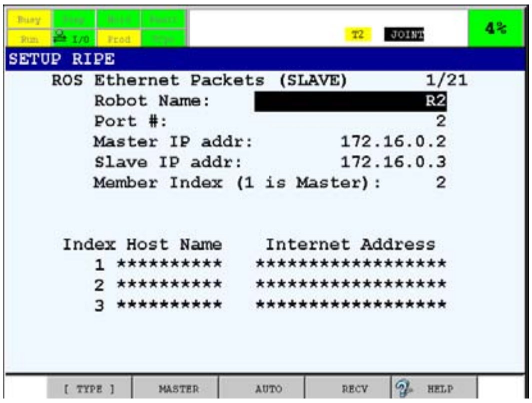
Slave Controller Setup

Set up the slave controller according to the following procedure.

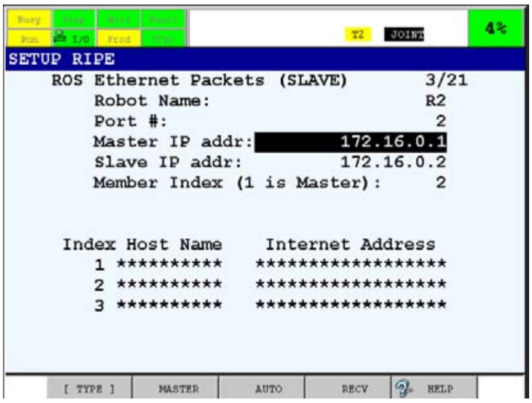
1. Press [MENUS] on the teach pendant.
2. Select [6 SETUP].
3. Select [Host Comm].
4. Select [RIPE], and press [ENTER] button.



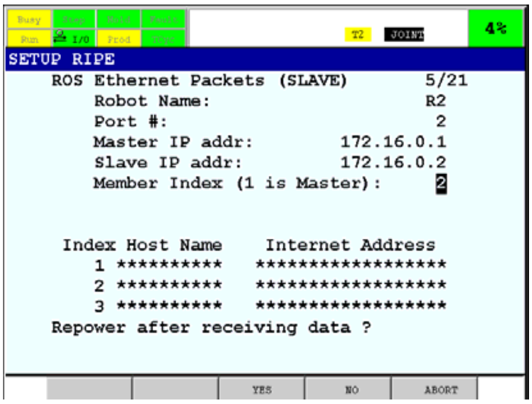
5. Press F2, [SLAVE], and display the screen “ROS Ethernet Packet (Slave)” as the following. When the screen of the slave controller is open and F2 button is [MASTER], this procedure is not necessary.



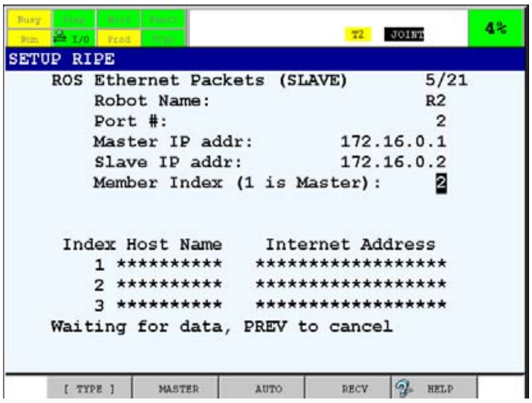
6. Set “Master IP address” to the IP address of the master controller in the port used in the robot ring.



7. Set [Member Index]. Set the IP address of the port used in the robot ring in “5.9.4.3 Setting IP Address” to the continuous number whose first number is the master controller. Input the order from this IP address. As a function of this setting, [Slave IP addr] is set automatically and verify that the displayed IP address and the IP address which is set this controller to are same.
8. Finish the above procedure, and press F3, [AUTO]. The following screen is displayed. You are asked whether this controller is cycled after receiving data, then press F3, [YES].



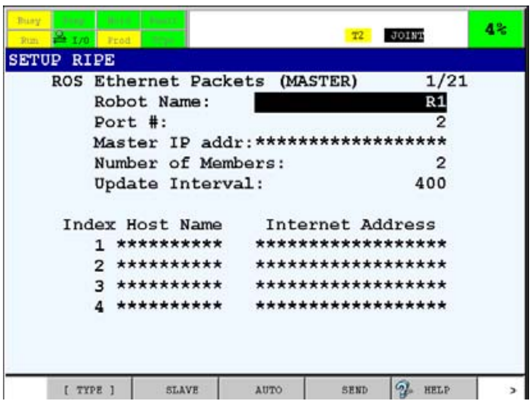
9. Perform the above procedure in all the slave controllers. All the slave controllers are allowed to wait to receive data from the master controller.



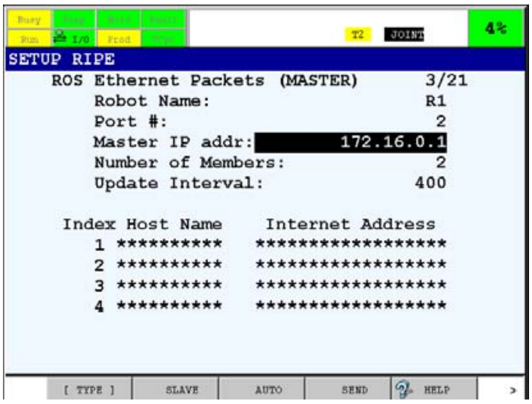
Master Controller Setup

After you finish setting all the slave controllers, set up the master controller according to the following procedure.

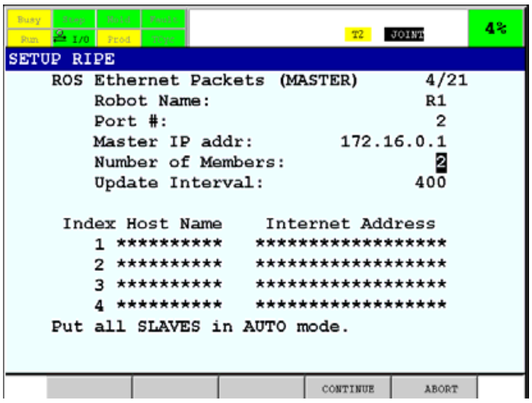
1. Press [MENUS] on the teach pendant.
2. Select [6 SETUP].
3. Select [Host Comm].
4. Select [RIPE], and press [ENTER] button.
5. When the screen “ROS Ethernet Packet (Slave)” is displayed, press F2, [MASTER], and change the following screen of the master controller. When the screen of the master controller is open and F2 button is [SLAVE], this procedure is not necessary.



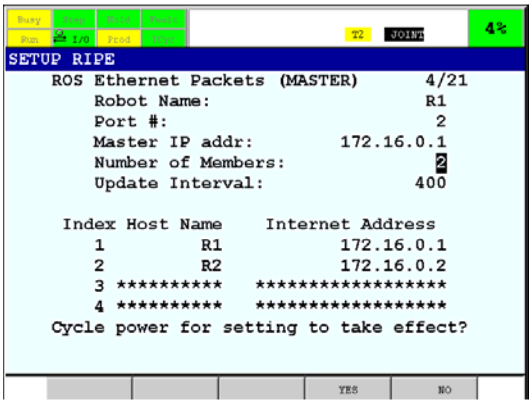
6. Set “Master IP addr” to the IP address of the master controller in the port used in the robot ring. This is the IP address of this controller.



7. Set “Number of Members” to the number of controllers which join the robot ring.
8. Set “Update Interval” as necessary. This is the interval to check whether the robot is online.
9. Finish the above procedure, and press F3, [AUTO]. The following screen is displayed.



10. The massage, “Put all SLAVES in AUTO mode.” is displayed. According to the previous Slave Controller Setup, all the slave controllers are allowed to wait to receive data from the master controller, and so press F4, [CONTINUE]. The following screen is displayed.



11. The robot names and IP addresses of all controllers join the robot ring are displayed on [Host Name]. You are asked to cycle the power of the controller in order to reflect the setting, and so press F4, [CONTINUE]. Then, all the controllers are cycled, and the setup is finished.

> ⚠**CAUTION**
> Don’t change the robot name and the host name after you finish setting the robot ring. The robot name and the host name need to be same. If you want to change them, restart the setting of robot ring from the beginning.

5.9.5 Ethernet Encoder Setup

Ethernet Encoder is set on Encoder setup menu.

Set up the Master Encoder

The following setup is performed on the controller connected physically with the encoder used as the Ethernet Encoder.

> ⚠**CAUTION**
> It is recommended that Encoder used as Ethernet Encoder is connected into RIPE Master Controller for decreasing the communication traffic.

1. Press [MENUS] on the teach pendant.
2. Select [6 SETUP].
3. Select [Encoders].
4. Set “Encoder Axis”. Set the servo axis number used for the tracking encoder. The valid number is from 1 to 32. For example, set “Encoder Axis” to 1 in case of one axis tracking. Set “Encoder Axis” of Encoder 1 to 1 and set “Encoder Axis” of Encoder 2 to 2 in case of two axes tracking.
5. Set “Ethernet Master RIPE Id” to RIPE id of this controller. On a RIPE Ethernet Network, each controller has a unique id. The system variable $PH_ROSIP.$MY_INDEX is this id. The controller with RIPE_id ($PH_ROSIP.$MY_INDEX) equal to 1 is the RIPE network master.

> ⚠**CAUTION**
> “Ethernet Master RIPE Id” is RIPE id of the controller connected with the master encoder.

6. Set “Ethernet Master Encoder” to the encoder number.
7. You should cycle power right away once you finish entering items. However, if you have more than one encoder need to be setup as Ethernet Encoder, you can wait until all of them are setup then cycle power.

For example, you want to set the encoder 1 of RIPE 1 controller (with $PH_ROSIP.$MY_INDEX = 1) as Master encoder, you set “Ethernet Master RIPE Id” to 1 and “Ethernet Master Encoder” to 1. The master controller screen is displayed. Encoder Set up menu is displayed as below.

```
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
```

Set up the Slave Encoder

The following setup is performed on the controller where you want to use Ethernet Encoder. You can set any encoder in any other controllers as slave encoder to that master encoder.

1. Press [MENUS] on the teach pendant.
2. Select [6 SETUP].
3. Select [Encoders].
4. Set “Ethernet Master RIPE Id” and “Ethernet Master Encoder” to the value set in Set up the Ethernet Master Encoder.
5. You should cycle power right away once you finish entering items. However, if you have more than one encoder need to setup as Ethernet Encoder, you can wait until all of them was set up then cycle power.

For example, when the master encoder was set as the Ethernet Master Encoder procedure example in the same RIPE network, you get into Encoder setup menu and set 1 to “Ethernet Master RIPE Id” and 1 to “Ethernet Master Encoder”.

```
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
```

5.9.6 Verify Setup

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

1. Go to the encoder set up menu of the master encoder on the controller where the master encoder was set.
2. Move the conveyor physically and verify “current count” change. When you want to verify it with the conveyor stopped, use “Simulate”. Set “Simulate: Enable:” to ON and “Rate” to 10, and verify “current count” change.

```
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
```

3. Go to the encoder set up menu of the slave encoder. The value of “current count” of the slave encoder should be continuously changing as a function of the value of “current count” of the master encoder. If it is not changed, check the Ethernet Encoder Setup.

```
SETUP   Encoders
             Encoder   Number:     3
1   Encoder   Axis:                                    1
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
```

4. Stop the conveyor, and verify that the change of “current count” is stopped. If Simulate is enable, set “Simulate: Enable” to OFF, and verify that the change of “current count” is stopped.

```
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
```

5. Go to the encoder set up menu of the slave encoder, and verify that the change of “current count” is stopped. Verify that “current count” of the master encoder and “current count” of the slave encoders are same. If they are same, the setup is correct. If they are not same, check the Ethernet Encoder Setup.

```
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
```

5.10 SERVO CONVEYER LINE TRACKING FUNCTION

5.10.1 Overview

Servo Conveyer Line Tracking function is the function for using an extended axis as a conveyer. Therefore, the robot can track the conveyer that is indexed with high accuracy. This function requires Line Tracking option and Servo Conveyer Line Tracking option.
Servo Conveyer Line Tracking option includes Multi Motion Group option and Continuous Turn option. These are used for keeping on moving an extended axis as a conveyer. In addition, Independent Auxiliary Axis option is necessary to control extended axis.

5.10.2 Setup

Please set up Servo Conveyer Line Tracking system by the following step.

5.10.2.1 Independent Extended Axis Setup
5.10.2.2 Servo Conveyer Setup
5.10.2.3 TP Program for Servo Conveyer
5.10.2.4 Tracking Schedule Setup
5.10.2.5 Example of TP Program

5.10.2.1 Independent extended axis setup

In Servo Conveyer Line Tracking function, you can use Independent Extended Axis as a conveyer. Please setup Independent Extended Axis on ROBOT MAINTENANCE MENU at Control Start.

Procedure: Controlled Start

Step

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 push the ENTER key. After a while a Control Start Menu appears.

Procedure: Robot Maintenance

Step

1. Press the MENUS key to display the screen menu. Select “9 MAINTENANCE” and push the ENTER key. ROBOT MAINTENANCE screen appears.

```
                                   CTRL START MENU
                                          1/1
      MENUS
     1 Tool SetUp
     2 S/W INSTALL
     3 S/W VERSION
     4 Variables
     5 File
     6 ALARM
     7 Port Init
     8 MEMORY
     9 MAINTENANCE
     0   NEXT --
[ TYPE ]
```

2. Select “Independent Axes” and push “F4:MANUAL” key.

```
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
```

3. “Independent Axes Setup Menu” appears. If you add axis, please select “2: Add Axis”. Push “2” key and the ENTER key and then “MOTOR SELECTION” appears. When axis setup is finished, you return to “Independent Axes Setup Menu”. If you finish the axis setup, please select “4 EXIT”.

```
*** Group 2 Total Axes Installed =   0
 1. Display/Modify Nobot Axis 1~4
 2. Add Axis
 3. Delete Axis
 4. EXIT
Select Item? 2
```

NOTE
Please refer to the mechanical specification for the following procedure.

4. Select a servo motor which is used as Independent Axis. Select “1 Standard Method” and then select MOTOR SIZE and MOTOR TYPE.

```
-- MOTOR SELECTION

1: Standard Method
2: Enhanced Method
3: Direct Entry Method
Select ==> 1
```

```
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
```

```
      MOTOR TYPE

1. /2000          11. /4000
2. /3000          12. /5000
      13. /6000
Select ==> 11
```

For example, if you want to select β*i*S2/4000*i*, please select 85 and 11.

5. Select a current limit for amplifier. If current limit for amplifier is 20A, please select 10.

```
  CURRENT LIMIT FOR AMPLIFIER

2.     4A          10.    20A
5.    40A          12. 160A
80A

Select==> 10
```

6. Select "2 Rotary Axis" as Independent axis type.

```
-- INDEPENDENT AXES TYPE –
1. Linear Axis
2. Rotary Axis

Select? 2
```

7. Enter the number of revolution of the motor which corresponds to one pitch of the conveyer. By this setting, a conveyer moves one pitch when Independent Axis moves 360 degrees.

Pitch:	The distance of conveyer when move conveyer to one bucket of conveyer
Bucket:	A part of a conveyer is divided by a constant distance.

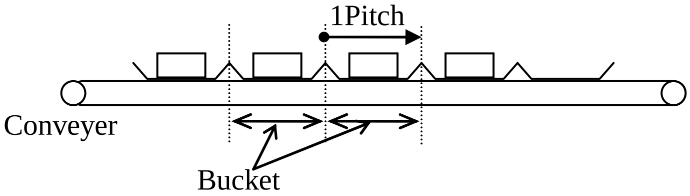


Fig. 5.10.2.1(a) Bucket and Pitch

```
-- GEAR RATIO --
Enter Gear Ratio?
```

If a conveyer rotates “N” times and a motor of it rotates “M” times, the gear ratio can be calculated by the following. Please enter the calculated value.

$$\mathrm{GearRatio} = \frac{M}{N \times \mathrm{Number\ of\ Buckets}}$$

- It is also possible to calculate the gear ratio from the set value (Motor Gear teeth and so on) of Servo Conveyer Setup. In this case, the gear ratio can be calculated by the following.

$$\mathrm{GearRatio} = \frac{\mathrm{Rotor\ Input\ Gear\ Teeth} \times \mathrm{Conveyer\ Belt\ Teeth}}{\mathrm{Motor\ Gear\ Teeth} \times \mathrm{Rotor\ Output\ Gear\ Teeth} \times \mathrm{Number\ of\ Flight}}$$

8. 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
```

9. 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.

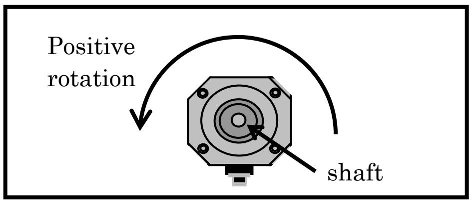


Fig. 5.10.2.1(b) Positive rotation of a motor

10. Enter the limit of the axis. Please 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
```

11. Enter the mastering position of the axis. Please 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)?
```

12. 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.

```
-- 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)?
```

    Next, enter the 2nd ACC/DEC time (acc_time2). Default value of acc_time2” is the default value. Please set half the value of acc_time1.

```
Default value of acc_time2 = 192 (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 need to change the value, choose “2:No Change”.

```
Enter Accel Time 2 (ms)?
```

13. 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)?
```

    min_acctime should be the sum of acc_time1 and acc_time2. Please choose “1:Change” and enter it.

```
Enter Minimum Accel Time (ms)?
```

14. 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 --

$$\text{Load Ratio} = \frac{\text{Load Inertia} + \text{Motor Inertia}}{\text{Motor Inertia}}$$

Enter Load Ratio -> (0:None 1->5:Valid)

15. Enter Amplifier Number.

```
--SELECT AMP NUMBER--
Enter amplifier number (1-56)?
```

16. 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?
```

17. Enter Brake Number.

```
--BRAKE SETTING --
Enter Brake Number (0~16)?
```

18. 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)
```

19. After the above setting, "Independent Axes Setup Menu" appears. Please select "4 EXIT" and finish the axis setup.

```
*** Group 2 Total Axes Installed =    0
  1. Display/Modify Nobot Axis 1~4
  2. Add Axis
  3. Delete Axis
  4. EXIT
Select Item? 4
```

20. After axis setup is finished, Press the FCTN key to display the function menu. Select “1 START (COLD)” and push the ENTER key and then Cold Start is executed.

```
ROBOT MAINTENANCE
                                   1/10
Setup Robot Sys      FUNCTION
                   1 START(COLD)
  Group Robot L    2 RESTORE/BACKUP
    1     M-3iA/(  3 PRINT SCREEN
    2     Indepen  4 UNSIM ALL I/O

[ TYPE ] ORD NO    AUTO    MANUAL
```

5.10.2.2 Servo conveyer setup

Please set up Servo Conveyer by the following step. The setting of encoder and the setting of Continuous Turn are updated automatically by setting Servo Conveyer.

Procedure 1-1 Servo Conveyer Setup

Step

1. Press MENUS.
2. Select “SETUP”.
3. Press “F1: TYPE”.
4. Select “Indexers”. You will see a screen as 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. Please refer to TP program for Servo Conveyer (INDXG*.TP).
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 push “F2: DETAIL” key or Enter Key. 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 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.

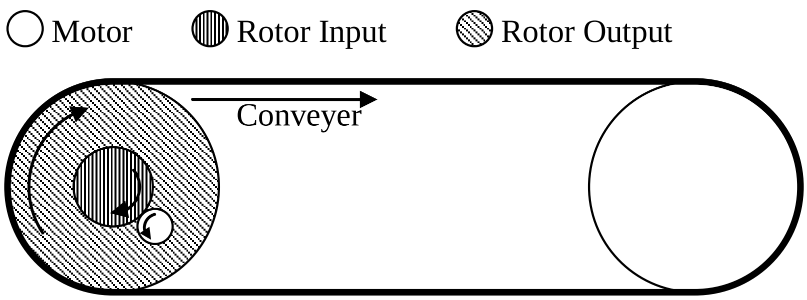


Fig. 5.10.2.2 Relationship between motor, rotor and conveyer

It is also possible to set up from the relation between the number of revolution of motor and the number of revolution of the conveyer. If a conveyer rotates “N” times and a motor of it rotates “M” times, please setup as following. “M” and “N” must be an integer.

a. Enter the value of “N” in Continuous Rotation to “Motor Gear teeth”
b. Enter the value of “M” in Continuous Rotation to “Rotor input Gear teeth”
c. Enter “1” to “Rotor output Gear teeth” and “Conveyer belt teeth”

14. Move the cursor to “Number of Flight” and enter the number of bucket 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, push “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.

5.10.2.3 How to create TP program for servo conveyer

It is necessary to prepare TP program for moving the servo conveyer on tracking because the servo conveyer is setup as extended axis. By following step, 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: How to create TP program for Servo Conveyer

Step

```
SETUP Indexers
                                          7/7
          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]
```

> **NOTE**
> Before generating indexer program, please set max payload which is used in an indexer tracking motion on MOTION/PAYLOAD SET display.

1. Move the cursor to “Create Index program” in “SETUP Indexers” and push “F2: DETAIL” key or ENTER key. 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 Indexer:   1
  1 Index Speed (part/min)               100
  2 Index Dwell (ms):                      0
  3 Indexer Register start                60
  4 Generate Index program
```

2. Move the cursor to “Index Speed (part/min)” and enter the value of Index Speed of the servo conveyer. Please set the number of pitches per minute.

3. 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. Please refer to following figures.
   If you want to use a conveyer as Index Conveyer, it is necessary to "0" and over as Index Dwell. If you want to move a conveyer at constant speed, it is necessary to set "-1" as Index Dwell.
   Movement Time of Conveyer per Pitch is calculated from Index Speed.
   Index Dwell is "0" and over:
   Movement Time of Conveyer per Pitch [ms] = (60*1000/Index Speed) – Index Dwell

   Index Dwell is "-1":
   Movement Time of Conveyer per Pitch [ms] = (2*60*1000/Index Speed)

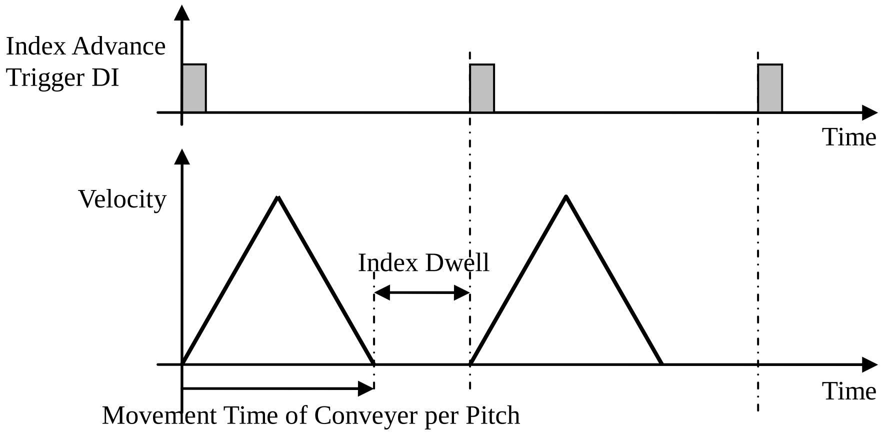


Fig. 5.10.2.3(a) Speed pattern if <u>Index Dwell is "0" and over</u>

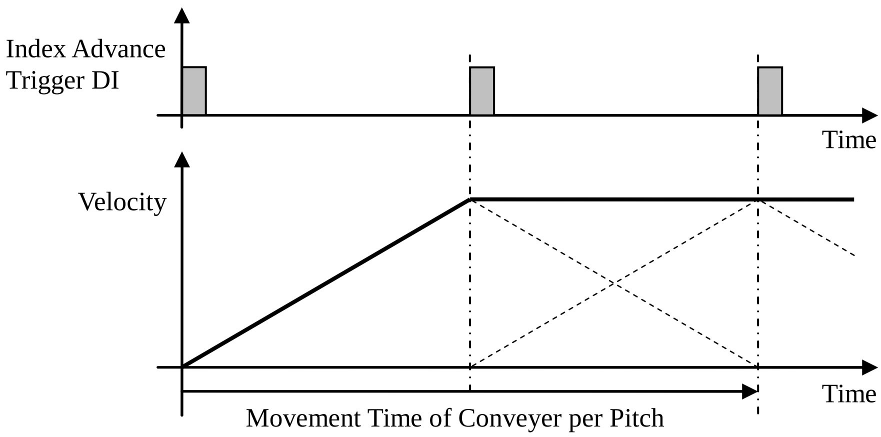


Fig. 5.10.2.3(b) Speed pattern if <u>Index dwell is "-1"</u>

4. 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 TP program.
   For example, R[60] and R[61] are used when "Indexer Resister start" is "60". R[60] is used for setting the number of DO. The specified DO by R[60] is used for checking whether the conveyer finishes moving to initial position. R[61] is used for counting the number of pitches of the conveyer. Please refer to example of INDXG*.TP.
5. After the above setting move the cursor to "Generate Index program" and push "F2: Create" key. 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. Please rename INDXG*.TP and use it.
- “Part Rate exceeds 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. Please adjust the setting (Index Speed, Index Dwell) so that the acceleration decreases.
- If there is INDXG*.TP and you push “F2: Create” key, “Index Program already exist” is displayed. In this case, INDXG*.TP is not created. If you want to create TP program, please push “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 current setting.
  - If you push only “F3 Replace” key, “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 push “SHIFT” key and “F3: Replace” key. Please adjust the setting so that the acceleration decreases.

Sample Program

The following is an example of INDXG*.TP.

Example of INDXG*.TP

1:J P[1] 50% CNT0 ;	Movement to P[1] as an initial position.
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] 180msec CNT100 INC ACC66 ;	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;	Count the number of pitches
11: IF R[61:G2 cur slot ID]=32,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 ;	Movement 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
};
```

- At line 1, the conveyer moves to the initial position P[1].
- DO is output to check whether the conveyer finishes moving to the initial position by instructions from line 2 to line 4.
- At line 8, the conveyer moves 1 pitch. In other words, the extended axis that is used as the conveyer moves 360 degrees.
- ”$INDEXER[*].$INDEX_MV=1” at line 7 and “$INDEXER[*].$INDEX_MV=0” at line 9 are setting for watching input of Index Advance Trigger DI (* is number of Servo Conveyor). Motion instruction at line 8 is not executed until the DI changes from OFF to ON.
- IF instruction at line11 checks whether the conveyer makes one revolution. In this example case, the conveyer makes one revolution per 32 pitches.
- At line 14, the conveyer moves to P[1] to correct iteration error. Every time the conveyer makes one revolution this line is executed.
- R[60] and R[61] are set by specifying “Indexer Register start”. R[60] is used for setting the number of DO. The specified DO by R[60] is used for checking whether the conveyer finishes moving to initial position. R[61] is used for counting the number of pitches of the conveyer.

5.10.2.4 Tracking schedule setup

Tracking Schedule Setup for servo conveyer is the same as ordinary Line Tracking function. Please refer to the operator’s manual for Line Tracking.

5.10.2.5 Example of main program

Basically, how to teaching TP program is the same as ordinary Line tracking function. However, it is necessary to run TP program for moving a servo conveyer by multitask. CNVY.TP for moving a servo conveyer is run at line 3 in the following Sample Program. It is possible to create TP program for moving a servo conveyer (INDXG*.TP) by the procedure of 5.10.2.3”How to Create TP Program for Servo Conveyer”.

Example of Main Program

```
1:   J P[1] 100% FINE ;
2:   LINE[1] ON ;
3:   RUN CNVY;                          CNVY.TP for moving Servo Conveyer is executed.
4:   LBL[1] ;
5:   WAIT DI[2]=ON      ;
6:   LINECOUNT[1] R[1] ;
7:   SETTRIG LNSCH[1] R[1] ;
8:   CALL TRACK      ;                  TRACK.TP for tracking motion is executed.
9:   CALL NORM      ;                   NORM.TP for non-tracking motion is executed.
10:   JMP LBL[1] ;
```

- Motion instruction at line 1 moves robot to the initial position.
- At line 3, “RUN” CNVY.TP for moving the conveyer.
- At line 5, wait until a workpiece on the conveyer is detected. DI[2] is used for detecting a workpiece.
- At line 6, 7, trigger value is got and set before tracking motion.
- At line 8, “CALL” TRACK.TP for tracking motion.
- At line 9, “CALL” NORM.TP for non-tracking motion.

5.10.3 Function for Servo Conveyer Line Tracking

5.10.3.1 Dynamic error tune variable

On Servo Conveyer Line Tracking, please adjust a dynamic error of the tracking motion by using the following parameter instead of $LNCFG_GRP[].$SOFT_DELAY, $SRVO_DELAY.
Please refer to 5.4 FINE TUNING HIGH SPEED ACCURACY of Line Tracking OPERATOR'S MANUAL too.

$SLTK_GRP[number of servo conveyer group].$SRVO_DELAY

5.10.3.2 Wait indexer stop function

This function is disabled by default. When this function is disabled, a robot starts/ends a tracking regardless of the motion of the servo conveyer. Therefore, for example, if robot starts a tracking motion while a servo conveyer accelerates, the acceleration of the robot is increased in order to catch up the conveyer and the motion of the robot might become aggressive.
When this function is enabled, Robot starts/ends a tracking motion while a servo conveyer stops. By this setting, the acceleration at the start/end of tracking motion is restricted.

NOTE
If you want to enable this function, it is necessary to set Index Dwell to 0 and over and generate TP program for Servo Conveyer again. If there is no time the conveyer stops, it is not possible to start/end a tracking motion.

The flag for switching the setting of this function is bit 1 and bit 2 of $INDX_TRACK[schedule number].
If you want to enable this function, please set these flag by the following procedure.

NOTE
If the setting of this function is changed, please create TP program for the servo conveyer again.

Procedure: Wait Indexer Stop Function Setup

Step

1. Setting bit 1 of $INDX_TRACK[schedule number]

1.1 Please divide the value of $INDX_TRACK[schedule number] by "2" and then check current setting by following step.
   - If the integer part of the calculated value is <u>Odd</u> number, bit 1 is TRUE.
   - If the integer part of the calculated value is <u>Even</u> number, bit 1 is FALSE.

1.2 Please change the value of $INDX_TRACK[schedule number] according to the current setting.
   - If bit 1 is TRUE and you want to <u>change it to FALSE</u>, please subtract "2" from the value of $INDX_TRACK[schedule number].
   - If bit 1 is FALSE and you want to <u>change it to TRUE</u>, please add "2" to the value of $INDX_TRACK[schedule number].

2. Setting bit 2 of $INDX_TRACK[schedule number]

2.1 Please divide the value of $INDX_TRACK[schedule number] by "4" and then check current setting by following step.
   - If the integer part of the calculated value is <u>Odd</u> number, bit 2 is TRUE.
   - If the integer part of the calculated value is <u>Even</u> number, bit 2 is FALSE.

2.2 Please change the value of $INDX_TRACK[schedule number] according to the current setting.
   - If bit 2 is TRUE and you want to <u>change it to FALSE</u>, please subtract "4" from the value of $INDX_TRACK[schedule number].

- If bit 2 is FALSE and you want to change it to TRUE, please add “4” to the value of $INDX_TRACK[schedule number].

3. After the above setting, please Power Off/ON.

5.10.3.3 KAREL program for servo conveyer line tracking

Servo Conveyer Line Tracking provides the following KAREL programs. By using these, it is possible to get a trigger value and save it every time Servo Conveyer moves a fixed distance. It is also possible to use Discard Line function and Stop Conveyer function.

- SLTKINIT
- SLTKREST
- SLTKPSHQ
- SLTKPOPQ
- SLTKDELQ
- SLTKRSTQ

This subsection explains these KAREL programs and Discard Line, Stop Conveyer.

> **NOTE**
> It is necessary to set $KAREL_ENB=1 to use KAREL program.

SLTKINIT

This program clears information about trigger values. Normally, this program is called once when the system is started.

Argument1: Specify the number of tracking schedule which Servo Conveyer Line Tracking uses.
Argument2: Specify the position register number that is saved an initial position of the conveyer.
Argument3: Specify the position register number.

The specified position register by argument 3 is set to the distance between the current conveyer position and an initial position of the conveyer. The value within 1 pitch (from 0 to 360 degrees) is set. This is available for adjusting the conveyer position at the start.

SLTKREST

Because the distance of one pitch of Servo Conveyer is constant, an encoder value can be got as a trigger value every a constant distance. This program is used for setting a reference value of the trigger. When SLTKREST is called, the encoder value is saved as a reference value of the trigger. It is necessary to call this program once just after the servo conveyer moves to an initial position.

Argument1: Specify the number of tracking schedule which Servo Conveyer Line Tracking uses.

In the following case, PR[2] is set to the distance between the current conveyer position and an initial position of the conveyer by SLTKINIT at line 1. Next, the conveyer moves to an initial position by the motion instruction and Reference position for trigger is saved by SLTKREST.

```
1:   CALL SLTKINIT(1,1,2)      ;          Initialization and Setting of Pos. Register.
2: J PR[2] 50% CNT0 INC        ;          Movement to an initial position.
3:   CALL SLTKREST(1) ;                   Setting of Reference Pos. for Trigger.
```

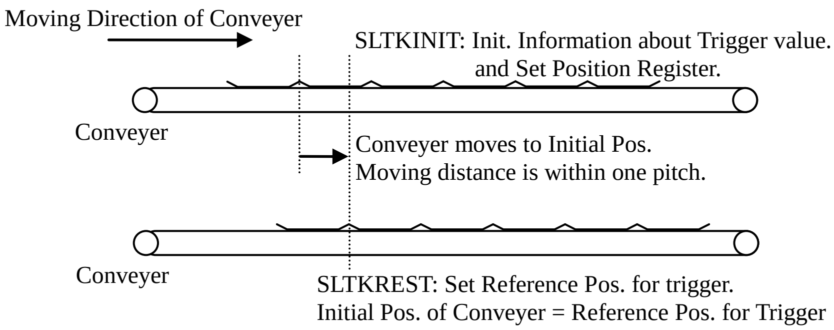


Fig. 5.10.3.3(a) SLTKINIT, SLTKREST

SLTKPSHQ

This is used for saving trigger value after the conveyer moves the number of pitch.

Argument1: Specify a number of tracking schedule which Servo Conveyer Line Tracking uses.
Argument2: Specify a number of pitch

If you want to change the save-able number of trigger values, please change the value of the following system variable and power off/on.

$SLTKSCH[n].$QUE_SIZE: Default value is 20 and max value is 100.

"n" is the number of tracking schedule which Servo Conveyer Line Tracking uses.

> **NOTE**
> When the saved number of trigger values become the specified value by $SLTKSCH[n].$QUE_SIZE, the oldest saved trigger value is removed.

It is necessary call this program after the conveyer moves the specified number of pitch. When this program is called, a trigger value is calculated from the reference trigger value and the specified number of pitch and it is saved. At the first, the reference trigger value is saved as a trigger value.
For example, if the conveyer moves 1 pitch and the trigger value is saved once, Argument2 should be set "1". If the conveyer moves 2 pitches and the trigger value is saved once, Argument2 should be set "2". The following shows the case that trigger values are saved during the conveyer moves four pitches.

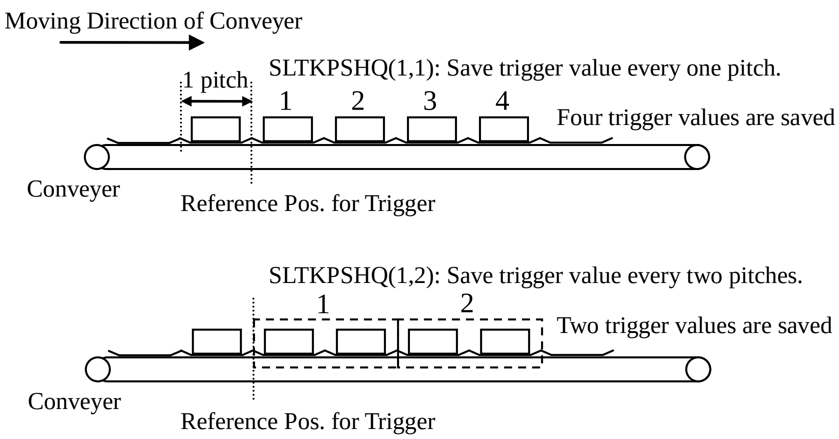


Fig. 5.10.3.3(b) SLTKPSHQ

SLTKPOPQ

This program is used for getting the saved trigger value. When this program is called, it is possible to get the oldest saved trigger value. If you succeed in getting the trigger value, the trigger value is removed and it is impossible to get it again.

Argument1: Specify a number of tracking schedule which Servo Conveyer Line Tracking uses.
Argument2: Specify a register number for getting a trigger value.
Argument3: Specify a register number for getting a status. If SLTKPOPQ fails to get a trigger value the value of status become nonzero.

The following shows the case that there are four saved trigger values.

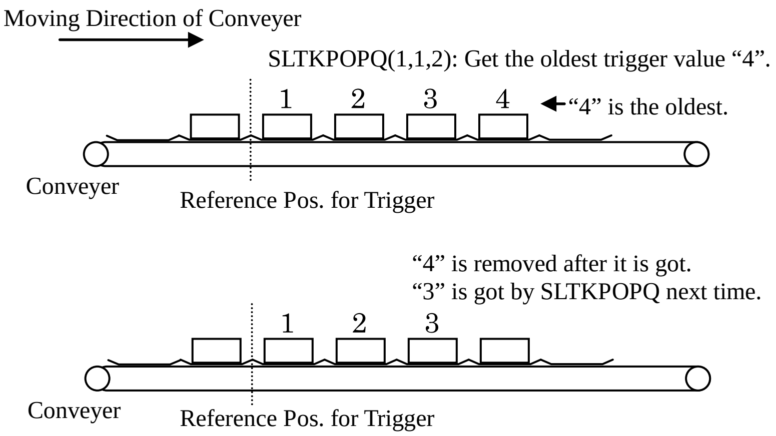


Fig. 5.10.3.3(c) SLTKPSHQ

SLTKDELQ

This program is used for deleting the specified the saved trigger value. It is not possible to get the deleted trigger value by SLTKPOPQ.

Argument1: Specify a number of tracking schedule which Servo Conveyer Line Tracking uses.
Argument2: Specify a number of trigger to delete.

For example, if argument2 is “2”, SLTKDELQ deletes the second trigger value “2” from the newest trigger value.

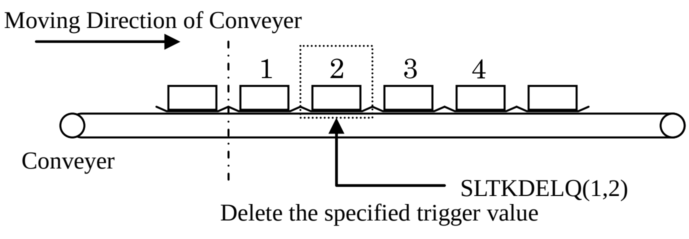


Fig. 5.10.3.3(d) SLTKDELQ

SLTKRSTQ

This program turns off the specified DO for Stop Conveyer. Please refer to STOP CONVEYER below.

Argument1: Specify a number of tracking schedule which Servo Conveyer Line Tracking uses.

DISCARD LINE

If a corresponding position to the trigger value passes the discard line, it is possible to discard the trigger value. It is necessary to set Discard Line on the basis of downstream boundary of the tracking area.

It is necessary to set following system variables in order to use this function.

$SLTKSCH[n].$SLQ_ENABLE:	TRUE
$SLTKSCH[n].$DISCARD_BND:	Specify the distance from the downstream boundary of the tracking area to the Discard Line. The unit is [mm]
$SLTKSCH[n].$SENS_X:	Specify the distance from the origin of Tracking frame to Reference Pos. along X axis of Tracking frame. The unit is [mm]

"n" is the number of tracking schedule which Servo Conveyer Line Tracking uses.

If $DISCARD_BND is negative, Discard Line is set up Down-stream boundary. Normally, you should set $DISCARD_BND to the negative value. If a corresponding position to the trigger value passes the discard line, it is not possible to get the trigger value by SLTKPOPQ. The oldest saved trigger value is used for checking whether this function discard the trigger value.
In the following case, because the corresponding position to the trigger value "4" passes the discard line, "4" is discarded. After that, the corresponding position to the trigger value "3" is checked whether this function discard the trigger value.

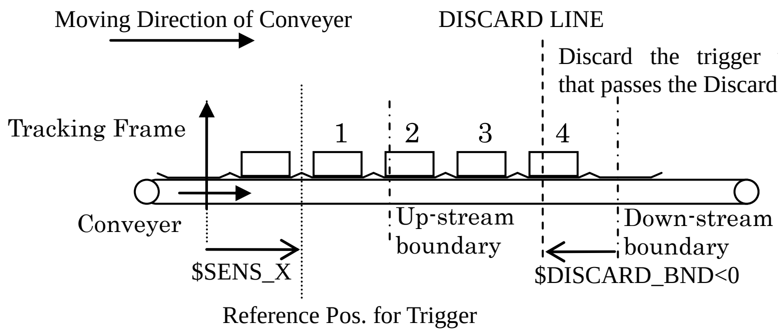


Fig. 5.10.3.3(e) Discard line

STOP CONVEYER

If a corresponding position to the trigger value passes Discard Line, the specified DO become ON. While the Specified DO is ON, you should not set "Index Advance Trigger DI" to ON in order to stop the conveyer.
It is necessary to set Discard Line so that the desired tracking motion can be done at the conveyer stop position.

It is necessary to set following system variables in order to use this function.

$SLTKSCH[n].$SLQ_ENABLE:	TRUE
$SLTKSCH[n].$DISCARD_BND:	Specify the distance from the downstream boundary of the tracking area to the Discard Line. The unit is [mm]
$SLTKSCH[n].$SENS_X:	Specify the distance from the origin of Tracking frame to Reference Pos. along X axis of Tracking frame. The unit is [mm]
$SLTKSCH[n].$STOPBELT:	TRUE
$SLTKSCH[n].$STOP_DONUM:	Specify DO number for Stop Conveyer

"n" is the number of tracking schedule which Servo Conveyer Line Tracking uses.

When Stop Conveyer function is enabled and the corresponding position to the trigger value passes Discard Line, the trigger value is kept. While the Specified DO is ON, it is possible to get the trigger value by SLTKPOPQ. If you want to change the specified DO from ON to OFF, it is necessary to use SLTKRSTQ. The next trigger value is used for checking the Stop Conveyer after SLTKRSTQ is executed.
In the following case, because the corresponding position to the trigger value "4" passes the discard line, the specified DO becomes ON. According the DO, it is necessary to stop the conveyer. The trigger value

“4” can be got by SLTKPOPQ. After a tracking motion is done at the corresponding position to the trigger value “4”, the DO becomes OFF by executing SLTKRSTQ. According the DO, it is necessary to resume the conveyer. When SLTKRSTQ is executed, the corresponding position to the trigger value “3” will be checked whether this function discard the trigger value.

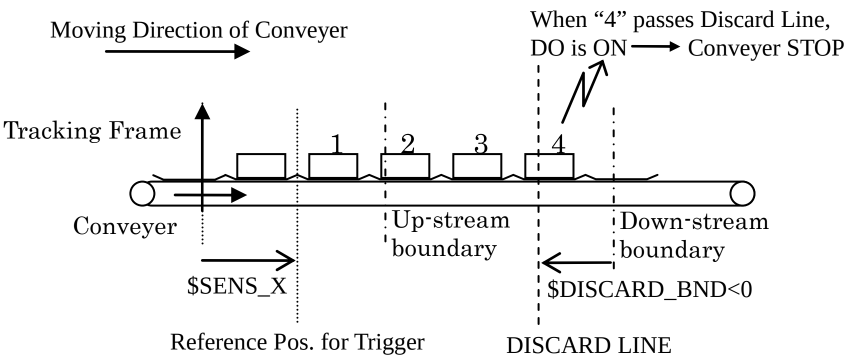


Fig. 5.10.3.3(f) Stop conveyer

> **NOTE**
> It is impossible to use both Discard Line and Stop Conveyer on the same Servo Conveyer.

Sample Program

The following are sample programs (Main Program and CNVY.TP for moving Servo Conveyer) in case of using Stop Conveyer.

- The used Tracking Schedule number is “1”.
- The initial position of the conveyer is saved to PR[1].
- The trigger value is saved every time the conveyer moves 2 pitches

In Main Program, the information about the trigger value is initialized by SLTKINIT at first. After that, TP program (CNVY.TP) for moving Servo Conveyer is executed by the other task.

Example of Main Program

1:J P[1] 100% FINE ;	
2: LINE[1] ON ;	
3: CALL SLTKINIT(1,1,2) ;	Information about trigger value is initialized and PR[2] is set.
4: RUN CNVY;	CNVY.TP for moving Servo Conveyer is executed.
5: LBL[1] ;	
6: WAIT .10(sec) ;	
7: CALL SLTKPOPQ(1,1,2) ;	The trigger value is got.
8: IF R[2]<>0,JMP LBL[1] ;	If it is impossible to get the trigger value, Jump to LBL[1]
9: SETTRIG LNSCH[1] R[1] ;	R[1] is set as a trigger value.
10: CALL TRACK ;	TRACK.TP for tracking motion is executed.
11: CALL SLTKRSTQ(1) ;	Turn off DO for Stop Conveyer.
12: CALL NORM ;	NORM.TP for non-tracking motion is executed.
13: JMP LBL[1] ;	

- At line 3, the information about the trigger value is initialized by SLTKINIT and PR[2] is set to the distance from current position to the position of PR[1].
- At line 4, RUN CNVY.TP in order to execute the program for moving Servo Conveyer by the other task.

- At line from 5 to 8, a trigger value is got by SLTKPOPQ every 100[msec]. At line 8, whether getting a trigger value succeed is checked by R[2]. If it is impossible to get a trigger value, jump to LBL[1] at line 5.
- At line 11, if DO for Stop Conveyer is ON, turn off the DO by SLTKRSTQ.

In TP program for moving Servo Conveyer, a trigger value is got every 2 pitches by SLTKPSHQ. R[3] is used for counting the number of pitches and SLTKPSHQ is executed according to the value of R[3].

Example of CNVY.TP

1:J PR[2] 50% CNT0 INC ;	Movement to the initial position.
2: CALL SLTKREST(1) ;	Setting of Reference Pos. for Trigger.
3: IF R[60:G2 Ready DO]=0,JMP LBL[3] ;	
4: DO[R[60]]=ON ;	Specified DO by R[60] become ON.
5: LBL[3] ;	
6: R[61:G2 cur slot ID]=0 ;	Reset R[61].
7: R[3]=0 ;	R[3] is the counter for SLTKPSHQ.
8: LBL[1] ;	
9: $INDEXER[1].$INDEX_MV=1 ;	Setting for waiting for DI trigger.
10:J P[2] 180msec CNT100 INC ACC66 ;	One pitch movement.
11: $INDEXER[1].$INDEX_MV=0 ;	Setting for waiting for DI trigger.
12: R[61:G2 cur slot ID]=R[61:G2 cur slot ID]+1;	Count the number of pitches.
13: R[3]=R[3]+1 ;	Count the number of pitches.
14: IF R[61:G2 cur slot ID]=32,JMP LBL[2] ;	
15: JMP LBL[4] ;	Jump to LBL[4] at line 19.
16: LBL[2] ;	
17:J PR[1] 100% CNT100 ;	Movement to PR[1] to correct iteration error.
18: R[61:G2 cur slot ID]=0 ;	Reset R[61]
19: LBL[4] ;	
20: IF R[3]<2,JMP LBL[1] ;	Check whether conveyer moves 2 pitches.
21: CALL SLTKPSHQ(1,2) ;	When conveyer moves 2 pitches, the trigger value is saved by SLTKPSHQ.
22: R[3]=0 ;	Reset R[3].
23: JMP LBL[1] ;	
/POS	
P[2:""]{ GP2: UF : 0, UT : 1, J1= 360.000 deg };	

- Because PR[2] is set to the distance from current position to the position of PR[1], at line 1, the conveyer moves from the current position to the position of PR[1].
- At line 2, Reference position for Trigger is set by SLTKREST.
- R[3] is the counter for SLTKPSHQ. At line 7, 13, 22, R[3] is initialized, counted and reset.
- At line 17, the conveyer moves to PR[1] to correct iteration error. Every time the conveyer makes one revolution this line is executed.
- At line 20, the number of pitches is checked. When the conveyer moves 2 pitches, SLTKPSHQ is executed to save to a trigger value.
- If you wan to change the number of pitches per trigger. Please change “2” at line 20, 21 to the desired number.

5.10.4 Limitations on Servo Conveyer Line Tracking

- Servo Conveyer Line Tracking function requires Line tracking function.
- It is necessary to separate a robot and an extended axis as a conveyer into different groups.
- 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. However, if you want to continue a tracking motion after a tracking motion of a different schedule, it is necessary to execute a normal motion once before next tracking motion starts after a tracking motion of a different schedule.
- 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.
- In a tracking program, Please add the wrist joint motion instruction (Wjnt) to all motion instruction or don’t add Wjnt to motion instruction. If there are a motion instruction with Wjnt and a motion instruction without Wjnt, TCP might be deviated from a destination at tracking motion.

APPENDIX

A CONVEYOR LIMITS AND TRACKING ACCURACY

A.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.

$$MaxConv.Speed_{mm/sec} = \frac{MaxError_{mm}}{ITPtime_{sec}}$$

For example, to obtain a tracking error of less than ±4 mm on a robot with an ITP time of 8 msec, the maximum conveyor speed would be:

500(mm/sec) = 4mm／8msec

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 (2 ms) so the conveyor can be much faster. However, this is bound by the process speed. Refer to Section 4.5.

A.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.

$$MaxConv.Accel_{mm/sec^2} = \frac{MinRobotAccel}{2}$$

For example:

$$200_{mm/sec^2} = \frac{400_{mm/sec^2}}{2}$$

The value of 200 mm/sec 2 is a general limit.

A.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 detects 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 (2 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.Speedmm/sec * 2 * ITPtimesec

For example:
3.2mm = 200mm/sec *2 * 8msec

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.Speedmm/sec * 2 * SystemTickmsec

For example:
0.8mm = 200mm/sec *2 * 2msec

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:

$$0.2_{mm} = 200_{mm/sec} * 1_{ms}$$

NOTE
The larger ErrorDist value will always be used.

NOTE
PaintTool disables the ACCUTRIG tracking accuracy feature.

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 msec, the resolution for a conveyor speed of 200 msec/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:

4 pulses/update =10 pulses/mm * 200 mm/sec * 0.002 sec/update

NOTE
Keep the encoder resolution above 10 pulses/update. Values around 40 or 50 are more desirable.

A.4 RESOLUTION OF THE ENCODER

When the robot seems to move too aggressively during tracking, one possible reason is low encoder resolution. The other possible reason is short tracking filter length.

- 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 software solution: 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 Section 3 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

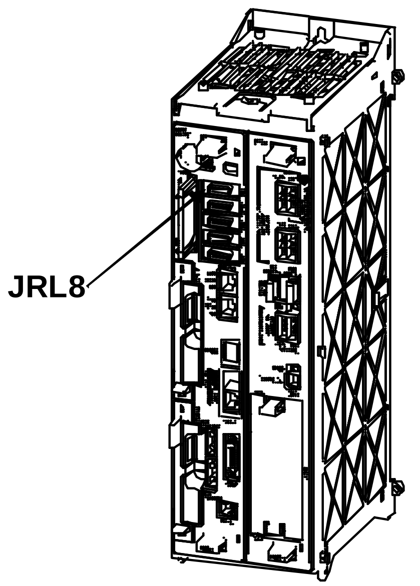
This occurs when the robot moves from a stationary position to the part on the conveyor the tracking filter is used. The filter length is the time robot has to catch up with the parts on the conveyor. When the filter length is short, the robot is heavy and the conveyor speed is high, the robot will be 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 (maximum 40) accordingly to a higher value.

B SCHEMATICS

B.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.

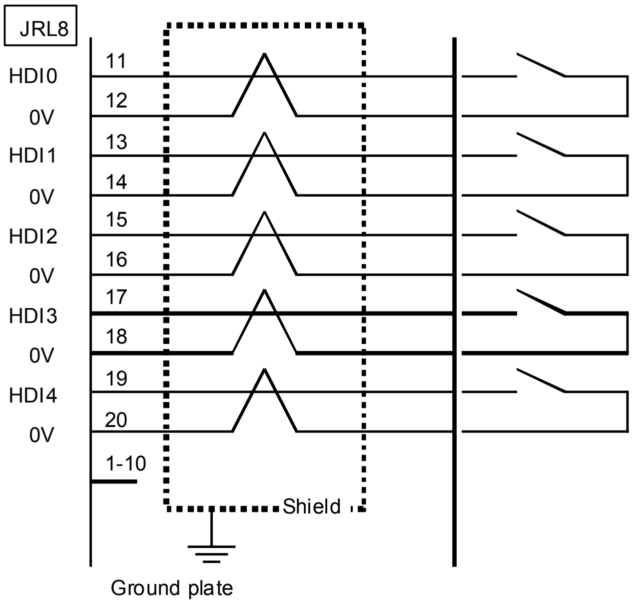


R-30*i*B

Main board
JRL8

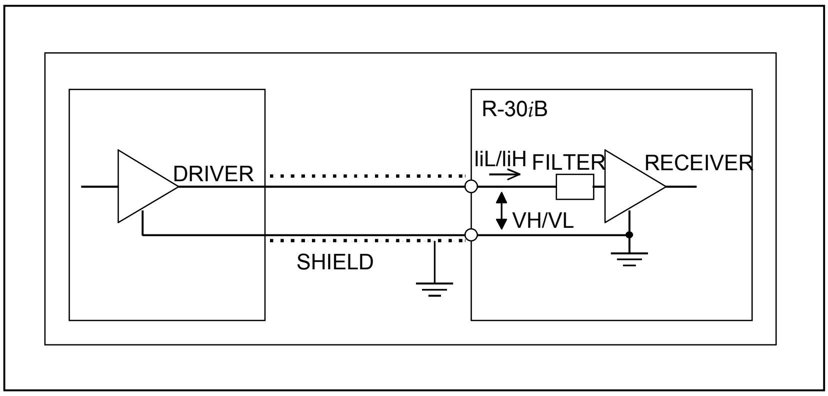
1	Reserved	11	HDI0
2	Reserved	12	0V
3	Reserved	13	HDI1
4	Reserved	14	0V
5	Reserved	15	HDI2
6	Reserved	16	0V
7	Reserved	17	HDI3
8	Reserved	18	0V
9	Reserved	19	HDI4
10	Reserved	20	0V

Fig. B.1(a) HDI interface



Recommended cable connector:
PCR-E20FA (Honda Tsushin Kogyo)
FI30-20S (Hirose Electric)
FCN-247J020-G/E (Fujitsu)
52622-2011 (Molex Japan)

Fig. B.1(b) HDI cable connections



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.

Fig. B.1 (c) Input signal rules for the high-speed skip (HDI)

C PULSECODER A860-0301-T001~T004

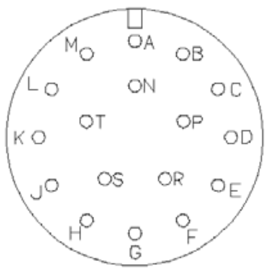
C.1 REQUIREMENTS

Required Components	R-30*i*B Controller A-Cabinet	R-30*i*B Controller B-Cabinet	Comments
Hardware			
Line Tracking Interface Board	A20B-8101-0421 (wide-mini slot) or A20B-8101-0601 (mini slot)	A20B-8101-0421 (wide-mini slot) or A20B-8101-0601 (mini slot)	Separate Detector Unit (SDU) - A02B-0323-C205 can be used in place of Line tracking Interface board. NOTE: SDU requires retrofit work to mount in the container (see Fig. C.3(a) through Fig. C.3(c)).
Fiber Optic (FSSB) Cable	A66L-6001-0023	A66L-6001-0023	
Pulsecoder Cable	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)	
	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)	
	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)	
Multiplexer cable	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)	When using multiple robots to track parts on the conveyor, use a Multiplexer to Controller cable, or use Ethernet encoder function (option).
	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)	

Required Components	R-30*i*B Controller A-Cabinet	R-30*i*B Controller B-Cabinet	Comments
Hardware			
Pulsecoder to Multiplexer cable	A05B-2451-K102(7M) A05B-2451-K103(14M)		

C.2 FIGURES

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	

Fig. C.2(a) Pulsecoder (A860-0301-T001) connection signal information

Item		Specification
Power voltage		5 (V) 5%
Current consumption		Up to 0.35 (A)
Working temperature range		0 to +60 (deg C)
Maximum response frequency		100×10^{3} (Hz)
Input shaft inertia		Up to 5×10^{-3} ($kg\cdot m^2$)
Input shaft startup torque		Up to 0.8 (Nm)
Rated loads	Radial	20 (N)
	Axial	10 (N)
Shaft diameter runout		0.02×10^{-3} (m)
Weight		Approx. 2.0 (kg)

Fig. C.2(b) Pulsecoder (A860-0301-T001) specifications

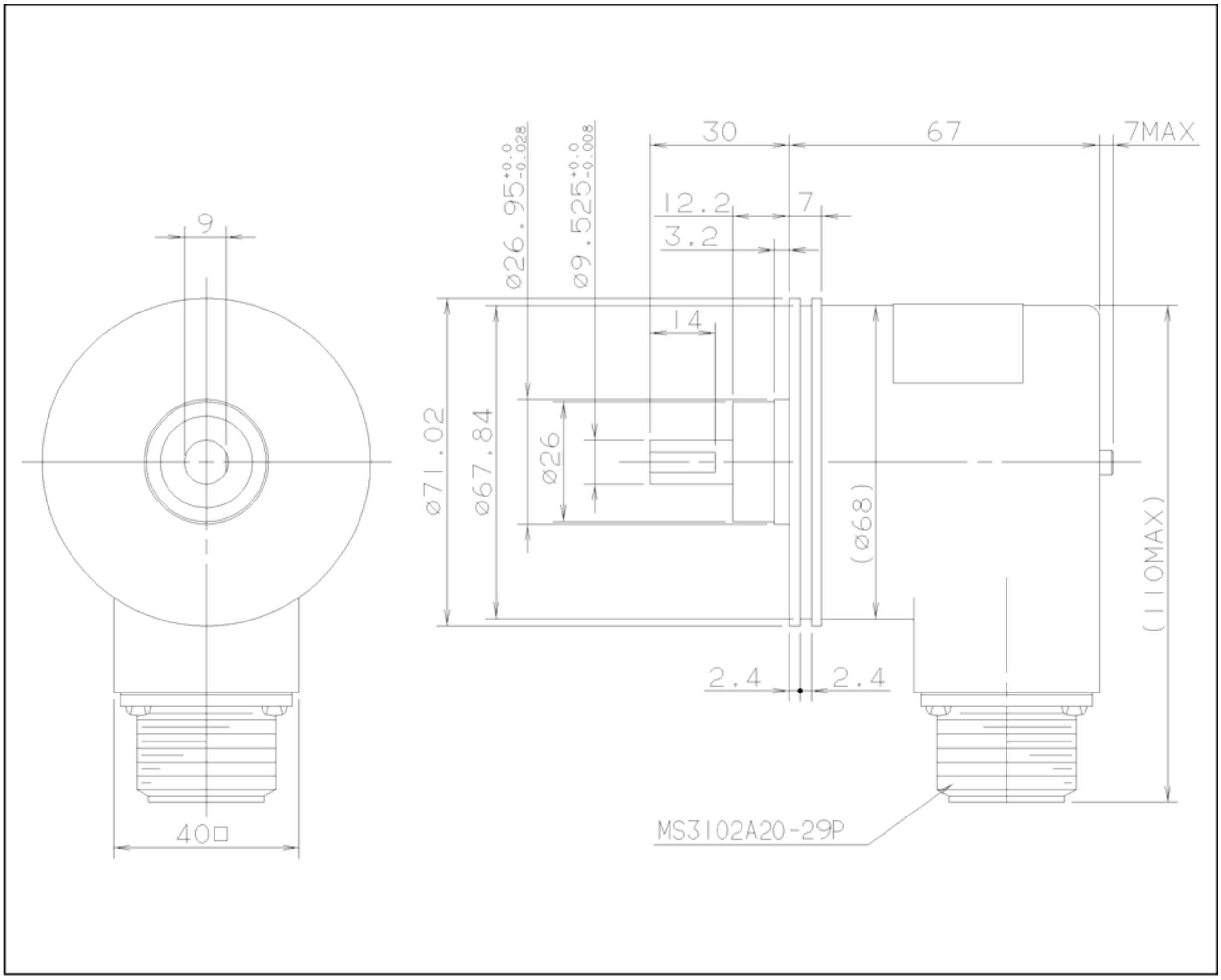
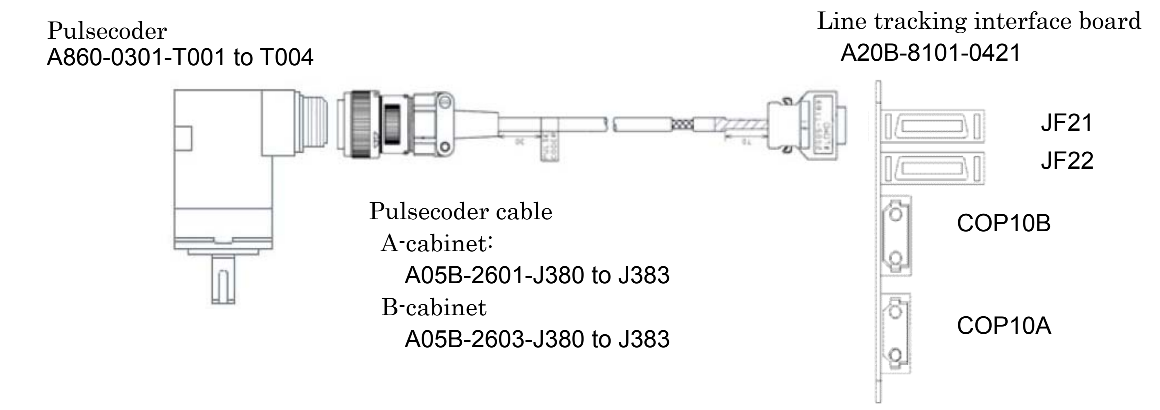
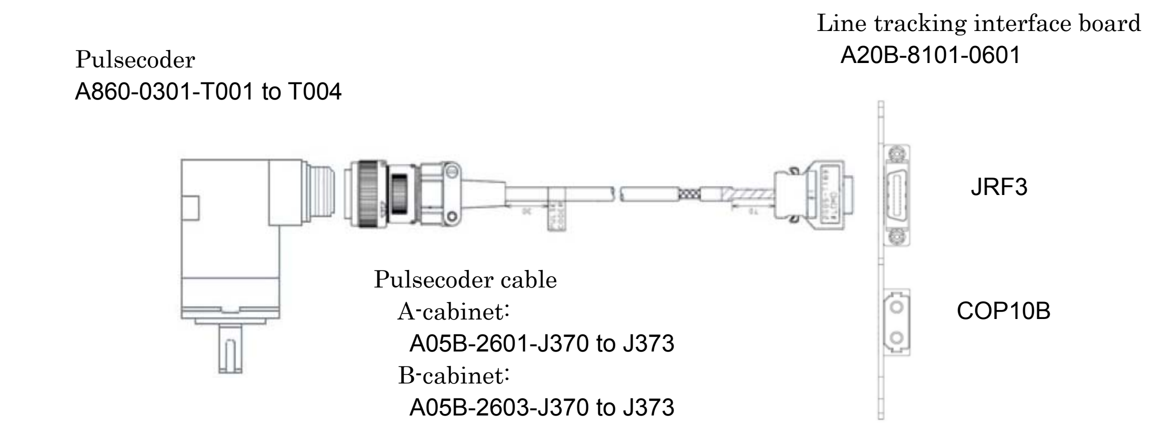


Fig. C.2(c) Pulsecoder dimensions (A860-0301-T001)

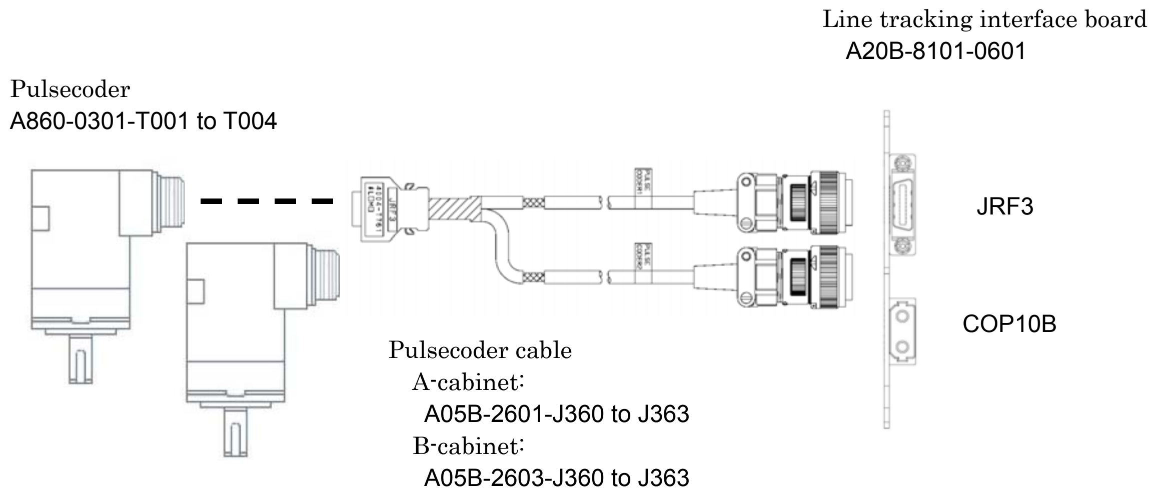
C.3 HOW TO CONNECT



(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.

Fig. C.3(a) Connecting cables (Pulsecoder, A860-0301-T001 to T004)

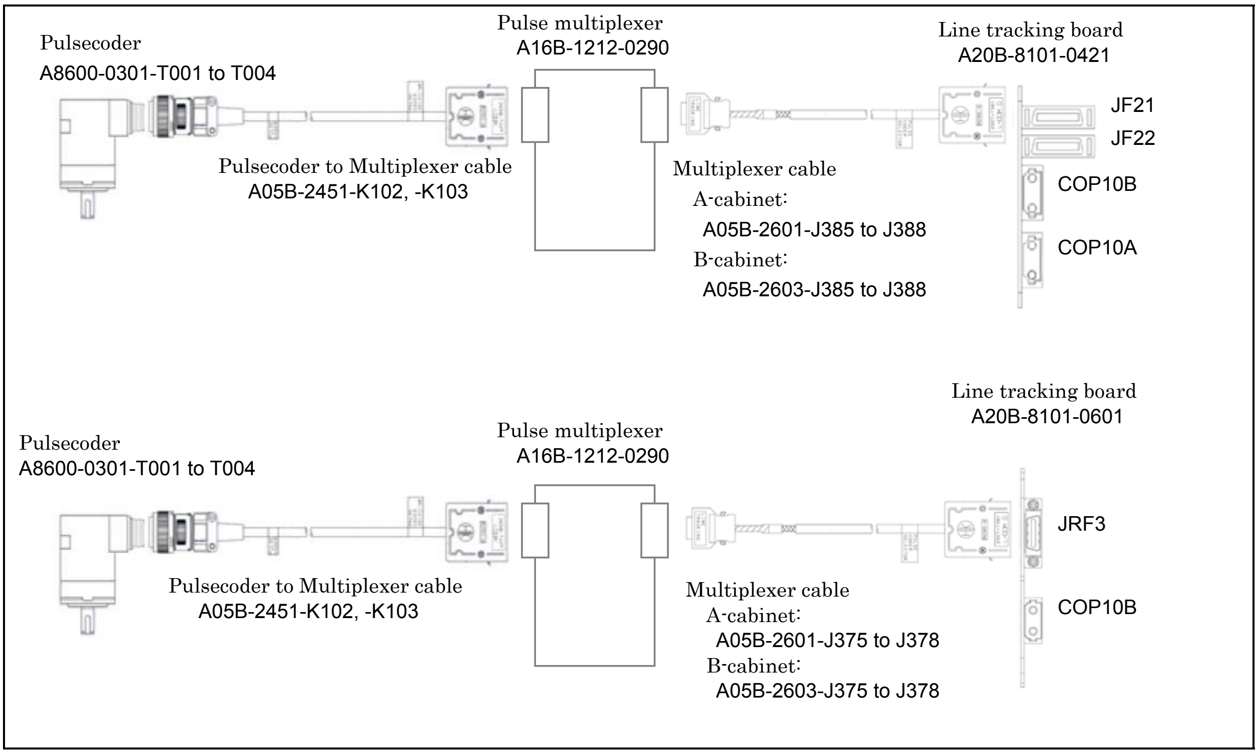


Fig. C.3 (b) Pulse multiplexer and connecting cables 1 (Pulsecoder, A860-0301-T001 to T004)

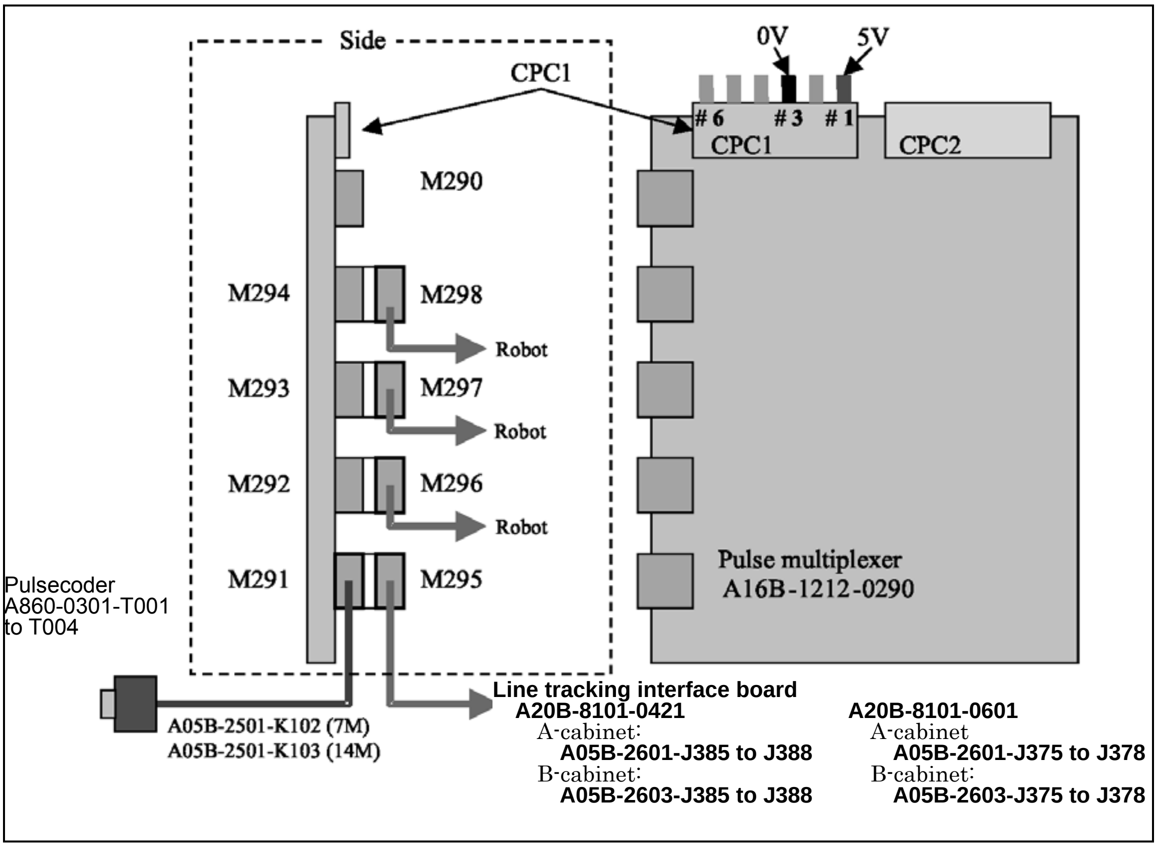


Fig. C.3(c) Connecting cables to the multiplexer 2 (Pulsecoder, A860-0301-T001 to T004)

NOTE

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

INDEX

<T>

<U>

<V>

<W>

REVISION RECORD

Edition	Date	Contents
01	Oct., 2012	

B-83474EN/01

* B - 8 3 4 7 4 E N / 0 1 *