

# ROBOT OPERATIONS

## Part 2

*EET 4144*

**FANUC**  
Robotics



Material Handling



Palletizing and Packing



Material Removal



Arc and Spot Welding



Painting



Dispensing and Assembly



**FANUC**  
Robotics



**CERTIFIED  
EDUCATION  
ROBOT TRAINING**

## Robot Operations



1 – Safety and Cycle Power

2 – Moving a Robot in JOINT and WORLD Jog Modes

**3 – Create and Change Teach Pendant Programs**

4 – Abort, Access, Test and Run Programs

## Teach Pendant Programs Module

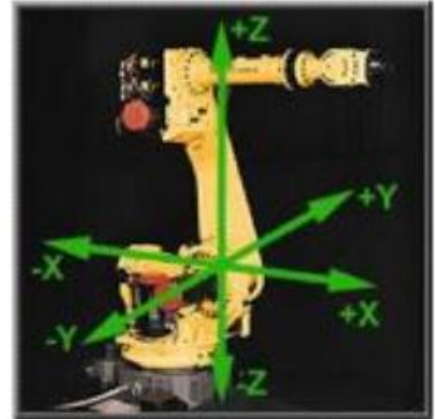
- **One of the key responsibilities operators are faced with is to instruct the robot and other equipment how to move and what to do to perform specific tasks.**
- **In this module you should learn how to create and make changes to a Teach Pendant Program (TPP).**

# Module Objectives

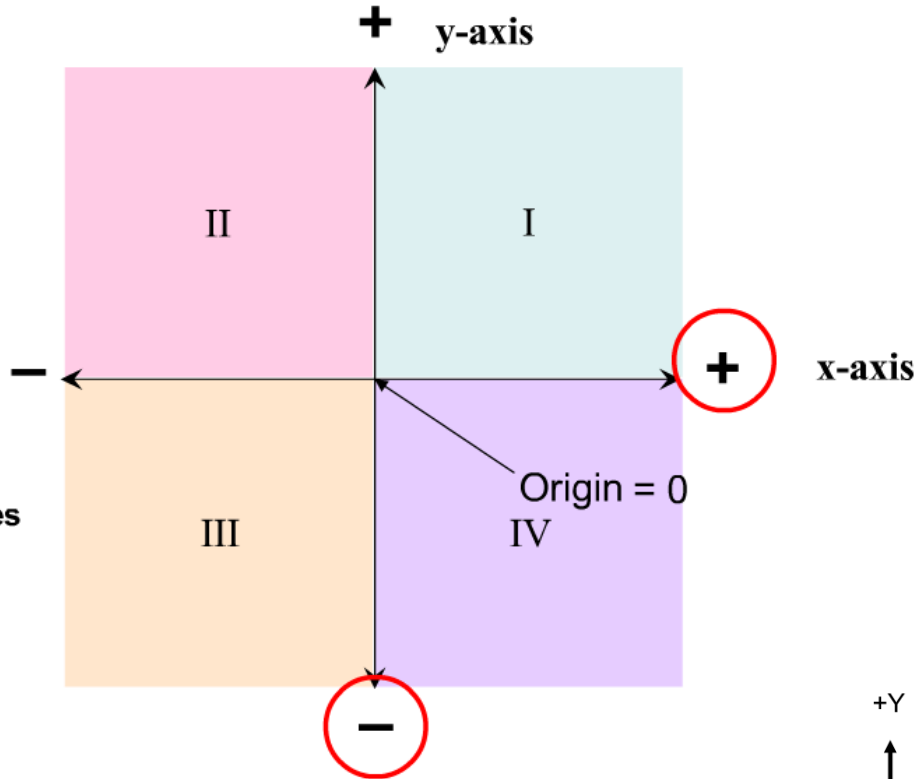
- **After successfully completing this module you should be able to:**
  - Define a point as it relates to robot coordinate systems
  - Describe the Cartesian Coordinate System
  - Describe or demonstrate how to teach one or more points
  - Identify the elements of a motion instruction
  - Re-teach positional data
  - Modify motion instruction components
  - Describe or demonstrate how to access the position detail screen for display of positional information
  - Describe or access the JOINT and WORLD screen display of positional data

# Define a Point

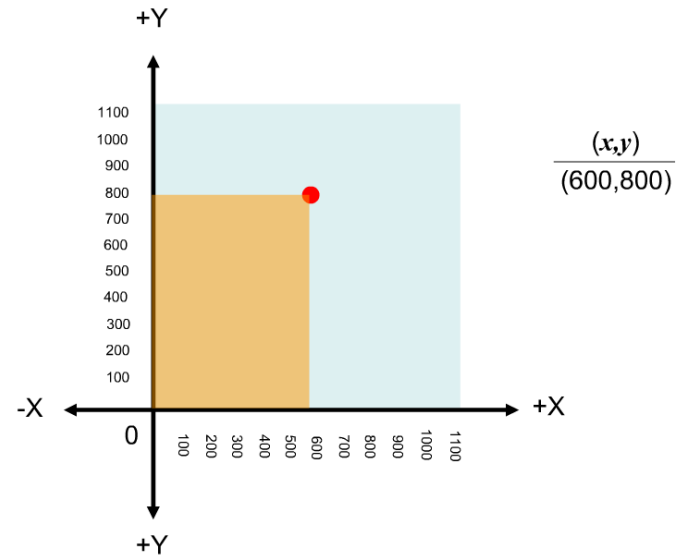
- A basic requirement for creating any Teach Pendant program is for operators to understand what a point is, and to be able to move a robot to a specified point in space in order to carry out a task. Teach Pendant programs commonly consist of movements to and from several points.
- **REMEMBER:** A point is a position in space which is usually defined based on the **WORLD Coordinate System** as a number of millimeters from the robot's origin. A point may also be displayed in **JOINT** with rotational joints being expressed in degrees and linear joints expressed in millimeters.



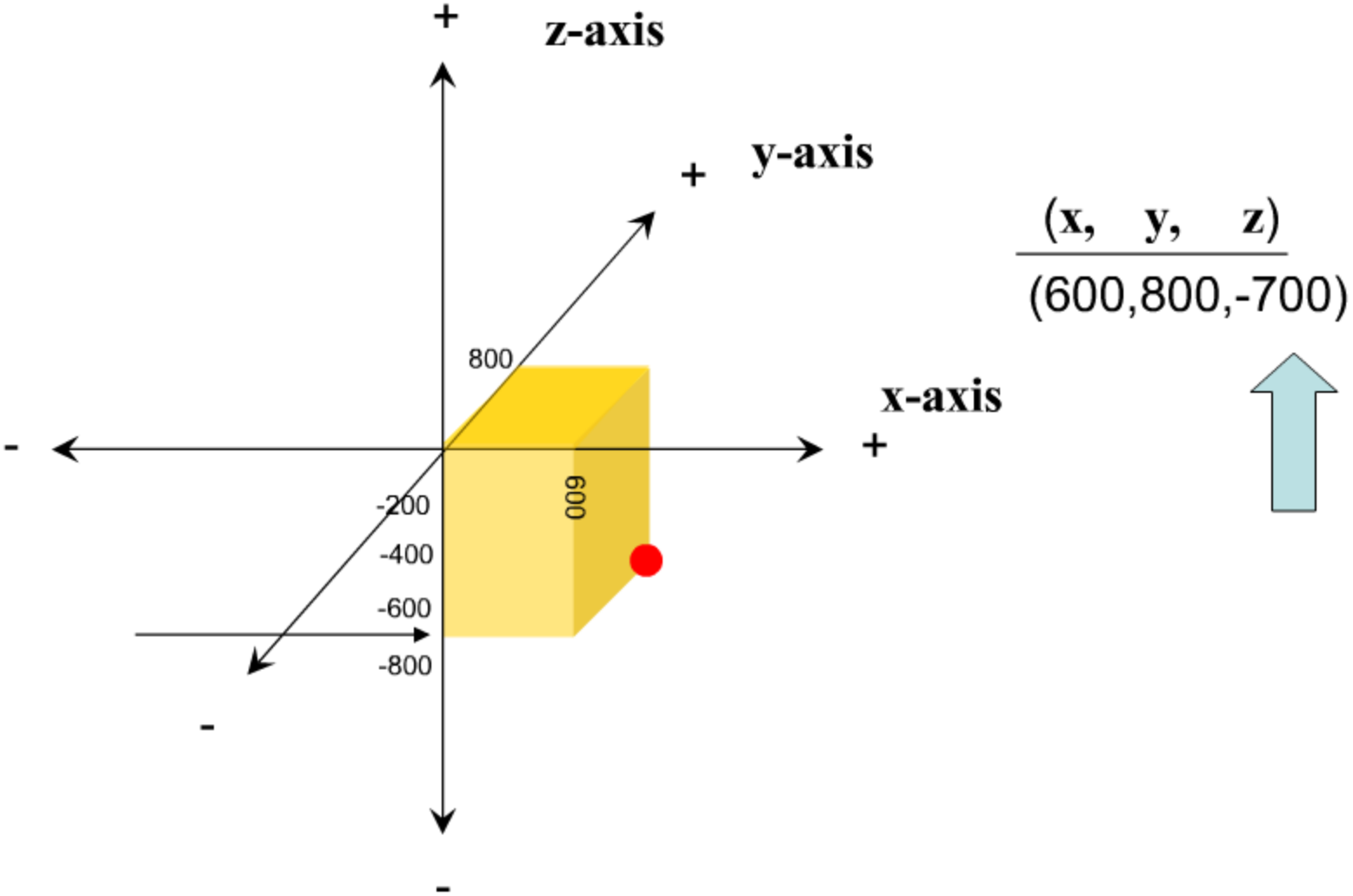
# Two Dimensional Cartesian Coordinate



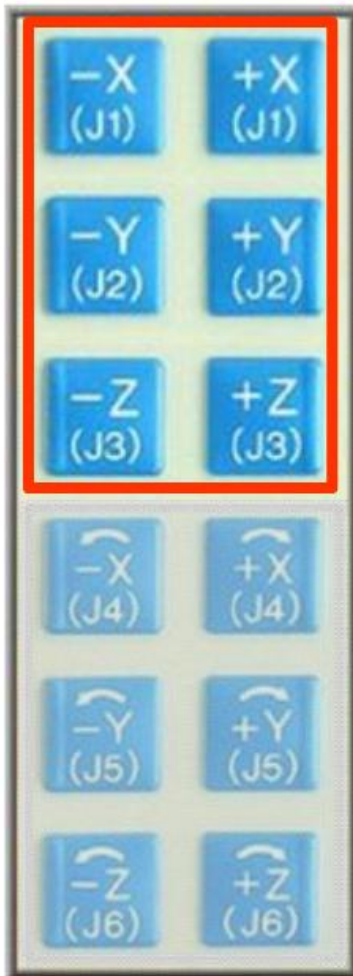
Quadrant	x values	y values
I	>0	>0
II	<0	>0
III	<0	<0
IV	>0	<0



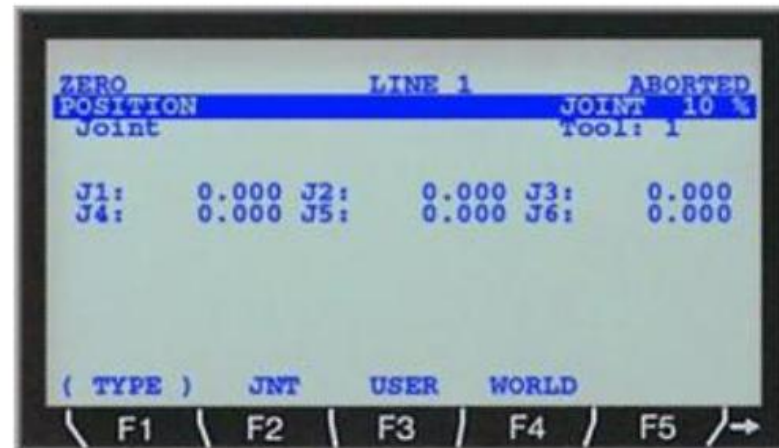
# Three Dimensional Cartesian Coordinate



# Positional Information



For every position that you can manually place a robot into, that position is definable for each axis or joint, and is measured in a positive (+) or negative (-) degree of rotation from the center or the zero position.



[How to find current position of the robot](#)

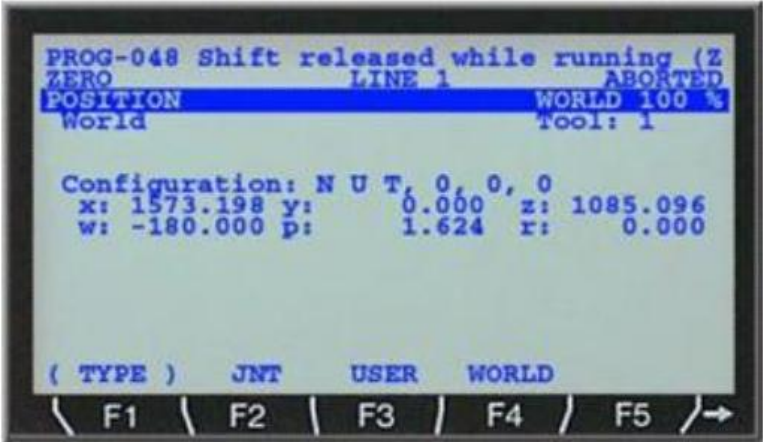
[Jog the robot](#)



# WORLD Screen Display of Positional Information



To observe robot motion based on WORLD jog mode, and desired positive or negative direction of motion, click on the axis of your choice from the listing. Try to determine where the robot will move before making your selection.



[World Jog Mode](#)

# Plan and Create a Program

If you were put in charge of creating the Teach Pendant programs for your organization, what would you do?

#1 Create a Plan

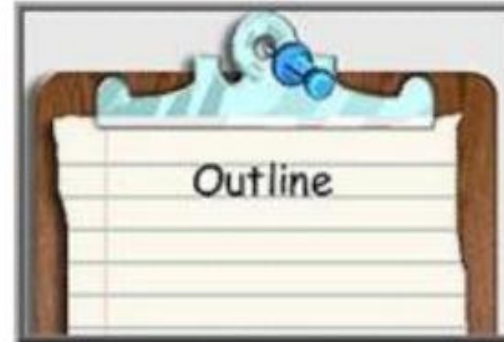


#2 Write the Program



# Plan and Create a Program

## #1 Create a Plan



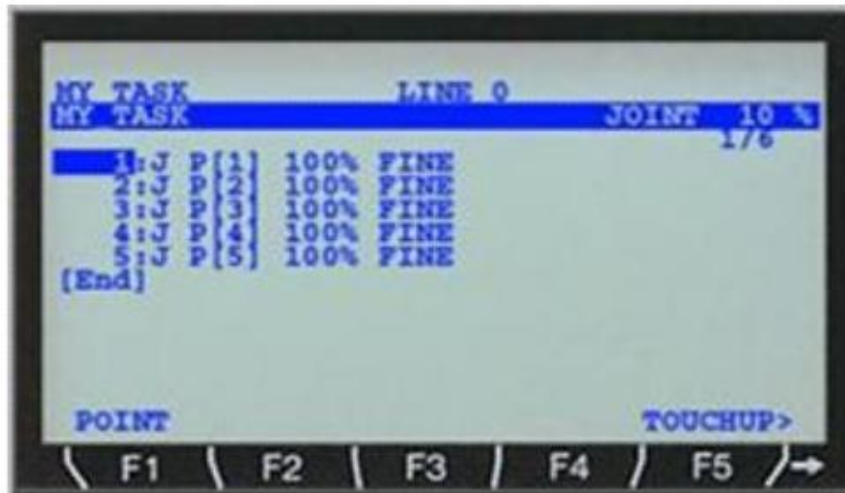
### TASK ANALYSIS:

- Describe the part
- Describe the production requirement
- List all activities to be performed
- Divide activities into subtasks
- Describe all positions or path nodes involved in the process
- Describe the environmental area and plant layout
- Identify environmental constraints

# Plan and Create a Program



- Create lines of instruction
- Tackle subtasks defined during planning
- Name the program
- Add each motion statement or line of program instruction
- Jog or move the robot
- Teach or record positions



[Creating the program](#)

# Define Elements of a Motion Instruction

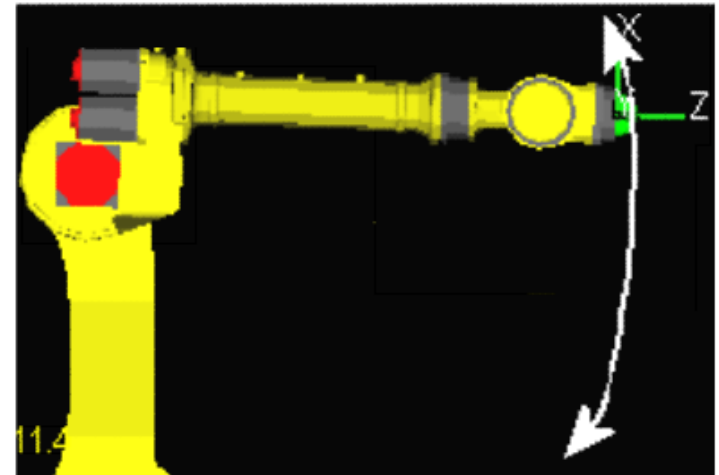
SHIFT + POINT = MOTION INSTRUCTION



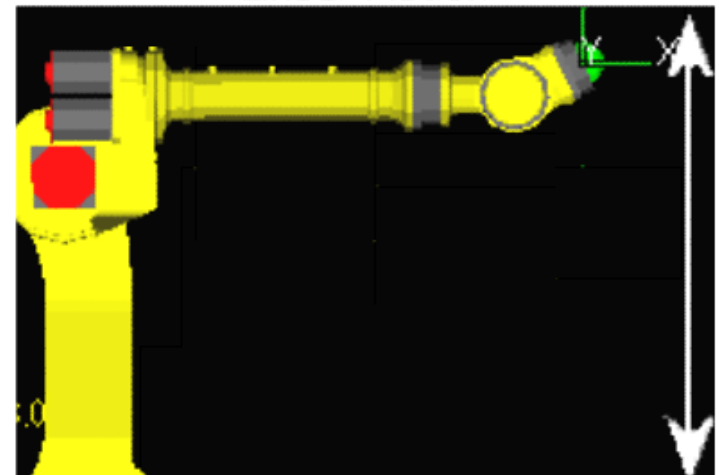
# Define Motion Types

- What is Joint motion? **Joint motion is the type used when it is not important how the robot moves from position to position. It is the fastest and most efficient path from one point to another, however it is not predictable.**
- What is Circular motion? **You use circular motion when the positions must be along the arc of a circle**
- What is Linear motion? **Linear motion is used when the robot must move in a straight line between two points, however it is slower and predictable.**

Joint Motion



Linear Motion



# Define Elements of a Motion Instruction

- **What does motion instruction tell us?**
  - A motion instruction directs the robot to move in a specified way to a specific location in the workcell using a specified speed.

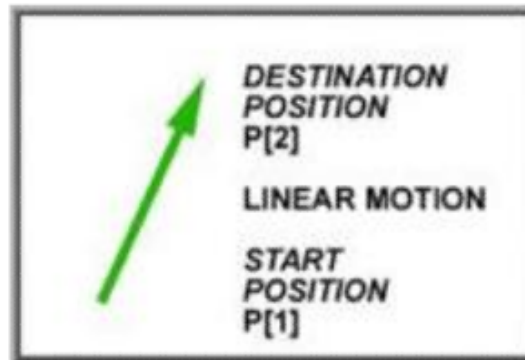
**Motion Type** - tells us how the robot will move from one position to another

**J** P[1] 100% FINE Arc Start [5]

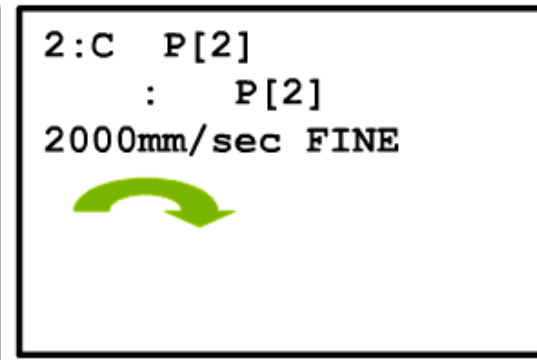
J: Joint motion



L: Linear motion



C: Circular motion



# Define Elements of a Motion Instruction

- **What does motion instruction tell us?**
  - A motion instruction directs the robot to move in a specified way to a specific location in the workcell using a specified speed.

**Positional** Information tells us where the robot moves. This includes the Position Type and the Position number

```
J P[1] 100% FINE Arc Start [5]
```

Position Type:

P: Position

PR: Position Register (predefined and recorded in a specified register)



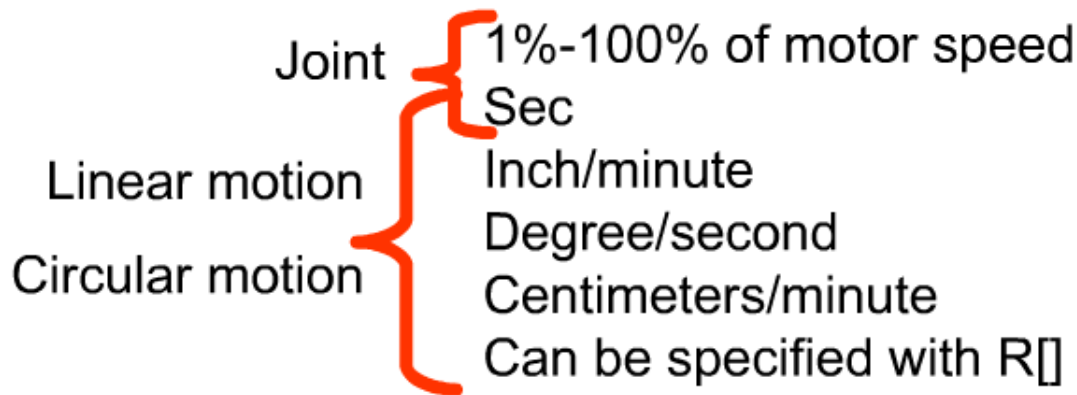
# Define Elements of a Motion Instruction

- **What does motion instruction tell us?**

- A motion instruction directs the robot to move in a specified way to a specific location in the workcell using a specified speed.

**Speed** - indicates how fast the robot moves to a position.

J P[1] **100%** FINE Arc Start [5]



# Define Elements of a Motion Instruction

- **What does motion instruction tell us?**
  - A motion instruction directs the robot to move in a specified way to a specific location in the workcell using a specified speed.

**Termination Type** is how the robot ends the move to the position.

J P[1] 100% **FINE** Arc Start [5]

FINE-Robot will stop at the destination point.

CNT (Continuous) - Robot will decelerate as it approaches the destination position but does not stop.

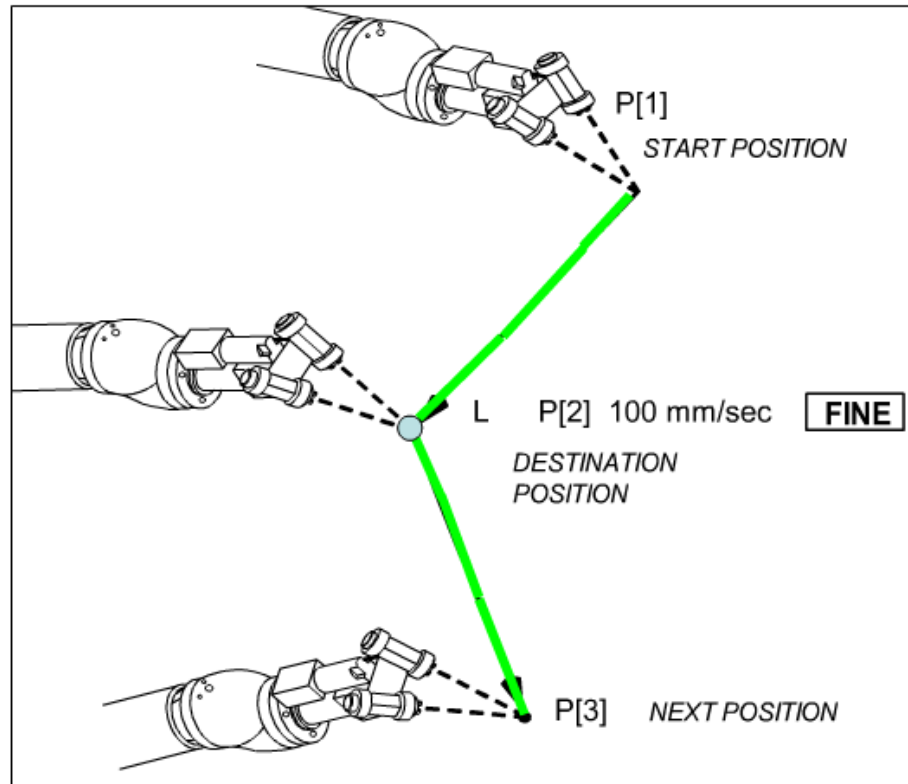
# Define Elements of a Motion Instruction

- **What does motion instruction tell us?**
  - A motion instruction directs the robot to move in a specified way to a specific location in the workcell using a specified speed.

J P[1] 100% FINE **Arc Start [5]**

Motion option refers to additional instructions given to the robot to determine what the robot will do when it reaches its destination. These options are used to provide information to perform specific tasks during robot motion.

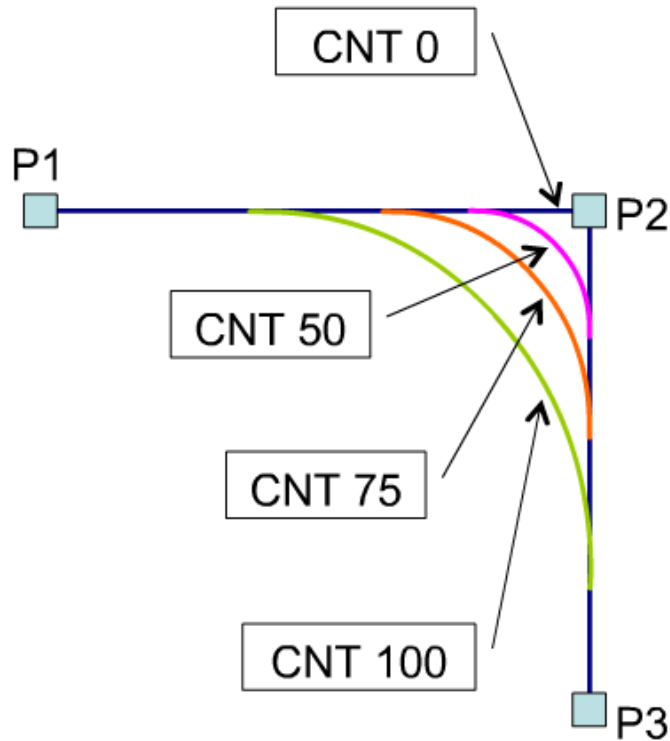
# Robot Motion with Fine Termination Type



Fine Termination

# CNT Termination

- The robot “rounds” at the end of each move



L P[2] 2000 mm/s CNT0  
L P[2] 2000 mm/s CNT100  
L P[2] 2000 mm/s CNT75  
L P[2] 2000 mm/s CNT50

[CNT Termination](#)

# CNT Directional Changes & Speed

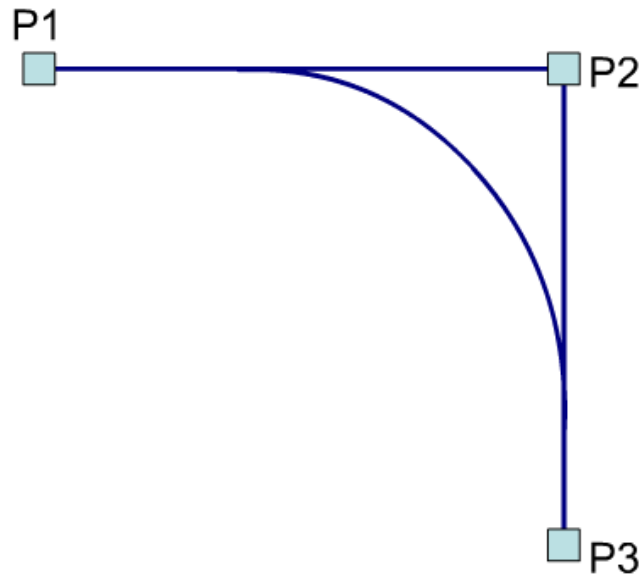
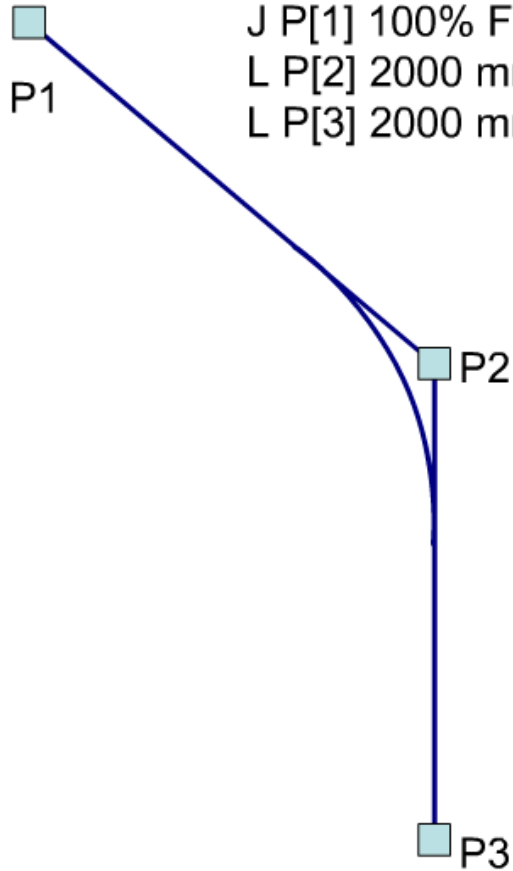
## Assumption:

General Override **Speed:100%** (Production)

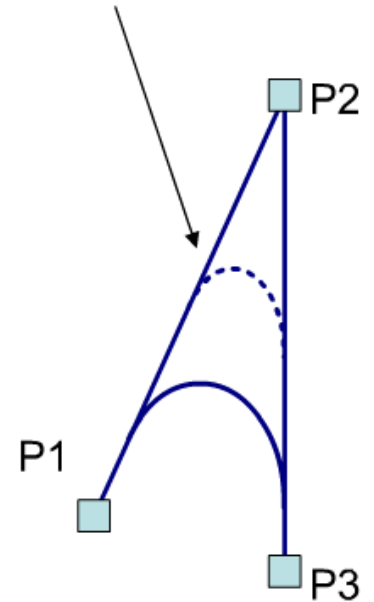
J P[1] 100% FINE

L P[2] 2000 mm/s **CNT100**

L P[3] 2000 mm/s **CNT100**

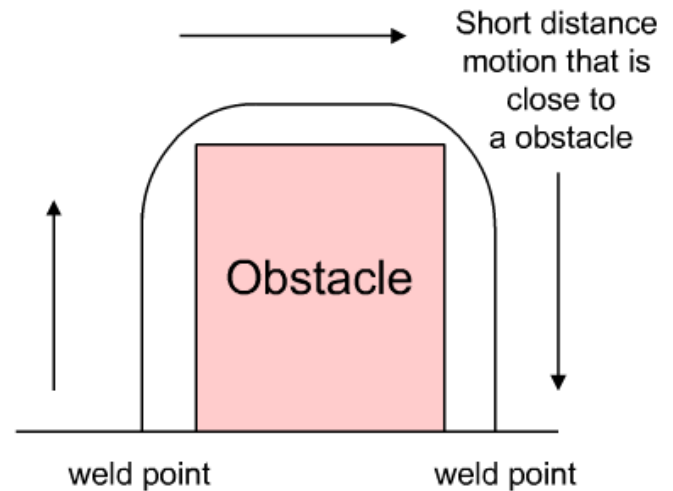
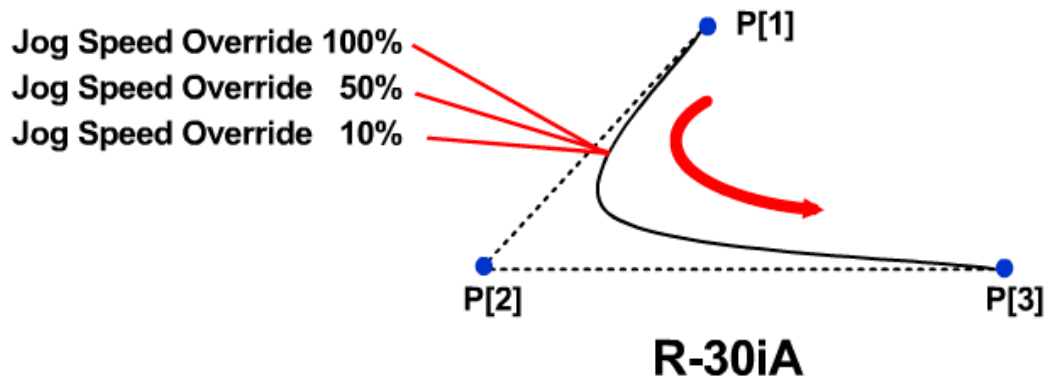


L P[2] **250** mm/s CNT100  
L P[3] **250** mm/s CNT100



# Constant Path for R-30iA Controller

- Constant path regardless of jog speed override setting
- Allows safely verifying a path that is close to an obstacle at low jog speed override
- Teaching becomes easier because path does not change at high jog speed override



# Re-Teach Positional Data

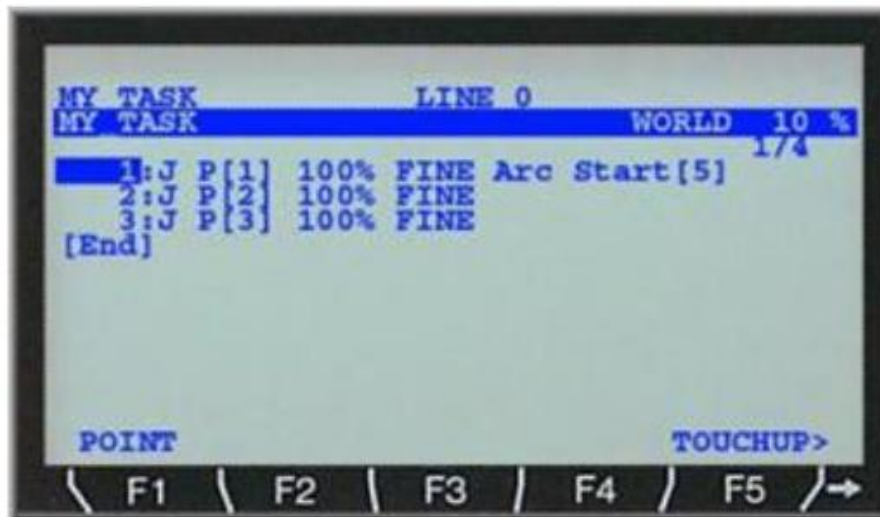
How can I move this robot just one more mm.?

Touching up specific points in a program is sometimes necessary for precision in robot movement. The process of re-teaching positional data is used in these instances. One might compare this process to that of editing a word processing document or editing a videotape to make necessary and critical adjustments. In this process, we will not only learn how simple touching up a point is, but also how to select an existing program before touching up a point.

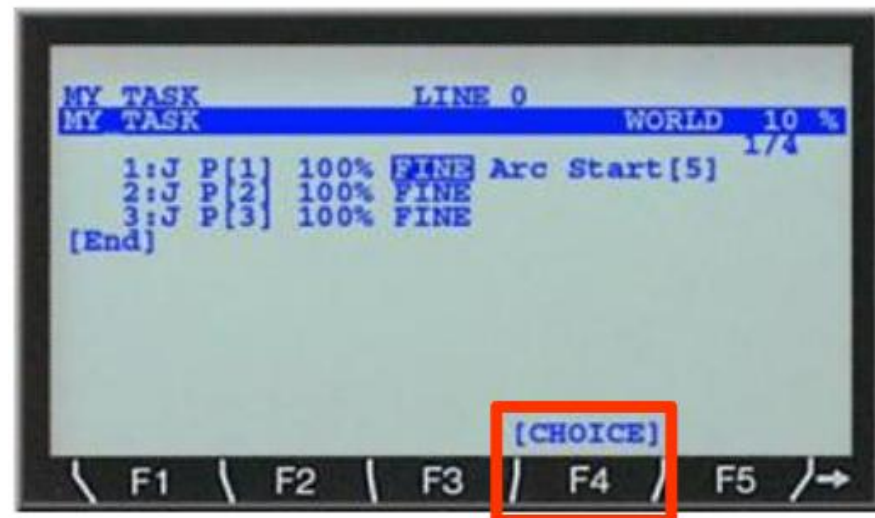
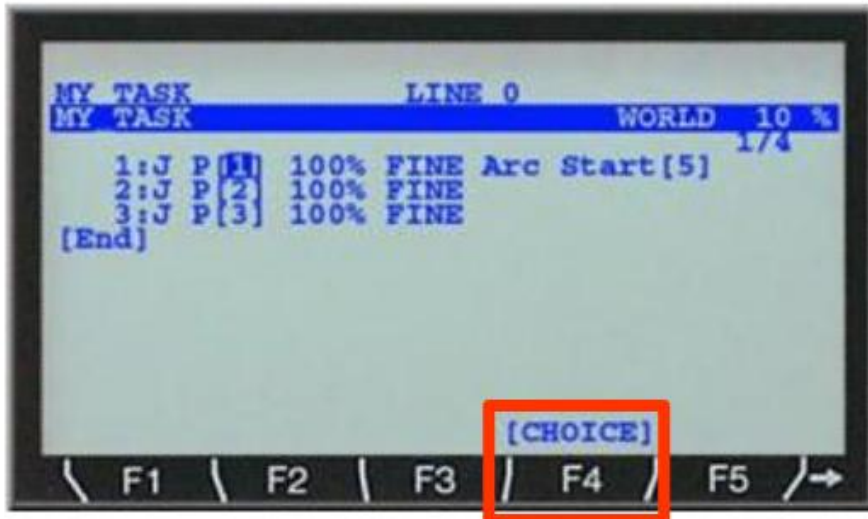
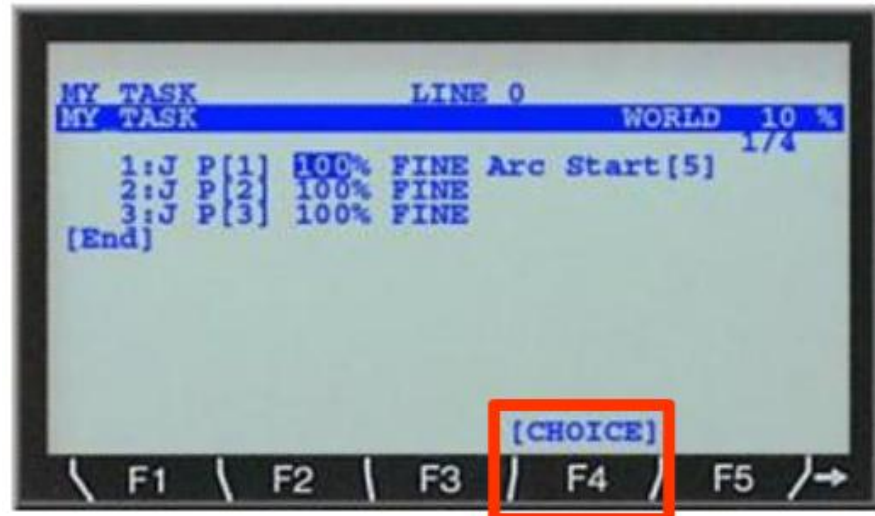
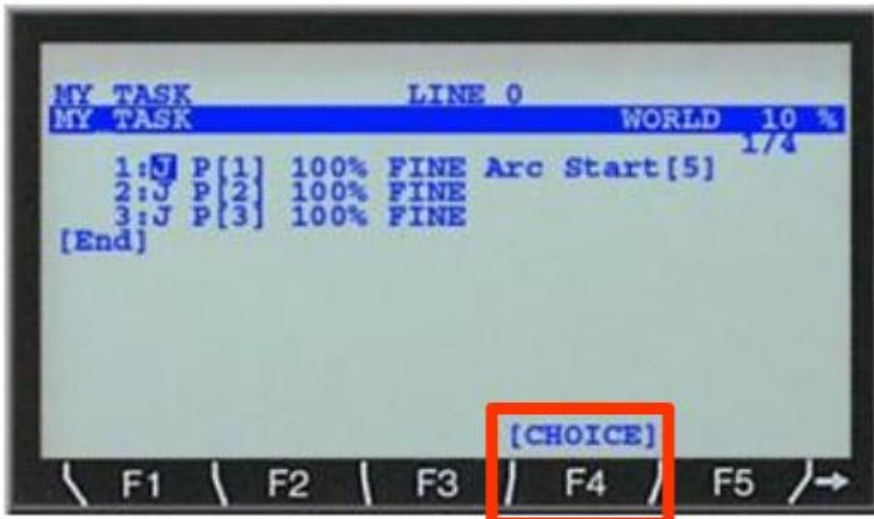




# Modify Motion Instruction Elements



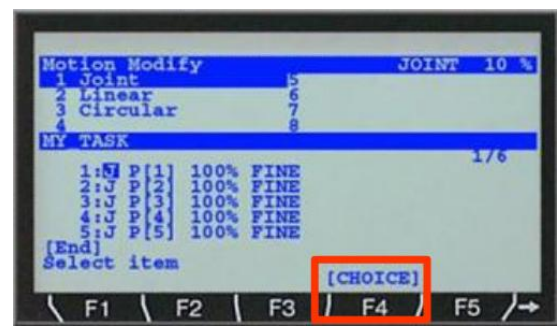
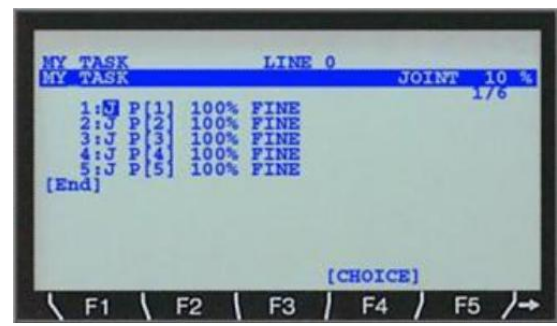
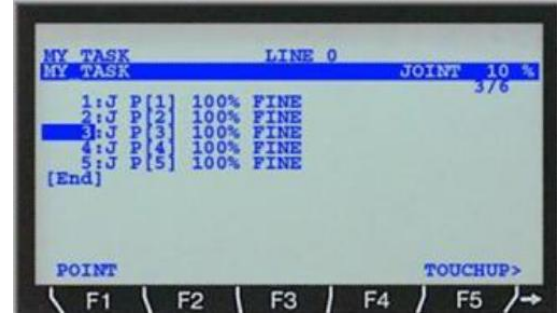
Previously we determined that teaching a point occurs by holding the SHIFT key and pressing POINT simultaneously. This action creates a program line called a Motion Instruction, and is based on default values set within the software. Sometimes it is necessary to make adjustments or changes to a specific element of a Motion Instruction for one reason or another. In cases like these, access the individual field on the Motion Instruction line to make the change using the soft key F4.



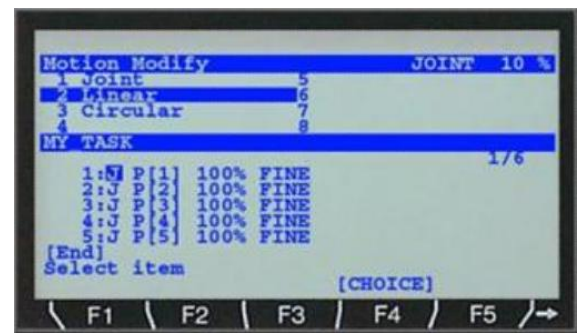
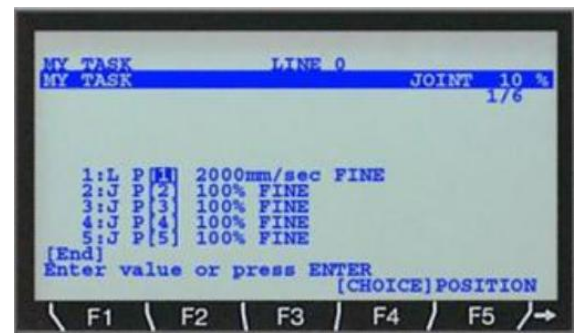
# Modify Motion Instruction Elements

The following are the steps needed when an element of a motion instruction needs to be changed.

1. Move the cursor to the required motion statement using the up and down arrow keys
2. Move the cursor to the element of the motion statement requiring change using the right and left arrow keys
3. Enter numeric value changes using the Teach Pendant numeric keypad, OR
4. Press CHOICE for available options
5. Highlight selection from options and press ENTER, or enter the numeric value that represents the option
6. Turn the Teach Pendant to the OFF position
7. Release the DEADMAN switch



## Touching Up Program



# Review

## Modify Motion Instruction Elements

Based on your recall, can you match each of the descriptions to the single element within this motion statement?

J P[1] 100% FINE Arc Start [5]

- A. [Motion Type](#)
- B. [Motion Option](#)
- C. [Termination Type](#)
- D. [Speed](#)
- E. [Positional Information](#)



## **Robot Operations**

1 – Safety and Cycle Power

2 – Moving a Robot in JOINT and WORLD Jog Modes

3 – Create and Change Teach Pendant Programs

**4 – Abort, Access, Test and Run Programs**

## **Abort, Access, Test and Run Programs**

- **Once programs have been created, it is important that operators are able to access the written programs, test, and then run the program in production successfully. In this module you should learn how to access, test, run programs in production, and abort programs.**

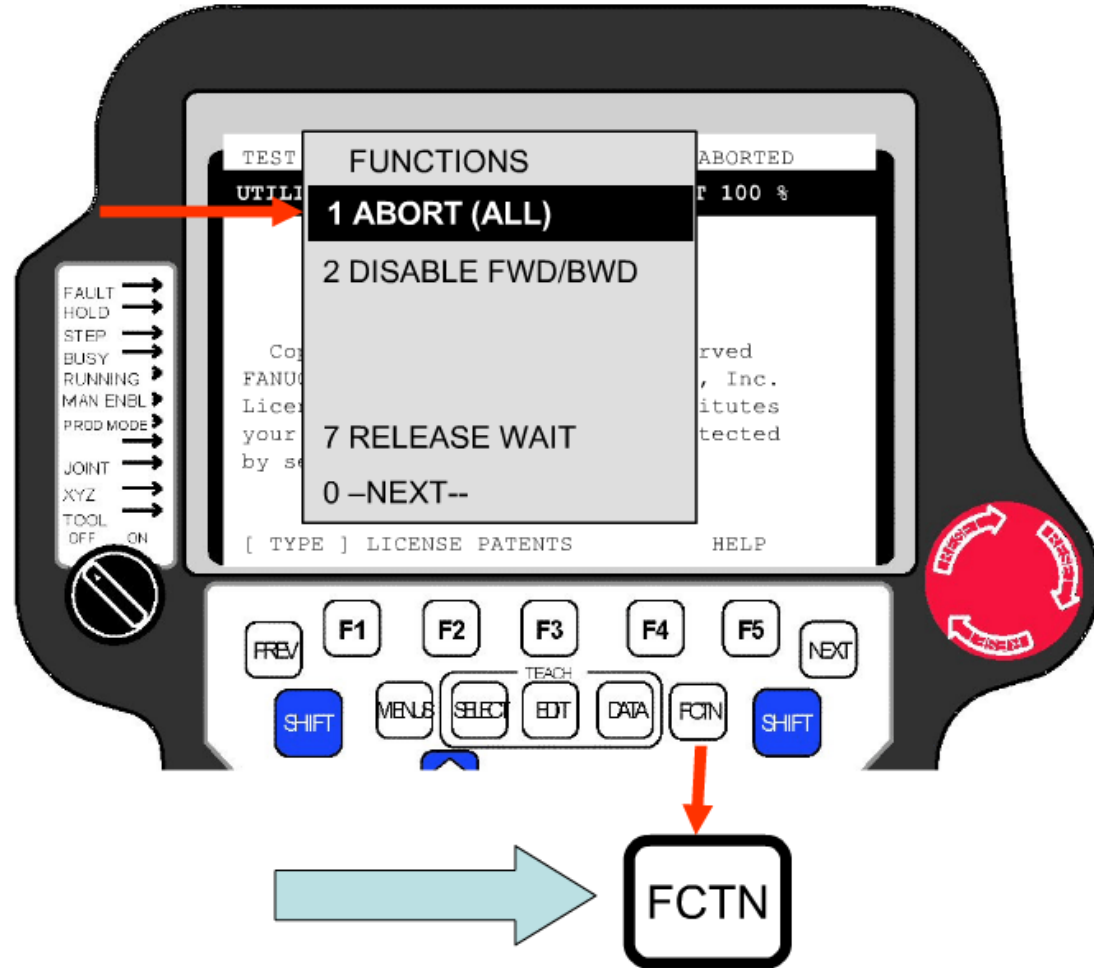
# Module Objective

- **After successfully completing this module you should be able to:**
  - Explain under which circumstances you would abort a program
  - Describe or demonstrate how to abort a program using the Teach Pendant
  - Describe or demonstrate how to select an existing program using the Teach Pendant Explain the purpose for single-step testing of robot programs
  - Describe or demonstrate how to single-step test a program using the Teach Pendant
  - Describe or demonstrate how to continuous test a program using a Teach Pendant or Standard Operator Panel (SOP)
  - Describe or demonstrate how to execute a program in a production environment
  - Describe or demonstrate how to pause a program when step testing or continuous testing a program in a production environment

# Purpose for Completely Aborting a Program

The benefits to aborting a program before selecting another program or continuing with the current program selected are that aborting:

1. **Ends the current program that may have been selected and cancels any motion pending or in progress.**
2. **Presents the operator an opportunity to return to the beginning of the current program to begin again.**
3. **Allows the operator to select another program for testing or production purposes.**





# Purpose for Single Step Testing a Program

- It is critical that the integrity of a program be tested before ever placing a newly created program into production. Once you have aborted all programs and selected the program, single-step test your program.
- Single-step testing allows operators to run each individual program line of instruction, one at a time, in the forward and backward direction. Step testing a program in the backward direction should be performed with caution.
- Step testing, in a sense, is a validation that the program makes the robot perform in a manner consistent with the intended application or task.



**CAUTION: Be careful when step-testing in backward direction!!**

[Single Step Testing](#)

# Purpose for Continuously Testing a Program

- **As further verification for the integrity of a program, continuous testing of a program is advised.**
- **Continuous testing a program involves running the program from beginning to end without stopping. Our recommendation is to continuously test a program:**
  - ONLY after you have successfully single step tested the program.
  - After aborting all the programs, making certain that Teach Pendant program will begin to be tested at the start of the program.

[Continuously Testing](#)

# Continuous Testing

Did you know that there are two ways to continuously test a program?

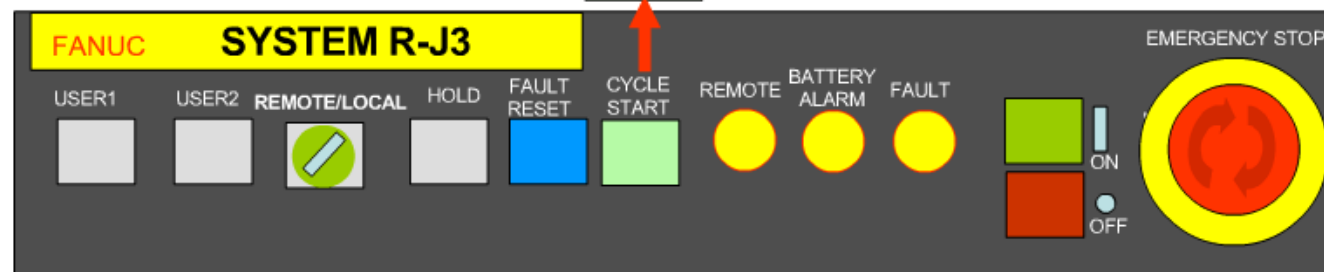
Teach Pendant



CYCLE START

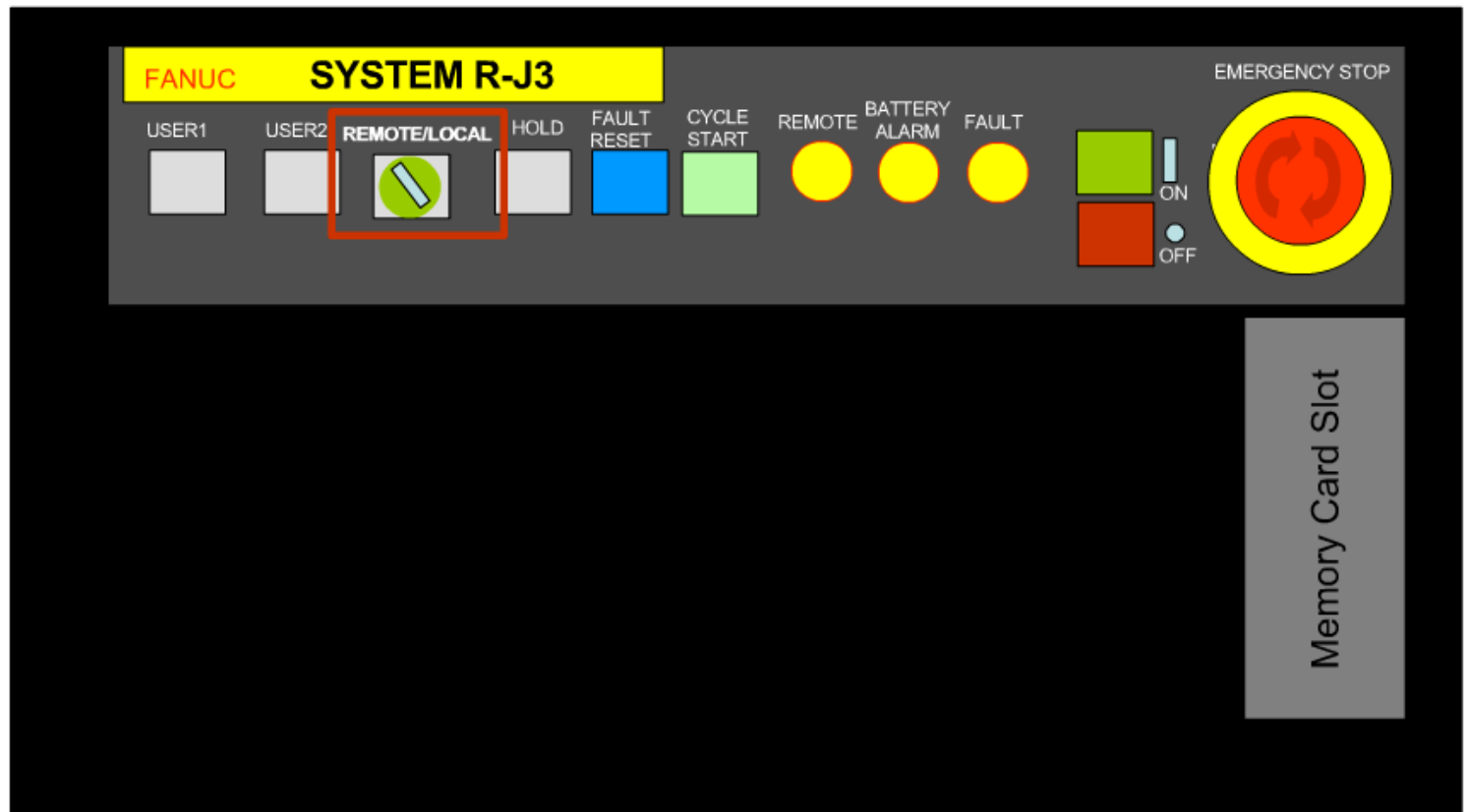


Standard Operator Panel



# Remote/Local Mode

- **Local Mode** is designed for application development and testing.
- **Remote Mode** is used when running production.



# Remote/Local Mode

- **Local Mode** is designed for application development and testing.
- **Remote Mode** is used when running production.

```

LINE 0      AUTO ABORTED
UTILITIES Hints          JOINT 100 %
  MENUS (N. A.)
  1 SELECT
  2 EDIT
  3 DATA
  4 STATUS rights Reserved
  5 POSITION cs America, Inc.
  6 SYSTEM use constitutes
  7 USER2 product protected
  8 .
  9
  0 -NEXT --
[ TYPE ] LICENSE PATENTS          HELP
  
```

```

SYSTEM Clock          JOINT 100 %
Clock Display

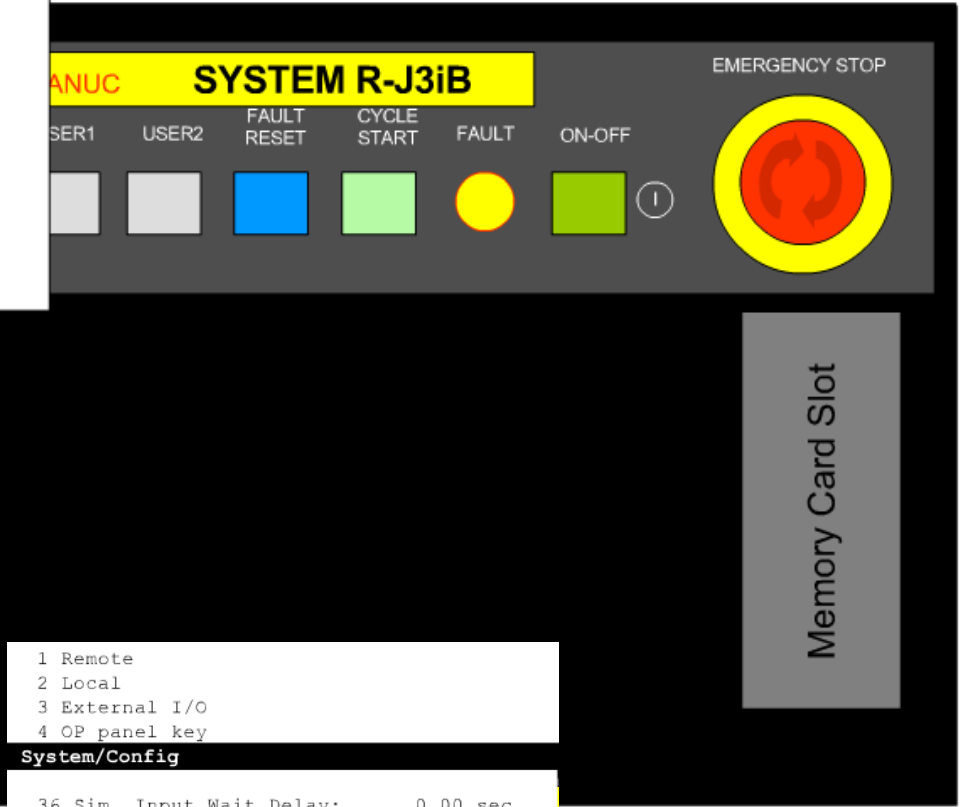
DATE          05/12/21

1 Clock          08:18:22
2 Variables
3 Master/Cal
4 OT Release
5 Axis Limits
6 Config
7 Motion function
[ TYPE ]          ADJUST
  
```

```

System/Config          JOINT 100 %
39/41
33 Signal to set in T2 mode DO[ 0]
34 Signal to set if E-STOP DO[ 0]
35 Set if INPUT SIMULATED DO[ 0]
36 Sim. Input Wait Delay: 0.00 sec
37 Set if Sim. Skip Enabled DO[ 0]
38 Hand broken : <*GROUPS*>
39 Remote/Local setup: Remote
40 External I/O(ON:Remote):RI [ 1]
41 UOP auto assignment: None

[TYPE]          [CHOICE]
  
```



```

1 Remote
2 Local
3 External I/O
4 OP panel key
  
```

```

System/Config
36 Sim. Input Wait Delay: 0.00 sec
37 Set if Sim. Skip Enabled DO[ 0]
38 Hand broken : <*GROUPS*>
39 Remote/Local setup: Remote
40 External I/O(ON:Remote):RI [ 1]
41 UOP auto assignment: None

[ TYPE ]          [ CHOICE ]
  
```

# Continuous Testing

What difference did you discover between single step testing and continuous testing of a program?

## SINGLE STEP TESTING

- Accomplished using only the Teach Pendant.
- Press the STEP key to enable single step testing and illuminate the Status Indicator.
- Ability to test individual program lines of instruction in both the forward direction and backward direction (with caution).

VS.

## CONTINUOUS TESTING

- Accomplished using either the Teach Pendant or from the Standard Operator Panel (SOP) if the external key for REMOTE/LOCAL switch has been turned to the LOCAL mode for R-J3 controllers (R-J3iB is controlled through the software configuration settings from the Teach Pendant).
- Press the STEP key to disable single step testing if the STEP status indicator is on.
- Ability to test a program in the forward direction only.

# Stopping a Program

There are times when testing programs or running programs in production, that stopping robot motion is necessary. It is critical that operators know how to stop a robot when they must.

### Stop the Robot when in production

