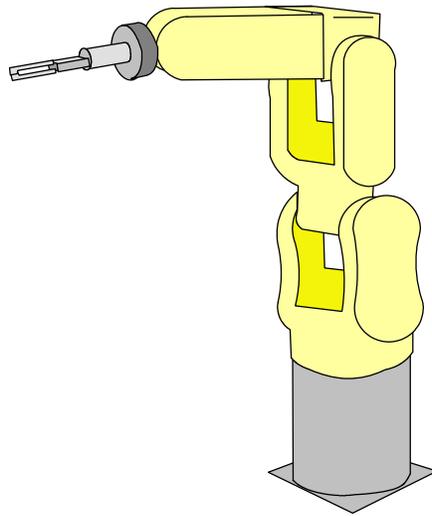


# Fanuc LR Mate 200iD/4S

## GUIDE



CHAP 1 - INTRO .....	3
1.1 Introduction.....	3
1.2 System Overview.....	3
1.3 Mechanical Unit.....	3
1.4 Servo Motors .....	3
1.5 Operator Panel .....	4
1.6 Safety .....	5
1.7 Power Up .....	5
1.8 Robo-Guide Software .....	5
CHAP 2 - TEACH PENDANT.....	6
2.1 Intro.....	6
2.2 Basic Control Using iPendant.....	6
2.3 Dead man Switch.....	7
2.4 Menus .....	7
2.5 Split Screens .....	8
CHAP 3 - COORDINATE FRAMES.....	8
3.1 Intro.....	8
3.2 Joint Coordinates .....	8
3.3 World Coordinates.....	8
3.4 Other Frames .....	9
CHAP 4 - JOGGING .....	9
4.1 Intro.....	9
4.2 Jog Speed.....	9
CHAP 5 - BASIC PROGRAMMING .....	10
5.1 Intro.....	10
5.2 Abort Previous Programs.....	10
5.3 Create a Program .....	10
5.4 Program Soft Key Menu.....	11
5.5 Important Program Functions .....	11
5.6 Teach Points.....	11
5.7 Motion Instructions.....	12
5.8 Touch Up .....	14

5.9	Circular Motion .....	14
5.10	Loops .....	15
5.11	Edit Commands .....	15
5.12	Testing Programs .....	15
5.13	Running programs .....	16
5.14	Running a program in AUTO .....	16
5.15	Open a Program .....	16
5.16	Existing Programs .....	16
5.17	Miscellaneous .....	17
CHAP 6	- INPUT OUTPUT .....	17
6.1	Intro.....	17
6.2	Gripper Control.....	17
6.3	Laser Pointer.....	17
6.4	Camera Light Control.....	17
CHAP 7	- FILE MANAGEMENT & DATA .....	18
7.1	Intro.....	18
7.2	File Navigation .....	18
7.3	Registers .....	18
CHAP 8	- SETUP & MAINTENANCE.....	19
8.1	Axis Limits .....	19
8.2	Fault and Alarms.....	20
8.3	Pendant Settings.....	20
8.4	Mastering & Batteries.....	20
8.5	Singularity.....	21
8.6	Maintenance Reminder .....	22
8.7	Lubrication.....	22
8.8	Other .....	22
CHAP 9	- xxx.....	22
9.1	Section A .....	22
9.2	Section B.....	22

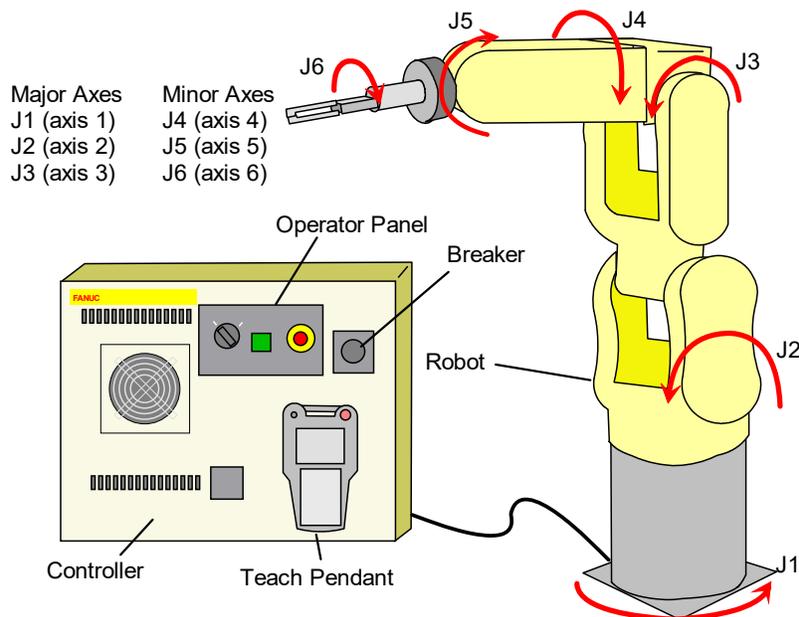
# CHAP 1 - INTRO

## 1.1 Introduction

The Fanuc LR Mate 200iD/4S is a **6-DOF** (degree-of-freedom) robot arm system. The robot is mounted in a cart and comes with a vision system, electronic safety fence, and a control pendant (iPendant). The iPendant be referred to simply as the "pendant" in this document.

## 1.2 System Overview

The robot system consists of the robot (mechanical unit), controller, pendant, and the FANUC software.



## 1.3 Mechanical Unit

The robot is a series of mechanical links driven by **AC servo motors**. The area between the links is called a JOINT or axis. The Fanuc robot has 3 major axes (J1, J2, J3) and 3 minor axes (J4, J5, J6). The first 3-DOF position the wrist in 3D space. The last 3-DOF orient the wrist about the x, y, and z-axes. The axes are numbered in the figure.

## 1.4 Servo Motors

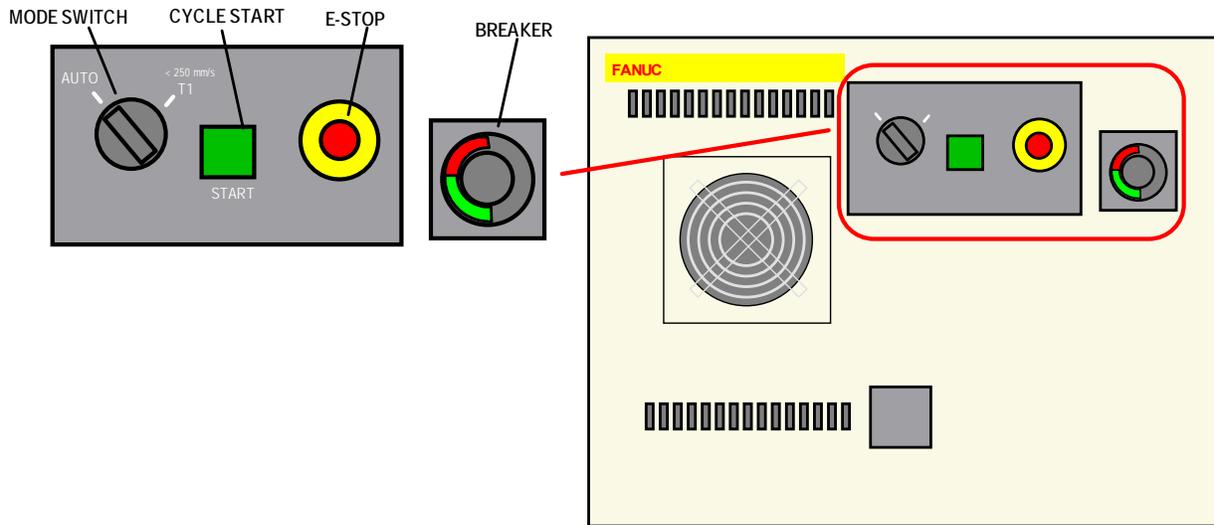
Each axis is driven by a Fanuc **AC servo motor**. The motors are driven by a pulse width modulator (PWM) signal delivered by the servo amplifier. The motor assembly comes as an integrated unit and includes the motor, the brake system, and the SPC (**serial pulse coder**) (basically an encoder).

**Refer to the CT GUIDE** on how encoders work. Encoder pulse counts encode how much the motor shaft has turned. Encoder pulse counts are stored in robot memory when the robot is powered down. They are also stored in each SPC and maintained by batteries found at the base of the robot.

Electrical energy RELEASES the brakes. When electricity is removed the brakes are applied. It is done this way as a safety feature.

## 1.5 Operator Panel

The operator panel is a box located on the bottom of the robot cart. This enclosure contains everything the robot needs to operate, including the processor, electronics, motor amplifiers, motor controller, and more. The front of the enclosure has the operator panel. This is where a user powers the robot on and off. It also has an emergency stop (E-stop) button.



### 1.5.1 Mode Select Switch

There are two basic modes of operation for the robot: (1) TRAINING, and (2) AUTOMATIC.

In training mode (T1), the robot is primarily controlled by the pendant. This mode is used to develop robot programs. In AUTO mode, the robot runs a pre-selected program automatically (no teach pendant).

Select mode with deadman switch RELEASED.

AUTO = AUTOMATIC MODE

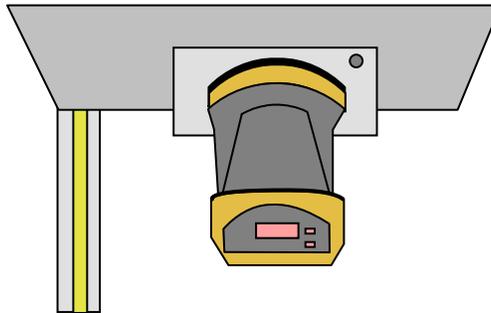
No Pendant	OFF
Safety fence	ACTIVE
GREEN button	Starts the program in auto mode

In AUTO mode the robot will run a pre-selected program. The iPendant is not needed. However, one can still e-stop the program with the pendant, and one can still alter movement speed with the pendant.

T1 - TRAINING mode.

In this mode movement speed is restricted to  $< 250$  mm/s  
Can only run programs with iPendant activated and deadman switch CENTERED.  
Safety fence is BYPASSED.

## 1.6 Safety



The Fanuc robot system comes with a **KEYENCE SAFETY FENCE** system. The electronic scanner system is housed UNDER the robot platform. It scans and collects distance data for any object in its field. This data is sent to the robot controller. In automatic mode, if a person gets too close then the robot will stop moving. Before powering up the robot be sure that no obstacles are in the robot's workspace.

### 1.6.1 Dual Check Safety (DCS)

Fanuc's DCS position/speed check features check the speed and position data of motors with 2 independent CPU's in the robot's controller. These functions can detect position and speed errors immediately and shut down the motor power by 2 independent channels.

## 1.7 Power Up

Here are the basic steps in powering up the robot.

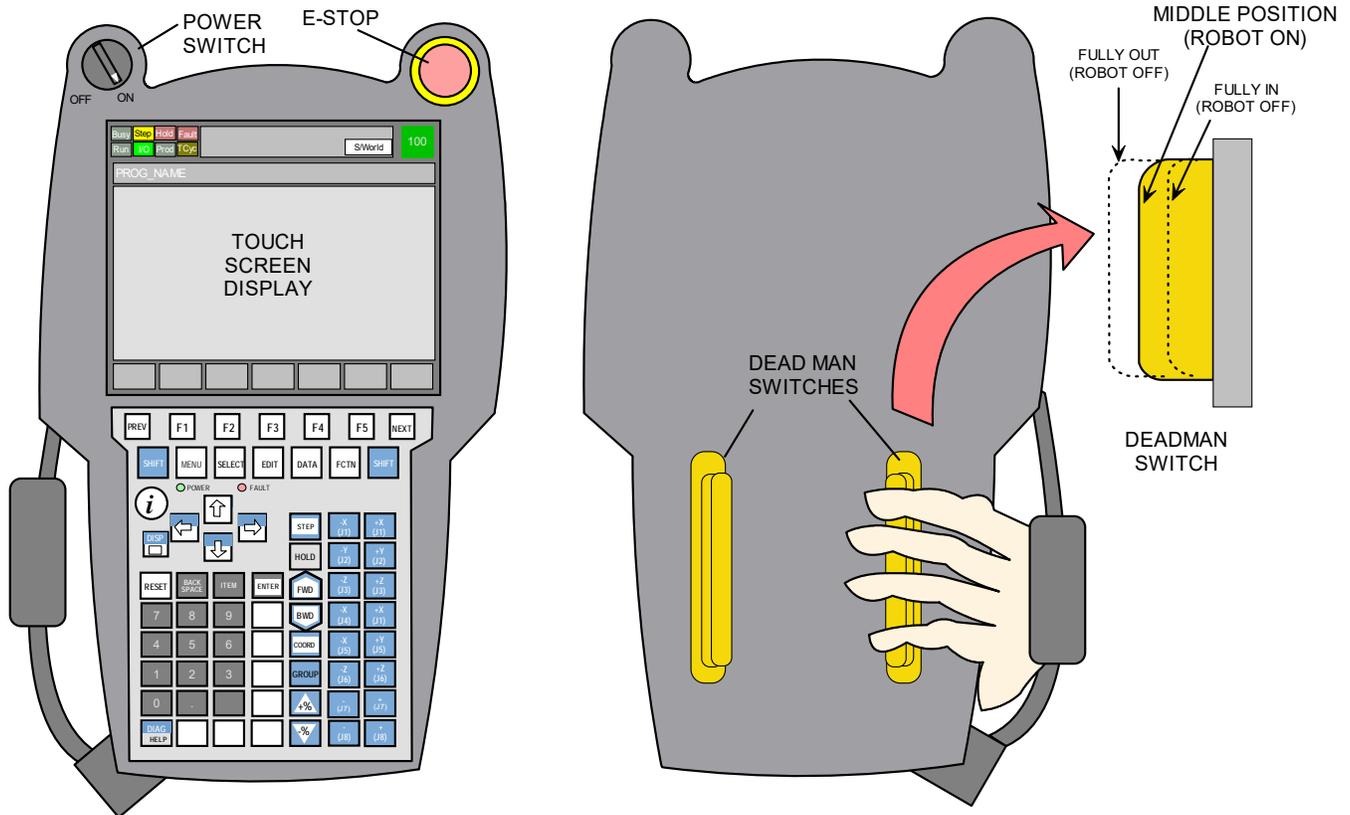
1. Before powering up the robot, ensure that no obstacles are in the robot's workspace.
2. Turn breaker switch clockwise (from 3 to 6 o'clock) on control panel. Allow the system to boot up.
3. Turn on the Pendant.
4. Allow system to boot up.
5. Address any faults (required before robot can be operated)

## 1.8 Robo-Guide Software

The Fanuc Robo-Guide software is a SIMULATOR software.  
The software has a virtual robot and virtual teach pendant.  
You can program the virtual robot in the software and run the virtual robot.

Unfortunately SAC was not approved to obtain this software.

# CHAP 2 - TEACH PENDANT



## 2.1 Intro

The teach pendant (iPendant or just "pendant" in this document) is a hand-held operator interface device that gives motion control solely to the operator and displays the software menus. It is connected to the controller via a cable that plugs into either the MAIN CPU board inside the controller.

### FEATURES

1. ON/OFF switch
2. E-STOP button
3. Menu system
4. USB port
5. Touch screen

## 2.2 Basic Control Using iPendant

The robot can be enabled using the power switch on the iPendant.

### STATUS BAR

MUST address and clear any FAULTS.

## LED INDICATORS

GREEN LED - power is ON  
RED LED - there is a FAULT

## EMERGENCY STOP BUTTON

The red E-stop button on the iPendant is used to stop the robot in EMERGENCY situations. This will cut off power to the motors and engage the brakes. For NON-EMERGENCY stops press the HOLD key. This is less stressful on the robot and braking system. Reset the e-stop by rotating the red knob clockwise until it clicks. Then it will pop out.

## 2.3 Dead man Switch

When operating and programming the robot using the teach pendant, the user must hold the teach pendant. The DEADMAN SWITCH on the back of the teach pendant that must be held in the CENTER position to run the robot. This is a safety feature. The deadman switch fully pushed in (e.g., users tend to grasp in response to an emergency) or fully released (e.g., user drops the pendant or is knocked out) will result in the robot stopping. There are 2 deadman switches on the back. Only 1 of them must be held in position.

The deadman switch is always used in conjunction with the SHIFT key on the pendant.

It can be tiring to hold the deadman switch for a long time. A user can be judicious on when to hold the switch. When typing in robot program commands (insert, copy, typing, deleting lines), you can let it go. However, you MUST hold it (plus SHIFT) to move, jog the robot, or run a robot program.

## 2.4 Menus

The Pendant has a menu system that allows users to access almost any function.

Press MENU to reach the menu system.

Sub-menus are accessible either by arrowing to the item or pressing the associated number key.

### QUICK / FULL MENUS

Someone may accidentally set to QUICK menus. To restore FULL menus, do the following

1. FCTN > 0 (Next) > 1 - Quick/Full Menus

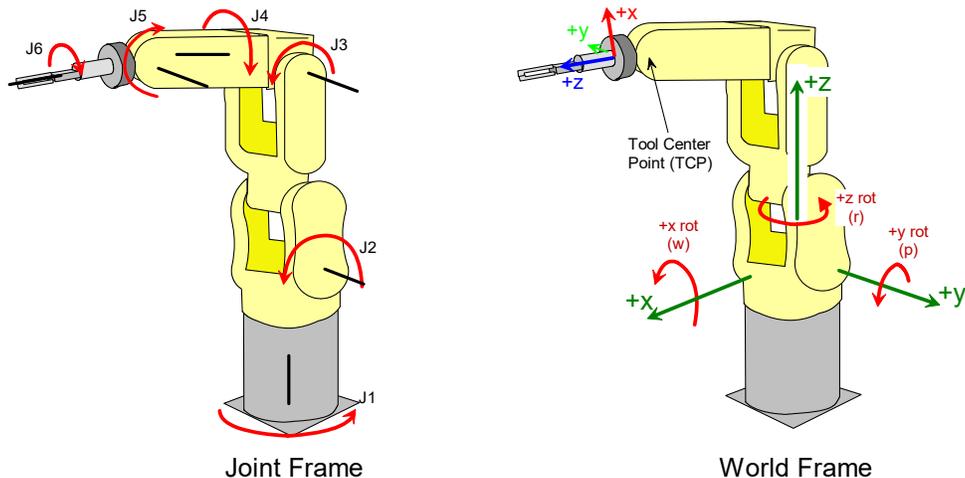
### SETTING MENU FAVORITE

1. Navigate to the desired menu item
2. Press MENU
3. Long press one of the available tiles at the bottom of the screen (3 sec).

## 2.5 Split Screens

The pendant has options for splitting the screen so information can be displayed in multiple windows at the same time.

# CHAP 3 - COORDINATE FRAMES



## 3.1 Intro

Generally a robot arm moves its END EFFECTOR (or sometimes called gripper) in space. This generally requires knowledge of the end effector's position in space. Position and movement in space must be done explicitly and unambiguously. This requires the use of COORDINATE FRAMES. There are different coordinate frames. You switch coordinate frames by pressing COORD (while NOT holding SHIFT).

## 3.2 Joint Coordinates

A user can jog the robot in joint coordinates. This simply means moving the robot one joint at a time, specifying the joint, and how much angular movement is made. Each joint is numbered (see previous section), and there are positive and negative rotation angles.

## 3.3 World Coordinates

There is a WORLD COORDINATE FRAME with an origin is where joint axis J1 & J2 intersect, the X-AXIS pointing straight out, the Y-AXIS points to the robot's LEFT, and the Z-AXIS points straight up. The coordinates in world frame describe the position (x,y,z) and orientation (w,p, r) of the TCP (tool center point).

### 3.4 Other Frames

Other frames can be created as well (USER, TOOL, etc.). There are different ways to set up these custom frames (2 pt +Z, 3 pt, 4 pt, 6-pt XZ, 6-pt XY). Many robot programs are written in those specific frames.

QQQ - are frame definitions global?

## CHAP 4 - JOGGING

### 4.1 Intro

JOGGING the robot involves commanding the robot using the Pendant so its end effector is in a desired position and orientation. Jogging is the primary way of programming the robot to move on a path. The robot is jogged to a desired position and that position is stored in the robot program. The robot program will often consist of a sequence of programmed positions, where each position is recorded using the this technique.

The main considerations in jogging are:

- 1.) How fast the robot moves
- 2.) Which coordinate system to use while jogging.

### 4.2 Jogging

Jog the robot by pressing SHIFT and the JOG KEYS (blue). You MUST be SHIFT pressed to jog the robot. Jog speed is based on the speed set (use speed override keys, Arrow UP and Arrow DOWN keys) and displayed in the upper right in green.

### 4.3 Jog Speed

The job speed determines how fast the robot moves when the jog keys are pressed. The job speed is a percentage of the maximum allowable job speed. The speed refers to the movement of the Tool Center Point (TCP).

Speed values	Joint	Cartesian
100... 1	% of jog speed	% of jog speed
FINE	~ 0.001 degrees	~ 0.023 mm
VFINE	~ 0.0001 degrees	~ 0.002 mm

#### 4.3.1 Position & Orientation

You can display the gripper's POSITION STATUS (its current position and orientation) by pressing the POSN button. After pressing POSN, joint angles (in degrees) and the gripper's location in space in WORLD coordinates in XYZ (in mm) and the gripper's orientation in w,p, r (in degrees) are displayed as below.

```
Joint                                Tool: 1
J1:  -16.877      J2:      17.136  J3: -12.672
J4:  -31.555      J5:     -34.272  J6: -62.304

World:

Configuration:  N, U, T, 0, 0, 0
:      2320.994   y:    -507.511   z:    556.492
w:      48.591   p:      19.990   r:    -93.546
```

## CHAP 5 - BASIC PROGRAMMING

### 5.1 Intro

There are different types of programs, but we will mainly create TP (teach pendant) programs. The user will press SELECT to show all of the available programs. Then the user can create, open, or delete a program. Newly-created programs are named, and then "edited" (i.e., write the program). Then the user will test the program (STEP or not STEP). Often a prior program will remain in memory so it is a good idea to ABORT that program before trying to run a new one. Finally a user can set up the program to run in AUTO (no pendant).

Note - in our case NEVER DELETE any program. The instructor will handle deleting any programs.

### 5.2 Abort Previous Programs

Sometimes a new program won't run because a prior program still resides in memory. In this case you may get an error like: "Run request failed Program-040 already locked by another task". In this case you must ABORT the prior program.

```
[FCTN] > [1] (Abort All) > [RESET]
```

### 5.3 Create a Program

Steps to creating a program

1. [SELECT] - shows a list of programs already written
2. {Create } - soft menu
3. Name the file (Options/Keybd > Keyboard > type name of file > Exit)
4. {Edit} - to start editing the program

Program starts with END command. You will type above it

## 5.4 Program Soft Key Menu

There are 2 pages of soft key menus.

Page 1:	[+POINT]	[TOUCHUP]
Page 2:	[INST]	[EDCMD]

[+POINT]	Press with SHIFT key records the current robot position as a motion instruction
[TOUCHUP]	Press with SHIFT key to overwrite previously-recorded position with current one
[+INST]	Gives access to programming instructions or logic or branching (e.g., I/O, etc.)
[EDCMD]	(Edit Command) Editing functions (insert, delete, copy, cut)

## 5.5 Important Program Functions

Here we summarize important tasks needed to writing a robot program. These include inserting lines, deleting lines, re-numbering lines, etc.

### 5.5.1 Insert Lines

EDCMD > Insert > Lines > Specify # lines

### 5.5.2 I/O Commands

Programmed I/O commands include, for instance, turning the laser on/off or opening/closing the gripper.

+INST > XX > XX

## 5.6 Teach Points

The main step in creating a robot program is to TEACH POINTS. The basic idea is to JOG the robot to a desired position and orientation, then "teach" (or save) that information.

STEPS

1. Get into program
2. Jog the robot to a desired position and orientation
3. (SHIFT) + [+POINT] (soft menu). A line of code will be inserted that looks like (for eg) below:

```
1: J P[1] 100% FINE
```

## 5.7 Naming Local Positions

You can NAME your local positions. Just having a list of positions (P[1],P[2], etc.) can become confusing. You may not remember what P[7] is. Naming the local points helps you to remember what they are.

## STEPS

1. Highlight P number.
2. Hit ENTER, then KEYBOARD
3. (SHIFT) + [+POINT] (soft menu). A line of code will be inserted that looks like (for eg) below:

```
1: J P[1:2nd box position] 100% FINE
```

## 5.8 Renumbering Local Positions

As you create, correct, and modify programs you may find that different positions are numbered out of sequence (e.g., P[5], P[3], P[1], etc.). You can renumber them. Then your P's will appear sequentially higher and higher. Note renumbering will keep track of re-used P's (eg, if P[3] appears originally as P[5] and it appears many times in the program, all of the P[5]'s will be changed to all P[3]'s).

## STEPS

1. Move cursor to top line (if you want start renumbering there)
2. Ensure SHIFT is OFF.
2. EDCMD > 6 Renumber > Yes

```
1: xxxxx
```

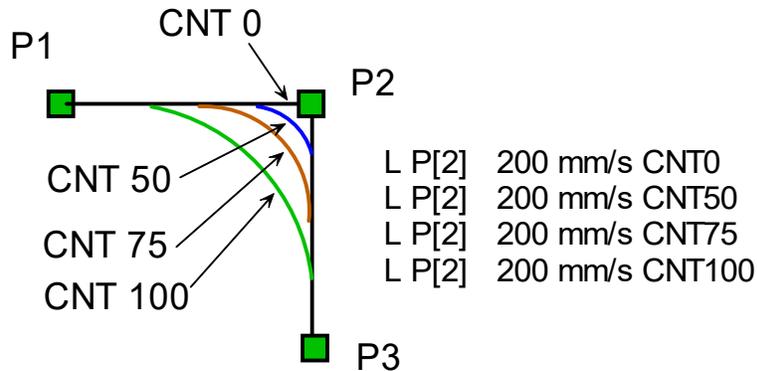
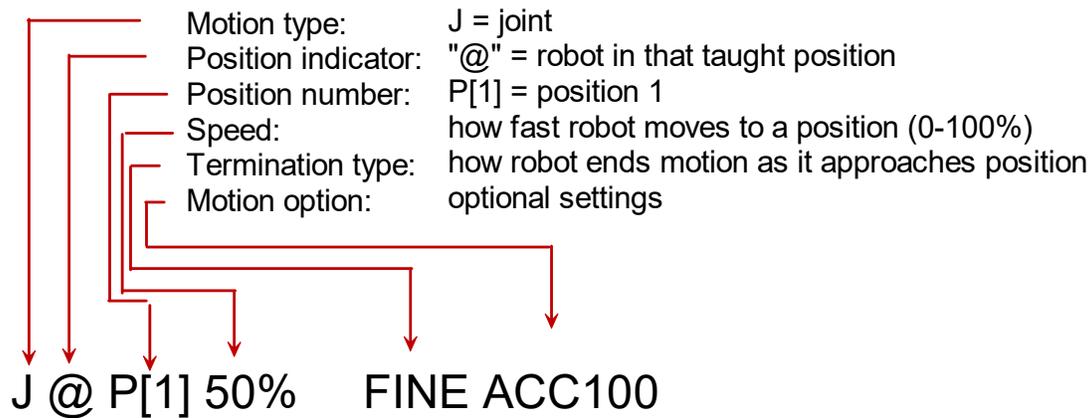
## 5.9 Motion Instructions

As you teach the robot to move to different points, there are a number of parameters to adjust that specifies HOW the robot will move about these points.

```
J @P[1] 50% FINE ACC 100
```

Motion type:	J (joint), L (linear), C (circle) (how to move)
Position indicator symbol:	@ ("@" appears is robot is in that position)
Position number	[1] example (where to move)
Speed	50% (1 - 100) (how fast to move)
Termination type	FINE, CNT (continuous 0-100) (what to do when you get there) 100 means less deceleration more deviation

- FINE - robot stops at destination point
- CNT - Continuous, robot slows down near destination point
- CNT 0 - is basically = FINE
- CNT 100 - "rounds" corners MORE (smoother but less accurate)



## MOTION TYPE (J, C, L)

Motion type refers to HOW the robot will move as it moves to the programmed point.

- J: JOINT - This is the simplest motion, causing the joints to start and end at the same time. The result is that the gripper tends to move in a gentle arc.
- L: LINEAR - this causes the gripper to move in a straight line. Use this option only if required. It requires much more complex computations by the controller. The joints all start and stop at different times and at different rates to achieve the straight motion.
- C: CIRCULAR - use this option if you need to move in a circular or arc motion.

## POSITION

P[x] This specifies the position where the robot is moving to. It can also be a position register (PR[]).

**SPEED** This specifies HOW FAST the robot should move as it goes to the desired position. There are different ways to specify speed. In J mode, you see percentage (0-100% where the percentage is that of the programmed speed). In L mode you'll see velocity units that you can change (in/min, deg/sec, mm/s, etc.)

**TERMINATION TYPE.** This specifies what to do when you arrive at the position.

FINE = robot goes to the position exactly (& pauses there briefly)

CNT = (CONTINUOUS)

0 = stops at the point

0-100 = medium deceleration

100 = minimum deceleration (robot does NOT slow down much at all),  
(note at 100 - the robot deviates MORE from the programmed point)

A program taught in one User Frame and Tool Frame combination WILL NOT RUN if a different User and Tool Frame combination is active.

## 5.10 Touch Up

You may have previously taught a point (say P[1]) but you decide you need to change P[1] to a different position. In the program arrow to the line where the robot moves to P[1]. Then jog the robot to the new desired position/orientation. Select TOUCHUP while pressing SHIFT. The program will set P[1] to the new position. All P[1]'s in the program will take on this new position.

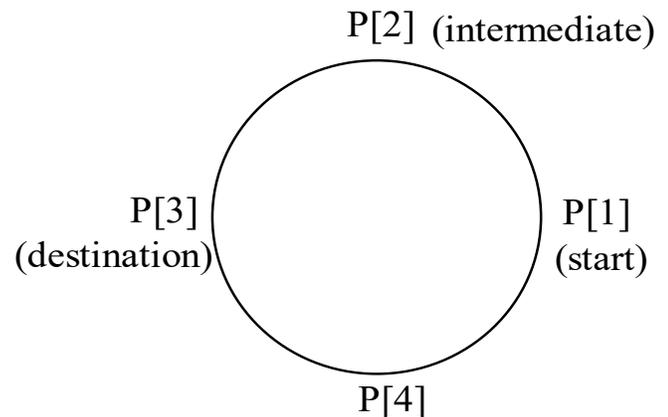
If the existing taught position (in this case P[1]) is used more than once, the software will prompt you with "set new id" just in case you didn't want to change the other instances. Select YES or NO WITHOUT PRESSING SHIFT.

## 5.11 Circular (Circle) Motion

Robot moves the tool center in ARC from start position thru an intermediate position to a destination position.

Program looks like the following

```
J P[1] 5% FINE
C P[2]
  P[3] 100 mm/sec FINE
J P[3] 5% FINE
C P[4]
  P[1] 100 mm/sec FINE
J P[1] 5% FINE
```



### STEPS

1. Teach positions P[1], P[2], P[3], & P[4].
2. Arrow program line of P[2] and change the motion type to "C"
4. A destination position line of code is automatically created.
5. Change the position # for the new line of code to "3"
6. Change movement speeds if desired (arrow to the value and enter a new value)
7. To complete the circle, create a 2nd arc using points: P[3] (start), P[4] (intermediate), and P[1] (end).
8. Repeat the steps above for the 2nd arc.

### CIRCULAR MOTION

Circular motion would consist of 2 arc commands. The program above will have the robot stopping at the taught points ("FINE"). If you want more continuous motion change the termination type from FINE to CNT (or continuous). A "CNT100" programs the smoothest movement.

### 5.12 Loops

Loops are done using LABELS and JUMP LABELS.

```
[EDCMD] > Insert line > 1 >
[+ INST] > JMP LBL
```

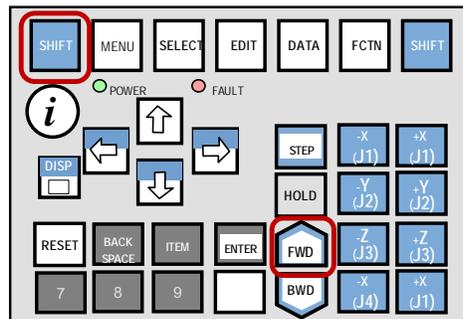
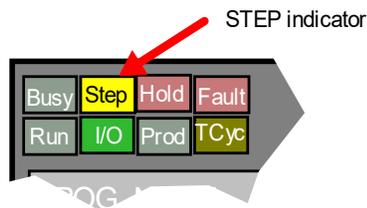
```
1:  LBL[1]
2:J  P[1]  5% FINE
3:J  P[2]  5% FINE
4:J  P[3]  5% FINE
5:J  P[4]  5% FINE
6:  JMP LBL[1]
```

### 5.13 Edit Commands

You may wish to edit lines of your program.

- DELETE LINES
- COPY/PASTE
- DISPLAY COMMENTS

### 5.14 Testing Programs



There are 2 options for testing a program using the iPendant.

1. STEP OFF - the program will run one line of code after the next.  
Run the program by pressing: [SHIFT] [FWD]
2. STEP ON - the program only runs ONE LINE AT A TIME. (yellow STEP will appear on screen)  
You must press FWD each time to run the next line of code

## 5.15 Running programs

Programs can either be run in TEST or AUTO. See other section on settings required for each.

Sometimes you may pause (HOLD) a program and restart it, or perhaps switch to another program.

The Fanuc software pop up a yellow window with a warning:

The cursor is on a different line from where the program PAUSED [2].  
Are you sure you want to run from this line?

[YES]                      NO

If this is okay, then select YES by pressing enter (for some reason the touch screen does not work for this).

This window can be avoided if you ABORT the prior program (FCTN > Abort all)

## 5.16 Running a program in AUTO

1. Select AUTO mode on controller
2. Check that E-stops are disengaged.
3. Choose program to be tested (it should have a loop)
4. Place cursor at the point to start the program
5. Ensure STEP status is NOT in STEP mode.
6. Ensure the manual speed is BELOW that specified in program (or you get an error during auto)
7. Clear any faults.
8. Turn Pendant OFF
9. Press CYCLE START (green button on controller).
10. You may exceed programmed speed using [SPEED UP] or [SPEED DOWN] arrows while it runs.

## 5.17 Open a Program

SELECT > Arrow to program > ENTER

## 5.18 Existing Programs

Open  
Modify  
Delete

## 5.19 Program Types

## 5.20 Miscellaneous

Be careful a previous program can leave a value in a register  
Often a good idea to clear registers at the start of your program

# CHAP 6 - INPUT OUTPUT

## 6.1 Intro

The robot must often communicate with various robot systems.

## 6.2 Gripper Control

The gripper (or the official term is "end effector") can either be opened or closed. The gripper is actuated via pneumatic air pressure. Control of the gripper opening and closing falls under ROBOT I/O.

Menu:            Menu > (5) I/O > (6) Robot I/O >

## 6.3 Laser Pointer

The laser pointer can be helpful when you are trying to position the robot gripper. The laser can be turned and off in the DIGITAL I/O section. Access the digital I/O section in either of 2 ways: menu, or [I/O] button. Note the laser is NOT centered in the robot gripper.

Menu:            Menu > (5) I/O > (3) Digital > DO [101] (arrow to it)  
or  
[I/O]            [I/O] > { Type } > (3) Digital > DO [101] (arrow to it)

Turn on/off using either the touch screen or the associated SOFT KEYS

## 6.4 Camera Light Control

The camera light is usually turned ON when using the robot's camera. The camera light can be turned on and off in the DIGITAL I/O section.

MENU

Menu > (5) I/O > (3) Digital > DO [102] (arrow to it) > On/off (touch screen or soft keys)

I/O BUTTON

[I/O] > {Type} > (3) Digital > DO [102] (arrow to it) > On/off (touch screen or soft keys)

## CHAP 7 - FILE MANAGEMENT & DATA

### 7.1 Intro

Files are managed on the teach pendant. In this section we will learn how to navigate the file system. We will also learn to create, save, and open programs.

### 7.2 File Navigation

#### SELECTING FILES

Select files by pressing [SELECT].

A list of files appears. Press [UP] or [DOWN] arrow keys to navigate through the list.

Pressing [SHIFT] while pressing [UP] or [DOWN] pages up and down.

#### COPYING FILES

Use [COPY] from menu choices.

#### RENAMING FILES

Use [COPY] from menu choices. Give the copy a new name (use keyboard). Delete the original file.

### 7.3 Registers

Registers are used to store information.

There are DATA REGISTERS and POSITION REGISTERS.

Registers are different than the programmed points in a teach pendant program.

Registers are GLOBAL (all programs can see them and have access to them).  
Position points in a program are LOCAL. Only that program sees them. No other programs can see them.

Example of data register: R[1].  
Example of position register: PR[1].

It is a good idea to clear out registers at the start of your program.

Note the "R" on the right (below) means RECORDED (or set up already)

**DATA Position Reg**

```
PR [ 1: HOME ]=R
PR [ 2: Cal Setup POS ]=R
PR [ 3: Target_pos ]=R
PR [ 4: Cal Setup POS ]=R
PR [ 5: ]=*
PR [ 6: ]=*
```

**DISPLAY REGISTERS**

Press [DATA]

**7.3.1 Position Registers**

**CREATING**

Arrow to new PR > Record (soft menu) >  
Pick > Enter > Keyboard > Type a name for register

**CLEARING**

Clear position register (select PR > Clear)

**CHAP 8 - SETUP & MAINTENANCE**

**8.1 History**

Purchased:  
Cost:  
Part of SC building project.

Curriculum - per Joe Ray - 50 student hours or 10 wks

## 8.2 Axis Limits

Axis limits allow the user to set limits for safety or other reasons.

1. MENU
2. 0 (NEXT)
3. Cursor to SYSTEM
4. Axis Limits

## 8.3 Fault and Alarms

### DIAGNOSING FAULTS OR ALARMS

1. Navigate to proper screen, active alarm, or history
2. Cursor next to target fault or message
3. Press SHIFT and DIAG (diagnose) together

### CLEARING FAULTS

You **MUST** clear faults (press SHIFT and RESET at same time) before you can jog the robot. Sometimes you can just push RESET.

There are sometimes circumstances where you must go to the ALARM screen to clear the fault

## 8.4 Pendant Settings

### MENU SETTING

Sometimes you are looking at a list of programs but the labels on the bottom are missing. This CAN be used to prevent someone from creating programs.

### QUICK VS. FULL MENUS

[FCTN] > Quick vs. Full Menus (KEEP FULL MENUS FOR NOW)

## 8.5 Mastering & Batteries

The Fanuc LR Mate 200iD/4S utilizes encoders to sense the robot joint angles. However, most encoders are INCREMENTAL, not absolute. The encoder values must be zeroed with the robot in a known configuration. Thus the robot must be HOMED or MASTERED (the term used by Fanuc) for the Fanuc to know its joint angles. This means the user must JOG the robot to specific angles as specified by "witness marks" (physical hash marks on the robot). The user must then enter those locations using the teach pendant. After mastering, the encoder counts are saved into memory. Unfortunately this memory is "volatile" requiring energy to maintain it. Batteries are used to maintain this data.

If the battery is allowed to go dead, the robot will give you a BZAL (battery zero alarm).

Note BLAL is a battery LOW alarm. In this state, you still have a chance to change the batteries without re-mastering the robot.

Batteries must be changed **WHILE THE CONTROLLER IS ON** or mastering data will be lost. If this happens the robot must be re-mastered. Batteries (4x C batteries) **should be replaced 2x per year**. The vendor has noted that re-chargeable batteries are not recommended.

## BATTERY CHANGING PROCESS

1. Turn ON the robot controller. Ensure the controller is powered on too.
2. Using #3 mm hex wrenches, remove the battery compartment cover.
3. Remove compartment cover. Use a spatula to assist if it's stuck.
4. Replace batteries one at a time to ensure you install them in correct direction.
5. The robot uses **FOUR (4) C-SIZED BATTERIES**
6. Tighten the outer door **LOOSELY** (leave some jiggle). It gets stuck otherwise.

### 8.5.2 Mastering Procedure

Menu > 0-Next > 6-System > 3 Master/Cal

7. Menu > 0 Next > 6 System > 3 Master/Cal
8. Then, RES\_PCA (reset pulse coder alarm) (F3)
9. "Reset?" Press YES (F4)
10. Clear the fault (deadman / shift > Reset)
11. Move each joint about 10 degrees (to let robot know data transfer is happening)
12. Get into JOINT coord
13. Jog motors (J1 thru J6) and align the witness marks
14. Menu > 0 Next > 6 System > 3 Master/cal > 2 Zero Position Master > Enter
15. Yes (F4)
16. 7 Calibrate, Yes (to zero out joints angles in the master position)
17. Yellow alarm - need to apply DCS param
18. Menu > 0 Next > 6 System > 9 DCS > 13 Mastering Parameters (appears red: have changed)
19. Apply (F2)
20. "Code Number (master): \_----- (enter 1111)
21. Wait
22. All (F3)
23. OK
24. Yellow alarm - cycle power to use new DCS params
25. Power off controller
26. Power back on
27. Move robot out of singularity (move J5 at least 10 degrees)

## 8.6 Singularity

SINGULARITY is a robot configuration where a DOF is lost. For the Fanuc this happens when Joints 4 and 6 are aligned (robot points straight out), and when J5 is 0 degrees.

### RECOVERING FROM SINGULARITY

Press COORD. Select JOINT  
Move J5 +/- 10 degrees

## 8.7 Maintenance Reminder

The xxxx

Maintenance Reminder > How many days since maintenance been done  
(if not in industrial environment, just click OK)

> Detail >

Mechanical Check  
First Controller Check

## 8.8 Lubrication

??

## 8.9 Other

Per rep Joe Ray, Timothy Mehring (of MI) has some nice youtube videos on Fanuc.  
Per rep Dan,  
Also there's Dan Staifer (youtube).

# CHAP 9 - xxx

## 9.1 Section A

### 9.1.1 sub-section

## 9.2 Section B

### 9.2.1 sub-section

ct\_text2

Here is some **EXAMPLE** text.

ct\_text0\_TITLE

**ct\_text0\_exercise**

ct\_x-programg

ct\_x-programg2

ct\_x-programg3

ct\_text1

1. ct\_8\_numberedList
2. ct\_8\_numberedList (hanging 0.1")
- 3.

END -----

ct\_text3

ct\_text4

ct\_0-FIGURE