Robot Programming

C. Sivakumar
Assistant Professor
Department of Mechanical Engineering
BSA Crescent Institute of Science and Technology

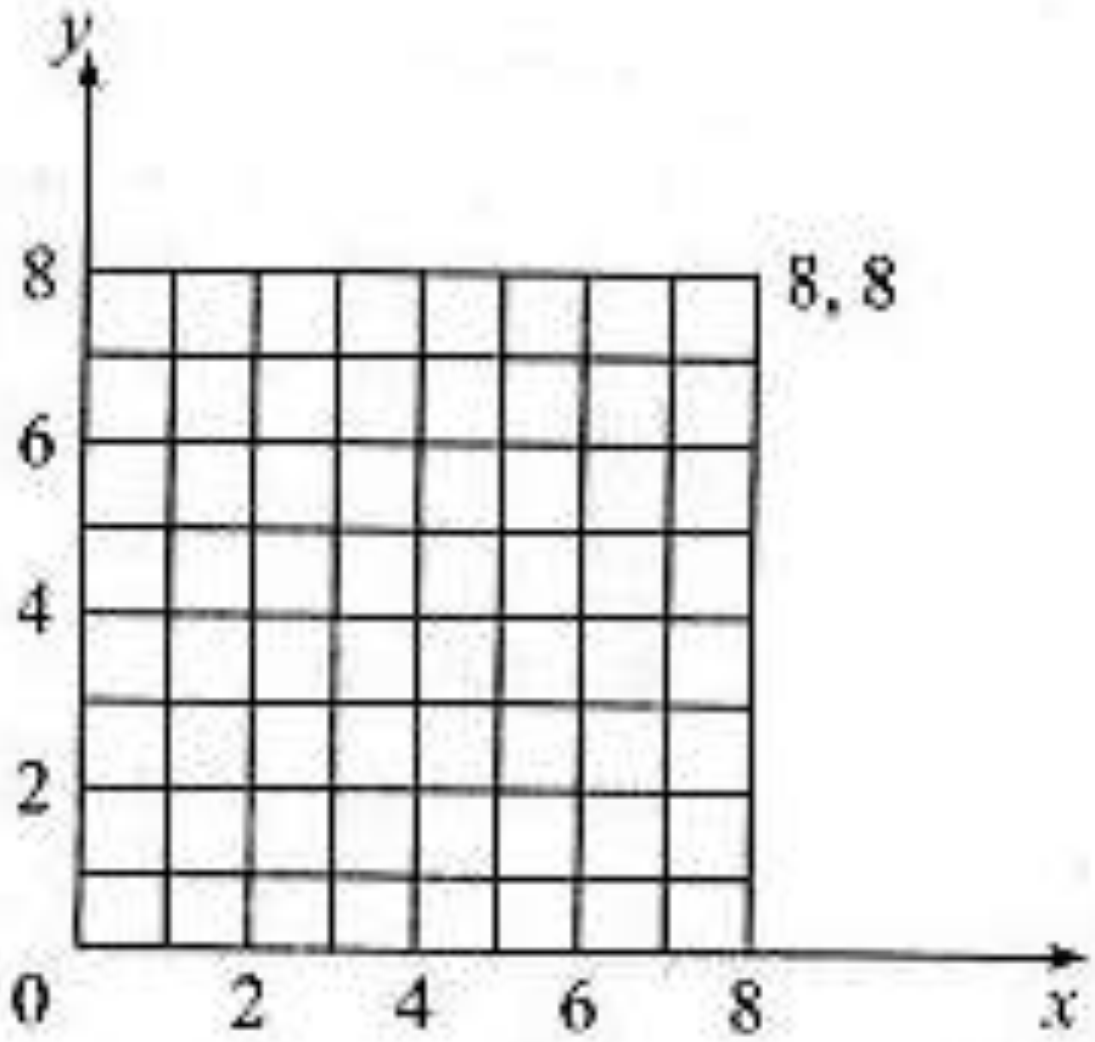
- Methods of programming
- Leadthrough programming
- Manual programming
- Robot program in space
- Methods of defining point in space
- Speed control

Teach Pendant

Motion Interpolation

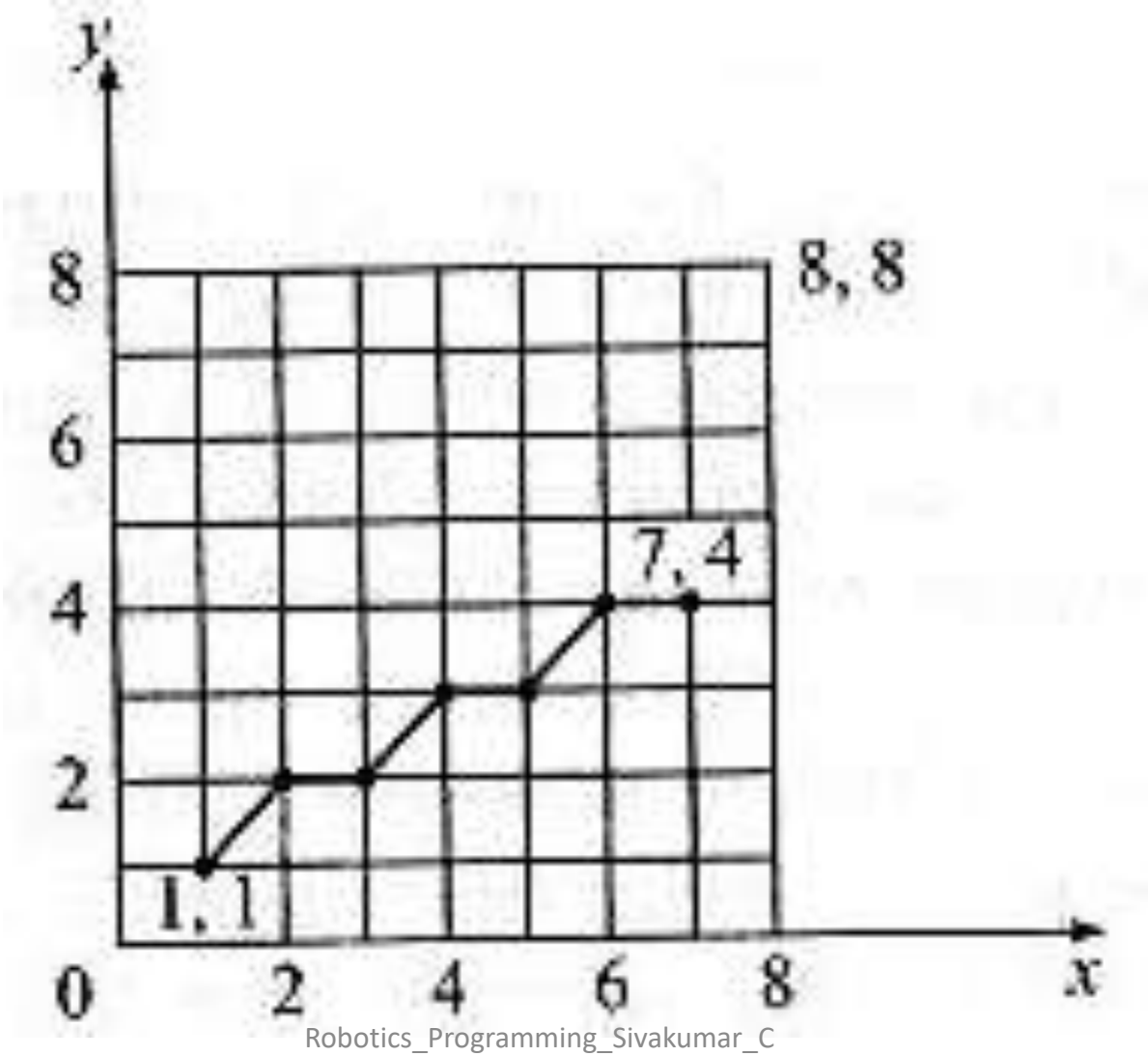
- Joint Interpolation
- Straight line Interpolation
- Circular Interpolation
- Irregular smooth motion

Joint Interpolation

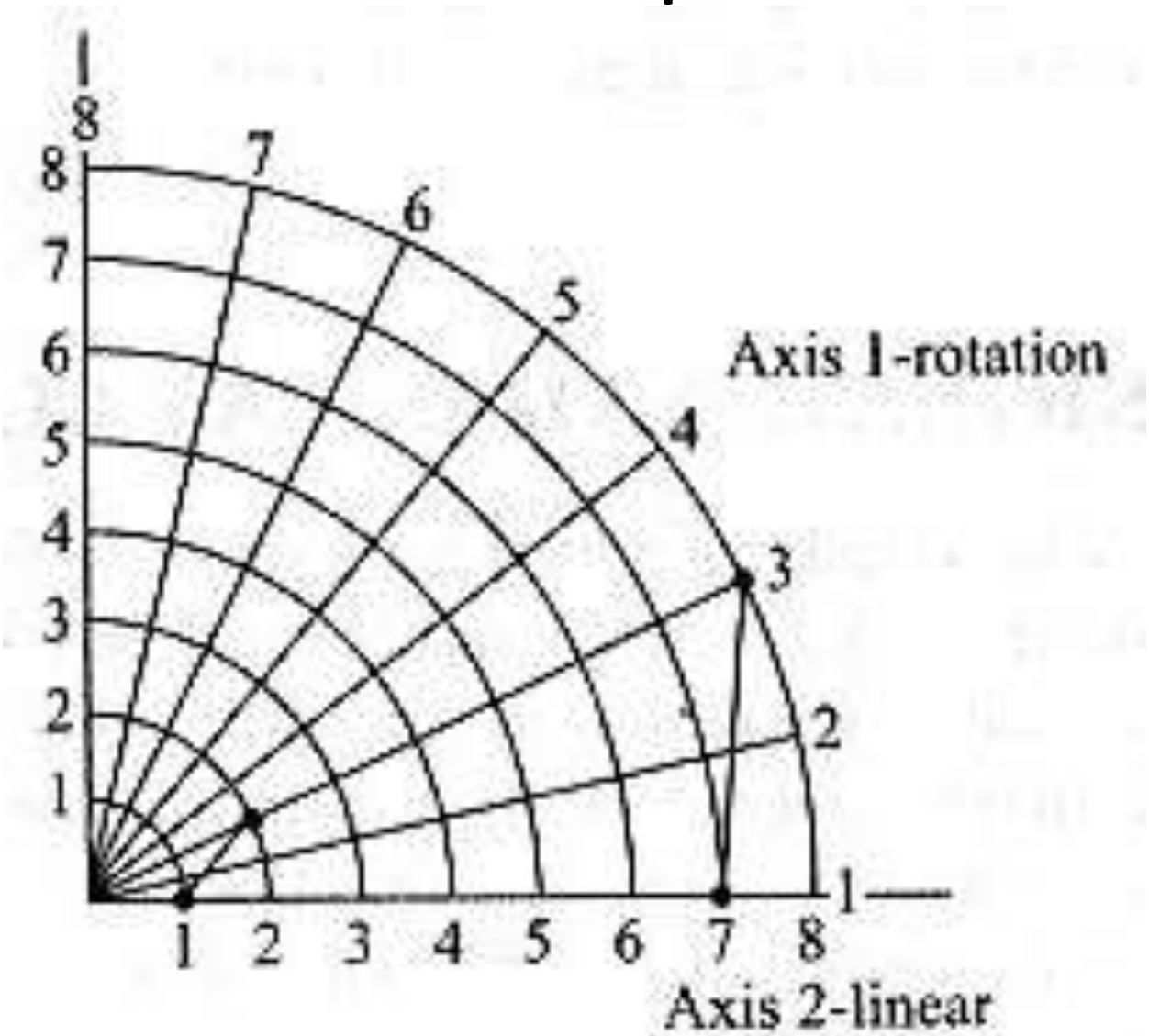


Joint Interpolation

Step	Move	Comments
1	1,1	User specified starting point
2	2,2	Internally generated interpolation point
3	3,2	Internally generated interpolation point
4	4,3	Internally generated interpolation point
5	5,3	Internally generated interpolation point
6	6,4	Internally generated interpolation point
7	7,4	User specified end point


y
8
6
4
2
0
2
4
6
8
x
8, 8
7, 4
1, 1

Circular Interpolation



WAIT, SIGNAL & DELAY COMMANDS

- All industrial robots are instructed to send signals or wait for signals
- These signals are called interlocks
- Common form is to actuate end effectors
- In grippers, its on or off or Binary
- Grippers involve 2 interlocks – open & close
- Feedback might be added to verify actuation

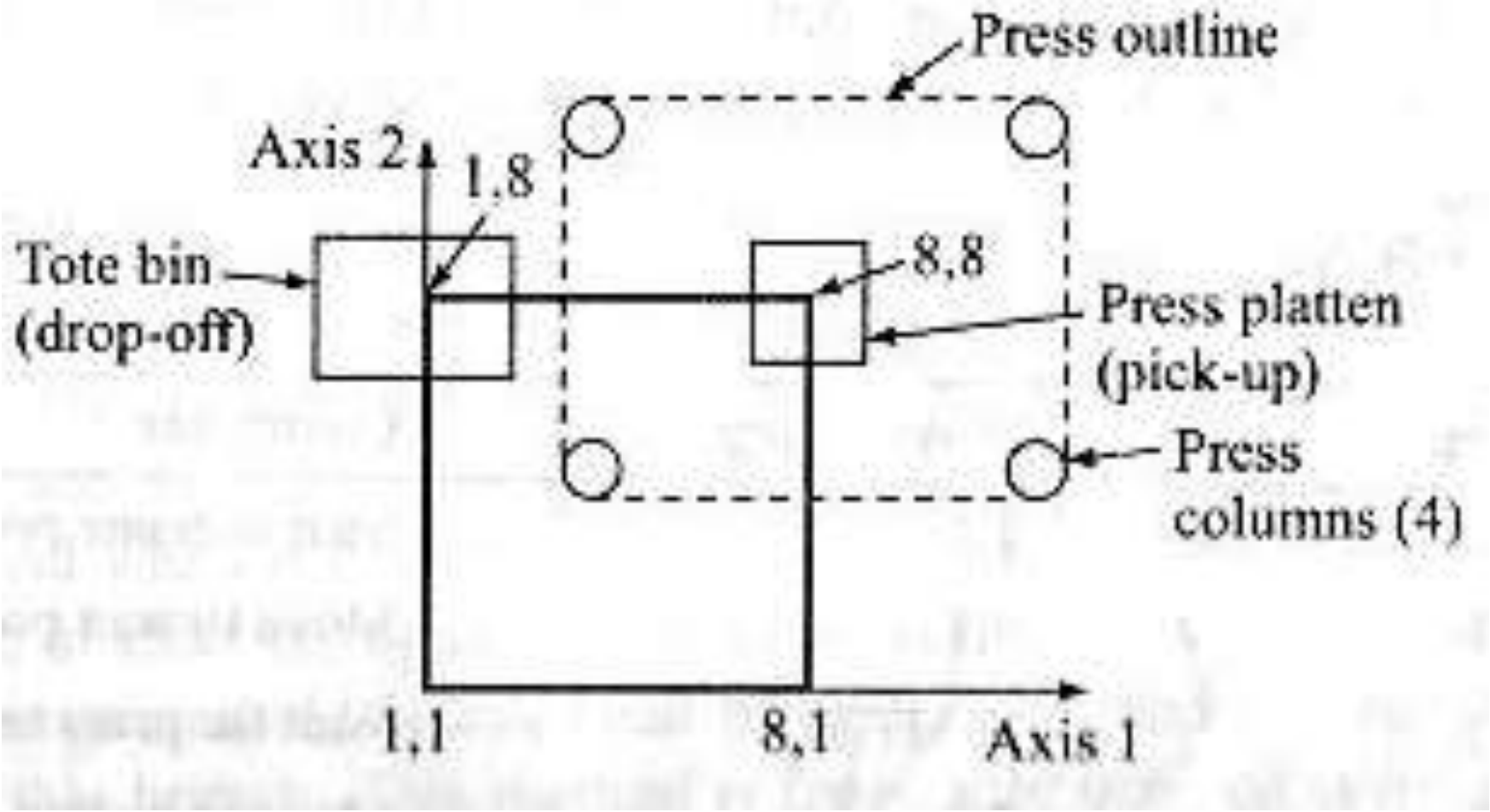
WAIT, SIGNAL & DELAY COMMANDS

- Communication with other devices is important.
- Ex of unloading from press
  - Stop robot entering before press is open
  - Remove gripper before press closes
- To do this we have 2 commands
- SIGNAL M – instructs to O/P signal thru M
- WAIT N – robot should wait until I/P thru N

Example of loading and unloading

- 8,8 – press
- 1,8 – tote bin or collecting tray
- 1,1 & 8,8 will be safe locations for waiting
- Controller port 1-10 = Output lines (SIGNAL)
- 4-actuate press, 5&6 – OPEN/CLOSE gripper
- Controller port 11-20 = Input lines (WAIT)
- 11- indicates gripper is open

Example of loading and unloading



Step	Move or signal	Comments
0	1,1	Start at home position
1	8,1	Move to wait position
2	WAIT 11	Wait for press to open
3	8,8	Move to pickup point
4	SIGNAL 5	Signal gripper to close
5	8,1	Move to safe position
6	SIGNAL 4	Signal press to actuate
7	1,1	Move around press column
8	1,8	Move to tote pan
9	SIGNAL 6	Signal gripper to open
10	1,1	Move to safe position

DELAY

- DELAY X SEC
- Robot should wait X seconds before going into next step

Step	Move or signal	Comments
0	1,1	Start at home position
1	8,1	Move to wait position
2	WAIT 11	Wait for press to open
3	8,8	Move to pickup point
4	SIGNAL 5	Signal gripper to close
5	DELAY 1 SEC	Wait for gripper to close
6	8,1	Move to safe position
7	SIGNAL 4	Signal press that hand is clear
8	1,1	Move around press column
9	1,8	Move to tote pan
10	SIGNAL 6	Signal hand to open
11	DELAY 1 SEC	Wait for gripper to open
12	1,1	Move to home position

BRANCHING

- Controllers allow dividing a program into one or more branches
- Allows program to be subdivided into convenient segments
- It is also subroutines and can be identified by a name
- Allows incoming signal to invoke branch
- Usage of interrupt

Pallet with 24 positions

General commands

Point name	Explanation
SAFE	Safe location to start and stop
PICKUP	Location of part pickup at end of chute
INTER	Intermediate point above chute to pass through
LOC 1	Location of first pallet position
LOC 2	Location of second pallet position
⋮	⋮
LOC 24	Location of 24th pallet position
ABOVE 1	Location above first pallet position
⋮	⋮
ABOVE 24	Location above 24th pallet position

Step	Command	Comments
1	MOVE SAFE	Move to the starting safe position
2	WAIT 11	Wait for start signal on line 11
(The following portion of the program directs the robot to pick up first part.)		
3	MOVE INTER	Go to the intermediate point above chute
4	WAIT 12	Wait for next part from chute
5	SIGNAL 5	Open gripper
6	MOVE PICK UP	Move gripper to pick-up part
7	SIGNAL 6	Close gripper
8	MOVE INTER	Depart to intermediate point above chute
9	MOVE ABOVE 1	Move to point above first pallet location
10	MOVE LOC 1	Position part in first pallet location
11	SIGNAL 5	Open gripper
12	MOVE ABOVE 1	Depart slowly from pic-kup point

(The next portion of the program directs the robot to pick up second part.)

13	MOVE INTER	Go to the intermediate point above chute
14	WAIT 12	Wait for next part from chute
15	SIGNAL 5	Open gripper
16	MOVE PICK UP	Move gripper to pick-up part
17	SIGNAL 6	Close gripper
18	MOVE INTER	Depart to intermediate point above chute
19	MOVE ABOVE 2	Move to point above second pallet location
20	MOVE LOC 2	Position part in second pallet location
21	SIGNAL 5	Open gripper
22	MOVE ABOVE 2	Depart slowly from pick-up point

(The preceding portions of the program are repeated for the next 21 part 5.)

⋮

(The next portion of the program directs the robot to pick up 24th part.)

232	MOVE INTER	Go to the intermediate point above chute
233	WAIT 12	Wait for next part from chute
234	SIGNAL 5	Open gripper
235	MOVE PICK UP	Move gripper to pick-up part
236	SIGNAL 6	Close gripper
237	MOVE INTER	Depart to intermediate point above chute
238	MOVE ABOVE 2	Move to point above second pallet location
239	MOVE IOC 2	Position part in second pallet location
240	SIGNAL 5	Open gripper
241	MOVE ABOVE 2	Depart slowly from pick-up point

(The pallet is now full.)

242	MOVE INTER	Go to the intermediate safe position
243	SIGNAL 7	Signal that pallet is full

This portion of the program is repeated below, indicating where the subdivision occurs. (The following is the 'fetch' subtask.)

13	MOVE INTER	Go to the intermediate point above chute
14	WAIT 12	Wait for next part from chute
15	SIGNAL 5	Open gripper
16	MOVE PICK UP	Move gripper to pick-up part
17	SIGNAL 6	Close gripper
18	MOVE INTER	Depart to intermediate point above chute

(The following is the 'place' subtask.)

19	MOVE ABOVE 2	Move to point above second pallet location
20	MOVE LOC 2	Position part in second pallet location
21	SIGNAL 5	Open gripper
22	MOVE ABOVE 2	Depart slowly from pick-up point

BRANCH FETCH	Indicates the following is branch FETCH
MOVE INTER	Go to the intermediate point above chute
WAIT 12	Wait for next part from chute
SIGNAL 5	Open gripper
MOVE PICK UP	Move gripper to pick-up part
SIGNAL 6	Close gripper
MOVE INTER	Go to intermediate point above chute
END BRANCH	This is the end of the branch

BRANCH PLACE	Indicates the following is branch PLACE
MOVE Z(–50)	Position part in pallet
SIGNAL 5	Open gripper to release part
MOVE Z(+50)	Depart from pick-up point
END BRANCH	This is the end of the branch

Step	Command	Comments
1	BRANCH FETCH	Indicates the following is branch FETCH
2	MOVE INTER	Go to the intermediate point above chute
3	WAIT 12	Wait for next part from chute
4	SIGNAL 5	Open gripper
5	MOVE PICK UP	Move gripper to pick-up part
6	SIGNAL 6	Close gripper
7	MOVE INTER	Depart to intermediate point above chute
8	END BRANCH	This is the end of the branch
9	BRANCH PLACE	Indicates the following is branch PLACE
10	MOVE Z(–50)	Position part in pallet
11	SIGNAL 5	Open gripper to release part
12	MOVE Z(+50)	Depart from pick up point
13	END BRANCH	This is the end of the branch
14	MOVE SAFE	Move robot to the starting safe position
15	WAIT 11	Wait for start signal on line 11
16	FETCH	Fetch first part
17	MOVE ABOVE 1	Move to first part
18	PLACE	Place first part
19	FETCH	Fetch second part
20	MOVE ABOVE 2	Move to second position
21	PLACE	Place second part
85	FETCH	Fetch 24th part
86	MOVE ABOVE 24	Move to 24th position
87	PLACE	Place 24th part
88	MOVE INTER	Go to intermediate safe position
89	SIGNAL 7	Signal that pallet is full

Capabilities of Leadthrough Methods

- Defining points & speed is easy through Teach pendant (TP)
- WAIT, SIGNAL & DELAY is done in TP through special buttons
- BRANCHING functions

Limitations of Leadthrough programming

- Robot cannot be used in production while being programmed
- More complex program cannot be done
- Not compatible with CIM and factory networking

Robot Languages

- WAVE – 1973 – Stanford AI Lab, feasibility of robot hand-eye coordination
  - AL – control multiple arm
- VAL – Victors Assembly Language (1979)
  - VAL II
- AML – A Manufacturing Language – developed by IBM
- MCL, RAIL, APT, HELP

First Generation Languages

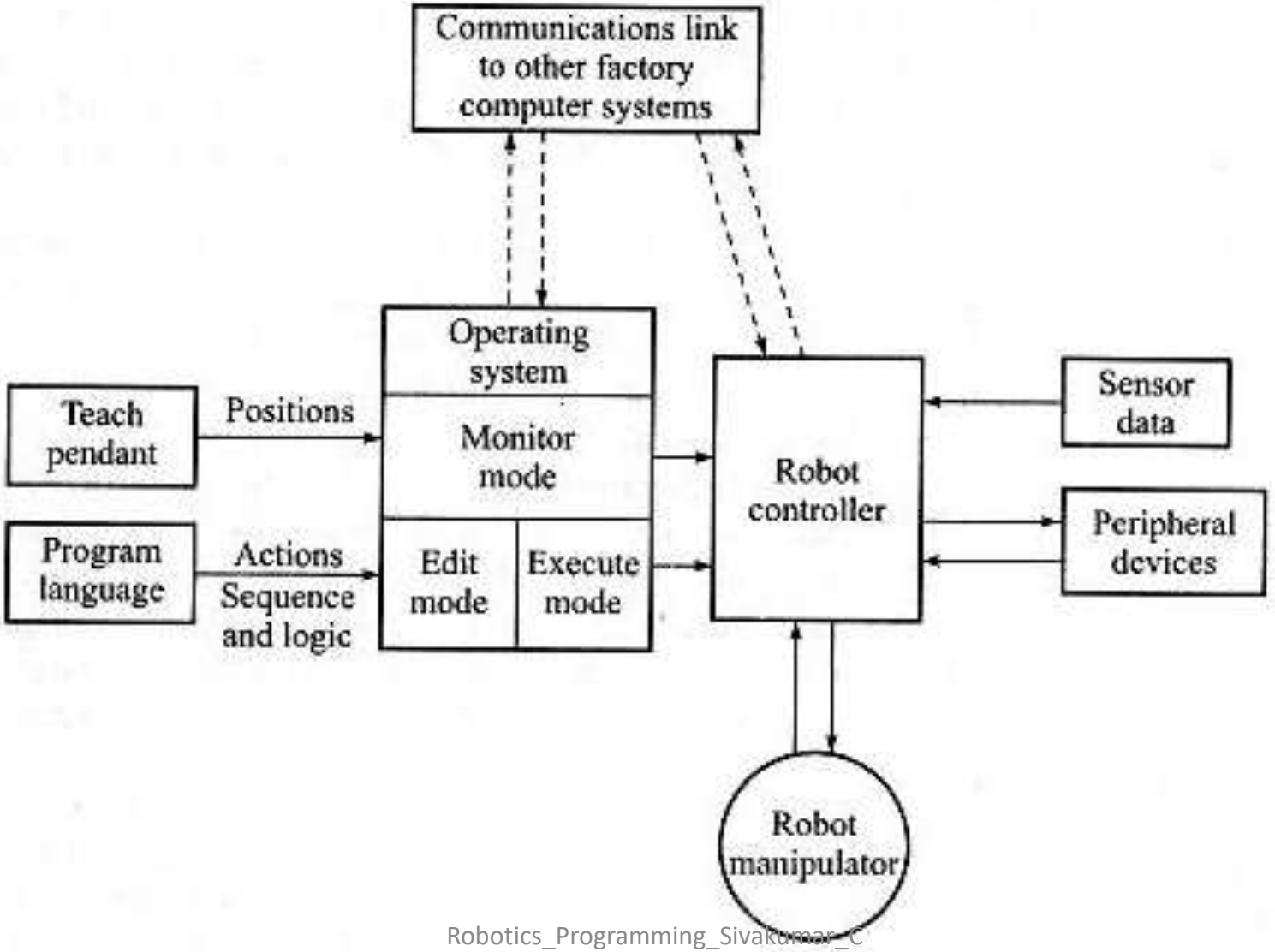
- Uses command statement and teach pendant
- Manipulator motion, straight line interpolation, branching, elementary sensor commands
- Low to medium complexity
- Ex VAL

Seconds Generation Languages

- Also called structured program language
- Ex - AML, RAIL, VAL II
- Make use of teach pendant to define location

- Motion control
- Advanced sensor capabilities
- Limited intelligence
- Communication and data processing

Robot language structure



Robot language structure - Operating system

- Some means of permitting the user to determine the action
- Monitor mode - supervisory control, speed control, store program, transfer program, change mode
- Run mode – execute robot program
- Edit mode – write new program and edit existing program

Robot language structure - Operating system

- Interpreter – a program in OS, executes each instruction of source program
- Compiler – passes thru the entire program and pretranslates all instructions into machine level code, that can be read and executed by robot controller

Robot Language Elements and Function

Constants and variables

- Character, string
- Numerical constants – integers and real numbers

- Aggregates and location variables

⟨50.526, 236.003, 14.581, 25.090, 125.750⟩
⟨'we', 'they'⟩

DEFINE A1 = POINT ⟨50.526, 236.003, 14.581, 25.090, 125.750⟩

Motion command

MOVE and related statements

- MOVE A1 (move to point A1)
- MOVES A1 (with straight line interpolation)
- MOVE A1 VIA A2 (moves to A1 thru A2)

- APPRO A1, 20 (approach at axial offset)
- MOVES A1
- SIGNAL (to close gripper)
- DEPART 50

DMOVE (incremental move)

- DMOVE (1,10) {link 1 moves thru 10 units}
- DMOVE (<4,5,6>,<30,-60,90>)

- MOVE ARM2 TO A1

SPEED control

- SPEED 60 IPS {60 inches per sec}(defines speed unit directly)
- SPEED 75 (defines speed as a % of max speed)

Definition of points in the workspace

- Through Teach Pendant
  - HERE A1

⟨50.526, 236.003, 14.581, 25.090, 125.750⟩

- Through typing

```
DEFINE A1 = POINT ⟨50.526, 236.003, 14.581, 25.090, 125.750⟩
```

Paths and frames

- DEFINE PATH1 = PATH(A1,A2,A3,A4)
- MOVE PATH1
- MOVES PATH1 (using straight line interpolation)
- DEFINE FRAME1 = FRAME(A1,A2,A3)
- DEFINE ROUTE:FRAME1=PATH(P1,P2,P3,P4,P5,P6,P7)
- MOVES ROUTE:FRAME1

END EFFECTOR AND SENSOR COMMANDS

End Effector command – Gripper

- OPEN or CLOSE (wait for next motion)
- OPENI or CLOSEI (immediate)
- CLOSE 40 MM or CLOSE 1.575 IN (int opening)
- CLOSE 3.0 LB (gripping force)
- CENTER (doesnt move the object)

End Effector command – Tools

- OPERATE TOOL(SPEED = 125RPM)
- OPERATE TOOL (TORQUE = 5 IN LB)
- OPERATE TOOL (TIME = 10 SEC)

SENSOR OPERATION

- SIGNAL 3, ON (binary output)
- SIGNAL 3, OFF (binary output)
- SIGNAL 105, 4.5 (analog voltage output)

```
SIGNAL 5, ON        Robot turns on the device
  WAIT 15, ON       Device signals back that it is on
      .
      .
      .
SIGNAL 5, OFF       Robot turns off the device
  WAIT 15, OFF      Device signals back that it is off
```

```
DEFINE MOTOR1 = OUTPORT 5
DEFINE SENSR3 = INPORT 15

        SIGNAL MOTOR1, ON
        WAIT SENSR3, ON
          .
          .
          .
        SIGNAL MOTOR1, OFF
        WAIT SENSR3, OFF

DEFINE VOL T1 = OUTPORT 105

        SIGNAL VOLT1
```

Thank you