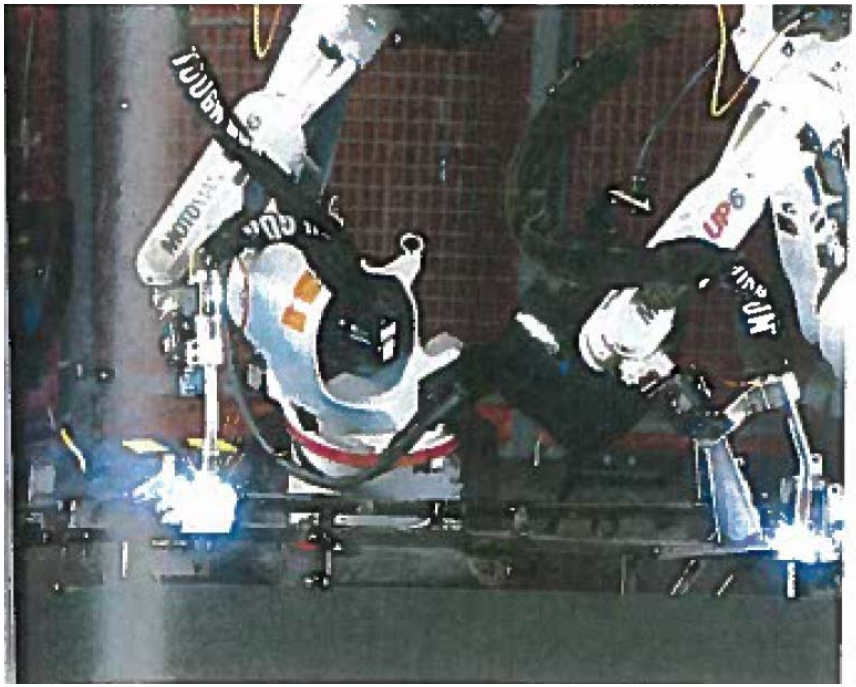


Chapter 3

Programming the Robot

Objectives

By the end of this lesson the learner should be able to:

- Identify and discuss the different motion control applications
- Explain various programming methods
- Discuss characteristics of the different types of programming
- Describe various peripheral applications, such as vision and voice recognition

Key Technical Terms

Artificial Intelligence (AI)	Task-Level Programming	Continuous Path Motion	Off-Line Programming
On-Line Programming	Teach Pendant Programming	End Stop	Pick-and-Place Motion
Voice Recognition	Hierarchical Control Programming	Point-to-Point Motion	High-Level Language
Sensory Feedback	Walk-Through Programming	Manual Programming	Sub Routine
WAVE			

By the end of this lesson the learner should be able to define and explain characteristics / actions related to these technical terms

Evolution of Programming

Three Generations of Robotics

- First Generation – 1950's to 1970's
- Second Generation – 70's to mid 80's
- Third Generation – mid 80's to Present
- Fourth (Future) Generation ????

Your Future

- Careers in Robotics: Software Engineer

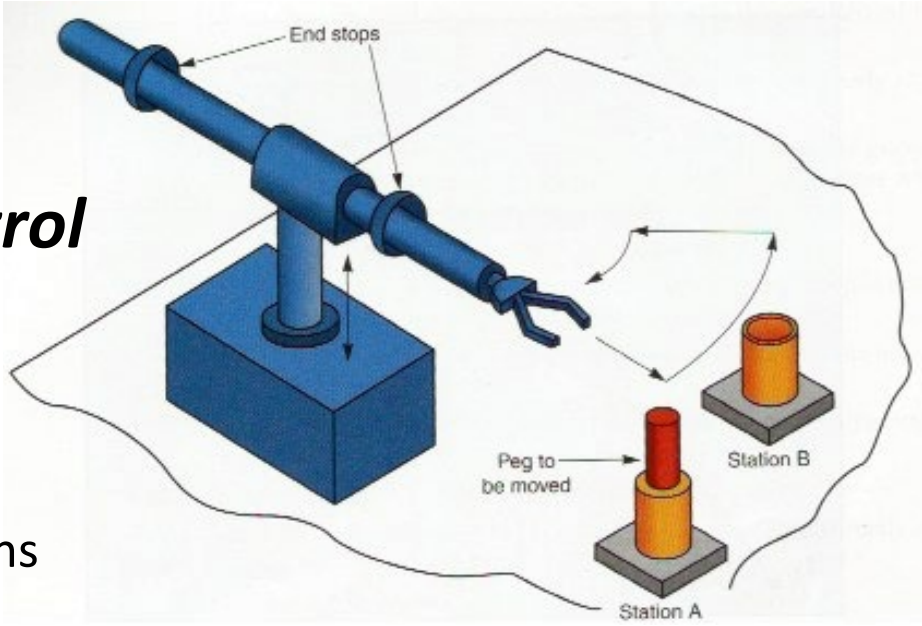
Motion Control

Three basic classifications of ***motion control***

- ***Pick-and-place motion control***
- ***Point-to-point motion control***
- ***Continuous-path motion control***

Motion Control

Pick-and-place motion control



- Repetitive, non-complex operations
- Positioning to designated point set by:
  - Mechanical stops (End Stops)
  - Limit switches (Signals program to turn valve on/off)
- Difficult and tedious to make additions or adjustments
- Low number of points compared to other motion control

Motion Control

Pick-and-place motion control (continued)

- Is pick-and-place a form of open or closed-loop control and why?

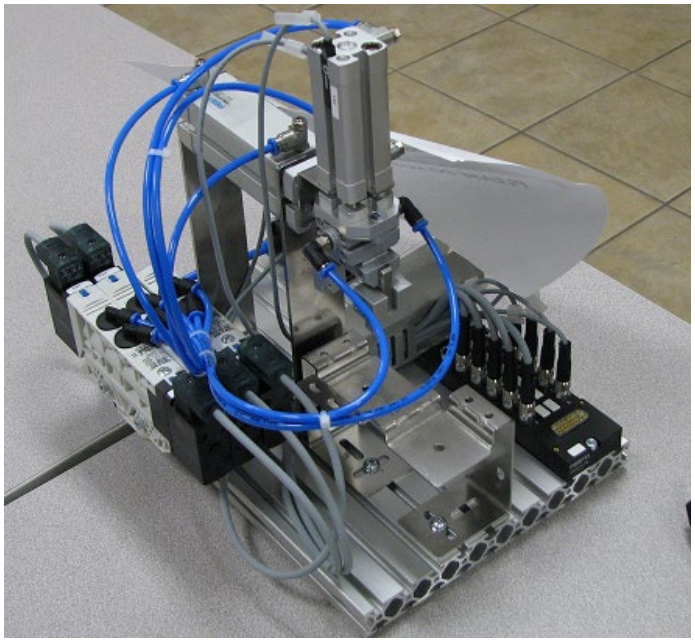
  Open loop because there is no feedback used

- What is another name for this type of control system?

  ***Non-servo* or *Open-loop* control system**

- What does this type of motion control use to control the length of travel along each axis?

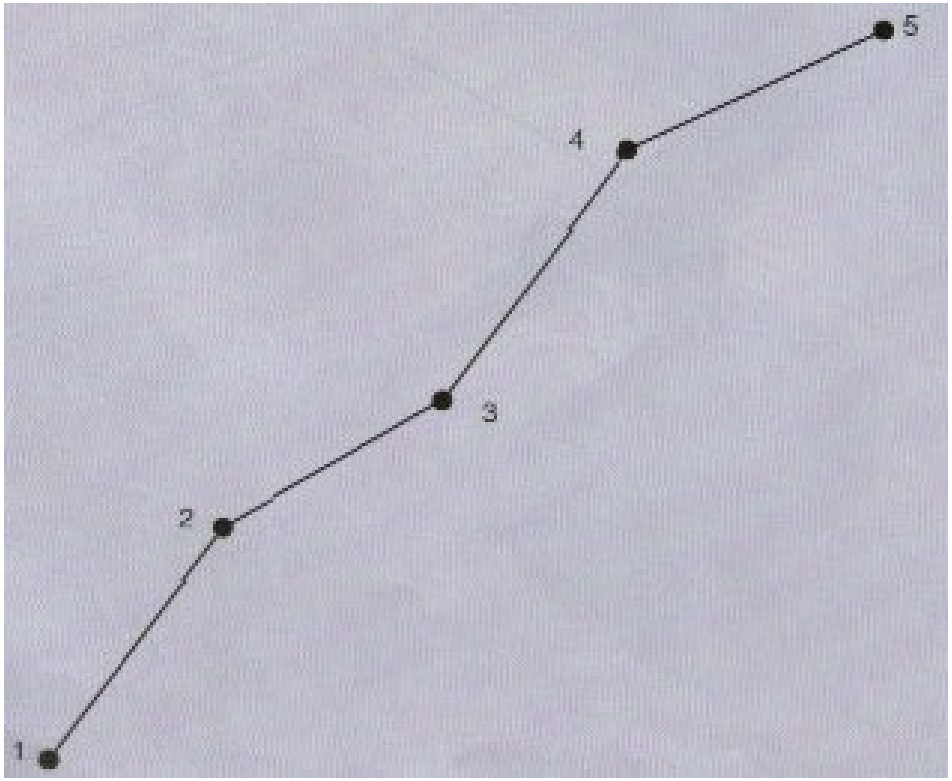
  End stops* and/or *limit switches



Motion Control

Point-to-point motion control

- Moves robot arm through a number of programmed points
- Each point position recorded and stored in memory
- Robot steps through the points as recorded
- The path is a series of straight lines connecting these points



Motion Control

Point-to-point motion control (continued)

- ***End stops*** are used for redundant safety or crash limitation
- Capable of storing ***hundreds of points***
- ***Acceleration***, ***deceleration***, and ***velocity (speed)*** – separate control

Motion Control

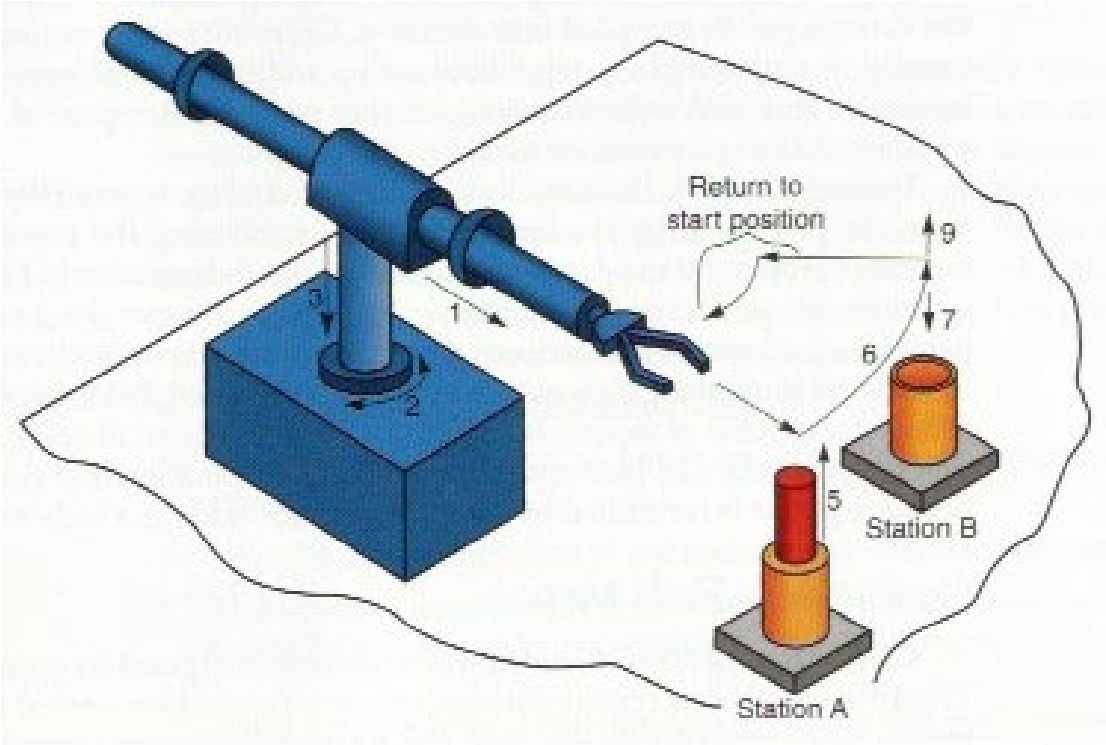
Point-to-point motion control (continued)

- Is point-to-point motion control a form of open or closed-loop control? Why?
  Open loop because as with pick-and-place, there is no feedback used

- What device may be used to control velocity, acceleration, and deceleration?
  Tachometer

Home position is arm "back / returned" with the gripper "open":

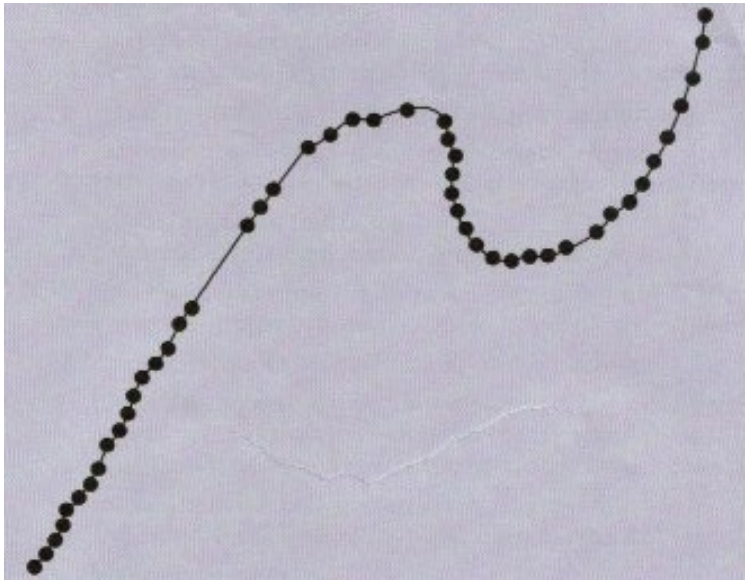
1. Arm extend - stop at pin 'A'
2. Gripper closed
3. Arm up
4. Arm left - stop above 'B'
5. Arm down
6. Gripper open
7. Arm retract
8. Arm right - "Home position"



Motion Control

Continuous-path motion control

- Similar to, but an extension of Point-to-point motion control
- Looking at the path diagram, how does this control differ?
  - Can include several thousand (not hundred) points
  - More points results in a more continuous and smooth motion
- Other differences
  - Robot typically moves “through” the various points
  - Points for each axis is recorded individually



Programming Methods

Manual programming

- Best suited for programming
  - Pick-and-place
  - Point-to-point
  - Open-loop controllers
- Suitable for:
  - Cost restrictive application
  - Less complex application
- Flexibility is limited
  - Typically two or thee degrees of freedom
  - Typically 10 to a few 100 program points

Programming Methods

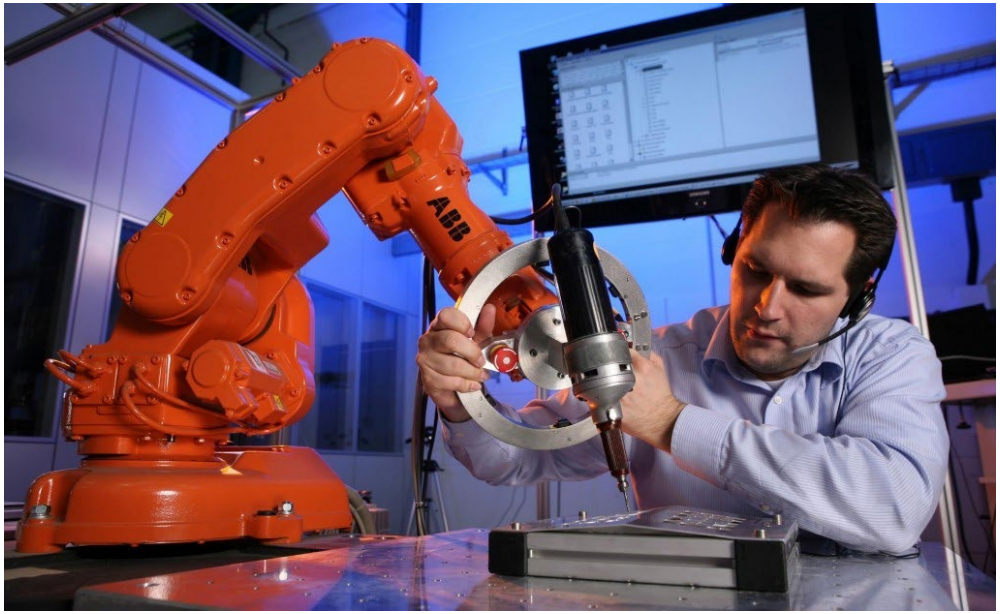
Teach pendant programming

- Operator virtually “leads” the robot through the various positions
- Points are recorded to create a point-to-point path
- Most popular and widely used programming method
  - Does not require a skilled computer programmer
  - Convenient
  - Simple to learn
  - Suitable for multiple industry tasks
- Must be “online” for this method

Programming Methods

Walk-through programming

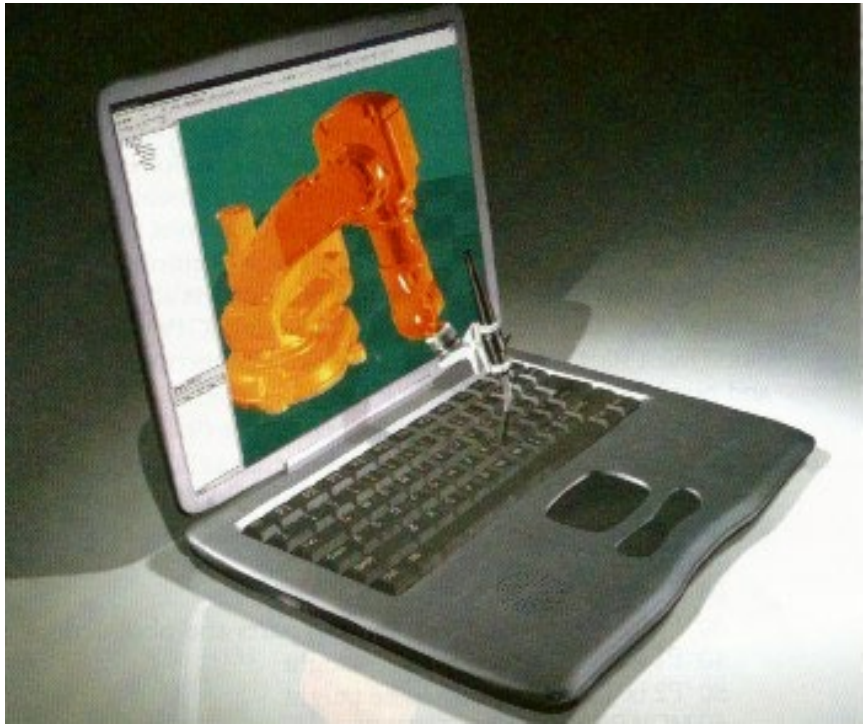
- An “experienced” operator “leads” robot through the various positions
- Points are recorded to create a point-to-point path
- More points – more smooth and continuous motion
- Operator does not have to have computer experience
- Operator must be highly skilled in the precise motion of tasks being programmed
- Must be “online” for this method of programming



Programming Methods

Programming using Computer

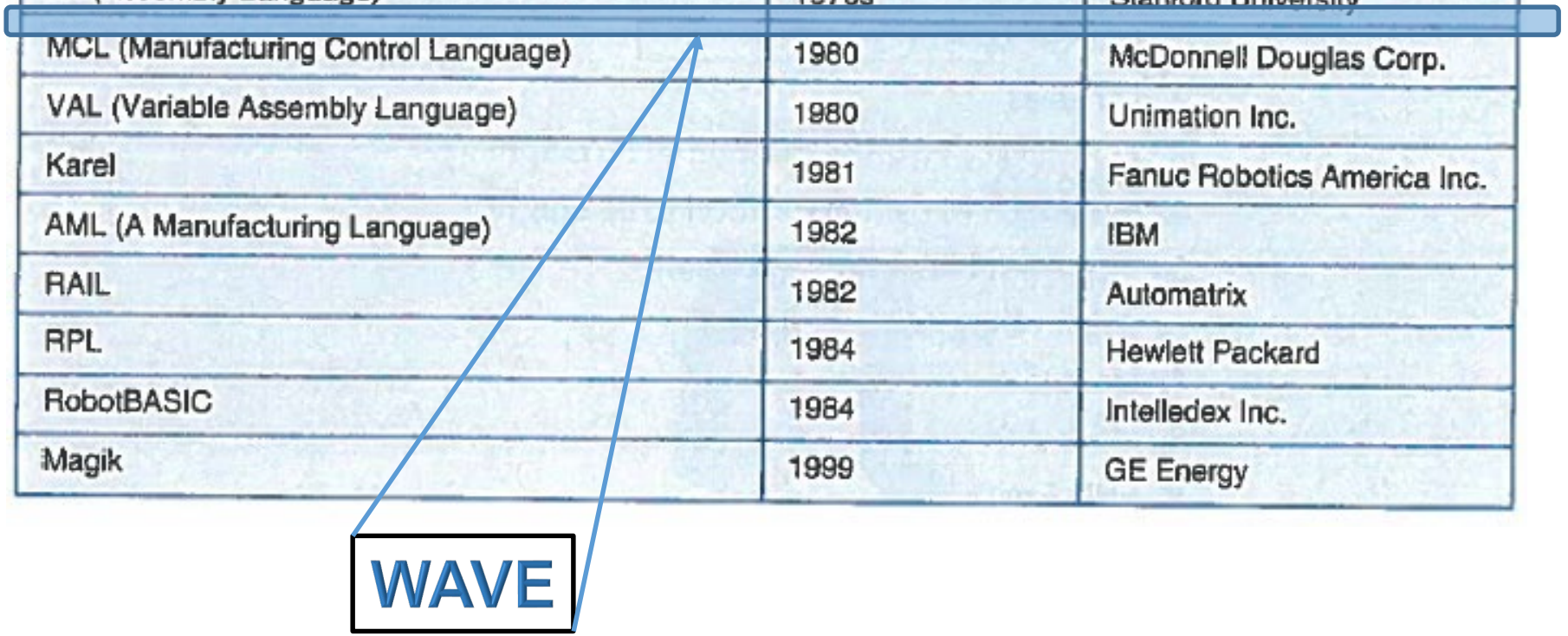
- Can be accomplished with the robot “online” or “off-line”
- Provides more flexibility
  - New program can be written or modified while another is running production
- Computer language allows more complex operations
- Requires a highly experience programmer



Programming Methods

Programming using Computer (continued)

Common Programming Languages		
Robotics Programming Language	**Year of Development**	**Originator**
SAIL (Stanford Artificial Intelligence Language)	1968	Stanford University
AL (Assembly Language)	1970s	Stanford University
MCL (Manufacturing Control Language)	1980	McDonnell Douglas Corp.
VAL (Variable Assembly Language)	1980	Unimation Inc.
Karel	1981	Fanuc Robotics America Inc.
AML (A Manufacturing Language)	1982	IBM
RAIL	1982	Automatrix
RPL	1984	Hewlett Packard
RobotBASIC	1984	Intelledex Inc.
Magik	1999	GE Energy

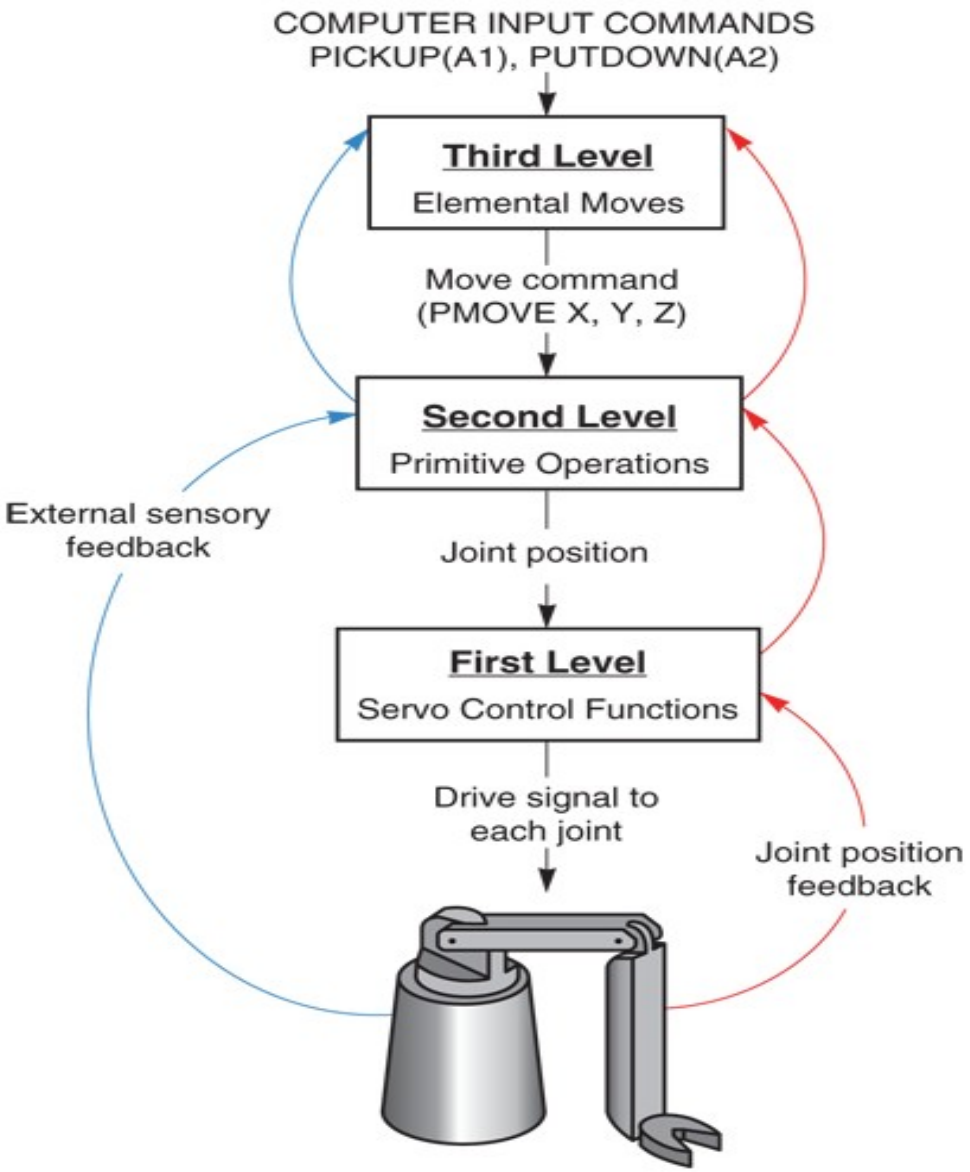


Types of Programming

Two primary types of programming

- Hierarchical control programming
- Task-Level programming

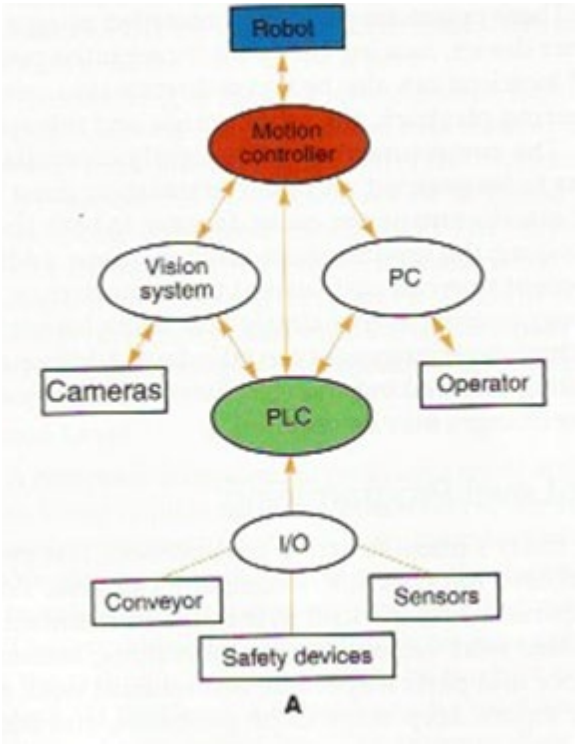
Hierarchical Control Programming

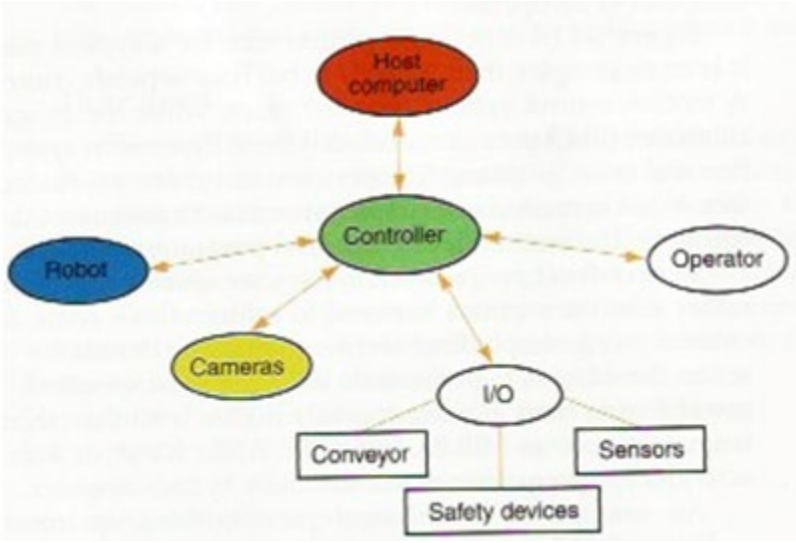


Types of Programming

Task-Level control programming

- User specified task goals – rather than programming motion
- Many activities can be programmed by computer
  - Responding to sensors and other inputs
- "User-friendly" programming





Types of Programming

Task-Level control programming (continued)

Common Task-level Programming Functions	
Function	**Programming**
Robot Movements	The first step in creating the robot program is to select the sequence of movements from a menu. For example, the MOVE statement tells the robot to pick up parts from one location and place them in another location. This type of sequence may be used to move a part from a conveyor belt to a pallet.
Location Information	A location database stores work cell locations. The height of each location, approach, and departure are taught or modified by making the appropriate selections and entries. The robot's speed, type of motion, and other details are also selected and stored.
Palletizing	The PALLET statement is commonly used in automatic palletizing functions. The operator can define spacing and the number of pallet locations.
Visual Sensors	The type of visual sensors needed can be specified by making the appropriate selections and entries on the program interface. For example, a visual inspection sensor can locate and evaluate parts, while a visual guidance sensor may track and orient the parts along a conveyor belt.
General Control Functions	From the control panel screen, the operator can start the operation, slow the speed of the robot, and step the robot through its motions to be sure the program is performing as intended. The control panel screen may also be used for debugging and cell control.

Types of Programming

What are the three levels of hierarchical control?

- **Main control**
- **Path control**
- **Actuator control**

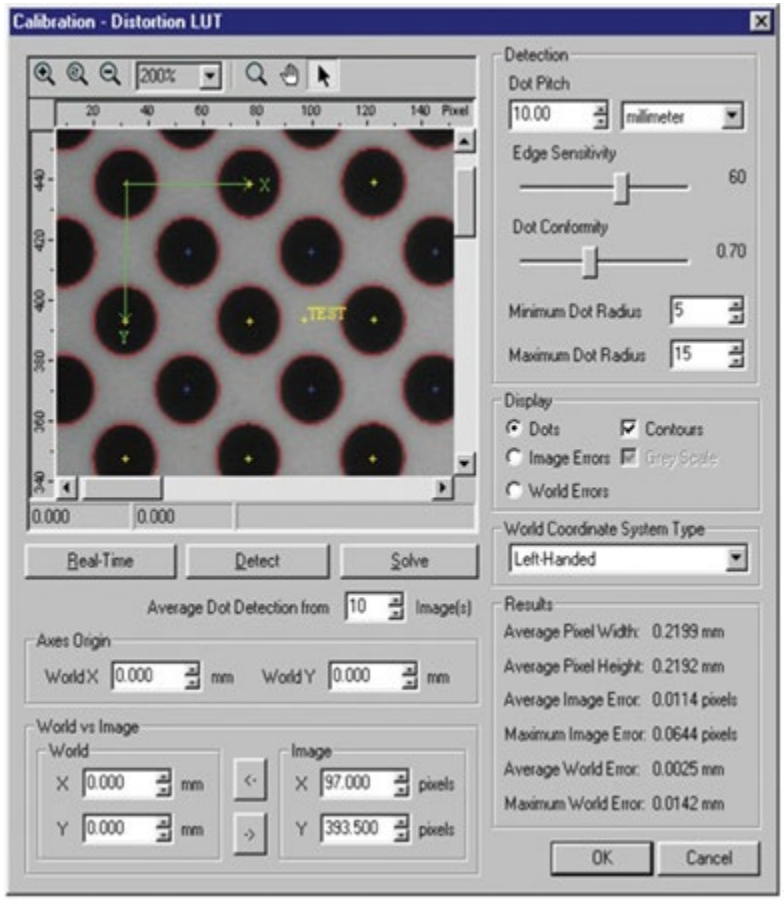
What type of control system using sensory feedback control?

- **Closed-loop or servo**

What type of programming is considered more “user friendly” and simplifies a complex set of instructions for a typical work cell?

- **Task-Level Control Programming**

Voice and Vision Recognition



A

Vision system calibrating for part recognition and locating

B

Vision system performing an inspection on a Integrated Circuit (IC) footprint