



**SATHYABAMA**

INSTITUTE OF SCIENCE AND TECHNOLOGY  
(DEEMED TO BE UNIVERSITY)

Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE  
[www.sathyabama.ac.in](http://www.sathyabama.ac.in)

**SCHOOL OF MECHANICAL ENGINEERING**  
**DEPARTMENT OF MECHANICAL ENGINEERING**

**UNIT – I – INTRODUCTION & ROBOT KINEMATICS – SPR1606**

## UNIT I INTRODUCTION AND ROBOTIC KINEMATICS

### Definition of a Robot

- "A reprogrammable, multifunctional manipulator designed to move material, parts, tools, or specialized devices through various programmed motions for the performance of a variety of tasks" .

Or a simpler version

- *An automatic device that performs functions normally ascribed to humans or a machine in the form of a human.*

### A general-purpose, programmable machine possessing certain anthropomorphic characteristics

- ☐ Hazardous work environments
- ☐ Repetitive work cycle
- ☐ Consistency and accuracy
- ☐ Difficult handling task for humans
- ☐ Multi shift operations
- ☐ Reprogrammable, flexible
- ☐ Interfaced to other computer systems

### Need of Industrial Robots

- Repetitive tasks that robots can do 24/7.
- Robots never get sick or need time off.
- Robots can do tasks considered too dangerous for humans.
- Robots can operate equipment to much higher precision than humans.
- May be cheaper over the long term
- May be able to perform tasks that are impossible for humans

Robots are also used for the following tasks:

- Dirty Tasks
- Repetitive tasks
- Dangerous tasks
- Impossible tasks
- Robots assisting the handicapped.

First, they are hardworking and reliable. They can do dangerous work or work that is very boring or tiring for humans. They can work around the clock without complaining and without needing rest, food or vacations. And robots can go places that humans cannot, such as the surface of Mars, deep under the ocean or inside the radioactive parts of a nuclear power plant

## Asimov's Laws of Robotics (1942)

- A robot may not injure a human being, or, through inaction, allow a human being to come to harm
- A robot must obey orders given it by human beings, except where such orders would conflict with the First Law.
- A robot must protect its own existence as long as such protection does not conflict with the First or Second Law.

## Robot Anatomy

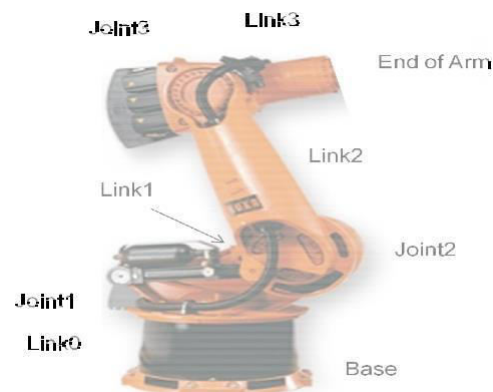


Fig 1.1 Robot Anatomy

## Main Components of Industrial Robots

Arm or Manipulator

End effectors

Drive Mechanism

Controller

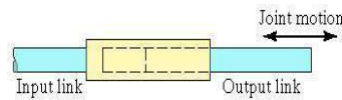
Custom features: e.g. sensors and transducers

Manipulator consists of joints and links

- ✓ Joints provide relative motion
- ✓ Links are rigid members between joints
- ✓ Various joint types: linear and rotary
- ✓ Each joint provides a “degree-of-freedom”
- ✓ Body-and-arm – for positioning of objects in the robot's work volume
- ✓ Wrist assembly – for orientation of objects

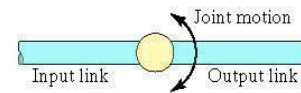
☐ **Translational motion**

☐ **(type L)**

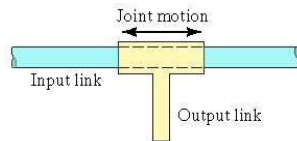


**Linear joint**

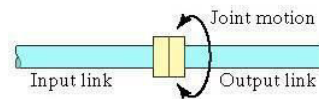
☐ **Rotational joint (type R)**



☐ **Orthogonal joint (type O)**



☐ **Twisting joint (type T)**



☐ **Rotary motion**

☐ **Revolving joint (type V)**

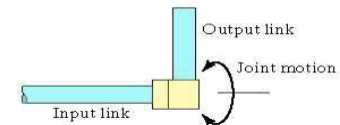


Fig 1.2 Robot Joints

### Classification of robots based on robots configuration Polar Coordinate Body-and-Arm Assembly

Consists of a sliding arm (L joint) actuated relative to the body, which can rotate about both a vertical axis (T joint) and horizontal axis (R joint) Notation TRL:

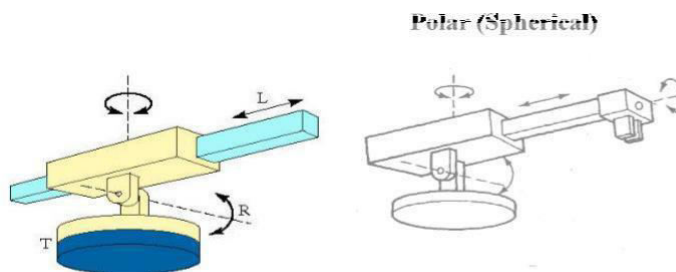


Fig 1.3 Polar Configuration

### Cylindrical Body-and-Arm Assembly

Consists of a vertical column, relative to which an arm assembly is moved up or down. The arm can be moved in or out relative to the column Notation TLO:

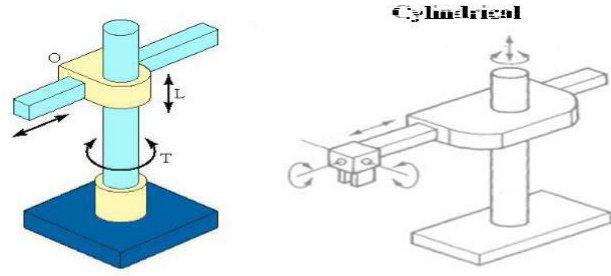


Fig 1.4 Cylindrical Configuration

### Cartesian coordinate Body-and-Arm Assembly

Consists of three sliding joints, two of which are orthogonal other names include rectilinear robot and x-y-z robot Notation LOO:

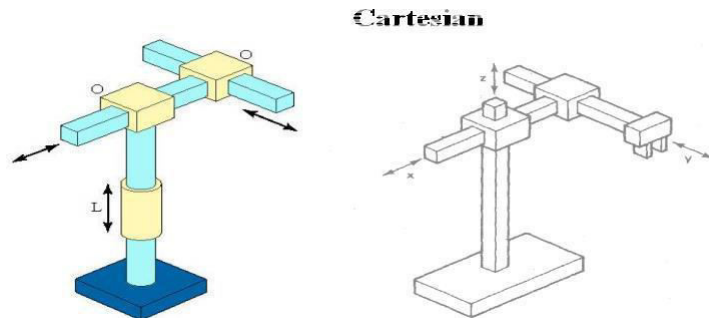


Fig 1.5 Cartesian Configuration

### Jointed-Arm Robot

Similar in appearance to human arm Rotated base, shoulder joint, elbow joint, wrist joint.

Notation TRR:

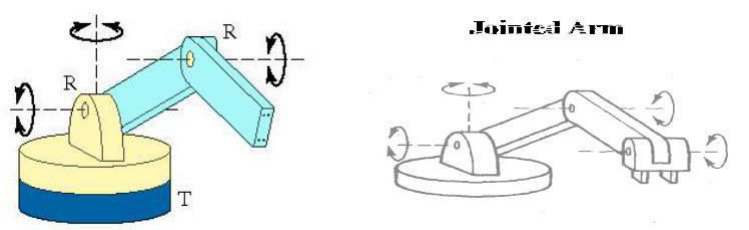


Fig 1.6 Jointed Arm Configuration

## Robot Wrist Configurations

- ☐ Wrist assembly is attached to end-of-arm
- ☐ End effectors is attached to wrist assembly
- ☐ Function of wrist assembly is to orient end effectors
  - ☐ Body-and-arm determines global position of end effectors
- ☐ Two or three degrees of freedom:
  - ☐ Roll
  - ☐ Pitch
  - ☐ Yaw
- ☐ Notation :RRT

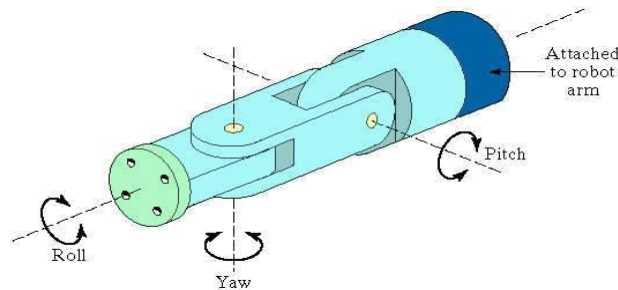


Fig1. 7 Gripper Configurations

- typically has 3 degrees of freedom
  - *Roll* involves rotating the wrist about the arm axis
  - *Pitch* up-down rotation of the wrist
  - *Yaw* left-right rotation of the wrist
- End effectors is mounted on the wrist

## Six Degree of Freedom of a Robot

Three arm and body movement and three wrist movements.

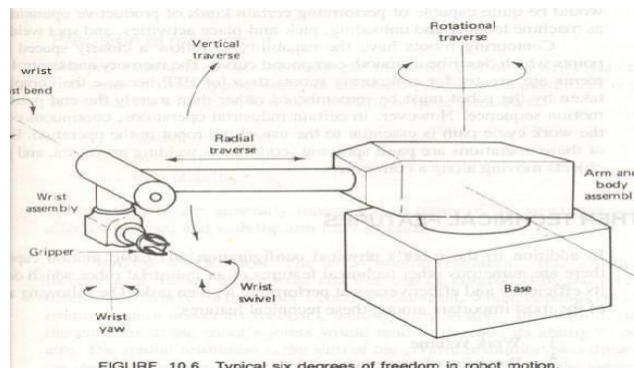


Fig 1.8 Six Degrees of Freedom Robot Manipulator

## Work volume of a robot

*Spatial region within which the end of the robot's wrist can be manipulated* Determined by

- Physical configurations
- Size
- Number of axes
- The robot mounted position (overhead gantry, wall-mounted, floor mounted, on tracks)
- Limits of arm and joint configurations
- The addition of an end-effectors can move or offset the entire work volume

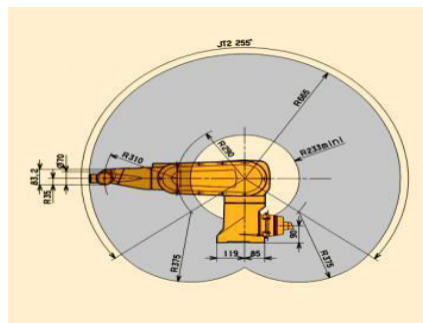


Fig 1.9 Robot Work volume

- Space within which a robot can operate.
- Determined by its configuration, size and the limits of its arms. Various shapes of work volume of robots
  - Polar coordinates – Partial Sphere
  - Cylindrical coordinates – Cylindrical
  - Cartesian – Rectangular
  - Jointed arm – Irregular

## Precision of movement

**The precision with which the robot can move the end of its wrist**

1. Spatial resolution
2. Accuracy
3. Repeatability

### Spatial resolution

*Smallest increment of motion at the wrist end that can be controlled by the robot* Depends on the position control system, feedback measurement, and mechanical accuracy

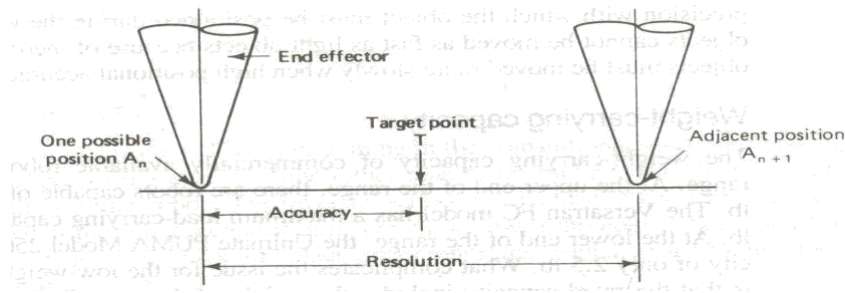


Fig 1.10 Spatial Resolution

Smallest increment of motion

Depends on the control system and feedback

$$\text{Control resolution} = \frac{\text{range}}{\text{Control increments}}$$

### Accuracy

*Capability to position the wrist at a target point in the work volume*

- One half of the distance between two adjacent resolution points
- Affected by mechanical Inaccuracies
- Manufacturers don't provide the accuracy (hard to control)

The ability of a robot to go to the specified position without making a mistake. Closely related to spatial resolution

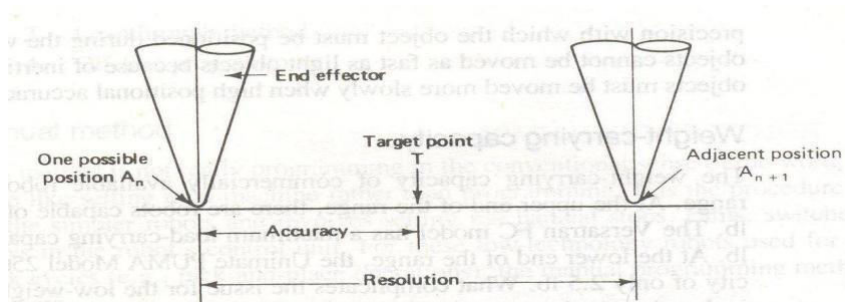


Fig 1.11 Accuracy

### Repeatability

*Ability to position back to a point that was previously taught*

- Repeatability errors form a random variable.
- Mechanical inaccuracies in arm, wrist components
- Larger robots have less precise repeatability values

Ability to position a wrist back to the previously visited point.

### Load Carrying Capacity

- The lifting capability provided by manufacturer doesn't include the weight of the end effectors
- Usual Range 2.5lb-2000lb



- Condition to be satisfied:

Load Capability > Total Wt. of work piece + Wt. of end effectors + Safety range

## Speed of movement

*Speed with which the robot can manipulate the end effectors* Acceleration/deceleration times are crucial for cycle time. •Determined by

Weight of the  
object Distance  
moved

Precision with which object must be positioned

The amount of distance per unit time at which the robot can move. Speed of the end effectors

Determined by the weight of the object

## End Effectors

- ☐ *The special tooling for a robot that enables it to perform a specific task*
- ☐ *Two types:*
  - ☐ *Grippers – to grasp and manipulate objects (e.g., parts) during work cycle*
  - ☐ *Tools – to perform a process, e.g., spot welding, spray painting*

*Device attached to the robot's wrist to perform a specific task*

## Grippers

A two-finger mechanical gripper for grasping rotational parts

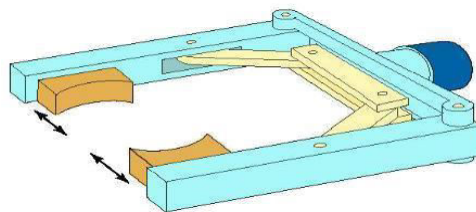


Fig 1.12 Robot Mechanical Gripper

- ☐ Dual grippers
- ☐ Interchangeable fingers
- ☐ Sensory feedback
  - ☐ To sense presence of object
  - ☐ To apply a specified force on the object
- ☐ Multiple fingered gripper (similar to human hand)

- ☐ Standard gripper products to reduce the amount of custom design required
- Mechanical Grippers
- Suction cups or vacuum cups
- Magnetized grippers
- Hooks
- Scoops (to carry fluids)
- Spot Welding gun
- Arc Welding tools
- Spray painting gun
- Drilling Spindle
- Grinders, Wire brushes
- Heating torches

## **Sensors in robotics**

### **Two basic categories of sensors used in industrial robots:**

1. Internal - used to control position and velocity of the manipulator joints
2. External - used to coordinate the operation of the robot with other equipment in the work cell
  - ☐ Tactile - touch sensors and force sensors
  - ☐ Proximity - when an object is close to the sensor
  - ☐ Optical -
  - ☐ Machine vision
  - ☐ Other sensors - temperature, voltage, etc.

### **Types of sensors:**

Tactile sensors (touch sensors, force sensors, tactile array sensors)

Proximity and range sensors (optical sensors, acoustical sensors, electromagnetic sensors)

Miscellaneous sensors (transducers and sensors which sense variables such temperature,

pressure, fluid flow, thermocouples, voice sensors) Machine vision systems

### **Uses of sensors:**

Safety monitoring

Interlocks in work cell control

Part inspection for quality control

Determining positions and related information about objects

### Desirable features of sensors:

- ✓ Accuracy
- ✓ Operation range Speed of response Calibration
- ✓ Reliability
- ✓ Cost and ease of operation

### World Coordinate System

- Origin and axes of robot manipulator are defined relative to the robot base

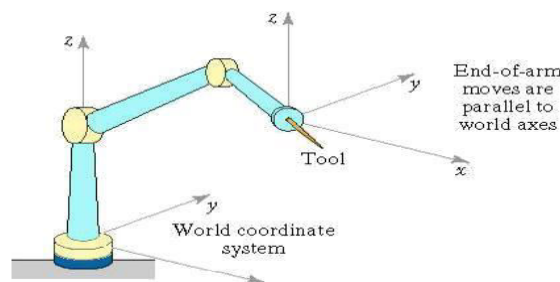


Fig 1.13 World Coordinate

### System Tool Coordinate System

- Alignment of the axis system is defined relative to the orientation of the wrist faceplate (to which the end effector is attached)

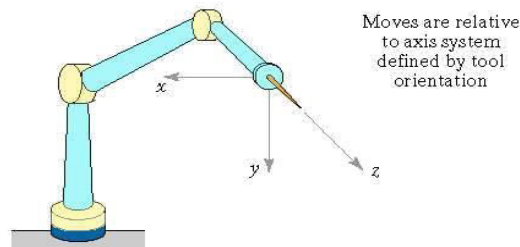


Fig 1.14 Tool Coordinate System

### Direct and inverse kinematics

Direct (forward) kinematics is a mapping from joint coordinate space to space of end-effectors positions. That is we know the position of all (or some) individual joints and we are looking for the position of the end effectors. Mathematically:  $\sim q \rightarrow T(\sim q)$  direct kinematics could be immediately used in coordinate measurement systems. Sensors in the joints will inform us about the relative position of the links, joint coordinates. The goal is to calculate the position of the reference point of the measuring system.

Inverse kinematics is a mapping from space of end-effectors positions to joint coordinate space. That is we know the position of the end effectors and we are looking for the coordinates

of all individual joints. Mathematically:  $T \rightarrow \sim q(T)$  Inverse kinematics is needed in robot control, one knows the required position of the gripper, but for control the joint coordinates are needed.

### Forward Kinematics (angles to position)

What you are given:    The length of each link  
                                 The angle of each joint

What you can find:     The position of any point  
                                 (i.e. it's (x, y, z) coordinates)

### Inverse Kinematics (position to angles)

What you are given:    The length of each link  
                                 The position of some point on the robot

What you can find:     The angles of each joint needed to obtain  
                                 that position

### Forward Kinematics

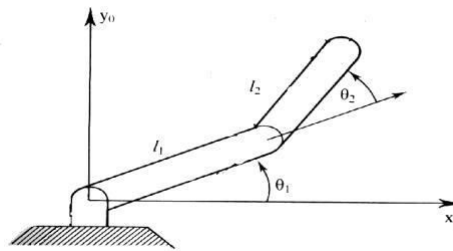


Fig 1.15 Forward Kinematics

### The Situation:

You have a robotic arm that starts out aligned with the  $x_0$ -axis. The first link to move by  $U_1$  and the second link to move by  $U_2$

#### 1. Geometric Approach

This might be the easiest solution for the simple situation. However, notice that the angles are measured relative to the direction of the previous link. (The first link is the exception. The angle is

measured relative to its initial position.) For robots with more links and whose arm extends into 3 dimensions the geometry gets much more tedious.

#### 2. Algebraic Approach

Involves coordinate transformations.

You are having a three link arm that starts out aligned in the x-axis. Each link has lengths  $l_1$ ,  $l_2$ ,  $l_3$ , respectively. You tell the first one to move by  $U_1$ , and so on as the diagram suggests.

Find the Homogeneous matrix to get the position of the yellow dot in the  $X^0 Y^0$  frame.

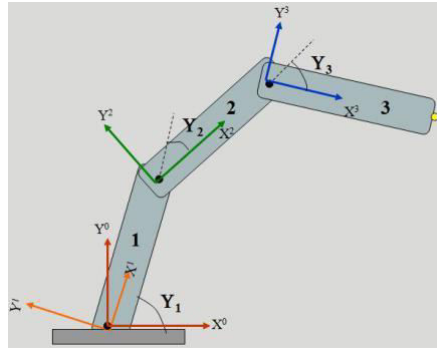


Fig 1.16 Kinematic Manipulator

$$H = R_z(U_1) * T_{x1}(l_1) * R_z(U_2) * T_{x2}(l_2) * R_z(U_3)$$

i.e. Rotating by  $U_1$  will put you in the  $X^1 Y^1$  frame.

Translate in the along the  $X^1$  axis by  $l_1$ .

Rotating by  $U_2$  will put you in the  $X^2 Y^2$  frame.

and so on until you are in the  $X^3 Y^3$  frame.

The position of the yellow dot relative to the  $X^3 Y^3$  frame is

$(l_3, 0)$ . Multiplying  $H$  by that position vector will give you the coordinates of the yellow point relative the  $X^0 Y^0$  frame.

$$H = R_z(U_1) * T_{x1}(l_1) * R_z(U_2) * T_{x2}(l_2) * R_z(U_3) * T_{x3}(l_3)$$

This takes you from the  $X^3 Y^3$  frame to the  $X^4 Y^4$  frame.

The position of the yellow dot relative to the  $X^4 Y^4$  frame is  $(0, 0)$ .

$$\begin{matrix} X & 0 \\ Y & 0 \\ Z & 0 \\ 1 & 1 \end{matrix} \quad \begin{matrix} \\ \\ H \\ \\ \end{matrix}$$

Notice that multiplying by the  $(0, 0, 0, 1)$  vector will equal the last column of the  $H$  matrix.

## Inverse Kinematics of a Two Link Manipulator

### From Position to Angles

**Given:**  $l_1, l_2, x, y$

**Find:**  $U_1, U_2$

**Redundancy:**

A unique solution to this problem does not exist. Notice, that using the “givens” two solutions are possible.

\*\*\*\*\*



**SATHYABAMA**

INSTITUTE OF SCIENCE AND TECHNOLOGY  
(DEEMED TO BE UNIVERSITY)

Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE

[www.sathyabama.ac.in](http://www.sathyabama.ac.in)

**SCHOOL OF MECHANICAL ENGINEERING**

**DEPARTMENT OF MECHANICAL ENGINEERING**

**UNIT – II – ROBOT DRIVES AND CONTROLS – SPR1606**

## UNIT II ROBOT DRIVES AND CONTROL

### Robot Control Systems

**Limited sequence control** – pick-and-place operations using mechanical stops to set positions

**Playback with point-to-point control** – records work cycle as a sequence of points, then plays back the sequence during program execution

**Playback with continuous path control** – greater memory capacity and/or interpolation capability to execute paths (in addition to points)

**Intelligent control** – exhibits behavior that makes it seem intelligent, e.g., responds to sensor inputs, makes decisions, communicates with humans

### Robot Control System

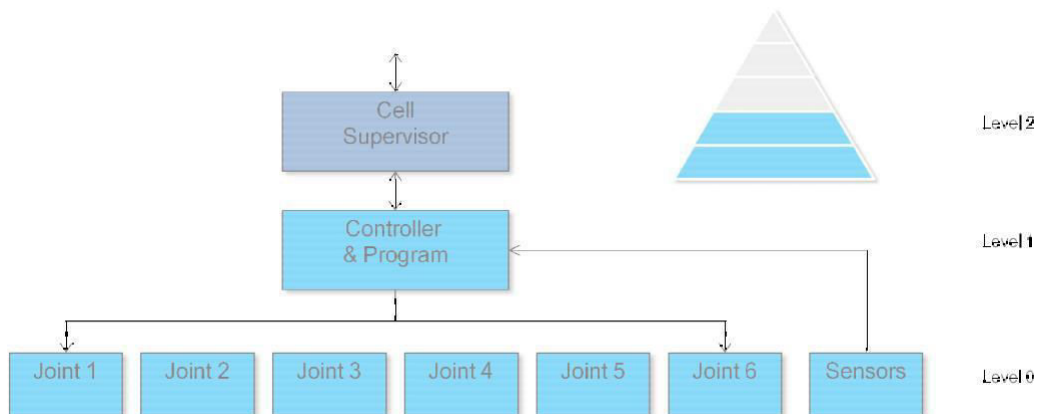


Fig 2.1 Robot Control system

### Motion Control

- Path control - how accurately a robot traces a given path (critical for gluing, painting, welding applications);
- Velocity control - how well the velocity is controlled (critical for gluing, painting applications)
- **Types of control path:**
  - Point to point control (used in assembly, palletizing, machine loading);
  - continuous path control/walkthrough (paint spraying, welding).
  - controlled path (paint spraying, welding)



- Limited sequence control – pick-and-place operations using mechanical stops to set positions
- Playback with point-to-point control – records work cycle as a sequence of points, then plays back the sequence during program execution
- Playback with continuous path control – greater memory capacity and/or interpolation capability to execute paths (in addition to points)
- Intelligent control – exhibits behaviour that makes it seem intelligent, e.g., responds to sensor inputs, makes decisions, communicates with humans

## Robot Control System

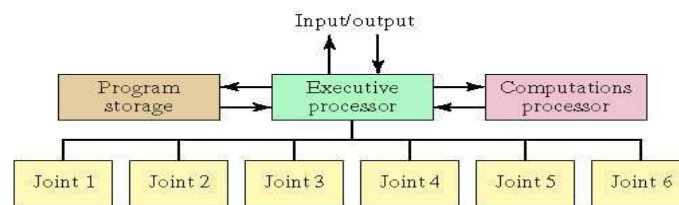


Fig 2.2 Robot Control System

Robot control consists in studying how to make a robot manipulator perform a task. Control design may be divided roughly in the following steps:

- Familiarization with the physical system under consideration,
- Modeling.

- Control specifications.

Control specifications Definition of control objectives: • Stability • Regulation • Trajectory tracking (motion control) • Optimization.

- Stability. Consists in the property of a system by which it goes on working at certain regime or ‘closely’ to it ‘forever’. – Lyapunov stability theory. – Input-output stability theory. In the case when the output  $y$  corresponds to the joint position  $q$  and velocity  $\dot{q}$ .
- Regulation “Position control in joint coordinates”

Trajectory tracking “Tracking control in joint coordinates”

## Control Methods

- **Non Servo Control**
  - implemented by setting limits or mechanical stops for each joint and sequencing the actuation of each joint to accomplish the cycle
  - end point robot, limited sequence robot, bang-bang robot
  - No control over the motion at the intermediate points, only end points are known

- Programming accomplished by
  - setting desired sequence of moves
  - adjusting end stops for each axis accordingly

### ***Servo Control***

- Point to point Control
- Continuous Path Control
- Closed Loop control used to monitor position, velocity (other variables) of each joint
- the sequence of moves is controlled by a “sequencer”, which uses feedback received
  - from the end stops to index to next step in the program
- Low cost and easy to maintain, reliable
- relatively high speed
- repeatability of up to 0.01 inch
- limited flexibility
- typically hydraulic, pneumatic drives

### **Point-to-Point Control**

- Only the end points are programmed, the path used to connect the end points are computed by the controller
- user can control velocity, and may permit linear or piece wise linear motion
- Feedback control is used during motion to ascertain that individual joints have achieved desired location
- Often used hydraulic drives, recent trend towards servomotors
- loads up to 500lb and large reach
- Applications
  - pick and place type operations
  - palletizing
  - machine loading
- In addition to the control over the endpoints, the path taken by the end effectors can be controlled
- Path is controlled by manipulating the joints throughout the entire motion, via closed loop control
- Applications:
  - spray painting, polishing, grinding, arc welding

### **Sensors in Robotics**

Two basic categories of sensors used in industrial robots:

1. Internal - used to control position and velocity of the manipulator joints

2. External - used to coordinate the operation of the robot with other equipment in the work cell
  - Tactile - touch sensors and force sensors
    - ☐ Proximity - when an object is close to the sensor
    - ☐ Optical -
    - ☐ Machine vision
    - ☐ Other sensors - temperature, voltage, etc.
  - ☐ Electric Drive system
    - ☐ Uses electric motors to actuate individual joints
    - ☐ Preferred drive system in today's robots
    - ☐ Electric motor (stepper, servo, less strength, better accuracy and repeatability)
  - ☐ Hydraulic Drive system
    - ☐ Uses hydraulic pistons and rotary vane actuators
    - ☐ Noted for their high power and lift capacity
    - ☐ Hydraulic (mechanical, high strength)
  - ☐ Pneumatic Drive system
    - ☐ Typically limited to smaller robots and simple material transfer applications
    - ☐ Pneumatic (quick, less strength)
  - ☐ **Hydraulic Drive system**
    - High strength and high speed
    - Large robots, Takes floor space
    - Mechanical Simplicity
    - Used usually for heavy payloads
  - ☐ **Electric Motor (Servo/Stepper) Drive system**
    - High accuracy and repeatability
    - Low cost
    - Less floor space
    - Easy maintenance
    - Smaller units, quick assembly
    - High cycle rate
    - Easy maintenance

### **Electro hydraulic servo valves**

An electro hydraulic servo valve (EHSV) is an electrically operated valve that controls how hydraulic fluid is ported to an actuator. Servo valves and servo-proportional valves are operated by transforming a changing analogue or digital input signal into a smooth set of movements in

a hydraulic cylinder. Servo valves can provide precise control of position, velocity, pressure and force with good post movement damping characteristics.

In its simplest form a servo or a servomechanism is a control system which measures its own output and forces the output to quickly and accurately follow a command signal, see Figure 1-1. In this way, the effect of anomalies in the control device itself and in the load can be minimized as well as the influence of external disturbances. A servomechanism can be designed to control almost any physical quantities, e.g. motion, force, pressure, temperature, electrical voltage or current.

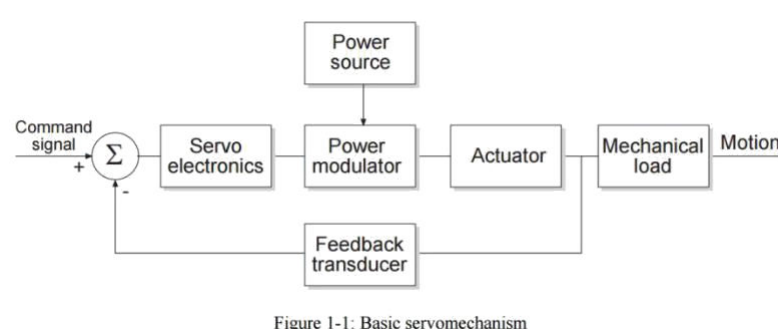


Figure 1-1: Basic servomechanism

Fig 2.3 Basic Servo Mechanics

Capabilities of electro-hydraulic servos When rapid and precise control of sizeable loads is required an electro-hydraulic servo is often the best approach to the problem. Generally speaking, the hydraulic servo actuator provides fast response, high force and short stroke characteristics. The main advantages of hydraulic components are.

- Easy and accurate control of work table position and velocity
- Good stiffness characteristics
- Zero back-lash
- Rapid response to change in speed or direction
- Low rate of wear

There are several significant advantages of hydraulic servo drives over electric motor drives:

- ♦ Hydraulic drives have substantially higher power to weight ratios resulting in higher machine frame resonant frequencies for a given power level.

- ◆Hydraulic actuators are stiffer than electric drives, resulting in higher loop gain capability, greater accuracy and better frequency response.
- ◆Hydraulic servos give smoother performance at low speeds and have a wide speed range without special control circuits.
- ◆Hydraulic systems are to a great extent self-cooling and can be operated in stall condition indefinitely without damage.
- ◆Both hydraulic and electric drives are very reliable provided that maintenance is followed.
- ◆Hydraulic servos are usually less expensive for system above several horsepower, especially if the hydraulic power supply is shared between several actuators.

### **End Effectors Types**

- 1) Standard Grippers (Angular and parallel, Pneumatic, hydraulic, electric, spring powered, Power-opened and Spring-closed)
- 2) Vacuum Grippers (Single or multiple, use venturi or vacuum pump)
- 3) Vacuum Surfaces (Multiple suction ports, to grasp cloth materials, flat surfaces, sheet material)
- 4) Electromagnetic Grippers (often used in conjunction with standard grippers)
- 5) Air-Pressure Grippers (balloon type)
  1. Pneumatic fingers
  2. Mandrel grippers
  3. Pin grippers
- 6) Special Purpose Grippers (Hooking devices, custom positioners or tools)
- 7) Welding (MIG /TIG, Plasma Arc, Laser, Spot)
- 8) Pressure Sprayers (painting, water jet cutting, cleaning)
- 9) Hot Cutting type (laser, plasma, de-flashers-hot knife)
- 10) Buffing/Grinding/De-burring type
- 11) Drilling/Milling type
- 12) Dispensing type (adhesive, sealant, foam)

## Mechanical Grippers

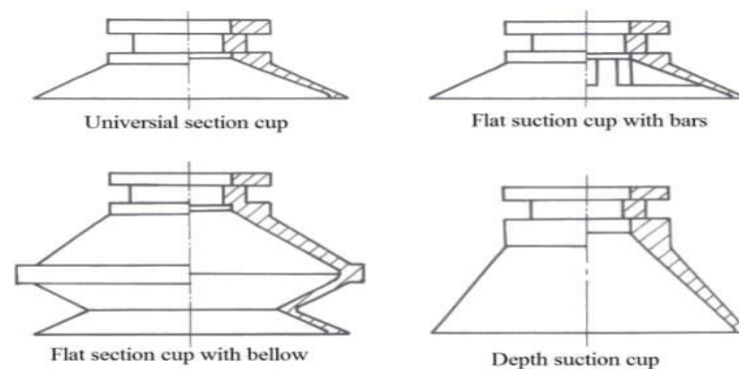
Mechanical grippers are used to pick up, move, place, or hold parts in an automated system. They can be used in harsh or dangerous

**VACUUM GRIPPERS:** for non-ferrous components with flat and smooth surfaces, grippers can be built using standard vacuum cups or pads made of rubber-like materials. Not suitable for components with curved surfaces or with holes.

### Vacuum grippers

Vacuum-grippers become in suction cups, the suction cups is made of rubber. The suction cups are connected through tubes with under pressure devices for picking up items and for releasing items air is pumped out into the suction cups. The under pressure can be created with the following devices:

The vacuum grippers use suction cups (vacuum cups) as pick up devices. There are different types of suction cups and the cups are generally made of polyurethane or rubber and can be used at temperatures between -50 and 200 °C. The suction cup can be categorized into four different types; universal suction cups, flat suction cups with bars, suction cups with bellow and depth suction cups as shown in figure 3.



*Figure 3 Different types of suction cups, picture taken from [5] page 204*

Fig 2.4 Suction Cups

The universal suction cups are used for flat or slightly arched surfaces. Universal suction cups are one of the cheapest suction cups in the market but there are several disadvantages with this type of suction cups. When the under pressure is too high, the suction cup decreases a lot which leads to a greater wear. The flat suction cups with bars are suitable for flat or flexible items that

need assistance when lifted. These types of suction cups provides a small movement under load and maintains the area that the under pressure is acting on, this reduces the wear of the flat suction cup with bars, this leads to a faster and safer movement. Suction cups with bellows are usually used for curved surfaces, for example when separation is needed or when a smaller item is being gripped and needs a shorter movement. This type of suction cups can be used in several areas but they allow a lot of movement at gripping and low stability with small under pressure. The depth suction cup can be used for surfaces that are very irregular and curved or when an item needs to be lifted over an edge. [5] Items with rough surfaces (surface roughness  $\leq 5 \mu\text{m}$  for some types of suction cups) or items that are made of porous material will have difficulty with vacuum grippers. An item with holes, slots and gaps on the surfaces is not recommended to be handled with vacuum grippers. The air in the suction is sucked out with one of the techniques described earlier, if the material is porous or has holes on its surface; it will be difficult to suck out the air. In such cases the leakage of air can be reduced if smaller suction cups are used. Figure 4 shows different types of suction cups.

**Magnetic Gripper:** used to grip ferrous materials. Magnetic gripper uses a magnetic head to attract ferrous materials like steel plates. The magnetic head is simply constructed with a ferromagnetic core and conducting coils. Magnetic grippers are most commonly used in a robot as end effectors for grasping the *ferrous* materials. It is another type of handling the work parts other than the mechanical grippers and vacuum grippers. Types of magnetic grippers:

The magnetic grippers can be classified into *two common types*, namely:

Magnetic grippers with



Fig 2.5 Magnetic Gripper

**Electromagnets:**

Electromagnetic grippers include a *controller unit* and a *DC power* for handling the materials. This type of grippers is easy to control, and very effective in releasing the part at the end of the operation than the permanent magnets. If the work part gripped is to be released, the polarity level is minimized by the controller unit before the electromagnet is turned off. This process will certainly help in *removing the magnetism* on the work parts. As a result, a best way of releasing the materials is possible in this gripper.

**Permanent magnets:**

The permanent magnets do not require any sort of external power as like the electromagnets for handling the materials. After this gripper grasps a work part, an additional device called as *stripper push – off pin* will be required to separate the work part from the magnet. This device is incorporated at the sides of the gripper.

The advantage of this permanent magnet gripper is that it can be used in hazardous applications like *explosion-proof apparatus* because of no electrical circuit. Moreover, there is no possibility of *spark production* as well.

**Benefits:**

This gripper only requires *one surface* to grasp the materials. The grasping of materials is done *very quickly*.

It does not require *separate designs* for handling different size of materials.

It is capable of grasping materials with *holes*, which is unfeasible in the vacuum grippers.

**Drawbacks:**

The gripped work part has the chance of *slipping out* when it is moving quickly. Sometimes *oil* in the surface can reduce the strength of the gripper.

The *machining chips* may stick to the gripper during unloading.





**SATHYABAMA**

INSTITUTE OF SCIENCE AND TECHNOLOGY  
(DEEMED TO BE UNIVERSITY)

Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE

[www.sathyabama.ac.in](http://www.sathyabama.ac.in)

**SCHOOL OF MECHANICAL ENGINEERING**

**DEPARTMENT OF MECHANICAL ENGINEERING**

**UNIT – III – ROBOT SENSORS – SPR 1606**

## **UNIT III ROBOT SENSORS**

Sensors are devices that can sense and measure physical properties of the environment,

E.g. temperature, luminance, resistance to touch, weight, size, etc. The key phenomenon is transduction

Transduction (engineering) is a process that converts one type of energy to another

### **Transducer**

a device that converts a primary form of energy into a corresponding signal with a different energy form Primary Energy Forms: mechanical, thermal, electromagnetic, optical, chemical, etc.

take form of a sensor or an

Actuator    Sensor    (e.g.,  
thermometer)

a device that detects/measures a signal  
or stimulus acquires information from  
the “real world”

### **Tactile sensing**

Touch and tactile sensor are devices which measures the parameters of a contact between the sensor and an object. This interaction obtained is confined to a small defined region. This contrasts with a force and torque sensor that measures the total forces being applied to an object. In the consideration of tactile and touch sensing, the following definitions are commonly used:

### **Touch Sensing**

This is the detection and measurement of a contact force at a defined point. A touch sensor can also be restricted to binary information, namely touch, and no touch.  
Tactile Sensing

This is the detection and measurement of the spatial distribution of forces perpendicular to a predetermined sensory area, and the subsequent interpretation of the spatial information. A tactile-sensing array can be considered to be a coordinated group of touch sensors.

### **Force/torque sensors**

Force/torque sensors are often used in combination with tactile arrays to provide information for force control. A single force/torque sensor can sense loads anywhere on the distal link of a manipulator and, not being subject to the same packaging constraints as a “skin” sensor, can generally provide more precise force measurements at higher bandwidth. If the geometry of the manipulator link is defined, and if single-point contact can be assumed (as in the case of a robot finger with a hemispherical tip contacting locally convex surfaces), then a force/torque sensor can provide information about the contact location by ratios of forces and moments in a technique called “intrinsic tactile sensing”

### **Proximity sensor**

A proximity sensor is a sensor able to detect the presence of nearby objects without any physical contact. A proximity sensor often emits an electromagnetic field or a beam of electromagnetic radiation (infrared, for instance), and looks for changes in the field or return signal. The object being sensed is often referred to as the proximity sensor's target. Different proximity sensor targets demand different sensors. For example, a capacitive or photoelectric sensor might be suitable for a plastic target; an inductive proximity sensor always requires a metal target. The maximum distance that this sensor can detect is defined "nominal range". Some sensors have adjustments of the nominal range or means to report a graduated detection distance. Proximity sensors can have a high reliability and long functional life because of the absence of mechanical parts and lack of physical contact between sensor and the sensed object.

Proximity sensors are commonly used on smart phones to detect (and skip) accidental touch screen taps when held to the ear during a call. They are also used in machine vibration monitoring to measure the variation in distance between a shaft and its support bearing. This is common in large steam turbines, compressors, and motors that use sleeve-type bearings.

## Ranging sensors

Ranging sensors include sensors that require no physical contact with the object being detected. They allow a robot to see an obstacle without actually having to come into contact with it. This can prevent possible entanglement, allow for better obstacle avoidance (over touch-feedback methods), and possibly allow software to distinguish between obstacles of different shapes and sizes. There are several methods used to allow a sensor to detect obstacles from a distance. Below are a few common methods ranging in complexity and capability from very basic to very intricate. The following examples are only made to give a general understanding of many common types of ranging and proximity sensors as they commonly apply to robotics.

## Sensors used in Robotics

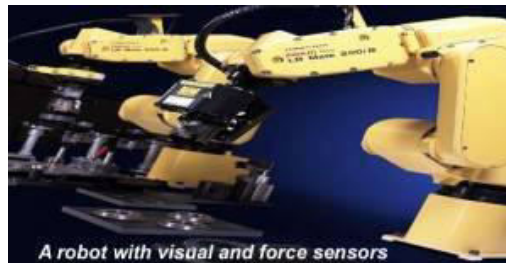


Fig 3.1 Industrial Robot with Sensor

The use of *sensors* in robots has taken them into the next level of creativity. Most importantly, the sensors have increased the performance of robots to a large extent. It also allows the robots to perform several functions like a human being. The robots are even made intelligent with the help of Visual Sensors (generally called as machine vision or computer vision), which helps them to respond according to the situation. The Machine Vision system is classified into six sub-divisions such as Pre-processing, Sensing, Recognition, Description, Interpretation, and Segmentation.

### Different types of sensors:

This type of sensor is capable of pointing out the availability of a component. Generally, the *proximity sensor* will be placed in the robot moving part such as end effector. This sensor will be turned ON at a specified distance, which will be measured by means of feet or millimeters. It is also used to find the presence of a human being in the work volume so that the accidents can be reduced.

### Range Sensor:

Range Sensor is implemented in the end effector of a robot to calculate the distance between the sensor and a work part. The values for the distance can be given by the workers on visual data. It can evaluate the size of images and analysis of common objects. The range is measured using the Sonar receivers & transmitters or two TV cameras.

### Tactile Sensors:

A sensing device that specifies the contact between an object, and sensor is considered as the Tactile Sensor. This sensor can be sorted into two key types namely: Touch Sensor and Force Sensor.



Fig 3.2 Touch Sensor and Force Sensor

The touch sensor has got the ability to sense and detect the touching of a sensor and object. Some of the commonly used simple devices as touch sensors are micro – switches, limit switches, etc. If the end effector gets some contact with any solid part, then this sensor will be handy one to stop the movement of the robot. In addition, it can be used as an inspection device, which has a probe to measure the size of a component.

The force sensor is included for calculating the forces of several functions like the machine loading & unloading, material handling, and so on that are performed by a robot. This sensor will also be a better one in the assembly process for checking the problems. There are several techniques used in this sensor like Joint Sensing, Robot – Wrist Force Sensing, and Tactile Array Sensing.

### Robotic applications of a machine vision system

A machine vision system is employed in a robot for *recognizing* the objects. It is commonly used to perform the *inspection* functions in which the industrial robots are not involved. It is usually mounted in a *high speed production line* for accepting or rejecting the work parts. The rejected work parts will be removed by other mechanical apparatuses that are in contact with the machine vision system.

## Machine Vision System

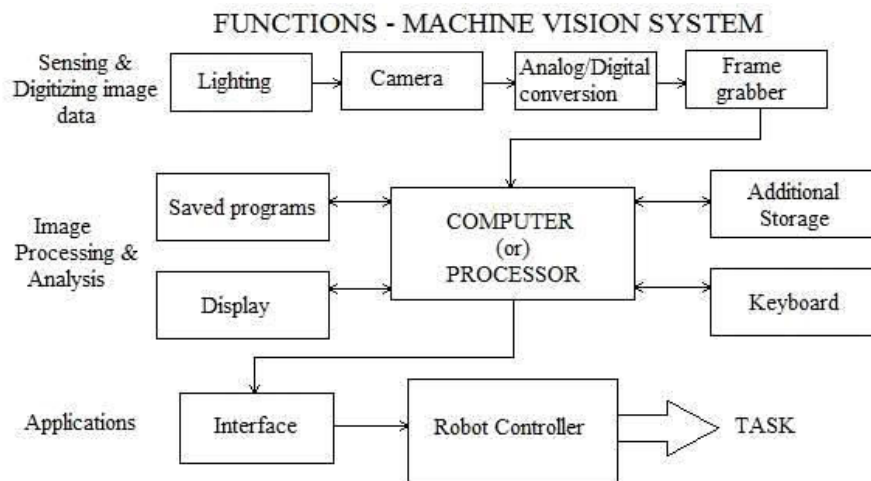


Fig 3.3 Block Diagram of Functions of Machine Vision System

Machine vision system is a *sensor* used in the robots for viewing and recognizing an object with the help of a computer. It is mostly used in the industrial robots for *inspection* purposes. This system is also known as *artificial vision or computer vision*. It has several components such as a camera, digital computer, digitizing hardware, and an interface hardware & software. The machine vision process includes three important tasks, namely:

- Sensing & Digitizing Image Data
- Image Processing & Analysis Applications

### Sensing & Digitizing Image Data:

A *camera* is used in the sensing and digitizing tasks for viewing the images. It will make use of special lighting methods for getting better picture contrast. These images are changed into the digital form, and it is known as *frame of the vision data*. A *frame grabber* is incorporated for taking digitized image continuously at 30 frames per second. Instead of scene projections, every frame is divided as a matrix. By performing sampling operation on the image, the number of pixels can be identified. The pixels are generally described by the elements of the matrix. A pixel is decreased to a value for measuring the intensity of light. As a result of this process, the intensity of every pixel is changed into the digital value and stored in the computer's memory.

## **Image Processing & Analysis:**

In this function, the *image interpretation* and *data reduction* processes are done. The threshold of an image frame is developed a binary image for reducing the data. The data reduction will help in converting the frame from raw image data to the feature value data. The feature value data can be calculated via computer programming. This is performed by *matching* the image descriptors like size and appearance with the previously stored data on the computer.

The image processing and analysis function will be made more effective by *training* the machine vision system regularly. There are several data collected in the training process like length of perimeter, outer & inner diameter, area, and so on. Here, the camera will be very helpful to identify the match between the computer models and new objects of feature value data.

### **Applications:**

Some of the important applications of the machine vision system in the robots are:

- Inspection
- Orientation
- Part Identification
- Location

## **Signal conversion**

Our interface modules are the links between the real physical process and the control system. Use the [EEx ia]-version of this function modules to assure a save data transmission from the potentially explosive area to the non-hazardous area and vice-versa. *Select the respective product properties below. The right-hand column adjusts the product list immediately and displays only products corresponding to your specifications.*

## **Image Processing**

Robotic vision continues to be treated including different methods for processing, analyzing, and understanding. All these methods produce information that is translated into decisions for robots. From start to capture images and to the final decision of the robot, a wide range of technologies and algorithms are used like a committee of filtering and decisions.

Another object with other colors accompanied by different sizes. A robotic vision system has to make the distinction between objects and in almost all cases has to track these objects. Applied in the real world for robotic applications, these machine vision systems are designed to duplicate the abilities of the human vision system using programming code and electronic parts. As human eyes can detect and track many objects in the same time, robotic vision systems seem to pass the difficulty in detecting and tracking many objects at the same time.

## **Machine Vision**

A robotic system finds its place in many fields from industry and robotic services. Even is used for identification or navigation, these systems are under continuing improvements with new features like 3D support, filtering, or detection of light intensity applied to an object.

Applications and benefits for robotic vision systems used in industry or for service robots:

- automating process;
- object detection;
- estimation by counting any type of moving;
- applications for security and surveillance;
- used in inspection to remove the parts with defects;
- defense applications;
- used by autonomous vehicle or mobile robots for navigation;
- for interaction in computer-human interaction;



## **Object tracking software**

A tracking system has a well-defined role and this is to observe the persons or objects when these are under moving. In addition, the tracking software is capable of predicting the direction of motion and recognizes the object or persons. OpenCV is the most popular and used machine vision library with open-source code and comprehensive documentation. Starting with image processing, 3D vision and tracking, fitting and many other features, the system include more than 2500 algorithms. The library interfaces have support for C++, C, Python and Java (in work), and also can run under Windows, Linux, Android or Mac operating systems.

## **SwisTrack**

Used for object tracking and recognition, SwisTrack is one of the most advanced tools used in machine vision applications. This tracking tool required only a video camera for tracking objects in a wide range of situations. Inside, SwisTrack is designed with a flexible architecture and uses OpenCV library. This flexibility opens the gates for implementing new components in order to meet the requirements of the user.

## **visual navigation**

Autonomous navigation is one of the most important characteristics for a mobile robot. Because of slipping and some incorrigible drift errors for sensors, it is difficult for a mobile robot to realize self-location after long distance navigation. In this paper, the perceptual landmarks were used to solve this problem, and the visual servoing control was adopted for the robot to realize self-location. At the same time, in order to detect and extract the artificial landmarks robustly under different illuminating conditions, the color model of the landmarks was built in the HSV color space. These functions were all tested in real time under experiment conditions.

## **Edge Detector**

Edge Detector Robot from IdeaFires is an innovative approach towards Robotics Learning. This is a simple autonomous Robot fitted with Controller and Sensor modules. The Edge Detector Robot senses the edges of table or any surface and turns the robot in such a way that it prevents it from falling.



**SATHYABAMA**

INSTITUTE OF SCIENCE AND TECHNOLOGY  
(DEEMED TO BE UNIVERSITY)

Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE

[www.sathyabama.ac.in](http://www.sathyabama.ac.in)

**SCHOOL OF MECHANICAL ENGINEERING**

**DEPARTMENT OF MECHANICAL ENGINEERING**

**UNIT – IV – ROBOT CELL DESIGN AND APPLICATIONS – SPR1606**

## Robot Work cell Layout

A robot cell is a complete system that includes the robot, controller, and other peripherals such as a part positioned and safety environment. Robot cells are sometimes referred to as work cells.

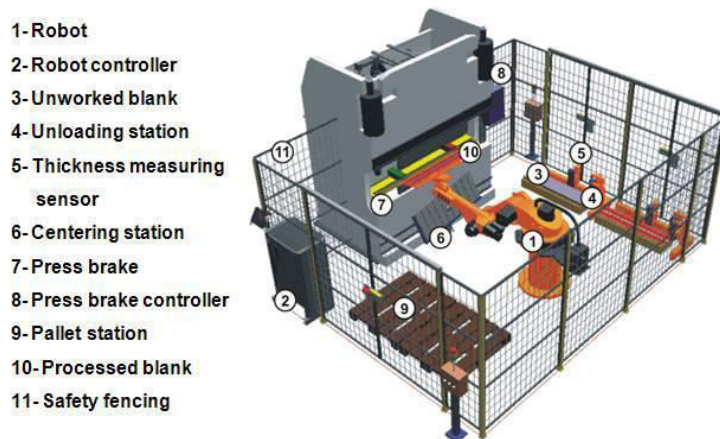


Fig 4.1 Robot Workcell

- Robot-centered work cell
- In-line robot work cell
- Mobile work cell

### Design of Robot-centered cell:

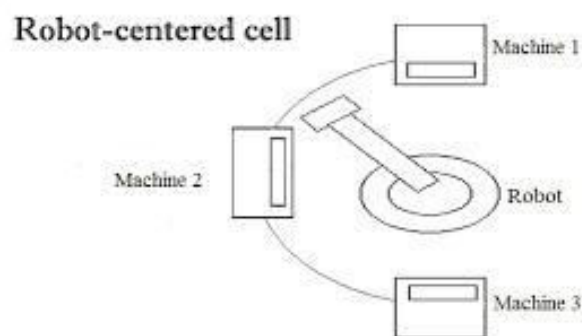


Fig 4.2 Robot Centred Cell

Robot-centered cell is one of the commonly used layouts in the industrial applications. In this arrangement, *a single robot* will be incorporated at the *center* of the work cell for performing operations on several machines that are set in a *semi-circle form*. This type of work

cell is shown in the below figure. This arrangement was organized because in 1960's *a single robot* was used to perform only *one function*, either carrying out a production process or examining a production machine. Those times, the robots were mostly employed in several operations like die casting, loading and unloading of parts, etc. where the *usage of robots* was *very low* when compared with the *machine usage times*. This *inequity in performance* was highly occurred in the *metal machining operations*. During this process, a robot remains *inactive* for a longer period. As a result, *Robot-centered cell* was introduced to maximize the usage of robots.

### **Robot-centered work cell**

- ☐ Center of work cell
- ☐ High utilization of robot
- ☐ Method of work part delivery (eg: conveyor, part-feeders, pallets)
- ☐ Install for single robot servicing 1@more production machines

### **Application of Robot-centered cell:**

Arc welding operation is an application of the Robot-centered cell. In this process, a robot is used to perform the *production process*, instead of examining the production machine.

### **In-line robot work cell**

An In-Line robot cell arrangement has a *movable conveyor* with the work parts. This conveyor travels near the *robots* for performing a function on the work parts. This type of arrangement is mostly used in the *automobile industries* for assembling the car bodies. To be more specific, the robots are placed beside the assembly line to carry out some operations like spot welding.

- ☐ 1 or more robots located along in-line conveyor
- ☐ Work is organized so each robot performs assembly operation on each part (eg: welding line)

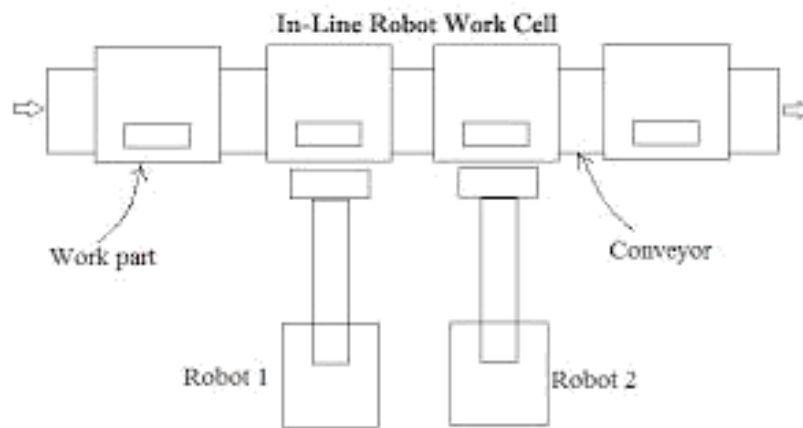


Fig 4.3 In-line Robot Cell

In the in-line robot cell design, anyone of the following transfer system can be used:

Synchronous or intermittent  
transfer system Continuous  
transfer system

Non synchronous transfer system

#### **Synchronous or Intermittent transfer system:**

In this transfer system, every work part is transported at the same time from one workplace and stored in another workstation. The robots are placed in a *fixed position* near the moving line in which the work parts travel. With the help of *start and stop motion*, the work parts are stopped next to a robot for performing the operations on it.

#### **Continuous transfer system:**

The *continuous transfer system* transfers the work parts *continuously* at a constant time along a conveyor. The continuous movement of work parts makes the robots tough to operate in a fixed position. As a result, there are two tracking system that helps to solve this problem such as:

*A moving baseline tracking system:* During an operation, this system allows the robot to move equivalent with the path of work parts. Hence, the traveling position of the robot and work parts will remain the same.

*A stationary baseline tracking system:* In this method, the robot is placed in a fixed position along a line. Instead of moving the robot, this system uses the manipulator for tracking the movable work parts. Here, tracking refers to the ability of a robot to sustain the locations of the programmed points relative to the work part even during the movement of the conveyor.

#### **Non synchronous transfer system:**

The *non-synchronous transfer system* allows the work parts to transfer *separately* along a conveyor. As like synchronous transfer, this system also moves with a start and stop motion. Sometimes, there is the possibility of *uneven arrival timings*. Therefore, *sensors* are used for specifying the starting time of the process to a robot. This transfer system is also called as *power – and – free system*.

#### **Mobile robot work cells**

*Mobile robot work cell* comes under the category of the *robot cell layouts*. Robot-centered work cell and In-line robot work cell are also one of the categories in this cell design. In the mobile robot work cells, the robots are arranged to move at different places in the cell. It is made possible by attaching the robot in a *movable bottom*, which is connected to the *rail system*. This system can be done by either any one of the following methods:

#### **Comparison of the rail systems:**

When comparing the floor track system and overhead rail systems, two things can be described:

The overhead rail system looks to be the best one in terms of floor space. The reason is that it requires *less floor space* than the floor track system.

The *cost* for constructing the floor track system is *very lower* than the overhead rail system.

#### **Design of mobile robot work cells:**

If a robot examines *several machines* for a *longer period*, then mobile robot work cell will be the perfect choice.

During this process, the robot will be mostly employed in the tasks than being in idle times.

## Mobile Robot Work Cell Overhead Rail System

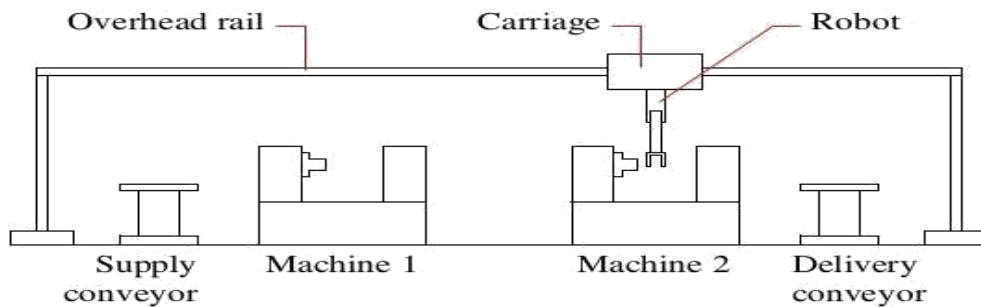


Fig 4.4 Mobile work cell layout

If this process is not applied, then *separate robot* will be required to perform this process, which causes *less usage of the robot*. It is said so because each robot should wait until the process gets finished in the production machine. A mobile robot cell has a problem in its design for identifying the *best possible machines* that a robot requires to examine. Moreover, this problem helps to *increase the machines* in the work cell without providing any idle times to the machines. Additionally, this operation requires *pallets, conveyors*, and other devices for providing the work parts to the cell. These devices will place the work parts in a *preferred area* for appropriate lifting. *Human workers* are also employed for performing the part loading and unloading tasks.

### Economic Analysis Of Robot

#### Background And Purpose:

Stroke is a leading cause of disability. Rehabilitation robotics have been developed to aid in recovery after a stroke. This study determined the additional cost of robot-assisted therapy and tested its cost-effectiveness.

#### Methods:

We estimated the intervention costs and tracked participants' healthcare costs. We collected quality of life using the Stroke Impact Scale and the Health Utilities Index. We analyzed the cost data at 36 weeks post randomization using multivariate regression models

controlling for site, presence of a prior stroke, and Veterans Affairs costs in the year before randomization.

### **Results:**

A total of 127 participants were randomized to usual care plus robot therapy (n=49), usual care plus intensive comparison therapy (n=50), or usual care alone (n=28). The average cost of delivering robot therapy and intensive comparison therapy was \$5152 and \$7382, respectively ( $P<0.001$ ), and both were significantly more expensive than usual care alone (no additional intervention costs). At 36 weeks postrandomization, the total costs were comparable for the 3 groups (\$17 831 for robot therapy, \$19 746 for intensive comparison therapy, and \$19 098 for usual care). Changes in quality of life were modest and not statistically different.

### **Conclusions:**

The added cost of delivering robot or intensive comparison therapy was recuperated by lower healthcare use costs compared with those in the usual care group. However, uncertainty remains about the cost-effectiveness of robotic-assisted rehabilitation pared with traditional rehabilitation.

Three methods to develop a robot with profit

Before starting the development of a robot, some of the data must be collected to carry out economic analysis effectively. They are:

- Type of robot to be installed.
- Cost to install a robot.
- Time taken to produce a robot.
- Savings and benefits in the development.

In an industry, the investment put on the development of a robot can be compared and analyzed by three common methods such as:

- Payback method
- EUAC (Equivalent Uniform Annual Cost)

ROI (Return on Investment Method)



**Payback method:**

The duration taken to equal the initial investment and net accumulated cash flow in the development of a robot is called payback period or payback method. If the net annual cash flows are identical to every year, then it can be stated by a formula given below.

Payback period = Investment Cost / Net Annual Cash Flow

**EUAC method:**

The EUAC is the short form of Equivalent Uniform Annual Cost method. It is used to alter the total cash flows and investments into the equivalent uniform costs over the expected time of developing a robot. It is done by employing different interest features that are connected with the calculations of engineering economy.

**ROI method:**

The Return on Investment is the expansion of ROI method. It is used to determine the return ratio of the current project, which is related to the anticipated expenditures and profits. If the rate of return is *low* to the expected cost of a company, then the investment made is *accepted*.



**SATHYABAMA**

INSTITUTE OF SCIENCE AND TECHNOLOGY  
(DEEMED TO BE UNIVERSITY)

Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE

[www.sathyabama.ac.in](http://www.sathyabama.ac.in)

**SCHOOL OF MECHANICAL ENGINEERING**

**DEPARTMENT OF MECHANICAL ENGINEERING**

**UNIT – V – ROBOT PROGRAMMING, ARTIFICIAL INTELLIGENCE,  
EXPERT SYSTEMS – SPR1606**

## Robot Programming

According to the *consistent* performance by the robots in industries, the robot programming can be divided in two common types such as:

- ☐ Leadthrough Programming Method
- ☐ Textual Robot Languages

### Leadthrough Programming Method:

During this programming method, the traveling of robots is based on the desired movements, and it is stored in the external controller memory. There are two modes of a control system in this method such as a *run mode* and *teach mode*. The program is taught in the teach mode, and it is executed in the run mode. The lead through programming method can be done by two methods namely:

- ☐ Powered Leadthrough Method
- ☐ Manual Leadthrough Method

#### a) Powered Leadthrough Method:

The powered leadthrough is the *common* programming method in the industries. A *teach pendant* is incorporated in this method for controlling the motors available in the joints. It is also used to operate the robot wrist and arm through a sequence of points. The playback of an operation is done by recording these points. The control of complex geometric moves is *difficult* to perform in the teach pendant. As a result, this method is good for *point to point* movements. Some of the key applications are spot welding, machine loading & unloading, and part transfer process.

#### b) Manual Leadthrough Method:

In this method, the robot's end effector is moved physically by the programmer at the desired movements. Sometimes, it may be difficult to move large robot arm manually. To get rid of it a teach button is implemented in the wrist for special programming. The manual leadthrough method is also known as Walk Through method. It is mainly used to perform continuous path movements. This method is best for spray painting and arc welding operations.

## **Textual Robot Languages:**

In 1973, WAVE language was developed, and it is the first textual robot language as well. It is used to interface the machine vision system with the robot. Then AL language was introduced in 1974 for controlling multiple robot arms during arm coordination. VAL was invented in 1979, and it is the common textual robot language. Later, this language was dated in 1984, and called as VAL II. The IBM Corporation has established their two own languages such as AML and AUTOPASS, which is used for the assembly operations.

Other important textual robot languages are Manufacturing Control Language (MCL), RAIL, and Automatic Programmed Tooling (APT) languages.

## **Robot Programming Methods**

There are three basic methods for programming industrial robots but currently over 90% are programmed using the each method.

### **Teach Method**

The logic for the program can be generated either using a menu based system or simply using a text editor but the main characteristic of this method is the means by which the robot is taught the positional data. A teach pendant with controls to drive the robot in a number of different co-ordinate systems is used to manually drive the robot to the desired locations.

These locations are then stored with names that can be used within the robot program. The co-ordinate systems available on a standard jointed arm robot are :-

### **Joint Co-ordinates**

The robot joints are driven independently in either direction.

### **Global Co-ordinates**

The tool centre point of the robot can be driven along the X, Y or Z axes of the robot's global axis system. Rotations of the tool around these axes can also be performed

### **Tool Co-ordinates**

Similar to the global co-ordinate system but the axes of this one are attached to the tool centre point of the robot and therefore move with it. This system is especially useful when the tool is near to the workpiece.

### ***Workpiece Co-ordinates***

With many robots it is possible to set up a co-ordinate system at any point within the working area. These can be especially useful where small adjustments to the program are required as it is easier to make them along a major axis of the co-ordinate system than along a general line. The effect of this is similar to moving the position and orientation of the global co-ordinate system.

This method of programming is very simple to use where simple movements are required. It does have the disadvantage that the robot can be out of production for a long time during reprogramming. While this is not a problem where robots do the same task for their entire life, this is becoming less common and some robotic welding systems are performing tasks only a few times before being reprogrammed.

### **Lead Through**

This system of programming was initially popular but has now almost disappeared. It is still however used by many paint spraying robots. The robot is programmed by being physically moved through the task by an operator. This is exceedingly difficult where large robots are being used and sometimes a smaller version of the robot is used for this purpose. Any hesitations or inaccuracies that are introduced into the program cannot be edited out easily without reprogramming the whole task. The robot controller simply records the joint positions at a fixed time interval and then plays this back.

### **Off-line Programming**

Similar to the way in which CAD systems are being used to generate NC programs for milling machines it is also possible to program robots from CAD data. The CAD models of the components are used along with models of the robots being used and the fixturing required. The program structure is built up in much the same

way as for teach programming but intelligent tools are available which allow the CAD data to be used to generate sequences of location and process information. At present there are only a few companies using this technology as it is still in its infancy but its use ncreas ng each year. The benefits of this form of programming are:-

- Reduced down time for programming.
- Programming tools make programming easier.
- Enables concurrent engineering and reduces product lead time.
- Assists cell design and allows process optimisation

### **Programming Languages for Robotics**

This article is all about giving an introduction about some of the programming languages which are used to design Robots.

There are many programming languages which we use while building Robots, we have a few programming languages which we always prefer to use in designing. Actually the programming languages which we use mainly depend on the hardware one is using in building robots. Some of them are- URBI, C and BASIC. URBI is an open source language. In this article we will try to know more about these languages. Let's start with URBI.

**URBI** : URBI stands for Universal Real-time Behavior Interface. It is a client/server based interpreted language in which Robot works as a client and controller as a server. It makes us to learn about the commands which we give to Robots and receive messages from them. The interpreter and wrapped server are called as "URBI Engine". The URBI Engine uses commands from Client and receives messages to it. This language allows user to work on basic Perception-action principle. The users just have to write some simple loops on the basis of this principle directly in URBI.

**PYTHON** : There is another language which is used in designing Robots. Python is an object-oriented language which is used to access and control Robots. Python is an interpreted language; this language has an application in working with mobile robots, particularly those manufactured by different companies. With python it is possible to use a single program for controlling many different robots. However Python is slower than C++ but it has so e good sides as well as it proved very easy to interact

with robots using this language, it is highly portable and can be run in windows and MAC OSX plus it can easily be extendable using C and C++ language. Python is a very reliable language for string manipulation and text processing.

**ROBOTC** : Other Languages which we use are C,C++ and C # etc. or their implementation, like ROBOTC, ROBOTC is an implementation of C language. If we are designing a simple

Robot, we do not need assembly code, but in complex designing we need well-defined codes. ROBOTC is another programming language which is C-based. It is actually a text based programming language. The commands which we want to give to our Robot, first written on the screen in the form of simple text, now as we know that Robot is a kind of machine and a machine only understands machine language. So these commands need to be converted in machine language so that robot can easily understand and do whatever it is instructed to do.

Although commands are given in text form (called as codes) but this language is very specific about the commands which is provided as instruction. If we do even a minor change in given text it will not accept it command. If the command which is provided to it is correct it colorizes that text, and we came to know that the given command in text form is correct (as we have shown in our example given below). Programming done in ROBOTC is very easy to do. Commands given are very straightforward. Like if we want our robot to switch on any hardware part, we just have to give code regarding to that action in text form. Suppose we want robot to turn motor of port, we just have to give command in this way:

Although program above is not exactly shown in the way in which it should be written, this is just to provide you a visualization of what we have told you. This is not written in an appropriate manner. ROBOTC provide advantage of speed, a Robot programmed in ROBOTC programming supports 45 times more speed than provided by other programming based on C plus it has a very powerful debugging feature.

## **ROBOTICS.NXT :**

ROBOTICS.NXT has a support for a simple message-based control. It direct commands, nxt-upload is one of its programs which is used to upload any file. It works on Linux. After getting introduction on programming languages, it becomes necessary to know something about MRDS as well, MRDS is an environment which is designed especially for controlling robots.

## **Microsoft Robotics Developer Studio**

Microsoft Robotics Developer Studio is an environment given for simulation purpose of Robots. It is based on a .net library concurrent implementation. This environment has support so that we can add other services as well. It has features which not only include creating and debugging Robot Applications but also it becomes easy to interact with sensors directly. C# programming language is used as a primary language in it. It has 4 main components:

- Visual Programming Language (VPL)
- Visual simulation environment (VSE)

Concurrency and coordination Runtime is a synchronous programming library based on .net framework. Although it is a component of MRDS but it can be used with any application. DSS is also a .net runtime environment, In DSS services are exposed as resources which one can access through programs. DSS uses DSSP (Decentralizes software services protocol) and HTTP.

If we want to graphics and visual effects in our programming, we use VPL. Visual Programming language is a programming language which allows us to create programs by doing manipulations in programming languages graphically. We use boxes and arrows in this kind of programming while we want to show dataflow kind of things.

Visual programming language has huge application in animations. The last component which we are going to describe is Visual Simulation Environment. VSE provides simulated physical objects. Visual Simulation environment is an integrated environment for picture-based, object oriented and component based applications of simulation.



Programming in robotics is a very vast topic that we can't cover in a single article. This is just an introduction for those who want to get an idea about using languages in building of robots

## **Motion Commands and the Control of Effectors**

Real-time systems are slaves to the clock. They achieve the illusion of smooth behavior by rapidly updating set of control signals many times per second. For example, to smoothly turn a robot's head to the right, the head must accelerate, travel at constant velocity for a while, and then decelerate. This is accomplished by making many small adjustments to the motor torques. Another example: to get the robot's LEDs to blink repeatedly, they must be turned on for a certain period of time, then turned off for another length of time, and so forth. To get them to glow steadily at medium intensity, they must be turned on and off very rapidly.

The robot's operating system updates the states of all the effectors (servos, motors, LEDs, etc.) every few milliseconds. Each update is called a "frame", and can accommodate simultaneous changes to any number of effectors. On the AIBO, updates occur every 8 milliseconds and frames are buffered four at a time, so the application must have a new buffer available every 32 milliseconds; other robots may use different update intervals. In Tekkotsu these buffers of frames are produced by the MotionManager, whose job is to execute a collection of simultaneously active MotionCommands (MCs) of various types every few milliseconds. The results of these MotionCommands are assembled into a buffer that is passed to the operating system (Aperios for the AIBO, or Linux for other robots). Suppose we want the robot to blink its LEDs on and off at a rate of once per second. What we need is a MotionCommand that will calculate new states for the LEDs each time the MotionManager asks for an update. LedMC, a subclass of both Motion Command and LedEngine, performs this service. If we create an instance of LedMC, tell it the frequency at which to blink the LEDs, and add it to the MotionManager's list of active MCs, then it will do all the work for us. There's just one catch: our application is running in the Main process, while the MotionManager runs in a separate Motion process. This is necessary to assure that potentially lengthy computations taking place in Main don't prevent Motion from running every few milliseconds. So how

can we communicate with our MotionCommand while at the same time making it available to the Motion Manager.

## **Applications of Industrial Robots**

### **Machine Loading**

Machine loading The first application of industrial robots was in unloading die-casting machines. In die casting the two halves of a mould or die are held together in a press while molten metal, typically zinc or aluminium, is injected under pressure. The die is cooled by water; when the metal has solidified the press opens and a robot extracts the casting and dips it in a quench tank to cool it further. The robot then places the casting in a trim press where the unwanted parts are cut off. The robot often grips the casting by the sprue. (The sprue is the part of the casting which has solidified in the channels through which molten metal is pumped to the casting proper. Several castings may be made at once; in this case they are connected to the sprue by runners. When the sprue and runners are cut off by the trim press, the press must automatically eject the casting(s) onto a conveyor.

### **Spray Painting**



Fig 5.1 Spray Painting

Spray painting the major application of industrial robot particularly in Automobile manufacturing industries. This technology is used to perform spray painting of automobile spare parts and all the automobile components.

## Artificial intelligence

Artificial intelligence (AI) is the intelligence exhibited by machines or software. It is also the name of the academic field of study which studies how to create computers and computer software that are capable of intelligent behavior.

- ☐ The intelligent control robot is capable of performing some of the functions and tasks carried out by human beings.
- ☐ It can detect changes in the work environment by means of sensory perception.
- ☐ Intelligent robot is equipped with a variety of sensors and sensor apparatus providing visual (computer vision) and tactile (touching) capabilities to respond instantly to variable situations.
- ☐ Much like humans, the robot observes and evaluates the immediate environment by perception and pattern recognition.
- ☐ Because its operation is so complex, powerful computers are required to control its movements and more- sophisticated sensing devices to respond to its actions.
- ☐ Extensive research has been and still concerned with how to equip robots with seeing “eyes” and tactile “fingers”.
- ☐ Artificial intelligence (AI) that will enable the robots to respond, adapt, reason, and make decisions to

react to change is also an inherent capability of the intelligent robot.

### Goals of AI

To create expert systems – the systems which exhibit intelligent behavior, learn, demonstrate, explain, and advice its users.

To implement human intelligence in machines – crea ng systems that understand, think, learn, and behave like humans.

The goals of artificial intelligence are:

Intelligent behavior by simulated persons or other actors in a computer game.

Mobile robots in hostile or dangerous environments that behave intelligently and act according to the situation and save the hostages with less blood loss.

Intelligent responses to user requests in information search systems such as Google or Wikipedia.

Process control systems that supervise complex processes and that are able to respond intelligently to a broad variety of malfunctioning and other unusual situations

Intelligent execution of user requests in the operating system and the basic services (document handling, electronic mail) of standard computers.

What is AI technique?

In the real world, the knowledge has some unwelcomed properties: its volume is huge, next to unimaginable.

It is not well-organized or well-formatted. It keeps changing constantly.

AI technique is a manner to organize and use the knowledge efficiently in such a way that: it should be perceivable by the people who provide it.

It should be easily modifiable to correct errors.

It should be useful in many situations though it is incomplete or inaccurate.

AI techniques elevate the speed of execution of the complex program it is equipped with.

What contributes to AI?

Artificial intelligence is a science and technology based on disciplines such as computer science, biology, psychology, linguistics, mathematics, and engineering. A major thrust of AI is in the development of computer functions associated with human intelligence, such as reasoning, learning, and problem solving. Out of the following areas, one or multiple areas can contribute to build an intelligent system.

## **Applications of AI**

### Game playing

You can buy machines that can play master level chess for a few hundred dollars. There is some ai in them, but they play well against people mainly through brute force computation--looking at hundreds of thousands of positions. To beat a world champion by brute force and known reliable heuristics requires being able to look at 200 million positions per second.

### Speech recognition

In the 1990s, computer speech recognition reached a practical level for limited purposes. Thus united airlines has replaced its keyboard tree for flight information by a system using speech recognition of flight numbers and city names. It is quite convenient. On the the other hand, while it is possible to instruct some computers using speech, most users have gone back to the keyboard and the mouse as still more convenient.

### Understanding natural language

Just getting a sequence of words into a computer is not enough. Parsing sentences is not enough either. The computer has to be provided with an understanding of the domain the text is about, and this is presently possible only for very limited domains.

## Computer vision

The world is composed of three-dimensional objects, but the inputs to the human eye and computers' tv cameras are two dimensional. Some useful programs can work solely in two dimensions, but full computer vision requires partial three-dimensional information that is not just a set of two-dimensional views. At present there are only limited ways of representing three-dimensional information directly, and they are not as good as what humans evidently use.

## Expert systems

A "knowledge engineer" interviews experts in a certain domain and tries to embody their knowledge in a computer program for carrying out some task. How well this works depends on whether the intellectual mechanisms required for the task are within the present state of ai. When this turned out not to be so, there were many disappointing results. One of the first expert systems was mycin in 1974, which diagnosed bacterial infections of the blood and suggested treatments. It did better than medical students or practicing doctors, provided its limitations were observed. Namely, its ontology included bacteria, symptoms, and treatments and did not include patients, doctors, hospitals, death, recovery, and events occurring in time. Its interactions depended on a single patient being considered. Since the experts consulted by the knowledge engineers knew about patients, doctors, death, recovery, etc., it is clear that the knowledge engineers forced what the experts told them into a predetermined framework. In the present state of ai, this has to be true. The usefulness of current expert systems depends on their users having common sense. Heuristic classification

One of the most feasible kinds of expert system given the present knowledge of ai is to put some information in one of a fixed set of categories using several sources of information. An example is advising whether to accept a proposed credit card purchase. Information is available about the owner of the credit card, his record of payment and also about the item he is buying and about the establishment from which he is buying it (e.g., about whether there have been previous credit card frauds at this establishment).

## **knowledge-based expert system (KBES)**

The knowledge-based expert system (kbes) is the first realisation of research in the field of artificial intelligence (ai), in the form of a software technology. For developers of application software, particularly in medical and engineering disciplines, it was a boon, as it addressed the decision-making process with the use of symbols rather than numbers. Tasks belonging to the classification and diagnosis category were the first to benefit from the emergence of kbes technology. Though ai researchers were carrying out symbolic processing much earlier, the results of such research could be taken from lab to field only when kbes was introduced as a software tool for addressing a class of problems that required simulation of the knowledge-based decision-making process.

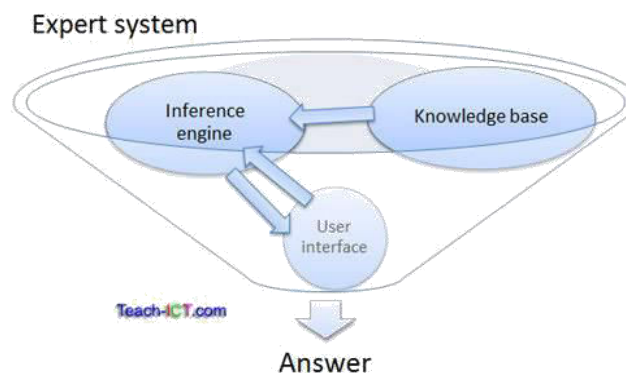
A knowledge-based system (kbs) is a computer program that reasons and uses a knowledge base to solve complex problems. The term is broad and is used to refer to many different kinds of systems. The one common theme that unites all knowledge based systems is an attempt to represent knowledge explicitly via tools such as ontologies and rules rather than implicitly via code the way a conventional computer program does. A knowledge based system has two types of sub-systems: a knowledge base and an inference engine. The knowledge base represents facts about the world, often in some form of subsumption ontology. The inference engine represents logical assertions and conditions about the world, usually represented via if-then rules.[1]

Knowledge-based systems were first developed by artificial intelligence researchers. These early knowledge-based systems were primarily expert systems. In fact the term is often used synonymously with expert systems. The difference is in the view taken to describe the system. Expert system refers to the type of task the system is trying to solve, to replace or aid a human expert in a complex task. Knowledge-based system refers to the architecture of the system, that it represents knowledge explicitly rather than as procedural code. While the earliest knowledge-based systems were almost all expert systems, the same tools and architectures can and have since been used for a whole host of other types of systems. I.e., virtually all expert systems are knowledge-based systems but many knowledge-based systems are not expert systems.

The first knowledge-based systems were rule based expert systems. One of the most famous was mycin a program for medical diagnosis. These early expert systems represented facts about the

world as simple assertions in a flat database and used rules to reason about and as a result add to these assertions. Representing knowledge explicitly via rules had several advantages:

1. Acquisition & maintenance. Using rules meant that domain experts could often define and maintain the rules themselves rather than via a programmer.
2. Explanation. Representing knowledge explicitly allowed systems to reason about how they came to a conclusion and use this information to explain results to users. For example, to follow the chain of inferences that led to a diagnosis and uses these facts to explain the diagnosis.
3. Reasoning. Decoupling the knowledge from the processing of that knowledge enabled general purpose inference engines to be developed. These systems could develop conclusions that followed from a data set that the initial developers may not have even been aware of.



As knowledge-based systems became more complex the techniques used to represent the knowledge base became more sophisticated. Rather than representing facts as assertions about data, the knowledge-base became more structured, representing information using similar techniques to object-oriented programming such as hierarchies of classes and subclasses, relations between classes, and behavior of objects. As the knowledge base became more structured reasoning could occur both by independent rules and by interactions within the knowledge base itself. For example, procedures stored as demons on objects could fire and could replicate the chaining behavior of rules.

Advancement was the development of special purpose automated reasoning systems called classifiers. Rather than statically declare the subsumption relations in a knowledge-base a



classifier allows the developer to simply declare facts about the world and let the classifier deduce the relations. In this way a classifier also can play the role of an inference engine.

The most recent advancement of knowledge-based systems has been to adopt the technologies for the development of systems that use the internet. The internet often has to deal with complex, unstructured data that can't be relied on to fit a specific data model. The technology of knowledge-based systems and especially the ability to classify objects on demand is ideal for such systems.

### **Applications of expert systems KBES**

Applications tend to cluster into seven major classes.

Diagnosis and troubleshooting of devices and systems of all kinds

This class comprises systems that deduce faults and suggest corrective actions for a malfunctioning device or process. Medical diagnosis was one of the first knowledge areas to which es technology was applied (for example, see shortliffe 1976), but diagnosis of engineered systems quickly surpassed medical diagnosis. There are probably more diagnostic applications of es than any other type. The diagnostic problem can be stated in the abstract as: given the evidence presenting itself, what is the underlying problem/reason/cause? Planning and scheduling

Systems that fall into this class analyze a set of one or more potentially complex and interacting goals in order to determine a set of actions to achieve those goals, and/or provide a detailed temporal ordering of those actions, taking into account personnel, materiel, and other constraints. This class has great commercial potential, which has been recognized. Examples involve airline scheduling of flights, personnel, and gates; manufacturing job-shop scheduling; and manufacturing process planning.

Configuration of manufactured objects from subassemblies

Configuration, whereby a solution to a problem is synthesized from a given set of elements related by a set of constraints, is historically one of the most important of expert system applications. Configuration applications were pioneered by computer companies as a means of facilitating the manufacture of semi-custom minicomputers (mcdermott 1981). The technique has found its way into use in many different industries, for example, modular home building, manufacturing, and other problems involving complex engineering design and manufacturing.

#### Financial decision making

The financial services industry has been a vigorous user of expert system techniques. Advisory programs have been created to assist bankers in determining whether to make loans to businesses and individuals. Insurance companies have used expert systems to assess the risk presented by the customer and to determine a price for the insurance. A typical application in the financial markets is in foreign exchange trading. Knowledge publishing

This is a relatively new, but also potentially explosive area. The primary function of the expert system is to deliver knowledge that is relevant to the user's problem, in the context of the user's problem. The two most widely distributed expert systems in the world are in this category. The first is an advisor which counsels a user on appropriate grammatical usage in a text. The second is a tax advisor that accompanies a tax preparation program and advises the user on tax strategy, tactics, and individual tax policy.

Systems falling in this class analyze real-time data from physical devices with the goal of noticing anomalies, predicting trends, and controlling for both optimality and failure correction. Examples of real-time systems that actively monitor processes can be found in the steel making and oil refining industries.

#### Design and manufacturing

These systems assist in the design of physical devices and processes, ranging from high-level conceptual design of abstract entities all the way to factory floor configuration of manufacturing processes.