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

SCHOOL OF MECHANICAL ENGINEERING

DEPARTMENT OF MECHANICAL ENGINEERING

		L	T	P	Credits	Total Marks
SMEA3012	**INDUSTRIAL ROBOTICS AND EXPERT SYSTEMS**	**3**	**0**	**0**	**3**	**100**

COURSE OBJECTIVES

- Understand the basics of robotics and Automation Systems.
- Learn the robot cell design, Robot Configuration and robot programming.
- Understand the application of artificial intelligence and expert systems in robotics.

UNIT 1 INTRODUCTION AND ROBOTIC KINEMATICS 9 Hrs.

Definition need and scope of industrial robots- Coordinate Systems Classification of Robot- Robot anatomy - work volume - Precision movement – End effectors - sensors. Robot kinematics – Basics about plane rotation – rotation matrix - Direct and inverse kinematics - Robot trajectories-Control of robot manipulators - Robot dynamics - Methods for orientation and location of objects. Pitch, Yaw, Roll, Joint Notations, Speed of Motion, Pay Load.

UNIT 2 ROBOT DRIVES AND CONTROL 9 Hrs.

Controlling the robot motion - Position and velocity sensing devices - Design of drive systems - Hydraulic and Pneumatic drives - D.C. Servo Motors, Stepper Motors, A.C. Servo Motors Linear and rotary actuators and control valves –Electro hydraulic servo valves, electric drives - Motors –Selection of Drives- designing of end effectors - Vacuum, magnetic and air operated grippers.

UNIT 3 ROBOT PATH PLANNING AND IMAGE PROCESSING 9 Hrs.

Introduction-Path planning overview- Road map path planning- Cell decomposition path planning-Potential field path planning-Obstacle avoidance- Robotic vision system - Image Gripping - Image processing and analysis - Image segmentation – Pattern recognition - Training of vision system.

UNIT 4 ROBOT CELL DESIGN AND FIELD ROBOTS 9 Hrs.

Robot work cell design and control - Safety in Robotics - Robot cell layouts - Multiple robots and machine interference - Robot cycle time analysis - Ariel robots- Collision avoidance-Robots for agriculture, mining, exploration, underwater, civilian and military applications, nuclear applications, Space applications.

UNIT 5 ROBOT PROGRAMMING, ARTIFICIAL INTELLIGENCE AND EXPERT SYSTEMS 9 Hrs.

Methods of robot programming - characteristics of task level languages lead through programming methods- Motion interpolation. Artificial intelligence - Basics - Goals of artificial Intelligence - AI techniques – problems representation in AI- Problem reduction and solution techniques - Application of AI. Elements of Knowledge Representation -Logic, Production Systems, Semantic Networks, Expert Systems Knowledge Building Environment Systems (KBES)-Humanoids.

Max. 45 Hrs.

COURSE OUTCOMES

On completion of the course, student will be able to

CO1 - Recall fundamental Concepts of Robots and Kinematics.
CO2 - Implement Drives concepts in End effectors.
CO3 - Analyze and design Path planning and Image processing.
CO4 - Design robot work cell for automation industries.
CO5 - Apply the knowledge about robot programming methods.
CO6 - Apply the concepts of AI and expert systems.

TEXT / REFERENCE BOOKS

1. K.S.Fu, R.CGonzalez and C.S.G. Lee, Robotics control, Sensing, Vision and intelligence", McGraw

Hill,1994.
2. Kozyrey, Yu, "Industrial Robotics", MIR Publishers Moscow, 1998.
3. Richard.D.,Klafter, Thomas.A, Chmielewski, Machine Negin "Robotics Engineering-An Integrated Approach", Prentice Hall of India, 1984.
4. Deb, S.R. "Robotics Technology and Flexible Automation", Tata McGraw Hill, 1994.
5. Mikell,P.Groover, Mitchell Weis, Roger N.Nagel, Nicholas Odrey "Industrial Robotics Technology, Programming and Applications", McGraw Hill, Int., 1986.
6. Timothy Jordonidesetal,"Expert Systems and Robotics", Springer-Verlag, New York, May 1991.

UNIT – I – INTRODUCTION AND ROBOTIC KINEMATICS– SMEA 3012

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

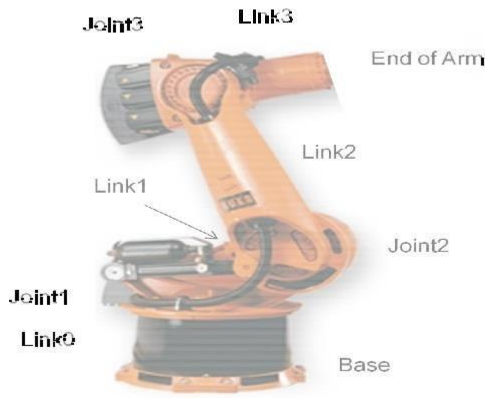


Fig 1.1 Robot Anatomy

Main Components of Industrial Robots

Arm or Manipulator End effectors

Drive Mechanism Controller

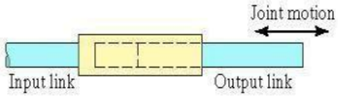
Custom features: e.g. sensors and transducers Manipulator consists of joints and links

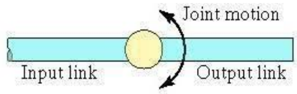
- ✓ Joints provide relative motion
- ✓ Links are rigid members between joints
- ✓ Various joint types: linear and rotary
- ✓ Each joint provides a “degree-of-freedom”
- ✓ Body-and-arm – for positioning of objects in the robot's work volume
- ✓ Wrist assembly – for orientation of objects

Translational motion

Linear joint (type L)

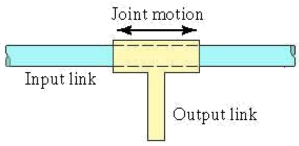
Rotational joint (type R)

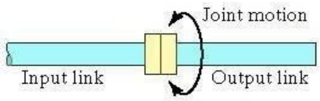




Orthogonal joint (type O)

Twisting joint (type T)





Revolving joint (type V)

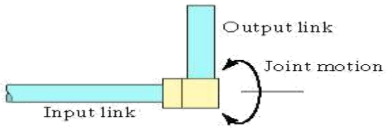


Fig 1.2 Robot Joints

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:

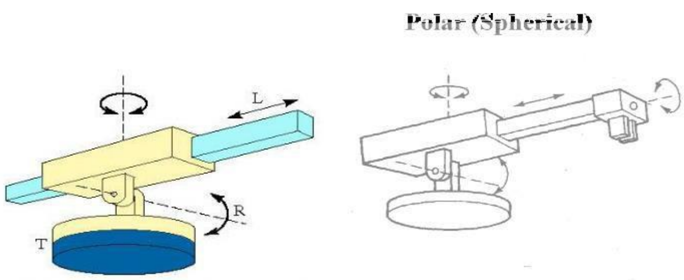


Fig 1.3 Polar Configurations

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:

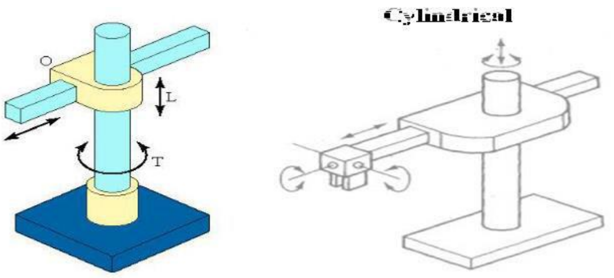


Fig 1.4 Cylindrical Configurations

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:

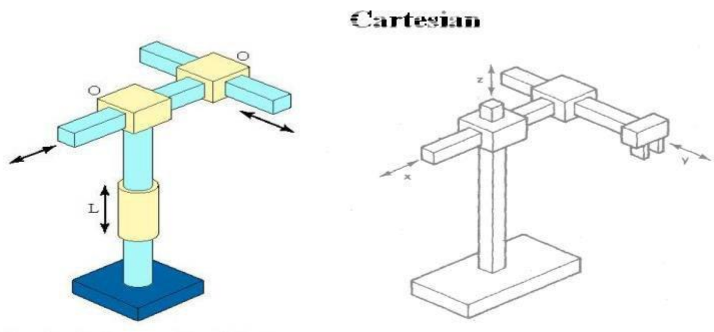


Fig 1.5 Cartesian Configurations

Jointed-Arm Robot

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

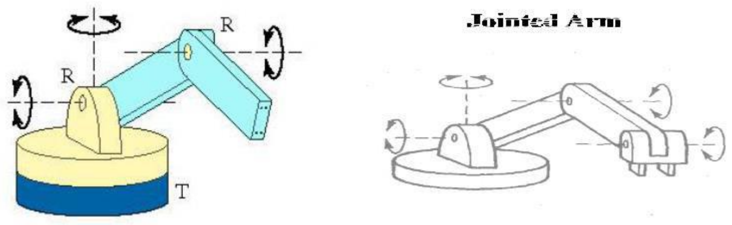


Fig 1.6 Jointed Arm Configurations

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

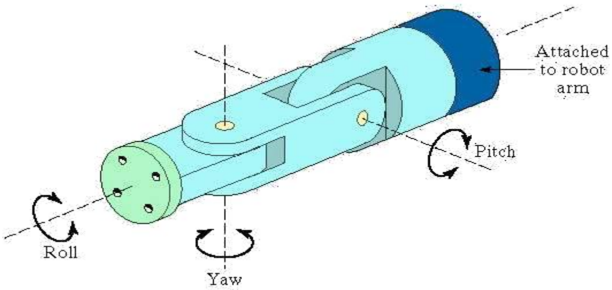


Fig1. 7 Gripper Configurations

- 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.

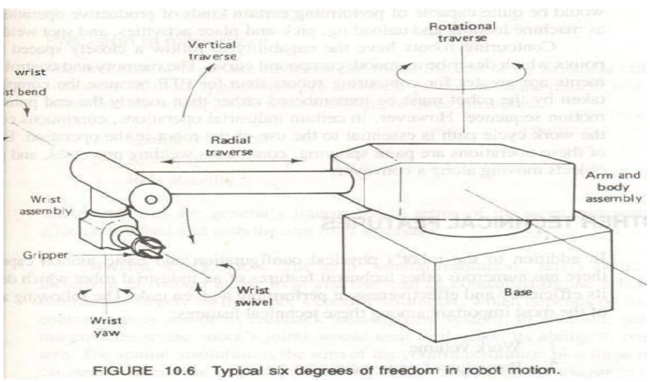


Fig 1.8 Six Degrees of Freedom Robot Manipulator

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

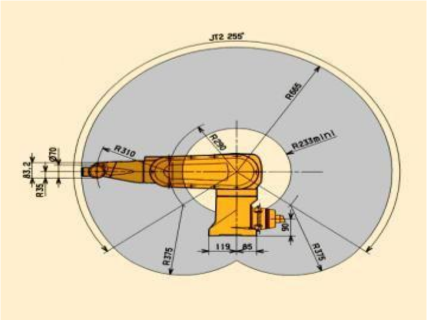


Fig 1.9 Robot Work volume

- 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

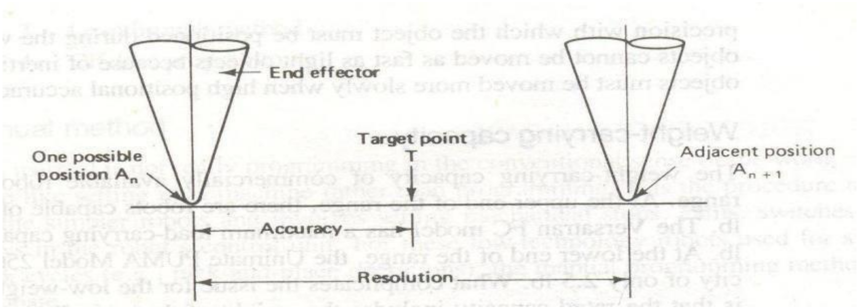


Fig 1.10 Spatial Resolutions

Smallest increment of motion

Depends on the control system and feedback

Control resolution = **range/ Accuracy** Control increment

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

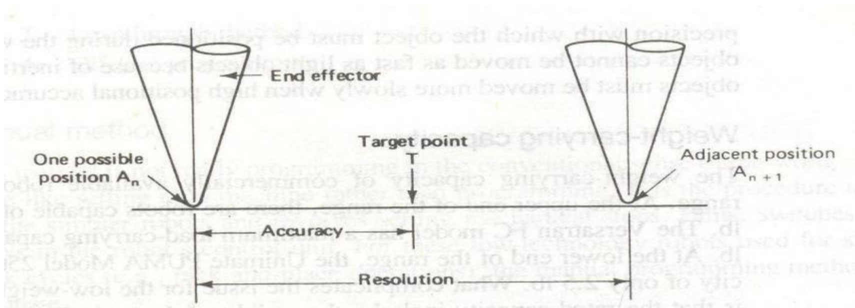


Fig 1.11 Accuracy

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

Speeds 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

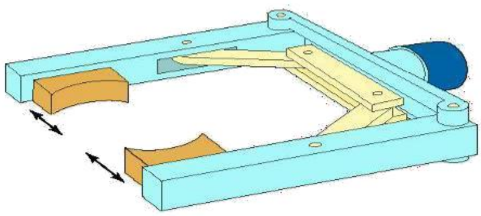


Fig 1.12 Robot Mechanical Gripper

- Dual gripper's
- 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

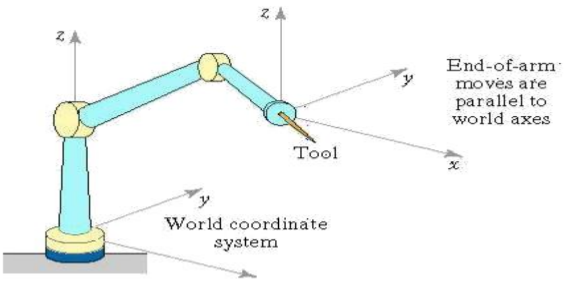


Fig 1.13 Worlds Coordinate System

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)

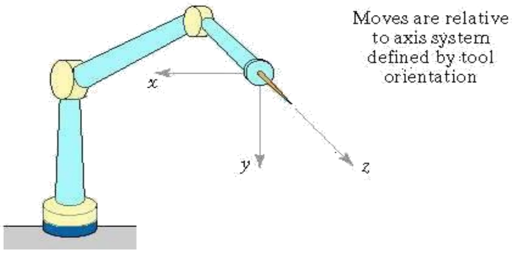


Fig 1.14 Tools Coordinate System

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: ~q → T (~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 → ~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

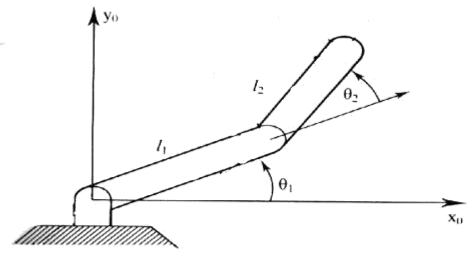


Fig 1.15 Forward Kinematics

The Situation:

You have a robotic arm that starts out aligned with the x_o-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^0Y^0 frame.

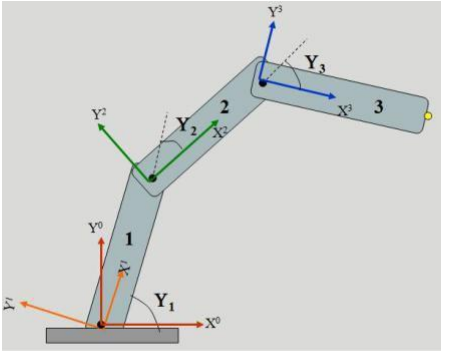


Fig 1.16 Kinematic Manipulator

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

i.e. Rotating by U_1 will put you in the X^1Y^1 frame. Translate in the along the X^1 axis by l_1. Rotating by U_2 will put you in the X^2Y^2 frame. and so on until you are in the X^3Y^3 frame.

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

(l_1, 0). Multiplying H by that position vector will give you the coordinates of the yellow point relative the the X^0Y^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_4Y_4 frame to the X^0Y^0 frame.

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

$$\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = H \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

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

SCHOOL OF MECHANICAL ENGINEERING

DEPARTMENT OF MECHANICAL ENGINEERING

UNIT – II – ROBOT DRIVES AND CONTROLS – SMEA 3012

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

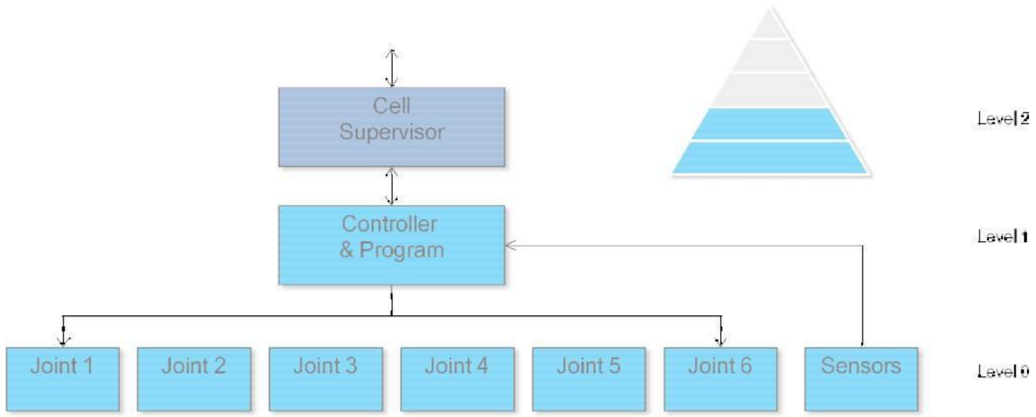


Fig 2.1 Robot Control system

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

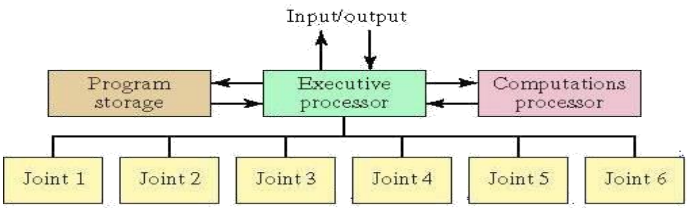


Fig 2.2 Robot Control System

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 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
3. Tactile - touch sensors and force sensors
4. Proximity - when an object is close to the sensor
5. Optical -Machine vision
6. 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
- **Pneumatic** Drive system
  - 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, se 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.

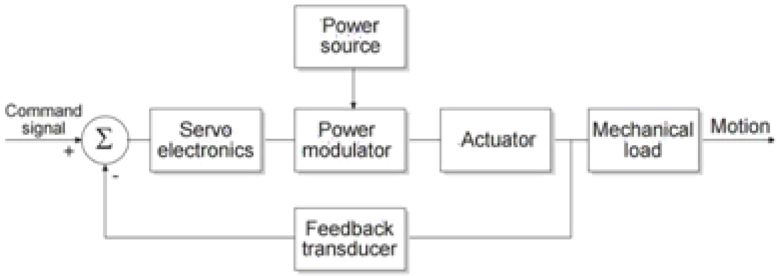


Fig 2.3 Basic Servo Mechanics

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)
6) Pneumatic fingers
7) Mandrel grippers
8) Pin grippers
9) Special Purpose Grippers (Hooking devices, custom positioners or tools)
    - Welding (MIG /TIG, Plasma Arc, Laser, Spot)
    - Pressure Sprayers (painting, water jet cutting, cleaning)
    - Hot Cutting type (laser, plasma, de-flashers-hot knife)
    - Buffing/Grinding/De-burring type
    - Drilling/Milling type
    - 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 suctions 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.

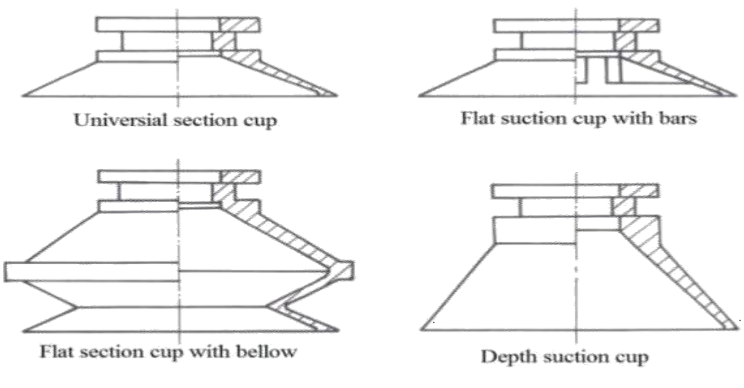


Figure 3 Different types of suction cups, picture taken from [5] page 204

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. Items with rough surfaces (surface roughness ≤ 5 μm 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.

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,

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.

SCHOOL OF MECHANICAL ENGINEERING

DEPARTMENT OF MECHANICAL ENGINEERING

UNIT – III – ROBOT PATH PLANNING AND IMAGE PROCESSING – SMEA 3012

UNIT 3
ROBOT PATH PLANNING AND IMAGE PROCESSING

Path Planning

As the term itself suggests, path planning is a computational problem that lets an autonomous mobile robot find the optimal path between two points. In other words, this problem deals with finding the shortest path between point A and point B such that there is no intersection with any configuration space obstacle**.**

The probabilistic roadmap planner is a motion planning algorithm in robotics, which solves the problem of determining a path between a starting configuration of the robot and a goal configuration while avoiding collisions. The basic idea behind PRM is to take random samples from the configuration space of the robot, testing them for whether they are in the free space, and use a local planner to attempt to connect these configurations to other nearby configurations. The starting and goal configurations are added in, and a graph search algorithm is applied to the resulting graph to determine a path between the starting and goal configurations.

At this point, the reader should know about the following important terms in robotics:

1. Configuration Space
2. Configuration Space Obstacle

Configuration Space

The set of all configurations or positions that a robot can attain is known as its configuration space. We also refer to it as c-space. We use Cartesian coordinates to define a robot's configuration.

Configuration Space Obstacles

It is the region in which either the robot collides with the physical obstacles or some specified links of the robot to collide with each other.

Moving back to the main topic, Probabilistic Roadmap planning is used to determine the shortest (and/or optimal) path between two specified points, let's refer to these points as nodes now.

A probabilistic roadmap (PRM) is a network graph of possible paths in a given map based on free and occupied spaces.

Let’s look at the steps involved in forming such a network graph

1. A random node is generated in the configuration space.

2. The system checks whether this node lies in free space or not i.e, whether this configuration intersects with an obstacle or not.

3. If the node is in free space, it is added to the graph.

4. This newly generated node is then connected to the closest nodes through a straight line.

5. The system then checks if the connection between two nodes lies in free space or not.

6. If it lies in free space, the connection is added to the graph. Let’s refer to theses connection as edges now.

The process of generating random nodes and then making edges is repeated n times.

Let’s say the system generates a random node x. Now, a function is called to check whether the x lies in free space. Let’s call this function collision check.

If collision check returns TRUE, the system will call another function to find the m (m can be any number) nearest points to this new node x. Here, this function will be called distance. It will return the coordinates of m nearest nodes

Then another function, check edge, is called to check if it is possible to construct an edge between the nodes that the distance function returns.

A visual representation of two consecutive iterations of this algorithm is given below (The blue shapes are obstacles)

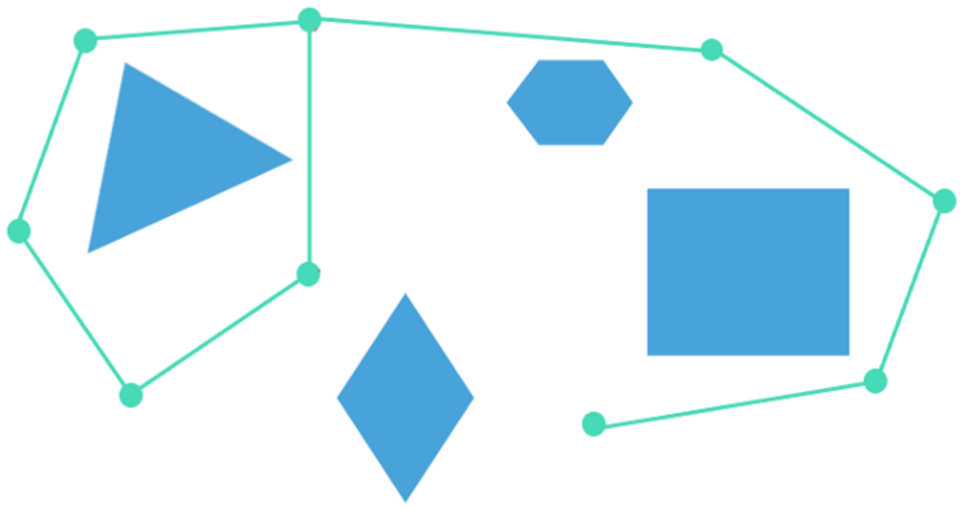


Fig 3.1 (k-1)th iteration

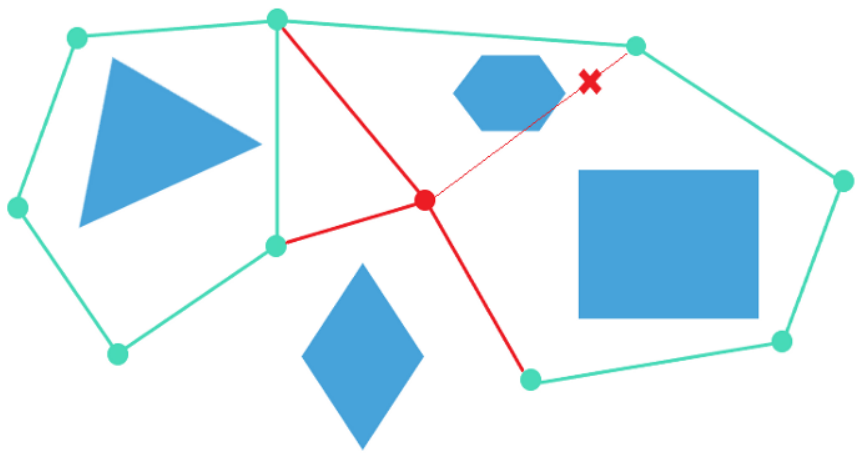


Fig 3.2 kth iteration

Now that we have a graph of nodes and edges, we can augment the **start** and **end** points in it to arrive at a solution to our path planning problem.

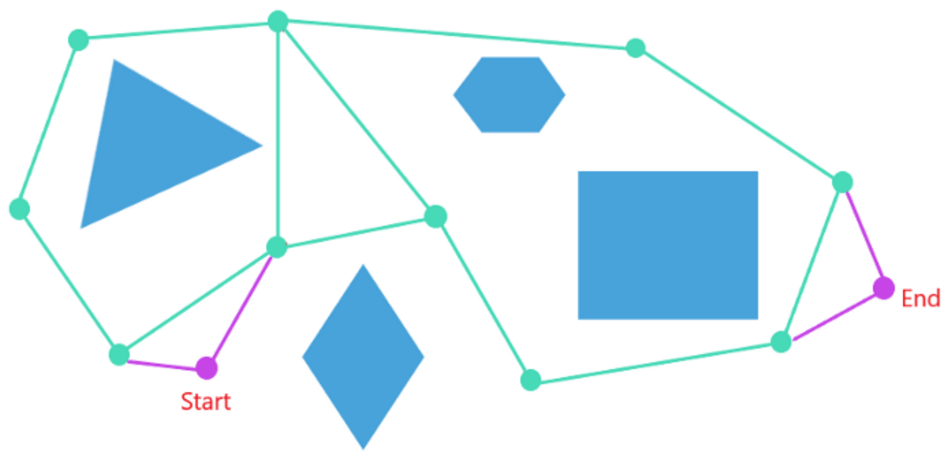


Fig 3.3 Final decision

The use of unmanned air vehicle or autonomous robot in place of human beings to carry out dangerous missions in adverse environments has been gradually increased since last decades. Path planning is one of the vital aspects in developing an autonomous vehicle that should traverse the shortest distance from a starting point to a target point while in a given mission for saving its resources and minimizing the potential risks. Therefore, it is crucial for a path planning algorithm to produce an optimal path. The path planning algorithm should also hold the completeness criterion which means that a path can be found if that exists. Moreover, the robot's safety, memory usages for computing and the real-time algorithms are also significant. Fig. illustrates the classification of path planning approaches.

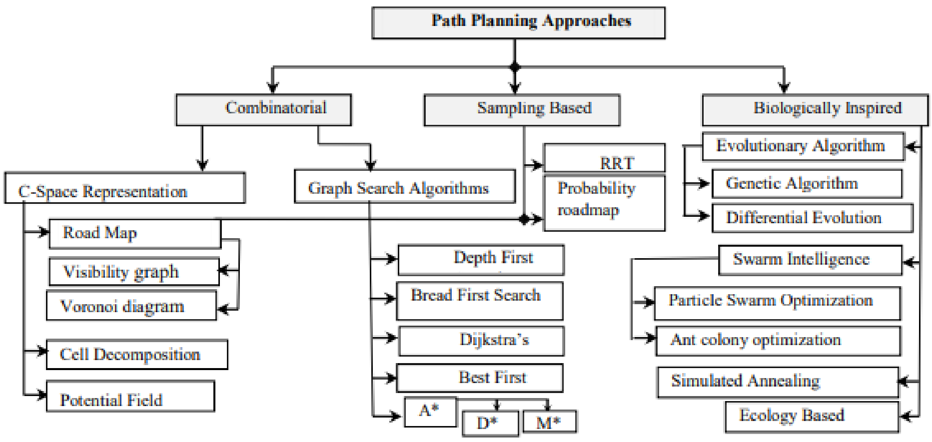


Fig 3.4 Classification of process planning

The bio-inspired methods are the nature-motivated/biologically inspired algorithms. A number of instances of bio-inspired approaches are the Genetic algorithm (GA), Simulated annealing (SA), Particle Swarm Optimization (PSO) plus Ant Colony Optimization (ACO).

GA uses the natural selection course of biological evolution that continuously fluctuate a populace of distinct results. Nonetheless, it cannot assure any optimal path. Local minima may occur in narrow environments and thus, it offers a lesser amount of safety and constricted corridor difficulty.

GA is computationally costly and ultimately it is not complete. SA algorithm is developed based on warming and cooling process of metals to regulate the internal configuration of its properties. Separate from very sluggish and very high cost functions, SA is not able to accomplish the optimal path.

PSO is a meta-heuristic population based approach and it has real-time outcome, but it tumbles into local optima easily in many optimization complications. Additionally, there is no general convergence concept appropriate for PSO in practice and its convergence period is mostly vague for multidimensional problems.

On the other hand, ACO emulates an ant to mark a path while the food source is confirmed. The ant separates its direction towards the food source with pheromones for tracing purpose. In ACO, the path in between the initial point and target point is arbitrarily produced. ACO does a blind exploration and therefore, it is not proper for efficient path planning due to the lack of optimal result.

In sampling based path planning, a method Rapidly Exploring Random Tree (RRT) does not require the establishment of the design space. In RRT, the first step is to define the starting and the target points. Then, it considers the starting point as the base for the tree, based on which different new branches are grown-up till it reaches Combinatorial Visibility graph Voronoi diagram C-Space Representation Cell Decomposition Road Map Potential Field Particle Swarm Optimization Evolutionary Algorithm Ant colony optimization Simulated Annealing Biologically Inspired Swarm Intelligence Genetic Algorithm Differential Evolution Ecology Based Graph Search Algorithms Path Planning Approaches D M Probability roadmap RRT Sampling Based Best First Depth First Search Dijkstra's A Bread First Search 3 the target point .

RRT is simple and easy way to handle problems with obstacles and different constraints for autonomous/unmanned robotic motion planning. Depending on the size of the engendered tree, the computation time is also escalated. The resulting path commencing by RRT is not optimal all the time. Nonetheless, it remains pretty easy to find a path for a vehicle with dynamic and physical constrictions and it also creates least number of edges.

Probabilistic roadmap (PRM) method is a path-planning algorithm that takes random samples from the configuration space by examining the accessible free space and dodging the crashes to find a way. A local planner is used to join these configurations with close-by configurations. PRM is costly without any possibilities to acquire the path. Combinatorial path planning consists of mainly two methods, i.e. C-space representation technique and graph search algorithm. In this case, the first step is to create the configuration space of the environment. Then, a graph search algorithm, for example Dijkstra's and A-star, is applied to search a path.

Depth-first search (DFS) is good to pick up a path among many possibilities without caring about the exact one. It may be less appropriate when there is only one solution. DFS is good because a solution can be found without computing all nodes. Breadth-first search that is suitable for limited available solutions uses a comparatively small number of steps. Its exceptional property finds the shortest path from the source node up to the node that it visits first time when all the graph's edges are either un-weighted or having similar weight. Breadth-first search is complete if one exists. Breadth-first search is good because it does not get trapped in dead ends and this algorithm does not assure to discover the shortest path because it bypasses some branches in the search tree.
It is a greedy search which is not complete and optimal. Dijkstra's algorithm is systematic search algorithm and gives shortest path between two nodes. In optimal cases, where there is no prior knowledge of the graph, it cannot estimate the distance between each node and the target. Usually, a large area is covered in the graph by Dijkstra's due to its edge selections with minimum cost at every step and thus, it is significant for the situation having multiple target nodes without any prior knowledge of the closest one.

A* is not very optimal because it needs to be executed a number of times for each target node to get them all. A* expands on a node only if it seems promising. It only aims to reach the target from the current node at the earliest and does not attempt to reach any other node. A* is complete because it always finds a path if one exists. By modifying the used heuristics and node's evaluation tactics of A*, other path-finding algorithm can be developed.

Configuration space gives complete information about the location of all points in the coordination and it is the space for all configurations such as real free space area for the motion of autonomous vehicle and guarantees that the vehicle must not crash with obstacles.

An illustration of a C-space for a circular vehicle. It assumes the robot as a point and adds the area of the obstacles so that the planning can be complete in a more capable way. C-space is obtained by adding the vehicle radius while sliding it along the edge of the obstacles and the border of the search space

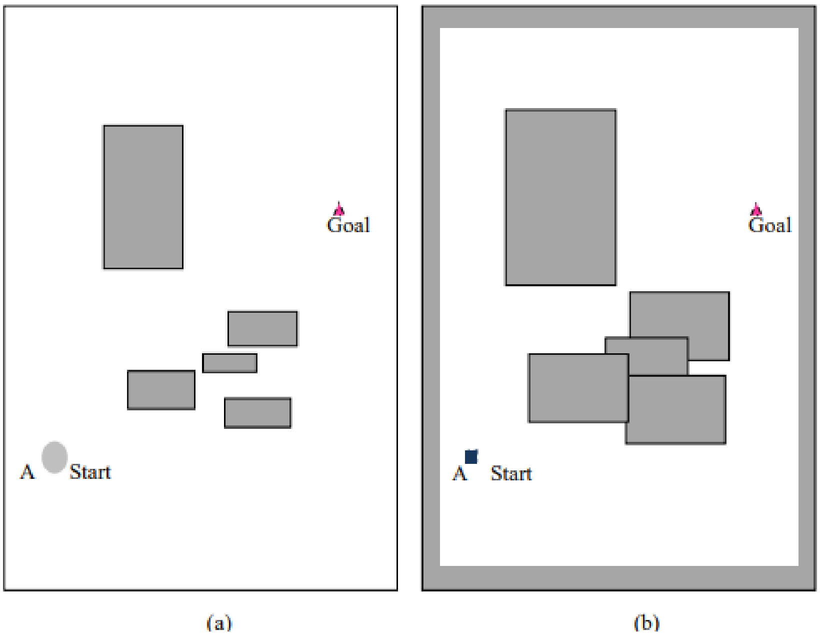


Fig 3.5 A scenario represented in (a) original form (b) configuration space. Note that the darker rectangles in (a) are those with actual dimensions while in (b) are those enlarged according to the size of robot A. The white areas represent free space.

Cell decomposition

Problem Formulation

In the basic problem, it is assumed that the robot is the only moving object in the workspace and the dynamic properties of the robot are ignored, thus avoiding temporal issues. The motions are resticted to non-contact motions, so that the issue related to the mechanical interaction between two physical objects in contact can be ignored. These assumptions essential transform the "physical" motion planning problem into a purely geometrical path planning problem. The geometrical issues are simplified even further by assuming that the robot ia single rigid object. The motions of this object are only constrained by the obstacles. The underlying idea of the configuration space is to represent the robot as a point in an appropriate space (the robot's configuration space) and to map the obstacles in this space. This mapping transforms the problem of planning the motion of a dimensioned object into the problem of planning the motion of a point.

Problem Solution

There exist a large number of methods for solving the basic motion planning problem. Not all of them solve the problem in its full generality. Despite many external differences, the methods are based on few different general approaches which are roadmap, cell decomposition and potential field. In this paper we study more deeply two of them, explained in the following subsections.

Cell decomposition

Cell decomposition can be used in path planning in the following way: The free space of the polygonal two-dimensional configuration space is determined.

The free space is partitioned into a collection of cells.

A connectivity graph is constructed by connecting the cells that share a common boundary (a hole in the bounding polygon corresponds to a cycle in the connectivity graph).

In the on-line phase: A sequence of cells, a channel, which the robot must traverse in order to go from the initial position to the goal position, is obtained from the connectivity graph. A free path is constructed from the channel.

If the robot is not a point and can turn in any direction then computing the free space is a major part of the calculation. Most methods assume a point-sized robot or a convex polygonal robot with fixed orientation and increase the thickness of the wall by the width of the robot.

The resulting free space is taken as an input. The triangular robot has a fixed orientation and a reference point p. Given the workspace defined by the interior of the bold polygon, the free space of the robot, with respect to the point p, is the white area with the bold line. However, it now becomes more difficult to describe the actual environment around a cell and most robots do not have a fixed orientation.

Potential field

Potential field method treats the robot represented as a point in configuration space as a particle under the influence of an artificial potential field U whose local variations are expected to reflect the "structure" of the free space.

The potential fields can be pictured in a mind either a charged particle navigating through a magnetic field or a marble rolling down a hill. The basic idea is that behavior exhibited by the particle/marble will depend on the combination of the shape of the field/hill. Unlike fields/hills where the topology is externally specified by environmental conditions, the topology of the potential fields that a robot experiences are determined by the designer. More specifically, the designer creates multiple behaviors, each assigned a particular task or function, represents each of these behaviors as a potential field, and combines all of the behaviors to produce the robot's motion by combining the potential fields.

The potential function is typically defined over free space as the sum of an attractive potential pulling the robot toward the goal configuration and a repulsive potential pushing the robot away from the obstacles. At each iteration an artificial force introduced by the potential function at the current configuration is regarded as the most promising direction of motion, and path generation proceeds along this direction by some increment.

The potential is calculated as the sum of two or more elementary potential functions: The attractive potential field can be simply defined as a parabolic well: where e is a positive scaling factor and denotes the Euclidean distance. The function is positive or null, and attains its minimum at q_{goal} , where

U_{att} is singular.

The main idea underlying the definition of the repulsive potential is to create a potential barrier around the C−obstacle area that cannot be traversed by the robots configuration. In addition, it is usually desirable that the repulsive potential does not affect the motion of the robot when it is sufficiently far away from C−obstacle.

$$U_{rep}(q) = \begin{cases} \frac{1}{2}\eta\left(\frac{1}{\rho}(q) - \frac{1}{\rho_0}\right) & \text{if} \quad \rho(q) \leq \rho_0 \\ 0 & \text{if} \quad \rho(q) > \rho_0 \end{cases}$$

where is a positive scaling factor, 0 denotes the distance from to the C−obstacle region, and is a positive constant called the distance of influence of the C−obstacle.

Exact cell decomposition algorithm

In this section, the exact cell decomposition method which was used for the software implementation of the algorithm is introduced. Polygonal configuration space method is the simple case of exact cell decomposition described in, where 2C and the C-obstacle region C (the union of the C-obstacles) forms a polygonal region in C. For simplifying the presentation, we assume that the robot's free space C=C/Cbeta is bounded. Figure 1 depicts such a configuration space

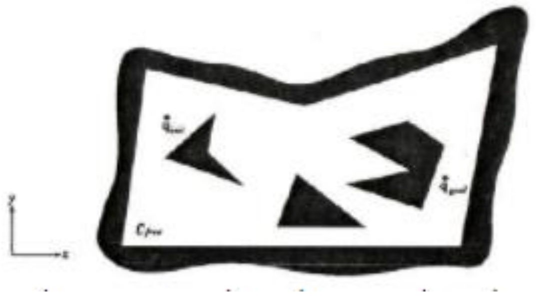


Fig 3.6 Two dimensional configuration spaces

The decomposition of C_{free} and the associated connectivity graph is defined as follows: A convex polygonal decomposition K of C_{free} is a finite collection of convex polygons, called cells, such that the interiors of any two cells do not intersect and the union of all the cells is equal to C_{free}. Two cells and in K are adjacent if only if is a line segment of non-zero length. The connectivity graph G associated with a convex polygonal decomposition K of C_{free} is the non-directed graph specified as follows: o G's nodes are the cells in K . Two nodes in G are connected by a link if and only if the corresponding cells are adjacent. Consider an initial configuration q_{init} and a goal configuration q goal in C_{free} .

Machine vision system

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 (engineering) it 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

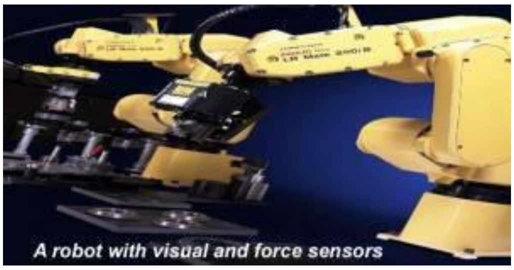


Fig 3.7 Industrial Robot with Sensor

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.

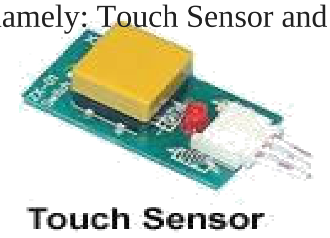


Fig 3.8 Touch Sensor and Force Sensor

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

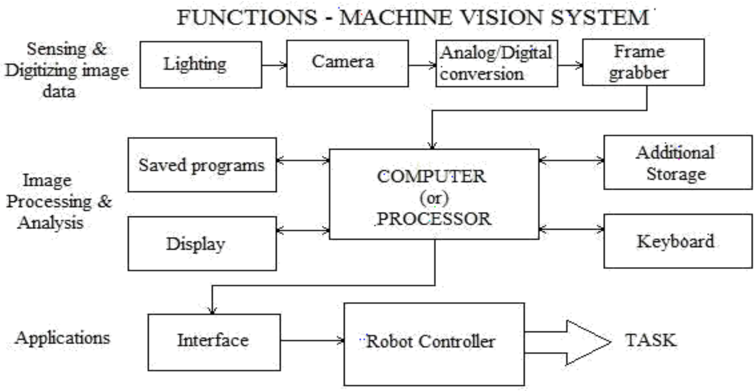


Fig 3.9 Block Diagram of Functions of Machine Vision System

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 gaining better picture contrast. These images are changed into the digital form, and it is known as the 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 matrix. By performing 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 on, 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 infor ation 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 tracking 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.

Swiss Track

Used for object tracking and recognition, Swiss Track 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, Swiss Track 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 serving 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 Idea Fires 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.

SCHOOL OF MECHANICAL ENGINEERING

DEPARTMENT OF MECHANICAL ENGINEERING

UNIT – IV – ROBOT CELL DESIGN AND FIELD ROBOTS – SMEA 3012

UNIT IV
ROBOT CELL DESIGN AND FIELD ROBOTS

Work cell control:

Industrial robots usually work with other things: processing equipment, work parts, conveyors, tools and perhaps human operators. A means must be provided for coordinating all of the activities which are going on within the robot workstations. Some of the activities occur sequentially, while others take place simultaneously to make certain that the various activities are coordinated and occur in the proper sequence, a device called the work cell controller is used. The work cell controller usually resides within the robots and has overall responsibility for regulating the activities of the work cell components.

Functions of work cell controller

1. Controlling the sequence of activities in the work cycles
2. Controlling simultaneous activities
3. Making decisions to proceed based on incoming signals
4. Making logical decisions
5. Performing computations
6. Dealing with exceptional events
7. Performing irregular cycles, such as periodically changing tool

Interlocks an interlock is the feature of work cell control which prevents the work cycle sequence from continuing until a certain conditions or set of conditions has been satisfied. In a robotic work cell, there are two types: outgoing and incoming.

The outer going interlock is a signal sent from the workstation controller to some external machine or device that will cause it to operate or not to operate for example this would be used to prevent a machine from initiating its process until it was commanded to process by the work cell controller, an incoming interlock is a single from some external machine or device to the work controller which determines whether or not the programmed work cycle sequence will proceed. For example, this would be used to prevent the work cycle program from continuing until the machine signaled that it had completed its processing of the work piece. The use of interlocks provides an important benefit in the control of the work cycle because it prevents actions from happening when they should not, and it causes actions occur when they should.

Interlocks are needed to help coordinate the activities of the various independent components in the work cell and to help avert damage of one component by another. In the planning of interlocks in the robotic work cell, the application engineer must consider both the normal sequences of the activities that will occur during the work cycle, and the potential malfunction that might occur. Then these normal activities are linked together by means of limit switches, pressure switches, photo electric devices, and other system components. Malfunction that can be anticipated are prevented by means of similar devices.

Robot work cell layout

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

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

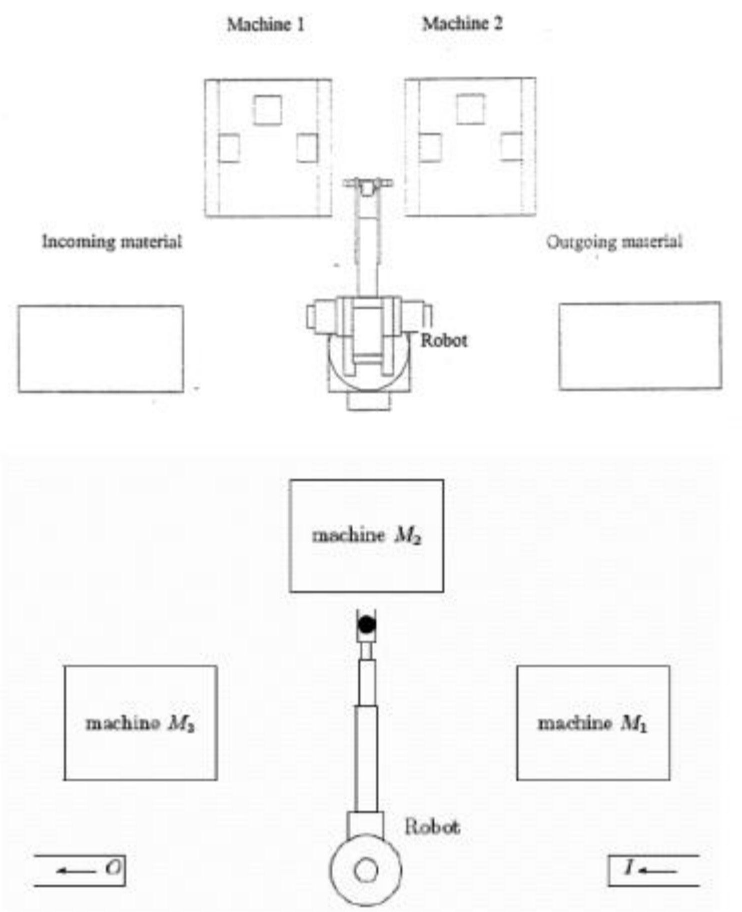


Fig 4.1 Robot centered work cell

Work cell Description

In-line robot work cell

1 @ more robots located along in-line conveyor

Work is organized so each robot performs assembly operation on each part (eg: welding line)

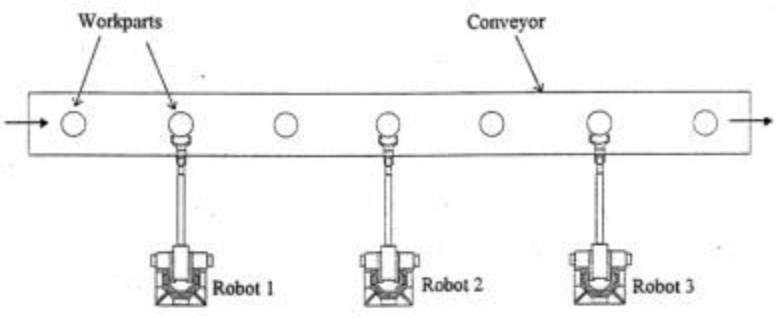


Fig 4.2 In-line robot work cell

There are 3 types of work part transport system used in in-line robot work cell.

1. Intermittent Transfer
2. Continuous Transfer
3. Non-Synchronous Transfer

Intermittent Transfer

The parts are moved in a start-and-stop motion from one station to another along the line. It is also called synchronous transfer since all parts are moved simultaneously to the next stop. The advantage of this system is that the parts are registered in a fixed location and orientation with respect to the robot during robot's work cycle.

Continuous Transfer

Work parts are moved continuously along the line at constant speed. The robot(s) has to perform the tasks as the parts are moving along. The position and orientation of the parts with respect to any fixed location along the line are continuously changing. These results in a "tracking" problem, that is, the robot must maintain the relative position and orientation of its tool with respect to the work part. This tracking problem can be solved. the moving baseline tracking system by moving the robot parallel to the conveyor at the same speed or by the stationary baseline tracking system i.e. by computing and adjusting the robot tool to maintain the position and orientation with respect to the moving part.

Non-synchronous Transfer System

This is a power and free system". Each work part moves independently of other parts. in a stop-and-go manner. When a work station has finished working on a work part, that part then proceeds to the next work station. Hence, some parts are being processed on the line at the same time that others are being transported or located between stations. Here. the timing varies according to the cycle time requirements of each station.

The design and operation of this type of transfer system is more complicated than the other two because each part must be provided with its own independently operated moving cart. However, the problem of designing and controlling the robot system used in the power-and free method is less complicated than for the continuous transfer method. Nonsynchronous Transfer System For the irregular timing of arrivals, sensors must be provided to indicate to the robot when to begin its work cycle. The more complex problem of part registration with respect to the robot that must be solved in the continuously moving conveyor systems are not encountered on either the intermittent transfer or the non-synchronous transfer.

Mobile work cell

In this arrangement, the robot is provided with a means of transport, such as a mobile base, within the work cell to perform various tasks at different locations. The transport mechanism can be floored mounted tracks or overhead railing system that allows the robot to be moved along linear paths.

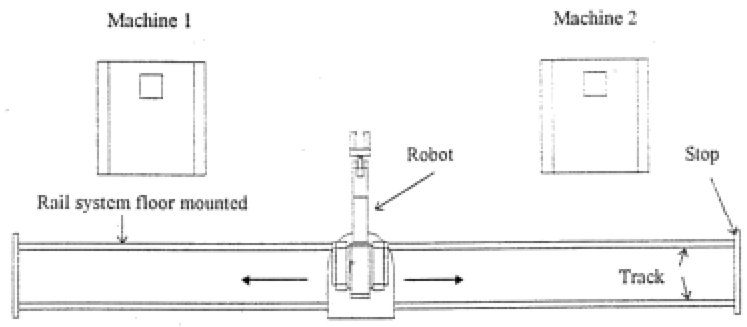


Fig 4.3 Mobile work cell

Mobile robot work cells are suitable for installations where the 1 robot must service more than one station (production machine) that has long processing cycles, and the stations cannot be arranged around the robot in a robot-centred cell arrangement. One such reason could be due to the stations being geographically separated by distances greater than the robot's reach. The type of layout allows for time-sharing tasks that will lower the robot idle time. One of the problems in designing this work cell is to find the optimum number of stations or machines for the robot to service.

Multiple robots and machine interference

Physical Interference of Robots

Here the work volumes of the robots in the cell are overlapping, posing dangers of collision. Collisions can be prevented by separating the robots so that their work volumes are not overlapping. – However, there are cases where the robots work on the same component piece or where the robots in turns, work on the component. Here the programmed work cycles of the robots must be coordinated so that they will not be near enough to risk a collision.

Machine Interference

This occurs when two or more machines are being serviced by one robot. The machine cycles are timed in such a way that idle time is experienced by one or more machines, while one machine is being serviced by the robot. Machine interference can be measured as the total idle time of all the machines in the cell as compared to the operator (or robot) cycle time. The measure is commonly expressed as a percent.

Calculation of Machine Interference

A three-machine cell in which a robot is used to load and unload the machines. Each of the three machines are identical with identical cycles times of 50 s. This cycle time is divided between run time (30 s) and service time (load/unload) by the robot (20 s). The organization of the cycle time is shown in the robot and machine process chart given below.

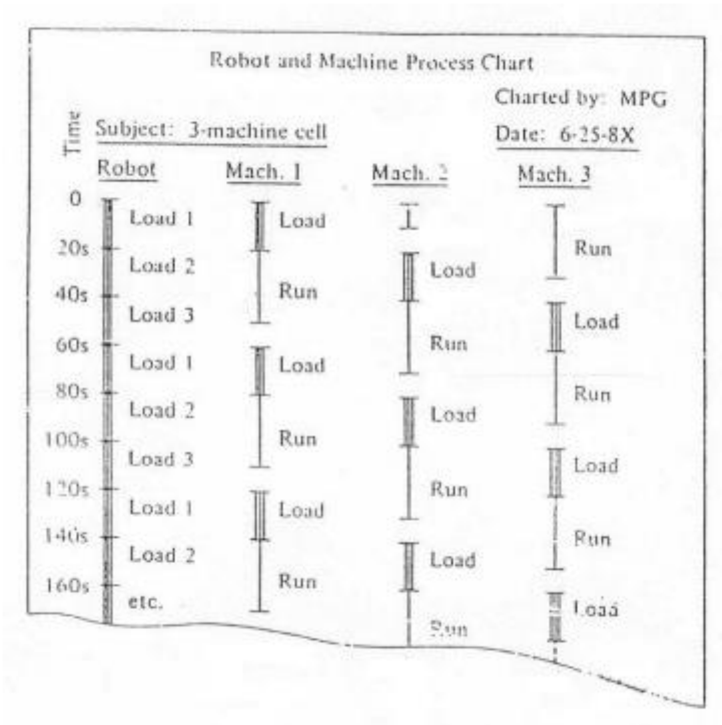


Fig 4.4 Process chart

It can be seen that each machine has idle time during its cycle of 10 s while the robot is fully occupied throughout its work cycle.

Total idle time of all three machines is 3 x 10 = 30 s The cycle time of the robot is 3 x 20 = 60 s Therefore, machine interference is 30 s / 60 s = 50%

In the example above, when the robot cycle time is greater than the machine cycle time, there will be resulting machine interference. – If the machine cycle time is greater than the robot cycle time, there will no machine interference, but the robot will be idle for part of the cycle. – In cases where the service and run times of the machines are different, the above relationships become complicated by the problem of determining the best sequence of servicing times for the machines into the robot cycle time.

Robot cycle time analysis

The cycle time analysis is important since it is related to production rate and economic viability of robot installation. An analysis method known as the RTM (Robot Time and Motion) has been developed at Purdue University. It is used for estimating the time needed to perform a work cycle before setting up a work cell and programming the robot. This would enable alternative robot tasks be evaluated. By comparing the performance for a given work cycle, it can also be used as a guide in the selection of the best robot for a particular application.

The robot work cycle elements can be broadly categorized into: Motion elements. These are manipulator movements, performed with or without load. Sensing elements. These are sensory activities performed by robots equipped with sensing capabilities such as vision sensing, force sensing and position sensing. End effector elements. These elements relate to the action of the gripper or tool attached to the robot wrist. Delay elements. These are delay times resulting from waiting and processing conditions in the work cycle.

There are four possible approaches that can be used to determine the element times and analyse a robot cycle with RTM: First approach: involves tables of elements, in which time values are determined for different elements listed in Table Second approach: Develop regression equations for the more complicated elements whose values are functionally related to several factors. Third approach: known as "Motion Control", and can be applied to those in the Motion elements. Motion control concerns with kinematics and dynamics analysis of movement. It determines the element time values by considering distances moves and velocities involved. It also considers the acceleration and deceleration at the beginning and end of motion. Fourth approach: known as "path geometry". It requires the specification of the motion path to be followed by the robot, as well as the robot joint and arm velocities. It turns out that most robot motions involve simultaneous actuation of several joints, but one of the joints usually predominates because its relative move is the largest.

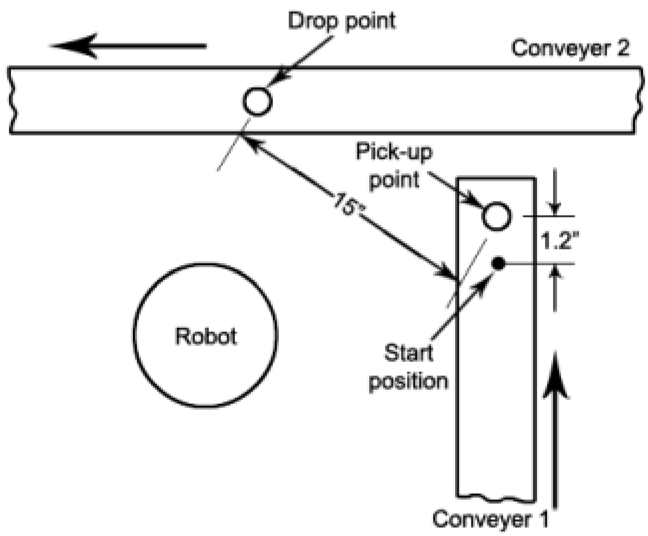


Fig 4.5 Work cell layout for cycle time analysis

A sketch of the workstation showing distances that the robot must move from one position to the next is shown in Fig 4.5 The conveyor delivers one part every 15 s, so that the work cycle is limited by the conveyor feed rate. The RTM analysis in this problem would be useful for determining whether the time for the robot motion cycle is compatible.

This example will illustrate the use of the RTM method. The work cycle consists of a simple task in which the robot must move parts weighing 3 lb from one conveyor to another conveyor. The sequence of the work cycle proceeds as follows:

Robot picks up part from first conveyor which has delivered the part to a known pickup position. 2. Robot transfers part to second conveyor and releases part. 3. Robot moves back to ready position at first conveyor.

Ariel robots

Using the definition, an aerial robot is a system capable of sustained flight. with no direct human control and able to perform. An unmanned aerial vehicle (UAV) is defined as a "powered, aerial vehicle that does not carry a human operator, uses aerodynamic forces to provide vehicle lift, can fly autonomously or be piloted remotely, can be expendable or recoverable, and can carry a lethal or nonlethal payload" UAV is a term that is commonly applied to military use cases. In addition to the software, autonomous drones also employ a host of advanced technologies that allow them to carry out their missions without human intervention, such as cloud computing, computer vision, artificial intelligence, machine learning, deep learning, and thermal sensors.

Classifications

Based on the weight

Based on their weight, drones can be classified into five categories — nano (weighing up to 250 g), Micro air vehicles (MAV) (250 g - 2 kg), Miniature UAV or small (SUAV) (2-25 kg), medium (25-150 kg), and large (over 150 kg)

Based on the degree of autonomy

Drones could also be classified based on the degree of autonomy in their flight operations. ICAO classifies unscrewed aircraft as either remotely piloted aircraft or fully autonomous. Some UAVs offer intermediate degrees of autonomy, for example, a vehicle that is remotely piloted in most contexts but has an autonomous return-to-base operation. Some aircraft types may optionally fly manned or as UAVs, which may include manned aircraft transformed into unscrewed or Optionally Piloted UAVs (OPVs).

Based on the altitude

- Hand-held 2,000 ft (600 m) altitude, about 2 km range
- Close 5,000 ft (1,500 m) altitude, up to 10 km range
- NATO type 10,000 ft (3,000 m) altitude, up to 50 km range
- Tactical 18,000 ft (5,500 m) altitude, about 160 km range
- MALE (medium altitude, long endurance) up to 30,000 ft (9,000 m) and range over 200 km
- HALE (high altitude, long endurance) over 30,000 ft (9,100 m) and indefinite range

- Hypersonic high-speed, supersonic (Mach 1–5) or hypersonic (Mach 5+) 50,000 ft (15,200 m) or suborbital altitude, range over 200 km
- Orbital low Earth orbit (Mach 25+)
- CIS Lunar Earth-Moon transfer
- Computer Assisted Carrier Guidance System (CACGS) for UAVs

Agricultural Robots

The main goal of agricultural robots is the automation of field activities, meaning saving farmers time on dull, regular tasks so that they could focus on more important work.

There is a variety of such robots for different purposes, including seeding, harvesting, weed control, tilling, chemicals application, and more.

Besides, the use of software and hardware complexes for replacing drivers of agricultural vehicles can ensure the reduction of waste and higher yields due to more precise land cultivation.

Ultimately, the world of agriculture slowly moves to automated farming based on the widespread use of mobile and stationary robots.

This is expected to lead to productivity gains coupled with improved margins, resulting in lower production costs.

Here are the prominent tasks of robotization in agriculture:

- Monitoring and forecasting
- Production costs reduction
- Activities precision and quality improvement
- Minimization of food production impact on the environment
- Support of medium and small agricultural businesses
- Increased food safety
- Ability to use agricultural robots in any weather, any time of the day.

Robots for mining

From the mica found in glittery eyeshadows to the coal that helps create energy to bring power to homes, the mining industry often impacts everyday life. But as one of the most dangerous jobs in the world, mining also can mean losing one's life.

Only less than a decade ago, mining jobs only accounted for one percent of the world's labor force but made up eight percent of accidents that end in a fatality, according to the International Labour Organization (ILO). Research also indicates that as many as 1,000 human miners die every year in the

United States for reasons, such as falling, an explosionormachinery. But thanks to advanced technology and automation, it's possible that mines across the world will be run by robots. In fact, mining robots are steadily replacing humans and saving lives simultaneously.

Benefits of Mining Robots

While mining robots are replacing humans in the field, these underground robots also offer the mining industry several benefits. Mining robots not only save human lives, but they also provide enhanced efficiency. That means mining companies can also save more money by having robots that work round-the-clock. Several types of mining robots can help achieve these goals, including self-driving ore trucks, deep-sea mining robots, and automated drill rigs.

Self-Driving Ore Trucks

The same technology that powers self-driving trucks is being used to automate self-driving trucks that carry ore. This makes it easier for mining companies to extract ore from dangerous underground locations that can risk human miners' safety. These autonomous ore-carrying trucks can also maneuver underground easily, thanks to radar and laser technology. For instance, Rio Tinto's driverless haul trucks avoid obstacles by using lasers and radar sensors. These autonomous trucks also use GPS systems for navigation in the ore mines.

Mining Robots that Explore Flooded Mines

Mining robots are also venturing into flooded mines that are abandoned to discover rare minerals. These "roaming" deep-sea mining robots can be constructed to work in confined spaces. They can also help locate unique minerals even with low visibility. These robots also help to reduce the cost of security investments for future explorations and existing exploration.

Drilling Ore from the Earth with Automated Drill Rigs

Mining robots also help save human lives by drilling ore from the ground. These automated drill rigs help carve ore from the Earth. Drilling ore from the ground poses a danger to humans as explosives are required to break apart rocks. Human miners would also need to use conventional equipment to drill the holes where they would place the explosives. But these automated drilling rigs also help speed up productivity to save time in addition to saving human lives. That's because the drilling rigs can create holes faster than human miners can with standard equipment.

Mining Robots of the Future

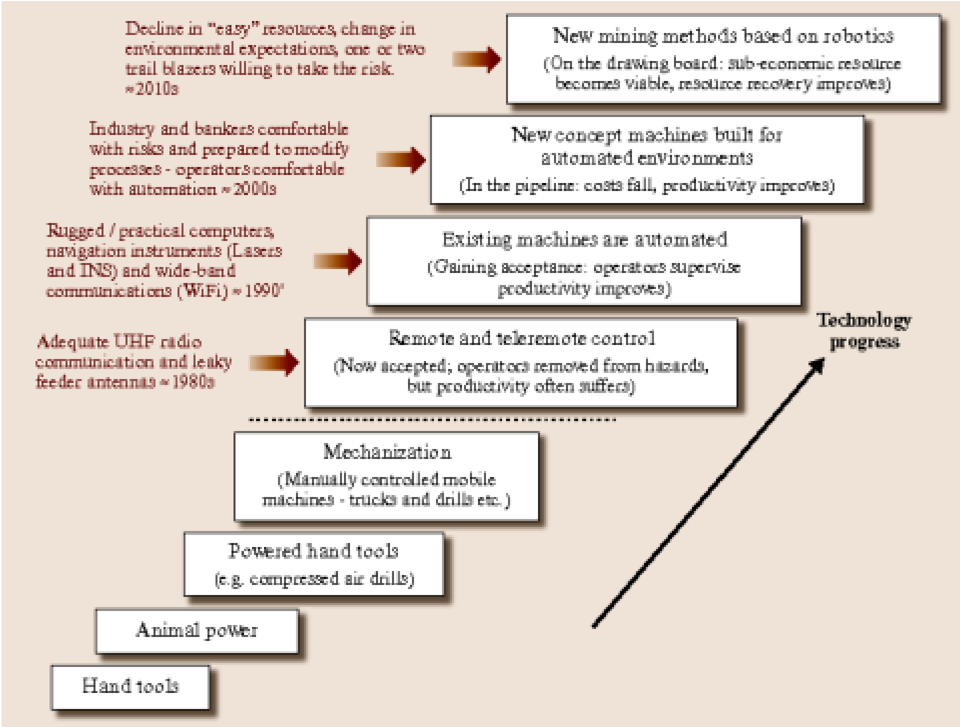


Fig 4.6 Evolution of mining technology

As technology advances, mining robots also have the potential to extract minerals from landscapes humans have never been able to explore successfully. For instance, mining robots will be able to extract minerals deep within the world's oceans, where increased pressure and low visibility makes it dangerous for humans to explore. Also, mining robots have the potential to extract rare minerals from space. Moreover, the mining industry could also see an increased prevalence of driverless trains and other automated mining robots to help improve safety in the industry and enhance productivity.

Robotics for Nuclear Power Plants

Use of robotics and computerized tools in Nuclear Power Plants (NPPs) has been identified as a highly recommended practice by IAEA. The key rationale of robotics application has always been to avoid human exposure to hazardous environments and tasks ranging from scrutiny and general maintenance to decontamination and post accidental activities. To execute these activities, robots need to incorporate artificial intelligence, improved sensors capability, enhanced data fusion and compliant human like leg and hand structures for efficient motions. Next generation robotic systems in NPPs are expected to work in full autonomous mode in contrast to the current semi-autonomous scenarios. Far future systems could deploy humanoid robots as well.

Robotics technology in NPPs was first implemented in US for radioactive material handling. The robot developed by Hughes Aircraft in 1958 was one of the foremost typical applications of robotics technology to reduce human risk associated with the contaminated hazardous environment.

To defuel the reactor core, a remotely controlled manipulator, Rosa, was employed. Moreover a series of three Remote Reconnaissance Vehicles (RRV) was deployed for inspection, surveying and cleaning operations. For the most contaminated zone in the basement of reactor, a RRV the Rover was engaged. Rover was designed for the radioactive inspection of the basement with the help of television cameras and instruments mounted on the robot. The second RRV was used in various parts of the reactor building to remove the radioactive deposits while the third RRV was designed to dismantle small structures in the building and to collect contaminated samples especially from the basement of the reactor. Robotics technology in fact made it possible to visit a no go location. Electric Power Research Institute (EPRI) was the principle contributor in that achievement.

One of the latest robotic systems realized at MIT to replace human inspectors in hazardous environment uses a spherical shaped robot which can change its orientation like an eyeball using a novel proposed mechanism.

Robotics for Space applications

Historically, the role of robotics in space exploration has been significant due to the uninhabitable conditions of non-terrestrial planets in the solar system. According to AZO Robotics, a robot is defined as “a self-controlled device consisting of electronic, electrical or mechanical units that can function in place of a living agent.” In this post, you'll learn some basic history about robots in space exploration and 5 of the most popular robotics systems used in space history.

Before scientists began sending robots to space, animals such as dogs or monkeys were often sent to complete tasks and conduct experiments in order to increase human knowledge of conditions on other planets and the Moon.

As robotic technology has improved, more of these experiments and tasks have been delegated to robots rather than to living beings; in this way robotics preserves lives. While humans have set foot on the Moon, scientific knowledge of conditions on planets including Mars, Venus, Titan and Jupiter comes almost exclusively from exploration conducted by robots. As technology progressively allows robots to reach space faster, humans are able to obtain significantly more data about conditions in space.

5 Incredible Space Robots Used for Space Exploration and Travel:

Space Robot 1: Sputnik 1

The first robot ever sent to space was Sputnik 1, sent by the USSR on October 4, 1957, according to NASA. Sputnik 1 was the first artificial Earth satellite and the first object created by humans to orbit Earth. The launch of Sputnik 1 marked the beginning of the notorious “space race” between the

United States and the USSR. Thereafter, engineers increasingly began constructing robots to be sent to non-terrestrial planets for a variety of purposes ranging from close-up photography of planets to determination of whether other planets sustain life.

Space Robot 2: Mariner 2 and 4

According to Universe Today, on December 14, 1962, the American space probe Mariner 2 became the first robotic space probe to complete a successful Venus flyby. Mariner 4, the first orbiter sent to space, then took the first proximal photos of Mars on July 14, 1965. While the role of landers is primarily to detect signs of life on planets, the role of orbiters is primarily to take photos for scientists to observe and analyze. Since the roles of these robots are different, orbiters and landers have frequently been sent to explore space in tandem.

Space Robot 3: Viking 1 and 2

On August 20, 1975, NASA's Viking project commenced when Viking 1 was launched to explore Mars, according to NASA. Shortly thereafter, on September 9, 1975, an identical spacecraft by the name of Viking 2 was launched with the same mission. Both spacecrafts were equipped with the same robotic technology as Mariners 2 and 4 - a lander and an orbiter. The orbiters' roles were to photograph Mars' surface, while the landers were to land on the planet and collect data in order for scientists to learn more about whether life exists on Mars. The orbiters from each spacecraft flew together, but the landers eventually separated to explore different regions of Mars' surface.

Space Robot 4: Voyager 1 and 2

Robots exited the solar system in 1977 when Voyagers 1 and 2 were launched to study the outer solar system. According to NASA, the two robots were designed to conduct close-up studies of Jupiter and Saturn, Saturn's rings, and both Jupiter and Saturn's largest moon. The two robots have continued on this journey for over 40 years and are presently closer to Pluto than to Earth or the sun. It was not until August 2012 that Voyager 1 entered into interstellar space, while Voyager 2 eventually followed on November 5, 2018. Both robots are still communicating information via the Deep Space Network (DSN) currently.

Space Robot 5: Dextre

More recently, the Canadian Space Association launched Dextre, a robotic arm designed to "install and replace small equipment such as exterior cameras or the 100-kg batteries used on the Space Station, replace defective components in the Station's electrical systems and test new tools and robotics techniques.” According to the Canadian Space Agency, Dextre is technically the most advanced space robot constructed thus far.

SCHOOL OF MECHANICAL ENGINEERING

DEPARTMENT OF MECHANICAL ENGINEERING

UNIT – V – ROBOT PROGRAMMING, ARTIFICIAL INTELLIGENCE AND EXPERT SYSTEMS– SMEA 3012

UNIT 5
ROBOT PROGRAMMING, ARTIFICIAL INTELLIGENCE AND EXPERT SYSTEMS

Robot Programming

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

- Lead through Programming Method
- Textual Robot Languages

Lead through 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 Lead through Method
- Manual Lead through Method

Powered Lead through Method:

The powered lead through 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.

a) Manual Lead through 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 larger arm manually. To get rid of it at each button is implemented in the wrist for special programming. The manual lead through 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 texture l 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 bas c 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 robots 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 center point of the robot and therefore move with it. This system is especially useful when the tool is near to the work piece.

Work piece 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 con roller 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 re 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 more s 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 increasing 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 optimization

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 pro essing.

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 progra ing 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 exp sed 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 graphic lly. We use boxes and arrows in this kind of programming while we want to show dataflow kind of hings.

Visual programming langu ge h s huge application in animations. The last component which we are going to describe is Visual Simulation Environment. VSE provides simulates 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 Motion Manager, whose job is to execute a collection of simultaneously active Motion Commands (MCs) of various types every few milliseconds. The results of these Motion Commands 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 Motion Command that will calculate new states for the LEDs each time the Motion Manager asks for an update. Led MC, a subclass of both Motion Command and Led Engine, performs this service. If we create an instance of Led MC, tell it the frequency at which to blink the LEDs, and add it to the Motion Manager'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 Motion Manager 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 Motion Command 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.