



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 ELECTRICAL AND ELECTRONICS ENGINEERING
DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING**

UNIT-1 INTRODUCTION TO ROBOTICS-SCSA1406

UNIT I

INTRODUCTION AND ROBOTICS

Unit 1 Introduction to Robotics

Types and components of a robot, Classification of robots, closed-loop and open-loop control systems. Kinematics systems; Definition of mechanisms and manipulators, Social issues and safety

Definition for Robot:

The Robot Institute of America (1969) defines robot as a re-programmable, multi-functional manipulator designed to move materials, parts, tools or specialized devices through various programmed motions for the performance of a variety of tasks.

Asimov's laws of robotics:

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

1.1 Components of a Robot:

- **Mechanical platforms or hardware base** is a mechanical device, such as a wheeled platform, arm, fixed frame or other construction, capable of interacting with its environment and any other mechanism involved with its capabilities and uses.
- **Sensors** systems is a special feature that rest on or around the robot. This device would be able to provide judgment to the controller with relevant information about the environment and give useful feedback to the robot.
- **Joints** provide more versatility to the robot itself and are not just a point that connects two links or parts that can flex, rotate, revolve and translate. Joints play a very crucial role in the ability of the robot to move in different directions providing more degree of freedom.
- **Controller** functions as the "brain" of the robot. Robots today have controllers that are run by programs - sets of instructions written in code. In other words, it is a computer used to command the robot memory and logic. So it, be able to work independently and automatically.

- **Power Source** is the main source of energy to fulfill all the robots needs. It could be a source of direct current as a battery, or alternate current from a power plant, solar energy, hydraulics or gas.
- **Artificial intelligence** represents the ability of computers to "think" in ways similar to human beings. Present day "AI" does allow machines to mimic certain simple human thought

processes, but cannot begin to match the quickness and complexity of the brain. On the other hand, not all robots possess this type of capability. It requires a lot of programming and sophisticated controllers and sensorial ability of the robot to reach this level.

- **Actuators** are the muscles of robot. An actuator is a mechanism for activating process control equipment by the use of pneumatic, hydraulic or electronic signals. There are several types of actuators in robotic arms namely synchronous actuator – brush and brushless DC servo, stepper motor and asynchronous actuator – AC servo motor, traction motor, pneumatic, hydraulic.

1.2 Classification of Robots:

The ways of classifying a robot as follows

1) According to the structural capability of robot – i) mobile or ii) fixed robot.

i) **Mobile robot:** A mobile robot is an automatic machine that is capable of locomotion. . Example: spying robot. Mobile robots have the capability to move around in their environment and are not fixed to one physical location. Mobile robots can be "autonomous" (AMR - autonomous mobile robot) which means they are capable of navigating an uncontrolled environment without the need for physical or electro-mechanical guidance devices. Alternatively, mobile robots can rely on guidance devices that allow them to travel a pre-defined navigation route in relatively controlled space (AGV - autonomous guided vehicle). By contrast, industrial robots are usually more-or-less stationary, consisting of a jointed arm (multi-linked manipulator) and gripper assembly (or end effector), attached to a fixed surface

ii) **Fixed Robot:** Most industrial robots are fixed with the base but the arms are moving.

2) According to the control

To perform as per the program instructions, the joint movements an industrial robot must accurately be controlled. Micro-processor-based controllers are used to control the robots. Different types of control that are being used in robotics are given as follows.

a. Limited Sequence Control:

It is an elementary control type. It is used for simple motion cycles, such as pick-and-place operations. It is implemented by fixing limits or mechanical stops for each joint and sequencing the movement of joints to accomplish operation. Feedback loops may be used to inform the controller that the action has been performed, so that the program can move to the next step. Precision of such control system is less. It is generally used in pneumatically driven robots.

b. Playback with Point-to-Point Control

Playback control uses a controller with memory to record motion sequences in a work cycle, as well as associated locations and other parameters, and then plays back the work cycle during program execution. Point-to-point control means individual robot positions are recorded in the memory. These positions include both mechanical stops for each joint, and the set of values that represent locations in the range of each joint. Feedback control is used to confirm that the individual joints achieve the specified locations in the program.

c. Playback with Continuous Path Control

Continuous path control refers to a control system capable of continuous simultaneous control of two or more axes. The following advantages are noted with this type of playback control: greater storage capacity—the number of locations that can be stored is greater than in point-to-point; and interpolation calculations may be used, especially linear and circular interpolations.

d. Intelligent Control

An intelligent robot exhibits behavior that makes it seem to be intelligent. For example, it may have capacity to interact with its ambient surroundings; decision-making capability; ability to communicate with humans; ability to carry out computational analysis during the work cycle; and responsiveness to advanced sensor inputs. They may also possess the playback facilities. However, it requires a high level of computer control, an advanced programming language for decision-making logic and other 'intelligence' into the memory.

ROBOT ANATOMY

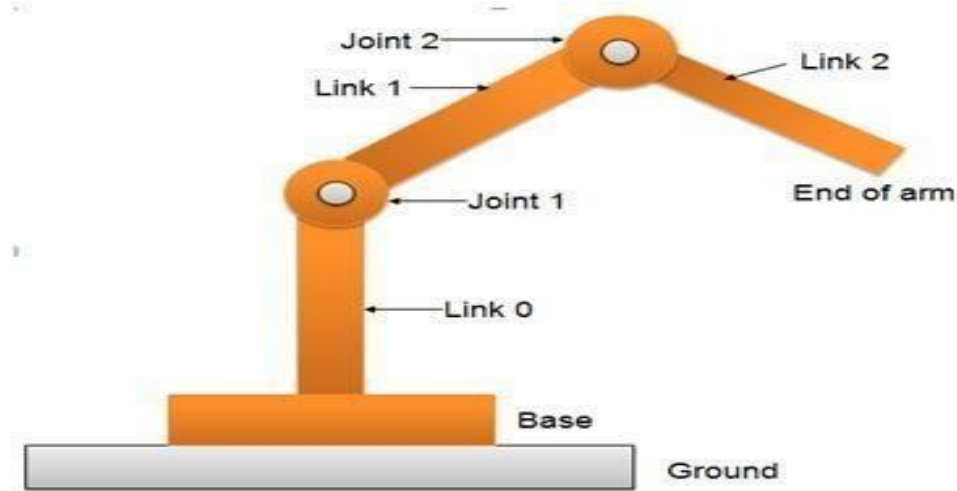


Fig. 1.1 Joint-link scheme for robot manipulator

Joints and Links:

The manipulator of an industrial robot consists of a series of joints and links. Robot anatomy deals with the study of different joints and links and other aspects of the manipulator's physical construction. A robotic joint provides relative motion between two links of the robot. Each joint, or axis, provides a certain degree-of-freedom (dof) of motion. In most of the cases, only one degree-of-freedom is associated with each joint. Therefore the robot's complexity can be classified according to the total number of degrees-of-freedom they possess.

Each joint is connected to two links, an input link and an output link. Joint provides controlled relative movement between the input link and output link. A robotic link is the rigid component of the robot manipulator. Most of the robots are mounted upon a stationary base, such as the floor. From this base, a joint-link numbering scheme may be recognized as shown in Figure 1.1. The robotic base and its connection to the first joint are termed as link-0. The first joint in the sequence is joint-1. Link-0 is the input link for joint-1, while the output link from joint-1 is link-1 which leads to joint-2. Thus link 1 is, simultaneously, the output link for joint-1 and the input link for joint-2. This joint-link-numbering scheme is further followed for all joints and links in the robotic systems.

Nearly all industrial robots have mechanical joints that can be classified into following five types as shown in Figure 1.2.

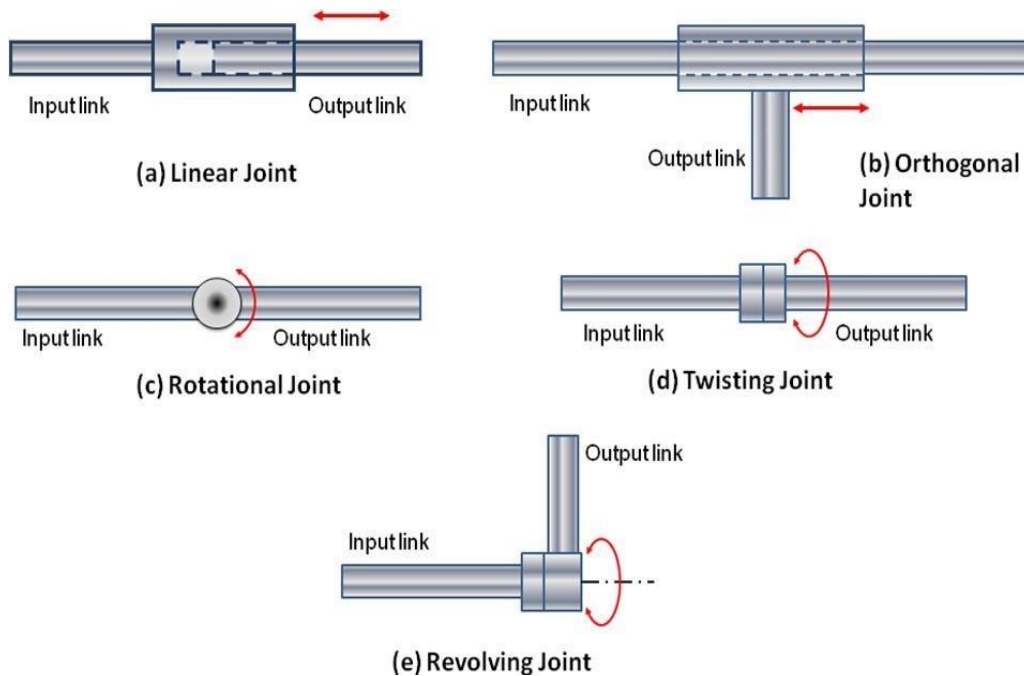


Fig. 1.2 Types of Joints

a) Linear joint (type L joint)

The relative movement between the input link and the output link is a translational sliding motion, with the axes of the two links being parallel.

b) Orthogonal joint (type U joint)

This is also a translational sliding motion, but the input and output links are perpendicular to each other during the movement.

c) Rotational joint (type R joint)

This type provides rotational relative motion, with the axis of rotation perpendicular to the axes of the input and output links.

d) Twisting joint (type T joint)

This joint also involves rotary motion, but the axis of rotation is parallel to the axes of the two links.

e) Revolving joint (type V-joint, V from the “v” in revolving)

In this type, axis of input link is parallel to the axis of rotation of the joint. However the axis of the output link is perpendicular to the axis of rotation.

Robotic arm configurations:

For body-and-arm configurations, there are many different combinations possible for a three-degree-of-freedom robot manipulator, comprising any of the five joint types.

Common body-and-arm configurations are as follows.

- 1) Polar coordinate arm configuration
- 2) Cylindrical coordinate arm configuration
- 3) Cartesian coordinate arm configuration
- 4) Jointed arm configuration

1) Polar coordinate arm configuration(RRP):

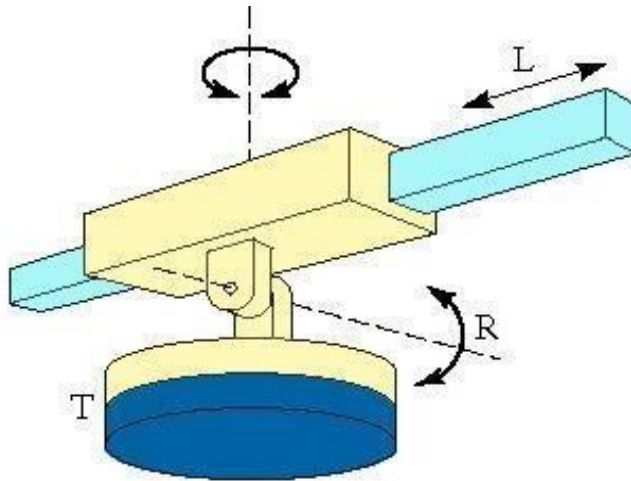


Fig 1.3: 3 DOF polar arm configuration

The polar arm configuration is shown in the fig 1.3. It consists of a prismatic joint that can be raised or lowered about a horizontal revolute joint. The two links are mounted on a rotating base. These various joints provide the capability of moving the arm endpoint within a partial spherical space. Therefore it is called as Spherical coordinated configuration. This configuration allows manipulation of objects on the floor.

- i. Low mechanical stiffness
- ii. Complex construction
- iii. Position accuracy decreases with the increasing radial stroke.

Example: Unimate 2000 series, MAKER 110

The cylindrical configuration uses two perpendicular prismatic joints and a revolute joint as shown in fig 1. 4. This configuration uses a vertical column and a slide that can be moved up or down along the column. The robot arm is attached to the slide, so that it can be moved radially with respect to column. By rotating the column, the robot is capable of achieving a workspace that approximates a cylinder. The cylindrical configuration offers good mechanical stiffness.

Applications: suitable to access narrow horizontal capabilities, hence used for machine loading operations. **Example:** GMF model M-1A.

3) Cartesian coordinate arm configuration (PPP):

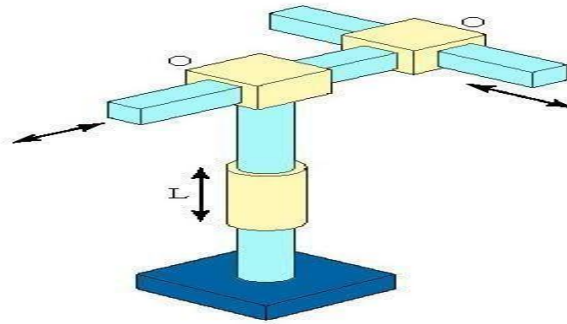


Fig 1.5: 3 DOF Cartesian arm configuration

From fig 1.5. Cartesian coordinate or rectangular coordinate configuration is constructed by three perpendicular slides, giving only linear motions along the three principal axes. It consists of three prismatic joints. The endpoints of the arm are capable of operating in a cuboidal space. Cartesian arm gives high precision and is easy to program.

Drawbacks:

- limited manipulatability
- low dexterity (not able to move quickly and easily)

Applications: use to lift and move heavy loads.

Example: IBM RS-1

4) Jointed arm configuration (RRR) or articulated configuration:

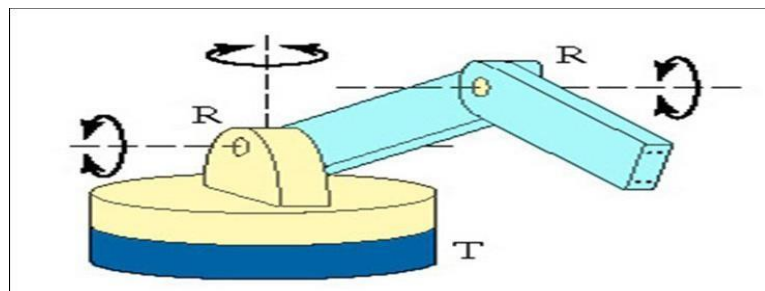


Fig 1.6. 3 DOF jointed arm configuration

From fig 1.6. jointed arm configurations are similar to that of human arm. It consists of two straight links, corresponding to human fore arm and upper arm with two rotary joint

corresponding to the elbow and shoulder joints. These two are mounted on a vertical rotary table corresponding to human waist joint. The work volume is spherical. This structure is the most dexterous one. This configuration is very widely used.

Applications: Arc welding, Spray coating.

Example: SCARA robot (Selective compliance Assembly Robot Arm)

Its full form is 'Selective Compliance Assembly Robot Arm'. It is similar in construction to the jointed-arm robot, except the shoulder and elbow rotational axes are vertical. It means that the arm is very rigid in the vertical direction, but compliant in the horizontal direction.

The SCARA body-and-arm configuration typically does not use a separate wrist assembly. Its usual operative environment is for insertion-type assembly operations where wrist joints are unnecessary. The other four body-and-arm configurations more-or-less follow the wrist-joint configuration by deploying various combinations of rotary joints viz. type R and T.

Robot Wrist:

Wrist assembly is attached to end-of-arm. End effectors are attached to wrist assembly. Function of wrist assembly is to orient end effectors. Body-and-arm determines global position of end effector. It has three degrees of freedom:

- **Roll (R)** axis – involves rotation of the wrist mechanism about the arm axis.
- **Pitch (P)** axis – involves up or down rotation of the wrist.
- **Yaw (Y)** axis - involves right or left rotation of the wrist.

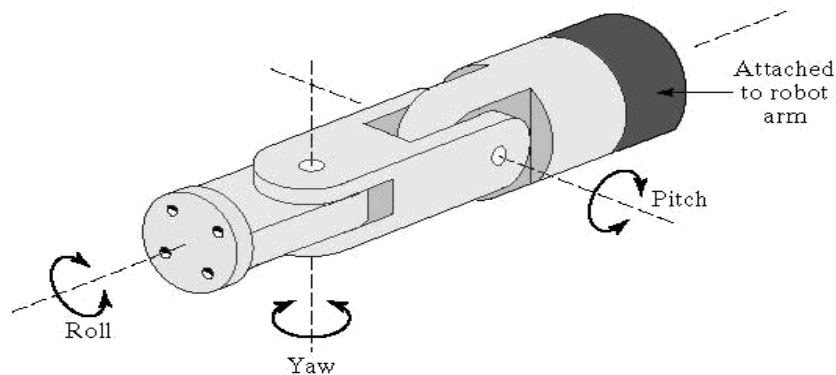


Fig 1.7: Robotic wrist

Robot wrist assembly consists of either two or three degrees of freedom. A typical three-degree-of-freedom wrist joint is depicted in Figure 1.7, the roll joint is accomplished by use of a T joint; the pitch joint is achieved by recourse to an R joint; and the yaw joint, a right-and-left motion, is gained by deploying a second R joint. Care should be taken to avoid confusing pitch and yaw motions, as both utilize R joints.

Degree of freedom:

In mechanics, the degree of freedom (DOF) of a mechanical system is the number of independent parameters that define its configuration. It is the number of parameters that determine the state of a physical system and is important to the analysis of systems of bodies in mechanical engineering, aeronautical engineering, robotics, and structural engineering.

The position and orientation of a rigid body in space is defined by three components of translation and three components of rotation, which means that it has six degrees of freedom.

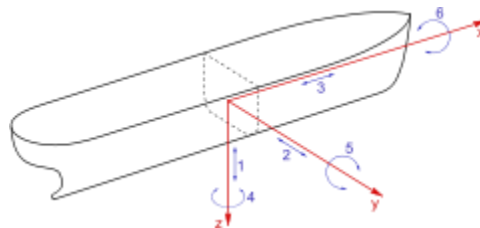


Fig 1.8. Six degrees of freedom of movement of a ship

The motion of a ship at sea has the six degrees of freedom of a rigid body, and is described as shown in fig 1.8.

Translation:

1. Moving up and down (heaving);
2. Moving left and right (swaying);
3. Moving forward and backward (surging);

Rotation:

4. Tilts forward and backward (pitching);
5. Swivels left and right (yawing);
6. Pivots side to side (rolling).

From fig 1.9. The trajectory of an airplane in flight has three degrees of freedom and its attitude along the trajectory has three degrees of freedom, for a total of six degrees of freedom.



Fig 1.9: Attitude degrees of freedom for an airplane.

Robot work volume:

A space on which a robot can move and operate its wrist end is called as a work volume. It is also referred as the work envelope and work space. For developing a better work volume, some of the physical characteristics of a robot should be considered such as:

- The anatomy of various robots
- The maximum value for moving a robot joint
- The size of the robot components like wrist, arm, and body

An industrial robot is a general-purpose, programmable machine possessing certain anthropomorphic characteristics that is, human-like characteristics that resemble the human physical structure, or allow the robot to respond to sensory signals in a manner that is similar to humans. Such anthropomorphic characteristics include mechanical arms, used for various industry tasks, or sensory perceptive devices, such as sensors, which allow robots to communicate and interact with other machines and make simple decisions.

Both robots and numerical control are similar in that they seek to have coordinated control of

multiple moving axes (called joints in robotics). Both use dedicated digital computers as controllers. Robots, however, are designed for a wider variety of tasks than numerical control. Typical applications include spot welding, material transfer (pick and place), machine loading, spray painting, and assembly. The general commercial and technological advantages of robot use are listed.

1.3 Open Loop and Closed Loop Control System:

Open Loop Control System

In this kind of control system, the output doesn't change the action of the control system otherwise; the working of the system which depends on time is also called the open-loop control system. It doesn't have any feedback. It is very simple, needs low maintenance, quick operation, and cost-effective. The accuracy of this system is low and less dependable. The example of the open-loop type is shown below. The main advantages of the open-loop control system are easy, needs less protection; operation of this system is fast & inexpensive and the disadvantages are, it is reliable and has less accuracy.

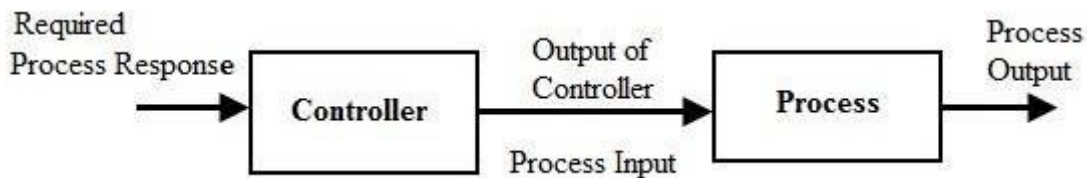


Fig. 1.10 Open Loop Control system

Example

The clothes dryer is one of the examples of the open-loop control system. In this, the control action can be done physically through the operator. Based on the clothing's wetness, the operator will fix the timer to 30 minutes. So after that, the timer will discontinue even after the machine clothes are wet. The dryer in the machine will stop functioning even if the preferred output is not attained. This displays that the control system doesn't feedback. In this system, the controller of the system is the timer.

Closed-Loop Control System

The closed-loop control system can be defined as the output of the system that depends on the input of the system. This control system has one or more feedback loops among its input & output. This system provides the required output by evaluating its input. This kind of system produces the error signal and it is the main disparity between the output and input of the system.



Fig. 1.11 Closed Loop Control system

The main advantages of the closed-loop control system are accurate, expensive, reliable, and requires high maintenance.

Example

The best example of the closed-loop control system is AC or air conditioner. The AC controls the temperature by evaluating it with the nearby temperature. The evaluation of temperature can be done through the thermostat. Once the air conditioner gives the error signal is the main difference between the room and the surrounding temperature. So the thermostat will control the compressor. These systems are accurate, expensive, reliable, and requires high maintenance.

1.4 Kinematics Systems:

Kinematics is the study of the relationship between a robot's joint coordinates and its spatial layout, and is a fundamental and classical topic in robotics. Kinematics can yield very accurate calculations in many problems, such as positioning a gripper at a place in space, designing a mechanism that can move a tool from point A to point B, or predicting whether a robot's motion would collide with obstacles. Kinematics is concerned with only the instantaneous values of the robot's coordinates, and ignores their movement under forces and torques. The kinematics problem may be rather trivial for certain robots, like mobile robots that are essentially rigid bodies, but requires involved study for other robots with many joints, such as humanoid robots and parallel mechanisms.

Robot kinematics describe the kinematics of several common robot mechanisms and define the concepts of configuration space and workspace. It will also present the process of forward kinematics, which performs the geometric calculations needed to map configuration space to workspace.

1 Configuration space and workspace

A robot's kinematic structure is described by a set of *links*, which for most purposes are considered to be rigid bodies, and *joints* connecting them and constraining their relative movement, for example, rotational or translational joints.

A robot's layout, at some instant in time, can be described by one of two methods:

1. A list of coordinates for each joint (typically an angle or translation distance) expressed relative to some *reference frame*, aka *zero position*.
2. A spatial representation of its links in the 2D or 3D world in which it operates, e.g., matrices describing the frame of each link relative to some world coordinate system.

The list of joint coordinates are known as the *configuration* of the robot. The 2D or 3D world in which the robot lives is known as its *workspace*.

The importance of a configuration is that it is a *non-redundant, minimal representation* of the robot's layout. This stands in contrast to the representation of storing each link's frame (also known as *maximal coordinates*), but the constraints imposed by each joint might not be satisfied by a given maximal coordinate representation.

1.1 Defining robot structures

A wide variety of robot mechanisms can be described by categorizing their arrangement of joints and joint types. For the moment we will ignore the size and shape of links, and simply focus on broad categorization.

First, there are three typical joint types, each describing the form of relative transformations allowed between the two links to which it is attached:

- Revolute: the attached links rotate about a common axis.
- Prismatic: the attached links translate about a common axis.
- Spherical: the attached links rotate about a point.

More exotic joints, like helical (screw) joints, may also exist. One may also speak of *fixed joints* where the attached links are rigidly fixed together; since mathematically the two links could be considered as one, this is primarily for representational convenience. It is customary to refer to one of the attached links as the *parent* and the other the *child*.

Second, mechanisms can be described by their *topology*, which describes how links and joints interconnect:

- Serial: the links and joints form a single ordered chain, with the child link of one joint being the parent of the next.
- Branched: each link can have zero or more child links, but cutting any joint would detach the system into two disconnected mechanisms. Like a human body, in which fingers are attached to the hand, toes are attached to the feet, and arms, legs, and head are attached to the torso.
- Parallel: the series of joints forms at least one closed loop. I.e., there exist joints that, if cut, would not divide the system into two disconnected halves.

The topology can be inspected by plotting a *link graph*, which is a network structure in which vertices are links and edges are joints. Serial mechanisms have a linear link graph, branched mechanisms are trees (i.e., graphs without loops), and parallel mechanisms have loops.

Serial mechanisms are usually characterized using an alphanumeric notation which lists the initials of the joint types in order from the base down the chain. For simplicity, when multiple joints of the same type are repeated, like "XXX", this is listed as "#X" where "#" is the number of repetitions. Examples include:

- 3P (PPP): xyz gantry
- 3P3R (PPPRRR): 6-axis CNC machine
- 6R (RRRRRR): revolute joint industrial robot

A third characterization defines whether the robot is affixed to the world or left free to move in space:

- Fixed base: a base link is rigidly affixed to the world, like in an industrial robot.
- Floating base: all links are free to rotate and translate in workspace, like in a humanoid robot.
- Mobile base: the workspace is 3D, but a base link can rotate and translate on a 2D plane, like in a car.

1.2 Configurations and configuration space

As mentioned above, the configuration of a robot is a minimal set of coordinates defining the position of all links. For serial or branched fixed-base mechanisms, this is simply a list of individual joint coordinates. For floating/mobile bases, the configuration is slightly more complex, requiring the introduction of *virtual linkages* to account for the movement of the base link. The situation for parallel mechanisms is even more complex, and we will withhold this discussion for later.

1.2.1 Degrees of freedom

The *degrees of freedom* (dof) of a system define the span of its freely and independently moving dimensions, and the number of degrees of freedom is also known as its *mobility* MM. In the case of a serial or branched fixed base mechanism, the degrees of freedom are the union of all individual joint degrees of freedom, and the mobility is the sum of the mobilities of all individual joints:

$$M = \sum_{i=1}^n f_i \dots\dots\dots (1)$$

where there are n joints and f_i is the mobility of the i 'th joint, with $f_i=1$ for revolute, prismatic, and helical joints, and $f_i=3$ for spherical joints.

The degrees of freedom for a single joint are expressed as the offset of the two attached links from their layout in a given *reference frame*. For revolute joints, the one dof is a joint angle defining the offset from a joint's zero position along its axis of rotation. For prismatic joints, the one dof is a translation along the axis relative to its zero position. Spherical joint dofs can be represented by Euler angles.

1.2.2 Floating bases and virtual linkages

For floating and mobile bases, the movement of the robot takes place not only via joint movement but also of the overall translation and rotation of the mechanism in space. As a result the number of degrees of freedom are increased. To represent this in a more straightforward manner, we treat floating base robots as fixed-base robots by means of attaching a *virtual linkage* that expresses the mobility of the root link.

It may be improper to think of a "base link" because there is no link attached to the environment, but it is customary to speak of a *root link* from which calculations begin. For a 2D floating base, the (x,y) translation and rotation θ of the robot's root link with respect to its reference frame can be expressed as a virtual linkage of additional 2PR manipulator. A similar construction gives the virtual linkage for a robot with a mobile base.

In 3D floating base robots, the virtual linkage is customarily treated as a 3P3R robot with degrees of freedom corresponding to the (x,y,z) translation of the root link and the Euler angle representation (ϕ, θ, ψ) of its rotation. Any Euler angle convention may be used for this linkage, except that it is often advisable not to use conventions that have a singularity at the identity. In the future we shall use roll-pitch-yaw (ZYX) convention.

As a result of the inclusion of the virtual linkage, for a floating base in 3D, the mobility is increased by 6: $M=6+\sum_{i=1}^n f_i$. In 2D, or for mobile bases in 3D, mobility is increased by 3.

1.2.3 Joint limits and configuration space

Joint mobility is usually limited by mechanical limitations or physical stops. Such prismatic and revolute joints will be associated with *joint limits*, which define an interval of joint values [a,b] that are valid irrespective of the configuration of the remaining links.

Some revolute joints may have no stops, such as a motor driving a drill bit or wheel, and these are known as continuous rotation joints. The revolute joints associated with virtual linkages also have continuous rotation. In these cases, the joint's degree of freedom moves in SO(2).

The Cartesian product of all joint ranges is the *configuration space* of the robot.

As an example, consider a 2RPR mechanism where all the axes are aligned with the Z axis. The first two joints define position in the (x,y)plane, and are limited to the range $[-\pi/2, \pi/2]$. The third joint moves a drill up and down in the range [zmin,zmax], and the final joint drives the continuous rotation of the drill bit. Here, the configuration space is $[-\pi/2, \pi/2]^2 \times [zmin, zmax] \times SO(2)$ (2)

1.2.4 Configurations for parallel mechanisms

Often, it is significantly harder to determine the configuration space of parallel mechanisms. We can no longer consider each joint independent, since the movement of each joint in a closed loop affects the movement of other joints. However, there is a formula to determine the mobility MM of these mechanisms.

Conceptually, the formula calculates the number of dofs of the maximal coordinate representation, and then subtracts the number of dofs removed by each joint. That is, if there are n links and m joints, each with mobility f_1, \dots, f_m , then the mobility is given by

$$M=3n-\sum_{j=1}^m (3-f_j) \dots\dots\dots(3)$$

in 2D and

$$M=6n-\sum_{j=1}^m (6-f_j) \dots\dots\dots(4)$$

in 3D.

As an example, for a 4-bar linkage in 2D, there are $n=4$ links and $m=4$ joints each with mobility 1. If one link is frozen to the environment, this could be considered a fifth fixed joint with mobility 0. Hence, the mobility is:

$$M=3 \cdot 4 - 4 \cdot (3-1) - (3-0) = 12 - 8 - 3 = 1$$

which indicates the entire structure has only a single degree of freedom.

1.3 End effectors and reachable workspaces

"Workspace" is somewhat of an overloaded term in robotics; it is also used to refer to the range of positions and orientations of a certain privileged link, known as the *end effector*. End effectors are typically at the far end of a serial chain of links, and are often where tool points are located since these links have the largest range of motion. Depending on context, the workspace may refer to positions only, both positions and orientations, or, less frequently, orientations only. (It is due to this ambiguity that some authors prefer the term "task space" to speak specifically of an end-effector's spatial range, but the dual usage of "workspace" is widespread in the field.)

The *reachable workspace* of an end effector is the region in workspace that can be reached by any valid configuration, in the absence of obstacles. Typically the notion of "validity" is defined such that joint limits are respected, but other constraints like self-collision avoidance may also be respected as well. The size and shape of the reachable workspace are important to consider when designing or selecting a robot for a given task, as well as determining the location to place a fixed-base robot in its workcell.

Calculating the reachable workspace in 2D or 3D space of an end effector's position can be done through a recursive geometric construction: first sweep the point about the range of motion of the last joint to obtain a curve, then sweep the curve about the range of motion of the second-to-last joint to obtain a surface, and then sweep the surface about the third-to-last to obtain a volume, and so on.

The process becomes more challenging when orientation is also considered, but this is nevertheless extremely important to consider for most robots since the orientation of the end effector must often be constrained to perform the desired function. In 2D space, the reachable workspace can be pictured as a 3D volume, with (x,y) components on the plane and θ plotted on the z axis. In 3D space the combined position and orientation workspace is 6D, which is very hard to compute or visualize. Instead, one may speak of a *fixed-orientation reachable workspace* which contains the range of end effector positions reachable with the orientation of the end effector held fixed at some useful angle (for example, pointing up or down or sideways, depending on the task).

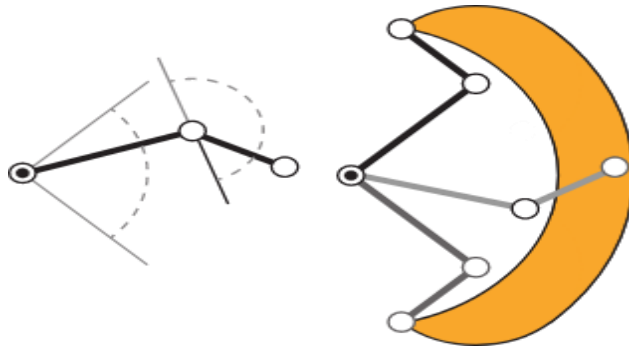


Fig . 1.12 : A 2R manipulator

with joint limits of $\pm 45^\circ$ for the first joint and $\pm 90^\circ$ for the second, and a second link somewhat shorter than the first. Treating the position of the end effector as the workspace coordinate, the reachable workspace is a portion of an annulus (planar donut shape).

1.5 Definition of mechanisms and manipulators

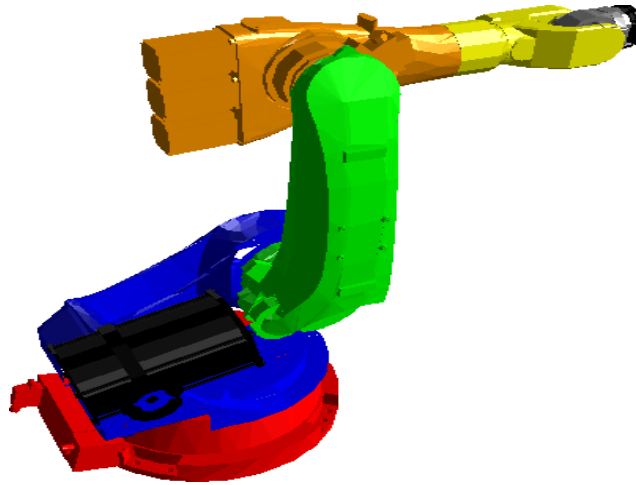


Fig. 1.13: Programmable Universal Manipulator Arm (PUMA)

A robot manipulator is an electronically controlled mechanism, consisting of multiple segments, that performs tasks by interacting with its environment. They are also commonly referred to as robotic arms. Robot manipulators are extensively used in the industrial manufacturing sector and also have many other specialized applications (for example, the Canadarm was used on space shuttles to manipulate payloads). The study of robot manipulators involves dealing with the positions and orientations of the several segments that make up the manipulators. This module introduces the basic concepts that are required to describe these positions and orientations of rigid bodies in space and perform coordinate transformations.

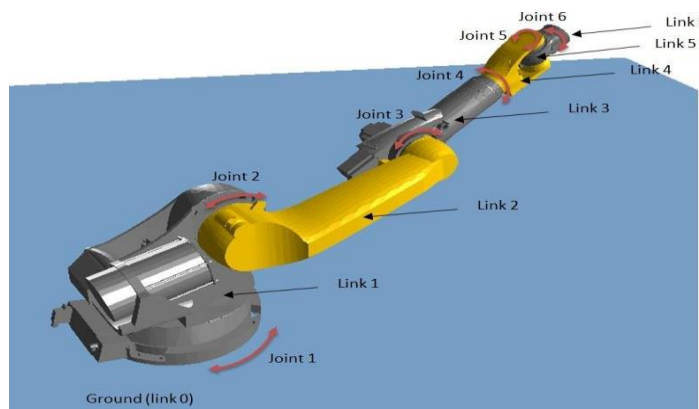







Fig. 1.14: Links and joints

Types of Joints

Joints allow restricted relative motion between two links. The following table describes five types of joints.

Table 1.1: Types of joints

Name of joint	Representation	Description
Revolute		Allows relative rotation about one axis.
Cylindrical		Allows relative rotation and translation about one axis.
Prismatic		Allows relative translation about one axis.
Spherical		Allows three degrees of rotational freedom about the center of the joint. Also known as a ball-and-socket joint.
Planar		Allows relative translation on a plane and relative rotation about an axis perpendicular to the plane.

Some Classification of Manipulators

Manipulators can be classified according to a variety of criteria. The following are two of these criteria:

By Motion Characteristics

Planar manipulator: A manipulator is called a planar manipulator if all the moving links move in

planes parallel to one another.

Spherical manipulator: A manipulator is called a spherical manipulator if all the links perform spherical motions about a common stationary point.

Spatial manipulator: A manipulator is called a spatial manipulator if at least one of the links of the mechanism possesses a general spatial motion.

By Kinematic Structure

Open-loop manipulator (or serial robot): A manipulator is called an open-loop manipulator if its links form an open-loop chain.

Parallel manipulator: A manipulator is called a parallel manipulator if it is made up of a closed-loop chain.

Hybrid manipulator: A manipulator is called a hybrid manipulator if it consists of open loop and closed loop chains.

Degrees of Freedom

The number of degrees of freedom of a mechanism are defined as the number of independent variables that are required to completely identify its configuration in space.

The number of degrees of freedom for a manipulator can be calculated as

.....(1)

$$n_{dof} = \lambda(n-1) - \sum_{i=1}^k (\lambda - f_i)$$

where n is the number of links (this includes the ground link), k is the number of joints, f_i is the number of degrees of freedom of the i th joint and λ is 3 for planar mechanisms and for spatial mechanisms.

Manipulators

Manipulators are composed of an assembly of links and joints. Links are defined as the rigid sections that make up the mechanism and joints are defined as the connection between two links. The device attached to the manipulator which interacts with its environment to perform tasks is called the end-effector. In Fig. 1.14, link 6 is the end effector.

Robot Kinematics are divided into two types namely

- i) Forward Kinematics
- ii) Inverse Kinematics

Forward Kinematics:

- We used known joint variables (i.e. servo motor angles, displacement of a linear actuator, etc.) to calculate the position and orientation of the end effector of a robotic arm (e.g. robotic gripper, robotic hand, vacuum suction cup, etc.) in 3D space. This is called **forward kinematics**.
- Forward kinematics asks the question: Where is the end effector of a robot (e.g. gripper, hand, vacuum suction cup, etc.) located in space given that we know the angles of the servo motors?

Inverse Kinematics:

Inverse kinematics is the forward kinematics problem in reverse. We know the position and orientation we want the end effector of a robotic arm to have, and we want to find the values of the joint variables that generate that desired position and orientation of the end effector.

- Joint Variables \longrightarrow Pose of the End effector of a Robotic Arm = **Forward Kinematics**
- Pose of the End effector of a Robotic Arm \longrightarrow Joint Variables = **Inverse Kinematics**

Inverse kinematics asks the question: What do the angles of the servo motors need to be given our desired position and orientation of the end effector of a robotic arm (e.g. gripper, hand, vacuum suction cup, etc.)?

1.6 Social issues and Safety:

“Unintelligent”, stationary industrial robots are used mostly in production systems equipped with NC (numerical controlled), machines as well as in CIM (computer integrated) or imms (intelligent manufacturing) - systems. Currently there are worldwide approximately 1.2 millions working in industry. With a 7th and 8th axis they can be limited movable to extend the working space. They are nowadays equipped with simple external sensors for “intelligent” operations e.g. assembly and disassembly, fuelling cars... and are called “intelligent” robots.

Mobile robots could be divided in three categories. “classic” mobile robots are partially intelligent mobile platforms. As “Autonomous Guided Vehicles – AGV’s “ they are available since some years in industry and equipped with additional external sensors (Intelligent Autonomous Guided Vehicles – Intelligent AGV’s) covering a broad application field. Movement possibilities are wheels, chains. Intelligent industrial and mobile robots are used for service tasks – “service robots”.

“Advanced” mobile robots are currently in development and exist mostly as prototypes.

Walking machines or mechanisms are well known since some decades. Usually they have more than 6 (snake), 4 (multi-legged) to 6 (hexapod), 2 (biped) or one leg (hopping). Walking on two legs is from

the view point of control engineering a complex stability problem. Biped walking machines equipped with external sensors are the basis for “humanoid” robots. Some prototypes of such robots are available today.

One of the current trends in robotics is cooperation. Industrial robots are connected by their controllers for synchronization or controlled by one controller. Latest developments deal with a modularization of the robots as well as the control system.

Mobile platforms with external sensors are available since some years and cover a broad field of new applications. They are the basis of mobile robot platforms. On such platforms various devices, like arms, grippers, transportation equipment, etc., can be attached. Communication between the On - board PC“ and the supervisory PC“ is carried out by WLAN or bus systems like CAN - communication with the environment can be accomplished by voice.

Possible applications including tele-operation or semi- autonomous operation of robot platforms in various scenarios could be: Factory automation: operation in hazardous environments, planetary and space exploration, deep-sea surveying and prospecting, services.

Service robots“ are mobile robots adapted for service tasks: For personal use - e.g. cleaning robots, lawnmowers, for healthcare e.g. assistance for handicapped, for leisure and hobby, e.g. game playing, sports (soccer,...).

Biped walking robots are much more flexible than robots with other movement possibilities. The main advantage of legged robots is the ability to move in a rough terrain without restrictions like wheeled and chained robots. Legged robots can work in environments which were until now reserved only for humans. Especially fixed and moved obstacles can be surmounted by legged robots. In addition to walking such robot could realize other movements like climbing, jumping. Intelligent robots – especially intelligent, mobile platforms and humanoid robots are able to work together on a common task in a cooperative way.

Service Robots are robots of several shapes and sizes and support and back up human operators. These robots can guarantee a better quality of life, providing that designers guarantee safety and security.

Humanoid robots realize the old dream of humans and are able to assist them. From the viewpoint of ethics they support a lot of activities in order to increase the human quality of life.

Intelligent machines can assist humans to perform very difficult tasks, and behave like true and reliable companions. In many ways and are or will be connected to the internet. This yields to the remote human-robot interaction for tele- operation and tele-presence. This and will permit robot-robot interaction for data-sharing and cooperative working and learning and is a pre-stage of cloud robots.

From a social and psychological standpoint, overuse could lead to technology addiction or invasion of privacy. Humans in robotized environments could face psychological problems.

Many applications of mobile robots are to explore, develop, secure, and feed our world and worlds beyond land e.g. demining, sea e.g. offshore, air (e.g. UAV) or space e.g. space exploration.

Robotic systems for health care - medical robots have made their way into the operating room. Biomechatronic human prostheses for locomotion, manipulation, vision, sensing, and other functions are still in use e.g. robotic tele-surgical workstations, robotic systems for diagnosis, robots for therapy, haptic interfaces for surgery/physiotherapy training, artificial limbs (legs, arms), internal organs (heart, kidney), senses (eye, ears, etc.), exoskeletons,. From the social and ethical standpoint, this is one of the fields in robotics that suffers from the most difficult safety and ethical problems, e.g. a robot can be only serve as an assistant of a human surgeon but it will never replace the surgeon.

Lifestyle of young people has changed. Robotics is a very good tool to teach technology while, at the same time, always remaining very tightly anchored to reality. Robots will enable us to build real environments. A very good example for *Edutainment* (Education by Entertainment) are robot competitions e.g. robot soccer.

The fascinating idea of using small robot cubicles to play soccer was born just a decade ago. Robot soccer was introduced to develop intelligent cooperative multi-robot (agents) systems (MAS) and to bring young generation the difficult scientific and engineering subjects easy in the way of playing. From the scientific viewpoint the soccer robot is an intelligent autonomous agent which carries out tasks with other agents in a cooperative, coordinated, and communicative way. It is also a good tool for spending leisure time and for education. There are numerous possibilities to use robots, where it could become unpleasant for humans. In addition, they can fulfill everyday tasks. This possibility causes many human concerns and fears.

Currently the ethical behavior of a robot is determined by the software. The features of the software depend directly from the programmers. That means roboethics is closely connected to the ethical

behavior of the software developer. If a robot is good or evil depends mostly from the software or from the ethical behavior of a human.

It is almost inevitable that human designers are inclined to replicate their own conception of intelligence in the intelligence of robots. In turn, the former gets wired into the control algorithm of the robots. Robotic intelligence is a learned intelligence, fed by the world models uploaded by the designers. It is a self-developed intelligence, evolved through the experience which robots have gained through the learned effects of their actions. Robot intelligence also includes the ability to evaluate and attribute a judgment to the actions carried out by robots.

TEXT/REFERENCE BOOKS:

- 1.Saha, S.K., “Introduction to Robotics, 2nd Edition, McGraw-Hill Higher Education, New Delhi, 2014.
2. Ashitava Ghosal., “Fundamental Concepts and Analysis”, Oxford, New Delhi, 2006.
3. Mikell.P.Groover, Mitchell Weiss, Roger.N.Nagel, Nicholas.G.Odrey, “Industrial Robotics-Technology, Programming and Applications”, Tata McGraw-Hill Publishing Company Limited, New Delhi, Third Reprint 2008



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 ELECTRICAL AND ELECTRONICS ENGINEERING
DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING**

UNIT II – ROBOT KINEMATICS AND DYNAMICS - SCSA1406

UNIT II

ROBOT KINEMATICS AND DYNAMICS

Unit 2: Robot Kinematics and Dynamics

Kinematic Modelling: Translation and Rotation Representation, Coordinate transformation, DH parameters, Jacobian, Singularity, and Statics Dynamic Modelling: Equations of motion: Euler-Lagrange formulation

Robot kinematics applies geometry to the study of the movement of multi-degree of freedom kinematic chains that form the structure of robotic systems. The emphasis on geometry means that the links of the robot are modeled as rigid bodies and its joints are assumed to provide pure rotation or translation.

Robot kinematics studies the relationship between the dimensions and connectivity of kinematic chains and the position, velocity and acceleration of each of the links in the robotic system, in order to plan and control movement and to compute actuator forces and torques. The relationship between mass and inertia properties, motion, and the associated forces and torques is studied as part of robot dynamics. The robot kinematics concepts related to both open and closed kinematics chains. Forward kinematics is distinguished from inverse kinematics.

SERIAL MANIPULATOR:

Serial manipulators are the most common industrial robots. They are designed as a series of links connected by motor-actuated joints that extend from a base to an end-effector. Often they have an anthropomorphic arm structure described as having a "shoulder", an "elbow", and a "wrist". Serial robots usually have six joints, because it requires at least six degrees of freedom to place a manipulated object in an arbitrary position and orientation in the workspace of the robot. A popular application for serial robots in today's industry is the pick-and-place assembly robot, called a SCARA robot, which has four degrees of freedom.



Fig 2.1 SCARA robot

STRUCTURE:

In its most general form, a serial robot consists of a number of rigid links connected with joints. Simplicity considerations in manufacturing and control have led to robots with only revolute or prismatic joints and orthogonal, parallel and/or intersecting joint axes. The inverse kinematics of serial manipulators with six revolute joints, and with three consecutive joints intersecting, can be solved in closed-form, i.e. analytically. This result had a tremendous influence on the design of industrial robots.

The main advantage of a serial manipulator is a large workspace with respect to the size of the robot and the floor space it occupies. The main disadvantages of these robots are:

- The low stiffness inherent to an open kinematic structure,
- Errors are accumulated and amplified from link to link,
- The fact that they have to carry and move the large weight of most of the actuators, and
- The relatively low effective load that they can manipulate.

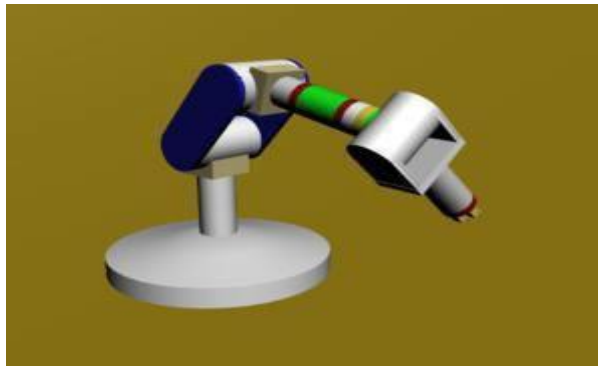


Fig. 2.2 Serial manipulator with six DOF in a kinematic chain

PARALLEL MANIPULATOR:

A parallel manipulator is a mechanical system that uses several computer-controlled serial chains to support a single platform, or end-effector. Perhaps, the best known parallel manipulator is formed from six linear actuators that support a movable base for devices such as flight simulators. This device is called a Stewart platform or the Gough-Stewart platform in recognition of the engineers who first designed and used them.

Also known as parallel robots, or generalized Stewart platforms (in the Stewart platform, the actuators are paired together on both the basis and the platform), these systems are articulated robots that use similar mechanisms for the movement of either the robot on its base, or one or more manipulator arms. Their 'parallel' distinction, as opposed to a serial manipulator, is that the end effector (or 'hand') of this linkage (or 'arm') is connected to its base by a number of (usually three or six)

separate and independent linkages working in parallel. 'Parallel' is used here in the computer science sense, rather than the geometrical; these linkages act together, but it is not implied that they are aligned as parallel lines; here *parallel* means that the position of the end point of each linkage is independent of the position of the other linkages.

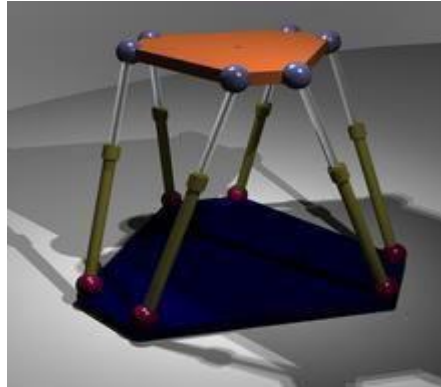


Fig: 2.3 Abstract render of a Hexapod platform (Stewart Platform)

Forward Kinematics:

It is used to determine where the robot's hand, if all joint variables are known)

Inverse Kinematics:

It is used to calculate what each joint variable, if we desire that the hand be located at a particular point.

ROBOTS AS MECHANISMS

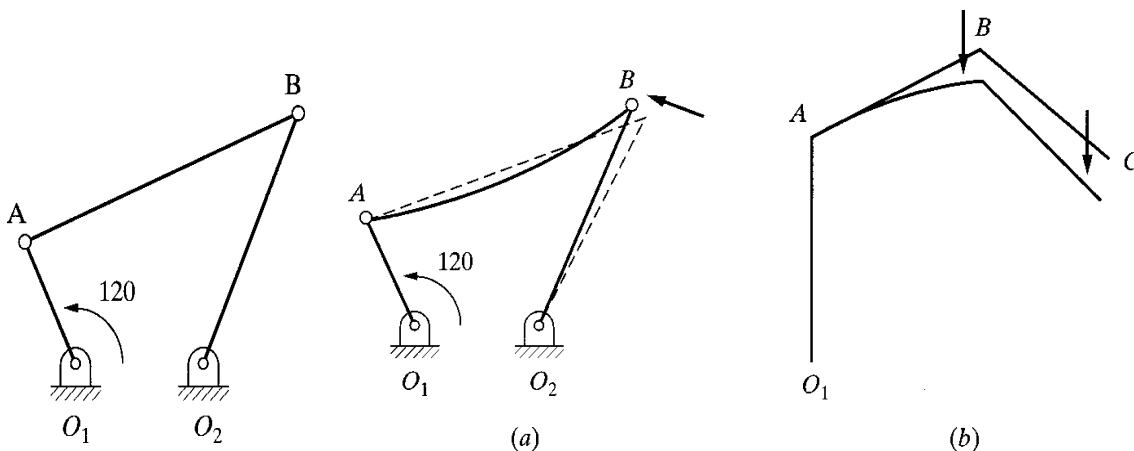


Fig. 2.4a one-degree-of-freedom closed-loop(a) Closed-loop versus (b) open-loop mechanism Four-bar mechanism

MATRIX REPRESENTATION

Representation of a Point in Space

A point P in space: 3 coordinate relative to a reference frame

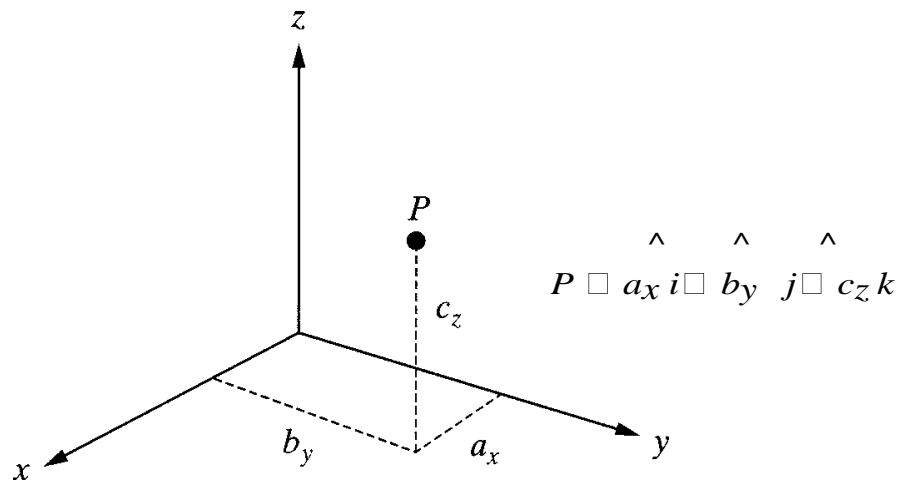


Fig 2.5 Representation of a point in space

Representation of a Vector in Space

A Vector P in space: 3 coordinates of its tail and of its head

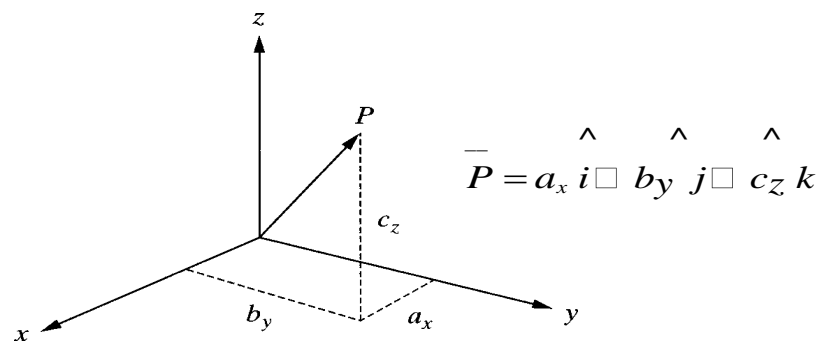


Fig 2.6 Representation of a vector in space

$$\vec{P} = \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix}$$

Representation of a Frame at the Origin of a Fixed-Reference Frame

Each Unit Vector is mutually perpendicular: normal, orientation, approach vector

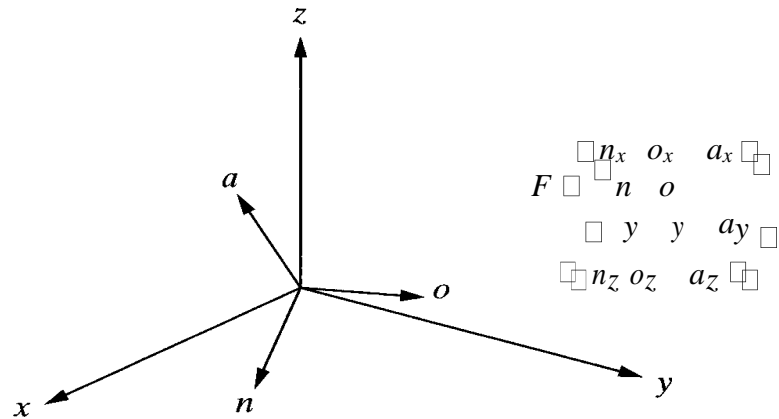


Fig. 2.7 Representation of a frame at the origin of the reference frame

Representation of a Frame in a Fixed Reference Frame

Each Unit Vector is mutually perpendicular: normal, orientation, approach vector

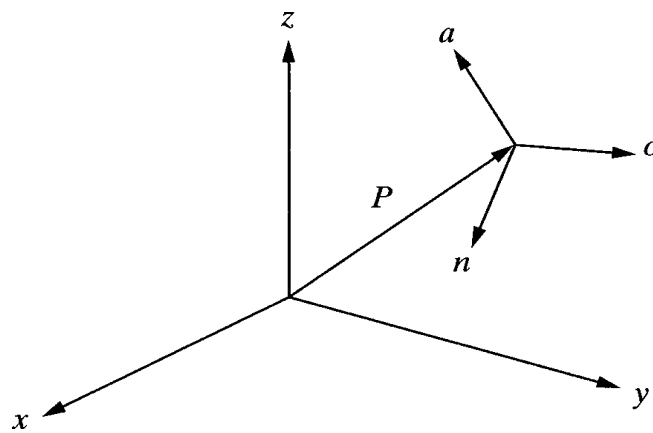


Fig.2.8 Representation of a frame in a frame

$$F = \begin{bmatrix} n_x & o_x & a_x & P_x \\ n_y & o_y & a_y & P_y \\ n_z & o_z & a_z & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Representation of a Rigid Body

An object can be represented in space by attaching a frame to it and representing the frame in space.

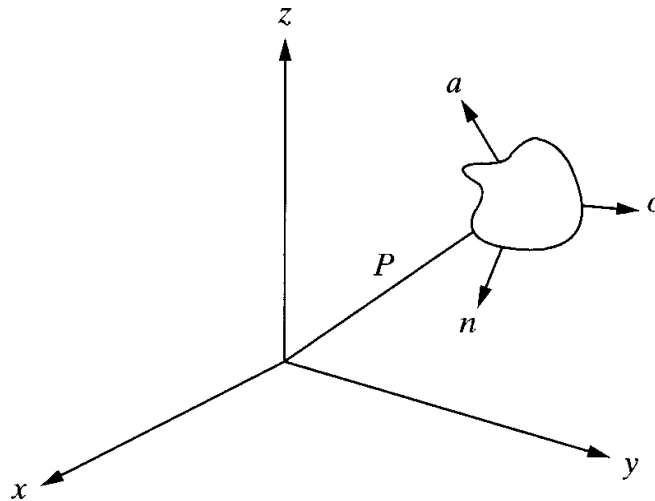


Fig. 2.9 Representation of an object in space

$$F_{object} = \begin{bmatrix} n_x & o_x & a_x & P_x \\ n_y & o_y & a_y & P_y \\ n_z & o_z & a_z & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

HOMOGENEOUS TRANSFORMATION MATRICES

Transformation matrices must be in square form. It is much easier to calculate the inverse of square matrices. To multiply two matrices, their dimensions must match.

Representation of a Pure Translation

- ◆ A transformation is defined as making a movement in space.
- ◆ A pure translation.
- ◆ A pure rotation about an axis.
- ◆ A combination of translation or rotations

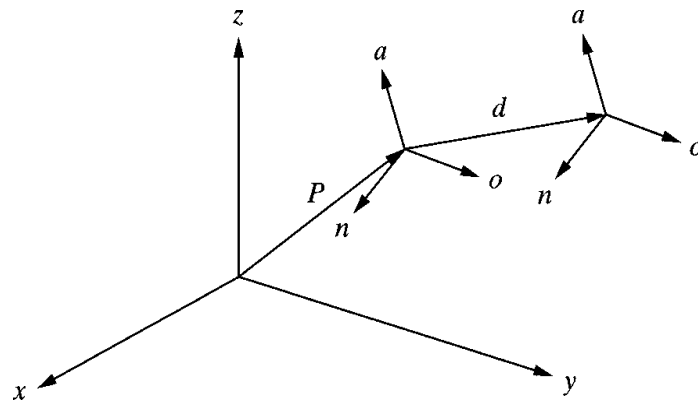


Fig. 2.10 Representation of a pure translation in space

$$T = \begin{bmatrix} 1 & 0 & 0 & d_x \\ 0 & 1 & 0 & d_y \\ 0 & 0 & 1 & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Representation of a Pure Rotation about an Axis

Assumption: The frame is at the origin of the reference frame and parallel to it.

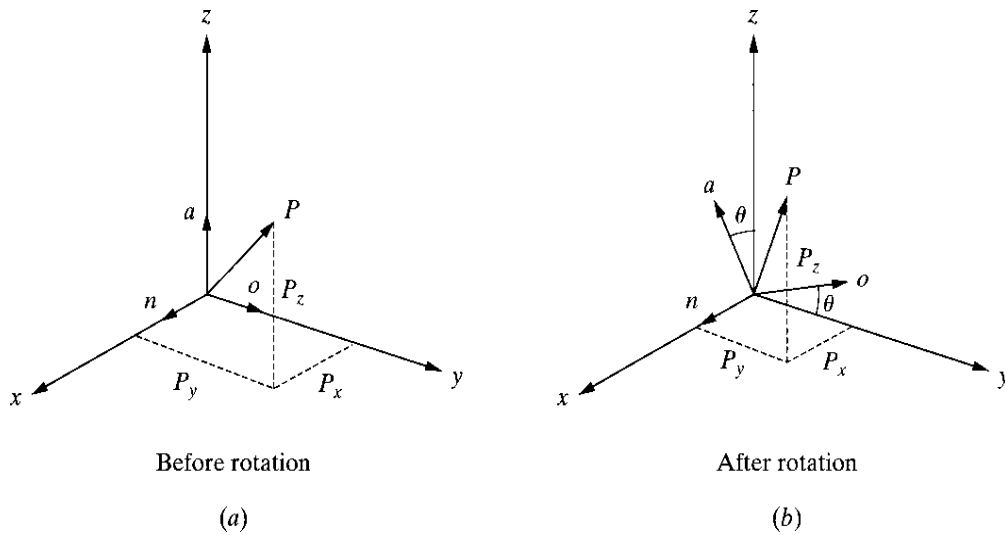


Fig.2.11 Coordinates of a point in a rotating frame before and after rotation



A number of successive translations and rotations



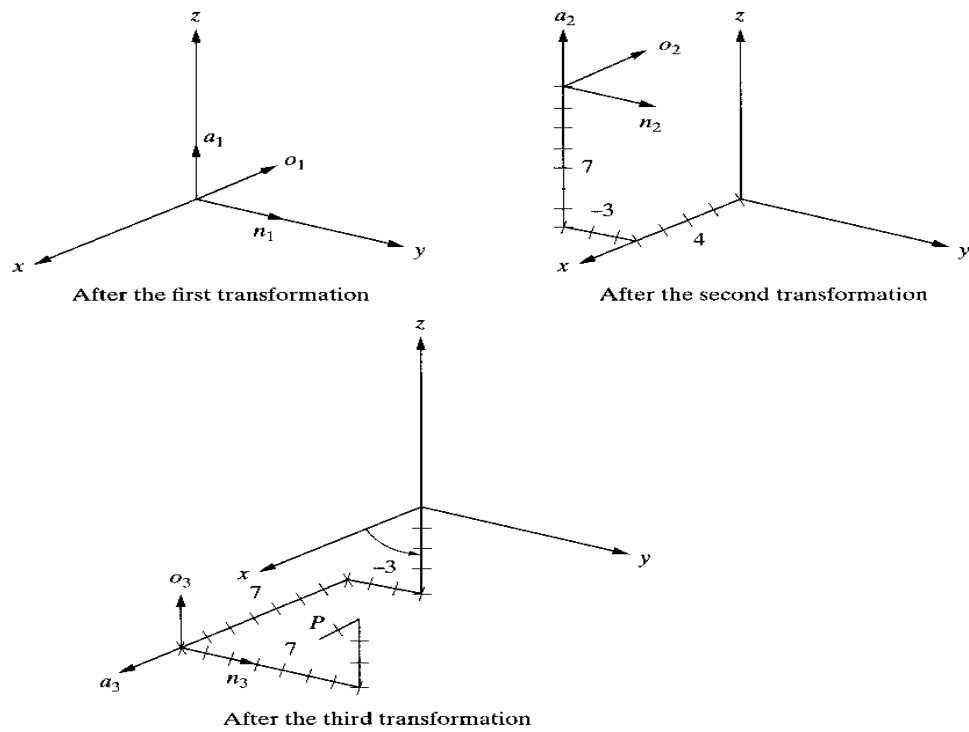


Fig 2.14 Changing the order of transformations will change the final result

Transformations Relative to the Rotating Frame

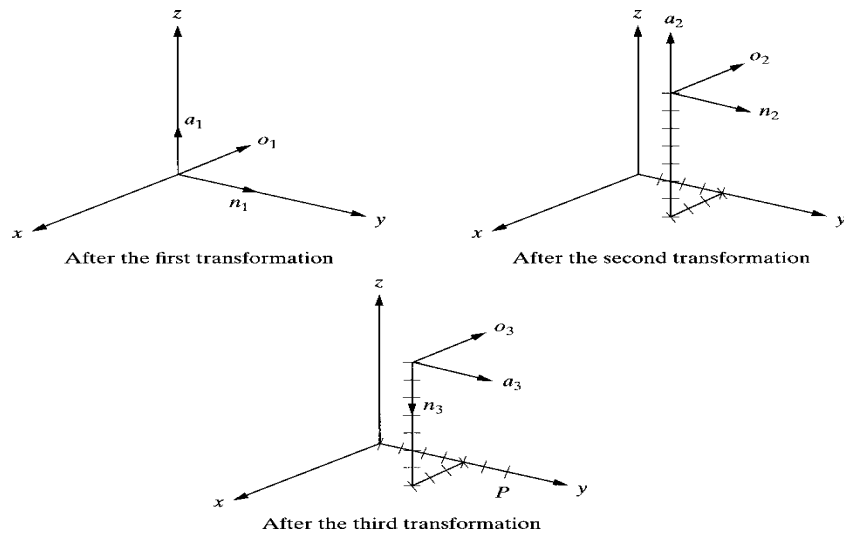


Fig.2.15 Transformations relative to the current frames

KINEMATICS EQUATIONS:

A fundamental tool in robot kinematics is the kinematics equations of the kinematic chains that form the robot. These non-linear equations are used to map the joint parameters to the configuration of the robot system. Kinematics equations are also used in biomechanics of the skeleton and computer animation of articulated characters.

Forward kinematics uses the kinematic equations of a robot to compute the position of the end-effector from specified values for the joint parameters. The reverse process that computes the joint parameters that achieve a specified position of the end-effector is known as inverse kinematics. The dimensions of the robot and its kinematics equations define the volume of space reachable by the robot, known as its workspace.

There are two broad classes of robots and associated kinematics equations serial manipulators and parallel manipulators. Other types of systems with specialized kinematics equations are air, land, and submersible mobile robots, hyper-redundant, or snake, robots and humanoid robots.

DENAVIT-HARTENBERG PARAMETERS:

The Denavit–Hartenberg parameters (also called DH parameters) are the four parameters associated with a particular convention for attaching reference frames to the links of a spatial kinematic chain, or robot manipulator.

Denavit-Hartenberg convention:

A commonly used convention for selecting frames of reference in robotics applications is the **Denavit and Hartenberg (D–H) convention**. In this convention, coordinate frames are attached to the joints between two links such that one transformation is associated with the joint, $[Z]$, and the second is associated with the link $[X]$. The coordinate transformations along a serial robot consisting of n links form the kinematics equations of the robot,

$$[T] = [Z_1][X_1][Z_2][X_2] \dots [X_{n-1}][Z_n],$$

Where, $[T]$ is the transformation locating the end-link.

In order to determine the coordinate transformations $[Z]$ and $[X]$, the joints connecting the links are modeled as either hinged or sliding joints, each of which have a unique line S in space that forms the joint axis and define the relative movement of the two links. A typical serial robot is characterized by a sequence of six lines $S_i, i=1, \dots, 6$, one for each joint in the robot. For each sequence of lines S_i and S_{i+1} , there is a common normal line $A_{i,i+1}$. The system of six joint axes S_i and five common normal lines $A_{i,i+1}$ form the kinematic skeleton of the typical six degree of freedom serial robot. Denavit and Hartenberg introduced the convention that Z coordinate axes are assigned to the joint axes S_i and X coordinate axes are assigned to the common normal's $A_{i,i+1}$.

This convention allows the definition of the movement of links around a common joint axis S_i by the screw displacement,

$$[Z_i] = \begin{bmatrix} \cos \theta_i & -\sin \theta_i & 0 & 0 \\ \sin \theta_i & \cos \theta_i & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

Where θ_i is the rotation around and d_i is the slide along the Z axis---either of the parameters can be constants depending on the structure of the robot. Under this convention the dimensions of each link in the serial chain are defined by the screw displacement around the common normal $A_{i,i+1}$ from the joint S_i to S_{i+1} , which is given by

$$[X_i] = \begin{bmatrix} 1 & 0 & 0 & r_{i,i+1} \\ 0 & \cos \alpha_{i,i+1} & -\sin \alpha_{i,i+1} & 0 \\ 0 & \sin \alpha_{i,i+1} & \cos \alpha_{i,i+1} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

Where $\alpha_{i,i+1}$ and $r_{i,i+1}$ define the physical dimensions of the link in terms of the angle measured around and distance measured along the X axis.

In summary, the reference frames are laid out as follows:

- the z -axis is in the direction of the joint axis
- the x -axis is parallel to the common normal: $x_n = z_{n-1} \times z_n$ If there is no unique common normal (parallel z axes), then d (below) is a free parameter. The direction of x_n is from z_{n-1} to z_n , as shown in the video below.
- the y -axis follows from the x - and z -axis by choosing it to be a right-handed coordinate system.

Four parameters

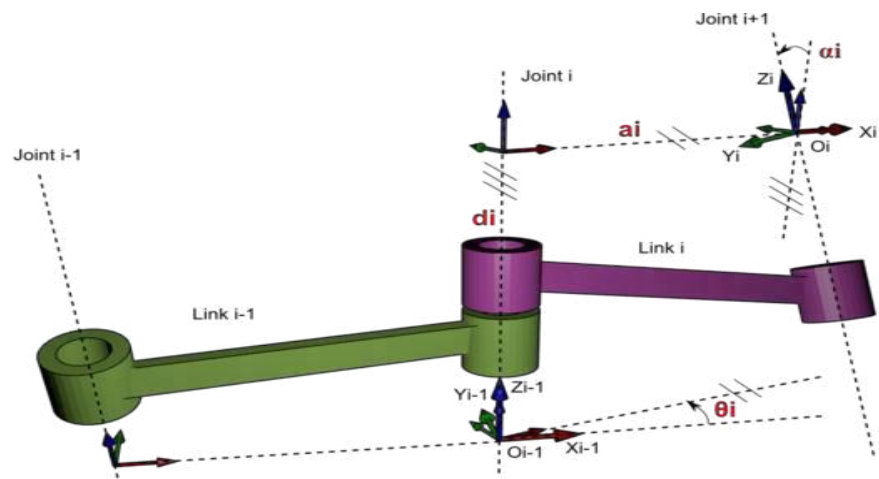


Fig 2.16 Conventional D-H representation of a general-purpose joint-link combination

The four parameters of classic DH convention are $\theta_i, d_i, a_i, \alpha_i$. With those four parameters, we can translate the coordinates from $O_{i-1}X_{i-1}Y_{i-1}Z_{i-1}$ to $O_iX_iY_iZ_i$.

The transformation the following four parameters known as D–H parameters:

d: offset along previous z to the common normal

θ : angle about previous z , from old x to new x

r: length of the common normal. Assuming a revolute joint, this is the radius about previous z .

α : angle about common normal, from old z axis to new z axis

There is some choice in frame layout as to whether the previous x axis or the next x points along the common normal. The latter system allows branching chains more efficiently, as multiple frames can all point away from their common ancestor, but in the alternative layout the ancestor can only point toward one successor. Thus the commonly used notation places each down-chain x axis collinear with the common normal, yielding the transformation calculations shown below.

We can note constraints on the relationships between the axes:

- x_n -axis is perpendicular to both the z_{n-1} and z_n axes
- x_n -axis intersects both z_{n-1} and z_n axes
- Origin of joint n is at the intersection of x_n and z_n
- y_n completes a right-handed reference frame based on x_n and z_n

Denavit-Hartenberg Matrix:

It is common to separate a screw displacement into the product of a pure translation along a line and a pure rotation about the line, ^{[5][6]} so that

$$[Z_i] = \text{Trans}_{Z_i}(d_i) \text{Rot}_{Z_i}(\theta_i),$$

And,

$$[X_i] = \text{Trans}_{X_i}(r_{i,i+1}) \text{Rot}_{X_i}(\alpha_{i,i+1}).$$

Using this notation, each link can be described by a coordinate transformation from the previous coordinate system to the next coordinate system.

$${}^{n-1}T_n = \text{Trans}_{z_{n-1}}(d_n) \cdot \text{Rot}_{z_{n-1}}(\theta_n) \cdot \text{Trans}_{x_n}(r_n) \cdot \text{Rot}_{x_n}(\alpha_n)$$

Note that this is the product of two screw displacements, the matrices associated with these operations are:

$$\text{Trans}_{z_{n-1}}(d_n) = \left[\begin{array}{ccc|c} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_n \\ \hline 0 & 0 & 0 & 1 \end{array} \right]$$

$$\text{Rot}_{z_{n-1}}(\theta_n) = \left[\begin{array}{ccc|c} \cos \theta_n & -\sin \theta_n & 0 & 0 \\ \sin \theta_n & \cos \theta_n & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 0 & 1 \end{array} \right]$$

$$\text{Trans}_{x_n}(r_n) = \left[\begin{array}{ccc|c} 1 & 0 & 0 & r_n \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 0 & 1 \end{array} \right]$$

$$\text{Rot}_{x_n}(\alpha_n) = \left[\begin{array}{ccc|c} 1 & 0 & 0 & 0 \\ 0 & \cos \alpha_n & -\sin \alpha_n & 0 \\ 0 & \sin \alpha_n & \cos \alpha_n & 0 \\ \hline 0 & 0 & 0 & 1 \end{array} \right]$$

This gives:

$${}^{n-1}T_n = \left[\begin{array}{ccc|c} \cos \theta_n & -\sin \theta_n \cos \alpha_n & \sin \theta_n \sin \alpha_n & r_n \cos \theta_n \\ \sin \theta_n & \cos \theta_n \cos \alpha_n & -\cos \theta_n \sin \alpha_n & r_n \sin \theta_n \\ 0 & \sin \alpha_n & \cos \alpha_n & d_n \\ \hline 0 & 0 & 0 & 1 \end{array} \right] = \left[\begin{array}{c|c} R & T \\ \hline 0 & 1 \end{array} \right]$$

Where R is the 3×3 sub matrix describing rotation and T is the 3×1 sub matrix describing translation.

DENAVIT-HARTENBERG REPRESENTATION OF FORWARD KINEMATIC EQUATIONS OF ROBOT:

Denavit-Hartenberg Representation:

1. Simple way of modeling robot links and joints for any robot configuration, regardless of its sequence or complexity.
2. Transformations in any coordinates are possible.

3. Any possible combinations of joints and links and all-revolute articulated robots can be represented

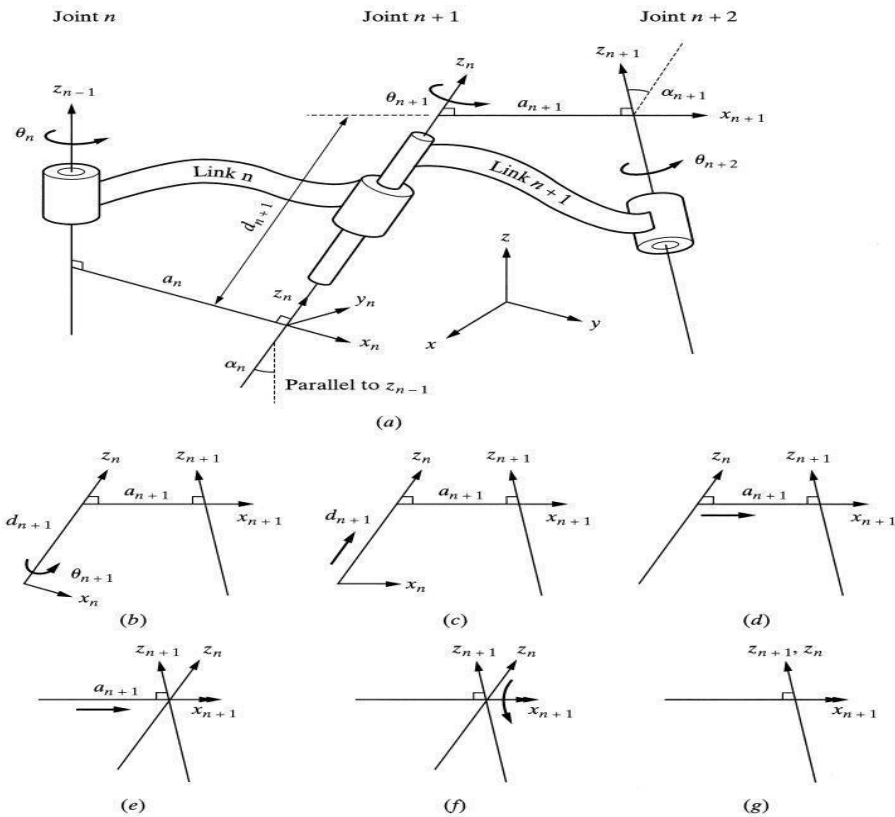


Fig 2.17 D-H representation of a general-purpose joint-link combination

DENAVIT-HARTENBERG REPRESENTATION PROCEDURES:

Start point:

- ◆ Assign joint number n to the first shown joint.
- ◆ Assign a local reference frame for each and every joint before or after these joints.
- ◆ Y-axis does not used in D-H representation.

Procedures for assigning a local reference frame to each joint:

All joints are represented by a z -axis. (Right-hand rule for rotational joint, linear movement for prismatic joint)

- ◆ The common normal is one line mutually perpendicular to any two skew lines.
- ◆ Parallel z -axes joints make a infinite number of common normal.
- ◆ Intersecting z -axes of two successive joints make no common normal between them (Length is 0.).

Symbol Terminologies:

- ◆ θ : A rotation about the z -axis.
- ◆ d : The distance on the z -axis.
- ◆ a : The length of each common normal (Joint offset).
- ◆ α : The angle between two successive z -axes (Joint twist)

Only θ and d are joint variables

The necessary motions to transform from one reference frame to the next.

- I) Rotate about the z_n -axis an angle of θ_{n+1} . (Coplanar)
- II) Translate along z_n -axis a distance of d_{n+1} to make x_n and x_{n+1} colinear.
- III) Translate along the x_n -axis a distance of a_{n+1} to bring the origins of x_{n+1} together.
- IV) Rotate z_n -axis about x_{n+1} axis an angle of α_{n+1} to align z_n -axis with z_{n+1} -axis.

Determine the value of each joint to place the arm at a desired position and orientation.

INVERSE KINEMATIC PROGRAM OF ROBOTS:

A robot has a predictable path on a straight line, or an unpredictable path on a straight line.

- ♦ A predictable path is necessary to recalculate joint variables. (Between 50 to 200 times a second)
- ♦ To make the robot follow a straight line, it is necessary to break the line into many small sections.
- ♦ All unnecessary computations should be eliminated.

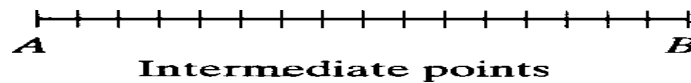


Fig. 2.18 Small sections of movement for straight-line motions

DEGENERACY AND DEXTERITY:

Degeneracy: The robot loses a degree of freedom and thus cannot perform as desired.

- ♦ When the robot's joints reach their physical limits, and as a result, cannot move any further.
- ♦ In the middle point of its workspace if the z -axes of two similar joints becomes co-linear.

Dexterity: The volume of points where one can position the robot as desired, but not orientate it.

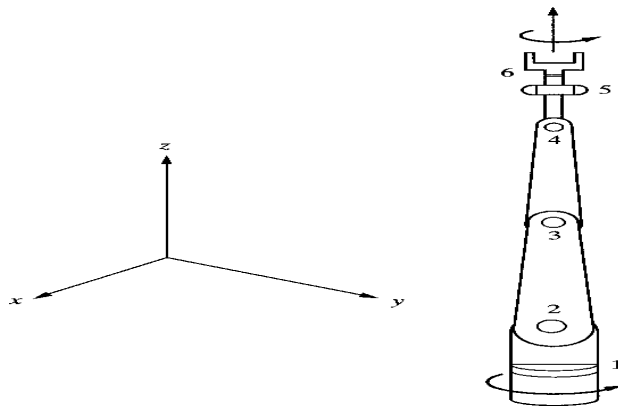


Fig. 2.19 An example of a robot in a degenerate position

THE FUNDAMENTAL PROBLEM WITH D-H REPRESENTATION:

Defect of D-H presentation: D-H cannot represent any motion about the y -axis, because all motions are about the x - and z -axis.

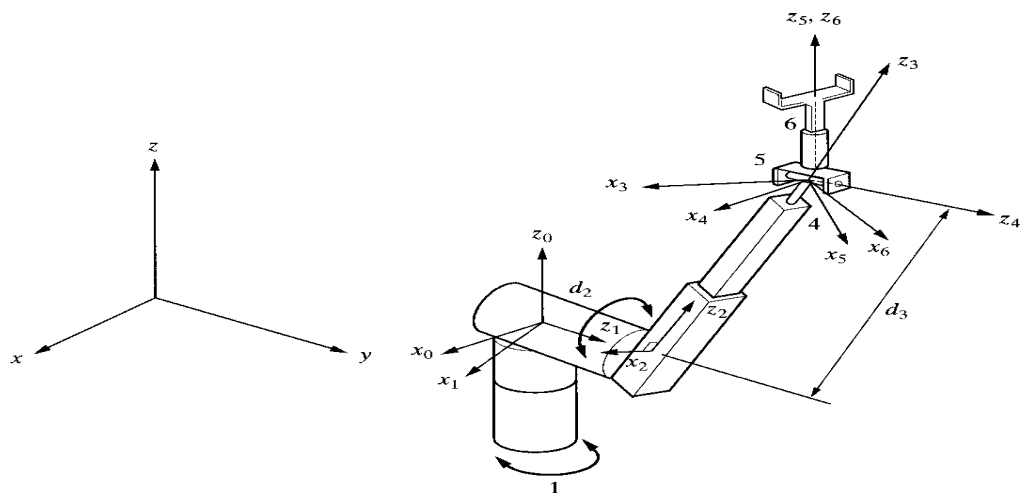


Fig. 2.20 Frames of the Stanford Arm.

#	\square	d	a	\square
1	$\square 1$	0	0	-90
2	$\square 2$	d_1	0	90
3	0	d_1	0	0
4	$\square 4$	0	0	-90
5	$\square 5$	0	0	90
6	$\square 6$	0	0	0

Table 2.1 Parameters Table for the Stanford Arm

INVERSE OF TRANSFORMATION MATIRICES

Inverse of a matrix calculation steps:

- ◆ Calculate the determinant of the matrix.
- ◆ Transpose the matrix.
- ◆ Replace each element of the transposed matrix by its own minor (ad-joint matrix).
- ◆ Divide the converted matrix by the determinant.

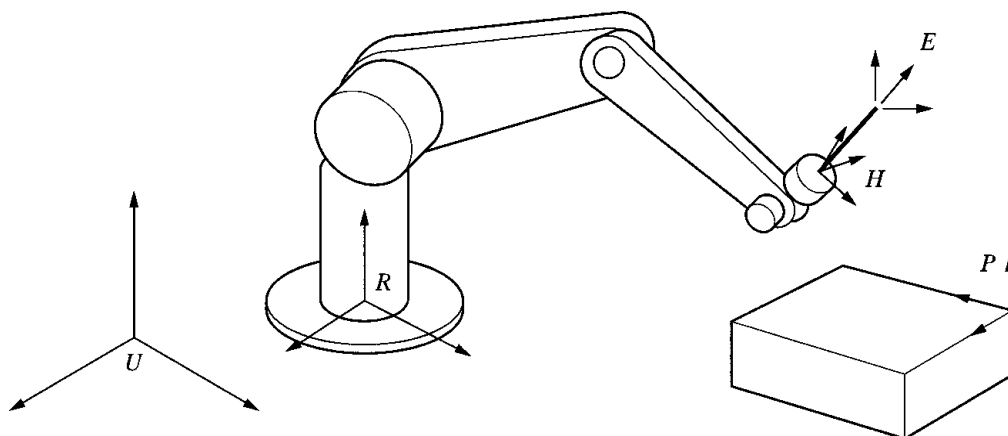


Fig 2.21 The Universe, robot, hand, part, and end effector frames

FORWARD AND INVERSE KINEMATICS OF ROBOTS:

Forward Kinematics Analysis:

- ◆ Calculating the position and orientation of the hand of the robot.
- ◆ If all robot joint variables are known, one can calculate where the robot is at any instant.

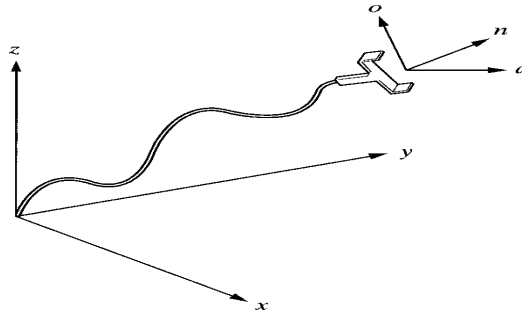


Fig.2.22 Handframe of the robot relative to the reference

frame Forward Kinematics and Inverse Kinematics equation for position analysis:

- Cartesian (gantry, rectangular) coordinates.
- Cylindrical coordinates.
- Spherical coordinates.
- Articulated (anthropomorphic, or all-revolute)

coordinates Forward and Inverse Kinematics Equations for
Position

(a) Cartesian (Gantry, Rectangular) Coordinates: IBM 7565 robot

- All actuator is linear
- A gantry robot is a Cartesian robot

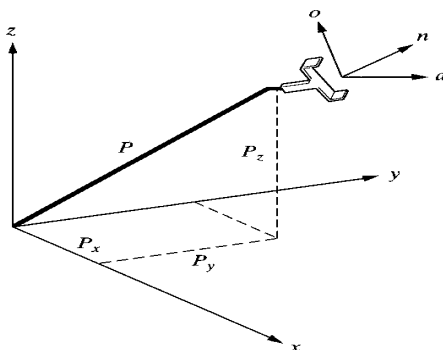


Fig. 2.23 Cartesian Coordinates

(b) Cylindrical Coordinates: 2 Linear translations and 1 rotation

- ◆ translation of r along the x -axis
- ◆ rotation of \square about the z -axis
- ◆ translation of l along the z -axis

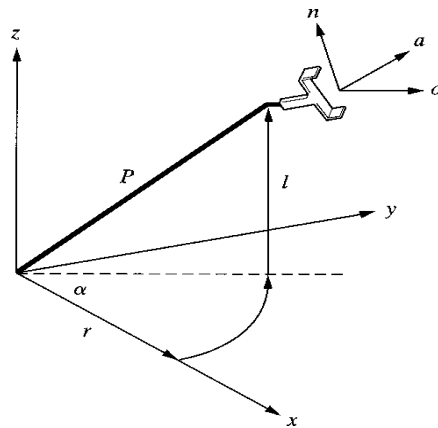


Fig. 2.24 Cylindrical Coordinates

(c) Spherical Coordinates: 1 linear translation and 2 rotations

- ◆ translation of r along the z -axis
- ◆ rotation of \square about the y -axis
- ◆ rotation of \square along the z -axis

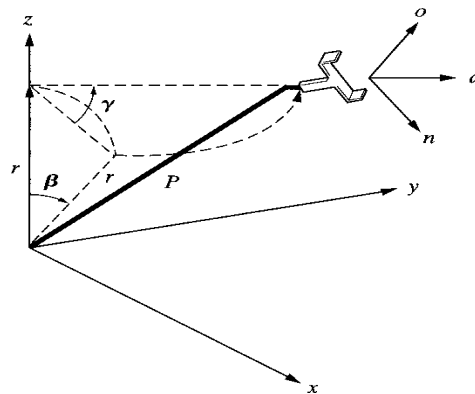


Fig.2.25 Spherical Coordinates

(d) Articulated Coordinates: 3 rotations -> Denavit-Hartenberg representation

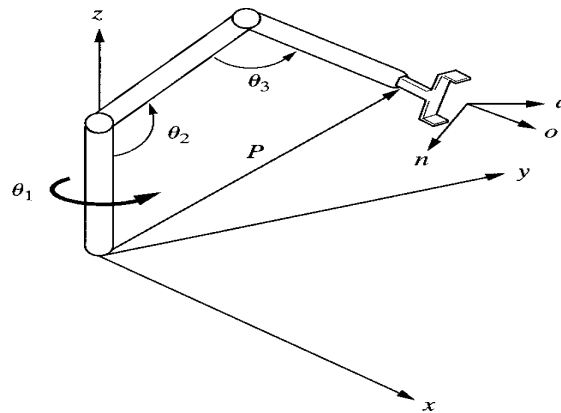


Fig. 2.26 Articulated Coordinates.

Forward and Inverse Kinematics Equations for Orientation

- ♦ Roll, Pitch, Yaw (RPY) angles
- ♦ Euler angles
- ♦ Articulated joints

(a) Roll, Pitch, Yaw (RPY) Angles

- Roll: Rotation of about - axis (z -axis of the moving frame)
- Pitch: Rotation of about - axis (y -axis of the moving frame)
- Yaw: Rotation of about - axis (x -axis of the moving frame)

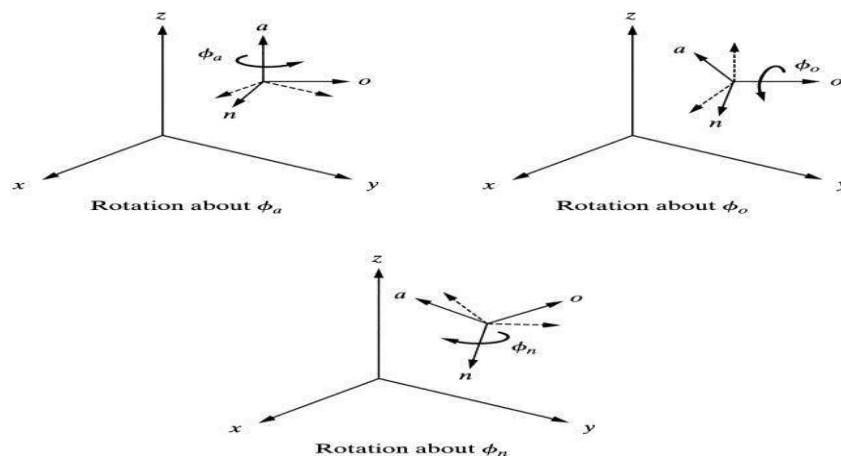


Fig. 2.27 RPY rotations about the current axes

(b) Euler Angles

- Rotation of about - axis (z-axis of the moving frame) followed by
- Rotation of about -axis (y-axis of the moving frame) followed by
- Rotation of about -axis (z-axis of the moving frame)

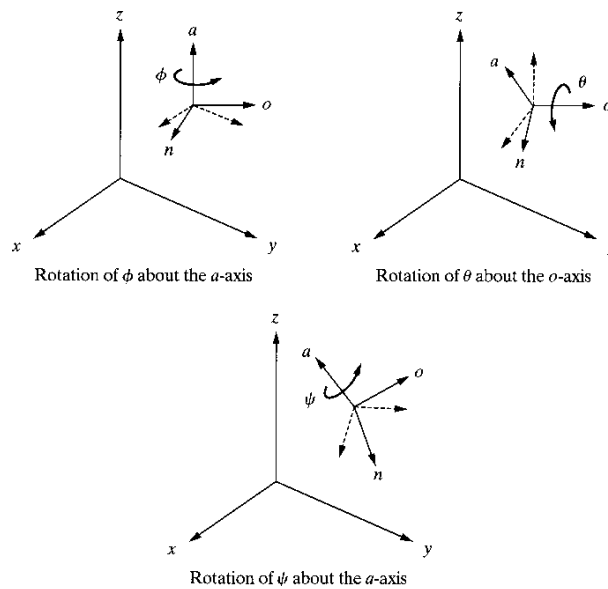


Fig. 2.28 Euler rotations about the current axes

Forward and Inverse Kinematics Equations for Orientation:

Assumption : Robot is made of a Cartesian and an RPY set of joints.

$${}^R T_H = T_{cart}(x, y, z) \cdot RPY(a, o, n)$$

Assumption : Robot is made of a Spherical Coordinate and an Euler angle.

$${}^R T_H = T(r, \theta, \phi) \cdot Euler(\alpha, \beta, \gamma)$$

Newton-Euler formulation, and the Lagrangian formulation

The dynamic behavior of robot mechanisms is described in terms of the time rate of change of the robot configuration in relation to the joint torques exerted by the actuators. This relationship can be expressed by a set of differential equations, called equations of motion, that govern the dynamic response of the robot linkage to input joint torques. In the next chapter, we will design a control system on the basis of these equations of motion.

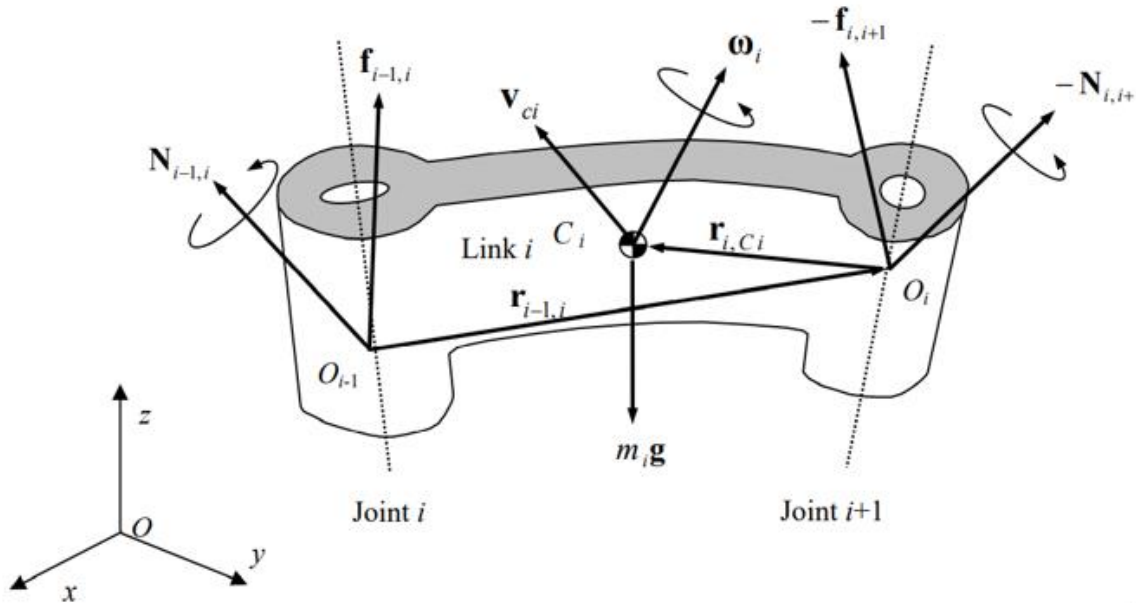
Two methods can be used in order to obtain the equations of motion: the *Newton-Euler formulation*, and the *Lagrangian formulation*. The Newton-Euler formulation is derived by the direct interpretation of Newton's Second Law of Motion, which describes dynamic systems in terms of force and momentum. The equations incorporate all the forces and moments acting on the individual robot links, including the coupling forces and moments between the links. The equations obtained from the Newton-Euler method include the constraint forces acting between adjacent links. Thus, additional arithmetic operations are required to eliminate these terms and obtain explicit relations between the joint torques and the resultant motion in terms of joint displacements. In the Lagrangian formulation, on the other hand, the system's dynamic behavior is described in terms of work and energy using generalized coordinates. This approach is the extension of the indirect method discussed in the previous chapter to dynamics. Therefore, all the workless forces and constraint forces are automatically eliminated in this method. The resultant equations are generally compact and provide a closed-form expression in terms of joint torques and joint displacements. Furthermore, the derivation is simpler and more systematic than in the Newton-Euler method.

The robot's equations of motion are basically a description of the relationship between the input joint torques and the output motion, i.e. the motion of the robot linkage. As in kinematics and in statics, we need to solve the inverse problem of finding the necessary input torques to obtain a desired output motion. This *inverse dynamics* problem is discussed in the last section of this chapter. Efficient algorithms have been developed that allow the dynamic computations to be carried out on-line in real time.

Newton-Euler Formulation of Equations of Motion

Basic Dynamic Equations: In this section we derive the equations of motion for an individual link based on the direct method, i.e. Newton-Euler Formulation. The motion of a rigid body can be decomposed into the translational motion with respect to an arbitrary point fixed to the rigid body, and the rotational motion of the rigid body about that point. The dynamic equations of a rigid body can also be represented by two equations: one describes the translational motion of the centroid (or center of mass), while the other describes the rotational motion about the centroid. The former is Newton's equation of motion for a mass particle, and the latter is called Euler's equation of motion. We begin by considering the free body diagram of an individual link. Figure 2.29 shows all the forces and moments acting on link i . The figure, which describes the static balance of forces, except for the inertial force and moment that arise from the dynamic motion of the link. Let \mathbf{v}_{ci} be the linear velocity of the centroid of link i with reference to the base coordinate frame O -xyz, which is an inertial reference frame. The inertial force is then given by $-m_i \dot{\mathbf{v}}_{ci}$, where m_i is the mass of the link and $\dot{\mathbf{v}}_{ci}$ is the time derivative of \mathbf{v}_{ci} . Based on D'Alembert's principle, the equation of motion is then obtained by adding the inertial force to the static balance of forces

$$\mathbf{f}_{i-1,i} - \mathbf{f}_{i,i+1} + m_i \mathbf{g} - m_i \dot{\mathbf{v}}_{ci} = \mathbf{0}, \quad i = 1, \dots, n$$



are the coupling forces applied to link i by links $i-1$ and $i+1$, respectively, and g is the acceleration of gravity.

Rotational motions are described by Euler's equations. In the same way as for translational motions, adding “inertial torques” to the static balance of moments yields the dynamic equations. We begin by describing the mass properties of a single rigid body with respect to rotations about the centroid. The mass properties are represented by an inertia tensor, or an inertia matrix, which is a 3×3 symmetric matrix defined by

$$\mathbf{I} = \begin{pmatrix} \int_{body} \{(y-y_c)^2 + (z-z_c)^2\} \rho dV & -\int_{body} (x-x_c)(y-y_c) \rho dV & -\int_{body} (z-z_c)(x-x_c) \rho dV \\ -\int_{body} (x-x_c)(y-y_c) \rho dV & \int_{body} \{(z-z_c)^2 + (x-x_c)^2\} \rho dV & -\int_{body} (y-y_c)(z-z_c) \rho dV \\ -\int_{body} (z-z_c)(x-x_c) \rho dV & -\int_{body} (y-y_c)(z-z_c) \rho dV & \int_{body} \{(x-x_c)^2 + (y-y_c)^2\} \rho dV \end{pmatrix}$$

Lagrangian Formulation of Robot Dynamics

Lagrangian Dynamics

In the Newton-Euler formulation, the equations of motion are derived from Newton's Second Law, which relates force and momentum, as well as torque and angular momentum. The resulting equations involve constraint forces, which must be eliminated in order to obtain closedform dynamic equations. In the Newton-Euler formulation, the equations are not expressed in terms of independent variables, and do not include input joint torques explicitly. Arithmetic operations are needed to derive the closed-form dynamic equations. This represents a complex procedure that requires physical intuition, as discussed in the previous section. An alternative to the Newton-Euler formulation of manipulator dynamics is the Lagrangian formulation, which describes the behavior of a dynamic system in terms of work and energy stored in the system rather than of forces and moments of the individual members involved. The constraint forces involved in the system are automatically eliminated in the formulation of Lagrangian dynamic equations. The closed-form dynamic equations can be derived systematically in any coordinate system. Let q_i be generalized coordinates that completely locate a dynamic system. Let T and U be the total kinetic energy and potential energy stored in the dynamic system. We define the Lagrangian L by

$$L(q_i, \dot{q}_i) = T(q_i, \dot{q}_i) - U(q_i)$$

Note that the potential energy is a function of generalized coordinates q_i and that the kinetic energy is that of generalized velocities as well as generalized coordinates q_i . Using the Lagrangian, equations of motion of the dynamic system are given by

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} = Q_i, \quad i = 1, \dots, n$$

where Q_i is the generalized force corresponding to the generalized coordinate q_i . Considering the virtual work done by non-conservative forces can identify the generalized forces acting on the system.

TEXT/REFERENCE BOOKS:

1. Saha, S.K., "Introduction to Robotics, 2nd Edition, McGraw-Hill Higher Education, New Delhi, 2014.
2. Ashitava Ghosal., "Fundamental Concepts and Analysis", Oxford, New Delhi, 2006.
3. Mikell.P.Groover, Mitchell Weiss, Roger.N.Nagel, Nicholas.G.Odrey, "Industrial Robotics-Technology, Programming and Applications", Tata McGraw-Hill Publishing Company Limited, New Delhi, Third Reprint 2008



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 ELECTRICAL AND ELECTRONICS ENGINEERING
DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING**

UNIT III- SENSORS AND VISION SYSTEM-SCSA1406

UNIT III

SENSORS AND VISION SYSTEMS

Unit 3: Sensors and Vision System

Sensor: Contact and Proximity, Position, Velocity, Force, Tactile etc. Introduction to Cameras, Camera calibration, Geometry of Image formation, Euclidean/Similarity/Affine/Projective transformations
Vision applications in robotics

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

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

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

Transducer

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

take form of a sensor or an Actuator Sensor (e.g., thermometer)

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

Tactile sensing

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

Touch Sensing

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

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

Force/torque sensors

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

Proximity sensor

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

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

Proximity Sensors

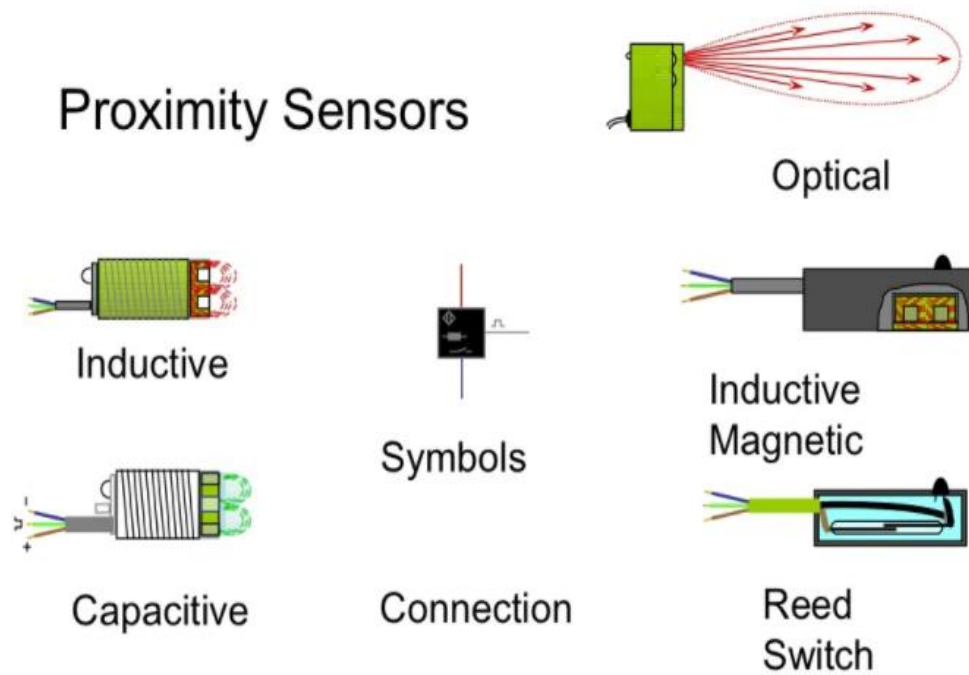


Fig.3.1 Types of Proximity Sensors

ELECTRONIC SENSORS : CAPACITIVE PROXIMITY SENSOR

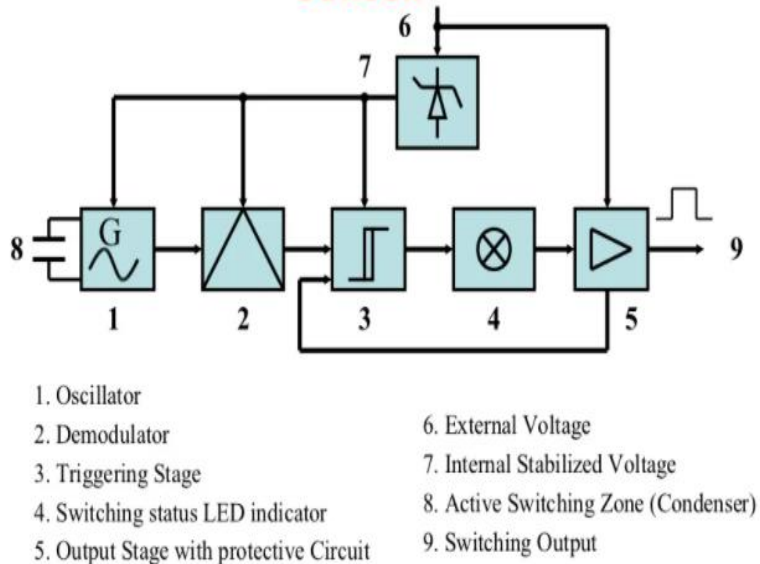


Fig.3.2 Capacitive Proximity Sensor

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

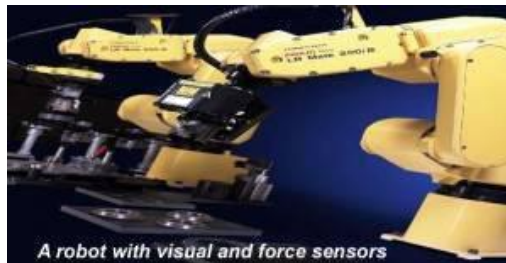


Fig 3.3 Industrial Robot with Sensor

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 3.4 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.

Camera Calibration:

Camera calibration is a necessary step in 3D computer vision. • A calibrated camera can be used as a quantitative sensor • It is essential in many applications to recover 3D quantitative measures about the observed scene from 2D images. Such as 3D Euclidean structure • From a calibrated camera we can measure how far an object is from the camera, or the height of the object, etc. e.g., object avoidance in robot navigation

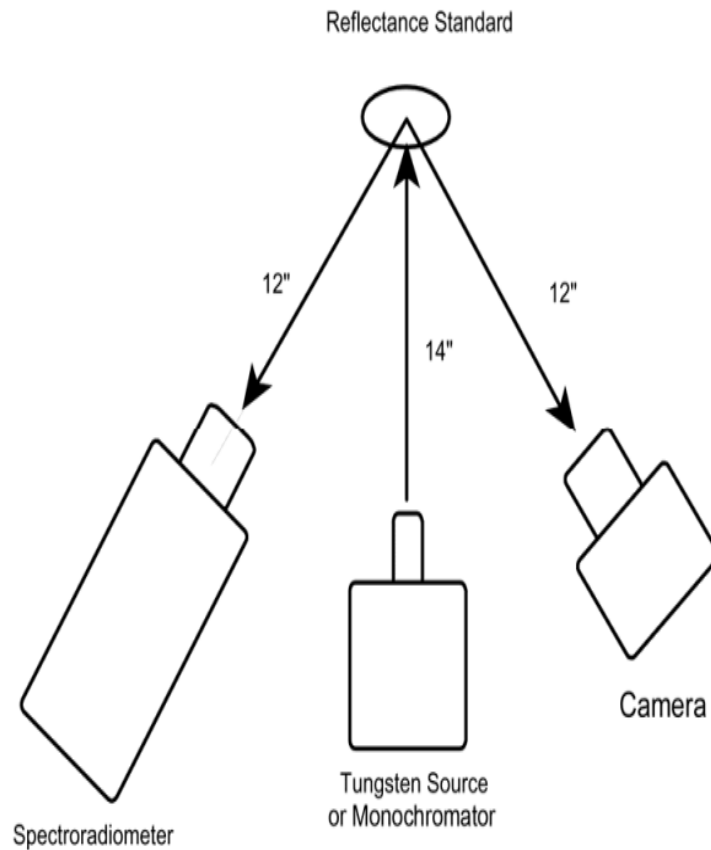


Fig 3.5 Camera Calibration

Suppose that the camera *F-number* is set to f , and that the shutter duration was set to T (in seconds). The effective aperture area when $f = 1.0$ is

$$a_0 = \frac{K}{1.0^2}$$

where K is a lens-dependent constant. The aperture area for an arbitrary *F-number* is

$$a = \frac{K}{f^2}$$

The relative area A is given by

$$A = \left(\frac{K}{f^2} \right) / \left(\frac{K}{1.0^2} \right) = \frac{1}{f^2} \quad (1)$$

Camera Models and Calibration:

Cameras provide a crucial sensing modality in the context of robotics. This is generally due to the fact that images inherently contain an enormous amount of information about the environment. However, while images do contain a lot of information, extracting the information that is relevant to the robot is quite challenging. One of the most basic tasks related to image processing is determining how a particular point in the scene maps to a point in the camera image, which is sometimes referred to as perspective projection. Last chapter, the pinhole camera model and the thin lens model were presented, and in this chapter the pinhole camera model is leveraged to further explore perspective projection⁴. All results also hold under the thin lens model, assuming the camera is focused at ∞ .

8.1 Perspective Projection

The pinhole camera model, shown graphically in Figure 8.1, can be used to mathematically define relationships between points P in the scene and points p on the image plane. Notice that any point P in the scene can be represented in two ways: in camera frame coordinates (denoted as PC) or in world frame coordinates (denoted as PW). The overall objective of this section is to find derive a mathematical model that can be used to map a point PW expressed in world frame coordinates to a point p on the image plane. To accomplish these two transformations are combined together, namely a transformation of P from world frame coordinates to camera frame coordinates (PW to PC) and a transformation from camera coordinates to image coordinates (PC to p).

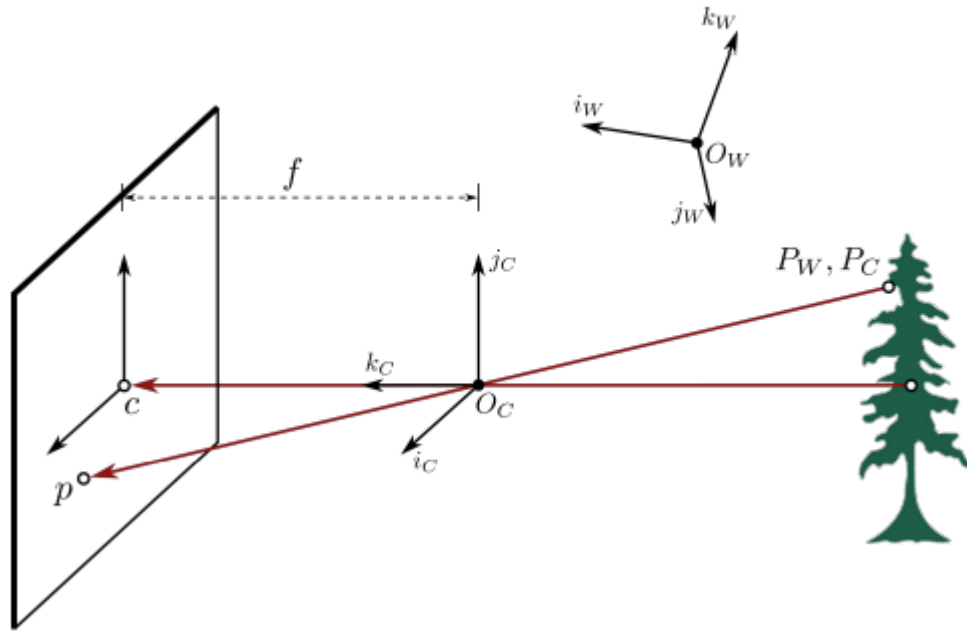


Fig.3.6 Graphical representation of the pinhole camera model

Mapping World Coordinates to Camera Coordinates ($P_W \rightarrow P_C$)

Recalling from Figure 3.6 that a point P in the scene can either be expressed in terms of camera frame coordinates P_C or world frame coordinates P_W . While the previous section discussed the use of the pinhole model to map P_C coordinates to pixel coordinates p , this section will discuss the mapping between the camera and world frame coordinates of the point P as in Figure 3.7 and it can be seen that P_C can be written as: $P_C = t + q$

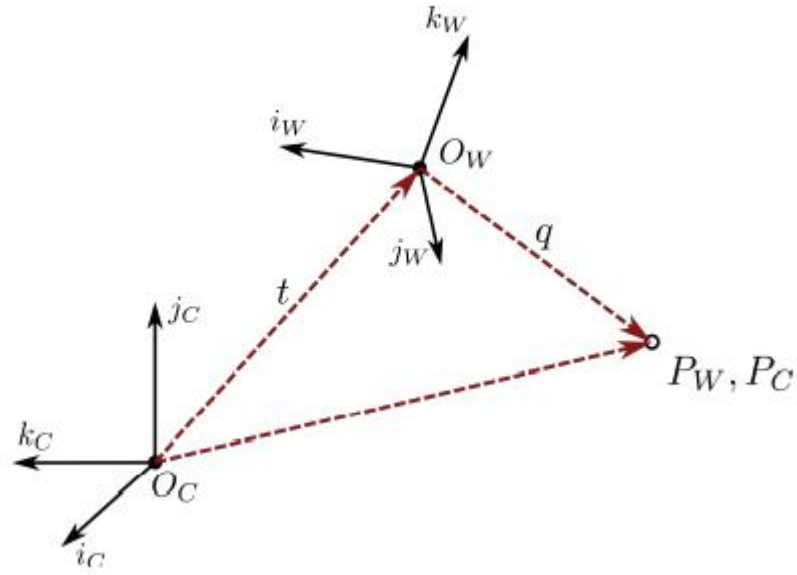


Fig.3.7 Graphical representation of the pinhole camera model

where t is the vector from O_C to O_W expressed in camera frame coordinates and q is the vector from O_W to P expressed in camera frame coordinates. However, the vector q is in fact the same vector as P_W , just expressed in different coordinates (i.e. with respect to a different frame). The coordinates can be related by a rotation:

$$q = RP_W,$$

where R is the rotation matrix relating the camera frame to world frame and is defined as:

$$R = \begin{bmatrix} i_w \cdot i & j_w \cdot i & k_w \cdot i \\ i_w \cdot j & j_w \cdot j & k_w \cdot j \\ i_w \cdot k & j_w \cdot k & k_w \cdot k \end{bmatrix},$$

where i, j , and k are the unit vectors that define the camera frame and i_w, j_w , and k_w are the unit vectors that define the world frame. To summarize, the point P_W can be mapped to camera frame coordinates P_C as:

$$P_C = t + RP_W,$$

where t is the vector in camera frame coordinates from O_C to O_W and R is the rotation matrix. Similar to the previous section, these expressions can also be equivalently expressed for the case where the points P_W and P_C are expressed in homogeneous coordinates:

$$\begin{pmatrix} P_C \\ 1 \end{pmatrix} = \begin{bmatrix} R & t \\ 0_{1 \times 3} & 1 \end{bmatrix} \begin{pmatrix} P_W \\ 1 \end{pmatrix}.$$

Geometry of Image Formation

- The two parts of the image formation process

The geometry of image formation which determines where in the image plane the projection of a point in the scene will be located.

The physics of light which determines the brightness of a point in the image plane as a function of illumination and surface properties.

- A simple model
 - The scene is illuminated by a single source.
 - The scene reflects radiation towards the camera.
 - The camera senses it via chemicals on film.

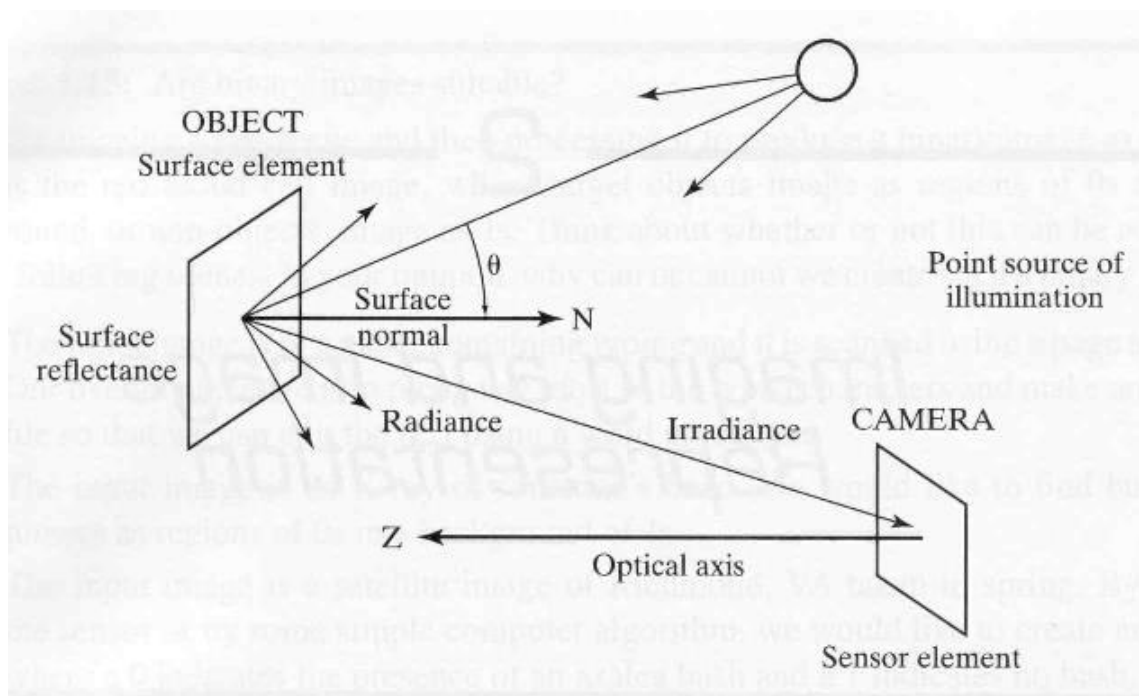


Fig. 3.8 Simple model

- **Camera Geometry**

- The simplest device to form an image of a 3D scene on a 2D surface is the "pinhole" camera.
- Rays of light pass through a "pinhole" and form an inverted image of the object on the image plane.

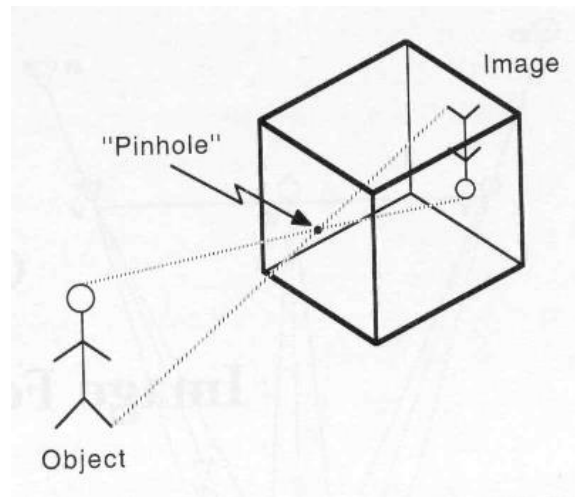


Fig. 3.9. Camera Geometry

Camera Optics

- In practice, the aperture must be larger to admit more light.
- Lenses are placed in the aperture to focus the bundle of rays from each scene point onto the corresponding point in the image plane.

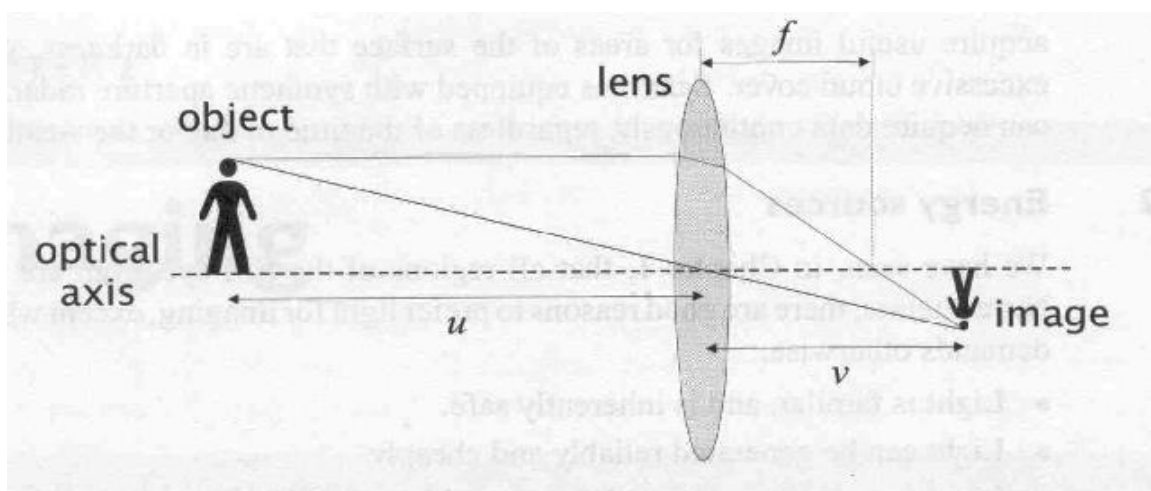


Fig. 3.10. Camera Optics

- **Diffraction and Pinhole Optics**

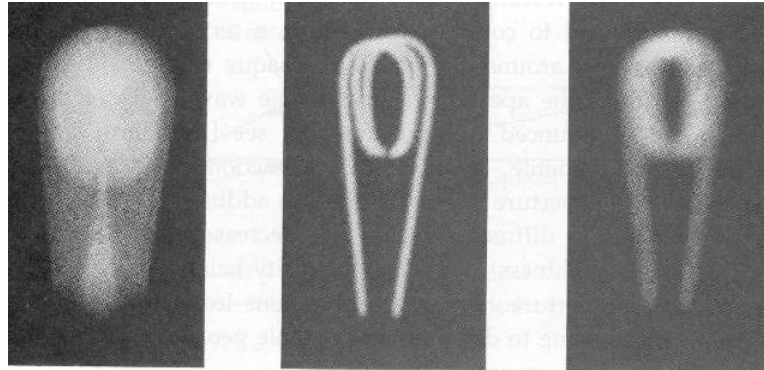


Fig. 3.11. Different pinhole positions

- If we use a wide pinhole, light from the source spreads across the image (i.e., not properly focused), making it blurry.

- If we narrow the pinhole, only a small amount of light is let in.

- * the image sharpness is limited by diffraction.

- * when light passes through a small aperture, it does not travel in a straight line.

- * it is scattered in many direction (this is a quantum effect).

- In general, the aim of using lens is to duplicate the pinhole geometry without resorting to undesirable small apertures.

- **Human Vision**

- At high light levels, pupil (aperture) is small and blurring is due to diffraction.

- At low light levels, pupil is open and blurring is due to lens imperfections.

- **CCD Cameras**

- An array of tiny solid state cells convert light energy into electrical charge.
- Manufactured on chips typically measuring about 1cm x 1cm (for a 512x512 array, each element has a real width of roughly 0.001 cm).
- The output of a CCD array is a continuous electric signal (*video signal*) which is generated by scanning the photo-sensors in a given order (e.g., line by line) and read- ing out their voltages.

Fig. 3.9. Camera Geometry

Fig. 3.9. Camera Geometry

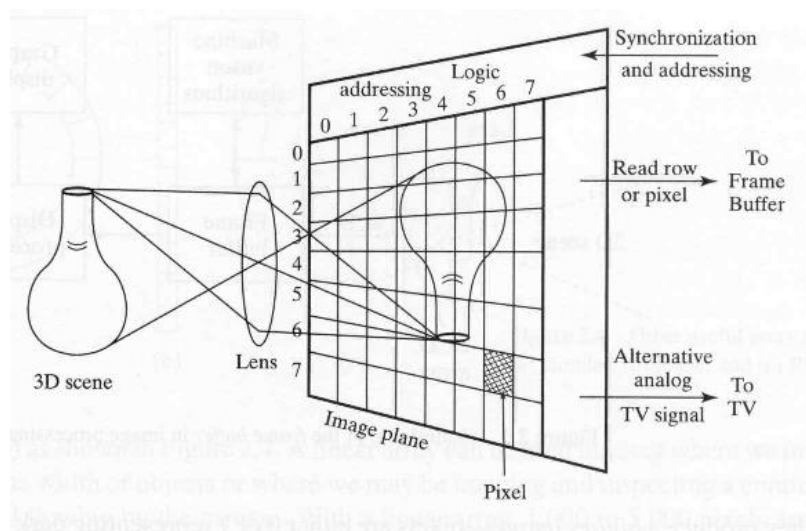


Fig. 3.12. CCD Camera

- **Frame grabber**

- The video signal is sent to an electronic device called the frame grabber.
- The frame grabber digitizes the signal into a 2D, rectangular array $N \times M$ of integer values, stored in the *frame buffer*

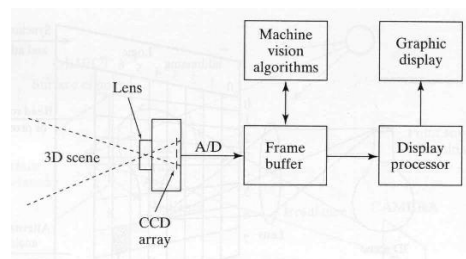


Fig. 3.12. CCD Camera

CCD array and frame buffer

- In a CCD camera, the *physical image plane* is the CCD array of $n \times m$ rectangular grid of photo-sensors.
- The *pixel image plane* (frame buffer) is an array of $N \times M$ integer values (pixels).
- The position of the same point on the image plane will be different if measured in CCD elements (x, y) or image pixels (x_{im}, y_{im}) .
- In general, $n \neq N$ and $m \neq M$; assuming that the origin in both cases is the upperleft corner we have:

$$x_{im} = \frac{N}{n} x \quad y_{im} = \frac{M}{m} y$$

where (x_{im}, y_{im}) are the coordinates of the point in the pixel plane and (x, y) are the coordinates of the point in the CCD plane.

- In general, it is convenient to assume that the CCD elements are always in one-to-one correspondence with the image pixels.
- Units in each case:

(x_{im}, y_{im}) is measured in pixels

(x, y) is measured, e.g., in millimeters.

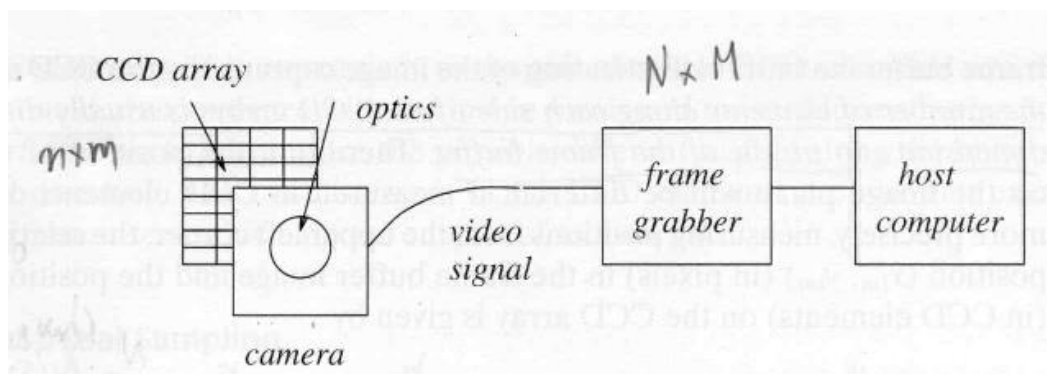


Fig. 3.13. CCD Camera with frame grabber

Reference Frames

- Five reference frames are needed for general problems in 3D scene analysis.

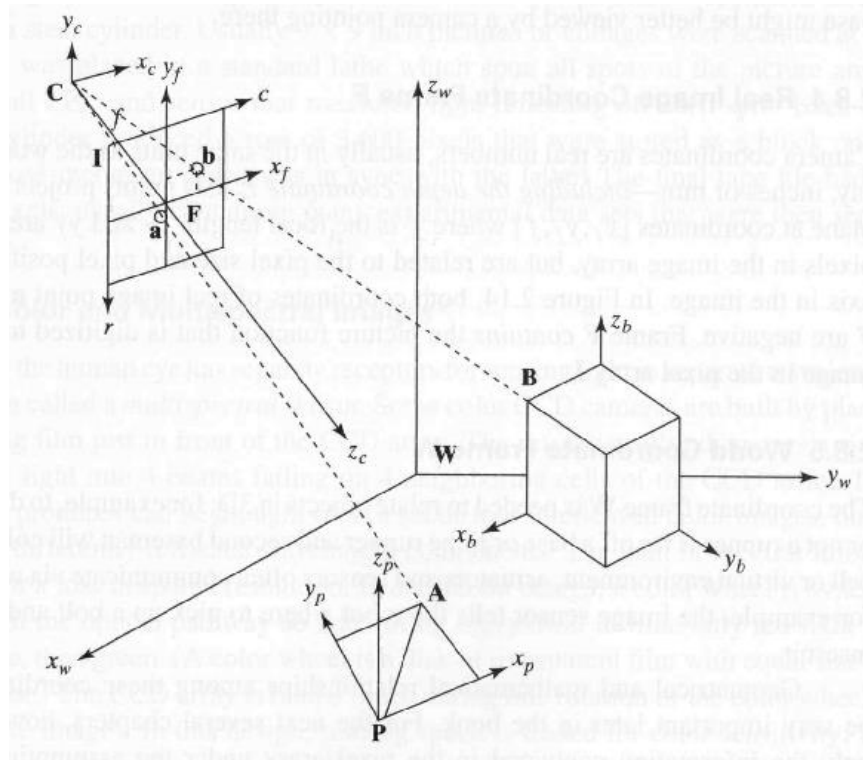


Fig 3.14 Reference Frames

Object Coordinate Frame

- This is a 3D coordinate system: x_b, y_b, z_b
- It is used to model ideal objects in both computer graphics and computer vision.
- It is needed to inspect an object (e.g., to check if a particular hole is in proper position relative to other holes)
- The coordinates of 3D point B, e.g., relative to the object reference frame are $(x_b, 0, z_b)$
- Object coordinates do not change regardless how the object is placed in the scene.

Notation: $(X_o, Y_o, Z_o)^T$

World Coordinate Frame

- This is a 3D coordinate system: x_w, y_w, z_w
- The scene consists of object models that have been placed (rotated and translated) into the scene, yielding object coordinates in the world coordinate system.
- It is needed to relate objects in 3D (e.g., the image sensor tells the robot where to pick up a bolt and in which hole to insert it).

Notation: $(X_w, Y_w, Z_w)^T$

Camera Coordinate Frame

- This is a 3D coordinate system (x_c, y_c, z_c axes)
- Its purpose is to represent objects with respect to the location of the camera.

Notation: $(X_c, Y_c, Z_c)^T$

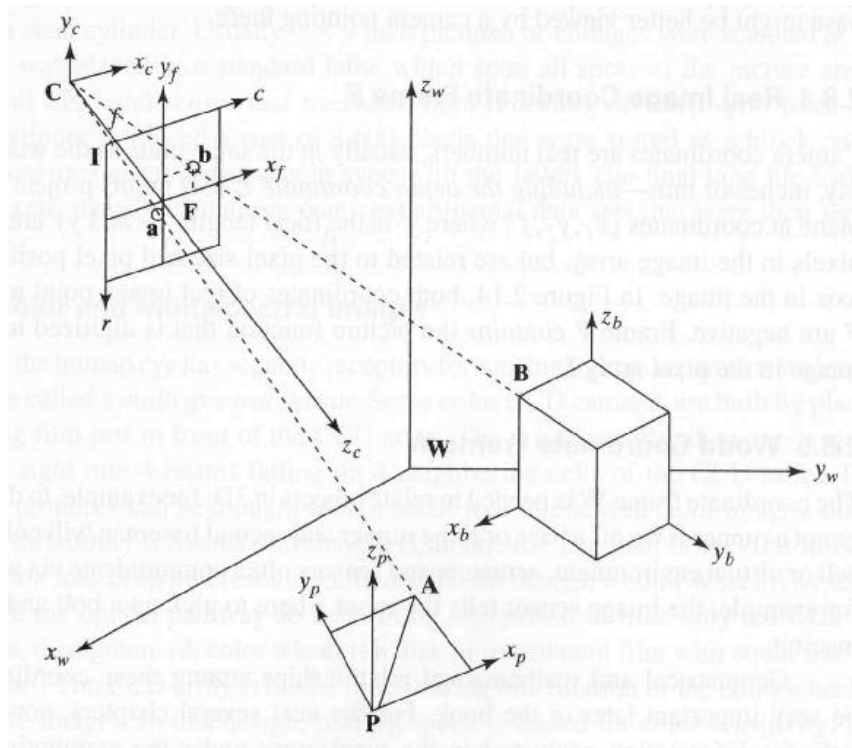


Fig 3.14 Camera Coordinat

- **Image Plane Coordinate Frame (CCD plane)**

- This is a 2D coordinate system (x_f, y_f axes)
- Describes the coordinates of 3D points projected on the image plane.
- The projection of A , e.g., is point a whose both coordinates are negative.

Notation: $(x, y)^T$

Pixel Coordinate Frame

- This is a 2D coordinate system (r, c axes)
- Each pixel in this frame has an integer pixel coordinates.
- Point A , e.g., gets projected to image point (a_r, a_c) where a_r and a_c are integer row and column.

Notation: $(x_{im}, y_{im})^T$

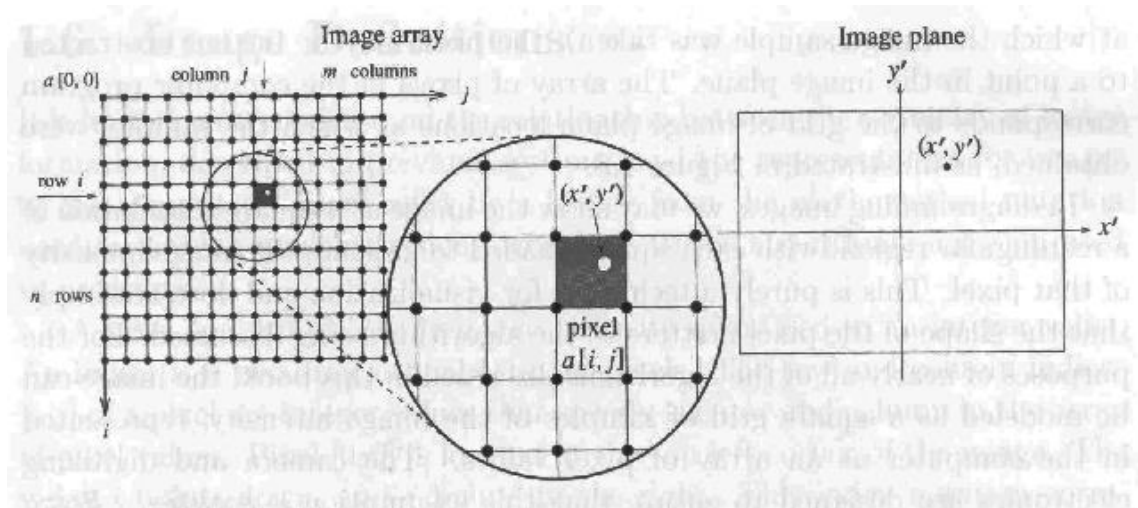
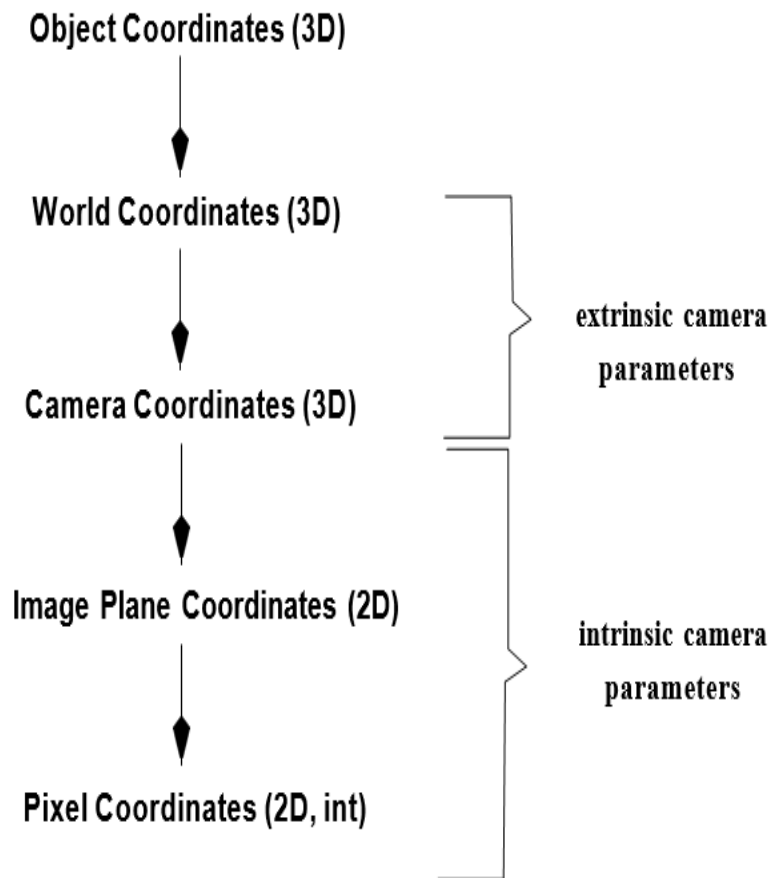


Fig 3.14 Pixel Coordinate Frames

Transformations between frames



Machine Vision System

n:

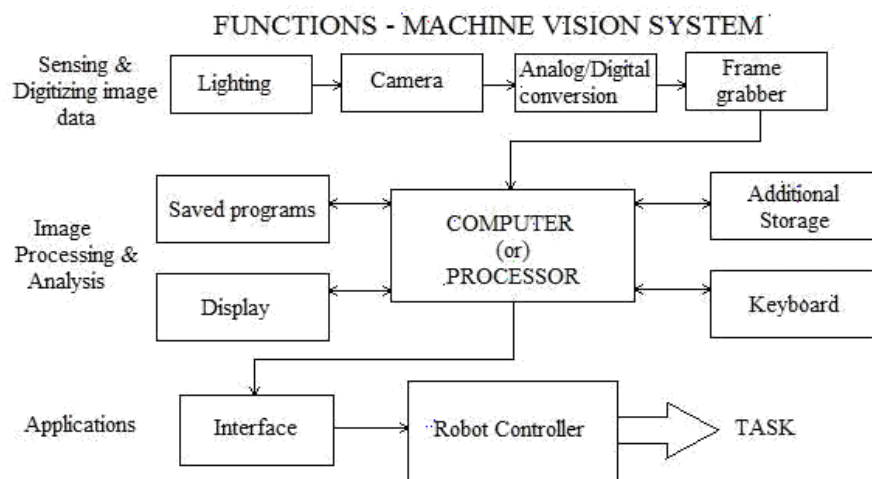


Fig 3.5 Block Diagram of Functions of Machine Vision System

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

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

Sensing & Digitizing Image Data:

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

Image Processing & Analysis:

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

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

Applications:

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

- Inspection
- Orientation
- Part Identification
- Location

Signal conversion

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

Image Processing

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

Another object with other colors accompanied by different sizes. A robotic vision system has to make the distinction between objects and in almost all cases has to 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.

SwisTrack

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

visual navigation

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

Edge Detector

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

TEXT/REFERENCE BOOKS:

- 1.Saha, S.K., “Introduction to Robotics, 2nd Edition, McGraw-Hill Higher Education, New Delhi, 2014.
2. [Ashitava Ghosal](#)., “Fundamental Concepts and Analysis”, Oxford, New Delhi, 2006.
- 3.Mikell.P.Groover, Mitchell Weiss, Roger.N.Nagel, Nicholas.G.Odrey, “Industrial Robotics-Technology, Programming and Applications”, Tata McGraw-Hill Publishing Company Limited, New Delhi, Third Reprint 2008



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 ELECTRICAL AND ELECTRONICS ENGINEERING
DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING**

UNIT IV – ROBOT CONTROL - SCSA1406

UNIT IV ROBOT CONTROL

Unit 4 Robot Control

Basics of control: Transfer functions, Control laws: P, PD, PID Non-linear and advanced controls

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

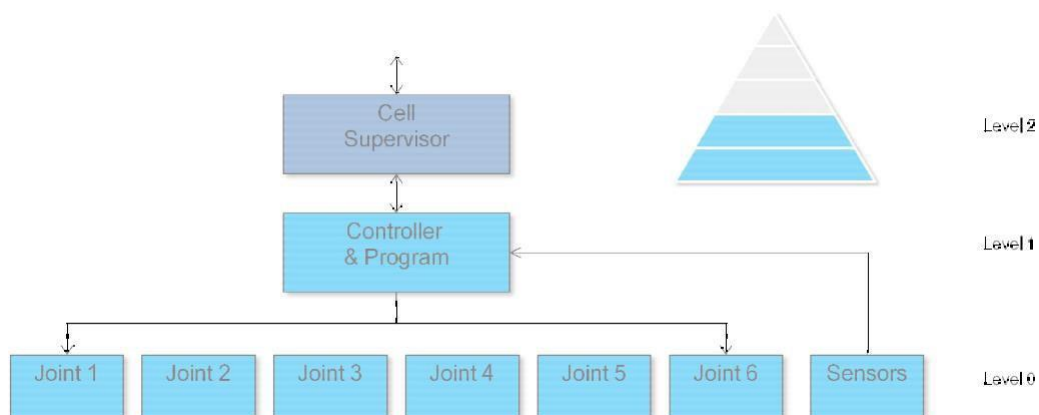


Fig 4.1 Robot Control system

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

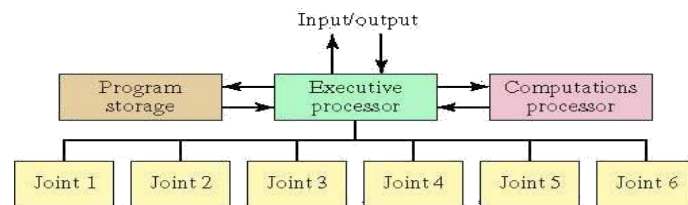


Fig 4.2 Robot Control System

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

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

- Control specifications.

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

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

Trajectory tracking “Tracking control in joint coordinates”

Control Methods

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

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

Servo Control

- Point to point Control
- Continuous Path Control
- Closed Loop control used to monitor position, velocity (other variables) of each joint
- the sequence of moves is controlled by a “sequencer”, which uses feedback received

from the end stops to index to next step in the program

- Low cost and easy to maintain, reliable
- relatively high speed
- repeatability of up to 0.01 inch
- limited flexibility
- typically hydraulic, pneumatic drives

Point-to-Point Control

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

Sensors in Robotics

Two basic categories of sensors used in industrial robots:

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

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

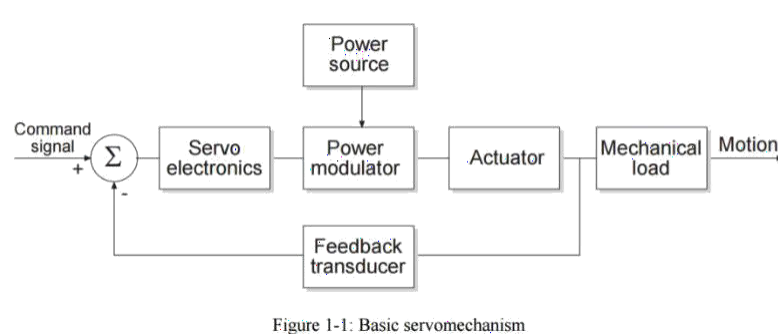


Figure 1-1: Basic servomechanism

Fig 4.3 Basic Servo Mechanics

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

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

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

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

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

End Effectors Types

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

Mechanical Grippers

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

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

Vacuum grippers

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

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

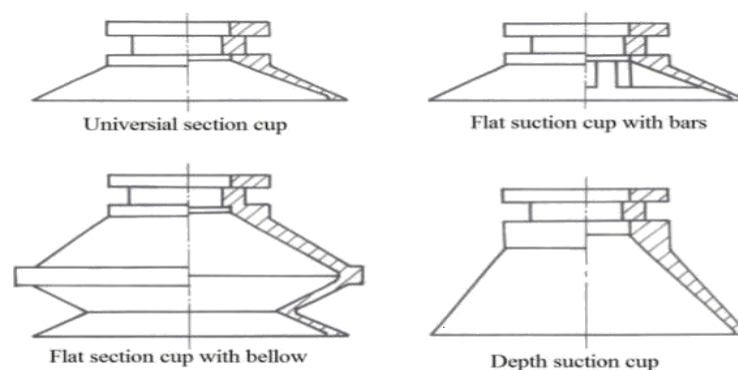


Figure 3 Different types of suction cups, picture taken from [5] page 204

Fig 4.4 Suction Cups

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

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

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

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

Magnetic grippers with



Fig 4.5 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.

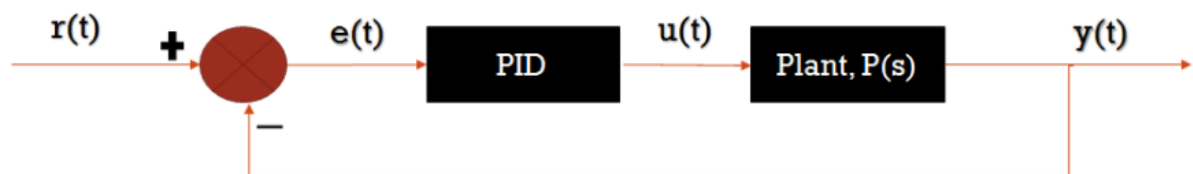
The *machining chips* may stick to the gripper during unloading.

PID is acronym for **Proportional Plus Integral Plus Derivative Controller**. It is a control loop feedback mechanism (controller) widely used in industrial control systems due to their robust performance in a wide range of operating conditions & simplicity. In This PID Controller Introduction, I have Tried To Illustrate The PID Controller With SIMPLE Explanations & BASIC MATLAB CODE To Give You

Idea About P,PI,PD & PID Controllers

For PID control, the actuating signal $u(t)$, consists of proportional error signal added with derivative and integral of error signal $e(t)$.

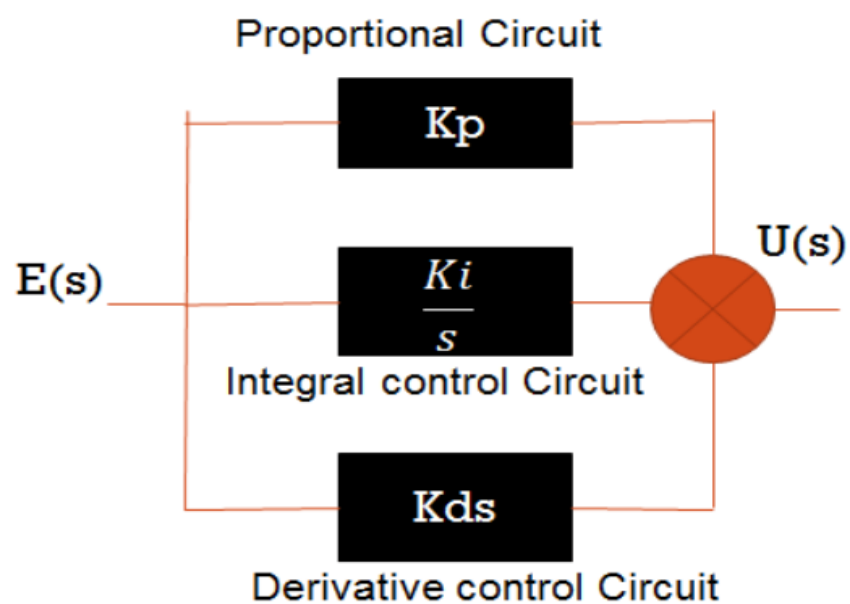
For PID control, the actuating signal $u(t)$, consists of proportional error signal added with derivative and integral of error signal $e(t)$



The Plant P is controlled by input $u(t)$ which is represented as

$$u(t) = K_p e(t) + K_d \frac{e(t)}{dt} + K_i \int e(t) dt$$

Physical Realisation of PID Controller



Physical realization of PID Controller

1. A **proportional controller** (K_p) will have the effect of reducing the rise time and will reduce, but never eliminate, the steady-state error.
2. An **integral control** (K_i) will have the effect of eliminating the steady-state error, but it may make **the transient response worse**.
3. A **derivative control** (K_d) will have the effect of increasing the stability of the system, reducing the overshoot, and improving the transient response but little effect on rise time

4. A **PD Controller** could add damping to a system, but the steady-state response is not affected.(steady state error is not eliminated)

5. A **PI Controller** could improve relative stability and eliminate steady state error at the same time, but the settling time is increased(System response sluggish)

But a **PID controller removes steady-state error** and decreases system settling times while maintaining **a reasonable transient response**

TEXT/REFERENCE BOOKS:

- 1.Saha, S.K., “Introduction to Robotics, 2nd Edition, McGraw-Hill Higher Education, New Delhi, 2014.
2. Ashitava Ghosal, “Fundamental Concepts and Analysis”, Oxford, New Delhi, 2006.
3. Mikell.P.Groover, Mitchell Weiss, Roger.N.Nagel, Nicholas.G.Odrey, “Industrial Robotics-Technology, Programming and Applications”, Tata McGraw-Hill Publishing Company Limited, New Delhi, Third Reprint 2008



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 ELECTRICAL AND ELECTRONICS ENGINEERING
DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING**

UNIT V- ROBOT ACUTUTAION SYSTEMS - SCSA1406

UNIT V

ROBOT ACTUATION SYSTEMS

Unit 5: Robot Actuation Systems

Actuators: Electric, Hydraulic and Pneumatic; Transmission: Gears, Timing Belts and Bearings, Parameters for selection of actuators. Control Hardware and Interfacing: Embedded systems: Architecture and integration with sensors, actuators, components, Programming for Robot Applications

Actuators are essential devices in robotics and widely common, particularly for industrial applications.

Actuation is the process of conversion of energy to mechanical form. An actuator is a hardware device that accomplishes this conversion. It converts a controller command signal into a change in a physical parameter.

Actuators come in various types and sizes, depending on the load associated with factors like force, torque, speed of operation, precision, accuracy, and power consumption. One of the prevalent types of actuators is electric motors such as servomotor, stepper motor, and direct current (DC) motors. A motor allows the robot to control a wheel, a switch, or even an arm.

Robot manufacturers usually use electric actuators since they are fast, efficient, and accurate. They are easy to control, can achieve high velocities (1000 – 10000 rpm), and have ideal torque for driving. At the same time, they are very weak or unpleasantly heavy because of their complexity.

Servo motor is a mechanism based on feedback control. It has a high maximum torque/force that allows high (de)acceleration. It is robust and has a high bandwidth that provides accurate and fast control.

Key advantages of servo motors are as follows:

- Power supply available everywhere.
- Low cost
- Large variety of products
- High power conversion efficiency
- Easy maintenance
- No pollution in the working environment

The disadvantages are:

- Overheating in static conditions
- Need special protection in flammable environments.

Stepper motors provide rotation in the form of discrete angular displacement. They can achieve precision angular rotation in both directions and are commonly employed to accommodate digital control technology. Stepper motors are, in general, heavier than

servomotors for the same power. The higher the voltage of electric motors, the better the power-to-weight ratio.

Then, we have electromechanical actuators that convert electrical energy into mechanical energy. Based on the basic principle of magnetism, they come in DC, AC, and stepper motors.

The third option is hydraulic and pneumatic actuators, which use fluid power and compressed air, respectively. Fluid power refers to the energy that is transmitted via a fluid under pressure. When pressure is applied to a confined chamber containing a piston, the piston will exert a force causing a motion. Pneumatic systems deliver the lowest power-to-weight ratio, while hydraulic systems have the highest power-to-weight ratio. Pneumatic actuators are mostly used for the opening and closing of grippers.

Hydraulic systems are very stiff and non-compliant, whereas pneumatic systems are easily compressed and thus are compliant. Notably, stiff systems have a more rapid response to changing loads and pressures and are more accurate. Although stiffness causes more responsive and more accurate systems, it also creates a danger if all things are not always perfect. Hydraulic actuators are very efficient, yet their cost is high.

Here are the advantages of a hydraulic actuator.

- Easy to control and accurate
- Simpler and easier to maintain
- Constant torque or force regardless of speed changes
- Easy to spot leakages of system
- Less noise

Disadvantages of the hydraulic actuator.

- Proper maintenance is required
- Expensive
- Leakage of the fluid creates environmental problems
- Wrong hydraulic fluid for a system can damage the components

Advantages of pneumatic actuators

- Clean, less pollution to the environment
- Inexpensive
- Safe and easy to operate

Disadvantages of pneumatic actuators

- Loud and noisy
- Lack of precision controls
- Sensitive to vibrations
- The fourth type of actuators is called piezoelectric actuators, which are successfully implemented in many applications today. They use the piezoelectric effect to create motion. When electricity flows through a piezoelectric material, it creates a physical deformation proportional to the applied electric field, known as the indirect piezoelectric effect.
- Piezoelectric actuators are used in loudspeakers, piezoelectric motors, acceleration sensors, vibration sensors, etc., and can be used to create either rotational or linear motion. These actuators' main advantages are their very high dynamics (up to 40 kHz), theoretically unlimited resolution (in the field of nanometers), high force, low consumption of electrical energy, and very compact construction.
- Another significant advantage is the possibility of having the actuator, force sensor, and position sensor contained in a one-piece unit. However, the main problem with implementing piezoelectric actuators is the small oscillating movements caused by its expansion and contraction.

Ideal Characteristics of Actuators

- Weight, power-to-weight ratio, operating pressure
- Stiffness against deformation
- Appropriate torque output.
- High torque density or the continuous output torque per mass
- High back drivability that protects the system against damage in environmental impacts, especially unexpected ones.
- High transparency and smooth energy flows between the actuator and end effector in both directions.
- High Efficiency.

Robot Programming

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

□ Leadthrough Programming

Method □ Textual Robot

Languages **Leadthrough**

Programming Method:

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

□ Powered

Leadthrough Method □

Manual Leadthrough

Method

a) **Powered Leadthrough Method:**

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

b) **Manual Leadthrough Method:**

In this method, the robot's end effector is moved physically by the programmer at the desired movements. Sometimes, it may be difficult to move large robot arm manually. To get rid of it a teach button is implemented in the wrist for special programming. The manual

leadthrough method is also known as Walk Through method. It is mainly used to perform continuous path movements. This method is best for spray painting and arc welding operations.

Textual Robot Languages:

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

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

Robot Programming Methods

There are three basic methods for programming industrial robots but currently over 90% are programmed using the teach 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 :-

JointCo-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 centre point of the robot and therefore move with it. This system is especially useful when the tool is near to the workpiece.

Workpiece Co-ordinates

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

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

Lead Through

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

Off-line Programming

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

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

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

Programming Languages for Robotics

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

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

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

PYTHON : There is another language which is used in designing Robots. Python is an object-oriented language which is used to access and control Robots. Python is an interpreted language; this language has an application in working with mobile

robots, particularly those manufactured by different companies. With python it is possible to use a single program for controlling many different robots. However Python is slower than C++ but it has so e good sides as well as it proved very easy to interact with robots using this language, it is highly portable and can be run in windows and MAC OSX plus it can easily be extendable using C and C++ language. Python is a very reliable language for string manipulation and text processing.

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

Robot, we do not need assembly code, but in complex de igring 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 w nt 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 langu ge. So these commands need to be converted in machine language so that robot can easily underst nd 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 provi ed 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 gi en below). Programming done in ROBOTC is very easy to do. Commands given are very str ghtforward. Like if we want our robot to switch on any hardware part, we just have to give code regarding to that action in text form. Suppose we want robot to turn motor of port, we just have to give command in this way:

Although program above is not exactly shown in the way in which it should be written, this is just to provide you a visualization of what we have told you. This is not written in an appropriate manner.ROBOTC provide advantage of speed, a Robot programmed in ROBOTC programming supports 45 times more speed than

provided by other programming based on C plus it has a very powerful debugging feature.

ROBOTICS.NXT :

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

Microsoft Robotics Developer Studio

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

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

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

If we want to graphics and visual effects in our programming, we use VPL. Visual Programming language is a programming language which allows us to create programs by doing manipulations in programming languages graphically. We

use boxes and arrows in this kind of programming while we want to show dataflow kind of things.

Visual programming language has huge application in animations. The last component which we are going to describe is Visual Simulation Environment. VSE provides 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 cant cover in a single article. This is just an introduction for those who want to get an idea about using languages in building of robots

Motion Commands and the Control of Effectors

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

The robot's operating system updates the states of all the effectors (servos, motors, LEDs, etc.) every few milliseconds. Each update is called a "frame", and can accommodate simultaneous changes to any number of effectors. On the AIBO, updates occur every 8 milliseconds and frames are buffered four at a time, so the application must have a new buffer available every 32 milliseconds; other robots may use different update intervals. In Tekkotsu these buffers of frames are produced by the MotionManager, whose job is to execute a collection of simultaneously active MotionCommands (MCs) of various types every few milliseconds. The results of these MotionCommands are assembled into a buffer that is passed to the operating system (Aperios for the AIBO, or Linux for other robots). Suppose we want the robot to blink its LEDs on and off at a rate of once per second. What we need is a MotionCommand that will calculate new states for

the LEDs each time the MotionManager asks for an update. LedMC, a subclass of both Motion Command and LedEngine, performs this service. If we create an instance of LedMC, tell it the frequency at which to blink the LEDs, and add it to the MotionManager's list of active MCs, then it will do all the work for us. There's just one catch: our application is running in the Main process, while the MotionManager runs in a separate Motion process. This is necessary to assure that potentially lengthy computations taking place in Main don't prevent Motion from running every few milliseconds. So how can we communicate with our MotionCommand while at the same time making it available to the Motion Manager.

Applications of Industrial Robots

Machine Loading

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

Spray Painting



Fig 5.1 Spray Painting

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

TEXT/REFERENCE BOOKS:

- 1.Saha, S.K., “Introduction to Robotics, 2nd Edition, McGraw-Hill Higher Education, New Delhi, 2014.
2. Ashitava Ghosal., “Fundamental Concepts and Analysis”, Oxford, New Delhi, 2006.
3. Mikell.P.Groover, Mitchell Weiss, Roger.N.Nagel, Nicholas.G.Odrey, “Industrial Robotics-Technology, Programming and Applications”, Tata McGraw-Hill Publishing Company Limited, New Delhi, Third Reprint 2008