INDUSTRIAL ROBOTS

2. ROBOTICS

The definition for a ***robot*** by *Robot Institute of America*, which defines the word "***robot***" relative to its role in manufacturing. The Institute has adopted the following definition:

*A **robot** is a reprogrammable multifunctional manipulator designed to move material, parts, tools, or specialized devices through variable programmed motions for the performance of a variety of tasks.*

This definition presumes that a robot must be:

1- ***reprogrammable;***

2- ***multifunctional.***

Which implies that the operation of a robot is ***flexible***.

Different countries have different standards for what they consider to be a robot. By American standards, a device must be easily ***reprogrammable*** to be considered a robot. Thus, manual-handling devices (i.e., a device that has multiple degrees of freedom and is actuated by an operator) or fixed-sequence robots (i.e., any device controlled by hard stops to control actuator motions on a fixed sequence and difficult to change) are not considered to be robots.

When a robot manipulator is used for a particular task such as spray painting, the manipulator may be reprogrammable depending on the computer, but it probably is not multifunctional; thus, the qualifications of being called a robot according to the foregoing definition may not be fulfilled. We observe that the definitions of a robot leave considerable freedom in these interpretations.

A ***manipulator*** is the main body of a robot and consists of the links, the joints, and other structural elements of a robot. Without other elements, the manipulator alone is not a robot.

A ***manipulator arm*** is one of the basic units in a robotic system. It is made of several links connected usually in series or by the joints to form an arm. The links can be numbered by starting from the base, which forms the zeroth link. A link is revolute or prismatic depending on the type of motion caused by the actuator attached to its joint:

- When the actuator of a joint causes rotational motion, the link is ***revolute*** (an articulated joint).
- When the actuator produces translational (linear) motion, the link is called ***prismatic***.

As an example, the Cincinnati robot manipulator shown in Figure (1.1). An end-effector, or a gripper, is attached to the arm by means of a wrist. A simple gripper usually has two opposing, moving plates for grasping an object Figure (1.2).

An industrial robot manipulator should be distinguished from a manipulator operated by a human. Such manipulators have been used, for example, in handling radioactive materials at nuclear power plants and in deep underwater operations. A robotic system in which there is a man-machine interaction (human, computer, manipulator) is usually referred to as a ***teleoperated manipulator***, or simply, a ***telerobot***.

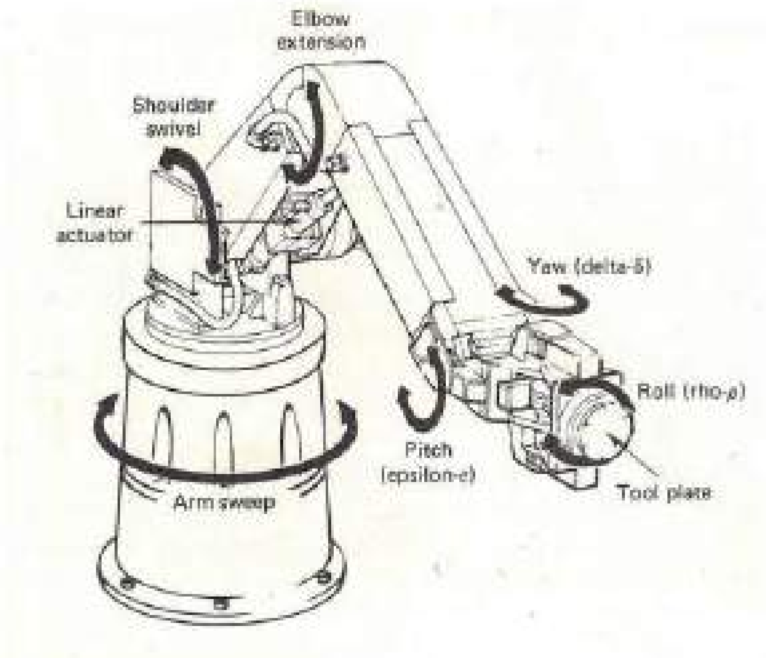


Figure (1.1). **Cincinnati Robot manipulator**

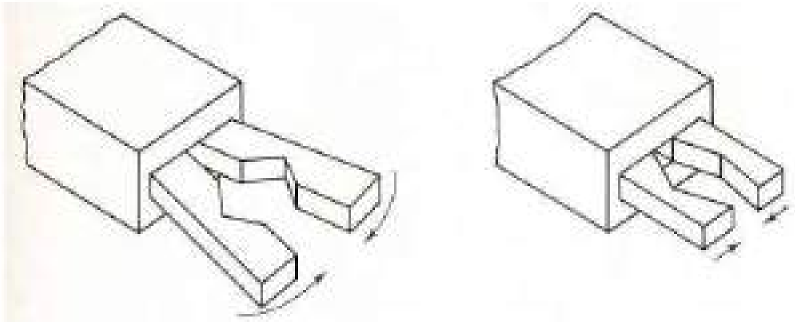


Figure (1.2). **Standard Angular and parallel Grippers**

The word ***robotics***, it is: the art, the knowledge base, and the know-how of designing, applying, the science and engineering dealing with robotic systems.

Robotic systems consist of not just robots, but also other devices and systems that are used together with the robots to perform the necessary tasks. Robots may be used in manufacturing environments, in underwater and space exploration, for aiding the disabled, or even for fun. In any capacity, robots can be useful, but need to be programmed and controlled. Robotics is an interdisciplinary subject that benefits from mechanical engineering, electrical and electronic engineering, computer science, biology, and many other disciplines.

1. CLASSIFICATION OF ROBOTS

The following is the classification of robots according to the Japanese Industrial Robot Association (***JIRA***):

Class 1: ***Manual-Handling Device:*** A device with multiple degrees of freedom that is actuated by an operator.

Class 2: ***Fixed-Sequence Robot:*** A device that performs the successive stages of a task according to a predetermined, unchanging method and is hard to modify.

Class 3: ***Variable-Sequence Robot:*** Same as class 2, but easy to modify.

Class 4: ***Playback Robot:*** A human operator performs the task manually by leading the robot, which records the motions for later playback. The robot repeats the same motions according to the recorded information.

Class 5: ***Numerical Control Robot:*** The operator supplies the robot with a movement program rather than teaching it the task manually.

Class 6: ***Intelligent Robot:*** A robot with the means to understand its environment and the ability to successfully complete a task despite changes in the surrounding conditions under which it is to be performed.

2. ADVANTAGES AND DISADVANTAGES OF ROBOTS

1. Robotics and automation can, in many situations, increase productivity, safety, efficiency, quality, and consistency of products.
2. Robots can work in hazardous environments without the need for life support, comfort, or concern about safety.
3. Robots need no environmental comfort, such as lighting, air conditioning, ventilation, and noise protection.
4. Robots work continuously without experiencing fatigue or boredom, do not get mad, do not have hangovers, and need no medical insurance or vacation.
5. Robots have repeatable precision at all times, unless something happens to them or unless they wear out.
6. Robots can be much more accurate than humans. Typical linear accuracies are a few thousands of an inch. New wafer-handling robots have micro-inch accuracies.
7. Robots and their accessories and sensors can have capabilities beyond that of humans.
8. Robots can process multiple stimuli or tasks simultaneously. Humans can only process one active stimulus.
9. Robots replace human workers creating economic problems, such as lost salaries, and social problems, such as dissatisfaction and resentment among workers.
10. Robots lack capability to respond in emergencies, unless the situation is predicted and the response is included in the system. Safety measures are needed to ensure that they do not injure operators and machines working with them. This includes:
    1) inappropriate or wrong responses;

2) a lack of decision-making power;
3) a loss of power;
4) damage to the robot and other devices;
5) human injuries.

11. Robots, although superior in certain senses, have limited capabilities in:
1) degrees of freedom;
2) dexterity;
3) sensors;
4) vision systems;
5) real-time response.

12. Robots are costly, due to:
1) initial cost of equipment;
2) installation costs;
3) need for peripherals;
4) need for training;
5) need for programming.

3. ROBOT COMPONENTS

A robot, as a system, consists of the following elements, which are integrated together to form a whole:

✓ **Manipulator**

This is the main body of the robot and consists of the links, the joints, and other structural elements of the robot. Without other elements, the manipulator alone is not a robot.

✓ **End effector**

This is the part that is connected to the last joint (hand) of a manipulator, which generally handles objects, makes connection to other machines or performs the required tasks. Robot manufacturers generally do not design or sell end effectors. In most cases, all they supply is a simple gripper. Generally, the hand of a robot has provisions for connecting specialty end effectors that are specifically designed for a purpose. This is the job of a company's engineers or outside consultants to design and install the end effector on the robot and to make it work for the given situation. A welding torch, a paint spray gun, a glue-laying device, and a parts handler are but a few of the possibilities. In most cases, the action of the end effector is controlled by the robot's controller.

- ✓ **Actuators and Drives**

Actuators are the "muscles" of the manipulators. Common types of actuators and drives are servomotors, stepper motors, pneumatic cylinders, and hydraulic cylinders. There are also other actuators that are more novel and are used in specific situations. Actuators are controlled by the robot controller.

- ✓ **Sensors**

Sensors are used to collect information about the internal state of the robot or to communicate with the outside environment. As in humans, the robot controller needs to know where each link of the robot is in order to know the robot's configuration. For robots; sensors integrated into the robot send information about each joint or link to the controller, which determines the configuration of the robot. Robots are often equipped with external sensory devices such as a vision system, touch and tactile sensors, speech synthesizers, etc., which enable the robot to communicate with the outside world.

- ✓ **Controller**

The controller receives its data from the computer, controls the motions of the actuators, and coordinates the motions with the sensory feedback information. Suppose that in order for the robot to pick up a part from a bin, it is necessary that its first joint be at 35°. If the joint is not already at this magnitude, the controller will send a signal to the actuator (a current to an electric motor, air to a pneumatic cylinder, or a signal to a hydraulic servo valve), causing it to move. It will then measure the change in the joint angle through the feedback sensor attached to the joint (a potentiometer, an encoder, etc.). When the joint reaches the desired value, the signal is stopped. In more sophisticated robots, the velocity and the force exerted by the robot are also controlled by the controller.

- ✓ **Processor**

The processor is the brain of the robot. It calculates the motions of the robot's joints, determines how much and how fast each joint must move to achieve the desired location and speeds, and oversees the coordinated actions of the controller and the sensors. The processor is generally a computer, which works like all other computers, but is dedicated to a single purpose. It requires an operating system, programs, peripheral equipment such as monitors, and has many of the same limitations and capabilities of a PC processor.

- ✓ **Software**

There are perhaps three groups of software that are used in a robot. One is the **operating system**, which operates the computer. The second is the **robotic software,** which calculates the necessary motions of each joint based on the kinematic equations of the robot. This information is sent to the controller. This software may be at many different levels, from machine language to sophisticated languages used by modern robots. The third group is the **collection of routines and application programs** that are developed in order to use the peripheral devices of the robots, such as vision routines, or to perform specific tasks.

It is important to note that in many systems, the controller and the processor are placed in the same unit. Although these two units are in the same box, and even if they are integrated into the same circuit, they have two separate functions.

4. ANATOMY OF A ROBOT

The industrial robots resemble the human arm in its physical structure. Like the hand attached to the human body the robot manipulator or robot arm is attached to the base. The chest, the upper arm and fore-arm in the human body compare with the links in the robot manipulators. The wrist, elbow and the shoulder in the human hand are represented by the joints in the robot arm. As the industrial robot arm compares with the human hand, they are also known as "anthropomorphic or articulated robots (Figure (8.1)).

Anatomy		*Representation*
1. Body	→	Base
2. Chest	→	Link
3. Shoulder	→	Joint
4. Upper arm	→	Link
5. Elbow	→	Joint
6. For-arm	→	Link
7. Wrist	→	Joint

The drives or motion to the links is provided at the joints. The joints motions can be **rotational** or **translatory (sliding)**. The tool known as **end-effector (gripper)** is attached to the **wrist**.

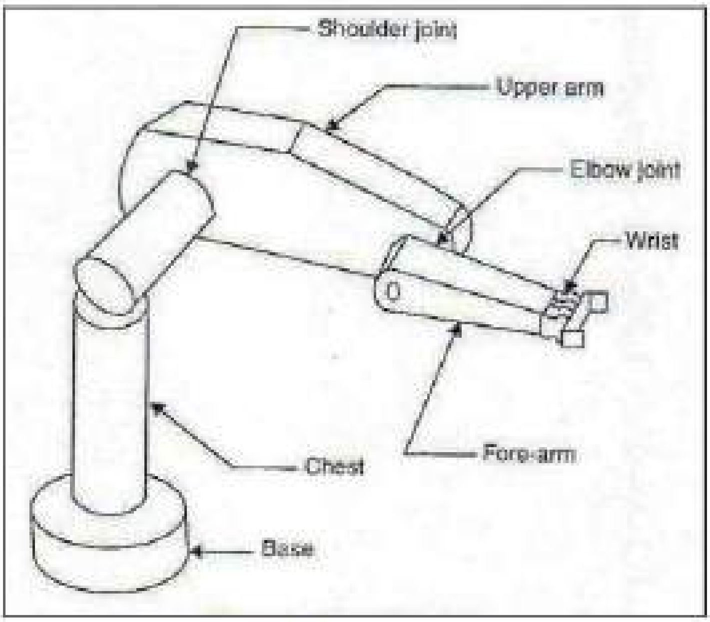


Figure (8.1). **Anatomy of a robot**

5. ROBOT DEGREES OF FREEDOM

The three Cartesian coordination (x, y, and z) are necessary and sufficient to define the location of the point. Although the three coordinates may be expressed in terms of different coordinate systems, they are always necessary.

A body suspended in space can have six positive degrees of freedom and six negative degrees of freedom. The three degrees of freedom are the translatory or linear degrees of freedoms along the positive cartesian axes and three along the negative cartesian axes which are opposite. Six rotary movements about the cartesian axes of which three are clockwise and remaining three are anticlockwise.

An object is said to have ***n*** **degrees of freedom** (DOF) if its configuration can be minimally specified by ***n*** parameters. Thus, the number of DOF is equal to the dimension of the configuration space. For a robot manipulator, the number of joints determines the number of DOF. A rigid object in three-dimensional space has six DOF: three for **positioning** and three for **orientation.** Therefore, a manipulator should typically possess at least six independent DOF. With fewer than six DOF the arm cannot reach every point in its work space with arbitrary orientation. Certain applications such as reaching around or behind obstacles may require more than six DOF. A manipulator having more than six DOF is referred to as a **kinematically redundant** manipulator.

6. ROBOT JOINTS

Robots may have different types of joints, such as linear, rotary, sliding, or spherical. Although spherical joints are common in many systems, since they possess multiple degrees of freedom, and thus, are difficult to control, spherical joints are not common in robotics, except in research. Most robots have either a linear (prismatic) joint or a rotary (revolute) joint.

Prismatic joints are linear; there is no rotation involved. They are either hydraulic or pneumatic cylinders, or they are linear electric actuators. These joints are used in gantry, cylindrical, or similar joint configurations.

Revolute joints are rotary, and although hydraulic and pneumatic rotary joints are common, most rotary joints are electrically driven, either by stepper motors or, more commonly, by servomotors.

7. ROBOT COORDINATES

Robot configurations generally follow the coordinate frames with which they are defined, as shown in Figure (8.2). Prismatic joints are denoted by P, revolute joints are denoted by R, and spherical joints are denoted by S. Robot configurations are specified by a succession of P's, R's, or S's. For example, a robot with three prismatic and three revolute joints is specified by 3P3R. The following configurations are common for positioning the hand of the robot:

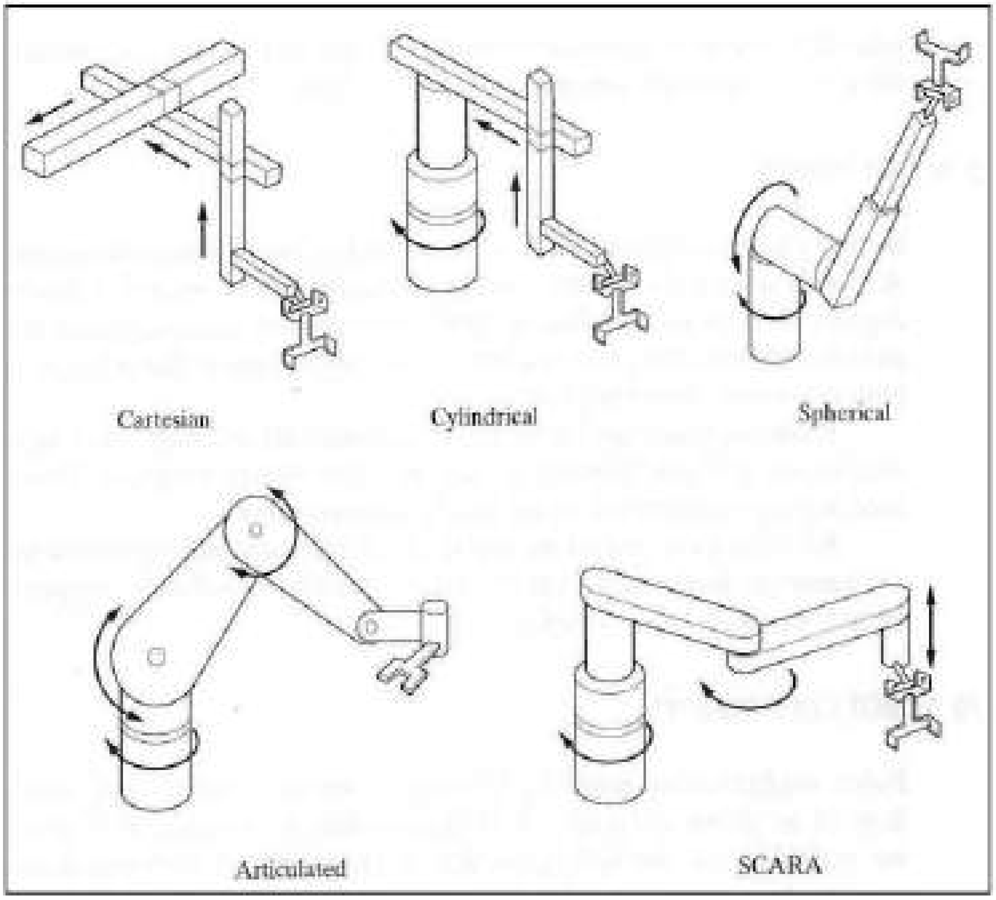


Figure (8.2). **Some possible robot coordinate frames**

✓ **Cartesian/rectangular/gantry (3P)**

These robots are made of three linear joints that position the end effector, which are usually followed by additional revolute joints that orientate the end effector.

✓ **Cylindrical(R2P)**

Cylindrical coordinate robots have two prismatic joints and one revolute joint for positioning the part, plus revolute joints for orientating the part.

- ✓ **Spherical(2RP)**

Spherical coordinate robots follow a spherical coordinate system, which has one prismatic and two revolute joints for positioning the part, plus additional revolute joints for orientation.

- ✓ **Articulated/anthropomorphic (3R)**

An articulated robot's joints are all revolute, similar to a human's arm. They are perhaps the most common configuration for industrial robots.

- ✓ **Selective Compliance Assembly Robot Arm (SCARA)**

SCARA robots have two revolute joints that are parallel and allow the robot to move in a horizontal plane, plus an additional prismatic joint that moves vertically. SCARA robots are very common in assembly operations. Their specific characteristic is that they are more compliant in the x-y-plane, but are very stiff along the z-axis, and thus have selective compliance.

7. ROBOT REFERENCE FRAMES

Robots may be moved relative to different coordinate frames. In each type of coordinate frame, the motions will be different. Usually, robot motions are accomplished in the following three coordinate frames (Figure 8.3):

1) World Reference Frame

This is a universal coordinate frame, as defined by x, y, z-axes. In this case, the joints of the robot move simultaneously so as to create motions along the three major axes. In this frame, for example, no matter where the arm is, a positive x-axis movement is always in the positive direction of the x-axis; this coordinate is used to:

- define the motions of the robot relative to other objects;
- define other parts and machines that the robot communicates with;
- define motion paths.

2) Joint Reference Frame

This is used to specify movements of each individual joint of the robot. Suppose that you want to move the hand of a robot to a particular position. In case of moving one joint at a time in order to direct the hand to the desired location, each joint may be accessed individually, and, thus, only one joint moves at a time. Depending on the type of joint used (prismatic, revolute, or spherical), the motion of the robot hand will be different. For instance, if a revolute joint is moved, the hand will move around a circle defined by the joint axis.

3) Tool Reference Frame

This specifies movements of the robot's hand relative to a frame attached to the hand. The x'-, y'-, and z'-axes attached to the hand define the motions of the hand relative to this local frame. Unlike the universal World frame, the local Tool frame moves with the robot. Suppose that the hand is pointed as shown in Figure (8.3). Moving the hand relative to the positive x-axis of the local Tool frame will move the hand along the x'-axis of the Tool frame. If the arm were pointed elsewhere, the same motion along the local x'-axis of the Tool frame would be completely different from the first motion. The same +x'-axis movement would be upward if the x'-axis were pointed upwards, and it would be downward if the x'-axis were pointed downward. As a result, the Tool reference frame is a moving frame that changes continuously as the robot moves, so the ensuing motions relative to it are also different, depending on where the arm is and what direction the Tool frame has. All joints of the robot must move simultaneously to create coordinated motions about the Tool frame. The Tool reference frame is an extremely useful frame in robotic programming, where the robot is to approach and depart from other objects or to assemble parts.

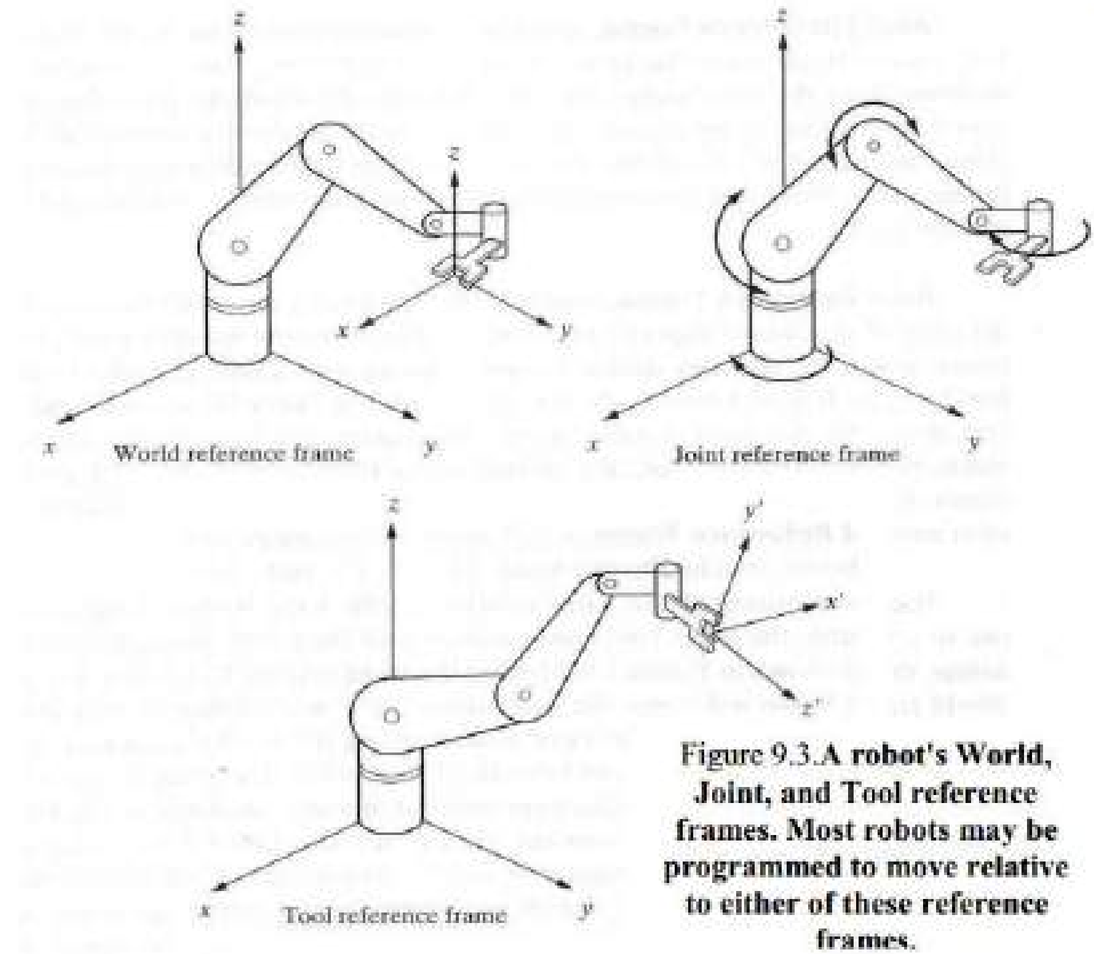


Figure 9.3.A robot's World, Joint, and Tool reference frames. Most robots may be programmed to move relative to either of these reference frames.

8. PROGRAMMING MODES

Robots may be programmed in a number of different modes, depending on the robot and its sophistication. The following programming modes are very common:

1) Physical Setup

In this mode, an operator sets up switches and hard stops that control the motion of the robot. This mode is usually used along with other devices, such as Programmable Logic Controllers (PLC).

2) Lead Through or Teach Mode

In this mode, the robot's joints are moved with a teach pendant. When the desired location and orientation is achieved, the location is entered (taught) into the controller. During playback, the controller will move the joints to the same locations and orientations. This mode is usually point to point, where the motion between points is not specified or controlled. Only the points that are taught are guaranteed to reach.

3) Continuous Walk-Through Mode

In this mode, all robot joints are moved simultaneously, while the motion is continuously sampled and recorded by the controller. During playback, the exact motion that was recorded is executed. The motions are taught by an operator, either through a model, by physically moving the end effector, or by directing the robot arm and moving it through its workspace. Painting robots, for example, are programmed by skilled painters through this mode.

4) Software Mode

In this mode of programming the robot, a program is written off-line or on-line and is executed by the controller to control the motions. The programming mode is the most sophisticated and versatile mode and can include sensory information, conditional statements (such as if...then statements), and branching. However, it requires the knowledge of the operating system of the robot before any program is written.

Most industrial robots can be programmed in more than one mode.

9. ROBOT CHARACTERISTICS

The following definitions are used to characterize robot specifications:

1) Payload

Payload is the weight a robot can carry and still remain within its other specifications. For example, a robot's maximum load capacity may be much larger than its specified payload, but at the maximum level, it may become less accurate, may not follow its intended path accurately, or may have excessive deflections. The payload of robots compared with their own weight is usually very small. For example, Fanuc Robotics LR Mate™ robot has a mechanical weight of 86 lb sand a payload of 6.6 lbs, and the M-16i™ robot has a mechanical weight of 594 lbs and a payload of 35 lbs.

2) Reach

Reach is the maximum distance a robot can reach within its work envelope. As we will see later, many points within the work envelope of the robot may be reached with any desired orientation (called dexterous). However, for other points, close to the limit of robot's reach capability, orientation cannot be specified as desired (called non dexterous point). Reach is a function of the robot's joint lengths and its configuration.

3) Precision (validity)

Precision is defined as how accurately a specified point can be reached. This is a function of the resolution of the actuators, as well as its feedback devices. Most industrial robots can have precision of 0.001 inch or better.

4) Repeatability (variability)

Repeatability is how accurately the same position can be reached if the motion is repeated many times. Suppose that a robot is driven to the same point 100 times. Since many factors may affect the accuracy of the position, the robot may not reach the same point every time, but will be within a certain radius from the desired point. The radius of a circle that is formed by this repeated motion is called repeatability. Repeatability is much more important that precision. If a robot is not precise, it will generally show a consistent error, which can be predicted and thus corrected through programming. As an example, suppose that a robot is consistently off 0.05 inch to the right. In that case, all desired points can be specified at 0.05 inch to the left, and thus the error can be eliminated. However, if the error is random, it cannot be predicted and thus cannot be eliminated.

Repeatability defines the extent of this random error. Repeatability is usually specified for a certain number of runs. More tests yield larger (bad for manufacturers) and more realistic (good for the users) results. Manufacturers must specify repeatability in conjunction with the number of tests, the applied payload during the tests, and the orientation of the arm. For example, the repeatability of an arm in a vertical direction will be different from when the arm is tested in a horizontal configuration. Most industrial robots have repeatability in the 0.001 inch range.

10. ROBOT WORKSPACE

Depending on their configuration and the size of their links and wrist joints, robots can reach a collection of points called a workspace. The shape of the workspace for each robot is uniquely related to its characteristics. **The workspace may be found mathematically** by writing equations that define the robot's links and joints and including their limitations, such as ranges of motions for each joint. Alternatively, **the workspace may be found empirically,** by moving each joint through its range of motions and combining all the space it can reach and subtracting what it cannot reach. Figure (8.4) shows the approximate workspace for some common configurations. **When a robot is being considered for a particular application, its workspace must be studied to ensure that the robot will be able to reach the desired points. For accurate workspace determination, please refer to manufacturers' data sheets.**

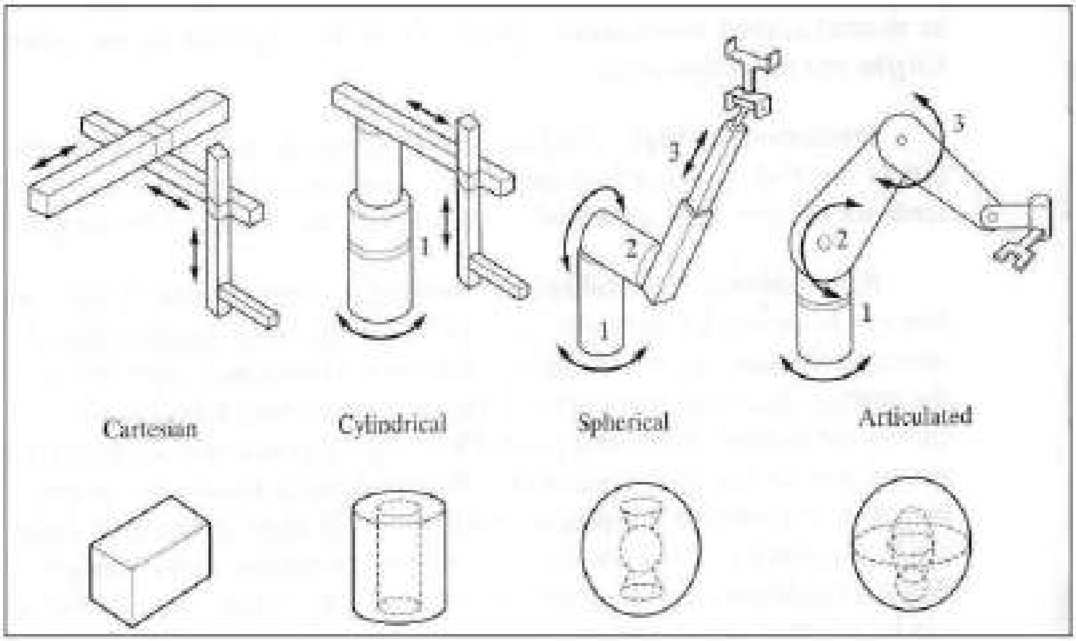


Figure (8.4). **Typical workspaces for common robot configurations**

11. ROBOT LANGUAGES

In order to use any particular robot, its programming language must be learned. Many robot languages are based on some other common language, such as Cobol, Basic, C, and Fortran. Other languages are unique and not directly related to any other common language.

Robotic languages are at different levels of sophistication, depending on their design and application. This ranges from machine level to a proposed human intelligence level.

✓ **High-level languages**

High-level languages are either interpreter based or compiler based:

1) Interpreter-based languages execute one line of the program at a time, and each line has a line number. The interpreter interprets the line every time it is encountered (by converting the line to a machine language that the processor can understand and execute) and executes each line sequentially. The execution continues until the last line is encountered or until an error is detected. The advantage of an interpreter-based language is in its ability to continue execution until an error is detected, which allows the user to run and debug the program portion by portion. Thus, debugging programs is much faster and easier. However, because each line is interpreted every time, execution is slower and not very efficient. Many robot languages, such as Unimation™ VAL® and IBM's AML® (A Manufacturing Language), are interpreter based.

2) Compiler-based languages use a compiler to translate the whole program into machine language (which creates an object code) before it is executed. Since the processor executes the object code during execution, these programs are much faster and more efficient. However, since the whole program must first be compiled, it is impossible to run any part of the program if any error is present. As a result, debugging compiler-based programs is much more difficult. Certain languages, such as AL©, are more flexible. They allow the user to debug the program in interpreter mode, while the actual execution is in compiler mode.

The following is a general description of different levels of robotic languages:

- **Microcomputer Machine Language Level**

In this level, the programs are written in machine language. This level of programming is the most basic and is very efficient, but difficult to understand and to follow. All languages will eventually be interpreted or compiled to this level. However, in the case of higher level programs, the user writes the programs in a higher level language, which is easier to follow and understand.

- **Point-to-Point Level**

In this level (such as in Funky and Cincinnati Milacron's T3), the coordinates of the points are entered sequentially, and the robot follows the points as specified. This is a very primitive and simple type of program; is easy to use, but not very powerful. It also lacks branching, sensory information, and conditional statements.

- **Primitive Motion Level**

In these languages, it is possible to develop more sophisticated programs, including sensory information, branching, and conditional statements (such as VAL by Unimation™). Most languages of this level are interpreter based.

- **Structured Programming Level**

Most languages of this level are compiler based, are powerful, and allow more sophisticated programming. However, they are also more difficult to learn.

- **Task-Oriented Level**

Currently, there are no actual languages of this level in existence. **Autopass**, proposed by IBM in the 1980s, to be task oriented. This means that instead of programming a robot to perform a task by programming each and every step necessary to complete the task, the user was simply to mention the task, while the controller would create the necessary sequence. Imagine that a robot is to sort three boxes by size. In all existing languages, the programmer will have to tell the robot exactly what to do, which means that every step must be programmed. The robot must be told how to go to the largest box, how to pick up the box, where to place it, go to the next box, etc. In Autopass, the user would only indicate "sort," while the robot controller would create this sequence automatically.

12. ROBOT APPLICATIONS

Robots have already been used in many industries and for many purposes. They can often perform better than humans and at lower costs. For example, a welding robot can probably weld better than a human welder, because the robot can move more uniformly and more consistently. In addition, robots do not need protective goggles, protective clothing, ventilation, and many other necessities that their human counterparts do. As a result, robots can be more productive and better suited for the job, as long as the welding job is set up for the robot for automatic operations and nothing changes and as long as the welding job is not too complicated. Similarly, a robot exploring the ocean bottom would require far less attention than a human diver. Also, the robot can stay underwater for long periods and can go to very large depths and still survive the pressure; it also does not require oxygen. The following is a list of some applications where robots are useful:

1) **Machine loading;**
2) **Pick and place operations;**
3) **Welding;**
4) **Painting;**
5) **Inspection.**
6) **Sampling.**
7) **Assembly operations.**
8) **Manufacturing.**
9) **Surveillance.**
10) **Medical applications.**
11) **Assisting disabled individuals.**
12) **Hazardous environments.**
13) **Underwater, space, and remote locations.**
14) **Other robot applications like military, Animatronics.**

ROBOT KINEMATICS

1. MATRIX REPRESENTATION

Matrices can be used to represent points, vectors, frames, translations, rotations, and transformations, as well as objects and other kinematic elements in a frame. We will use this representation throughout this book to derive equations of motion for robots.

1.1. Representation of a Point in Space

A point P in space (Fig. (9.1)) can be represented by its three coordinates relative to a reference frame:

$$P = a_x\hat{\mathbf{i}} + b_y\hat{\mathbf{j}} + c_z\hat{\mathbf{k}},$$

where a_x, b_y and c_z are the three coordinates of the point represented in the reference frame. Obviously, other coordinate representations can also be used to describe the location of a point in space.

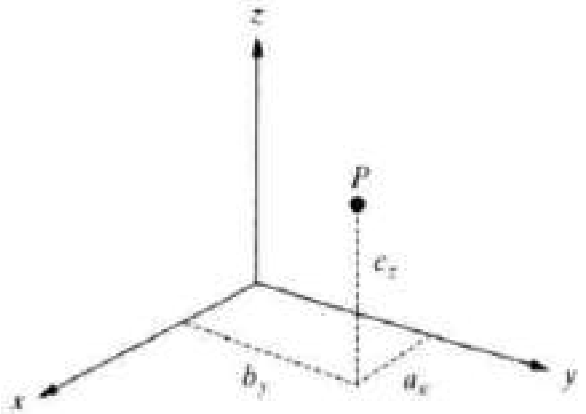


Figure (9.1). **Representation of a point in space**

1.2. Representation of a Vector in Space

A vector can be represented by three coordinates of its tail and of its head. If the vector starts at a point A and ends at point B, then it can be represented by

$$\bar{P}_{AB} = (B_x - A_x)\hat{\mathbf{i}} + (B_y - A_y)\hat{\mathbf{j}} + (B_z - A_z)\hat{\mathbf{k}}.$$

Specifically, if the vector starts at the origin (Fig. (9.2)), then:

$$\bar{P} = a_x\hat{\mathbf{i}} + b_y\hat{\mathbf{j}} + c_z\hat{\mathbf{k}},$$

where a_x, b_y and c_z are the three components of the vector in the reference frame. In fact, point P in the previous section is in reality represented by a vector connected to it at point P and expressed by the three components of the vector. The three components of the vector can also be written in a matrix form, as:

$$\bar{P} = \begin{bmatrix} a_x \\ b_y \\ c_z \end{bmatrix}.$$

This representation can be slightly modified to also include a scale factor w such that if x, y and z are divided by w, they will yield a_x, b_y and c_z. Thus, the vector can be written as:

$$\bar{P} = \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix}, \text{where } a_x = \frac{x}{w}, b_y = \frac{y}{w}, c_z = \frac{z}{w}.$$

Variable w may be any number, and as it changes, it can change the overall size of the vector. This is similar to zooming a picture in computer graphics. As the value of w changes, the size of the vector changes accordingly. If w is greater than unity, all vector components enlarge; if w is less than unity, all vector components become smaller. This is also used in computer graphics for changing the size of pictures and drawings.

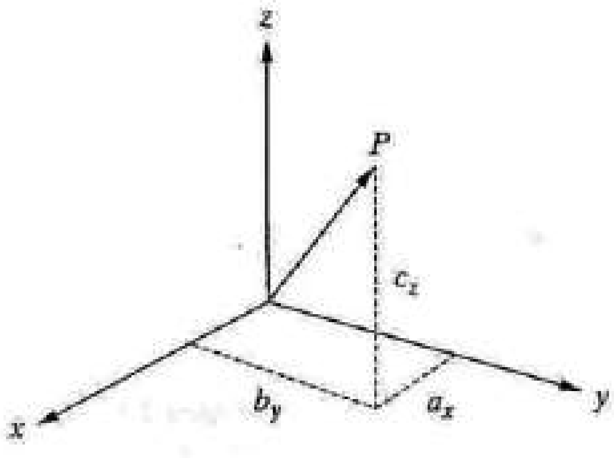


Figure (9.2). **Representation of a vector in space**

If w is unity, the size of the components remain unchanged. However, if $w = 0$, then a_x, b_y and c_z will be infinity. In this case, x, y and z (as well as a_x, b_y and c_z) will represent a vector whose length is infinite, but nonetheless, is in the direction represented by the vector. This means that a **directional vector** can be represented by a scale factor of $w = 0$, where the length is not of importance, but the direction is represented by the three components of the vector.

Example 9.1.

A vector is described as $\bar{P} = 3\hat{\mathbf{i}} + 5\hat{\mathbf{j}} + 2\hat{\mathbf{k}}$. Express the vector in matrix form:

1) with a scale factor of 2;
2) if it were to describe a direction as a unit vector.

Solution: The vector can be expressed in matrix form with a scale factor of 2, as well as 0 for direction, as:

$$\bar{P} = \begin{bmatrix} 6 \\ 10 \\ 4 \\ 2 \end{bmatrix} \quad \text{and} \quad \bar{P} = \begin{bmatrix} 3 \\ 5 \\ 2 \\ 0 \end{bmatrix}.$$

However, in order to make the vector into a unit vector, we will normalize the length such that the new length will be equal to unity. To do this, each component of the vector will be divided by the square root of the sum of the squares of the three components:

$$\lambda = \sqrt{p_x^2 + p_y^2 + p_z^2} = 6.16, \text{where } p_x = \frac{3}{6.16} = 0.481, p_y = \frac{5}{6.16}, \text{ect.,}$$

$$\text{and } \bar{P}_{\text{unit}} = \begin{bmatrix} 0.487 \\ 0.811 \\ 0.324 \\ 0 \end{bmatrix}.$$

1.3. Representation of the Reference Frame at the Origin

A frame centered at the origin of a reference frame is represented by three vectors, usually mutually perpendicular to each other, called unit vectors $\bar{n}$, $\bar{o}$, $\bar{a}$, for *normal, orientation,* and *approach* vectors (Fig. (9.3)). Each unit vector is represented by its three components in the reference frame. Thus, a frame F can be represented by three vectors in a matrix form as:

$$F = \begin{bmatrix} n_x & o_x & a_x \\ n_y & o_y & a_y \\ n_z & o_z & a_z \end{bmatrix}.$$

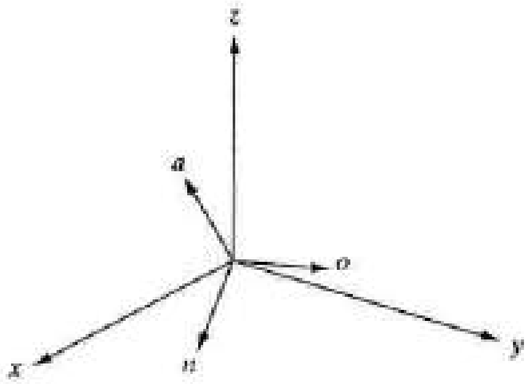


Figure (9.3). **Representation of a reference frame at the origin**

1.4. Representation of a Frame in Space Relative to the Reference Frame

If a frame is not at the origin, then the location of the origin of the frame relative to the reference frame must also be expressed. In order to do this, a vector will be drawn between the origin of the frame and the origin of the reference frame describing the location of the frame (Fig. (9.4)). This vector is expressed through its components relative to the reference frame. Thus, the frame can be expressed by three vectors describing its directional unit vectors, as well as a fourth vector describing its location as follows:

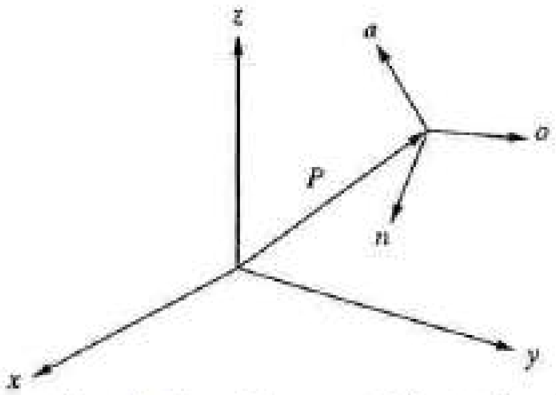


Figure (9.4). **Representation of a frame in space relative to the reference frame**

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

As shown in the last Equation above, the first three vectors are directional vectors with $w = 0$ representing the directions of the three unit vectors of the frame $\bar{n}$, $\bar{o}, \bar{a}$, while the fourth vector with $w = 1$ represents the location of the origin of the frame relative to the reference frame. Unlike the unit vectors, the length of vector P is important to us, and thus we use a scale factor of one. A frame may also be represented with a 3×4 matrix without the scale factors.

Example 9.2.

The frame F shown in Fig. (9.5) is located at 3, 5, 7 units, with its n-axis parallel to x, its o-axis at 45° relative to the y-axis, and its a-axis at 45° relative to the z-axis. The frame can be described by:

$$F = \begin{bmatrix} 1 & 0 & 0 & 3 \\ 0 & 0.707 & -0.707 & 5 \\ 0 & 0.707 & 0.707 & 7 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

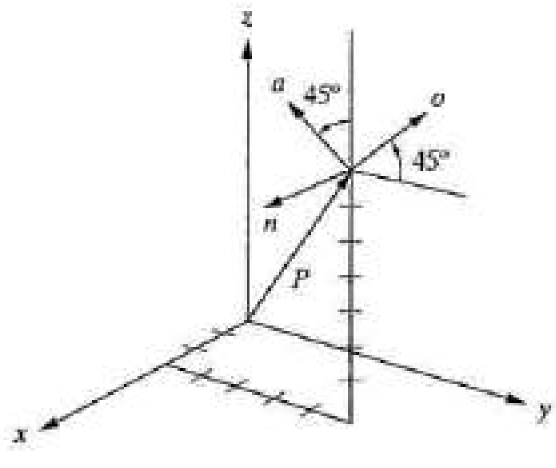


Figure (9.5). **An example of representation of a frame in space**

1.5. Representation of a Rigid Body

An object can be represented in space by attaching a frame to it and representing the frame in space. Since the object is permanently attached to this frame, its position and orientation relative to this frame is always known (Fig. (9.6)). A rigid body in space has 6 degrees of freedom, meaning that not only it can move along three axes of x, y and z, but can also rotate about these three axes. Thus, all that is needed to completely define an object in space is 6 pieces of information describing the position of the origin of the object in the reference frame relative to the three reference axes, as well as its orientation about the three axes. However, as can be seen in the Equation, 12 pieces of information are given, 9 for orientation, and 3 for position. Obviously, there must be some constraints present in this representation to limit the preceding to 6. Thus, we need 6 constraint equations to reduce the amount of information from 12 to 6 pieces. The constraints come from the known characteristics of the frame:

- the three unit vectors $\bar{n}$, $\bar{o}, \bar{a}$, are mutually perpendicular;
- each unit vectors length must be equal to unity.

These constraints translate into the following six constraint equations:

1) $\bar{n} \cdot \bar{o} = 0$. (The dot product of $\bar{n}$ and $\bar{o}$ vectors must be zero);
2) $\bar{n} \cdot \bar{a} = 0$;
3) $\bar{a} \cdot \bar{o} = 0$;
4) $|n| = 1$. (The magnitude of the length of the vector must be 1);
5) $|o| = 1$;
6) $|a| = 1$.

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

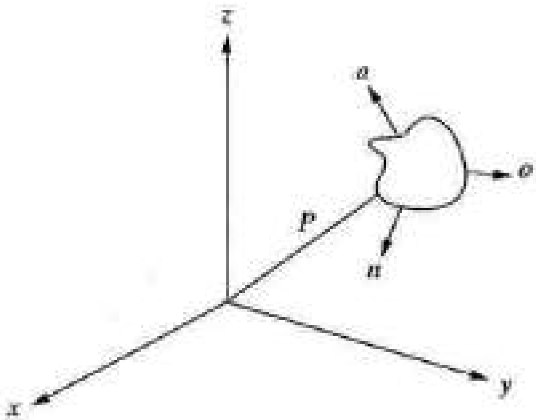


Figure (9.6). **Representation of an object in space**

2. HOMOGENEOUS TRANSFORMATION MATRICES

For a variety of reasons, it is desirable to keep transformation matrices in square form, either (3 × 3) or (4 × 4):

- it is much easier to calculate the inverse of square matrices than rectangular matrices;
- in order to multiply two matrices, their dimensions must match, such that the number of columns of the first matrix must be the same as the number of rows of the second matrix.

To keep representation matrices square, if we represent both orientation and position in the same matrix, we will add the scale factors to the matrix to make it a 4 × 4 matrix. If we represent the orientation alone, we may either drop the scale factors and use 3 × 3 matrices, or add the 4th column with zeros for position in order to keep the matrix square. Matrices of this form are called **homogeneous matrices**, and we write them as follows:

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

Combining the rotation matrix (3 x 3), the homogeneous translational vector (4 x 1) and a perspective matrix (1 x 3) a homogeneous transformation matrix (4 x 4) is arrived at.

$$\underset{(4\times 4)}{F=}\begin{bmatrix} \underset{(3\times 3)}{Rotation\ Matrix} & \underset{(3\times 1)}{Position\ Vector} \\ \underset{(1\times 3)}{Perspective\ Transformation} & \underset{(1\times 1)}{Scale\ Factor} \end{bmatrix}$$

3. REPRESENTATION OF TRANSFORMATIONS

A transformation is defined as making a movement in space. When a frame (a vector, an object, or a moving frame) moves in space relative to a fixed reference frame, we can represent this motion in a form similar to a frame representation. This is because a transformation itself is a change in the state of a frame (representing the change in its position and orientation), and thus it can be represented as a frame. A transformation may be in one of the following forms:

- a pure translation;
- a pure rotation about an axis;
- a combination of translations and rotations.

To see how these can be represented, we will study each one separately.

3.1. Representation of a Pure Translation

If a frame (representing an object) moves in space without any change in its orientation, the transformation is a pure translation. In this case, the directional unit vectors remain in the same direction and thus do not change. All that changes is the location of the origin of the frame relative to the reference frame, as shown in Fig. (9.7).

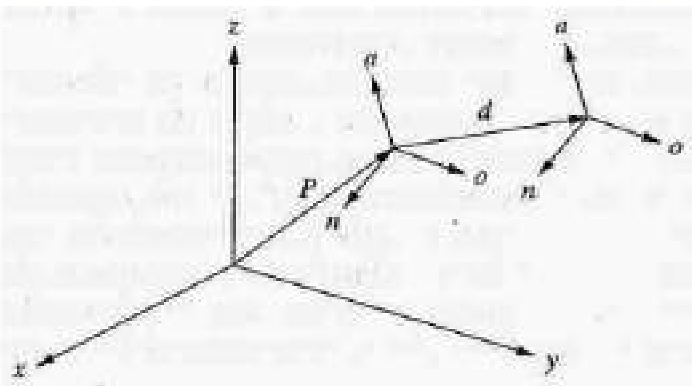


Figure (9.7). **Representation of a pure translation in space**

The new location of the frame relative to the fixed reference frame can be found by adding the vector representing the translation to the vector representing the original location of the origin of the frame. In matrix form, the new frame representation may be found by multiplying the frame with a matrix representing the transformation. Since the directional vectors do not change in a pure translation, the transformation T will simply be:

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

$$F_{\text{new}} = \begin{bmatrix} 1 & 0 & 0 & d_x \\ 0 & 1 & 0 & d_y \\ 0 & 0 & 1 & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} n_x & o_x & a_x & P_x \\ n_y & o_y & a_y & P_y \\ n_z & o_z & a_z & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} n_x & o_x & a_x & P_x + d_x \\ n_y & o_y & a_y & P_y + d_y \\ n_z & o_z & a_z & P_z + d_z \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

This equation is also symbolically written as:

$$F_{\text{new}} = \text{Trans}\left(d_x, d_y, d_z\right) \times F_{\text{old}}.$$

✓ First, by multiplying the transformation matrix with the frame matrix, the new location can be found. This, in one form or another, is true for all transformations, as we will see later.

✓ Second, you notice that the directional vectors remain the same after a pure translation, but that as vector addition of $\bar{d}$ and $\bar{P}$ would result, the new location of the frame is $\bar{d} + \bar{P}$.

✓ Third, you also notice how homogeneous transformation matrices facilitate the multiplication of matrices, resulting in the same dimensions as before.

Example 9.3.

A frame F has been moved nine units along the x-axis and five units along the z-axis of the reference frame. Find the new location of the frame:

$$F = \begin{bmatrix} 0.527 & -0.574 & 0.628 & 5 \\ 0.369 & 0.819 & 0.439 & 3 \\ -0.766 & 0 & 0.643 & 8 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Solution.

$$F_{\text{new}} = \text{Trans}\left(d_x, d_y, d_z\right) \times F_{\text{old}} = \text{Trans}\ (9, 0, 5) \times F_{\text{old}}$$

and

$$F = \begin{bmatrix} 1 & 0 & 0 & 9 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 5 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 0.527 & -0.574 & 0.628 & 5 \\ 0.369 & 0.819 & 0.439 & 3 \\ -0.766 & 0 & 0.643 & 8 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 0.527 & -0.574 & 0.628 & 14 \\ 0.369 & 0.819 & 0.439 & 3 \\ -0.766 & 0 & 0.643 & 13 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

3.2. Representation of a Pure Rotation about an Axis

To simplify the derivation of rotations about an axis, let's assume that a frame ($\bar{n}$, $\bar{o}, \bar{a}$), located at the origin of the reference frame ($\bar{x}$, $\bar{y}, \bar{z}$), will rotate through an angle of θ about the x-axis of the reference frame. Let's also assume that attached to the rotating frame ($\bar{n}$, $\bar{o}, \bar{a}$), is a point P, with coordinates P_x, P_y and P_z relative to the reference frame and P_n, P_o and P_a relative to the moving frame. As the frame rotates about the x-axis, point P attached to the frame, will also rotate with it. Before rotation, the coordinates of the point in both frames are the same. (Remember that the two frames are at the same location and are parallel to each other). After rotation, the P_n, P_o and P_a coordinates of the point remain the same in the rotating frame ($\bar{n}$, $\bar{o}, \bar{a}$),but P_x, P_y and P_z will be different in the ($\bar{x}$, $\bar{y}, \bar{z}$), frame (Fig. (9.8)). We desire to find the new coordinates of the point relative to the fixed-reference frame after the moving frame has rotated.

Let's look at the same coordinates in 2-D as if we were standing on the x-axis. The coordinates of point P are shown before and after rotation in Fig. (9.9). The coordinates of point P relative to the reference frame are P_x, P_y and P_z while its coordinates relative to the rotating frame (to which the point is attached) remain as P_n, P_o and P_a.

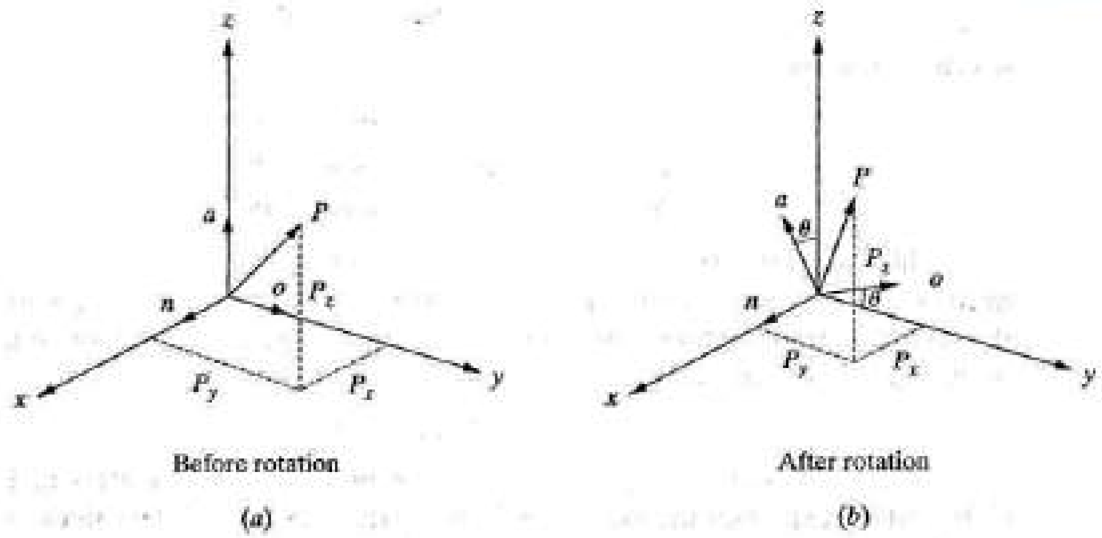


Figure (9.8). **Coordinates of a point in a rotating frame before and after rotation**

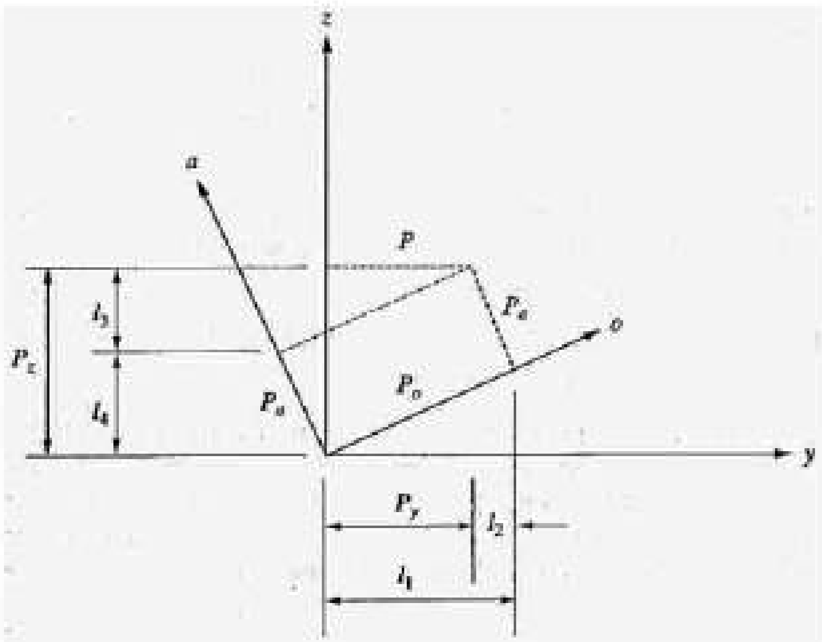


Figure (9.9). **Coordinates of a point relative to the reference frame and rotating frame as viewed from the x-axis**

From Fig. (9.9) the value of P_x does not change as the frame rotates about the x-axis, but the values of P_y and P_z do change. verify that:

$$P_x = P_n,$$

$$P_y = l_1 - l_2 = P_o\cos\theta - P_a\sin\theta,$$

$$P_z = l_3 + l_4 = P_o\sin\theta + P_a\cos\theta,$$

which is in matrix form

$$\begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} P_n \\ P_o \\ P_a \end{bmatrix}.$$

This means that the coordinates of the point (or vector) P in the rotated frame must be multiplied by the rotation matrix, as shown, to get the coordinates in the reference frame. This rotation matrix is only for a pure rotation about the x-axis of the reference frame and is denoted as:

$$P_{xyz} = \mathrm{Rot}(x, \theta) \times P_{noa}.$$

Also notice that the first column of the rotation matrix in Equation, which expresses the location relative to the x-axis, has 1, 0, 0 values, indicating that the coordinate along the x-axis has not changed. Desiring to simplify writing of these matrices, it is customary to designate $C\theta$ to denote $\cos\theta$ and $S\theta$ to denote $\sin\theta$. Thus, the rotation matrix may be also written as:

$$\mathrm{Rot}(x, \theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & C\theta & -S\theta \\ 0 & S\theta & C\theta \end{bmatrix}.$$

Doing the same for the rotation of a frame about the y- and z-axes of the reference frame. Verify that the results are:

$$\mathrm{Rot}(y, \theta) = \begin{bmatrix} C\theta & 0 & S\theta \\ 0 & 1 & 0 \\ -S\theta & 0 & C\theta \end{bmatrix} \quad \text{and} \quad \mathrm{Rot}(z, \theta) = \begin{bmatrix} C\theta & -S\theta & 0 \\ S\theta & C\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Also we can write the equation of rotation $P_{xyz} = \mathrm{Rot}(x, \theta) \times P_{noa}$ in a conventional form, which assists in easily following the relationship between different frames. Denoting the transformation as ${}^{U}T_{R}$ (and reading it as the transformation of frame R relative to frame U (for Universe)), denoting P_{noa} as ${}^{R}P$ *(P relative to frame R),* and denoting P_{xyz} as ${}^{U}P$ *(P* relative to frame *U*), the last Equation simplifies to:

$${}^{U}P = {}^{U}T_{R} \times {}^{R}P.$$

Example 9.4.

A point $P\,(2,3,4)^{T}$ is attached to a rotating frame. The frame rotates 90° about the x-axis of the reference frame. Find the coordinates of the point relative to the reference frame after the rotation, and verify the result graphically.

Solution. Of course, since the point is attached to the rotating frame, the coordinates of the point relative to the rotating frame remain the same after the rotation. The coordinates of the point relative to the reference frame will be

$$\begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & C\theta & -S\theta \\ 0 & S\theta & C\theta \end{bmatrix} \times \begin{bmatrix} P_n \\ P_o \\ P_a \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} 2 \\ 3 \\ 4 \end{bmatrix} = \begin{bmatrix} 2 \\ -4 \\ 3 \end{bmatrix}.$$

As you notice in Fig. (9.10), the coordinates of point P relative to the reference frame after rotation are 2, -4, 3, as obtained by the preceding transformation.

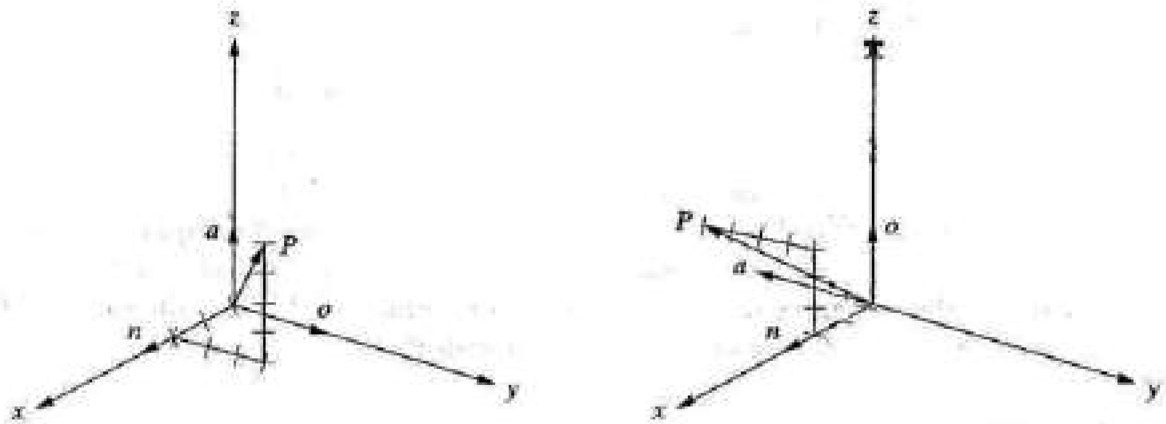


Figure (9.10). **Rotation of a frame relative to the reference frame**

3.3. Representation of Combined Transformations

Combined transformations consist of a number of successive translations and rotations about the fixed reference frame axes. Any transformation can be resolved into a set of translations and rotations in a particular order. For example, rotation a frame about the x-axis, translate about the x-, y-, and z-axes, and then rotate about the y-axis in order to accomplish the transformation that is needed.

To see how combined transformations are handled, let's assume that a frame $(\bar{n}, \bar{o}, \bar{a})$ is subjected to the following three successive transformations relative to the reference frame (x, y, z):

1) rotation of α degrees about the x-axis;

2) followed by a translation of $[l_1, l_2, l_3]$ (relative to the x-, y-, and z-axes, respectively);

3) followed by a rotation of β degrees about the y-axis.

Also, the point P_{noa} is attached to the rotating frame at the origin of the reference frame. As the frame ($\bar{n}$, $\bar{o}$, $\bar{a}$) rotates or translates relative to the reference frame, the point P within the frame moves as well, and the coordinates of the point relative to the reference frame change. After the first transformation, as in the previous section, the coordinates of point P relative to the reference frame can be calculated by

$$P_{1,xyz} = Rot(x,\alpha) \times P_{noa},$$

where $P_{1,xyz}$ is the coordinates of the point after the first transformation relative to the reference frame. The coordinates of the point relative to the reference frame at the conclusion of the second transformation will be:

$$P_{2,xyz} = Trans(l_1, l_2, l_3) \times P_{1,xyz} = Trans(l_1, l_2, l_3) \times Rot\ (x,\alpha) \times P_{noa}.$$

Similarly, after the third transformation, the coordinates of the point relative to the reference frame will be:

$$\begin{aligned} P_{xyz} = P_{3,xyz} &= Rot(y,\beta) \times P_{2,xyz} \\ &= Rot(y,\beta) \times Trans(l_1, l_2, l_3) \times Rot(x,\alpha) \times P_{noa} \end{aligned}$$

Example 9.5

A point $P(7,3,2)^T$ is attached to a frame ($\bar{n}$, $\bar{o}$, $\bar{a}$), and is subjected to the transformations described next. Find the coordinates of the point relative to the reference frame at the conclusion of transformations:

1) rotation of 90° about the z-axis;
2) followed by a rotation of 90° about the y-axis;
3) followed by a translation of [4, -3, 7].

Solution. The matrix equation representing the transformation is:

$$P_{xyz} = Trans(4,-3,7) \times Rot(y,90) \times Rot(z,90) \times P_{noa} =$$

$$\begin{bmatrix} 1 & 0 & 0 & 4 \\ 0 & 1 & 0 & -3 \\ 0 & 0 & 1 & 7 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 0 & -1 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 7 \\ 3 \\ 2 \\ 1 \end{bmatrix} = \begin{bmatrix} 6 \\ 4 \\ 10 \\ 1 \end{bmatrix}.$$

The first transformation of 90° about the z-axis rotates the $(\bar{n}, \bar{o}, \bar{a})$ frame as shown in Fig. (9.11), followed by the second rotation about the y-axis, followed by the translation relative to the reference frame x-, y-, z-axes. The point P in the frame can then be found relative to the $\bar{n}$-, $\bar{o}$-, $\bar{a}$-axes, as shown graphically. The final coordinates of the point can be traced on the x-, y-, z-axes to be $4 + 2 = 6$, $-3 + 7 = 4$, and $7 + 3 = 10$.

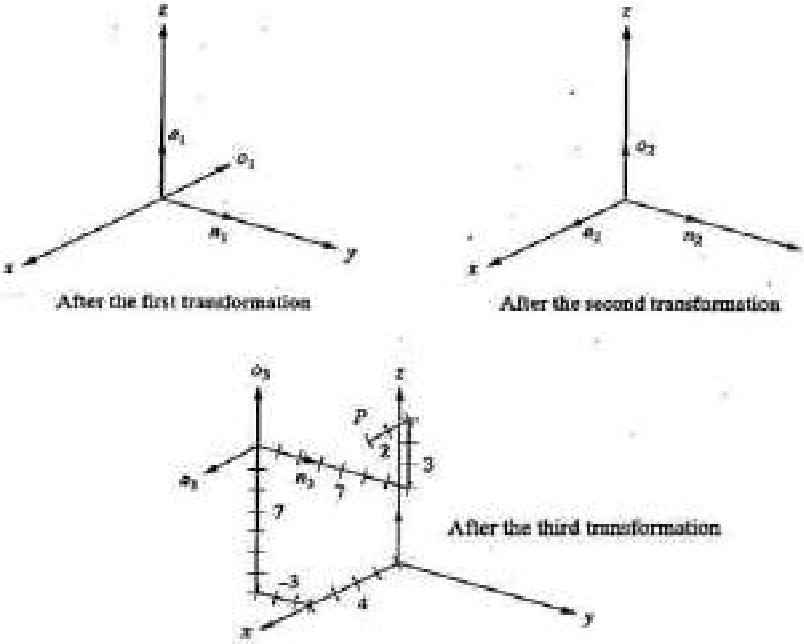


Figure (9.11). **Effects of three successive transformations**

Example 9.6

In this case, assume that the same point $P(7,3,2)^T$ attached to a frame $(\bar{n}, \bar{o}, \bar{a})$ is subjected to the same transformations, but that the transformations are performed in a different order, as shown. Find the coordinates of the point relative to the reference frame at the conclusion of transformations:

1) a rotation of 90° about the z-axis;
2) followed by a translation of [4, -3, 7];
3) followed by a rotation of 90° about the y-axis.

Solution. The matrix equation representing the transformation is:

$$P_{xyz} = Rot(y,90) \times Trans(4,-3,7) \times Rot(z,90) \times P_{noa} =$$

$$\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 & 4 \\ 0 & 1 & 0 & -3 \\ 0 & 0 & 1 & 7 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 7 \\ 3 \\ 2 \\ 1 \end{bmatrix} = \begin{bmatrix} 9 \\ 4 \\ -1 \\ 1 \end{bmatrix}.$$

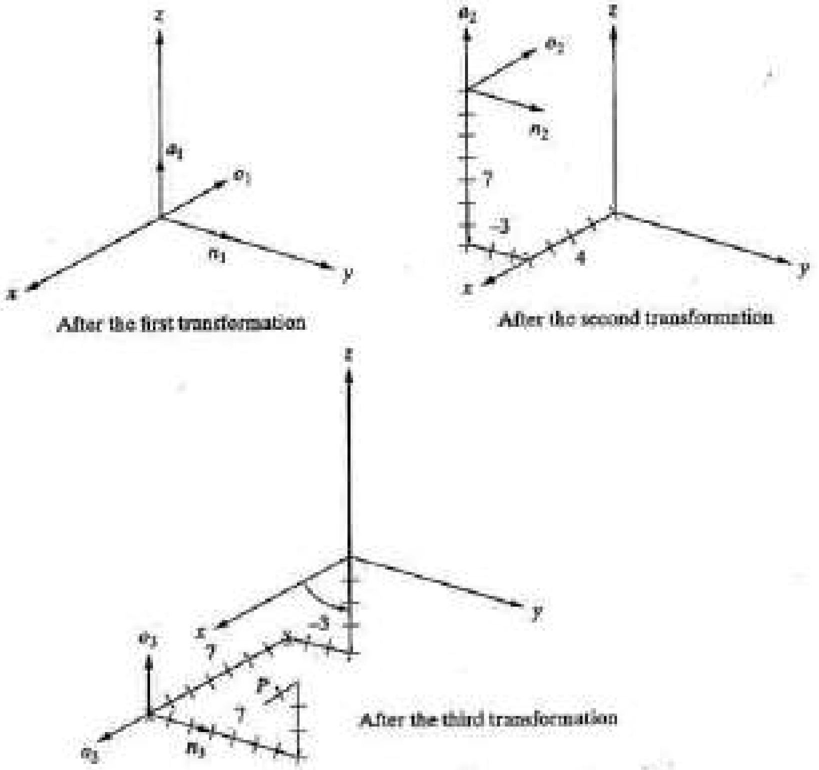


Figure (9.12). **Effects of three successive transformations**

The transformations are exactly the same as in the **previous Example** since the order of transformations is changed, the final coordinates of the point are completely different from the previous example. This can clearly be demonstrated graphically as in Fig. (9.12). In this case although the first transformation creates exactly the same change in the frame, the second transformation's result is very different, as the translation relative to the reference frame axes will move the rotating $(\bar{n}, \bar{o}, \bar{a})$, frame ***outwardly***. As a result of the third transformation, this frame will rotate about the reference frame y-axis, thus rotating ***downwardly***. The location of point P attached to the frame is also shown. The coordinates of this point relative to the reference frame are: 7 + 2 = 9, -3 + 7 = 4 and -4 + 3 = -1, which is the same as the analytical result.

ROBOT KINEMATICS

1. ROBOT ARM KINEMATICS

Robot arm kinematics *is the science that explains the analytical description of the motion geometry of the manipulator with reference to a robot coordinate system fixed to a frame, without the consideration of the forces or the moments causing the movements. The motion is described as a function of time.* For the **Forward Kinematics** the inputs are the joint angle vectors and the link length parameters. The output of the forward kinematics is the orientation and the position of the tool or the gripper. The block diagram representation of the forward kinematics is as given under in Fig. 11.1(a).

In certain situations it is possible to know the position and orientation of the objects placed in the work place or envelope and it is desired to know the joint vectors given the link parameters of the robots. Such a problem is known as the **Inverse Kinematic** problem, represented by the block diagram in Fig. 11.1(b).

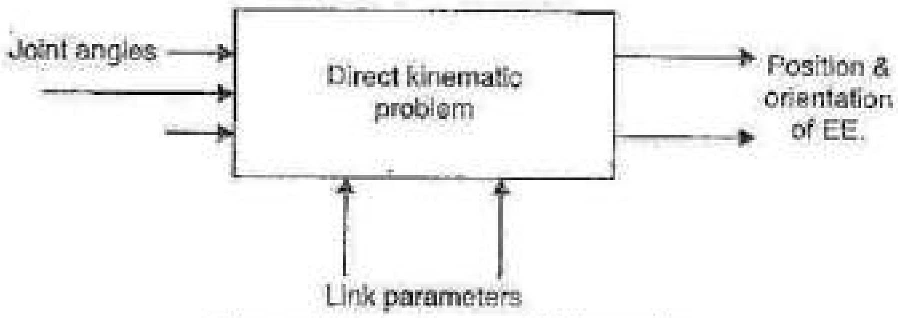


Figure 11.1(a). **Forward kinematics**

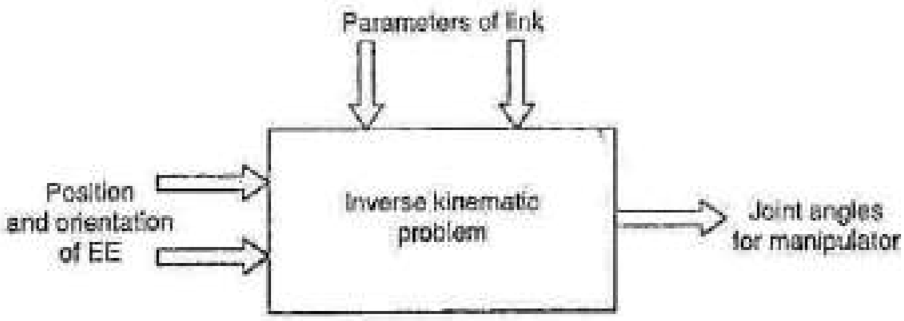


Figure 11.1(b). **Inverse kinematics**

2. MANIPULATOR PARAMETERS

A robot manipulator is a chain of rigid bodies, called links, connected in sequence by joints, known as lower pair joints. The links remain in contact at the joints with two surfaces sliding over one another relatively. There are totally six possible lower pair joints: prismatic (sliding) joint, revolute (rotary) joint, cylindrical, screw, spherical and planar joints. The robot manipulators are generally designed with prismatic joints, revolute joints or together prismatic and revolute joints. In a serial open loop formation each link forms connection, at the most, with two other links. Each pair of a link and a joint contributes single degree of freedom (DOF). N numbers of pairs provides N degrees of freedom for a manipulator. Link 1 forms a joint '0' with the base which establishes an inertial coordinate frame for a dynamic system analysis of an industrial robot. The last link at its free end accommodates a tool or a gripper. Both the base and the gripper are not considered as the part of a robotic manipulator.

In general the link 'k' gets connected at the two ends with link (k – 1) and link (k + 1), forming two joints at the ends of connections. The link is characterized by:

1) The distance (d_i)
2) The angle (θ_i)
3) The length (a_i)
4) The twist angle (α_k)

The manipulator parameters determine structure and relative position of links m the arm Fig. (11.2).

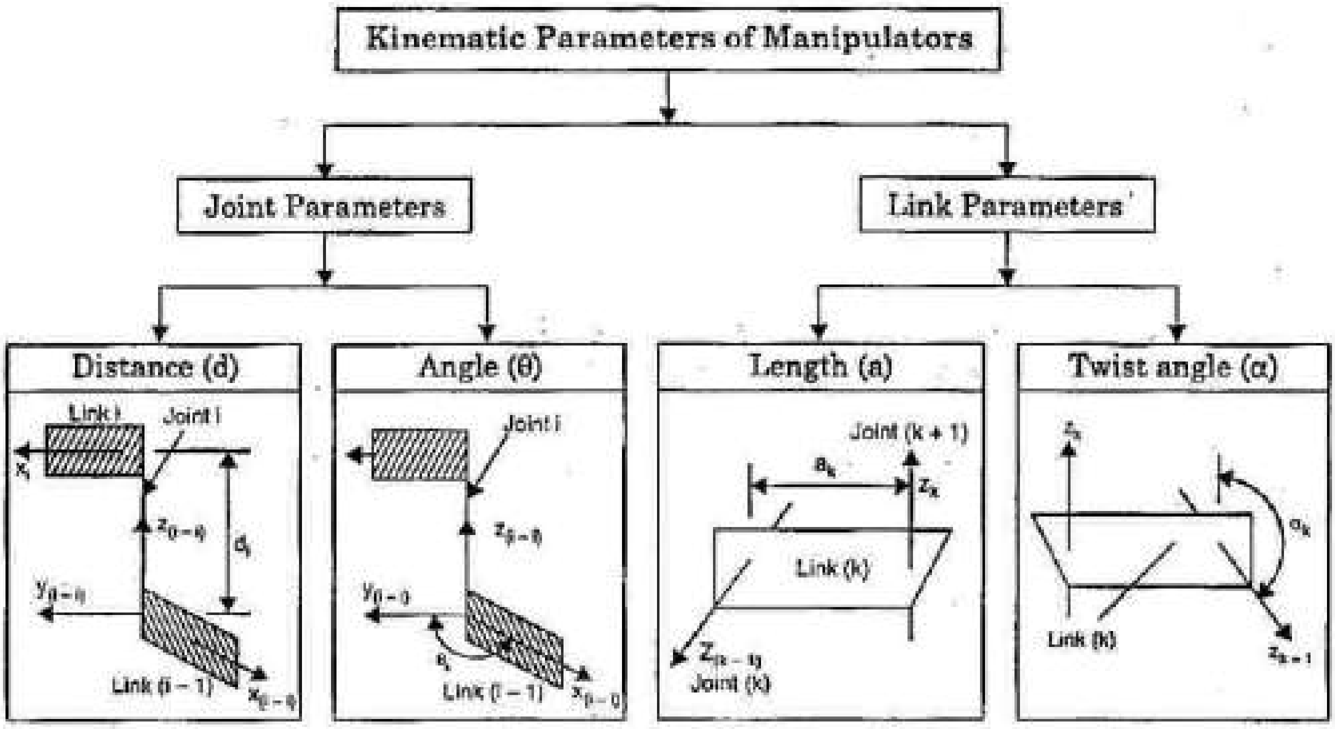


Figure (11.2). **Kinematic Parameters of Joints and Links**

1. THE DENAVIT-HARTENBERG (D-H) REPRESENTATION

In an open kinematic manipulator (Fig. (11.3)) chain of links and joints, each link frame is assigned according the right hand rule coordinate and the manipulator parameters through a systematic procedure proposed by **Denavit** and **Hartenberg**. This procedure of assigning coordinate frame to the links helps in arriving at 4 x 4 homogeneous transformation matrix at each joint with respect to the previous joint or frame leading to the formulation of the kinematic arm equation.

D-H algorithm is the two pass procedure concept. The first pass (Frame Pass) is responsible for the assignment of frames starting from base to the distal end and the second pass (Parameter Pass) computes the kinematic link-joint parameters as shown in tables 11-1 and 11-2.

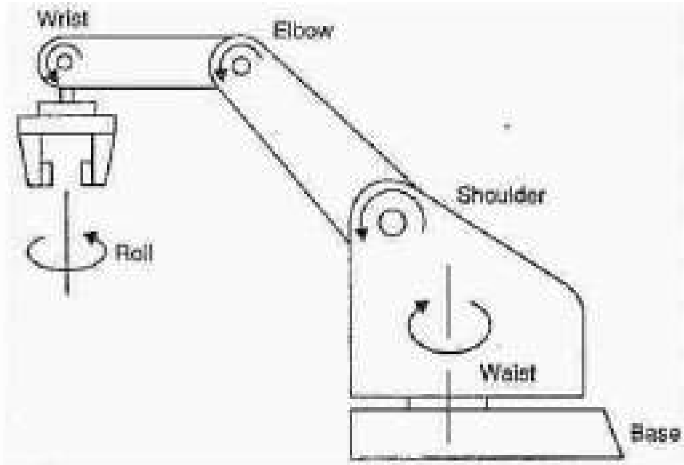
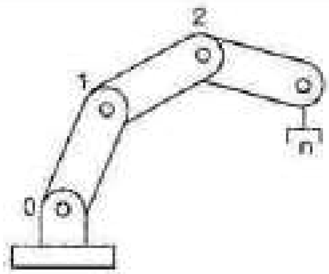
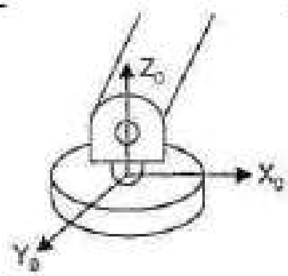
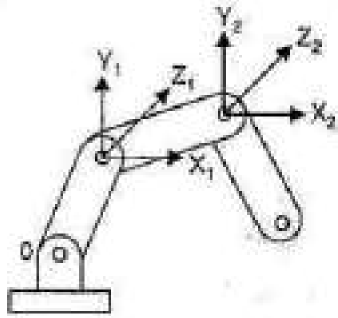


Figure (11.3). **Open kinematic manipulator**

Table 11-1. **Frame Pass**

Step	Assignment	Where to	Figures
1	Number the joints from 0 to *n*.	Start with the base and end with the tool in order.	
2	Assign right hand rule coordinate frame A_0. Z_0 should align with the axis of joint '0'.	To the robot base	
3	Assign Z_i with the other joints.	Along the axis of joints (i+1)	

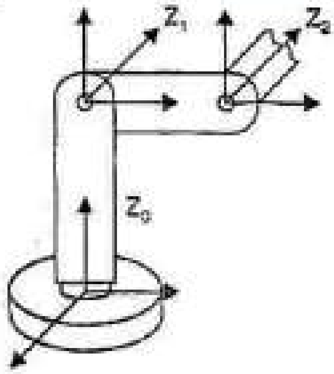
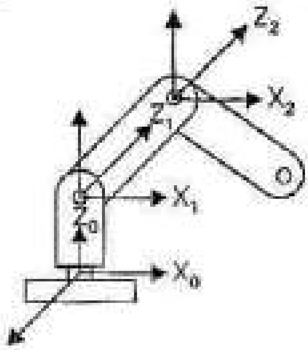
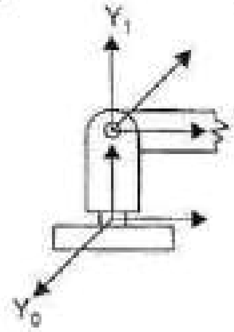
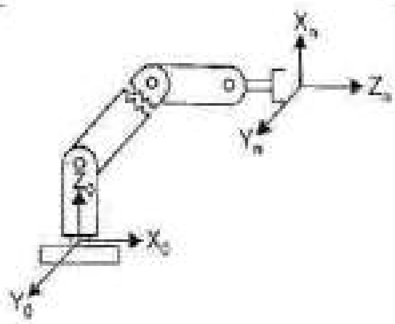
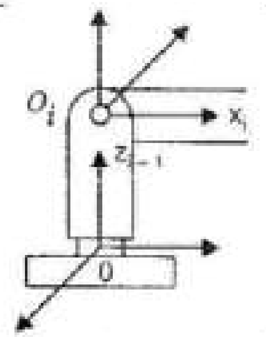
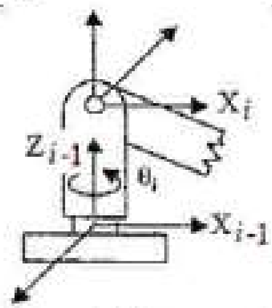
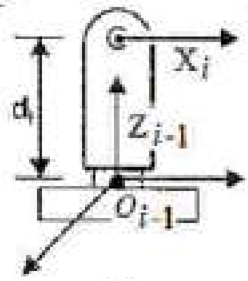
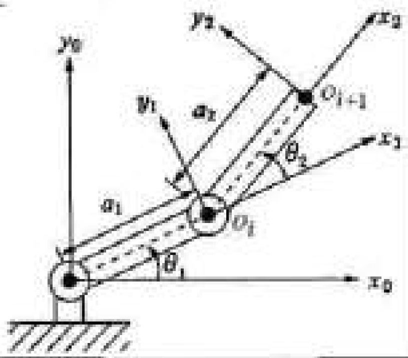
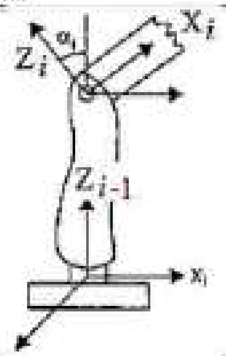
4	Locate the origin of A_i with the joints.	At the intersection of Z_i and Z_{i-1}. If they do not intersect use the meet point of Z_i and the common normal between Z_i and Z_{i-1}.	
5	• Selection of X_i. • If Z_i and Z_{i-1} are parallel.	• Orthogonal to Z_i and Z_{i-1}. • Point X_i away from Z_{i-1}.	
6	Assign y-coordinate axis as Y_i.	So as to form a coordinate frame with right hand rule.	
7	Set $i = i + 1$ and continue	If $i < n$, go to step 3.	
8	Select origin of A_n • Align Z_n with tool approach • Align Y_n with the sliding vector • Align X_n with the normal vector + set $i = 1$	At the tool tip or end-effector tip.	

Table 11-2. **Parameter Pass**

Step	Assignment	Where to	Figures
1	Locating point O_i. If do not intersect	At the intersection of X_i and Z_{i-1} axis. Use common normal between X_i and Z_{i-1}.	
2	Compute θ_i – the angle of rotation of joint *i*.	The angle from of X_{i-1} to X_i measured about Z_{i-1}. If joint *i* is revolute, θ_i is variable.	
3	Compute d_i – distance between the joints.	Distance along Z_{i-1} from O_{i-1} to the intersection of X_i and Z_{i-1} axes. If joint *i* is prismatic, d_i is variable.	
4	Compute a_i – link distance.	Distance along X_i from the intersection of X_i and Z_{i-1} axes to O_i.	
5	Compute α_i – the twist angle of rotation of link	The angle from Z_{i-1} and Z_i measured about X_i.	
6	Set $i = i + 1$	If $i \leq n$, go to step 1.	

3. ARM MATRIX

After assigning coordinate frames to all the links, according to D-H representation, it is possible to establish the relation between successive frames i and $(i+1)$ by the following rotations and translations in sequence.

- Rotation about Z_i by an angle θ_i.
- Translate along Z_i by a distance d_i.
- Translate along rotated $X_i = X_{i+1}$ through length a_i.
- Rotation about X_i by twist angle α_i.

where $i = 0, 1, 2, ..., n$.

This may be expressed as a product of four homogeneous transformations relating coordinate frame of $(i+1)$ link to that of link (i). This relation is known as Arm matrix (A).

$$A_{i+1}^{i} = \mathrm{T}(z,\theta)\ \mathrm{T}_{\mathrm{trans}}(0,0,d)\ \mathrm{T}_{\mathrm{trans}}(a,0,0)\ \mathrm{T}(x,\alpha)$$
$$= \mathrm{T}(z,\theta)\ \mathrm{T}_{\mathrm{trans}}(\alpha,0,d)\ \mathrm{T}(x,\alpha).$$

Hence

$$A_{i+1}^{i} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha & 0 \\ 0 & \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \cos\theta & -\sin\theta\cos\alpha & \sin\theta\sin\alpha & a\cos\theta \\ \sin\theta & \cos\theta\cos\alpha & -\cos\theta\sin\alpha & a\sin\theta \\ 0 & \sin\alpha & \cos\alpha & d \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

For a prismatic joint $\alpha = 0$.

The coordinate frame at the end of the manipulator is related to the base reference frame by the 'T' matrix in terms of (A) matrices, as below:

$$T_6^0 = A_1^0\, A_2^1\, A_3^2\, A_4^3\, A_5^4\, A_6^5.$$

Example 11.1. Planar Elbow Manipulator

Consider the two-link planar arm of Fig. (11.4). Solve to find the forward kinematic transformation matrix?

Solution. The joint axes z_0 and z_1 are normal to the page. The base frame $O_0X_0Y_0Z_0$ as shown. The origin is chosen at the point of intersection of the z_0-axis with the page and the direction of the x_0-axis is completely arbitrary. Once the base frame is established, the $O_1X_1Y_1Z_1$ frame is fixed as shown by the D-H convention, where the origin O_1 has been located at the intersection of Z_1 and the page.

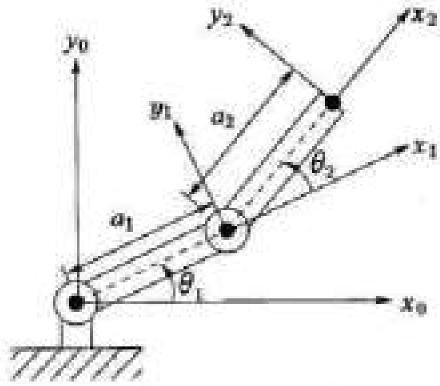


Figure (11.4). **Two-link planar arm**

Table 11.3. **D-H parameters for 2-link planar manipulator**

Link	a_i	α_i	d_i	θ_i
1	a_1	0	0	θ_1^*
2	a_2	0	0	θ_2^*

* variable

The final frame $O_2X_2Y_2Z_2$ is fixed by choosing the origin O_2 at the end of link 2 as shown. The A-matrices are determined from Equation:

$$A_1 = \begin{bmatrix} c_1 & -s_1 & 0 & a_1c_1 \\ s_1 & c_1 & 0 & a_1s_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad A_2 = \begin{bmatrix} c_2 & -s_2 & 0 & a_2c_2 \\ s_2 & c_2 & 0 & a_2s_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The T-matrices are thus given by:

$T_1^0 = A_1$

$$T_2^0 = A_1 A_2 = \begin{bmatrix} c_{12} & -s_{12} & 0 & a_1 c_1 + a_2 c_{12} \\ s_{12} & c_{12} & 0 & a_1 s_1 + a_2 s_{12} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Notice that the first two entries of the last column $T_2^0$of are the x and y components of the origin O_2 in the base frame; that is,

$$x = a_1 c_1 + a_2 c_{12}$$

$$y = a_1 s_1 + a_2 s_{12}$$

are the coordinates of the end-effector in the base frame. The rotational part of T_2^0 gives the orientation of the frame $O_2X_2Y_3Z_4$relative to the base frame, O.

Example 11.2. Three Link Cylindrical Manipulator (Fig. (11.5))

Consider now the three-link cylindrical robot represented symbolically by Fig. 11.5. We establish O_0, O_1, O_2 and O_3 as shown at the joints 1-3. Note that the placement of the origin O_0 along Z_0 and the direction of the x_0-axis are arbitrary. The X_1-axis is parallel to X_0 when $\theta_1 = 0$ but, of course its direction will change since $\theta_1$is variable. Since Z_1 and Z_2 intersect, the origin O_2 is placed at this intersection. The direction of X_2 is chosen parallel to X_1 so that θ_2 is zero. Finally, the third frame is chosen at the end of link 3 as shown.

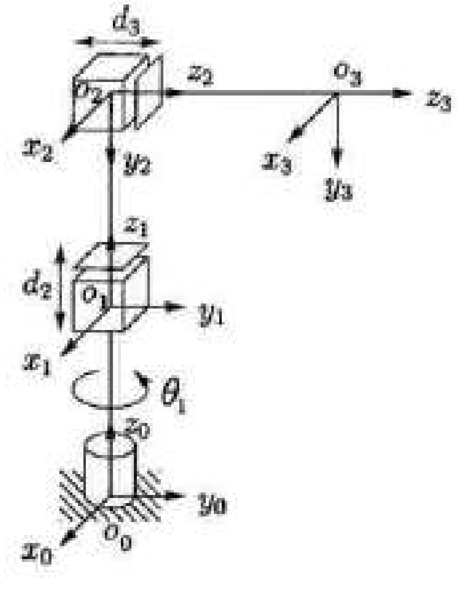


Figure (11.5). **Three-link cylindrical manipulator**

The D-H parameters are shown in Table 11.4. The corresponding *A* and *T* matrices are:

Table 11.4. **D-H parameters for three-link cylindrical robot manipulator**

Link	a_i	α_i	d_i	θ_i
1	0	0	d_1	θ_1^*
2	0	-90	d_2^*	0
3	0	0	d_3^*	0

* variable

$$A_1 = \begin{bmatrix} c_1 & -s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad A_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$A_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

$$T_2^0 = A_1 A_2 A_3 = \begin{bmatrix} c_1 & 0 & -s_1 & -s_1 d_3 \\ s_1 & 0 & c_1 & c_1 d_3 \\ 0 & -1 & 0 & d_1 + d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$