CHAPTER 1 Robotic Manipulation:

UNIT STRUCTURE:

1.1 Automation and Robots

Definition of a robot:

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: (Definition by Robot Institute of America, 1979)

WHAT IS ROBOTICS?

- It is a Gadget which tries to mimics human behavior.
- A ROBOT IS
  - An electromechanical device that is;
  - Reprogrammable
  - Multifunctional
  - Sensible for environment

LAWS OF ROBOTICS: Science fiction writer Isaac Asimov proposed three "Laws of Robotics".

- First Law (Human safety):A human being or, through inaction, allow a human being to come to harm.
- Must obey orders given it by human beings except where such orders would conflict with the First Law.

- Must protect its own existence as long as such protection does not conflict with the First or Second Law.

DEFINTION OF Automation:

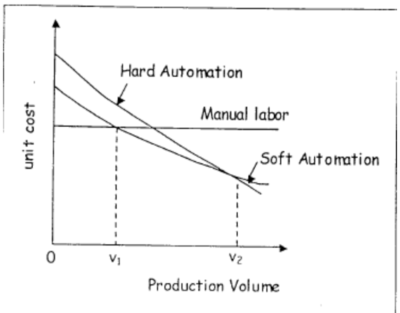
Technology associated with electronic, mechanical and computer based system in their process control operation and production environment.

1.1.1 Hard Automation

- It is a type of fixed automation .
- Periodic modification of the hardware may be required hence called as Hard automation.
- Provides limited flexibility.
- Number of operations performed by the equipment are fixed.
- Useful for mass production manufacturing
- COST: Initial high investment
- Hardware set up useful for one kind of product manufacturing.

1.1.2 Soft Automation

- Preferred when production is high volume
- Different kind of products to be manufactured in a short time period.
- EX: Using computer control robotic manipulators can perform different tasks such as spot welding, spray painting, etc.
- Can be termed as **Programmable automation**
- Preferred when different configuration of products to be designed and manufactured.
- Following figure shows that when robotic systems are less expensive, production volume range volumes [v1, v2] expands continuously at the production end spectrum.



Economic Characteristic

1.2 Classification

1.2.1 Drive Technologies

- **Most popular drive technologies:**

1. **Electrical**
2. **Pneumatic**
3. **Hydraulic**

- Electric: all robots use electricity as the primary source of energy.
  - ✓ Electricity turns the pumps that provide hydraulic and pneumatic pressure.
  - ✓ It also powers the robot controller and all the electronic components and peripheral devices.
  - ✓ In all electric robots, the drive actuators, as well as the controller, are electrically powered.
  - ✓ Because electric robot does not require a hydraulic power unit, they conserve floor space and decrease factory noise.
  - ✓ No energy conversion is required.
- **Pneumatic:** these are generally found in relatively low-cost manipulators with low load carrying capacity.
  - ✓ Pneumatic drives have been used for many years for powering simple stop-to-stop motions.
  - ✓ It is inherently light weight, particularly when operating pressures are moderate.
- **Hydraulic:** are either linear position actuators or a rotary vane configuration.
  - ✓ Hydraulic actuators provide a large amount of power for a given actuator.
  - ✓ The high power-to-weight ratio makes the hydraulic actuator an attractive choice for moving moderate to high loads at reasonable speeds and moderate noise level.
  - ✓ Hydraulic motors usually provide a more efficient way of energy to achieve a better performance, but they are expensive and generally less accurate.

1.2.2 Work Envelope Geometries

1. **Cartesian coordinate robot**
2. **Cylindrical coordinate robot**
3. **Spherical coordinate robot**

4. Articulate coordinate robot

WORK ENVELOPE

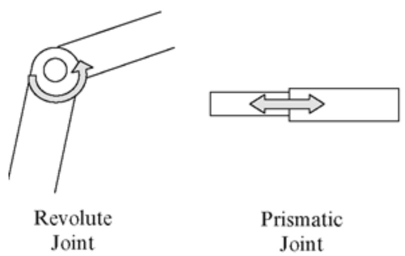
- The volume in space that a robot's end-effector can reach, both in position and orientation.
- In Cartesian, the endpoint of the arm is capable of operating in a cuboidal space, called workspace.

WORKSPACE

- Represents the portion of space around the base of the manipulator that can be accessed by the arm endpoint.

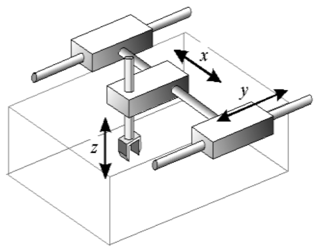
Axes	Type	Function
1-3	Major	Position the wrist
4-6	Minor	Orient the tool

TYPES OF ROBOT JOINTS:

TYPE	**NOTATION**	**Description**	**Examples**
Revolute	**R**	**Rotary motion about an axis**	
Prismatic	**P**	**Translate /Linear/Sliding motion along an axis**	

1. Cartesian coordinate robot

Cartesian Robot

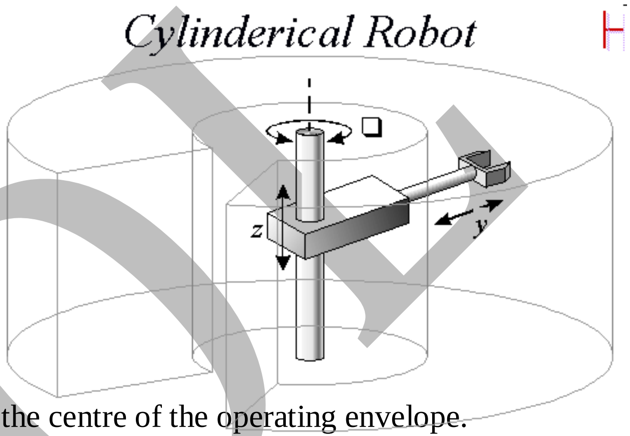


- The first type of robot is called the Cartesian robot.
- This type of robot uses the X, Y, Z three dimensional coordinate system to control movement and location.
- PPP

- **Gantry Robot**

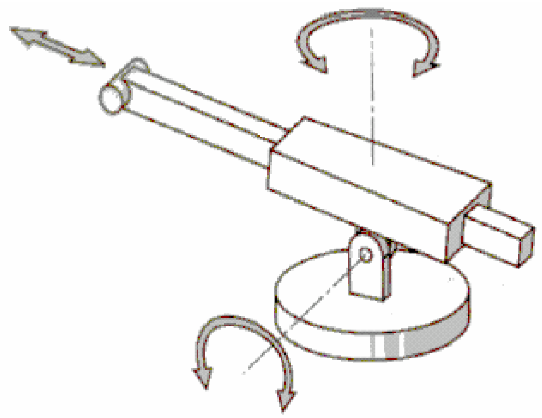
- In a gantry robot, each of these motions
    - are arranged to be perpendicular to each other and
    - are typically labeled X, Y, and Z.
- X and Y are located in the horizontal plane and Z is vertical.
- A gantry robot can move things anywhere within this envelope or perform some **operation on an item within the envelope.**

2. Cylindrical coordinate robot



Cylindricalrobots have a main axis that is in the centre of the operating envelope.

- It can reach into tight areas without sacrificing speed or repeatability.
- RPP
- If the first joint of a Cartesian- coordinate robot is replaced with a revolute joint to form the configuration RPP, this produces a cylindrical-coordinate robot.
- The revolute joint swings the arm back and forth about a vertical base axis.
- The prismatic joints then move the wrist up and down along the vertical axis and in and out along a radial axis.



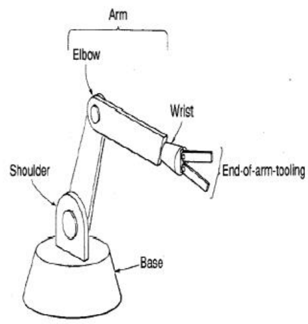
3. Spherical coordinate robot

Spherical or polar robots are similar to a cylindrical robot, but form a spherical range of motion using a polar coordinate system.

- RRP
- The first revolute joint swings the arm back and forth about a base axis that can also be thought of as a vertical shoulder axis.

- The second revolute joint swings the forearm back and forth about a vertical elbow axis.
- Thus the two revolute joints control motion in a horizontal plane.
- The vertical component of the motion is provided by the third joint, a prismatic joint which slides the wrist up and down.

Articulate coordinate robot



- Articulated arm robots have at least three rotary joints. They are frequently called an anthropomorphic arm because they closely resemble a human arm.
- Note: 3 degrees of freedom are necessary for position (x, y, z) and 3 degrees of freedom are necessary for orientation (α, β, γ).

1.2.3 Motion Control Methods

Point-to-point: these robots are most common and can move from one specified point to another but cannot stop at arbitrary points not previously designated.

Controlled path: is a specialized control method that is a part of general category of a point-to-point robot but with more precise control. The controlled path robot ensures that the robot will describe the right segment between two taught points. Controlled-path is a calculated method and is desired when the manipulator must move in the perfect path motion.

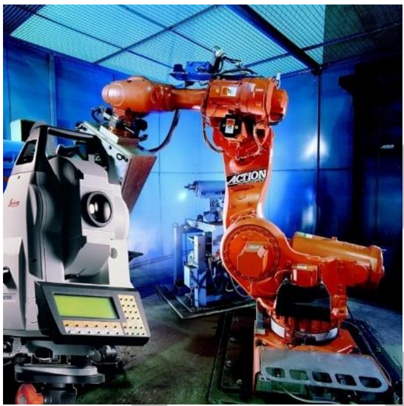
Continuous path: is an extension of the point-to-point method. This involves the utilization of more points and its path can be arc, a circle, or a straight line. Because of the large number of points, the robot is capable of producing smooth movements that give the appearance of continuous or contour movement.

1.3 Application

Robots in Automobile Industries

In the automobile industry, robotic arms are used in diverse manufacturing processes including assembly, spot welding, arc welding, machine tending, part transfer, laser processing, cutting, grinding, polishing, deburring, testing, painting and dispensing.

Robots have proved to help automakers to be more agile, flexible and to reduce production lead times.Robots are ideal for doing precise, repetitive or dangerous tasks. Around 90% of robots are used in factories with half of these being used in the automobile industry.



1.4 Specifications

SPEED

- Speed is the amount of distance per unit time at which the robot can move, usually specified in inches per second or meters per second.
- The speed is usually specified at a specific load or assuming that the robot is carrying a fixed weight.
- Actual speed may vary depending upon the weight carried by the robot.

Load Bearing Capacity

- Load bearing capacity is the maximum weight-carrying capacity of the robot.
- Robots that carry large weights, but must still be precise are expensive.

Accuracy

- Accuracy is the ability of a robot to go to the specified position without making a mistake.
- It is impossible to position a machine exactly.
- Accuracy is therefore defined as the ability of the robot to position itself to the desired location with the minimal error (usually 0.001 inch).

REPEATABILITY

- Repeatability is the ability of a robot to repeatedly position itself when asked to perform a task multiple times.
- Accuracy is an absolute concept, repeatability is relative.
- Note that a robot that is repeatable may not be very accurate.
- Likewise, an accurate robot may not be repeatable.

PRECISION

- Precision is the 'fineness' with which a sensor can report a value.
- For example, a sensor that reads 2.1178 is more precise than a sensor that reads 2.1 for the same physical variable.
- Precision is related to significant figures.
- The number of significant figures is limited to the least precise number in a system of sensing or string of calculations.

Number of Axes

Each robotic manipulator has number of axes about which its links rotate or alongwhich its links translates.

The Major axes determine the shape of work envelope. The Minor axes determinethe arbitrary orientation of the tool in 3D space. The Mechanism for activating thetool is not regarded as independent axis, because it does not contribute to either theposition or the orientation of the tool. The Redundant axes are useful for reachingaround obstacles in the workspace or avoiding undesirable geometricalconfigurations of the manipulator.

Load Bearing Capacity

The maximum weight-carrying capacity of the robot.

Repeatability, Accuracy & Precision

• Accuracy: The measure of the ability of a robot to place the tool tip at anarbitrarily prescribed location in the work envelope.

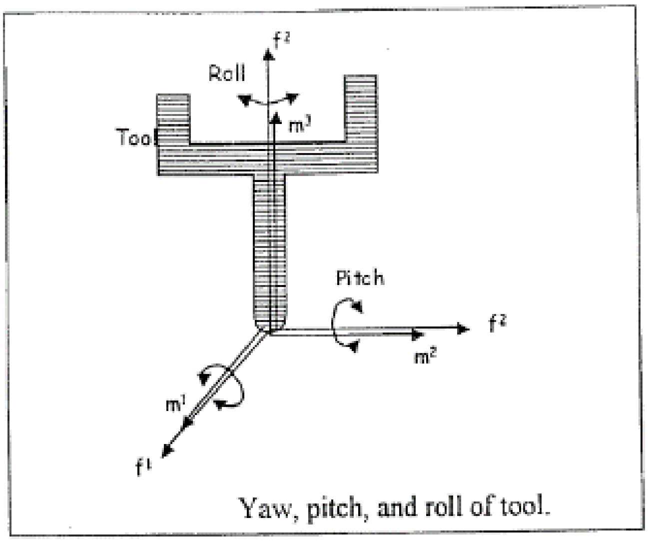
• Repeatability: The measure of the ability of the robot to position the tool tipthe same place repeatedly.

• Precision: The measure of the spatial resolution with which the tool can positioned within the work envelope.

Tool Orientation

Position: The translational (straight-line) location of something.

Orientation: The rotational (angle) location of something. A robot's orientation is



Yaw, pitch, and roll of tool.

measured by roll, pitch, and yaw angles.

To specify the tool orientation, a mobile coordinate frame M= {m1, m2, m3} is attached to the tool and moves with the tool. Initially,the mobile tool frame M starts out coincident with a fixed wrist coordinate frame F {f1, f2, f3}

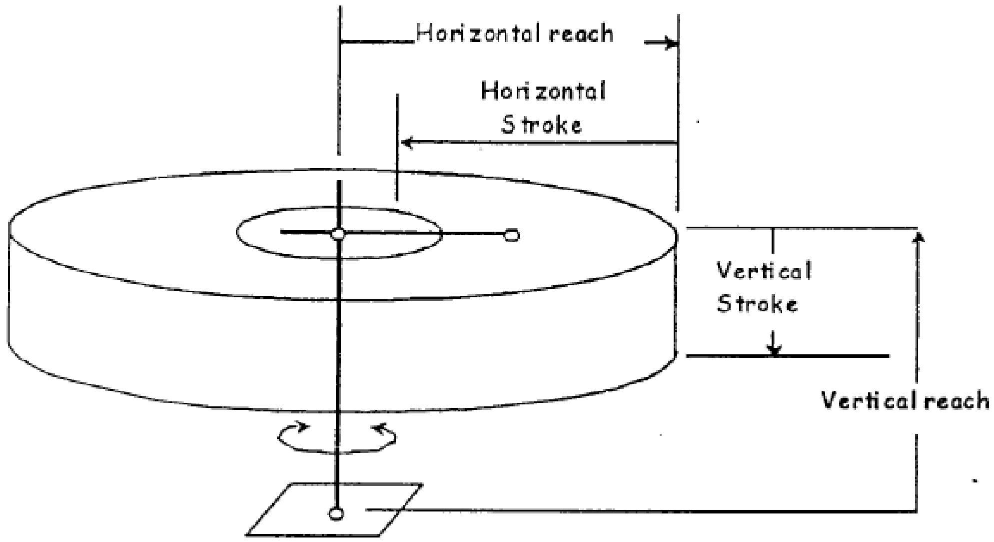
Reach and Stroke

The horizontal reach maximum radial distance be positioned from the vertical axis about which the robot rotates.

• The horizontal stroke defined as the total radial distance the wrist can travel.

• The vertical reach maximum elevation above the work surface that the tool can reach.

The vertical stroke is defined as the total vertical distance the wrist can travel.



Reach and stroke of a cylindrical robot

1.5 Notations

Vectors:

The following is a column vector arranged in an n X 1 array represented in square brackets.

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}.$$

T superscript: Transpose of a vector, matrix

x= [x_1, x_2, x_3] T

Sets: Curly brackets or braces are used to enclose the components of vectors and matrices to enclose members of a set.

$\Gamma = \{x^{1,} x^2, \ldots\ldots., x^{n\}}$

Matrices:

Represented by a 2 dimensional array with the scalar components as shown:

A= A_{11} A_{12}

A_{21} A_{22}

Coordinate Transformation:

Coordinate frames are associated with the different parts of the robot arm, sensors and objects in the workspace. RH orthonormal coordinate frame are associated with the links of the robot, begins from 0 i.e. the base and ends with the link n, i.e. the tool as given

$L_k = \{xk, y_k, z_k\}$ $0 \leq k \leq n$

- ✓ Column vectors are denoted with lowercase letters.
- ✓ Matrices and sets are denoted with uppercase letters.
- ✓ Single subscripts denote scalar components of column vectors
- ✓ Double subscripts denote scalar components of matrices.
- ✓ In the case of matrices:
- ✓ The first subscript is the index of the row
- ✓ The second subscript is the index of the column.

Entity	**Notation**	**Examples**
Scalars	Subscripted	a_1, a_2, α_1, α_2, A_{11}, A_{12}
Column vectors	Lowercase	a, b, c, α, β ,γ , x^1 , x^2
Matrices, sets	Uppercase	A, B, C, D, Γ, Ω, Ψ, Y

REFERENCES:

1.Robert Shilling, “Fundamentals of Robotics-Analysis and control”, PHI.

2. Fu, Gonzales and Lee, “Robotics”, McGraw Hill

3. J.J, Craig, “Introduction to Robotics”, Pearson Education

CHAPTER 2 Direct Kinematics:

UNIT STRUCTURE:

INTRODUCTION:

Given vector of joint variables of a robotic manipulator, determine the position and orientation of the tool with respect to a coordinate frame attached to the robot base.

2.1 Dot and cross products

Dot Product

Two vectors x and y in R^n is denoted

$x.\ y \triangleq \sum_{k=1}^{n} x_k y_k$

Following are the properties of dot product

1. $x \cdot x \geq 0$
2. $x \cdot x = 0 \Leftrightarrow x = 0$
3. $x \cdot y = y \cdot x$
4. $(\alpha x + \beta y) \cdot z = \alpha(x \cdot z) + \beta(y \cdot z)$

Orthogonality

Two vectors x and y in R^n are orthogonal if and only if x.y=0

Completeness:

An orthogonal set of vectors {x1, x2, x3,Xn} in R^n is complete if and only if

$y.x^k = 0$ for $1 \leq k \leq n \quad => \quad y = 0$

Norm:Norm of a vector x in R^n is ||x||, || x || $\triangleq (x\ .\ y)^{1/2} = (\sum_{k=1}^{n} x_k^2)^{1/2}$

Dot product

Let x and y be the arbitrary vectors in R ^{3} and let theta be the angle from x and y

$$\|u \times v\| = \|u\| \, \|v\| \sin \theta$$

Cross Product:

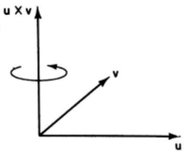


Figure 2-3 Cross product of *u* with *v*.

$$\|u \times v\| = \|u\| \, \|v\| \sin \theta$$

$$w \triangleq \det \left\{ \begin{bmatrix} i^1 & i^2 & i^3 \\ u_1 & u_2 & u_3 \\ v_1 & v_2 & v_3 \end{bmatrix} \right\} = \begin{bmatrix} u_2 v_3 - u_3 v_2 \\ u_3 v_1 - u_1 v_3 \\ u_1 v_2 - u_2 v_1 \end{bmatrix}$$

2.2 Co-ordinate frames

Let p be a vector in Rn

Let X = {x1,x2,x3,….xn} be a complete orthonormal set for Rn.

The coordinates of p w.r.t X are denoted as $[p]^X$

Can be defined as:

$$p = \sum_{k=1}^{n} [p]_k^X x^k$$

Coordinate Transformation:

F={f1,f2,f3,…fn}

M={m1,m2,m3,….mn}

F and M are coordinate frames for Rn

$$[P]_k^F = P.f^k \left(k^{th} \text{ coordinate of } P \text{ w.r.t. frame } F\right)$$

$$= \left(\sum_{j=1}^{n} [P]_j^M m^j \right) \cdot f^k$$

$$= \sum_{j-1}^{n} [P]_j^M \left(m^j . f^k \right)$$

$$= \sum_{j=1}^{n} \left(m^j . f^k \right) [P]_j^m$$

$$= \sum_{j=1}^{n} \left(f^k . m^j \right) [P]_j^m$$

$$= \sum_{j=1}^{n} A_{kj} [P]_j^m$$

$$\therefore [P]^F = A [P]^m$$

Inverse Coordinate Transformation:

F and M = orthonormal coordinate frames in Rn

A= CTM maps M coordinates into F coordinates

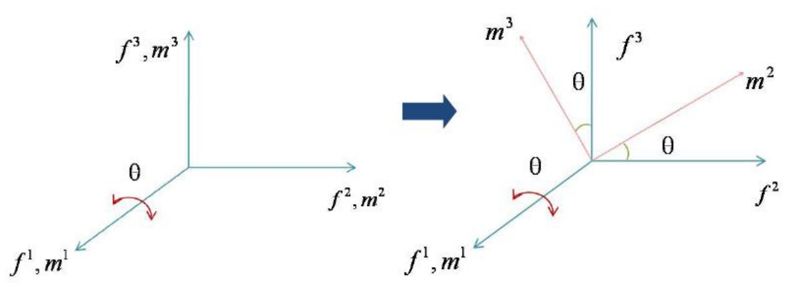
$A^{-1} = A^T$

$$\begin{aligned}(A^{-1})_{kj} &= m^k \cdot f^j \\ &= f^j \cdot m^k \\ &= A_{jk} \\ &= (A^T)_{kj}\end{aligned}$$

2.3 Rotations

2.3.1 Fundamental Rotations

A fixed frame 'f' is attached to the base of the robot, where as a mobile co-ordinate frame is attached to the tool.



The Rotation is represented by a 3x3 matrix

$$R_k(\theta)=\begin{bmatrix} f^1.m^1 & f^1.m^2 & f^1.m^3 \\ f^2.m^1 & f^2.m^2 & f^2.m^3 \\ f^3.m^1 & f^3.m^2 & f^3.m^3 \end{bmatrix}$$

Rotation about 1st axis

$$R_1(\theta)=\begin{bmatrix} 1 & 0 & 0 \\ 0 & Cos\theta & -Sin\theta \\ 0 & Sin\theta & Cos\theta \end{bmatrix}$$

Rotation about 2nd axis

$$R_2(\theta)=\begin{bmatrix} Cos\theta & 0 & Sin\theta \\ 0 & 1 & 0 \\ -Sin\theta & 0 & Cos\theta \end{bmatrix}$$

Rotation about 3rd axis

$$R_3(\theta)=\begin{bmatrix} Cos\theta & -Sin\theta & 0 \\ Sin\theta & Cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

2.3.2 Composite Rotations

Multiplication of the number of fundamental rotation matrices together and the product which contains a sequence of rotations about the unit vectors, such type of multiple rotations is called as composite rotations.

$$YPR(\theta)_{fixed} = RPY(\theta)_{mobile} = R_3(\theta_3)R_2(\theta_2)R_1(\theta_1) =$$

$$\begin{bmatrix} C\theta_3 & -S\theta_3 & 0 \\ S\theta_3 & C\theta_3 & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} C\theta_2 & 0 & S\theta_2 \\ 0 & 1 & 0 \\ -S\theta_2 & 0 & C\theta_2 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 \\ 0 & C\theta_1 & -S\theta_1 \\ 0 & S\theta_1 & C\theta_1 \end{bmatrix}$$

$$YPR(\theta) = R_3(\theta_3)\,R_2(\theta_2)\,R_1(\theta_1)$$

$$= \begin{bmatrix} C_3 & -S_3 & 0 \\ S_3 & C_3 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} C_2 & 0 & S_2 \\ 0 & 1 & 0 \\ -S_2 & 0 & C_2 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & C_1 & -S_1 \\ 0 & S_1 & C_1 \end{bmatrix}$$

$$= \begin{bmatrix} C_3 & -S_3 & 0 \\ S_3 & C_3 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} C_2 & S_1S_2 & C_1S_2 \\ 0 & C_1 & -S_1 \\ -S_2 & S_1C_2 & C_1C_2 \end{bmatrix}$$

$$= \begin{bmatrix} C_2C_3 & S_1S_2C_3 - C_1S_3 & C_1S_2C_3 + S_1S_3 \\ C_2S_3 & S_1S_2S_3 + C_1C_3 & C_1S_2S_3 - S_1C_3 \\ -S_2 & S_1C_2 & C_1C_2 \end{bmatrix}$$

Equivalent Angle-Axis:

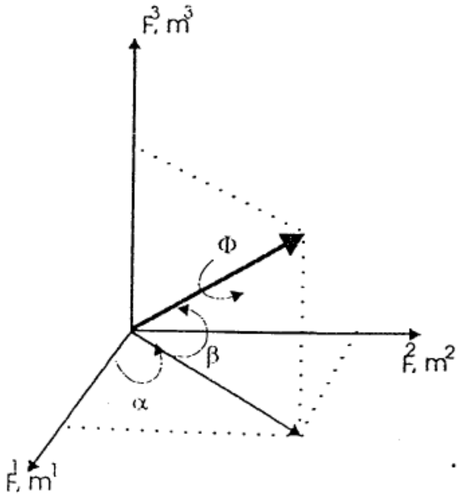
F , M= 2 orthonormal coordinate frames in R3

M initially coincident with F

U= unit vector

M is rotated about u by an angle ϕ The equivalent angle axis rotation matrix R(ϕ ,u)

In which M maps coordinates into F coordinates is



$$R(\phi, u) = \begin{bmatrix} u_1^2 V\phi + C\phi & u_1 u \\ u_1 u_2 V\phi + u_3 S\phi & u \\ u_1 u_3 V\phi - u_2 S\phi & u_2 u \end{bmatrix}$$

2.4 Homogeneous Coordinates

In 3-D space , a physical point is located and if we want to change from one coordinate to another frame then we need to use 4 X 4 homogenous transformation matrix.

- It consists of 4 sub-matrices as given below:
- 3 X 3 sub - matrix R is a rotation matrix
- 3 X 1 column vector p is a translation matrix
- 1 X 3 $\eta^T$is a perspective vector
- Scalar σ is a non-zero scale factor set to unity.

$$T = \begin{bmatrix} R & P \\ \eta & \sigma \end{bmatrix}$$

Translations and Rotations:

Consider the two orthonormal coordinate frames F and M are initially coinciding with each other.

Case11: Translate the mobile coordinate frame by a distance p_k along the kth unit vector of F. In terms of homogenous coordinates, we obtain the following 4 X 4 matrix called as the fundamental homogenous translation matrix.

$$\text{Tran}(p) \triangleq \left[\begin{array}{ccc:c} 1 & 0 & 0 & p_1 \\ 0 & 1 & 0 & p_2 \\ 0 & 0 & 1 & p_3 \\ \hdashline 0 & 0 & 0 & 1 \end{array}\right]$$

Case11: Rotate the mobile coordinate frame by an amount angle ϕ about the kth unit vector of F. In terms of homogenous coordinates, we obtain the following 4 X 4 matrix called as the fundamental homogenous rotation matrix.

$$\text{Rot}(\phi, k) \triangleq \left[\begin{array}{ccc:c} & & & 0 \\ & R_k(\phi) & & 0 \\ & & & 0 \\ \hdashline 0 & 0 & 0 & 1 \end{array}\right]$$

Composite Homogenous Transformations

A Homogeneous transformation matrix represents both a rotation and a translation of the mobile frame with respect to the fixed frame. A product of fundamental homogenous transformation matrices is a sequence of individual rotations and translations. But the order as well as the axis of rotation (F or M) is important since AB is not equal to BA in matrix multiplication. Hence the following algorithmis used:

1. Initialize the transformation matrix to T = I, which corresponds to the orthonormal coordinate frame

2. F and M being coincident. (No rotation and no translation)
3. Represent rotations and translations using separate homogeneous transformation matrices.
4. Represent composite rotations as separate fundamental homogeneous rotation matrices.
5. If the mobile coordinate frame M is to be rotated about or translated along a unit vector of fixed coordinate frame F, pre-multiply (multiplication on left)
6. If the mobile co-ordinate frame M is to be rotated about or translated along one of its own unit vectors,past-multiply (multiplication on right).
7. If there are more fundamental rotationsor translations to be performed, go to step iv; else stop.

INVERSE HOMOGENOUS TRANSFORMATION:

$$T^{-1} = \left[\begin{array}{c|c} R^T & -R^T P \\ \hline 0\ 0\ 0 & 1 \end{array}\right]$$

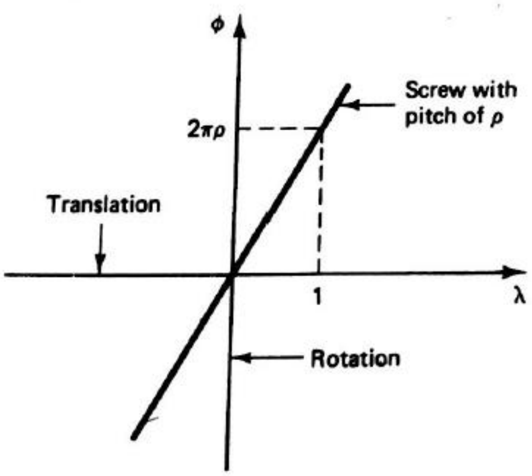
Screw Transformations:

The threading and the unthre3ading operation is called as a screw transformation in which there is a linear displacement along an axis in combination with an angular displacement about the same axis.

$$\text{Screw}(\lambda, \phi, k) \triangleq \text{Rot}(\phi, k)\,\text{Tran}(\lambda i^k) \triangleq \text{Tran}(\lambda i^k)\,\text{Rot}(\phi, k)$$

Pitch of the screw $p \triangleq \frac{\phi}{2\pi\lambda}$ threads per unit length

- For a **pure rotation**, the linear displacement $\lambda = 0$, hence pure rotation is a **screw with infinite pitch**.
- For a **pure translation,** the angular rotation $\phi = 0$; hence pure translation is **a screw with zero Pitch**



2.5 Co-ordinates (Kinematic Parameters)

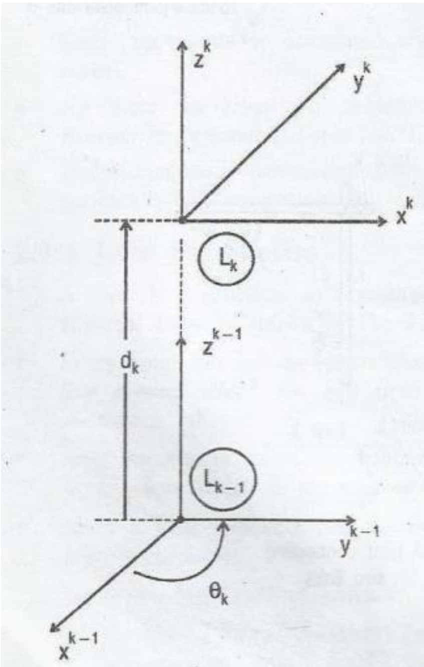
Link and Joint Parameters

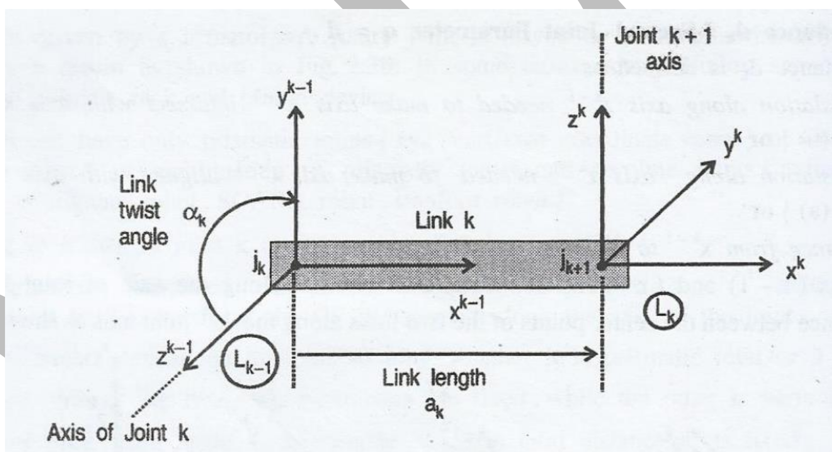
• Joint angle: the angle of rotation from the Xi-1 axis to theXiaxis about the Zi-1 axis. It is the joint variable if joint i isrotary.

• Joint distance : the distance from the origin of the (i-1)coordinate system to the intersection of the Zi-1 axis andthe Xi axis along the Zi-1 axis. It is the joint variable if joint I is prismatic.

• Link length : the distance from the intersection of the Zi-1axis and the Xi axis to the origin of the ith coordinatesystem along the Xi axis.

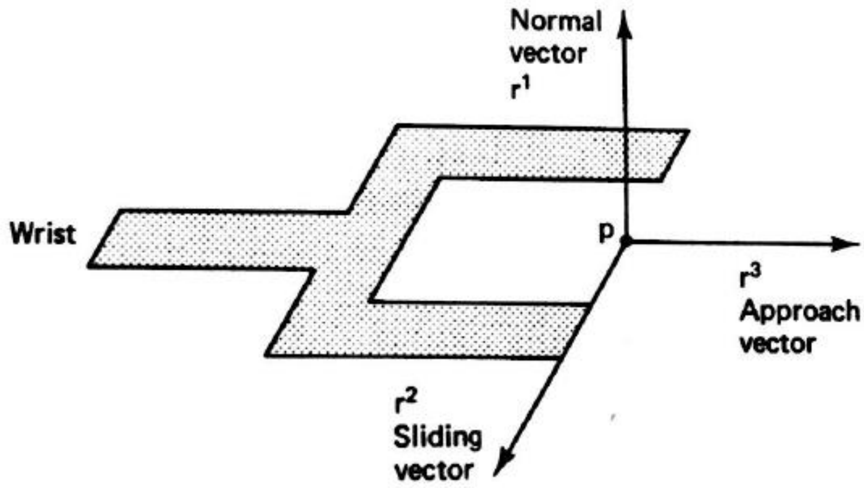
• Link twist angle : the angle of rotation from the Zi-1 axis tothe Ziaxis about the Xi axis.





Arm Parameter	Symbol	Revolute Joint (R)	Prismatic Joint (P)
Joint angle	θ	Variable	Fixed
Joint distance	d	Fixed	Variable
Link Length	a	Fixed	Fixed
Link twist angle	α	Fixed	Fixed

Normal, Sliding and Approach vectors:



- In rectangular coordinates the orientation of the tool is expressed by a Rotation matrix R ={ r1,r2,r3}
- 3 columns of R coordinates correspond to the normal, sliding and approach as shown in the above diagram.
- r3= approach vector i.e. it is aligned along with the Tool Roll axis and always point away from the wrist.
- r2= sliding vector i.e. orthogonal to the approach vector, align to the opening and closing of the gripper.
- r1= normal vector i.e. orthogonal to the plane defined by the r3 and r2 and to complete Right –handed Orthonormal coordinate frame.

2.6 Link co-ordination arm equation

2.6.1 D-H algorithm

0. Number thejoints from 1 ton starting with the base and ending with the tool Yaw, pitch and roll in that order.

1. Assign a right handed orthonormal coordinate frame L0 to the robot base, making sure that z^0 aligns with the axis of joint 1. Set k = 1.

2. Align i with the axis of joint k+l.

3. Locate the origin of L, at the intersection of the i and i·1 axes. If they donot intersect, use the intersection of zk with a common normal between i and i·1•

4. Select x" to be orthogonal to both i and zk-i _ If i and i-1 are parallel, pointxkaway from z^{k-1}

5. Select y^k to form a right handed orthonormal coordinate frame Lk.

6. Set k = k + 1. If k < n, goto step 2; else continue.

7. Set the origin of L, at the tool tip. Align zn with the approach vector, y with sliding vector, and x"with the normal vector of the tool. Set k = 1.

8. Locate point bk at the intersection of the xk and z^{k-1} axes. If they do not intersect, use the intersectionof x" with a common normal between xk and i·1.

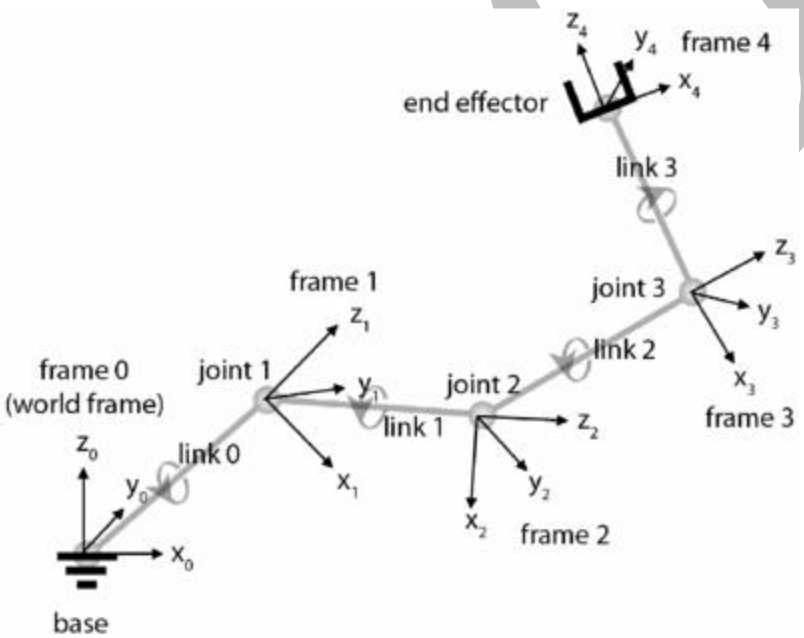
9. Compute as the angle of rotation from xk-l toxk measure dba out zk·l .

10. Compute d^k as the distance from the origin of frame L^{k-1} to point bk measured along i·1•

11. Compute a^k as the distance from point bk to the origin of frame Lk measured alongx^k.

12. Compute a^kas the angle of rotation from z^{k-1} toz^k measured about x",

13. Set k = k + I. lf k s n, go to step 8; else, stop.



2.6.2 ARM Matrix

In order to construct the homogenous transformation matrix we require to map the frame of the coordinates k into the frame of the coordinates k-1 and this steps are related to the four kinematic parameters given in the table as shown below:

Operation	Description
1	Rotate L_{k-1} about z^{k-1} by θ_k
2	Translate L_{k-1} along z^{k-1} by dk
3	Translate L_{k-1} along x^k by a_k
4	Rotate L_{k-1} about x_k by α_k

Operation 1: To make x^{k-1} with || to x^k

Operation 2: x^{k-1} is aligned to x^k

Operation 3: origins of Lk-1 and Lk coincides with each other

Operation 4:axis z^{k-1} aligns with axis z^k

Operations 1 and 2 = gives screw transformation along the axis z^{k-1}

Operations 3 and 4 = gives screw transformation along the axis x^{k-1}

Composite Homogenous Transformation is the combination of 2 screw transformation :

$$T_{k-1}^{k}(\theta_k, d_k, a_k, \alpha_k) = \text{screw}(d_k, \theta_k, 3)\ \text{screw}(a_k, \alpha_k, 1)$$

Link Coordinate Transformation matrix

$$[q]^{k-1} = T_{k-1}^{k}[q]^{k}$$

$$T_{k-1}^{k} = \left[\begin{array}{ccc|c} C\theta_k & -C\alpha_k S\theta_k & S\alpha_k S\theta_k & a_k C\theta_k \\ S\theta_k & C\alpha_k C\theta_k & -S\alpha_k C\theta_k & a_k S\theta_k \\ 0 & S\alpha_k & C\alpha_k & d_k \\ \hline 0 & 0 & 0 & 1 \end{array}\right]$$

Inverse Link-Coordinate Transformation :

$$T_k^{k-1} = \left[T_{k-1}^{k}\right]^{-1} =$$

$$T_k^{k-1} = \left[\begin{array}{ccc|c} C\theta_k & S\theta_k & 0 & -a_k \\ -C\alpha_k S\theta_k & C\alpha_k C\theta_k & S\alpha_k & -d_k S\alpha_k \\ S\alpha_k S\theta_k & -S\alpha_k C\theta_k & C\alpha_k & -d_k C\alpha_k \\ \hline 0 & 0 & 0 & 1 \end{array}\right]$$

2.6.3 Joint coupling

From the above arm matrix equation, we get the following equations repeatedly.

$\Theta_{23} = \Theta_2 + \Theta_3$

$\Theta_{234} = \Theta_2 + \Theta_3 + \Theta_4$

Θ_{23} = Global Elbow angle measures relative to the x0 y0 work surface.

Θ_{23} = 0 i.e horizontal arm pointing outwards

Θ_{23}= $\Pi/2$ i.e vertical arm pointing in the upward direction.

Θ_{234} = vertical tool pointing in the downward direction

Θ_{234}= horizontal tool pointing outward straight.

In some of the robots the joints such as shoulder, elbow, and tool pitch joints are coupled in order to control the robot at the joint level.

But the D-H algorithm consists of the joint control which are independent and the joints are not dependent on each other i.e no coupling exists.

In order to some mechanical coupling we need to use software with the following parameters:

$\Delta\theta$ = joint movement command

Δh = provides the number of encoders

C = coupling matrix

P = precision

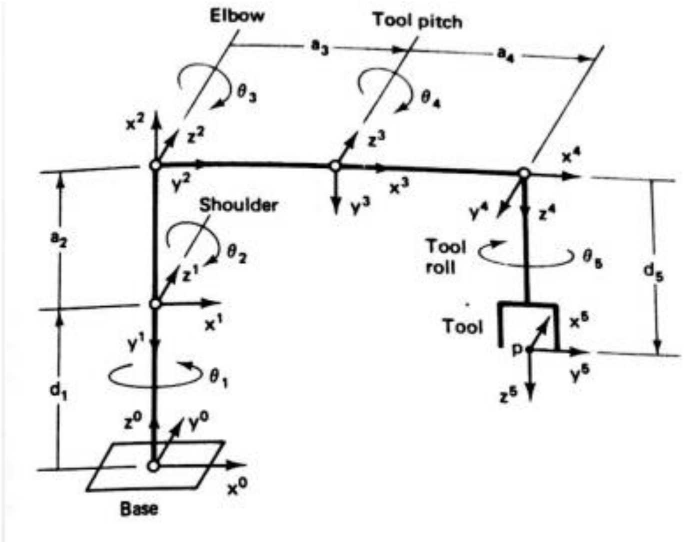
Example: Rhino XR-3 robot , following is the encoders count need to produce in order obtain the movement of theta degrees as shown below:

$$\Delta h = \begin{bmatrix} \rho_1 & 0 & 0 & 0 & 0 \\ 0 & \rho_2 & 0 & 0 & 0 \\ 0 & 0 & -\rho_3 & 0 & 0 \\ 0 & 0 & 0 & -\rho_4 & 0 \\ 0 & 0 & 0 & 0 & \rho_5 \end{bmatrix}^{-1} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \Delta\theta = P^{-1}C\,\Delta\theta$$

2.7 Five-axis robot

Following is the link coordinate diagram based on the Denvait Hartenberg algorithm.

Using the steps from 0 to 7 we obtain the link coordinate diagram for the Rhino XR-3 robot.



Using the D-H algorithm steps 8 to 12 we obtain the following set of kinematic parameters.

d5 = tool length which can vary from robot to robot depending on which tool is installed.

Values for the joint distances d and link length a of the Rhino XR-3 robot are as follows:

d1=26.04cm, d5= 16.83 cm

a2=22.86 cm, a3= 22.86 cm, a4 = 0.95cm

Axis	θ	d	a	α	Home
1	q_1	d_1	0	$-\pi/2$	0
2	q_2	0	a_2	0	$-\pi/2$
3	q_3	0	a_3	0	$\pi/2$
4	q_4	0	a_4	$-\pi/2$	0
5	q_5	d_5	0	0	$-\pi/2$

The arm matrix equation is divided into 2 parts:

T_{base}^{wrist} and T_{wrist}^{tool}

$$T_{\text{base}}^{\text{wrist}} = T_0^1 T_1^2 T_2^3$$

$$= \begin{bmatrix} C_1 & 0 & -S_1 & 0 \\ S_1 & 0 & C_1 & 0 \\ 0 & -1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} C_2 & -S_2 & 0 & a_2 C_2 \\ S_2 & C_2 & 0 & a_2 S_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} C_3 & -S_3 & 0 & a_3 C_3 \\ S_3 & C_3 & 0 & a_3 S_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} C_1 C_2 & -C_1 S_2 & -S_1 & a_2 C_1 C_2 \\ S_1 C_2 & -S_1 S_2 & C_1 & a_2 S_1 C_2 \\ -S_2 & -C_2 & 0 & d_1 - a_2 S_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} C_3 & -S_3 & 0 & a_3 C_3 \\ S_3 & C_3 & 0 & a_3 S_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} C_1 C_{23} & -C_1 S_{23} & -S_1 & C_1(a_2 C_2 + a_3 C_{23}) \\ S_1 C_{23} & -S_1 S_{23} & C_1 & S_1(a_2 C_2 + a_3 C_{23}) \\ -S_{23} & -C_{23} & 0 & d_1 - a_2 S_2 - a_3 S_{23} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_{\text{base}}^{\text{wrist}}(\text{home}) = \left[\begin{array}{ccc:c} 1 & 0 & 0 & a_3 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & d_1 + a_2 \\ \hdashline 0 & 0 & 0 & 1 \end{array}\right]$$

$$T_{\text{wrist}}^{\text{tool}} = T_3^4 T_4^5$$

$$= \begin{bmatrix} C_4 & 0 & -S_4 & a_4 C_4 \\ S_4 & 0 & C_4 & a_4 S_4 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} C_5 & -S_5 & 0 & 0 \\ S_5 & C_5 & 0 & 0 \\ 0 & 0 & 1 & d_5 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} C_4 C_5 & -C_4 S_5 & -S_4 & a_4 C_4 - d_5 S_4 \\ S_4 C_5 & -S_4 S_5 & C_4 & a_4 S_4 + d_5 C_4 \\ -S_5 & -C_5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The arm matrix from base to the tool for the 5-axis articulated – coordinate robot is as follows:

$$\left[\begin{array}{ccc:c} C_1 C_{234} C_5 + S_1 S_5 & -C_1 C_{234} S_5 + S_1 C_5 & -C_1 S_{234} & C_1(a_2 C_2 + a_3 C_{23} + a_4 C_{234} - d_5 S_{234}) \\ S_1 C_{234} C_5 - C_1 S_5 & -S_1 C_{234} S_5 - C_1 C_5 & -S_1 S_{234} & S_1(a_2 C_2 + a_3 C_{23} + a_4 C_{234} - d_5 S_{234}) \\ -S_{234} C_5 & S_{234} S_5 & -C_{234} & d_1 - a_2 S_2 - a_3 S_{23} - a_4 S_{234} - d_5 C_{234} \\ \hdashline 0 & 0 & 0 & 1 \end{array}\right]$$

The final expression depends on all the kinematic parameters and if we evaluate the arm matrix with the soft home position we obtain the following:

$$T_{\text{base}}^{\text{tool}}(\text{home}) = \left[\begin{array}{ccc:c} 0 & 1 & 0 & a_3 + a_4 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & d_1 + a_2 - d_5 \\ \hdashline 0 & 0 & 0 & 1 \end{array}\right]$$

2.8 Four axis robot

Following is the link coordinate diagram based on the Denvait Hartenberg algorithm.

Using the steps from 0 to 7 we obtain the link coordinate diagram for the four-axis SCARA robot.

Next we need to apply D-H algorithm steps 8 to 13 in order to obtain the kinematic parameters as shown below:

Axis	θ	d	a	α	Home
1	q_1	d_1	a_1	π	0
2	q_2	0	a_2	0	0
3	0	q_3	0	0	100
4	q_4	d_4	0	0	$\pi/2$

d1-877 mm, d4= 200mm

d3 = d3 because it's a joint variable where value can range from 0 to 195 mm

a1= 425 mm, a2= 375 mm

The vector of joint variables q= {ϴ1, ϴ2,d3, ϴ4}

ϴ1, ϴ2 = revolute variables for the tool position p

D3= prismatic variable

ϴ4 = revolute variable controls the tool orientation R

ARM Matrix equation T_{base}^{tool} for the SCARA robot is as given below:

There is no need for portioning and we calculate the arm matrix equation as follows:

$$T_{base}^{tool} = T_0^1 T_1^2 T_2^3 T_3^4$$

$$= \begin{bmatrix} C_1 & S_1 & 0 & a_1 C_1 \\ S_1 & -C_1 & 0 & a_1 S_1 \\ 0 & 0 & -1 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} C_2 & -S_2 & 0 & a_2 C_2 \\ S_2 & C_2 & 0 & a_2 S_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & q_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} C_4 & -S_4 & 0 & 0 \\ S_4 & C_4 & 0 & 0 \\ 0 & 0 & 1 & d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} C_{1-2} & S_{1-2} & 0 & a_1 C_1 + a_2 C_{1-2} \\ S_{1-2} & -C_{1-2} & 0 & a_1 S_1 + a_2 S_{1-2} \\ 0 & 0 & -1 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & q_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} C_4 & -S_4 & 0 & 0 \\ S_4 & C_4 & 0 & 0 \\ 0 & 0 & 1 & d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} C_{1-2} & S_{1-2} & 0 & a_1 C_1 + a_2 C_{1-2} \\ S_{1-2} & -C_{1-2} & 0 & a_1 S_1 + a_2 S_{1-2} \\ 0 & 0 & -1 & d_1 - q_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} C_4 & -S_4 & 0 & 0 \\ S_4 & C_4 & 0 & 0 \\ 0 & 0 & 1 & d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

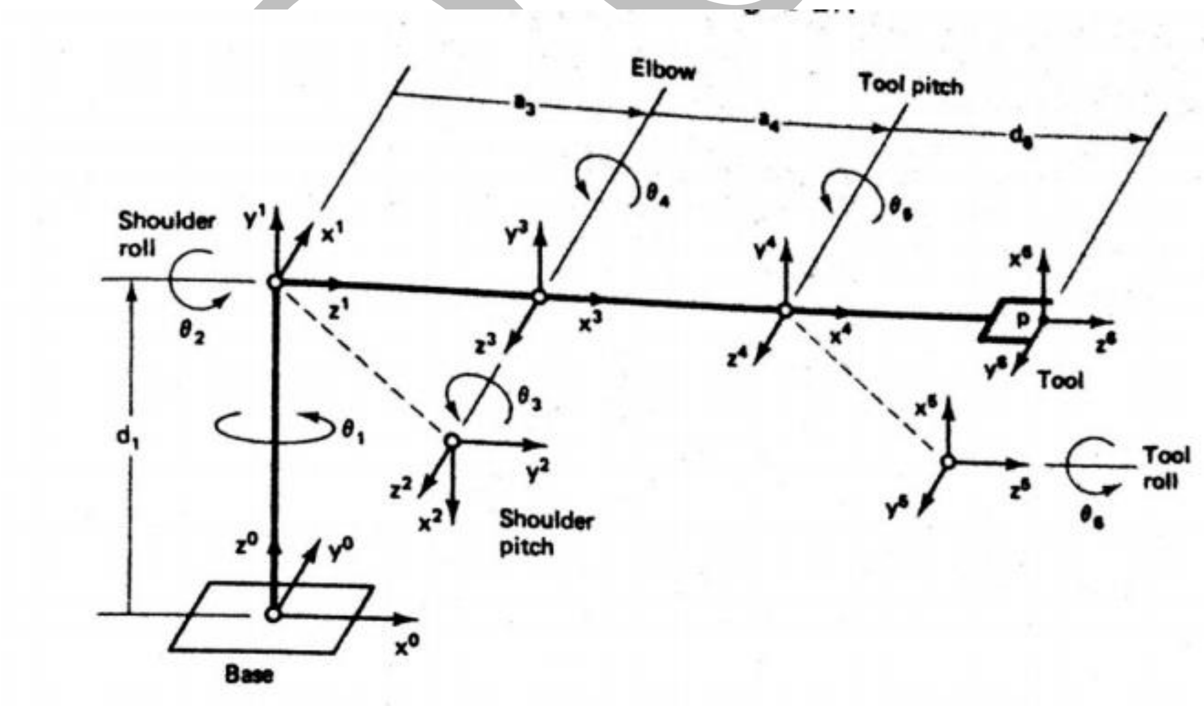
$$T_{base}^{tool} = \left[\begin{array}{ccc|c} C_{1-2-4} & S_{1-2-4} & 0 & a_1 C_1 + a_2 C_{1-2} \\ S_{1-2-4} & -C_{1-2-4} & 0 & a_1 S_1 + a_2 S_{1-2} \\ 0 & 0 & -1 & d_1 - q_3 - d_4 \\ \hline 0 & 0 & 0 & 1 \end{array}\right]$$

Such robots are mostly used in applications of assembly operations where components are required to insert in the circuit boards.

2.9 Six axis robot

Following is the link coordinate diagram for the six axis Intelledex 660 manipulator which is a high precision light –assembly industrial robot used for applications like assembly, material handling applications,etc.

By applying the D-H algorithm steps 0 to 7 we construct a link coordinate diagram as shown below:



Next we need to apply D-H algorithm steps 8 to 13 in order to obtain the kinematic parameters as shown below:

Axis	θ	d	a	α	Home
1	q_1	d_1	0	$\pi/2$	$\pi/2$
2	q_2	0	0	$\pi/2$	$-\pi/2$
3	q_3	0	a_3	0	$\pi/2$
4	q_4	0	a_4	0	0
5	q_5	0	0	$\pi/2$	$\pi/2$
6	q_6	d_6	0	0	0

d6= tool length varies which depends on the type of tool installed.

d1= 373.4 cm, d6= 228.6 mm

a3=304.8cm, a4=304.8 mm

In order to obtain the arm matrix equation, we divide the arm equation into two parts:

T_{base}^{elbow} and T_{elbow}^{tool}

T_{base}^{elbow} = Transformation provides the position and orientation of the L3 frame i.e. elbow relative to the L0 frame i.e. base frame.

$$T_{base}^{elbow} = T_0^1 T_1^2 T_2^3$$

$$= \begin{bmatrix} C_1 & 0 & S_1 & 0 \\ S_1 & 0 & -C_1 & 0 \\ 0 & 1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} C_2 & 0 & S_2 & 0 \\ S_2 & 0 & -C_2 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} C_3 & -S_3 & 0 & a_3C_3 \\ S_3 & C_3 & 0 & a_3S_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} C_1C_2 & S_1 & C_1S_2 & 0 \\ S_1C_2 & -C_1 & S_1S_2 & 0 \\ S_2 & 0 & -C_2 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} C_3 & -S_3 & 0 & a_3C_3 \\ S_3 & C_3 & 0 & a_3S_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} C_1C_2C_3 + S_1S_3 & -C_1C_2S_3 + S_1C_3 & C_1S_2 & (C_1C_2C_3 + S_1S_3)a_3 \\ S_1C_2C_3 - C_1S_3 & -S_1C_2S_3 - C_1C_3 & S_1S_2 & (S_1C_2C_3 - C_1S_3)a_3 \\ S_2C_3 & -S_2S_3 & -C_2 & d_1 + S_2C_3a_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_{base}^{elbow}(\text{home}) = \left[\begin{array}{ccc|c} 1 & 0 & 0 & a_3 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & d_1 \\ \hline 0 & 0 & 0 & 1 \end{array}\right]$$

T_{elbow}^{tool} = provides the position and orientation of the tool –tip frame to the elbow i.e. from L3 frame to L6 frame

$$T_{elbow}^{tool} = T_3^4 T_4^5 T_5^6$$

$$= \begin{bmatrix} C_4 & -S_4 & 0 & a_4 C_4 \\ S_4 & C_4 & 0 & a_4 S_4 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} C_5 & 0 & S_5 & 0 \\ S_5 & 0 & -C_5 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} C_6 & -S_6 & 0 & 0 \\ S_6 & C_6 & 0 & 0 \\ 0 & 0 & 1 & d_6 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} C_{45} & 0 & S_{45} & a_4 C_4 \\ S_{45} & 0 & -C_{45} & a_4 S_4 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} C_6 & -S_6 & 0 & 0 \\ S_6 & C_6 & 0 & 0 \\ 0 & 0 & 1 & d_6 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} C_{45} C_6 & -C_{45} S_6 & S_{45} & a_4 C_4 + S_{45} d_6 \\ S_{45} C_6 & -S_{45} S_6 & -C_{45} & a_4 S_4 - C_{45} d_6 \\ S_6 & C_6 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Following equations are for the position and orientation of the tool frame relative to the base frame for a 6-axis articulated –coordinated robot

$$p = \begin{bmatrix} C_1 C_2(a_3 C_3 + a_4 C_{34} + d_6 S_{345}) + S_1(a_3 S_3 + a_4 S_{34} - d_6 C_{345}) \\ S_1 C_2(a_3 C_3 + a_4 C_{34} + d_6 S_{345}) - C_1(a_3 S_3 + a_4 S_{34} - d_6 C_{345}) \\ d_1 + S_2(a_3 C_3 + a_4 C_{34} + d_6 S_{345}) \end{bmatrix}$$

$$R = \begin{bmatrix} (C_1 C_2 C_{345} + S_1 S_{345}) C_6 + C_1 S_2 S_6 & C_1 S_2 C_6 - (C_1 C_2 C_{345} + S_1 S_{345}) S_6 & -S_1 C_{345} + C_1 C_2 S_{345} \\ (S_1 C_2 C_{345} - C_1 S_{345}) C_6 + S_1 S_2 S_6 & S_1 S_2 C_6 - (S_1 C_2 C_{345} - C_1 S_{345}) S_6 & C_1 C_{345} + S_1 C_2 S_{345} \\ S_2 C_{345} C_6 - C_2 S_6 & -C_2 C_6 - S_2 C_{345} S_6 & S_2 S_{345} \end{bmatrix}$$

REFERENCES:

1.Robert Shilling, "*Fundamentals of Robotics-Analysis and control",* PHI.

2. Fu, Gonzales and Lee, "*Robotics",* McGraw Hill

3. J.J, Craig, "*Introduction to Robotics"*, Pearson Education

CHAPTER 3 Inverse Kinematics:

UNIT STRUCTURE:

3.1 General properties of solutions

3.2 Tool configuration

3.3 Five axis robots

3.4 Three-Four axis

3.5 Six axis robot (Inverse kinematics)

INTRODUCTION:

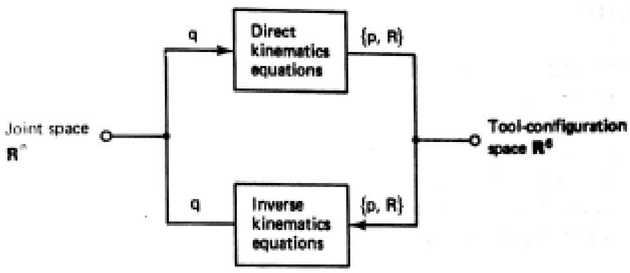
DIRECT Kinematics

- For any robotic manipulator, given the joint variable vector, q(t) i.e. q is either a rotary or a prismatic variable and the geometric link parameter i.e.a's and d's,
- What is the position p and orientation R of the robotic arm in the 3D space w.r.t a fixed coordinate frame from the reference position?
- To find position and orientation
- ARM matrix i.e. Composite Homogenous Coordinate Transformation Matrix: 4 * 4 matrix
- 1^{st} 3 columns give the 3 possible orientations (Yaw, Pitch, Roll) of the tool and the last column gives the position of the tool tip p, thus solving the DKP.
- If we give this matrix as input to the robot, the robot will go and stop in that particular position and in that particular orientation.

INVERSE:

- In Direct Kinematics it is given:joint variable vector 'q' and Geometric Link Parameters
- We need to find find: Position andRotation in 3D space
- Inverse is reverse of Direct Kinematics.
- Also called as arm solutions or backward solutions
- Normally tasks are formulated in terms of Position and Orientation rather than joint variables.

BLOCK DIAGRAM :



EXAMPLE: When external sensors like camera are used then information is in terms of P and R of the objects which needs to be manipulated.

DEFINITION of Inverse Kinematics:

- Given: Position, Rotation and Geometric Link Parameters
- Find: sets of joint variable vectors q
- Imp: Satisfying the same Position and Rotation
- Mapping from Tool Configuration Space to Joint Space
- Tool Configuration Space -------- 6 dimension
- Joint Space ------ 3 dimension
- Given: Tool Configuration Vector w and Geometric Link Parameter
- Find: sets of joint variable vector q
- Satisfying same Position and Rotation
- Given a desired Position and Rotation
- Find the joint variables q
- Solving the direct kinematics problem is equivalent to finding the mapping from joint space to tool configuration space.
- Solving the inverse kinematics problem is equivalent to finding an inverse mapping from tool configuration space back to joint space.
- When solutions do exist, typically they are not unique.
- For ex: some robots are designed with n axes where n > 6.
- For these robots, infinitely many solutions to the inverse kinematics problem typically exist.
- Hence n >6 robots are referred as kinematically redundant robots, because they have more degrees of freedom than are necessary to establish arbitrary tool configuration

Why not unique?

- Space is 3 dimensional in nature i.e. 3D

- Even if the space is 2D there are n no of ways to move an object from one point to another point on the table
- When the arm matrix is equated with the Soft home positionmatrix, we get 16 equations called as the arm equations
- These 16 eq are non-linear equations and are functions of trig functions of sine and cosine of q1 to q6 angles and are independent of each other.
- 16 non-linear eq contains 6 angles,
- Unknown may be
- 2 for 2 axis robot
- 3 for 3 axis robot
- 4 for 4 axis robot
- 5 for 5 axis robot
- 6 for 6 axis robot
- Since the no of simultaneous equation 16 to be solved are greater than the no of unknowns 6, definitely multiple solution exists.
- Out of these ‘n’ no of solutions, it is very difficult for the user to select the most appropriate solution i.e. shortest path solution.
- Inverse Kinematics problem is robot dependent.
- A particular class of robots can be treated in the same manner i.e. IK analysis is generic in nature and is same for same class of robots, but is different for different generations of robots
- IK is difficult when coupling exists between the joints.
- Extraction of joint variables needs to be done with the use of intermediate variables to remove the coupling between the joints.
- IKS becomes simplifies if cameras and feedback devices such as sensors are used.
- The key to obtain IK solution is to make use of cameras, feedback devices and spherical wrists.
- If robot is redundant one, then infinite solutions are possible to reach P.
- IKP depends on:
- Degree of Freedom
- Work space
- No of joint variables
- No of constraints

- Greater DOF more difficult to extract Joint variable
- Apart from 12 constraints:
- 6 independent constraints
- 6 unknown joint variables
- 12 constraints------ output of DKP
- Like DK the IK does not have any unique method.
- Multiple methods can be used to solve IKP

3.1 General properties of solutions

- **Existence of Solutions**
- No solution will exist if the desired tool tip outside the work envelope.
- Even when P is within the work envelope, the solution may or may not exist.
- Consider the arm equation below:

$$T_{base}^{tool}(q) = \begin{bmatrix} R_{11} & R_{12} & R_{13} & p_1 \\ R_{21} & R_{22} & R_{23} & p_2 \\ R_{31} & R_{32} & R_{33} & p_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Last row of the column will be a constant always
- Arm equation contains 12 simultaneous nonlinear algebraic equations where there arean unknown q component.
- 12 equations are dependent of each other.
- R= coordinate transformation matrix i.e. pure rotation
- Mutual orthogonality puts three constraints on the columns of R

$$r^1 \cdot r^2 = 0$$
$$r^1 \cdot r^3 = 0$$
$$r^2 \cdot r^3 = 0$$

$\|r^k\| = 1 \quad 1 <= k <= 3$

- 12 constraints of the arm equation indicate 6 independent constraints
- Present on the unknown components of the vector of q joint variable.
- In order to provide a GENERAL SOLUTION for the Inverse Kinematics problem, for which tool configuration can be generated if we can find q.
- Number of independent constraints should match the number of unknowns.
- If the p i.e. tool tip position and R i.e. orientation of the tool is given, then p wrist i.e. position of the wrist can be determined from p by tracing it in the backward direction along the approach vector.

  $P^{wrist} = p – dn\ r^3$

  Where dn = joint distance i.e tool length

  r3 = last column of the rotation matrix.

- P^{wrist}can be obtained with the help of p,R,dn , q1,q2,q3 joint variables used for positioning the wrist and it can be obtained from the following reduced arm equation

$$T^{wrist}_{base}(q_1, q_2, q_3)i^4 = \begin{bmatrix} p - d_n r^3 \\ 1 \end{bmatrix}$$

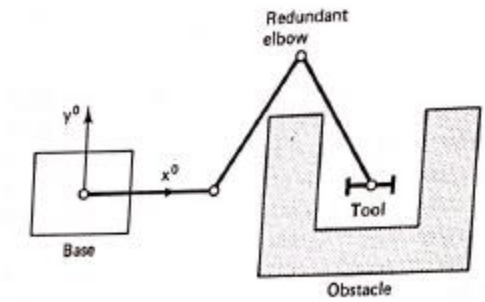
Uniqueness of Solution

- The existence of a solution to the inverse kinematics problem is not the only issue that needs to be addressed.
- Inverse kinematics provides the solutions which are not unique.
- There are numerous solutions which exist

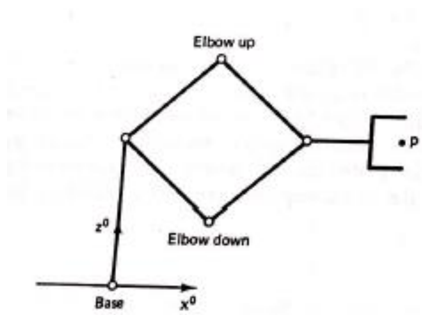
  Example: For robot's n > 6 there are infinite solutions to solve the inverse kinematics problem.

  Such robots are called Kinematically REDUNDANT robots

- **Ex: SCARA robot requires only 2 joint in order to establish the tool horizontal position and its second elbow is redundant.**



- Ex: Articulated coordinate robot has limitations on the travelling range of the shoulder, elbow and tool pitch joints because they are sufficiently large and we have 2 distinct solutions to perform the job of placing the tool in front of the robot.
- 2 solutions are: elbow-up and elbow –down
- In TCS both the solutions are same since they provide the same p and R as they are distinct in joint space.
- And mostly elbow up solution is considered in order to avoid the collision related to the links of the robot arm and obstacles on the work space.



3.2 Tool configuration

To solve the inverse kinematics problem, the input is the desired tool configuration.

In order to obtain the tool, roll angle from a scaled approach we should use an invertible function of the roll angle qn to scale the r3 length.

When qn is bounded then we obtain the following positive, invertible, exponential scaling function.

$$f(q_n) \triangleq \exp \frac{q_n}{\pi}$$

Tool Configuration vector(TCV):

- Assume that the p and R are the position and orientation of the tool with respect to the Base frame.
- qn = toll roll angle
- **TCV = vector in R^6**

$$w \triangleq \begin{bmatrix} w^1 \\ \hline w^2 \end{bmatrix} \triangleq \begin{bmatrix} p \\ \hline [\exp(q_n/\pi)] r^3 \end{bmatrix}$$

- W1=p= tool tip position
- W2= last 3 components present the tool orientation
- qn can be determined from the TCV i.e. w

Tool ROLL: following equation provides that the tool roll angle can be obtained from the TCV w

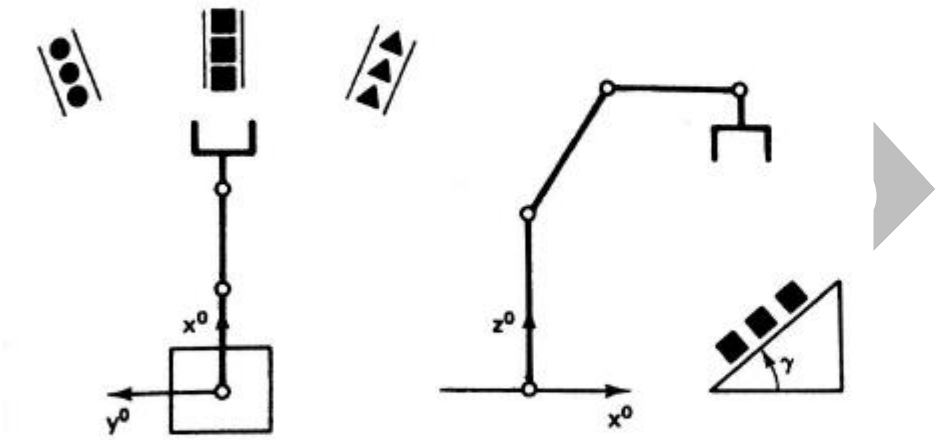
$$q_n = \pi \ln (w_4^2 + w_5^2 + w_6^2)^{1/2}$$

- The TCS parameters i.e. p and R are always associated with a vector 'w' which is a subset of R6 .
- This R6 is called as TCS and is 6D frame.

TCS R6

- Defined as the space in which the tool is configured.
- The tool or the gripper or the end – effector is lying in its own space or rotating in its own space. That space is called as TCS.**Joint Space or Joint Coordinate Space**
- Defined as the space in which the n *1 vector of joint variables is defined.
- The n dimensional space in which all the n no of joints is situated is called as space or the vector space and is denoted by Rn.

3.2.1 Tool configuration of a Five-axis Articulated Robot



Consider the example of 5-axis articulated-coordinate robot which doesn't have tool yaw motion.In the figure, there are part feeders arranged in concentric manner around the robot. The part feeders are kept at an angle Y which is in the range of the tool pitch angle which can be realizable by the robot. Since the tool yaw angle is fixed as seen in the figure therefore the approach vector of the tool only lies in the vertical plane w.r.t base axis. This limitation can be seen in mathematical form in terms of the arm matrix components as:

$R_{13}\, p_2 = R_{23}\, p_1$

In practical situation, tool can do manipulation of the objects from different views such as above, front, back and also from below depending on the limits of the tool pitch angle ranges.

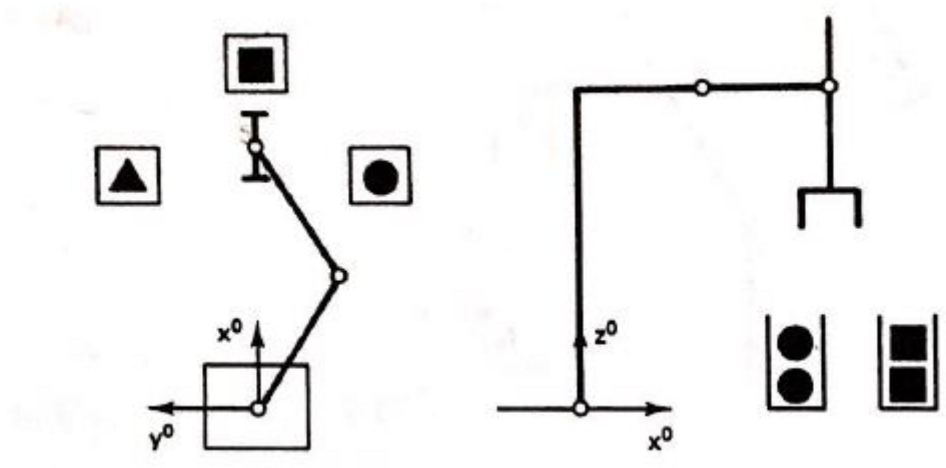
We can see that such robots cannot perform operations of manipulation of objects from the side. If we substitute the above equation in

$$w \triangleq \begin{bmatrix} w^1 \\ \hline w^2 \end{bmatrix} \triangleq \begin{bmatrix} p \\ \hline [\exp(q_n/\pi)]r^3 \end{bmatrix}$$

we obtain the tool configuration locus as shown below:

$$w = [w_1, w_2, w_3, \beta w_1, \beta w_2, w_6]^T$$

3.2.2 Tool configuration of a Four-axis Articulated Robot



Consider the case where we need to perform the parts manipulation where they need to be approached from above i.e. a part may be required to picked up from a horizontal surface as we can see in the figure and also the part need to be placed on the horizontal surface.

For such operations of task manipulation, we consider either the SCARA robot or 4-axis horizontal or jointed robot.

In this the approach vector r3 is constraint to be along a vertical line within the same plane such as:

$r^3 = -i^3$

In 4-axis SCARA robot: first 3-axes: major axes for position of the tool tip

4^{th}axis: minor axis is for to orient the tool by changing the sliding vector.

When we substitute the above equation in

$$w \triangleq \begin{bmatrix} w^1 \\ --- \\ w^2 \end{bmatrix} \triangleq \begin{bmatrix} p \\ \text{-----------} \\ [\exp(q_n/\pi)]r^3 \end{bmatrix}$$

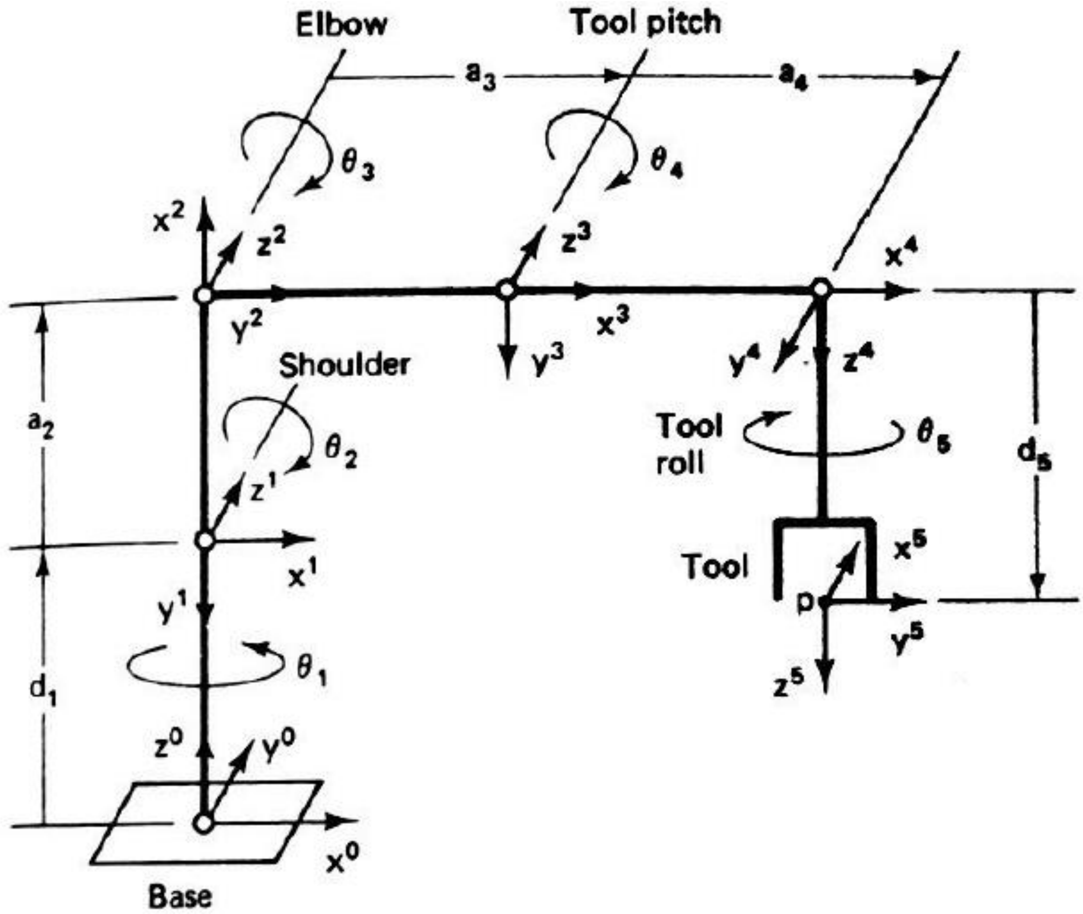
we obtain the following:

$$w = \left[p_1, p_2, p_3, 0, 0, -\exp\frac{q_4}{\pi}\right]^T$$

We obtain 4 degrees of freedom in which 2 components of the TCV are always zero.

3.3 Five axis robots

Following is the kink-coordinate diagram for a five axis articulated robot:



To solve the inverse kinematics, we use the following expression of tool configuration vector w(q) obtained from the arm matrix equation.

$$
w(q) = \begin{bmatrix} C_1(a_2C_2 + a_3C_{23} + a_4C_{234} - d_5S_{234}) \\ S_1(a_2C_2 + a_3C_{23} + a_4C_{234} - d_5S_{234}) \\ d_1 - a_2S_2 - a_3S_{23} - a_4S_{234} - d_5C_{234} \\ \hdashline -[\exp(q_5/\pi)]C_1S_{234} \\ -[\exp(q_5/\pi)]S_1S_{234} \\ -[\exp(q_5/\pi)]C_{234} \end{bmatrix}
$$

For the joint variable vector q to obtain from the nonlinear TCV w(q) we use row operations to the w components and obtain various trigonometric identities.

BASE JOINT:

Consider w1 and w2, divide w2 by w1 we obtain S1/C1 and the base angle can be extracted as

$$q_1 = \text{atan2}\,(w_2, w_1)$$

ELBOW JOINT:

This angle is difficult to extract because it is associated with 2 angles i.e. shoulder and tool pitch angle i.e. they are coupled with each other.

To decouple them we need to isolate them by using an intermediate variable q234 which is known as Global Tool Pitch angle. q234= q2 + q3 + q4

Global tool pitch angle can be calculated using the following equation:

$$q_{234} = \text{atan2}\,[-(C_1 w_4 + S_1 w_5),\ -w_6]$$

To decoupled the shoulder and elbow angles, we have to use 2 intermediate variables

$$b_1 \triangleq C_1 w_1 + S_1 w_2 - a_4 C_{234} + d_5 S_{234}$$

$$b_2 \triangleq d_1 - a_4 S_{234} - d_5 C_{234} - w_3$$

Consider the w components in TCV and substitute in the b1 and b2 expressions, we obtain the following:

$$b_1 = a_2 C_2 + a_3 C_{23}$$

$$b_2 = a_2 S_2 + a_3 S_{23}$$

Thus the coupling between shoulder and elbow angles is removed. To isolate elbow angle we need to perform $||b||^2$

Use trigonometric identities and simplify we obtain the following:

$$\|b\|^2 = a_2^2 + 2a_2 a_3 C_3 + a_3^2$$

||b||= depends on the q3 elbow angle since ||b|| represents the distance between the shoulder L1 frame and L3 wrist frame.

When we solve the above equation, we can obtain q3 which gives 2 solutions i.e. elbow-up and elbow-down solution.

$$q_3 = \pm \arccos \frac{\|b\|^2 - a_2^2 - a_3^2}{2a_2 a_3}$$

SHOUDER JOINT:

From b1 and b2 equations we can isolate the q2 shoulder angle.

Expanding C_{23} and S_{23}, use cosine and sine of the sum with the help of trigonometric identities, we obtain the following equation by rearranging the terms:

$$b_1 = (a_2 + a_3 C_3)C_2 - (a_3 S_3)S_2$$

$$b_2 = (a_2 + a_3 C_3)S_2 + (a_3 S_3)C_2$$

Perform the row operations to solve the above two simultaneous linear equations for the unknowns C2 and S2 and we obtain the

$$C_2 = \frac{(a_2 + a_3 C_3)b_1 + a_3 S_3 b_2}{\|b\|^2}$$

$$S_2 = \frac{(a_2 + a_3 C_3)b_2 - a_3 S_3 b_1}{\|b\|^2}$$

The shoulder angle can be obtained as:

$$q_2 = \text{atan2}\,[(a_2 + a_3 C_3)b_2 - a_3 S_3 b_1,\ (a_2 + a_3 C_3)b_1 + a_3 S_3 b_2]$$

TOOL PITCH JOINT:

The shoulder angle q2, the elbow angle q3 and the global tool pitch angle q234 are known to us , therefore we can obtain the tool pitch angle q4 as:

$$q_4 = q_{234} - q_2 - q_3$$

TOOL ROLL JOINT q5:

This can be obtained from the last three components of w as:

$$q_5 = \pi \ln (w_4^2 + w_5^2 + w_6^2)^{1/2}$$

Using the rotation matrix we can obtain the q5 as:

$$q_5 = \text{atan2}\,(S_1 R_{11} - C_1 R_{21},\ S_1 R_{12} - C_1 R_{22})$$

Entire Solution:

The following solution is applicable to the robots such as:

Rhino XR-3 and Alpha II robot

$$a_1 = \text{atan2}(w_2, w_1)$$

↓

$$q_{234} = \text{atan2}\left[-(C_1 w_4 + S_1 w_5), -w_6\right]$$
$$b_1 = C_1 w_1 + S_1 w_2 - a_4 C_{234} + d_5 S_{234}$$
$$b_2 = d_1 - a_4 S_{234} - d_5 C_{234} - w_3$$

↓

$$q_3 = \pm\arccos\frac{\|b\|^2 - a_2^2 - a_3^2}{2a_2 a_3}$$

↓

$$q_2 = \text{atan2}\left[(a_2 + a_3 C_3)b_2 - a_3 S_3 b_1, (a_2 + a_3 C_3) b_1 + a_3 S_3 b_2\right]$$

↓

$$q_4 = q_{234} - q_2 - q_3$$

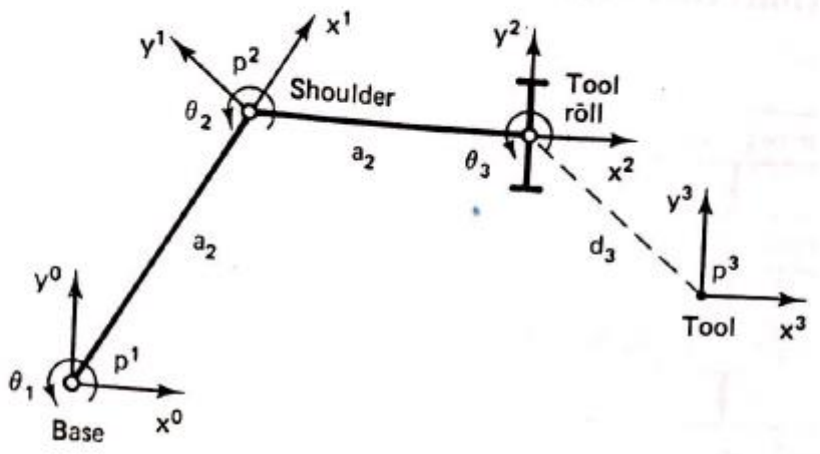
↓

$$q_5 = \pi \ln\left(w_4^2 + w_5^2 + w_6^2\right)^{1/2}$$

3.4 Three- axis

In order to solve the Inverse Kinematics of a robot we need to first solve the Direct Kinematics of a three-axis robot.

Using steps 0 to 7 of the D-H algorithm we obtain the following link-coordinate diagram for a three axis articulated robot:



Then using the steps 8 to 12 of D-H algorithm we obtain the following Kinematic Parameters as shown below:

Axis	θ	d	a	α	Home
1	q_1	0	a_1	0	$\pi/3$
2	q_2	0	a_2	0	$-\pi/3$
3	q_3	d_3	0	0	0

To solve the inverse kinematics, we use the arm matrix equation.

$$T_{\text{base}}^{\text{tool}} = T_0^1 T_1^2 T_2^3$$

$$= \begin{bmatrix} C_1 & -S_1 & 0 & a_1C_1 \\ S_1 & C_1 & 0 & a_1S_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} C_2 & -S_2 & 0 & a_2C_2 \\ S_2 & C_2 & 0 & a_2S_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} C_3 & -S_3 & 0 & 0 \\ S_3 & C_3 & 0 & 0 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} C_{12} & -S_{12} & 0 & a_1C_1 + a_2C_{12} \\ S_{12} & C_{12} & 0 & a_1S_1 + a_2S_{12} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} C_3 & -S_3 & 0 & 0 \\ S_3 & C_3 & 0 & 0 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} C_{123} & -S_{123} & 0 & a_1C_1 + a_2C_{12} \\ S_{123} & C_{123} & 0 & a_1S_1 + a_2S_{12} \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

To obtain the following tool configuration vector w(q) for 3-axis robot

$$W(q) = \begin{bmatrix} a_1C_1 + a_2C_{12} \\ a_1S_1 + a_2S_{12} \\ 0 \\ 0 \\ \exp(q_3/\pi) \end{bmatrix}$$

SHOULDER JOINT:

The shoulder angle q2 is obtain from the distance between base frame L0 frame and wrist frame L2. Taking the square of the w1 and w2 components and adding the terms we obtain the following:

$$w_1^2 + w_2^2 = (a_1C_1 + a_2C_{12})^2 + (a_1S_1 + a_2S_{12})^2$$
$$= a_1^2C_1^2 + 2a_1a_2C_1C_{12} + a_2^2C_{12}^2 + a_1^2S_1^2 + 2a_1a_2S_1S_{12} + a_2^2S_{12}^2$$
$$= a_1^2 + 2a_1a_2(C_{12}C_1 + S_{12}S_1) + a_2^2$$
$$= a_1^2 + 2a_1a_2C_2 + a_2^2$$

Solving the above equation, we obtain:

$$q_2 = \pm\arccos\frac{w_1^2 + w_2^2 - a_1^2 - a_2^2}{2a_1a_2}$$

BASE JOINT:

Expanding the terms C_{12} and S_{12} and taking the sum of the trigonometric terms, using the coefficients of C1 and S1 we obtain the following:

$$(a_1 + a_2C_2)C_1 - (a_2S_2)S_1 = w_1$$
$$(a_2S_2)C_1 + (a_1 + a_2C_2)S_1 = w_2$$

By using row operations, and solving the above two simultaneous equations we obtain the following:

$$C_1 = \frac{(a_1 + a_2C_2)w_1 + a_2S_2w_2}{(a_1 + a_2C_2)^2 + (a_2S_2)^2}$$
$$S_1 = \frac{(a_1 + a_2C_2)w_2 - a_2S_2w_1}{(a_1 + a_2C_2)^2 + (a_2S_2)^2}$$

Consider a2=a1 then for C1 and S1 equations, the numerators and denominators goes to zero we obtain q2= Π. This means to a workspace singularity where the base axis and the tool roll axis collinear to each other. To avoid the singularity , C1 and S1 are unknown then the base angle q1 can be obtained as:

$$q_1 = \text{atan2}\,[(a_1 + a_2C_2)w_2 - a_2S_2w_1, (a_1 + a_2C_2)w_1 + a_2S_2w_2]$$

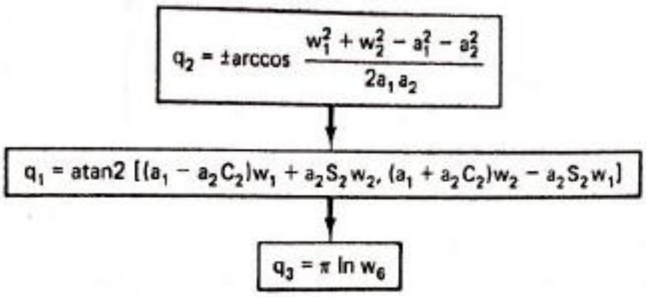
TOOL ROLL JOINT:

From the last component of the tool-configuration vector , the tool roll angle q3 can be obtained as:

$q_3 = \pi \ln w_6$

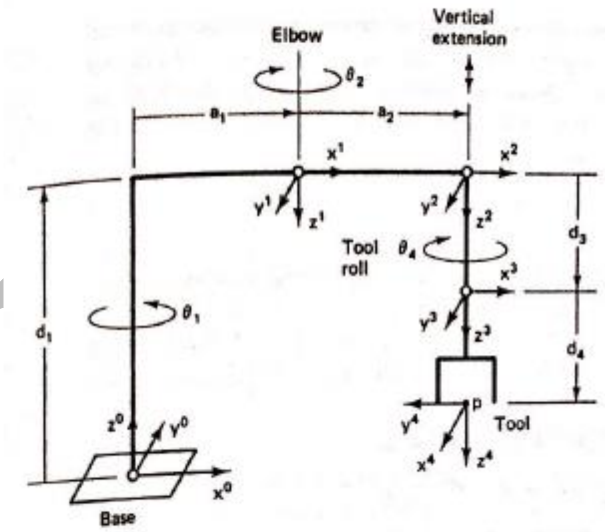
Complete Solution:

The following is the entire algorithm for the 3-axis planar articulated robot



Four-axis robot:

Following is the kink-coordinate diagram for a four axis articulated robot:



To solve the inverse kinematics, we use the tool configuration vector as follows:

$$W(q) = \begin{bmatrix} a_1C_1 + a_2C_{1-2} \\ a_1S_1 + a_2S_{1-2} \\ d_1 - q_3 - d_4 \\ 0 \\ 0 \\ -\exp(q_4/\pi) \end{bmatrix}$$

ELBOW JOINT:

Using the first 2 components of the TCV (w) we can extract the elbow angle.

Squaring w1 and w2 and then adding the terms we obtain the radial component to the tool position. Solving and simplify by using the trigonometric identities we obtain the following :

$$w_1^2 + w_2^2 = a_1^2 + 2a_1a_2C_2 + a_2^2$$

Solving the above equation we can obtain q2 which gives 2 solutions one called left-handed solution and other called as right-handed solution

$$q_2 = \pm\arccos \frac{w_1^2 + w_2^2 - a_1^2 - a_2^2}{2a_1a_2}$$

BASE JOINT:

In the expressions of w1 and w2 components we expand C_{1-2} and S_{1-2} terms. Taking the difference and sine of the difference trigonometric, rearranging the coefficients we obtain

$$(a_1 + a_2C_2)C_1 + (a_2S_2)S_1 = w_1$$
$$(-a_2S_2)C_1 + (a_1 + a_2C_2)S_1 = w_2$$

Using row operations to solve this linear system we obtain the following equations:

$$S_1 = \frac{a_2S_2w_1 + (a_1 + a_2C_2)w_2}{(a_2S_2)^2 + (a_1 + a_2C_2)^2}$$

$$C_1 = \frac{(a_1 + a_2C_2)w_1 - a_2S_2w_2}{(a_2S_2)^2 + (a_1 + a_2C_2)^2}$$

The above expressions are for both cosine and the sine of the base angle, thus using the atan2 we obtain the following:

$$q_1 = \text{atan2}\,[a_2S_2w_1 + (a_1 + a_2C_2)w_2,\ (a_1 + a_2C_2)w_1 - a_2S_2w_2]$$

VERTICAL EXTENSION JOINT:

In this robot we have q3 as a prismatic joint variable where its slides the tool up and down along the tool roll axis.

$$q_3 = d_1 - d_4 - w_3$$

TOOL ROLL JOINT:

The last component of the TCV(w) can recovered the final joint variable i.e tool roll angle q4 as shown below:

$$q_4 = \pi \ln |w_6|$$

After obtaining the global tool roll angle, the tool roll angle q4 can be calculated from q1 and q2 as shown:

$$q_4 = q_1 - q_2 - q_{1-2-4}$$

COMPLETE SOLUTION:

Following is the complete algorithm for a four-axis SCARA robot.

$$q_2 = \pm\arccos \frac{w_1^2 + w_2^2 - a_1^2 - a_2^2}{2a_1 a_2}$$

↓

$$q_1 = \text{atan2}\left[a_2S_2w_1 + (a_1 + a_2C_2)\,w_2,\ (a_1 + a_2C_2)w_1 - a_2S_2w_2\right]$$

↓

$$q_3 = d_1 - d_4 - w_3$$

↓

$$q_4 = \pi \text{ in } |w_6|$$

3.5 Six axis robot (Inverse kinematics)

Following is the kink-coordinate diagram for a six axis robot:

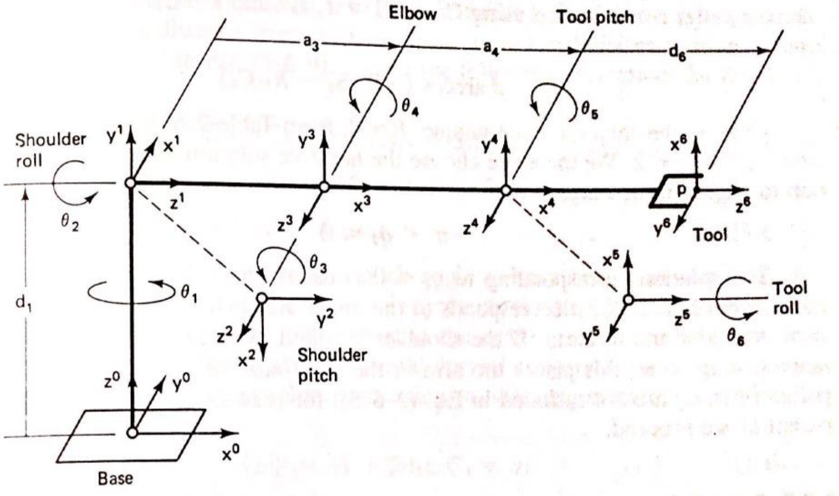


Figure 3-9 Link coordinates of a six-axis articulated robot (Intelledex 660T).

Following is the tool configuration vector for the Intellexdex 660 robotic arm as shown below:

$$R = \left[\begin{array}{c:c:c} (C_1C_2C_{345} + S_1S_{345})C_6 + C_1S_2S_6 & C_1S_2C_6 - (C_1C_2C_{345} + S_1S_{345})S_6 & -S_1C_{345} + C_1C_2S_{345} \\ (S_1C_2C_{345} - C_1S_{345})C_6 + S_1S_2S_6 & S_1S_2C_6 - (S_1C_2C_{345} - C_1S_{345})S_6 & C_1C_{345} + S_1C_2S_{345} \\ S_2C_{345}C_6 - C_2S_6 & -C_2C_6 - S_2C_{345}S_6 & S_2S_{345} \end{array}\right]$$

TOOL ROLL JOINT:

From the last 3 components of the Tool configuration vector w we can obtain the tool roll angle q6 as :

$$q_6 = \pi \ln (w_4^2 + w_5^2 + w_6^2)^{1/2}$$

SHOULDER ROLL JOINT:

Using the third components of the normal vector r1 and the sliding vector r2 , toll roll angle we can obtain the shoulder roll angle q2 as follows :

$$q_2 = \pm \arccos(-R_{31}S_6 - R_{32}C_6)$$

The solution of Inverse kinematics is not UNIQUE.

We consider the negative solution and limit the angles in the range as seen below:

$$-\pi < q_2 < 0$$

BASE JOINT:

Using the first and second components of the normal vector r1 and the sliding vector r2, we can obtain the base angle q1.

We first calculate the following :

$$R_{11}S_6 + R_{12}C_6 = C_1S_2$$

From the shoulder roll joint two equations we can see that S2 is a non-zero quantity. Using q6 , we have to isolate C1. In order to isolate S1, we have to recover the base angle over the entire range [-Π,Π]. We can obtain with the help of the second components of the normal and sliding vector. Using the trigonometric identities we obtain the following:

$$R_{21}S_6 + R_{22}C_6 = S_1S_2$$

Dividing the above 2 equations, we can obtain the base angle q1 as shown below:

$$q_1 = \text{atan2}(R_{21}S_6 + R_{22}C_6, R_{11}S_6 + R_{12}C_6)$$

ELBOW JOINT:

The elbow joint is difficult to extract because it is tightly coupled to the shoulder pitch angle q3 and the tool pitch angle q5. We need to partitioned at the wrist in order to remove the effects of the tool pitch angle q5. The result is b given as :

$$b \triangleq p - d_6 r^3 - d_1 i^3$$

1st 2 components of b = x and y coordinates of L4 relative to the L0

3rd component of b = vertical distance from L4 to L2

Substituting p and r3 we obtain the following results:

$$b_1 = C_1C_2(C_3a_3 + C_{34}a_4) + S_1(S_3a_3 + S_{34}a_4)$$
$$b_2 = S_1C_2(C_3a_3 + C_{34}a_4) - C_1(S_3a_3 + S_{34}a_4)$$
$$b_3 = S_2(C_3a_3 + C_{34}a_4)$$

Thus we have removed the effects of the tool pitch angle q5.

||b||= linear distance between shoulder joint and wrist joint

||b|| = depends on q4

After simplification we can calculate $\|b\|^2$ given as:

$$\|b\|^2 = a_3^2 + 2a_3a_4C_4 + a_4^2$$

Solving the above equation we obtain q4 as :

$$q_4 = \pm\arccos\frac{\|b\|^2 - a_3^2 - a_4^2}{2a_3a_4}$$

Thus we observe once again that the solution is not unique.

+ve = elbow-down solution

-ve= elbow –up solution

SHOULDER PITCH JOINT:

Consider the following equations:

$$b_1 = C_1C_2(C_3a_3 + C_{34}a_4) + S_1(S_3a_3 + S_{34}a_4)$$
$$b_2 = S_1C_2(C_3a_3 + C_{34}a_4) - C_1(S_3a_3 + S_{34}a_4)$$
$$b_3 = S_2(C_3a_3 + C_{34}a_4)$$

From b1 and b2 , we obtain the following :

$S_1 b_1 - C_1 b_2 = S_3 a_3 - S_{34} a_4$

Solving for $S_{34}$and rearrange the coefficients we obtain the following:

$$S_1 b_1 - C_1 b_2 = (a_3 + C_4 a_4) S_3 + (S_4 a_4) C_3$$

From b3 we obtain

$b_3 / S_2 = C_3 a_3 + C_{34} a_4$

Solving for $C_{34}$and rearrange the coefficients we obtain the following:

$$\frac{b_3}{S_2} = (-S_4 a_4) S_3 + (a_3 + C_4 a_4) C_3$$

NOTE: The above equation as seen is divided by S_2 because S_2 is not equal to zero.

Now solving the S_{34} and C_{34} using row operations we obtain the following result:

$$C_3 = \frac{S_4 a_4 (S_1 b_1 - C_1 b_2) + (a_3 + C_4 a_4) b_3 / S_2}{(S_4 a_4)^2 + (a_3 + C_4 a_4)^2}$$

$$S_3 = \frac{(a_3 + C_4 a_4)(S_1 b_1 - C_1 b_2) - S_4 a_4 b_3 / S_2}{(S_4 a_4)^2 + (a_3 + C_4 a_4)^2}$$

Performing atan2 function with C3 and S3 we obtain the following:

$$q_3 = \text{atan2}\left[(a_3 + C_4 a_4)(S_1 b_1 - C_1 b_2) - \frac{S_4 a_4 b_3}{S_2}, \; S_4 a_4 (S_1 b_1 - C_1 b_2) + \frac{(a_3 + C_4 a_4) b_3}{S_2}\right]$$

TOOL PITCH JOINT:

The q5 tool pitch angle can be calculated using the sum angle q235 using the R matrix.

$R_{33} = S_2 S_{345}$

Since S2 is known and nonzero , then q345 is given over the entire interval [-Π/2 ,Π/2].

In order to obtain the solution over the interval [- Π, Π] we need to isolate C_{345}.

Using the remaining components of the third row of R .

From the rotation matrix and $R_{31} C_6 - R_{32} S_6 = S_2 C_{345}$, we get

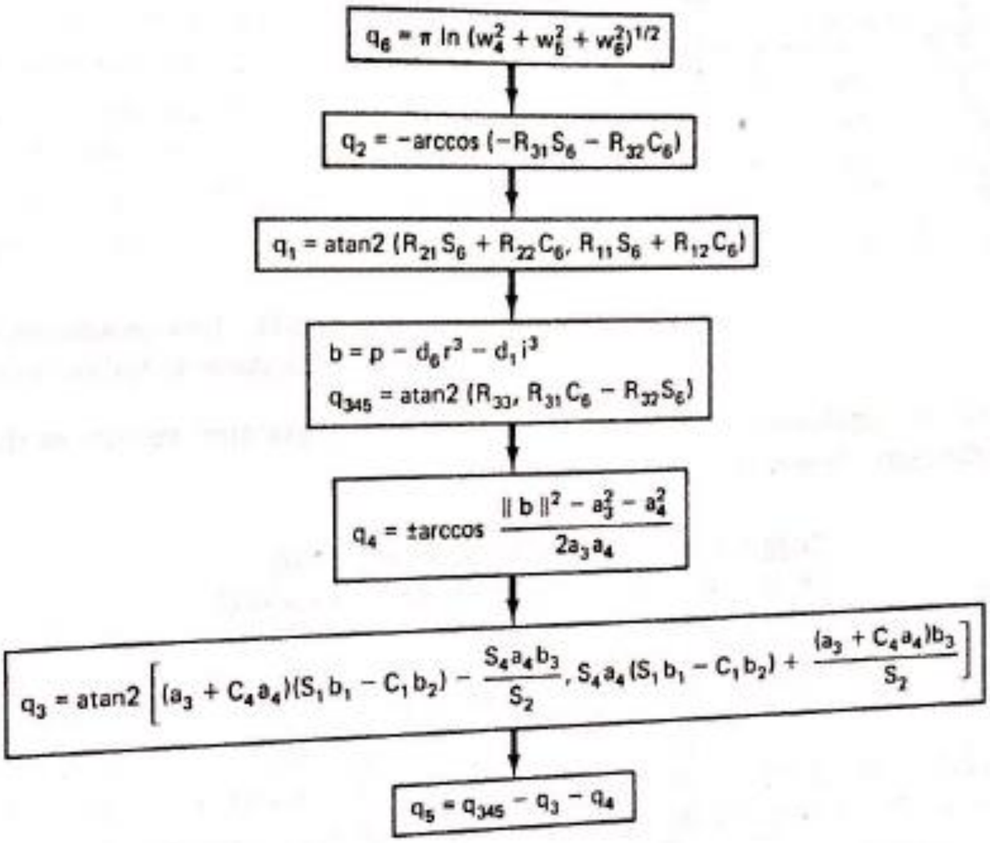
$$q_{345} = \text{atan2}\,(R_{33}, R_{31} C_6 - R_{32} S_6)$$

Knowing q3, q4 and q345 we obtain the following:

$$q_5 = q_{345} - q_3 - q_4$$

COMPLETE SOLUTION:

Following is the summarized algorithm for the 6-axis Intellexdex 660 robot as shown:



REFERENCES:

1.Robert Shilling, "*Fundamentals of Robotics-Analysis and control",* PHI.

2. Fu, Gonzales and Lee, "*Robotics",* McGraw Hill

3. J.J, Craig, "*Introduction to Robotics"*, Pearson Education

CHAPTER 7: Moment of Inertia

Definition: The property of virtue of which it resists any change in its state of rest or of uniform motion is called as Inertia.

Types of Inertia:

1. Translatory Inertia: Inertia generated by the body translation when a force is applied and it depends on the mass and acceleration. Such inertia is called Mass(M)
2. Rotational Inertia: Inertia generated by the body rotation when a torque is applied and it depends on the moment of inertia and angular acceleration. Such inertia is called **Moment of Inertia(MI)**

Robot Dynamics:

- The study of motion of the robot arm considering the forces or torques which results the motion.

Two Dynamic models of robots are:

1. Euler-Lagrange method
2. Euler-Newton method

- Both the models give a closed-form solution that provides a real time control to the robotic arms.
- Robot has to accelerate, need to move at constant speed and decelerate and then follow the trapezoidal speed profile curve during the pick-place operation.
- The variation of the time w.r.t to position and orientation of the robot is called as dynamic behaviour.
- The variation in the torque is applied through the joint motors at the joints in order to balance the internal as well as external forces.
- Due to the motion of the links, it results in internal forces and due to the forces of the external environment, it results into external forces.
- The links and joints need to work properly to withstand the by the forces in order to balance between them.
- The Dynamic response caused through the input actuators can be referred through the set of Equations of Motion.
- To execute a task, calculation of the torque and associated forces is required to study the dynamic model of a robot.

- To simulate and design some control algorithms, the dynamic behaviour of the robot gives the relation of the joints and the link motion.

Control Problems due to Moment of Inertia

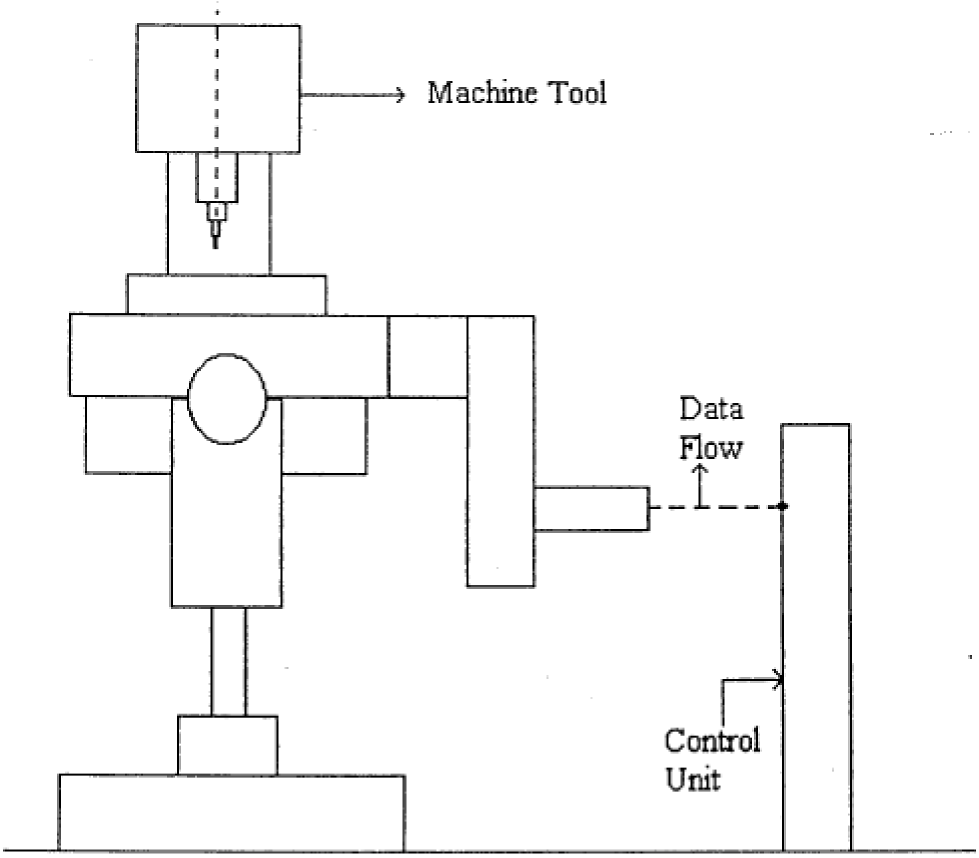
- Dynamics of a robot depends on the mass and moment of inertia.
- Moment of Inertia depends on the area i.e more is the area then more is the MI. Thus more amount of inertia demands more torque will be required to move the robot manipulator links which will it heavy and slow in its function.
- If the mass, weight of the manipulators increases, size of the motor will also increase then the torque of the motors rating required will also be high.
- To overcome the acceleration of the MI we need to use the torque of the link-joint motors, thus the useful torque required is available to lift and move the load.
- In order to drive the joints, more power will be required.
- Thus to drive the actuators, power amplifier of higher rating will be required.
- Following points to be considered while controlling a robot.
    - MI depends on various factors such as weight of the link, payload. Therefore, we need to reduce the link weight for effective performance.
    - Damping should be between 0 and 1
    - MI also depends on the axis i.e more is the area then more is MI and more bending force will be required. If less is the area, less is the MI and less bending force will be required.
    - Light weight joint –link movement is required.
    - Base should contain heavy motors and gear parts so that power is transmitted from base to other respective joints.
    - Need to reduce the robot links mass so that MI also gets reduced and subsequently less torque is enough for the movement of the motors of the joint. Therefore, motors should be less weight, and small in size.

CHAPTER 8: Principles of NC and CNC Machines:

Introduction

- NC and CNC machines are one type of automation systems (soft automation)
- SA is a technique in which a variety of products of different size, shape, volumes can be manufactured by changing the control software.

Numerically Controlled Machines(NC)



A numerical Controlled Machine

- Defined as the control of a machine tool by means of recorded information on punched tape/cards or by means of a prepared program which consists of blocks or series of numbers.
- These numbers define the required position of each slide, its feeds, cutting speeds etc.
- Also defined as a form of software controlled automation, in which the process is controlled by alphanumeric characters or symbols.
- It is a form of digital control of a machining process.
- A NC machine (machine tool + control system)

- According toEIA (Electronic institute of America), a NC machine is a system in which the actions are controlled by the direct insertion of numeric data at some point.
- System must automatically interpret at least some portion of data.

Constructional features of NC

- Machine
  - Machine units
  - Machine tools
  - Control system – electronic HW
  - Feedback components
  - Software
- Driver units

SOFTWARE

- Program or set of instruction
- Languages
- Punched cards
- Magnetic tape
- Information processing systems
- Software controls the sequence movements of an NC.
- NC machines are called as software controlled machines using numeric data.
- In NC, the numeric data which is required for producing a part is maintained on a punched type.
- Preparing the data for a NC machine tool requires a part programmer and this type of programming of a NC machine is called as part programming.

Machine Control Unit(MCU)

- Consists of electronic circuitry that reads NC program, interprets it and conversely translates it into the mechanical actions of the machine tool.
- MCU consists of 2 parts:

1. Data Processing Unit
2. Control Loop Unit

- MCU consists of 3 types:

1. Housed
2. Swing around
3. Standalone

Sub units of MCU

- Input/reader unit
- Data buffer(memory)
- Processor
- Output channels
- Actuators
- Control panel
- Feedback channels
- Transducers
- Electronic components
- Electrically operated control equipment such as the starters, relays,
- manual controls consist of push buttons, switches used by the operator

Machine Tools(MT)

- Main component which executes the operations.
- Simple drilling machine
- Includes different parts or sub-assemblies such as the work table, fitting tools, fixtures, motors for driving the spindle and the tool
- Coolant and lubricating system

Classification of NC machines

- Based on feedback control

1. Open loop
2. Closed loop
3. Semi-closed loop

- Based on motion control system

1. Point to point
2. Straight line
3. Continuous path
4. Combined motion control system

- Based on circuit technology

1. Analog control systems
2. Digital Control systems

- Based on programming type

1. Incremental
2. Absolute

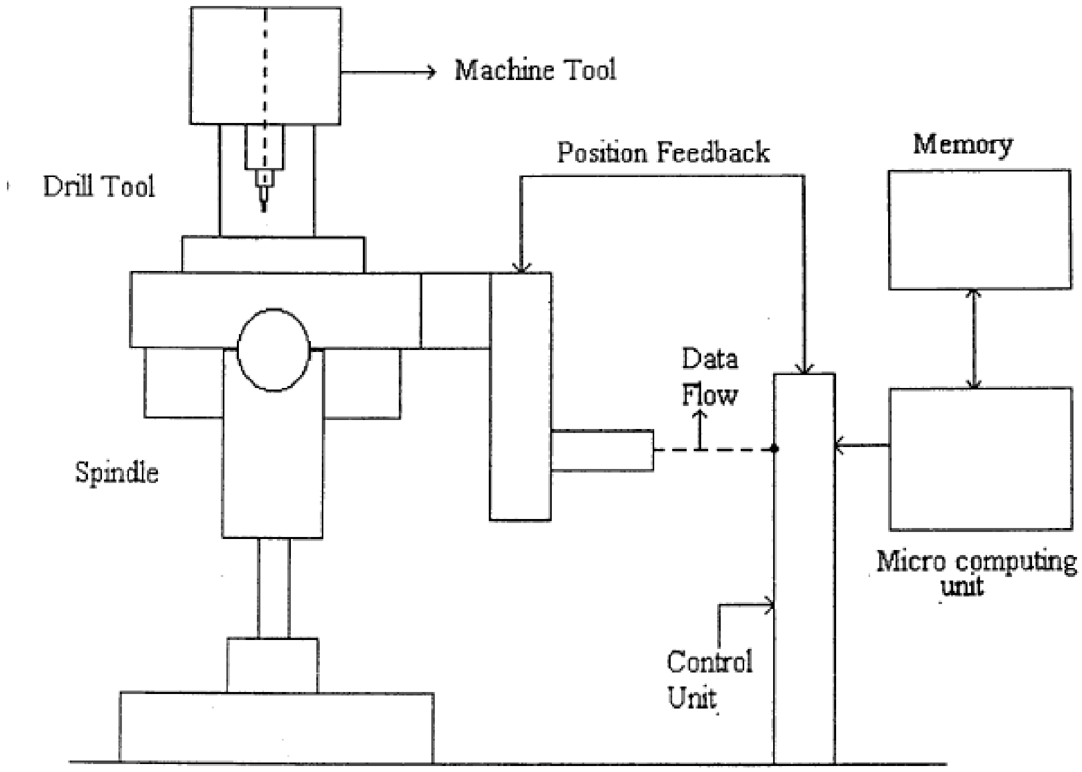
EXAMPLES:

- Drilling machines
- Grinding machines
- Boring machines
- Milling machines

ADVANATGES

- Increased flexibility
- Increased productivity
- Reduced floor space
- Lesser human error
- Reduced set-up time
- Machine utilization
- Machine accuracy
- Production of complex parts
- Lesser scrap
- Longer tool life
- Lower labour cost
- Reduction in transportation cost
- Effective machine utilization

Computer Numerically Controlled Machines(CNC)



A Computer Numerically Controlled Machine

- NC machines switched over the controlling from the electronic HW system control to the computer control due to the advent of the computers.
- Use of computers emphasized on controlling the NC with advanced software using sophisticated computers instead of HW.
- Hence this machines are called as CNC machines.
- It is a NC system where in a dedicated stored program computer is used to perform the same or all of the basic NC functions in accordance with a control program stored in the computer.
- In CNC a mini computer is used to control the machine tool functions from stored information's or punched type input or a computer terminal input.
- In short,
- CNC machine = NC machine + a computer
- CNC needs only the drawing specifications of a part to be manufactured and the computer automatically generates the part program for the loaded part.
- CNC is used when the control system of the NC system utilizes an internal computer.

Computer allows

- Storage of additional programs apart from the main programs
- Machine and control diagnostics
- Program editing
- Running the programs from memory
- Special routines, inch/metric/micron accuracy

DEFINITION:

- A self-contained NC system for a single machine tool which uses a dedicated minicomputer, controlled by the instruction stored in its memory, to perform all the basic numerical control functions.

CONSTRUCTION DESIGN

- Mechanical system design
- Control system design
- Machine tooling's design
- Software design

TOOLINGS:

- Cutting tools
  - ✓ Preset tools
  - ✓ Diamond tools
  - ✓ Solid tools
  - ✓ Qualified tools
  - ✓ High carbon steels tools
- Holding devices for tools and jobs

Drive systems

- Electrical Drive
- Mechanical load section

SOFTWARE

- A computer needed for the software control of the machine and to do the machining operation on the given part.
- Various types of programming languages are used: computer aided part programming (CAD programming) and part program.
- Part program is defined as a sequence of instructions which describe the work which has to be done on a part, in the form required by a computer under the control of a CNC computer program

ADVANTGES

- Machines are operated automatically
- Complex part shapes can have machined easily
- Reduced tooling costs
- Uniformity of production
- Used in industries where high accuracy of machined components is required
- On line editing of program
- Reduced data reading error

DISADVANTGES

- High initial cost
- Higher maintenance cost
- Not suitable for long run applications
- Machines have to be installed in air conditioned places
- Proper environmental conditions are to be maintained
- Skilled CNC personnel required to operate the system

APPLICATIONS

- Used in a wide range of manufacturing processes such as
- Drilling
- Metal carving
- Welding
- Used to manufacture small, minute parts where fine finishing/precision is required such as manufacturing of watch gears, needles

Difference between NC/CNC and Robots

Points	NC/CNC Machines	Robots
Capital investment	more	More
DOF	Fixed i.e 3	> 6
Co-ordinate system	xyz system	Xyz, cylindrical, polar/spherical, articulated type
Control system	Elec/elec HW (NC) or software(CNC)	Computers(soft automation)
sensors	no	Huge number
Mobile , stationary	not	Yes
applications	less	Huge
Environment friendly	No	Yes
Adaptability to environment	No	Yes
Construction: Rigid type	Yes	Yes
Size	huge	Small to huge

Points	NC/CNC Machines	Robots
Automation	Soft automation	Soft automation
Work applications	Cannot do all the work that a robot does	Variety of robots
Maintenance cost	High	High
Flexibility	less	More
Skilled Labour	Required	Required
Precision jobs	Can be done	Can be done
Quantity of jobs	Small	Small and bulk
Programming languages	Low level as well as high level	High level

CHAPTER 4

WORKSPACE ANALYSIS AND TRAJECTORY PLANNING

CONTENTS

INTRODUCTION

The end effector of a manipulator is required to move in a particular motion in a multidimensional space.The kinematics model is used as a tool to follow the define trajectory in order to perform meaningful manipulation tasks. We formulate the trajectory planning problem by defining the workspace and analyse the workspace of the robot, because it is in this workspace, the robot does useful work.

The main goal of trajectory planning is to describe the motion of the robot end- effector or joints as a time base sequence location and its derivatives which are generated by interpolating or approximating the desired path by polynomial functions and to achieve a smooth motion of the end-effector.

4.1 WORKSPACE ANALYSIS

Robot Workspace

The area in which the robot can do useful work like pick and place operations. The area within the total work space envelope of the robots is called workspace

Definition: Joint-Space Work Envelope (Q)

The work envelope that is traced by the joints in the joint space R^n dimensional space or the set of all values that the joint variables q can assume in the workspace is called the joint-space work envelope.

It is of the form:

$Q \triangleq \{q \in R^n: q^{min} \leq Cq \leq q^{max}\}$

4.2 Work envelope:

The work envelope Y is the locus of all points $p \in R^3$ that are reachable by the tool tip. It can be expressed in terms of Q.

It is of the form: $Y = \{ p(q) \in R^3 : q \in Q\}$

Factors on which work space of a Robot depends:

1. Robot’s physical configuration
2. Geometric link parameters
3. Type of robot
4. Number of degree of freedom

5. Type of application
6. Limits of robot joints and its movements
7. Nature of the axes.

4.2.1 Work envelope of a Five axis articulated robot

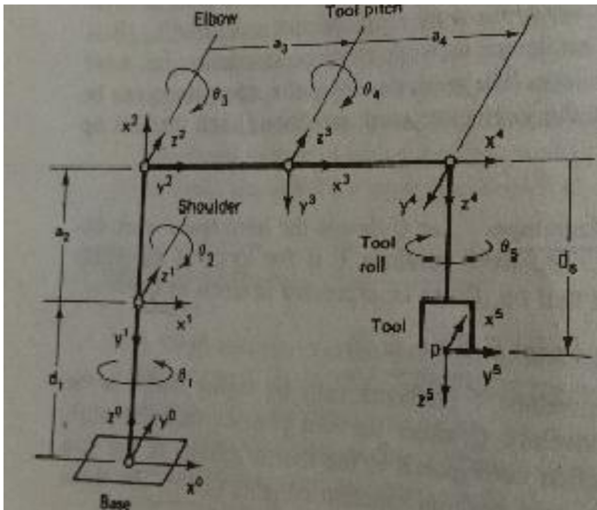


Fig 4.2.1 Link coordinates of a five axis articulated robot. (Rhino XR-3) Courtesy: Fundamentals of Robotics (Analysis and Control) by Robert Schilling

Step 1: Expression for the tool-configuration vector w (q).

The expression for the tool configuration vector of the five-axis articulated robot in fig 4.1 is:

$$w(q) = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ \hdashline w_4 \\ w_5 \\ w_6 \end{bmatrix} = \begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ \hdashline \left\{\exp\left(\frac{q_5}{\pi}\right)\right\} R_{13} \\ \left\{\exp\left(\frac{q_5}{\pi}\right)\right\} R_{23} \\ \left\{\exp\left(\frac{q_5}{\pi}\right)\right\} R_{33} \end{bmatrix} = \begin{bmatrix} C_1 (a_2 C_2 + a_3 C_{23} + a_4 C_{234} - d_5 S_{234}) \\ S_1 (a_2 C_2 + a_3 C_{23} + a_4 C_{234} - d_5 S_{234}) \\ d_1 - a_2 S_2 - a_3 S_{23} - a_4 S_{234} - d_5 C_{234} \\ \hdashline -\left[\exp\left(\frac{q_5}{\pi}\right)\right] C_1 S_{234} \\ -\left[\exp\left(\frac{q_5}{\pi}\right)\right] S_1 S_{234} \\ -\left[\exp\left(\frac{q_5}{\pi}\right)\right] C_{234} \end{bmatrix}$$

Eq. 4.2.1

The joint variables of the minor axes used to orient the tool are the tool pitch angle q_4 and the tool roll angle q_5. The tool roll angle has no effect on the tool-tip position p. However, the tool-tip position changes with the pitch angle q_4. The tool-tip position changes with the base q_1, shoulder q_2, elbow q_3 and tool pitch angle q_4. The global tool pitch angle q_{234} is measured relative to the x^0y^0 plane or the horizontal work surface.

Step 2: Joint-space work envelope equation (JSWE)

The joint angles are constrained to lie in the following JSWE equation Q of the robot given by

Q= {qk Є $R^{n,}$ $q_k^{min} \le Cq_k \le q_k^{max}$}; $1 \le k \le n$; ‘n’ is the degree of freedom DOF =5; k is particular joint of the robotic arm.

$$\begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \\ q_5 \\ q_{23} \\ q_{234} \end{bmatrix}^{min} \le C_{7\times 5} \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \\ q_5 \\ q_{23} \\ q_{234} \end{bmatrix} \le \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \\ q_5 \\ q_{23} \\ q_{234} \end{bmatrix}^{max}$$ Eq 4.2.2

$$\begin{bmatrix} -\pi \\ -3\pi/4 \\ \pi/4 \\ -5\pi/4 \\ -\pi \\ -3\pi/4 \\ -5\pi/4 \end{bmatrix} \le \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \end{bmatrix} q \le \begin{bmatrix} \pi \\ \pi/4 \\ 3\pi/4 \\ \pi/4 \\ \pi \\ \pi/4 \\ \pi/4 \end{bmatrix}$$ Eq 4.2.3

From the above equation, the following conclusions can be made

- There is no constraint on the base angle q_1 as it can range over one complete cycle [-π, π].
- The constraints on the shoulder angle $q_{2,}$ elbow angle $q_{3,}$ and tool pitch angle q_4 prevent adjacent links from colliding.
- The tool roll angle q_5 also ranges over one complete cycle [-π, π].
- The constraints on the global elbow angle q_{23} limit the orientation of the forearm relative to the work surface.

Reach constraint equation

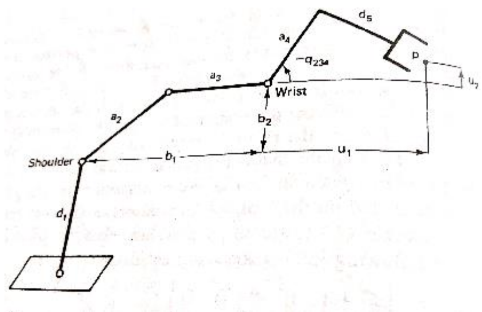


Fig: 4.2.2 evaluating reach of a five-axis articulated robot

The global tool pitch angle q_{234} which is selected to find the reach equation specifies the outer surface of the work envelope Y. The basic constraint on the reach is illustrated in the above figure which shows that the distance from the shoulder frame L_2 to the wrist frame L_4 can never be larger than $a_2 + a_3$, where a_2 is the length of the upper arm and a_3 is the length of the forearm; b_1 and b_2 are the horizontal and vertical components which are dependent on position.

I b I pronounced as mod b is the radial distance from the origin of the shoulder frame L_1 to the pitch frame L_3.

Therefore, $b_1 = a_2C_2 + a_3C_{23}$ and $b_2 = a_2S_2 + a_3S_{23}$

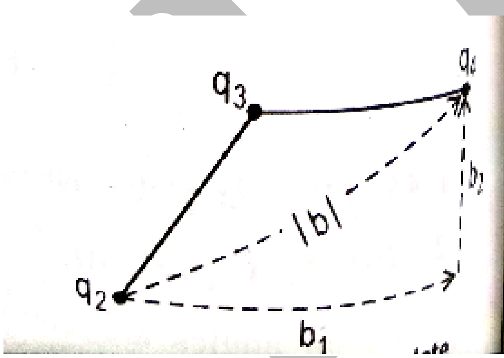


Fig: To calculate b_1 and b_2

Using Pythagoras theorem, when the two links a_2 and a_3 are straight, the reach constraint equation is given by $(b_1)^2 + (b_2)^2 \leq \| b \|^2$

Therefore, $(b_1)^2 + (b_2)^2 \leq (a_2 + a_3)^2$

The above equation specifies that the distance from the origin of L_1 coordinate frame to the L_3 coordinate frame cannot be greater than $(a_2 + a_3)$

4.2.2 Different reaches in a Rhino robot

There are two types of reaches in a rhino robot

1) Horizontal reach 2) Vertical reach
1. Horizontal reach : It is defined as the distance the arm can travel in the horizontal direction
2. Vertical reach : It is defined as the distance the arm can travel in the vertical direction

Horizontal stroke is the difference between maximum and minimum horizontal reach.

Vertical stroke is the difference between maximum and minimum vertical reach.

*The combination of all the four reaches gives the work space envelope of a Rhino robot.

4.2.2.1 Reach constraint

For an articulated robot like rhino robot, the tool-tip position p must satisfy the following reach constraint once the global tool pitch angle q_{234} has been selected.

$[(p_1^2 + p_2^2)^{1/2} - u_1]^2 + (p_3 - d_1 - u_2)^2 \leq (a_2 + a_3)^2$

a) Limits on p_1
   From the above reach constraint equation, the maximum value of p_1 occurs when
   $(b_1)^2 + (b_2)^2 \leq (a_2 + a_3)^2$ and $p_2 = 0$ and $p_3 = d_1 + u_2$ be minimum.
   Therefore, the upper bound on p_1 for a given tool orientation q_{234} is
   $| p | = u_1 + a_2 + a_3$
b) Limits on p_2
   Once the first coordinate of p is selected as maximum, p_2 automatically becomes equal to zero.
   The robot is in the xz plane and the constraint on p_3 gets reduced to $d_1 + u_2$.
c) Limits on p_3
   Once q_{234}, p_1 and p_3 is selected, the bound on p_3 reduces to $p_3 \leq (d_1 + u_2)$

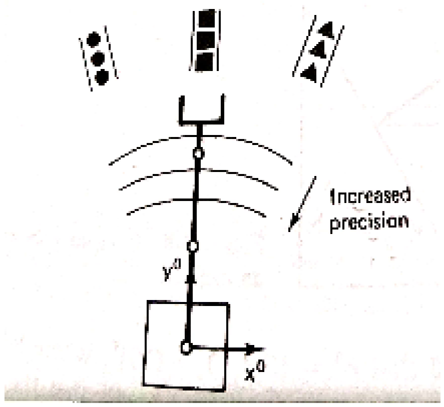
4.3 Workspace fixtures

Workspace fixtures are accessories and add-on of a robot. These fixtures form a part of a robot to perform a particular task. Work space fixtures used in the work space of the robot are:

1) Part feeders 2. Transport devices 3. Part-holding devices

1.) Part feeders – Devices used to feed the parts to the robot.

For the object to be successfully grasped by the robot, the part must be presented to the robot in the manner compatible with the tool configurations realizable by the manipulator. Part feeders are usually arranged concentrically around the robot base near the outer boundary of the work envelope. Part feeders are activated electromechanically and electrically. Part feeders activated electrically are basically vibratory-bowl devices and those that are activated electromechanically are gravity fed part feeders.



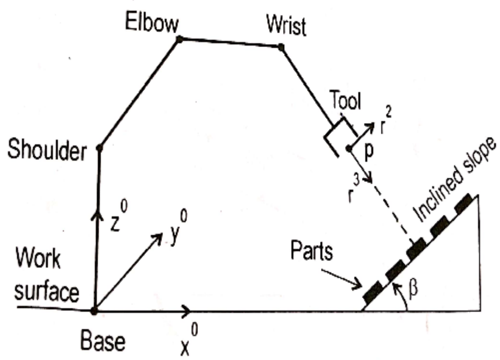


Fig: Concentric layout of multiple part feeders   fig: A gravity fed part feeder

2.) Transport devices- Devices which are used to move parts from one robotic work cell to another work cell. Conveyors and Carousels are examples of transporting devices.

Conveyers and carousels are special part feeders. They transport parts between robots and from one robotic work cell to another linearly. The conveyor belt transports the parts horizontally either from left to right or from right to left but in one direction only. They are driven by electric motors.

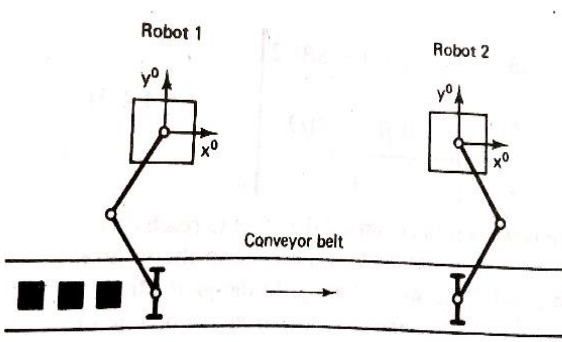


Fig: A conveyor belt as a linear transport device

Carousels are rotary transport devices. It transfers the parts back and forth between two or more than two robots. It consists of a rotating table with a central pedestal.

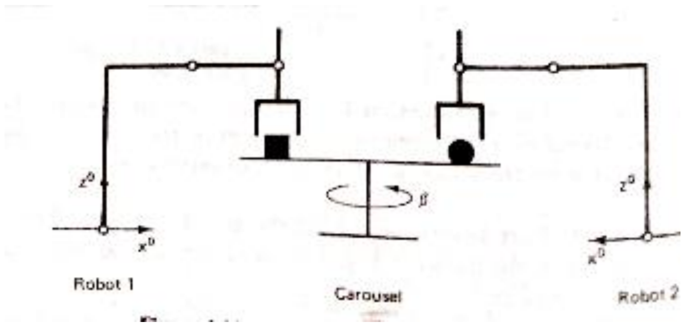


Fig : Carousel

From the figure, the parts can be easily approached from above. Parts can be easily exchanged between the two robots even when the carousel can turn in only one direction.

Advantage: 1) Conveyor can easily accommodate many robots.

2) Carousel is limited to relatively few robots; otherwise the diameter of the turntable would become too large.

3.) Part-holding devices- Devices which are used to hold a sub-part in exact position and orientation. It is mainly used in assembly line task. Most robots are one-handed and have only one tool in operation at any given time. Work-holding fixtures are needed to place the subassembly part at a workstation. The robot then completes the manipulation task given by adding the new part to the sub-assembly.

A work-holding fixture is considered as a fixed tool. It is fixed and immobile. It cannot move or reorient itself. Examples: jigs, clamps etc.

Example: Robot performing nut fastening operation

Suppose the robot has already picked up the bolt and placed it in a fixed tool, or computer-controlled clamp. While planning the nut-fastening trajectory for the tool, the nut will be twisted using the tool roll motion of the robot. The approach vector must be constant and pointing perpendicular to the work surface x^0y^0 plane. During the threading operation, sliding vector and normal vector gets twisted in threading operation.

To determine the speed of the threading operation

Suppose it takes T seconds to roll the tool one complete revolution

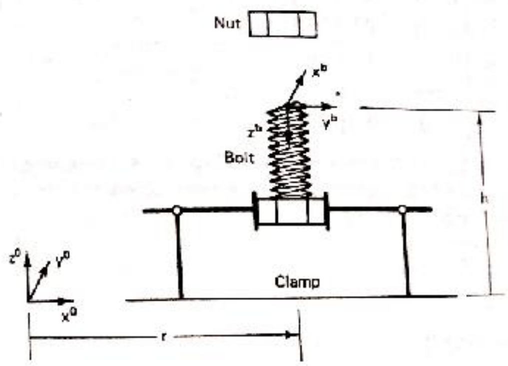


fig: A fixed tool for part holding

If a screw transformation is applied to the bolt frame B= { x^b , y^b ,z^b } as shown in the fig above. The angular displacement of the screw is defined by ϕ(t)

Therefore, $\phi(t) = 2\pi t / T$ radians

To determine the linear displacement of a screw transformation, the tool has to slide down fast down the bolt when the nut is turned in the clockwise direction. If the pitch of the bolt is β threads/mm and the tool performs 1 rotation in 1 sec, then the linear displacement of the screw is given by

$\lambda(t) = t / \beta T$ mm , if the length of the bolt is b mm long, total threading time t = β b T seconds.

Position and orientation of the threading frame:

Threading operation is done w.r.t. to the bolt using a screw transformation matrix

$$T_{bolt}^{thread}(t) = \text{Screw}[\lambda(t), \phi(t), 3] \qquad 0 \leq t \leq b\rho T.$$

Screw transformation is the product of rotation and translation. The homogeneous transformation matrix which maps bolt coordinates into base coordinates is given by

$$T_{base}^{bolt} = \begin{bmatrix} 0 & 1 & 0 & r \\ 1 & 0. & 0 & 0 \\ 0 & 0 & -1 & h \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The homogeneous screw transformation matrix is given by

$$
\begin{aligned}
T_{base}^{thread}(t) &= T_{base}^{bolt}\, T_{bolt}^{thread}(t) \\
&= T_{base}^{bolt}\ \text{Screw}\,[\lambda(t), \phi(t), 3] \\
&= T_{base}^{bolt}\ \text{Rot}\,[\phi(t), 3]\ \text{Tran}\,[\lambda(t) i^3] \\
&= \begin{bmatrix} 0 & 1 & 0 & r \\ 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & h \\ 0 & 0 & 0 & 1 \end{bmatrix}
\begin{bmatrix} \cos[\phi(t)] & -\sin[\phi(t)] & 0 & 0 \\ \sin[\phi(t)] & \cos[\phi(t)] & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \lambda(t) \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
&= \left[\begin{array}{ccc|c} \sin(2\pi t/T) & \cos(2\pi t/T) & 0 & r \\ \cos(2\pi t/T) & -\sin(2\pi t/T) & 0 & 0 \\ 0 & 0 & -1 & h - t/\rho T \\ \hline 0 & 0 & 0 & 1 \end{array}\right]
\end{aligned}
$$

4.6 THE Pick-and-Place Operation

The main and the most fundamental function of any robotic manipulator is to pick up an object from one place which is in one position and orientation and place it in another place in its position and orientation. The pick-and-place motions are used to alter the distributions of parts within the workspace. This operation is basically used for loading and unloading of machines. The robot during its pick-and-place operations moves along a particular path known as trajectory in a particular time. The path taken by the robot from pick point to the place point is called as four discrete point minimal pick and place trajectory.

4.4.1 Pick and Lift-Off points

Pick point is the first point in the pick-and-place trajectory .The coordinate frame is denoted by

T_{base}^{pick}.This represents the initial position and orientation of the part being manipulated. The pick position p^{pick} is considered to be center of mass or centroid. The object is picked up using the approach vector r^3 which is perpendicular to the work surface x^0y^0 plane on which the part is resting and the sliding vector r^2 moving inwards along the closing axis of gripper or tool. d^{pick} represents the vertical distance from the horizontal work surface to the pick point of the object.

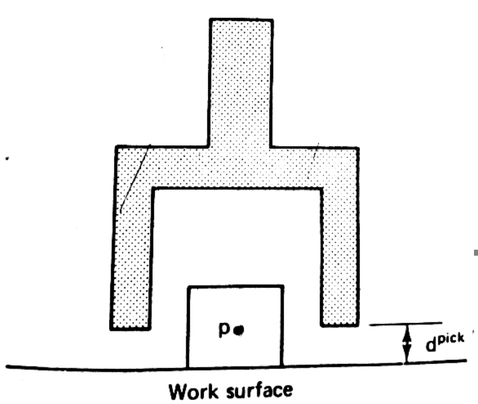


Fig: Tool configuration at the pick point

Lift-off points

The second point of pick-and-place trajectory is lift-off points. The coordinate frame is denoted by T_{base}^{lift}. The lift-off point is a point near pick point that the robot move to, before the robot reach down to pick up the part. The lift-off point is an intermediate point that is inferred from the pick point. The tool orientation at the lift-off point is identical to the tool orientation at the pick point. The tool orientation remains fixed as the tools moves from the lift-off point to the pick point. The tool position at the lift-off point is obtained by starting at the pick position and moving backward a safe distance v along the approach vector, away from the pick point.

$$T_{base}^{lift} = \left[\begin{array}{ccc|c} & R^{pick} & & p^{pick} - \nu R^{pick} i^3 \\ \hline 0 & 0 & 0 & 1 \end{array}\right]$$

Equation for lift-off point frame

Place and Set-down points

Place point is the third point of the pick-and-place trajectory. The coordinate frame is denoted by T_{base}^{place}.This represents the final position and orientation of the part being manipulated.The place orientation R^{place} is selected in such a way that the approach vector r^3 is orthogonal to the surface on which the part will come to rest. The distance d^{place} between the place position and the place surface, measured along the approach vector should be identical to the distance between the pick position and the pick surface.

$$d^{place} = d^{pick}$$

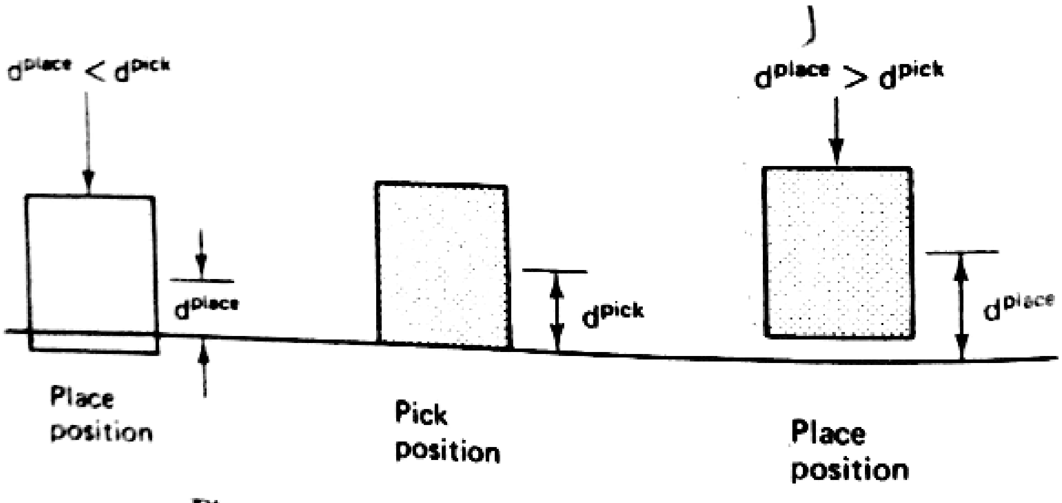


Fig: Constraints on pick and place positions

1) $d^{place} = d^{pick}$

   Object is picked up from the work surface and placed exactly touching the work surface.

2) $d^{place} < d^{pick}$

   Robot penetrates the object into the work surface when part is placed, as a result of which the part slides in between the fingers of the grippers.

3) $d^{place} > d^{pick}$

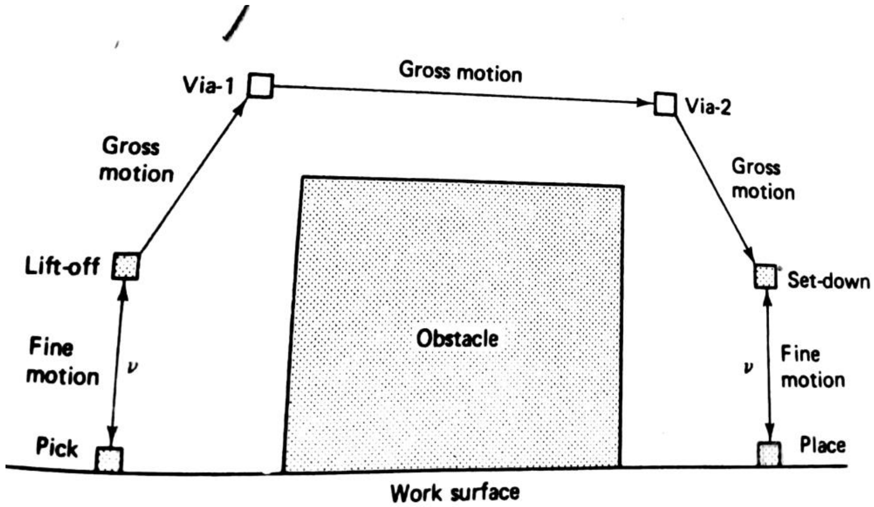
   The placed part is unsupported when it reaches the destination and the robot opens its fingers and the object or the part falls down because of gravity.

Set-down point

The last point in the pick-and-place trajectory is the set-down point, denoted by T_{base}^{set} .It is a point near the place point that the robot moves to before an attempt is made to place the part. The set-down point is an intermediate point that is inferred from the place point. The set down point can be computed as:

$$T_{base}^{set} = \left[\begin{array}{ccc|c} & R^{place} & & p^{place} - \nu R^{place} i^3 \\ \hline 0 & 0 & 0 & 1 \end{array}\right]$$

Pick-and-Place trajectory



The sequence of motions needed to execute a pick-and-place operation with minimal four point trajectory. If part feeders, conveyors ,carousels are used to present parts from the robot, then the pick-and-plane surfaces are at different levels or at different orientations. If the workspace contains obstacles, then to avoid collisions one or more via points can be inserted between lift-off and set-down points.

Speed Variation

For the robot to do a particular task, speed is the most important consideration. The robot controller has to be designed in such a way so as to control the speed. The gross motion between the lift-off and set-down points can be carried out at high speed for efficiency. The fine motions near the two end points of the trajectory should be carried out at reduced speed.

Summary of a typical pick-and-place sequence

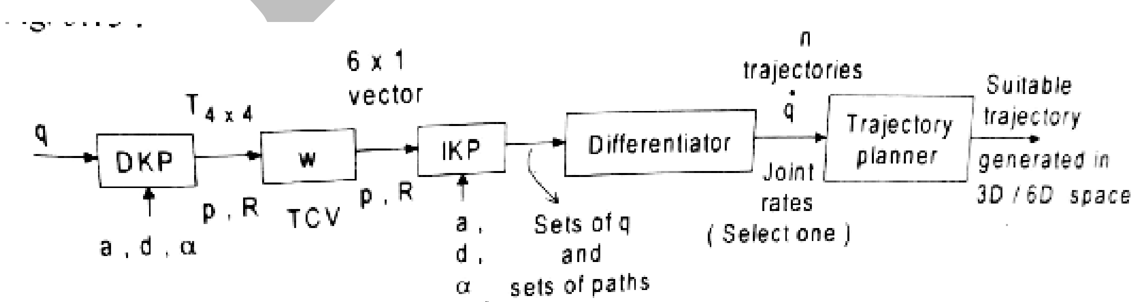
Destination	Type of Motion	Speed
Lift-off	Gross	Fast
Pause	Fine	Zero
Pick	Fine	Very slow
Grasp	Fine	Slow
Lift-off	Fine	Slow
Via	Gross	Fast
Set-down	Gross	Fast
Pause	Fine	Zero
Place	Fine	Very slow
Release	Fine	Slow
Set-down	Fine	Slow

4.5 Continuous Path Motion

This motion or trajectory is exhibited by the continuous path robots. It is known as continuous path motion. Once the path is specified, the robot moves continuously along the specified path. The trajectory path is explicitly specified by the user and the robot moves continuously along the specified path. The robot end-effector moves along a particular path in 3D space specified by the user and the speed of the motion may vary. Continuous path robots have got the capability to follow a smooth curved path. Controlling the path of motion is difficult since all the joints have to be activated and controlled simultaneously to keep the desired tool orientation along the specified path.

Examples: Spray painting, arc welding, line welding, inspection of parts along assembly line etc.

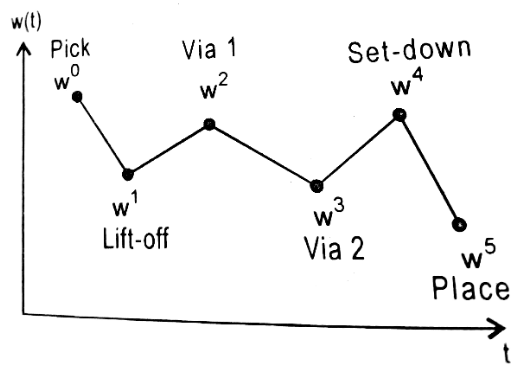
The block diagram of a continuous path motion control is shown in the following figure



Continuous path can be calculated for any five axes, four axes, three axes articulated robot.

4.6 Interpolated Motion

Interpolated motion is a type of motion exhibited by point to point robots. In many instances, the path will not be completely specified. Knot-points such as intermediate via-points and end points will be specified. The trajectory planning interpolate between the knot points to produce a smooth trajectory that can be executed using continuous-path motion control techniques.



In a general interpolation problem, the sequence of m knot points in tool configuration space is given by $\Gamma_m = \{ \mathbf{w}^0, \mathbf{w}^1, \ldots, \mathbf{w}^{n-1} \}$

The path w (t) between the m knot points should be smooth. It should have minimum two continuous derivatives in order to avoid the need for infinite acceleration.

4.6.1 Cubic Polynomial Paths

Consider a simple path i.e. from w^0 to w^1 an interpolated motion between two knot points w^0 to w^1.

The equation for the path from w^0 to w^1 is represented by a cubic polynomial given by

$w(t) = a\,t^3 + b\,t^2 + c\,t + d$; $0 \le t \le T$; $T > 0$; where w(t) is a cubic polynomial trajectory in Tool Configuration Space R^6 and a, b, c, d are the unknown polynomial coefficients which has to be determined.

Once, the equation for the path is determined, the robot moves along the specified path.

Interpolation:

There are four constraints on the four unknown coefficients of the cubic interpolating polynomial.

Consider the equation $w(t) = a t^3 + b t^2 + c t + d$

At the start of the path, $t = 0$ in $w(t)$, Putting $t=0$ in $w(t)$

$w(0) = d = w^0$

At the end of the path, $t = T$, Putting $t = T$ in $w(t)$

$w(T)= a T^3 + b T^2 + c T + d = w^1$

If the tool starts with velocity of v^0 at the beginning of the trajectory i.e. $t=0$ and ends with velocity v^1 at the end of trajectory $t= T$, then the two end-point velocity constraints are

Velocity $= \frac{d}{dt}[a t^3 + b t^2 + c t + d]$

Therefore, $\dot{w}(t) = 3 a t^2 + 2 b t + c$

At the start of the path, $t= 0$, Putting $t =0$ in $\dot{w}(t)$; we get

$\dot{w}(0) = c = v^0$

At the end of the path $t =T$, Putting $t =T$ in $\dot{w}(t)$; we get

$\dot{w}(T) = 3 a T^2 + 2 b T + c = v^1$

To get the values of a and b; solving by trial and error method

For finding the value of a ;

1) Add v^1 to v^0 and multiply the sum by T

2) Subtract w^0 from w^1 and multiply the result by 2

We get,

$$a = \frac{T(v^1 + v^0) - 2(w^1 - w^0)}{T}$$

For finding the value of b;

1) Take v^1, add it to twice v^0 and multiply the result by T^2

2) Subtract w^0 from w^1 and multiply the result by 3

3) Result of step 1 and result of step 2, divide by T^2

4) Take negative of the entire result

We get,

b= $-\frac{T[(v^1 + 2v^0) - 3(w^1 - w^0)]}{T^2}$

4.6.2 Linear Interpolation with parabolic blends

Piecewise linear interpolation between the knot points effectively decouples the original m-point problem into m-1 separate two –point problems

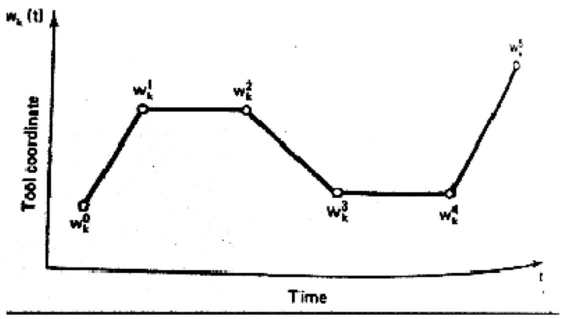


Fig: Piecewise-linear interpolation between knot points

Advantage: Piecewise-linear interpolation is computationally very efficient when the number of knot points is maximum two and the path generated is linear.

Disadvantage: The path is not smooth.

Since the velocity changes every knot point the tool passes, a infinite instantaneous tool acceleration is required.

Solution: To overcome this disadvantage, piecewise linear interpolation with parabolic blends is used for a smooth path.

4.6.2.1 Linear Interpolation with parabolic blends

or infinite .

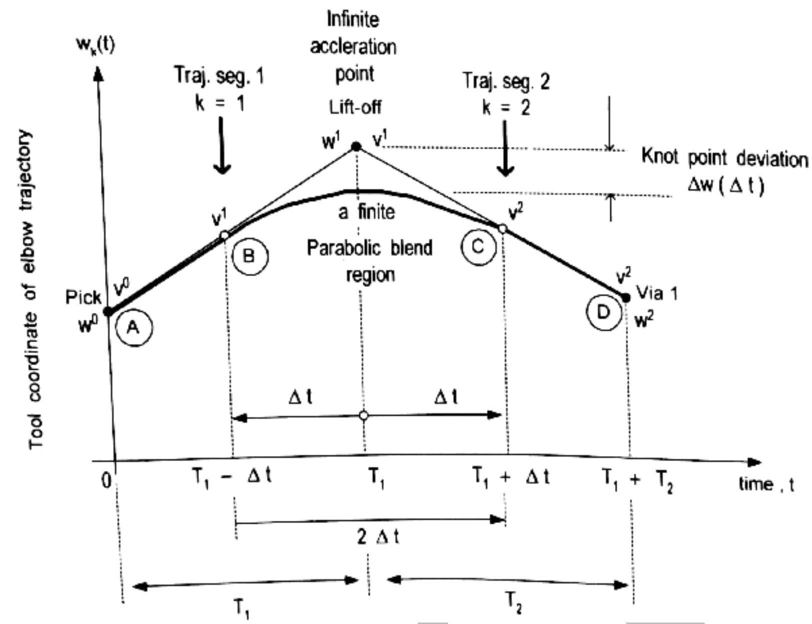


Fig: Piecewise linear interpolation with parabolic blends

Consider two-segment trajectory. Suppose the segment $\{w^{k-1}, w^k\}$ is to be traversed in time T_k constant velocity v^k for $1 \le k \le 2$.

Velocity with which the segment is traversed is given by

Velocity = $\frac{Distance\ travelled}{Time\ Taken}$ = $\Delta w^k / T_k$

If $v^2 \ne v1$, then an ∞ acceleration is required at time T1 to achieve the sudden change in velocity (note that there is a sudden direction change at w^1). To obtain this sudden change in direction, very high acceleration is required at time T_1, which is highly dangerous to the robotic system. Hence, to avoid this, apply a constant or a finite acceleration starting at $t = (T_1 - \Delta T)$ and ending at time $t = (T_1 + \Delta T)$ by making use of a parabolic blend, so that there is smooth transition in velocity v^1 at time $(T_1 - \Delta T)$ velocity v^2 at time $(T_1 + \Delta T)$ and the acceleration is finite and not zero or infinite.

During the blend portion of the trajectory, constant acceleration is used to achieve a smooth velocity transition. The two parabolic blends are assumed to be identical to the left and right side of the knot point. Therefore, blends neat the path points are of the same duration and the whole trajectory is symmetric about the half way point in time and position. Therefore, the trajectory gives continuity of position, velocity and acceleration throughout.

This will place the tool at time $(T_1 + \Delta T)$, at the same point where it would have been if the original piecewise linear path and been followed as a result of which, the two velocities $v^2 = v^1$, the two points B and C are the same and the ∞ acceleration factor will be eliminated. If $v^2 = v^1$ then, there is a smooth transition in velocity and acceleration at the knot point is avoided. During the parabolic blend, acceleration is finite and the robot follows a parabolic path given by a quadratic equation w(t).

The parabolic path w(t) represented by a quadratic polynomial over the transition period, is given by

$$w(t) = \frac{a(t - T1 + \Delta T)^2}{2} + b(t - T1 + \Delta T) + c$$

Solving the quadratic equation, we get

$$a = \frac{\text{Change in velocity}}{\text{Total time taken}} = \frac{T_1 \Delta w^2 - T_2 \Delta w^1}{2 T_1 T_2 \Delta T}$$

$b = \frac{\Delta w^1}{T_1}$; since velocity at the start of the blend is $v^1 = \frac{\Delta w^1}{T_1}$

$c = w(T_1 - \Delta T)$

4.6.2.2 Knot Point Deviation

It is defined as the deviation of the parabolic path from the knot point ,the distance from w^1 at time T_1 to the parabolic path.

Knot point deviation is given by : $\Delta w(\Delta T) \rightarrow 0$ as $\Delta T \rightarrow 0$

$\Delta w(\Delta T) = w(T_1) - w^1$

$$a = \frac{T_1 \Delta w^2 - T_2 \Delta w^1}{2 T_1 T_2 \Delta T}$$

From the above equation for acceleration, the following conclusions are devised

1. When $\Delta T > 0$; acceleration is finite
2. When ΔT goes on decreasing, a goes on increasing which implies the parabolic blend trajectory moves towards the knot point.
3. When $\Delta T = 0$, $a = \infty$ and parabolic trajectory tends to the original linear piecewise trajectory.

4.7 Straight Line Motion

Straight line motion of the robot payload represents the shortest distance between two points in the workspace. The straight line motion from the source to the goal covered in a specific amount of time is known as straight line trajectory. It is always required in Tool Configuration space R^6. Inverse kinematics equation is used to get straight line motion.

Applications of Straight Line Motion

- Conveyor belt operations
- Inserting peg into a hole
- Straight line arc welding
- Screw transformations
- Inserting electronic components onto PCB

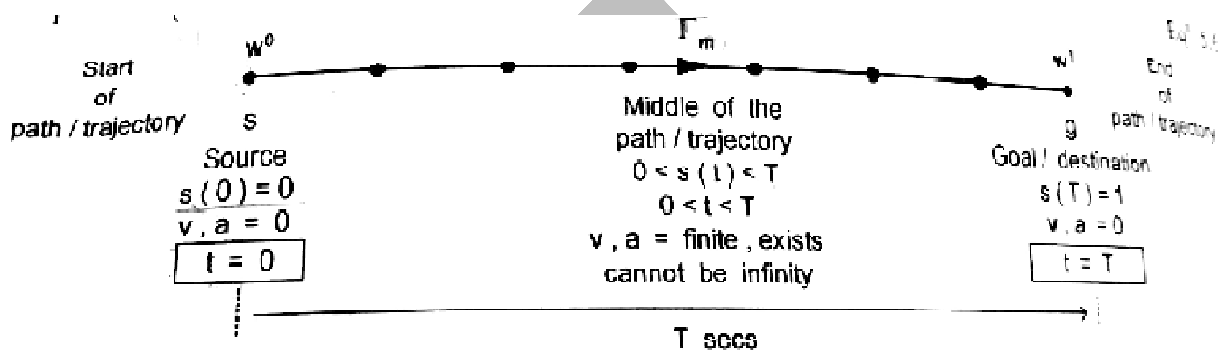


Fig: Straight line path

The equation for straight line path is given by ,

$w(t) = [\, 1 - S(t)\,]w^0 + s(t)\, w^1$; $0 \le t \le T$

s(t) is speed distribution function given by $s(t) = \frac{t}{T}$ which maps into (0,T)

At the start of the trajectory, t= 0

$w(t) = w^0$

At the end of the trajectory, t = T

$w(t) = w^1$

4.7.1 **Bounded Deviation Algorithm**

It is an algorithm which is used to obtain an approximated straight line motion in tool configuration space R^6 by using an articulated robot by selecting the number of knot points, minimizing the knot points and distributing them along the trajectory in an optimal manner.

Since the inverse kinematic equations have to be solved at each knot point, it is desirable and necessary to minimize the knot points and distribute the m along the trajectory in an optimal manner.

The technique used for selecting the knot points is proposed by Taylor and the method is known as Bounded Deviation algorithm. It is used for approximating straight line motion with a point-to-point robot

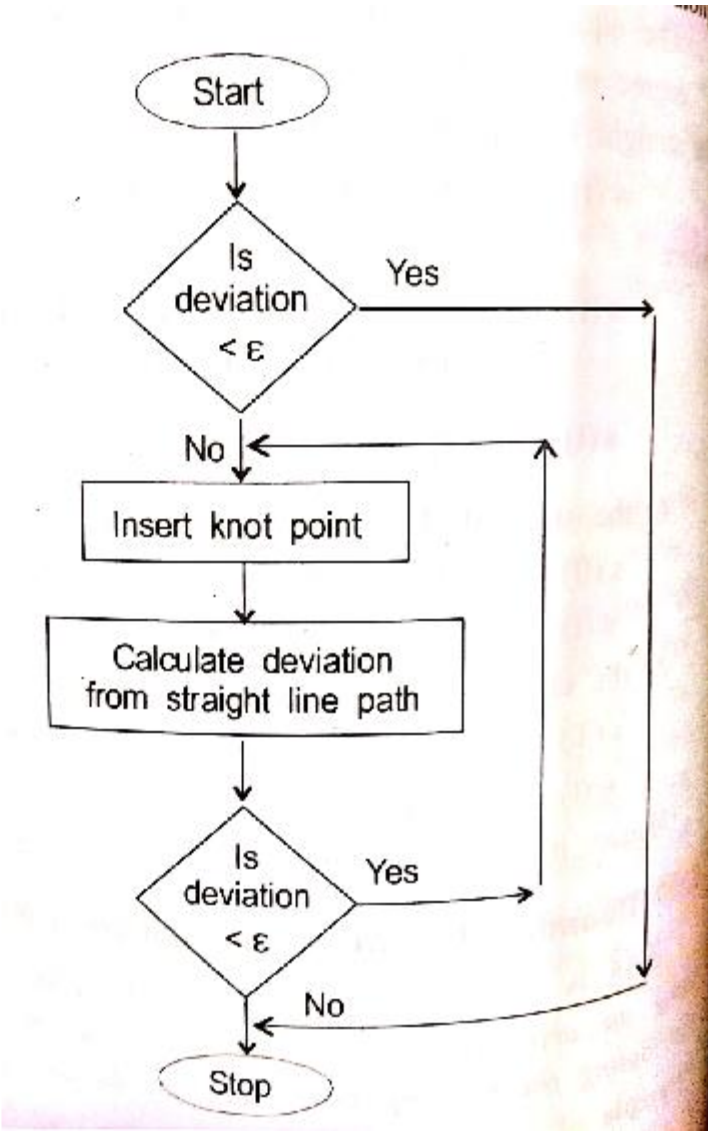


Fig: Flowchart Bounded Deviation

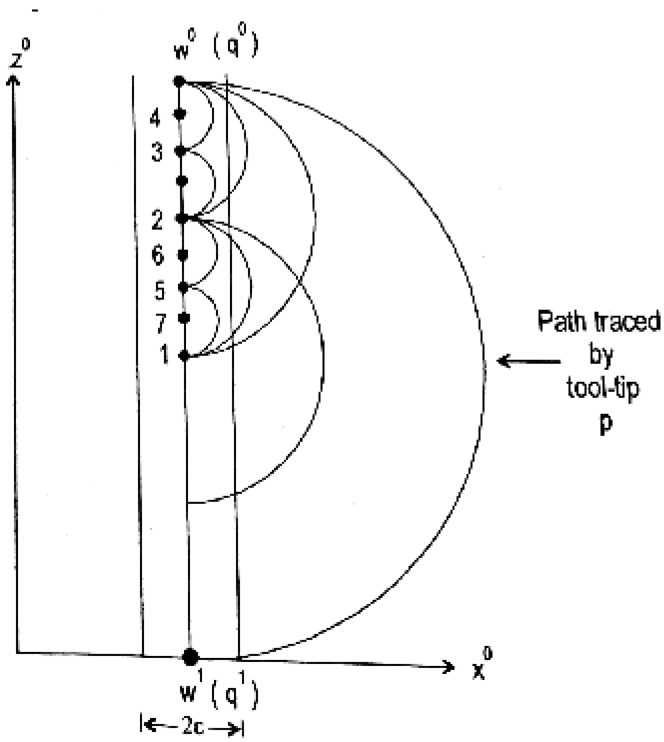


Fig: Interpolation of joint space approximation to the straight line motion

Algorithm for obtaining straight line motion

1. Select Ɛ > 0. Use the inverse kinematics equation to compute { q^0 , q^1 }, joint vectors associated with { w^0 , w^1 }.
2. Compute the joint-space midpoint $q^m = (q^0 + q^1) / 2$
3. Use q^m and the tool- configuration function w to compute the associated tool-configuration space midpoint : $w^m = w(q^m)$
4. Compute the exact tool-configuration space midpoint
   $W^M = (w^0 + w^1) / 2$
5. If the deviation $\| w^m - w^M \| \le Ɛ$, then stop; else ,insert w^M as a knot point between w^0 and w^1.
6. Apply the algorithm recursively to the new two segments { w^0 , w^M } and { w^M , w^1 }.

   The algorithm does not distribute the knot points uniformly; it will depend on Ɛ, the robot geometry and the location of the straight line path within the workspace.

Questions:

1. What is workspace analysis? Define Joint-Space work envelope.
2. Give the work envelope of a five-axis articulated robot with a neat labeled diagram.
3. Explain the reach constraints for an articulated robot.
4. What are workspace fixtures? Explain
5. Write a short note on Conveyors and Carousels.
6. What are fixed tools? Explain fixed tool for part holding operation.
7. Explain briefly the pick-and-place operation.
8. State the speed variation of a pick-and-place sequence.
9. Discuss about continuous-path motion in brief.
10. What is interpolated motion? How it is calculated?
11. Explain linear interpolation with parabolic blends.
12. Write a short note on Knot-point deviation.
13. Briefly explain straight line motion.
14. State the Bounded Deviation algorithm.

5 ROBOT VISION

5.1 Introduction

Robot vision is defined as the process of capturing, extracting, characterizing and interpreting the information obtained from images through a camera or a charge coupled device.

A basic robot vision system has a single stationary camera mounted over the workspace. Several stationary cameras are used in order to gain multiple perspectives. This is an active form of sensing in which the robot is used to position the sensor on the basis of the results of pervious measurements.

Robot vision systems supply valuable information that can be used to automate the manipulation of objects. With the use of robotic vision, the position, orientation, identity and condition of each part in the scene can be obtained. The information obtained can then be used to plan robot motion so as to determine how to grasp a part and how to avoid collisions with obstacles.

5.1.1 Image Representation

Goal: To examine how the visual data from the camera is represented.

- **A camera is a device which converts a 3D physical object in to a 2D image of the object.**
- **The depth information is lost, objects becomes inverted and scaled and is analog in nature.**
- **The 2D image is the output of the camera and is a visual data.**
- **The image is made of spatial coordinates (x, y) and the value of the spatial coordinate represents the reflected light intensity or the brightness that spatial coordinate (x, y).**

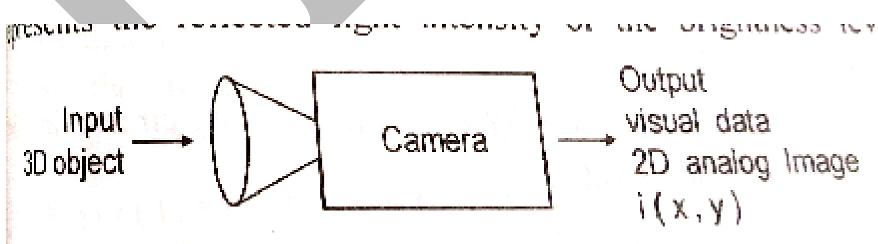


Fig: Input and Output of a Camera

- **A computer cannot process the analog images or the raw images i(x, y). It has to be converted to digital image. The process of converting an analog image into its equivalent digital image is called as digitization of images.**
- **Digitization is a three step process: sampling, quantization and coding of images.**

- **Digital Image I(k ,j) is defined as an analog image i(x,y) which is sampled to a spatial resolution of (m x n) pixels and quantized i.e. discretized both in spatial coordinates and in brightness.**
- **A digital image is represented as a matrix of cells of width Δx and height Δy.**
- **It is made up of number of rows and columns, the size of the image being (m x n).Each row/column of the image is called as cell or a pixel.**

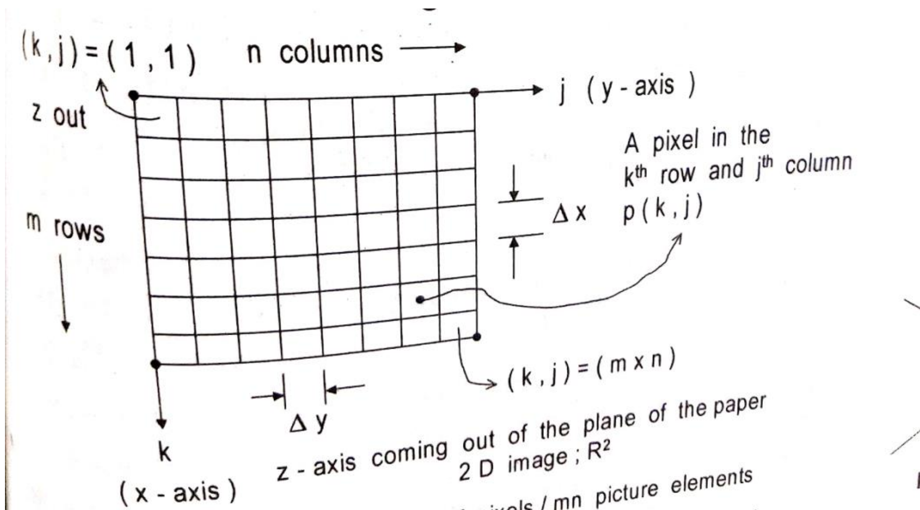


Fig: A 2D digital image

The average light intensity over the picture element is given by

$$I_a(k,\ j) \triangleq \frac{\int_0^{\Delta x}\int_0^{\Delta y} i[(k-1)\Delta x + x,\ (j-1)\Delta y + y]\, dy\, dx}{\Delta x\, \Delta y}$$

5.2 Template Matching

Template matching is a technique which is used to recognize whether a given part belongs to a particular class of parts.

The characteristic features of each class of parts are placed in front of the camera, one at a time, and their images are obtained and stored in a library of parts.

LIBRARY = LIB = {Ti (k, j): 1 ≤ k ≤ m0 , 1 ≤ j ≤ n0 , 1 ≤ i ≤ N}

The i^{th} representative image T_i (k, j) is referred as template or mask for a class i.

The objective is to search for a translation (x, y) which will minimize the performance index ρ_i (x,y).

If $\rho_i(x, y) = 0$, then an exact match has been detected. The algorithm determines the part i contained in the image I (k, j) and also the location of the part (x, y) is also identified.

Algorithm for Template matching

1. Initialize i= 1, x =0, y =0, $\varepsilon > 0$, found = true.
2. Compute performance index ρ_i (x,y) using

$$\rho_i(x, y) \triangleq \sum_{k=1}^{m0} \sum_{j=1}^{n0} \left| I(k + x, j + y) - T_i(k, j) \right| \qquad 1 \le i \le N \qquad (8\text{-}$$

3. If ρ_i (x,y) $\le \varepsilon$, stop
4. Set x = x+ 1.If x ≤ m – m0, go to step 2
5. Set x =0, y = y +1 .If y ≤ n –n0, go to step 2
6. Set x =0 ,y = 0, i = i + 1.If i ≤ N , go to step 2
7. Set found = false

Advantages of performance index method

1. Whether a part belongs to a particular class of parts or not can be found out.
2. Location of a part can be found out.
3. Used in pattern recognition

Disadvantages of performance index method

1. Getting best match is difficult due to noise in analog image, gray scale/digital image
2. Due to proper illumination
3. Due to sampling and quantization noise
4. Presence of shadows
5. Difference in the intensity levels of the image and the template
6. It can be affected by scaling, rotations and perspective changes of the part.

5.3 **Polyhedral objects** (Edge Detection and Corner point detection)
Robots always manipulate polyhedral objects. Objects having more than two surfaces are called as polyhedral objects.

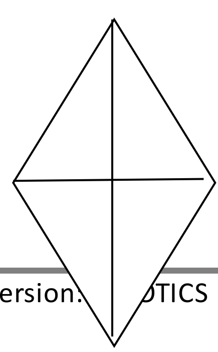


Fig: A simple polyhedral solid

5.3.1 Edge Detection

The reflected light intensity will be constant or vary continuously over each face of the object, since the faces of the polyhedral object are smooth , homogenous and opaque. However, the reflected light intensity will have discontinuity at the boundary between adjacent faces.

At the edges of the polyhedral object, the gradient of the light intensity $\nabla i(x, y)$will be infinite.

Digital approximation of the gradient operator is required to locate an edge.

To detect an edge, the magnitude or size of the gradient vector is computed.

$|| \nabla I(k, j) || = [\nabla_1^2(k, j) + \nabla_2^2(k, j)]^{½}$

An edge is present at pixel (k, j) if the magnitude $|| \nabla I(k, j) ||$ is greater than some edge threshold Ɛ > 0.

The threshold Ɛ should be small enough to pick up all the edges, and it must be large enough to avoid false edges due to noise in the image.

Algorithm in Edge Detection

1. Initialize k = 1 , j = 1 , Ɛ > 0
2. Compute $|| \nabla I(k, j) ||$ as per the equation above
3. If $|| \nabla I(k, j) ||$ ≥Ɛ, set L (k, j) = 1; otherwise set L(k,j) = 0.
4. Set j = j+ 1.if j < n, go to step 2.
5. Set L (k, n) = 0.
6. Set j = 1, k = k + 1. If k < m, go to step 2.
7. For j = 1 to n , set L(m, j) = 0

Working:

Consider an m x n gray scale image as an input and produce an m x n binary image as output, where 1s represent edges.

The first m -1 row and n- 1 columns are scanned for edge pixels. The key parameter is the edge threshold Ɛ.

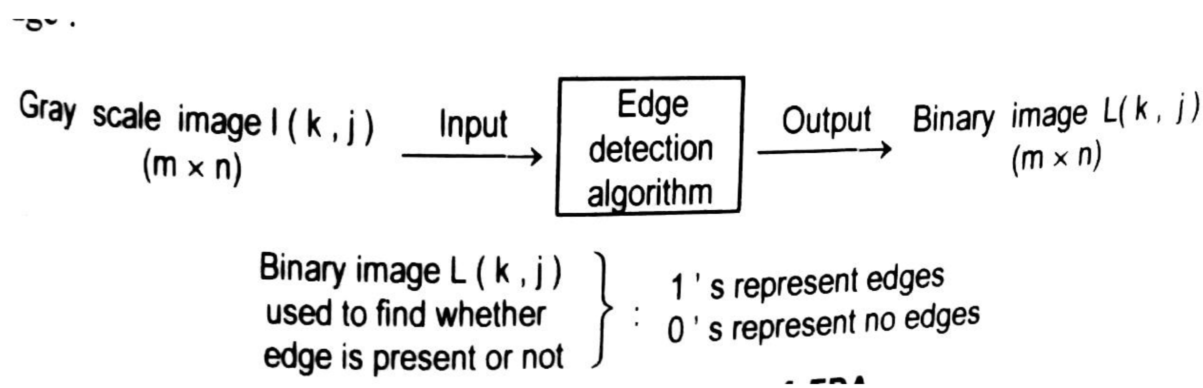
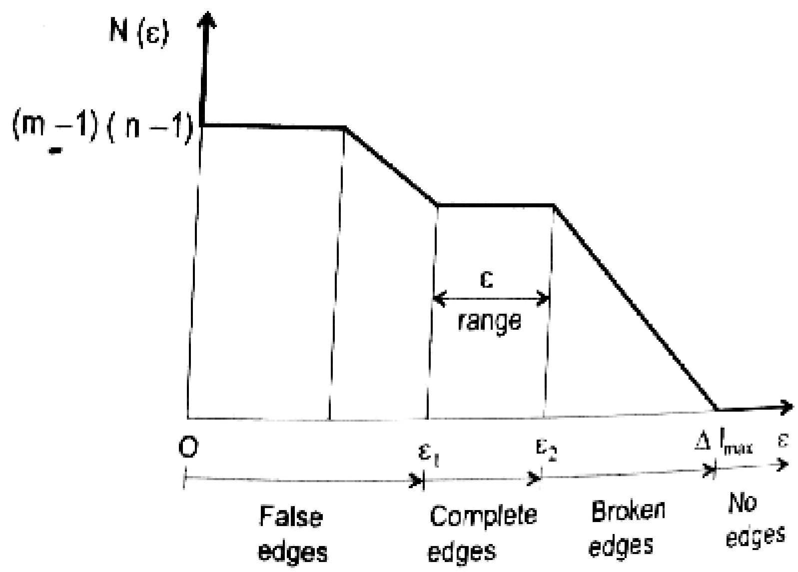


Fig: Block diagram of edge detection algorithm

Edge pixel distribution



Edge threshold has to be selected properly to recover all the correct edge pixels of the object in the image.

When

1. Ɛ = 0; All scanned pixels are edge pixels , we get false edges
2. Ɛ is increased from zero ; false edges begins to disappear from the binary image
3. Ɛ is between $Ɛ_1$ and $Ɛ_2$; all edges of the object will appear in the binary image
4. Ɛ is increased further; broken edges appear
5. Ɛ is very large; threshold very high; no edge pixels are detected.

Conclusion: In order to receive all the edge pixels of the object in the image, edge threshold has to lie between $Ɛ_1$ and $Ɛ_2$

5.3.2 Corner Point Detection

It is a method of determining the corner points or vertices of an object in the gray scale image which has only two gray levels. Each of the interior pixels has eight adjacent pixels called its neighbors.

By examining the intensity pattern of the neighbors, we can determine whether a given pixel is a corner point or vertex. This can be done by scanning over the image with 3 x 3 corner-point templates or masks which represents all possible types of corner-points.

The set of corner points templates is generated by taking the corner pattern appearing in the upper right portion of the first template and rotating it counterclockwise to generate the seven templates in multiples of $\frac{\pi}{4}$.

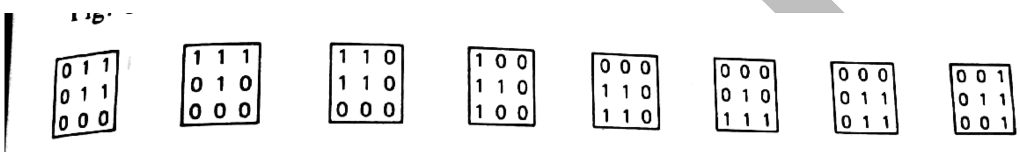


Fig : Corner point templates

To search for the corner points, scan the image with the templates using normalized cross-correlation function. We first scan the (m- 2) (n-2) pixels in the interior of I(k ,j) to find candidate pixels , pixels with a value of 1. Once a candidate pixel is located, its eight neighbors can be calculated by cross-correlating with 3 x 3 templates. If the normalized cross correlation evaluates to 1; the process can be terminated since a corner-point pixel has been identified.

5.4 SHAPE ANALYSIS

Shape analysis is a method of finding the shape of irregular objects using two types of descriptors. It is used when the objects are not polyhedral. They are line and area descriptors.

5.4.1 **Line descriptors**: They are used to find out the length of the irregular boundary or curve of an irregular object in terms of pixels and make use of two encoding names namely absolute representation and relative representation.

Chain code

A curve C (a) is represented by a sequence of chain codes a ∈ $R^{n,}$ where n is the length of the curve in pixels.

Chain code is more efficient than storing the coordinates of the points because it takes only 3 bits per point.

Example

Consider the following image

0	0	0	0	1	1	0	0
0	0	1	1	0	0	1	0
0	1	0	0	0	0	1	0
0	1	0	0	0	0	1.	0
0	0	1	0	0	1	0	0
0	0	0	1	1	0	0	0

3	2	1
4	p.	0
5	6	7

1. Mark a dot ". " at the right most pixel of the binary image.(already marked in the image)
2. The chain code is generated as follows
   a = [2,2,3,4,5,4,5,6,7,7,0,1,1]
   C (a) = 13 pixels i.e. the length of the curve is 13 pixels.

5.4.2 Area Descriptors

The descriptors which are based on the analysis of points enclosed by the boundary are called area descriptors. Area descriptors are more robust than line descriptors.

Consider R representing a region in an image I(k,j) . R is a connected set that is for each pair of pixels in R, there is a path in R which connects the pair. R corresponds to a single part but may have holes. Foreground is represented by 1's and background region is represented by 0's. A circular object with a hole inside is also present.

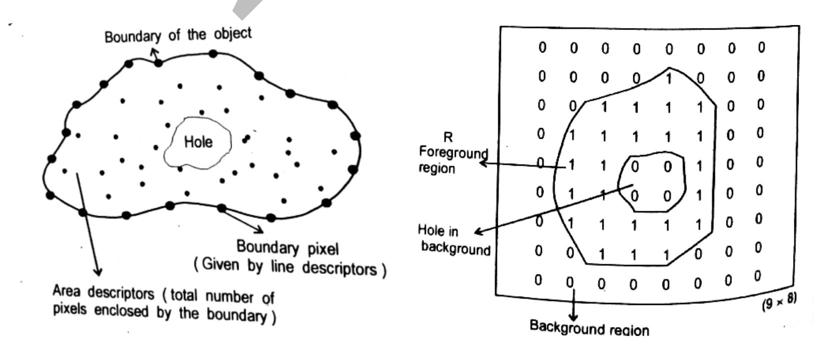


Fig: Area descriptors

5.4.3 Shape analysis using moments

To analyze and characterize the shape of the foreground region R, we compute moments. Moments gives the characteristic features of the objects such as shape, area, centroid, moment of inertia and orientation of the object in the image.

Moments: It is defined as a sequence of numbers m_{kj}, which are used to characterize the shape of the foreground object in an image.

$$m_{kj} \triangleq \sum_{x,y \in R} x^k \, y^j$$

Physical Significance

Let {mkj} be the ordinary moments of the foreground region R of a binary image. Let A be the area of the region R and $\{x_c, y_c\}$ be the position of its centroid of the region R .

Then,

Area = A = m_{00}

$x_c = \frac{m_{10}}{m_{00}}$

$y_c = \frac{m_{01}}{m_{00}}$

(x_c, y_c) = Centroid = $\{ \frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}} \}$

- Zeroth order m_{00} moment gives the area of the foreground region R
- m_{10} gives the first order moment along x-axis
- m_{01} gives the first order moment along y-axis

Central moments of a foreground region R

Central moments are so called because they are obtained using the centroid (x_c, y_c). The central moments of foreground region R are ordinary moments translated by an amount equal to the centroid so that the centroid coincides with the origin, as a result of which the first order central moments are equal to zero and are invariant to translations of the foreground region R

Central moments $\mu_{kj} = \sum_{x,y \in R}(x - x_c)^k \, (y - y_c)^{\,j}$; (k, j) ≥ 0

The physical significance of the central moments is that they give area and moment of inertia and they are invariant to translations but variant to scale changes and rotations.

5.4.4Principal angle

Lower order moments, central moments and normalized central moments characterize the region R and are invariant to translations and scaling of R, but are variant to rotations of the foreground region R.

Invariance to rotation of R is obtained by finding the principal angle ϕ. It is a measure of the orientation of the region R. It can be expressed in terms of second-order central moments.

Principal angle of R is defined as $\phi = \frac{1}{2} \text{atan2} (2 \mu_{11}, \mu_{20} - \mu_{02})$

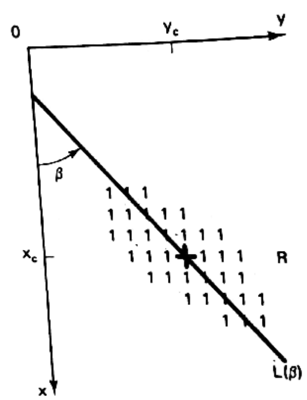


Fig: Principal angle of a region R

To interpret the principal angle, consider a line L(β) drawn through the centroid of R at an angle of β with respect to the x-axis. The moment of inertia will depend on the angle β. The angle at which the moment of inertia is minimized is the principal angle β = ϕ. The principal angle is defined for elongated object.

5.5 SEGMENTATION

The process of separating out the parts in an image into distinct connected regions is called segmentation. It involves partitioning the image into connected regions, each region being homogenous. The set of all connected pixels with a particular gray level attribute is identified as a region and is assigned a unique label.

5.5.1 Thresholding

Gray level threshold is the simplest way to segment a gray scale image into background and foreground. It converts the gray scale image into a binary image where foreground is represented by 1's and background region by 0's.

The thresholding is not very useful when the image is having multiple foreground objects of varying gray levels. Alternative method of segmenting is performing gray scale histogram.

5.5.2 Region Growing and labeling algorithm

Step 1: Select any pixel as 'seed' which will be a non-background pixel .Check its gray level and assign it a label 1.

Step 2: Evaluate each unlabelled non-background pixel in neighborhood of the seed pixel. Assign same label to all the neighboring pixels having same gray level.

Step 3: Select one of the neighboring pixels with the same label and call it the seed, go to step 2.if none of the neighbors of all the pixels in the region has same gray level go to step 4. If no unlabelled pixel is found, go to step 6.

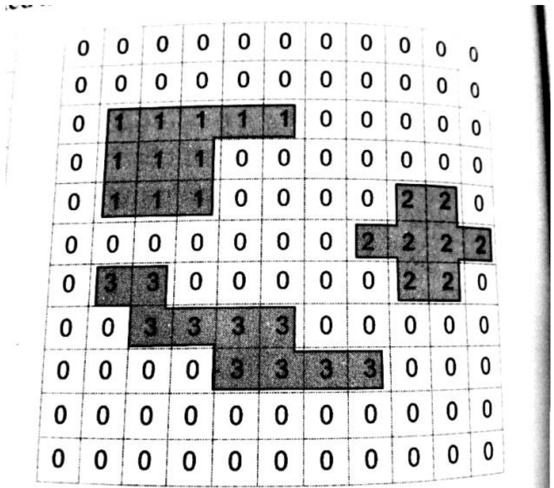
Step 4: Select any unlabelled non-background pixel with a gray-level as seed and assign it the next label. If no such pixel is found then go to step 6.

Step 5: go to step 2

Step 6: The image scan is over and the image is segmented into regions identified by the pixels of the same label.

0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	1	1	1	1	1	0	0	0	0	0
0	1	1	1	0	0	0	0	0	0	0
0	1	1	1	0	0	0	0	1	1	0
0	0	0	0	0	0	0	1	1	1	1
0	1	1	0	0	0	0	0	1	1	0
0	0	1	1	1	1	0	0	0	0	0
0	0	0	0	1	1	1	1	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0

(a) A binary image



(b) Segmented image after applying region growing

5.6 ITERATIVE PROCESSING

Binary images contain noise due to variety of sources. Shadows and lighting can cause the boundary regions to appear jagged, with narrow appendages and inlets.

5.6.1 Shrink Operators

Shrink operators are iterative operators which convert 1 into 0 i.e. converts a foreground pixel which is present in the background of the object into a background pixel.

If p(k , j) is the pixel function evaluated at (k , j) and 1(.) is a unit step function, then the shrink operator acting on the pixel p(k,j) of a digital image is defined as

Shrink (i). I (k,j) = I(k,j) AND 1(i- 1 –[8 – p(k ,j)]) ; 0≤ i ≤ 8

p(k , j) returns the number of foreground pixels

[8 – p(k ,J)] returns the number of background pixels.

5.6.2 Swell operators

Swell operators are iterative operators which convert 0 into 1 i.e. convert background pixel which is present in the foreground of an object into a foreground pixel and are dual of shrink operators.

If p(k , j) is the pixel function evaluated at (k , j) and 1(.) is a unit step function, then the swell operator acting on the pixel p(k,j) of a digital image is defined as

Swell (i). $I(k,j) = I(k,j)$ OR $1(p(k,j) - 1])$; $0 \le i \le 8$

$p(k, j)$ returns the number of foreground pixels

5.6.3 Euler number

Euler number is defined as the number of parts minus the number of holes in a given image. The part is a connected foreground region and a hole is an isolated background region enclosed by a part.

Euler number of A = 1 -1 =0; B = 1 – 2 = -1; C= 1- 0= 1

Connectedness

A pixel is 4-connectedness to its neighbors if and only if at least one of the four pixels to the east, north, west , south has the same value. A pixel is 8-connected to its neighbors if and only if at least one of its eight neighbors has the same value.

Foreground region: 4-connected

Background region : 8-connected

Skeleton operator

An iterative operator which shrinks the image as much as possible while preserving the Euler number is called skeleton operator.

When skeleton operator is applied to a simple region, it reduces to a point and whereas if it is applied to a region with a single hole, the final result is a thin ring.

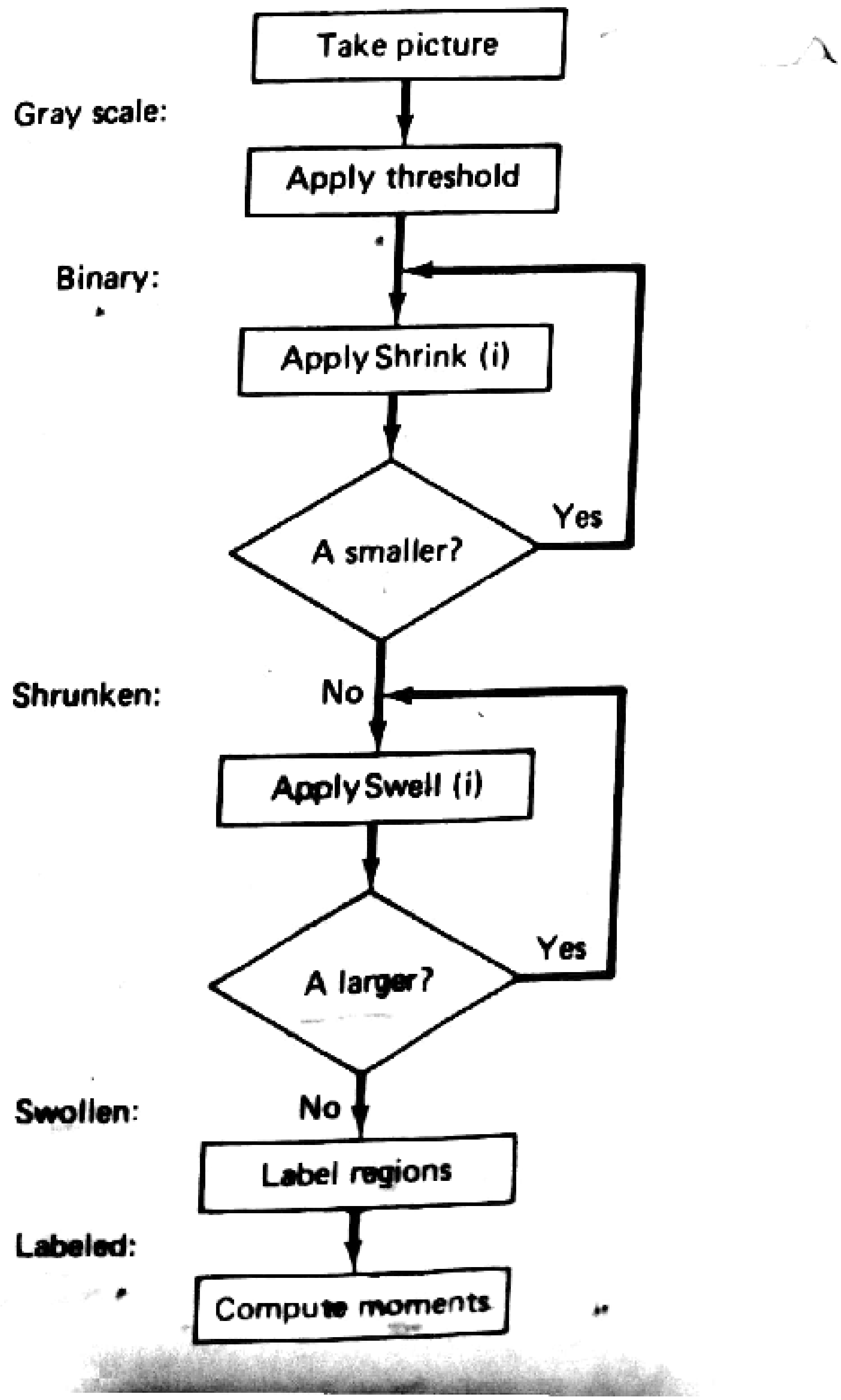
Bulk operator

An iterative operator which swells the image as much as possible while preserving the Euler number is called bulk operator.

It is the dual of skeleton operator

They are used to highlight defects in parts. However, they are not effective in removing salt-and-pepper noise from an image.

5.7 IMAGE ANALYSIS



Steps in image analysis

1. The analog image is obtained using a camera. The image is converted into a digital image using digitization.
2. Using thresholding, the gray scale image is converted into binary image which consists of foreground and background pixels.
3. Shrink and Swell operators are used to remove noise in the binary image.
4. The noise free region is then labeled using region labeling and growing algorithm.
5. Shape analysis is carried out using area/line descriptors
6. Moments are computed.

5.8 PERSPECTIVE TRANSFORMATIONS

A robot vision system can be used to identify the coordinates of certain points on a part. When a camera is used to do automated manipulation task the coordinates of the camera or the coordinates of the image will come in the perspective vector η^T in the Homogenous coordinate transformation matrix. Hence, the perspective vector will not be zero and the global scaling factor will be less than 1. The camera converts a 3D object into a 2D object, image becomes inverted, depth information is lost and the object is scaled or compressed ($\sigma < 1$)

$$T = \begin{bmatrix} Rk(\theta) & p \\ \eta T & \sigma \end{bmatrix}$$

5.8. 1 Inverse perspective transformation

The key to using robot vision to plan the motion of a robotic arm is the inverse perspective transformation. It gives the 3D information of the object w.r.t. the camera coordinate frame. Perspective transformation gives only (x, y) coordinate of the object. With inverse perspective transformation, the depth information can be recovered.

5.9CAMERACALIBRATION:

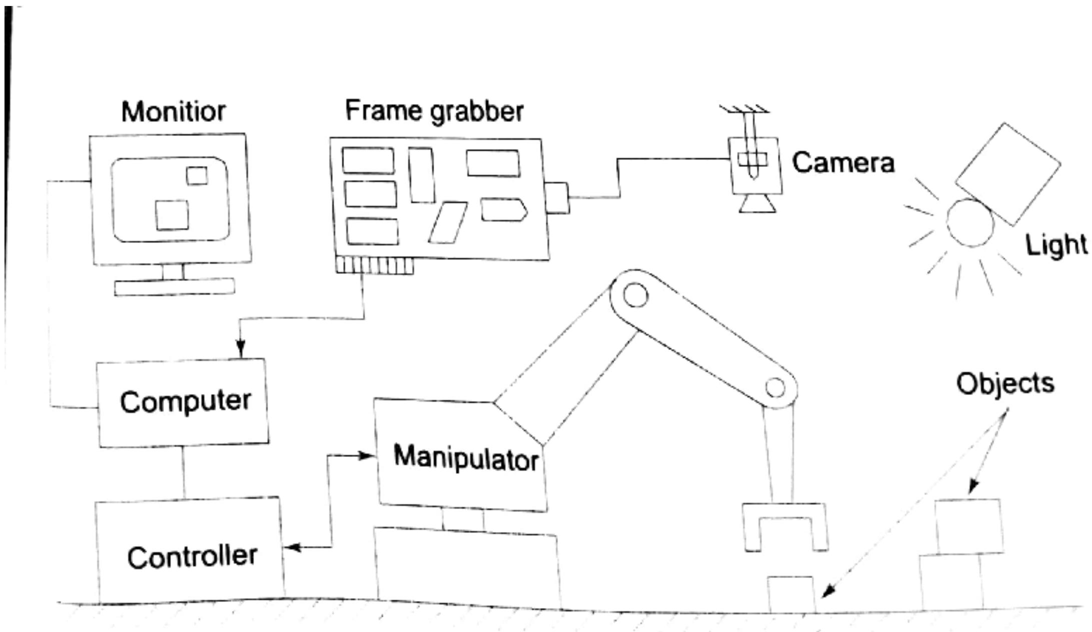


Fig : A robotic vision system

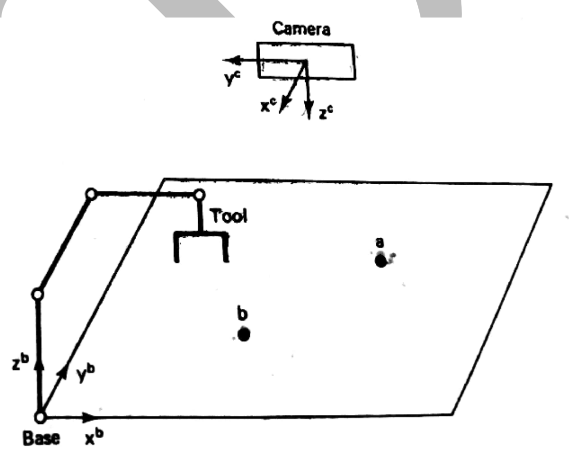


Fig : Camera Calibration

The main objective of robotic applications is to determine the position and orientation of each part relative to the base frame of the robot. Once this information is known, the proper tool configuration can be selected and then the joint-space trajectory can be computed, to manipulate the part. With the aid of robot vision, we can determine the position and orientation of a part relative to the camera. To compute this, accurate transformation from camera to base coordinates is required. Determining this information is called camera calibration. Camera calibration requires determining the position as well as the orientation of the camera.

6 TASK PLANNING

6.1 Task –level programming

6.2 Uncertainty

6.3 Configuration space

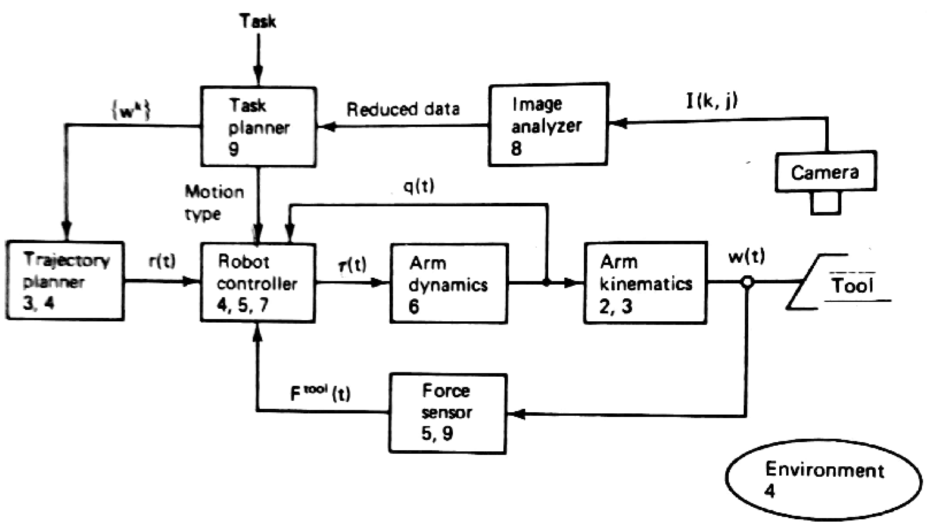
6.4 Gross-Motion planning

6.5 Grasp planning

6.6 Fine-Motion planning

Introduction

A task is a job or an operation that has to be done by the robot, whether it is a stationary robot or mobile robot. The term 'Planning' means deciding on a course of action before acting. The way the task has to be performed in its workspace has to be planned, before the robot does a particular task. This is called Task planning. In a highly sophisticated robot environment such as in automatic production plants, assembly lines, higher level of planning of robot motion is required. The planning at this level is called as task planning. The robot task planning requires the task planners to achieve these goals.



The primary input to the task planner is a task specification supplied by the user. To plan the motions needed to accomplish a task, a task planner uses an internal world model of its environment plus on-line data from sensors, specially the vision system. The raw images I (k ,j) are first processed by an image analyzer in order to reduce the data to a form that is more usable by the task-planning software.

Once a specific movement is planned in the form of a discrete tool-configuration trajectory $\{w^k\}$, the information is sent to the task planner. The trajectory planner uses interpolation and inverse kinematics techniques to convert the trajectory to a continuous-time joint-space reference trajectory r(t). The joint-space trajectory then serves as input to the robot controller which contains a torque regulator for the joints of the robotic arm. The feedback data can then be used by the robot controller to implement complaint motion and guarded motion as required.

6.1 Task Level Programming

Robot programming is done at a level that requires detail knowledge of the robot characteristics, the manipulation task, and the environment in which the task is to be performed. There are three general approaches to robot programming.

1. Teach-pendant method
2. Lead-through method
3. On-line and off-line programming

The robot is guided through the desired motions manually. The robot controller records the motions, which are play backed and edited as needed until the desired motion is obtained.

Drawbacks:

1. It requires human operator to program the robot.

2. It takes the robot out of useful service

3. The parts or objects being manipulated are to be presented to the robot at the same positions and orientation each time.

4. The teach method is an open-loop type of approach which does not incorporate uncertainties in the environment.

5. Off-line programming requires detailed specification of the manipulation task and the layout of the parts in the workspace.

To overcome the drawbacks of all the three approaches, task level programming was introduced. It is a higher-level programming technique.

In this approach, a series of goals specifying the desired positions and orientations of the objects being manipulated are supplied by the programmer. The task specification is robot-independent. The task planner, however requires a detailed model, knowledge base, ,robot characteristics, its environment and the manipulation task .

6.2 UNCERTAINTY

Uncertainty in robot task planning knows the exact position and orientation of the objects/parts being manipulated in the external environment of the robot. An uncertainty always occurs when a robot is being made to do a particular task. The factors for uncertainty are various non-linearities that are present in the robotic system such as tolerances, backlash, weight of the system, mass, inertia, friction, resistance etc.

Uncertainty reduces the positional accuracy of the system. In uncertainty, the variables represent the exact position and the orientation of the parts will have a nominal value with an error term added known as uncertainty. The exact position and orientation of the part which are represented by exact variables

$$v^{exact} = v^{nominal} + \Delta v$$

$$\| v \| \leq \Delta v^{max}$$

The error bound Δv^{max} represent tolerances in the size of a machined part. A strategy must be devised which allows for completion of the task inspite of the uncertainties in the world model. For example, if a command is given to the robot to move at 10 cm/sec ,it will not move at 10 cm /sec but it will be moving at 9.9 cm/sec. $\pm$ 0.1 is the uncertainty or the tolerance band.

6.3 Configuration Space

During a Pick and place operation, the robot grasps the object from the source using fine motion, lifts it up using the fine motion ,transports it from the life position to the set down position using the gross motion and keeps the object in the place position using the fine motion. Both the source and goal points should be within the workspace envelope of the robot. This fine and gross motion path should be planned first and take care of all the obstacles.

The task planner must address gross-motion path planning. The objective is to plan the path to move a part from given source position and orientation s to a desired goal position and orientation g in the presence of obstacles. The motion is gross such that the mobile part is assumed to be free of contact with other parts at both the ends of the path. The planned path should be optimal such as the shortest selected path stays far away from collisions or tries to avoid collisions.

Any path planning problem requires a search in six-dimensional space , as a moving part always has three translational and three rotational degrees of freedom.

A fundamental analytical tool for solving motion-planning problems is the configuration space framework developed by Lozano-Perez.

Definition:

Configuration of a part

A configuration of a part is a set of parameters which uniquely specify the position of every point on the part.

Configuration of a space

The configuration space is defined as the set of all the possible configurations of the mobile part that is obtained around the obstacle.

6.3.1 Translations

The configuration space will be two –dimensional or three-dimensional depending upon whether or not the mobile polygon is allowed to rotate.

Consider a scene with two polygonal parts

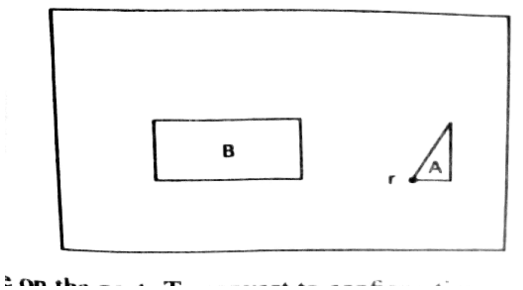


Fig: A scene with two polygonal parts

Suppose triangle A is the mobile part and rectangle B is an obstacle. A can translate but not rotate. The mobile part might represent a payload being manipulated by the robot. For a mobile part A as the figure above, consider a reference point r somewhere on the part. To convert to configuration space , shrink the mobile part to its reference point r while simultaneously growing the obstacles to compensate. By sliding part A along the boundary of obstacle B and tracing out the locus of points traversed by the reference point r which makes the obstacles grow in case of a pure translations.

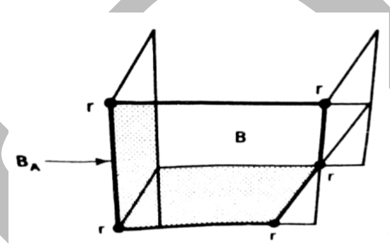


Fig: Generating Configuration space obstacle

From the figure, triangle A is shrunk to its reference point r, rectangle B grows into a five- sided polygon B_A . It becomes enlarged obstacle B_A as a configuration space obstacle induced by A . The robot workspace can be regarded as a hole in a large obstacle. Therefore , the walls of the workspace has grown inward which corresponds to sliding triangle A along the boundary of the workspace which can be referred from the figure :

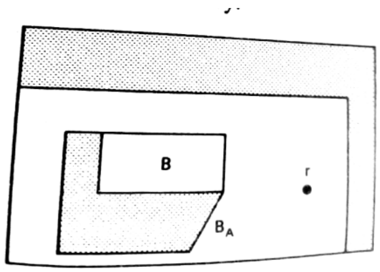


Fig: Configuration space induced by part A

As the reference point of triangle A stays outside the configuration space obstacle $B_{A,}$ triangle A will not collide with obstacle B. Hence the configuration space obstacle contains the original obstacle as a subset. The mobile part and the obstacle are both convex and so is the obstacle. Configuration space is very effective when applied to purely translational motion in a plane as the shortest path can readily be found.

6.3.2. Rotations

When the mobile polygon is allowed to rotate, the configuration-space obstacles are no longer polygonal. The surfaces of configuration space obstacles are curved in the orientation dimension. To handle rotations, enlarged polygons which encloses the mobile part over a range of orientations is employed.

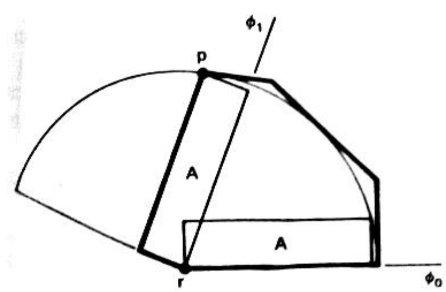


Fig: Enclosing the rotating part in a convex polygon

The configuration space obtained is no longer a convex polygon but an enlarged polygon which is curved and encloses the mobile part over a range of orientations. The advantage of rotational approach of generating the configuration space obstacle is, it converts the problem of finding a path for a part that rotates and translates in to a simple problem of finding a path for an enlarged part that only translates. Since, the rotated polygonal part will be contained in a circular sector of radius d and angle $\Delta\phi \geq (\phi_1 - \phi_0)$. If a path is found, then the mobile part can assume any orientation in the interval $\{\phi_0, \phi_1\}$ at each point along the path.

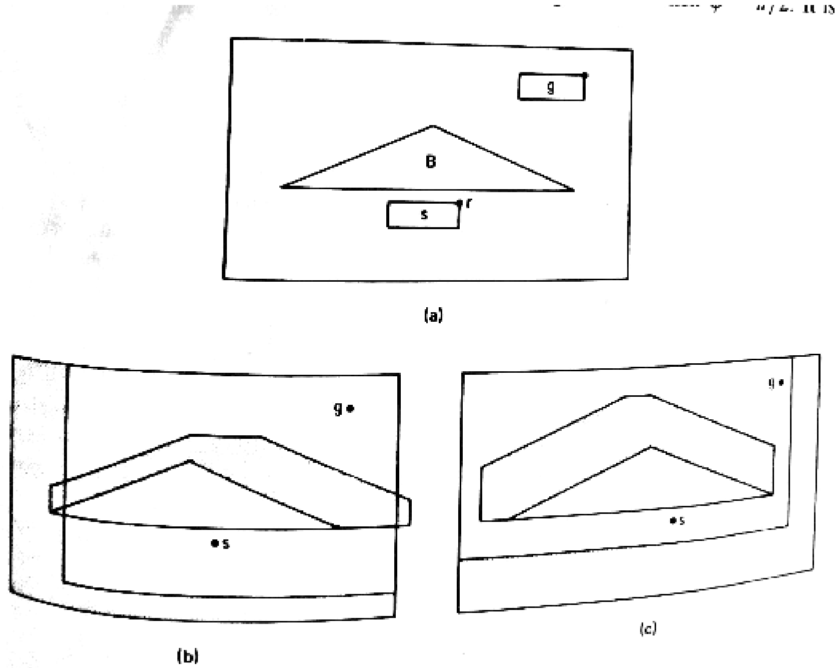


Fig : Configuration space for a rotating part

6.4 Gross motion planning

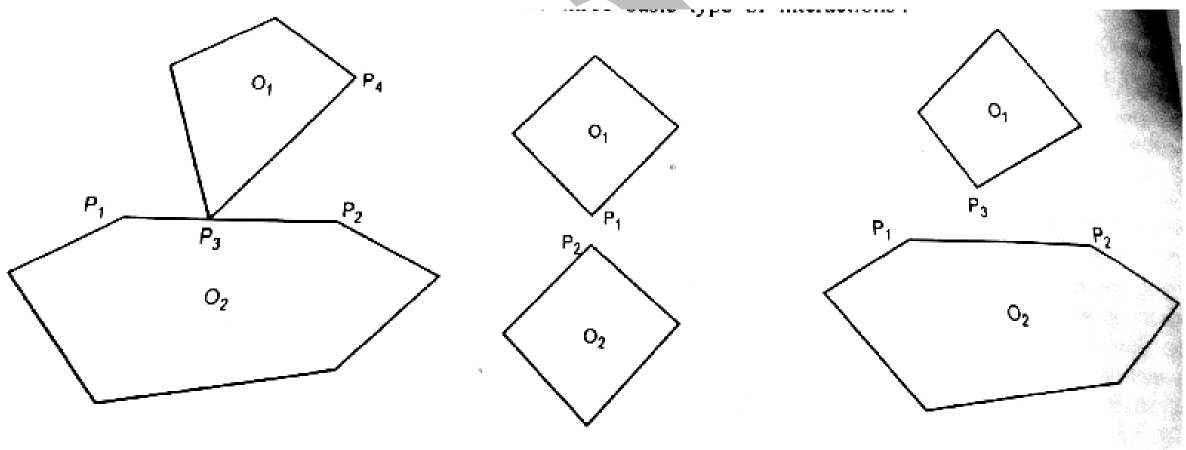
An alternative to the configuration space approach to solve gross-motion path planning problem is to search the free space directly. An explicit representation was proposed by Brooks(1983) for free space based on overlapping generalized cones having straight spines and non-increasing radii. They are referred as freeways. It typically generates path for the mobile part that are away from the obstacles. When the workspace is very sparsely populated with obstacles, this method is fast and effective. The principal drawback is when the workspace is cluttered with closely spaced obstacles.

The freeway method of designing the gross motion path is known as the Generalized Voronoi Diagram. It is used for obtaining a obstacle collision free path in the work space of the robot from source to the goal. A GVD in free space is defined as the locus of all points which are equidistant from two or more than two obstacle boundaries.

Types of GVD

The robot workspace consists of a number of obstacles. The parameters of any obstacles are edges and the vertices. So, while constructing the GVD from S to the G , four basic types of interactions occur. The interactions are

1. Interaction between a pair of edges
2. Interaction between a vertex and an edge
3. Interaction between a pair of vertices
4. GVD induced by a skew edge



a) Interaction between pair of edges b) a pair of vertices c) a vertex and an edge

6.5 Grasp Planning

The grasp-planning problem is determining the configuration for the robotic tool that produces a safe, reachable and secure grasp of the part at both ends of the path along which it is to be transported.

Safe grasp: The robot must be safe at the initial grasp and final grasp configuration like a pick and place operation. The part to be manipulated should not be in contact with any obstacles. The robot or the object will be safe if it is moving along the GVD path.

Reachable grasp: The robot should be within the workspace and the pick point should be reachable by the robot. It is defined as the one which is within the work envelope of the manipulator and one for which a collision free path to the grasp configuration is available. Example, a gravity part feeder is a typical example for reachable grasp.

Secured/Stable grasp: It is defined as the one for which the object or the part will not move or slip during the part transfer. The grasp should be stable during the transportation of the object from the pick point to the place point

6.6 Fine motion planning

It is the type of motion planning technique when a mobile part or an object comes into physical contact with the surrounding environment of the robot. Moving at slow speed is also called as fine motion planning.

It can be achieved by using impedance control technique, guarded motion or compliant motion.

Guarded motion: It is defined as moving an object or a part in a specified direction towards the goal surface until an event occurs. The event is generation of a reaction force by the surface.

Compliant motion: The motion required after guarded motion in order to place the object in desired position and orientation is complaint motion. There should be continuous physical contact with the environment. The coefficient friction should be low.

7 Moment of Inertia

7.1 **Introduction**

The concepts on which a body or any matter depends upon are the relative distribution of area, mass and the weight .Quantitative estimates of the relative distribution of the area and the mass over a region of interest is provided by Newton's Law of motion.

Moment of Inertia plays a very important role in the dynamics and kinematics of robot mechanism.

Properties of any physical body or rigid body are factors like Area, Mass , Weight , Moment of Inertia, Boundary , Corner points , Vertices, Volume , Centroid , Orientation , Edges, Position , Shape, Physical dimensions and Inertia

This chapter focuses on moment of inertia of three dimensional objects which are based on the assumption that the objects are homogeneous with a density of ρ.

6.2 Definition:- Inertia of a body

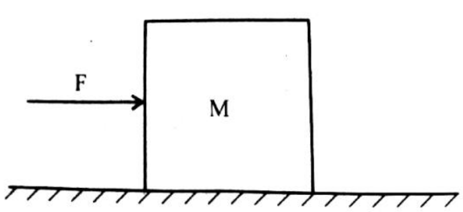
The property by virtue of which it resists any change in its state of rest or of uniform motion is called as inertia either by translation or rotation of a body due to application of a force.

Moment of Inertia is a mathematical expression which is used in engineering mechanics. When a force is applied to a body, which is fixed at one end, the body rotates about the axis with an angular velocity of ω radians/ sec and with an angular acceleration of α. The body rotates because of the production of torque, which is the moment of force about an axis.

Therefore, the torque is the force multiplied by the perpendicular distance from the axis given by:

T = F x perpendicular distance of force to the axis of rotation.

$T = I_m \frac{d^2\theta}{dt^2}$; where I_m is the moment of inertia of all the rotating masses and that of the load.



A body subject to a force

6.3 Types of Moments of Inertia(MI)

7.3.1 Moment of Inertia of Area

It originates whenever one computes the moment of a distributed load that varies linearly with the moment axis

It can be calculated about x and y –axis given by I. It is also calculated about z-axis denoted by J , known as polar Moment of Inertia.

The mathematical definition of MI, I = $\int \rho^2 \, dA$ which indicates that an area is divided into small parts dA and each area is multiplied by the square of its moment arm about the reference axis.

Since the objects are homogeneous with a density ρ with volume V, the moment of Inertia about the x, y, and z axes of an orthonormal coordinate frame are computed as follows:

$I_{xx} = \iiint (y^2 + z^2)\, \rho \, dx\, dy\, dz$

$I_{yy} = \iiint (x^2 + z^2)\, \rho \, dx\, dy\, dz$

$I_{zz} = \iiint (x^2 + y^2)\, \rho \, dx\, dy\, dz$

7.3.2 Moment of Inertia of mass I_m

The dynamics of rigid bodies involving the mathematical expression $\int \rho^2 \, dm$ is known as moment of inertia of mass m. The expression is for rotating bodies and is therefore regarded as a measure of mass distribution or of the resistance of the body to angular acceleration.

Mass moment of inertia for a differential element dm of body about any one of the three axes is defined as the product of mass of element and the square of the shortest distance from the axis to the element.

Homogeneous objects with regular shapes have moments of inertia that can be expressed in terms of the total mass m = ρV. It is given below:

Object	I_{xx}	I_{yy}	I_{zz}
Cylinder	$\frac{m(3r^2 + 4h^2)}{12}$	$\frac{m(3r^2 + 4h^2)}{12}$	$\frac{mr^2}{2}$
Thin cylinder	$\frac{mh^2}{3}$	$\frac{mh^2}{3}$	0
Cylindrical shell	$\frac{m(3r^2 + 2h^2)}{6}$	$\frac{m(3r^2 + 2h^2)}{6}$	mr^2
Prism	$\frac{m(b^2 + 4c^2)}{12}$	$\frac{m(a^2 + 4c^2)}{12}$	$\frac{m(a^2 + b^2)}{12}$
Cone	$\frac{m(3r^2 + 2h^2)}{20}$	$\frac{m(3r^2 + 2h^2)}{20}$	$\frac{3mr^2}{10}$
Pyramid	$\frac{m(b^2 + 2h^2)}{20}$	$\frac{m(a^2 + 2h^2)}{20}$	$\frac{m(a^2 + b^2)}{20}$

7.4 Dynamic Performance of the Robot Arm

The dynamics of any robotic system depends on its mass and moment of inertia. It is directly proportional. The increased MI increases the torque required to move the links and makes the robot heavy and slow.

The following factor has to be considered while modeling and controlling the robot:

- MI depends on the weight of the links.
- Mass of the robot links has to be less, so that MI is reduced and less torque is required to drive the joint motors.
- Joint –links to be light in weight
- Motors and gear mechanism to be mounted at the base which will increase the positional accuracy and reduce the overall MI of the robot.
- The length of the arm should reduce. MI depends on the axis.
- Damping should lie between 0 and 1

  Therefore, robots are designed in such a way that the overall weight of the arm reduces, MI reduces, torque required to drive the joint reduces.