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

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

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

* variable

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

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

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

LECTURE 11. **INVERSE KINEMATICS**

1. INVERSE KINEMATICS (GEOMETRIC APPROACH)

1.1. Two-Link Planar Robot

In order to command the robot to move to location A we need the joint variables θ_1, θ_2 in terms of the x and y coordinates of A. This is the problem of inverse kinematics. Since the forward kinematic equations are nonlinear, a solution may not be easy to find, nor is there a unique solution in general. In the case of a two-link planar mechanism that there may be no solution, for example if the given (x, y) coordinates are out of reach of the manipulator. If the given (x, y) coordinates are within the manipulator's reach there may be two solutions (elbow up and elbow down configurations) as shown in Fig. (11.1), or there may be exactly one solution if the manipulator must be fully extended to reach the point.

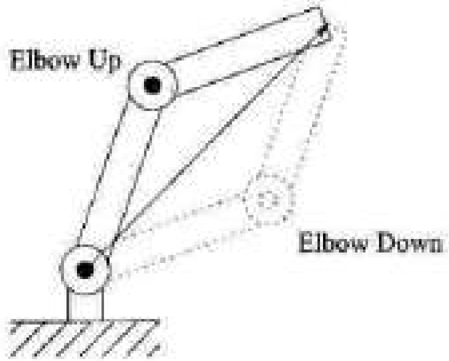


Figure (11.1). **The two-link elbow robot has two solutions to the inverse kinematics**

Consider the diagram of Fig. (11.2). Using the law of cosines that the angle θ_2 is given by:

$$\cos\theta_2 = \frac{x^2 + y^2 - a_1^2 - a_2^2}{2a_1a_2} := D$$

Mathematically we could now determine θ_2 as $\theta_2 = \cos^{-1}(D)$, but a better way to find θ_2 by:

$$\sin\theta_2 = \pm\sqrt{1 - D^2}$$

and, hence, θ_2 can be found by:

$$\theta_2 = \tan^{-1}\frac{\pm\sqrt{1-D^2}}{D}.$$

The advantage of this latter approach is that both the elbow-up and elbow-down solutions are recovered by choosing the negative and positive signs in the Equation respectively.

Thus, θ_1 can be found by:

$$\theta_1 = \tan^{-1}(y/x) - \tan^{-1}\frac{a_2\sin\theta_2}{a_1 + a_2\cos\theta_2}.$$

Notice that the angle θ_1 depends on which solution is chosen for θ_2.

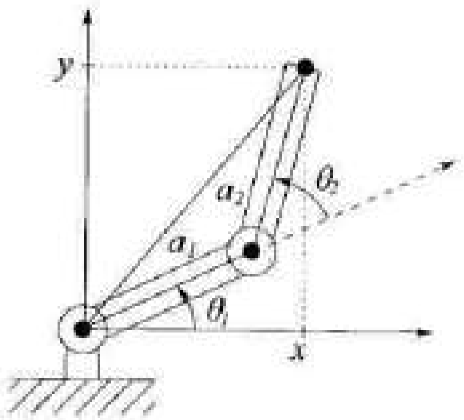


Figure (11.2). **Solving for the joint angles of a two-link planar arm**

1.2. Articulated Configuration

For the common kinematic arrangements we can use a geometric approach to find the position variables θ_1, θ_2 and θ_3 because of the most manipulator designs are kinematically simple (usually consisting of one of the five basic configurations in Lecture 9). The general idea of the geometric approach is to solve for joint variable θ_i by projecting the manipulator onto the X_{i-1}, Y_{i-1} plane and solving a simple trigonometry problem.

The projection of the wrist center of the **Elbow manipulator** at the $x_o - y_o$ plane (x_c, y_c, z_c) (Fig. (11.3) and (11.4)) gives:

$$\theta_1 = \text{Atan2}(x_c, y_c).$$

The function $Atan2$ denotes the two argument arctangent function, and thus the second solution for θ_1 will be:

$$\theta_1 = \pi + \text{Atan2}(x_c, y_c).$$

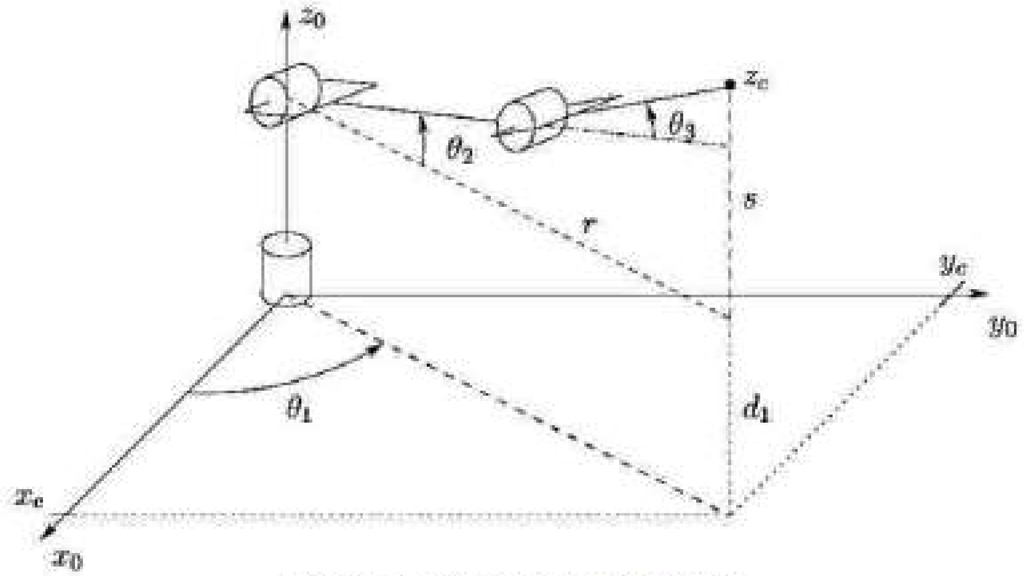


Figure (11.3). **Elbow manipulator**

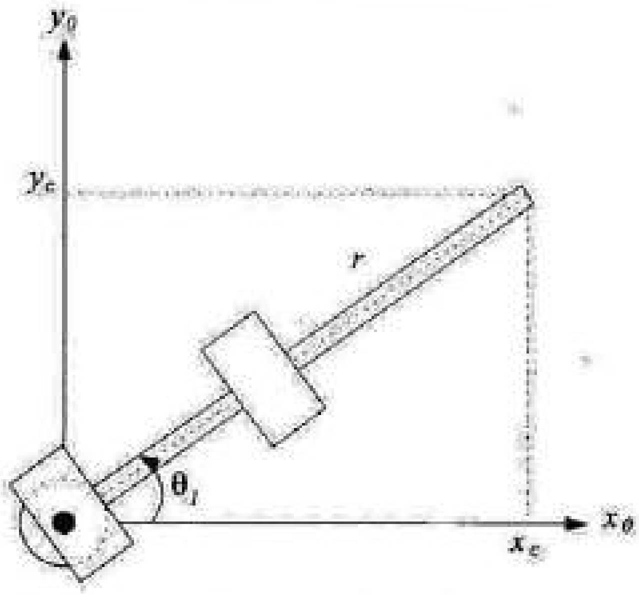


Figure (11.4). **Projection of the wrist center onto x_0-y_0 plane**

These two solutions for $\theta_1$are valid unless $x_c = y_c = 0$. In this case the equation of the first solution is undefined and the manipulator is in a singular configuration, shown in Fig. (11.5). In this position the wrist center O_c intersects $Z_o$hence any value of $\theta_1$leaves O_c fixed. There are thus infinitely many solutions for θ_1 when O_c intersects Z_o.

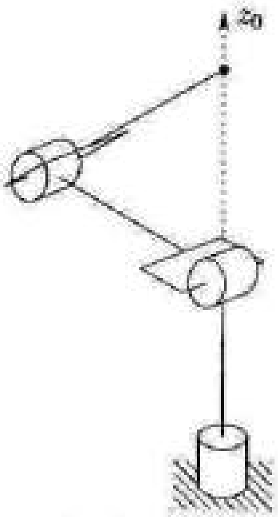


Figure (11.5).**Singular configuration in which the wrist center lies on the $Z_o$axis**

If there is an offset as shown in Fig. (11.6) $(d \neq 0)$ then the wrist center cannot intersect Z_o.In this case, the *D-H* parameters have been assigned, we will have $d_2 = d$, and $d_3 = d$there will be, in general, only two solutions for θ_1. These correspond to the **left arm** and **right arm** configurations as shown in Fig. (11.7) and (11.8).

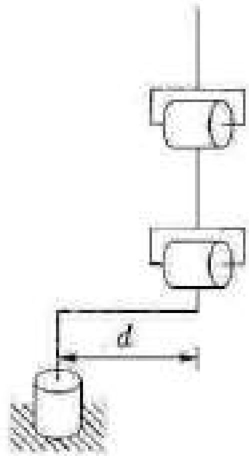


Figure (11.6).**Elbow manipulator with shoulder offset**

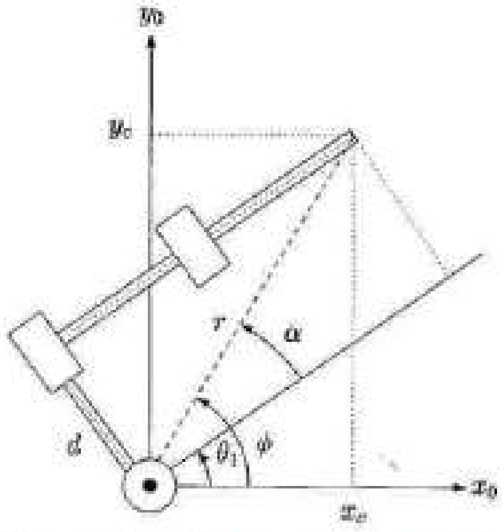


Figure (11.7).**Left arm configuration**

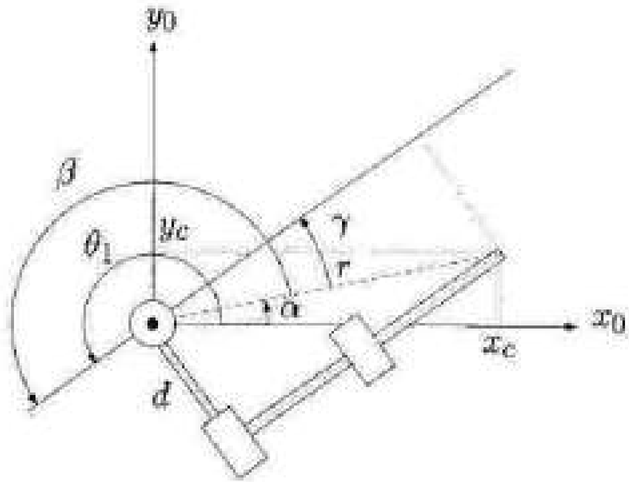


Figure (11.8). **Right arm configuration**

Fig. (11.7) shows the left arm configuration. From this figure, we see geometrically that:

$$\theta_1 = \phi - \alpha,$$

in which

$$\phi = \text{Atan2}(x_c, y_c);$$

$$\alpha = \text{Atan2}\left(\sqrt{r^2 - d^2}, d\right) = \text{Atan2}(\sqrt{x_c^2 + y_c^2 - d^2}, d).$$

The second solution, given by the right arm configuration shown in Fig. (11.8) is given by:

$$\theta_1 = \text{Atan2}(x_c, y_c) - \text{Atan2}\left(-\sqrt{r^2 - d^2}, -d\right).$$

To see this, note that

$$\theta_1 = \alpha + \beta;$$
$$\alpha = \text{Atan2}(x_c, y_c);$$
$$\beta = \gamma + \pi;$$
$$\gamma = \text{Atan2}\left(\sqrt{r^2 - d^2}, d\right).$$

which together imply that

$$\beta = \text{Atan2}\left(-\sqrt{r^2 - d^2}, -d\right)$$

since $\cos(\theta + \pi) = -\cos(\theta)$ and $\sin(\theta + \pi) = -\sin(\theta)$.

To find the angles θ_2, θ_3 for the elbow manipulator, we consider the plane formed by the second and third links as shown in Fig. (11.9). Since the motion of second and third links is planar, the solution is analogous to that of the two-link manipulator:

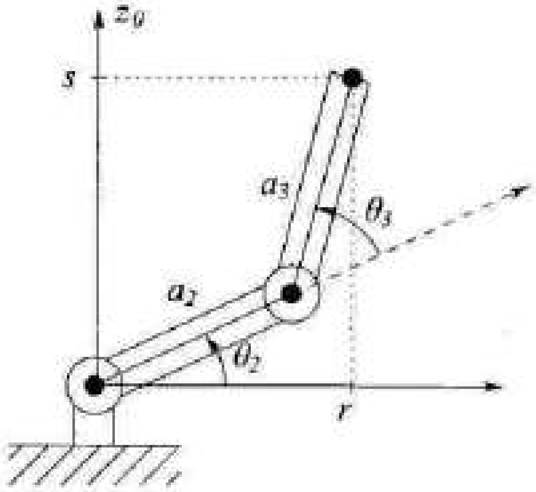


Figure (11.9). **Projecting onto the plane formed by links 2 and 3**

$$\cos\theta_3 = \frac{r^2 + s^2 - a_2^2 - a_3^2}{2a_2a_3} = \frac{x_c^2 + y_c^2 - d^2 + (z_c - d_1)^2 - a_2^2 - a_3^2}{2a_2a_3} := D$$

since $r^2 = x_c^2 + y_c^2 - d^2$ and $s = z_c - d_1$. Hence, θ_3 is given by

$$\theta_3 = \text{Atan2}(D, \pm\sqrt{1 - D^2}).$$

The two solutions for θ_3 correspond to the elbow-down position and elbow-up position, respectively. Similarly θ_2 is given as:

$$\theta_2 = \text{Atan2}(r, s) - \text{Atan2}(a_2 + a_3c_3, a_3s_3)$$

$$= \text{Atan2}\left(\sqrt{x_c^2 + y_c^2 - d^2}, z_c - d_1\right) - \text{Atan2}(a_2 + a_3c_3, a_3s_3).$$

An example of an elbow manipulator with offsets is the PUMA shown in Fig. (11.10). There are four solutions to the inverse position kinematics as shown. These correspond to the situations

- left arm-elbow up;
- left arm-elbow down;
- right arm-elbow up;
- right arm-elbow down.

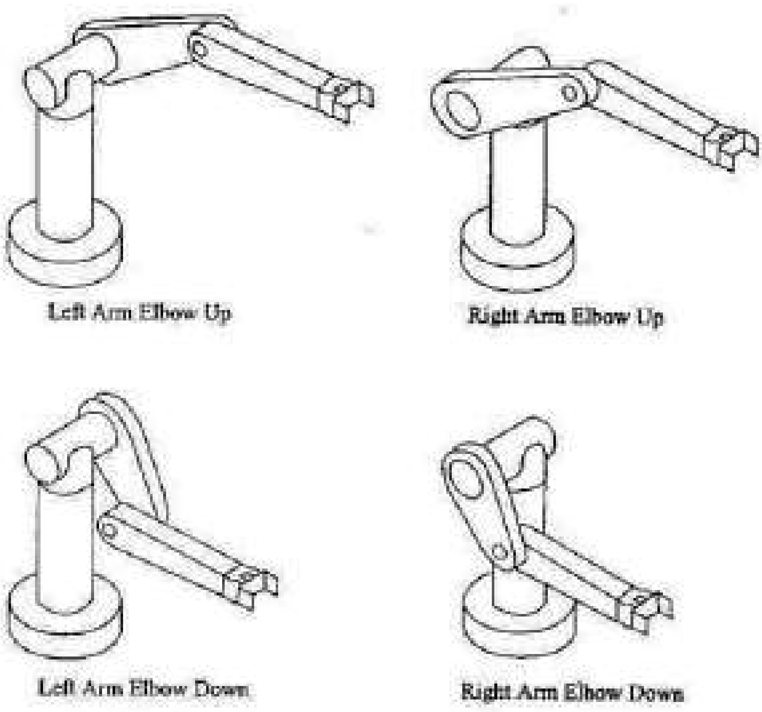


Figure (11.10). **Four solutions of the inverse position kinematics for the PUMA manipulator**

Atan2 COMPUTATION

In terms of the standard ($\tan^{-1}$) function, that is with range of ($-\pi/2$, $\pi/2$), Atan2 can be expressed as follows:

$$Atan2\,(M,N) = \begin{cases} tan^{-1}\left(\frac{N}{M}\right) & if\ M > 0 \\ \pi + tan^{-1}\left(\frac{N}{M}\right) & if\ N \geq 0\,, M < 0 \\ -\pi + tan^{-1}\left(\frac{N}{M}\right) & if\ N < 0\,, M < 0 \\ \frac{\pi}{2} & if\ N > 0\,, M = 0 \\ -\frac{\pi}{2} & if\ N < 0\,, M = 0 \\ undefined & if\ N = 0\,, M = 0 \end{cases}$$

- Where M represent the X-axis and N represent the Y-axis.
- This produces results in the range [$-\pi$, π], which can be mapped to [0, 2π] by adding 2π to negative values.

USEFUL TRIGONOMETRIC FORMULAS

- **Reduction Formulas**

$$\sin(-\theta) = -\sin\theta \qquad \sin\left(\frac{\pi}{2} + \theta\right) = cos\theta$$

$$\cos(-\theta) = \cos\theta \qquad \tan\left(\frac{\pi}{2} + \theta\right) = -\cot\theta$$

$$\tan(-\theta) = -\tan\theta \qquad \tan(\theta - \pi) = tan\theta$$

$$\sin(x \pm y) = \sin x \cos y \pm \cos x \sin y$$

$$\cos(x \pm y) = \cos x \cos y \mp \sin x \sin y$$

$$\tan(x \pm y) = \frac{\tan(x) \pm \tan(y)}{1 \mp \tan x \tan y}$$

Low of Cosines

If a triangle has sides of length a, b, and c, and θ is angle opposite the side of length c, then

$$c^2 = a^2 + b^2 - 2ab \cdot \cos\theta$$

1. PATH AND TRAJECTORY PLANNING

A path is defined as a sequence of robot configurations in a particular order without regard to the timing of these configurations. So, if a robot goes from point (and thus, configuration) ***A*** to point ***B*** to point ***C*** in Fig. (5.1), the sequence of the configurations between ***A*** and ***B*** and ***C*** constitutes a path. However, a trajectory is concerned about when each part of the path must be attained, thus specifying timing. Also planning the trajectory depends on the velocities and accelerations. Thus points *B* and *C* may be reached at different times, creating different trajectories.

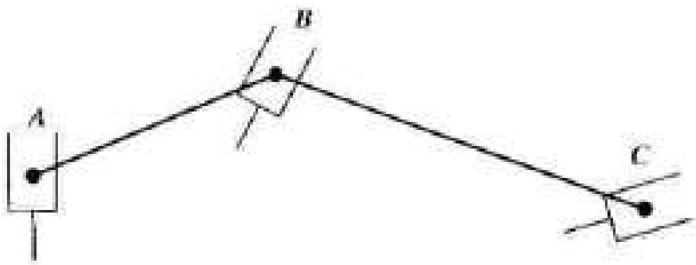


Figure (5.1). **Sequential robot movements in a path**

2. JOINT-SPACE VS. CARTESIAN-SPACE DESCRIPTIONS

Consider a 6 DOF robot at a point ***A*** in space, which is directed to move to another point ***B***. Using the inverse kinematic equations of the robot, we can calculate the total joint displacements that the robot needs to get to the new location. The joint values thus calculated can be used by the controller to drive the robot joints to their new values and, consequently, move the robot arm to its new position. The description of the motion to be made by the robot by its joint values is called ***joint-space description***. In this case, although the robot will eventually move to the desired position, the motion between the two points is **unpredictable**, as will be seen later.

Now assume that a straight line is drawn between points ***A*** and ***B*** and that it is desirable to have the robot move from point ***A*** to point ***B***, but it is also desirable that it follow the straight line between the two points. To do this, it will be necessary to divide the straight line into small portions, as shown in Fig. (5.2), and to move the robot through all intermediate points. To accomplish this task, at each intermediate location, the robot's inverse kinematic equations are solved, a set of joint variables is calculated, and the controller is directed to drive the robot to the next segment.

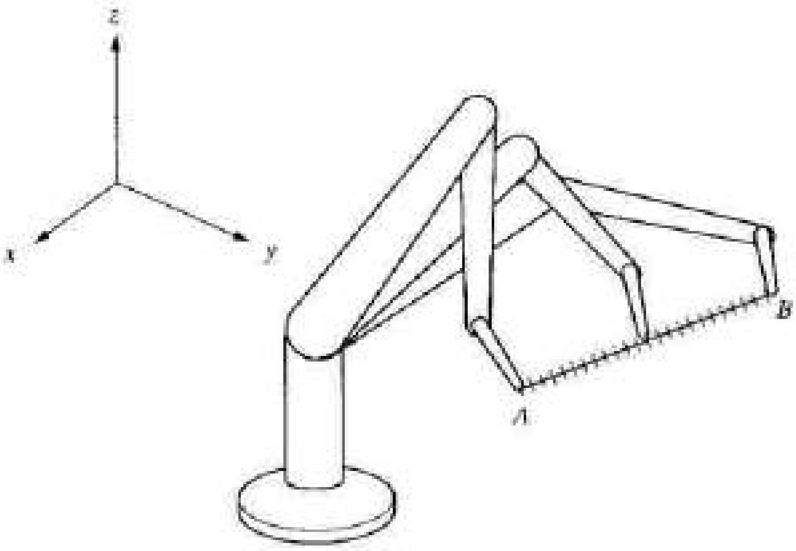


Figure (5.2). **Sequential motions of a robot to follow a straight line**

When all segments are completed, the robot will be at point ***B***, as desired. However, in this case, unlike the joint-space case, the motion is known at all times. The sequence of movements that the robot makes is **described in *Cartesian space*** and is converted to *joint-space* at each segment. As is clear from this simple example, *the **Cartesian-space description*** is much more computationally intensive than the joint-space description, but yields a **controlled and known** path. Both joint-space and Cartesian-space descriptions are very useful and are used in industry. However, each one has its own advantages and disadvantages.

Cartesian-space trajectories are very **easy to visualize**. Since the trajectories are in the common Cartesian space in which we all operate, it is easy to visualize what the end-effector's trajectory must be. However, Cartesian space trajectories are **computationally extensive**, and **require a faster processing time** for similar resolution as joint-space trajectories. Additionally, although it is easy to visualize the trajectory, it is **difficult to visually ensure that singularities will not occur**. For example, consider the situation in Fig. (5.3(a)). If not careful, one may specify a trajectory that requires the robot to run into itself or to reach a point outside of the work envelope, which, of course, is impossible and yields an unsatisfactory solution. This is true because it may be impossible to know whether the robot can actually make a particular location and orientation before the motion is made. Also, as shown in Fig. (5.3(b)), the motion between two points may require an instantaneous change in the joint values, which is impossible to make. Some of these problems may be solved by a joint space description with specifying via points through which the robot must pass in order to avoid obstacles and other similar singularities.

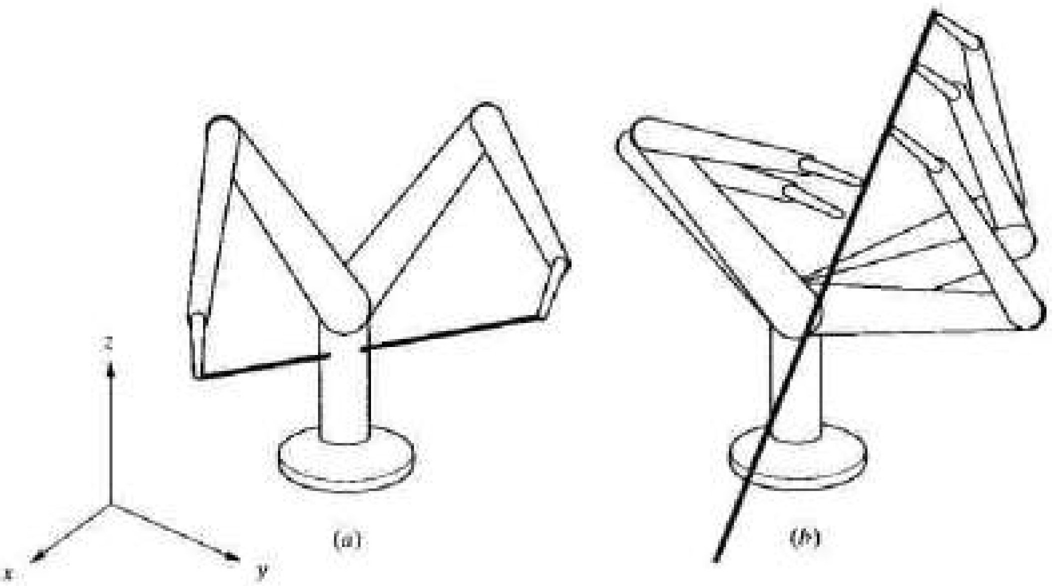
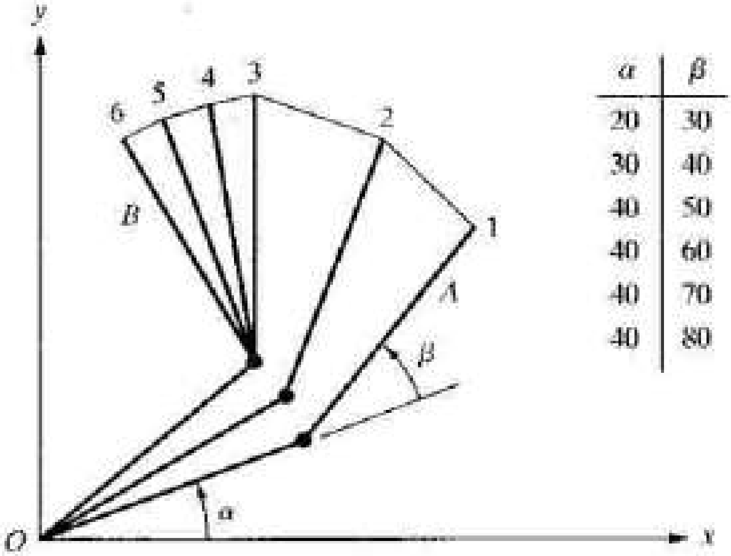


Figure (5.3). **Cartesian-space trajectory problems: (a) The trajectory specified in Cartesian coordinates may force the robot to run into itself, and (b) the trajectory may require a sudden change in the joint angles**

3. BASICS OF TRAJECTORY PLANNING

To understand the basics of planning a trajectory in joint-space and Cartesian-space, let's consider a simple two-degree of freedom robot. In this case, as shown in Fig. (5.4), we desire to move the robot from point $\boldsymbol{A}$ to point $\boldsymbol{B}$. The configuration of the robot at point $\boldsymbol{A}$ is shown, with $\boldsymbol{\alpha} = 20°$ and $\boldsymbol{\beta} = 30°$. Suppose that it has been calculated that in order for the robot to be at point $\boldsymbol{B}$, it must be at $\boldsymbol{\alpha} = 40°$ and $\boldsymbol{\beta} = 80°$. Also suppose that **both joints of the robot can move at the maximum rate of**

10 degrees/sec. One way to move the robot from point ***A*** to point ***B*** is to run both joints at their maximum angular velocities. **This means that after two seconds the lower link of the robot will have finished its motion, while the upper link continues for another three seconds**, as shown in the figure. As indicated, the path is irregular, and the distances traveled by the robot's end are not uniform.

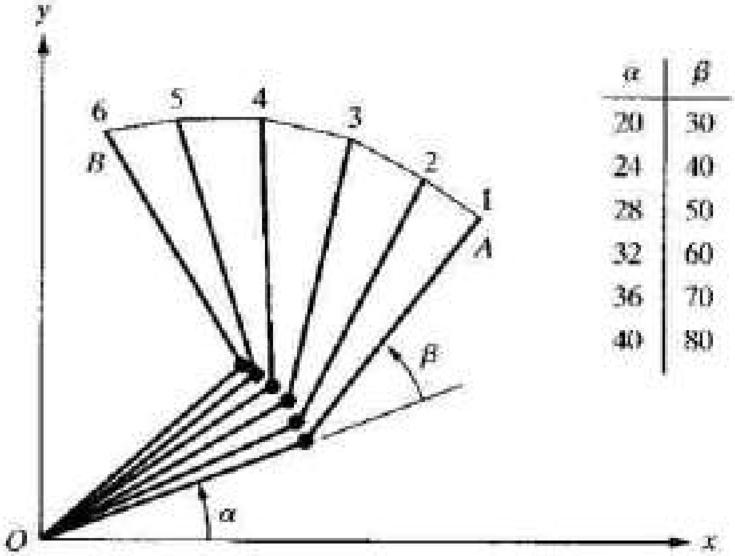


α	β
20	30
30	40
40	50
40	60
40	70
40	80

Figure (5.4). **Joint-space non-normalized movements of a robot with 2 DOF**

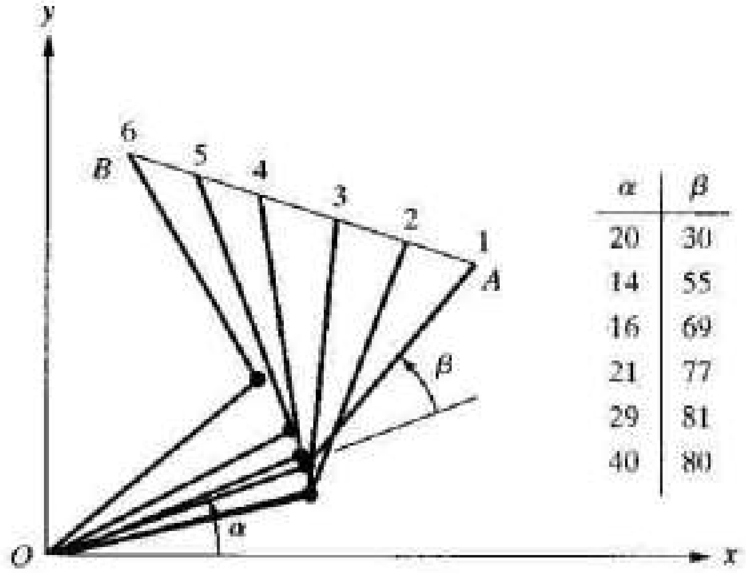
Now suppose that the motions of both joints of the robot are normalized by a common factor such that the joint with smaller motion will move proportionally slower and that both joints will start and stop their motion simultaneously. In this case, both joints move at different speeds, but move continuously together, ***α*** **will change 4 degrees per second** while ***β*** **changes 10 degrees per second**. The resulting trajectory will be different, as shown in Fig. (5.5). You notice that the segments of the movement are much more similar to each other than before, but that the path is still irregular (and different from the previous case). Both of these cases were planned in joint-space. The only calculation needed was the joint values for the destination and, in the second case, normalization of the joint velocities.

Now suppose we desire that the robot's hand follows a known path between points ***A*** and ***B***, say, **in a straight line**. The simplest solution would be to draw a line between points ***A*** and ***B***, divide the line into, say, five segments, and solve for necessary angles ***α*** and ***β*** at each point, as shown in Fig. (5.6). This is called ***interpolation*** between points ***A*** and ***B***. In this case, the path is a straight line, but that the joint angles are not uniformly changing. Although the resulting motion is a straight (and thus, known) trajectory, it is necessary to solve for the joint values at each point. Obviously, many more points must be calculated for better accuracy, as with so few segments, the robot will not exactly follow the lines at each segment. This trajectory is in Cartesian-space, since all segments of the motion must be calculated based on the information expressed in a Cartesian frame. In this case, it is assumed that the robot's actuators are strong enough to provide large forces necessary to accelerate and decelerate the joints as needed.



α	β
20	30
24	40
28	50
32	60
36	70
40	80

Figure (5.5). **Joint-space normalized movements of a robot with 2 DOF**



α	β
20	30
14	55
16	69
21	77
29	81
40	80

Figure (5.6). **Cartesian-space movements of a robot with 2 DOF**

4. JOINT-SPACE TRAJECTORY PLANNING

The motions of a robot can be planned in joint-space with controlled characteristics. A number of different schemes, such as polynomials of different orders and linear functions with parabolic blends, can be used in joint-space trajectory planning. These schemes specify joint values and not Cartesian values.

4.1. Third-Order Polynomial Trajectory Planning

In this application, the initial location and orientation of the robot is known, and using the inverse kinematic equations, we find the final joint angles for the desired position and orientation. Now consider one of the joints, which at the beginning of the motion segment at time t_i is at θ_i. We desire to have the joint move to a new value of θ_f at time t_f. Using a polynomial to plan the trajectory, such that:

- the initial and final boundary conditions match the values of θ_i and θ_f which are known;
- the velocities at the beginning and the end of the motion segment are zero (or other known values).

These four pieces of information allow us to solve for four unknowns (or a third-order polynomial) in the form of:

$$\theta(t) = c_0 + c_1 t + c_2 t^2 + c_3 t^3,$$

where the initial and final conditions are:

$$\theta(t_i) = \theta_i;$$
$$\theta(t_f) = \theta_f;$$
$$\dot{\theta}(t_i) = 0;$$
$$\dot{\theta}(t_f) = 0.$$

Taking the first derivative of the polynomial Equation:

$$\theta(t_i) = c_0 = \theta_i;$$
$$\theta(t_f) = c_0 + c_1 t_f + c_2 t_f^2 + c_3 t_f^3;$$
$$\dot{\theta}(t_i) = c_1 = 0;$$

$$\dot{\theta}(t_f) = c_1 + 2c_2 t_f + 3c_3 t_f^2 = 0.$$

By solving these four equations simultaneously, we get the necessary values for the constants as follows:

$$\theta(t) = c_0 + c_1 t + c_2 t^2 + c_3 t^3;$$
$$\dot{\theta}(t) = c_1 + 2c_2 t + 3c_3 t^2.$$
At $t = 0$ ($t = t_i =$ Zero)
$$\theta(0) = c_0 = \theta_i;$$
$$\dot{\theta}(0) = c_1 = 0. \qquad \text{(initial velocity = Zero)}$$
At $t = t_f$:
$$\theta(t_f) = \theta_f = \theta_i + c_2 t_f^2 + c_3 t_f^3$$
$$\rightarrow c_2 t_f^2 + c_3 t_f^3 = \theta_f - \theta_i. \quad \ldots\ldots\ldots\ldots\ldots \quad (1)$$
$$\dot{\theta}(t_f) = 2c_2 t_f + 3c_3 t_f^2 = 0 \quad \text{(Zero velocity at the final time } t_f\text{)}$$
$$\rightarrow 2c_2 t_f + 3c_3 t_f^2 = 0. \quad \ldots\ldots\ldots\ldots\ldots \quad (2)$$
Solving Equation (1) and (2) gives:
$$c_3 = \frac{-2(\theta_f - \theta_i)}{t_f^3};$$
$$c_2 = \frac{3(\theta_f - \theta_i)}{t_f^2}.$$

- This allows us to calculate the joint position at any interval of time, which can be used by the controller to drive the joint to position.
- The same process must be used for each joint individually, but they are all driven together from start to finish.
- If the initial and final velocities are nonzero, the given values can be used in these equations.
- If more than two points are specified such that the robot will go through the points successively, the boundary velocities and positions at the conclusion of each segment can be used as the initial values for the next segments.

Positions and velocities are continuous; accelerations are not, which may cause problems.

Example 5.1.

It is desired to have the first joint of a 6 DOF robot go from initial angle of 30° to a final angle of 75° in 5 seconds. Using a third-order polynomial, calculate the joint angle at 1, 2, 3 and 4 seconds.

Solution

Substituting the boundary conditions into the Equations above:

$$\begin{cases} \theta(t_i) = c_0 = 30, \\ \theta(t_f) = c_0 + c_1(5) + c_2(5^2) + c_3(5^3) = 75, \\ \dot{\theta}(t_i) = c_1 = 0, \\ \dot{\theta}(t_f) = c_1 + 2c_2(5) + 3c_3(5^2) = 0, \end{cases} \rightarrow \begin{cases} c_0 = 30, \\ c_1 = 0, \\ c_2 = 5.4, \\ c_3 = -0.72. \end{cases}$$

This will result in the following cubic polynomial equation for position, as well as the velocity and acceleration equations:

$$\theta(t) = 30 + 5.4t^2 - 0.72t^3,$$

$$\dot{\theta}(t) = 10.8t - 2.16t^2,$$

$$\ddot{\theta}(t) = 10.8 - 4.32t.$$

Substituting the desired time intervals into the motion equation gives:

$$\theta(1) = 34.68°; \quad \theta(2) = 45.84°; \quad \theta(3) = 59.16°; \quad \theta(4) = 70.32°.$$

The joint angles, velocities, and accelerations are shown in Fig. (5.7). The acceleration needed at the beginning of the motion is 10.8 degrees/sec^2 and -10.8 degrees/sec^2 deceleration at the end of the motion.

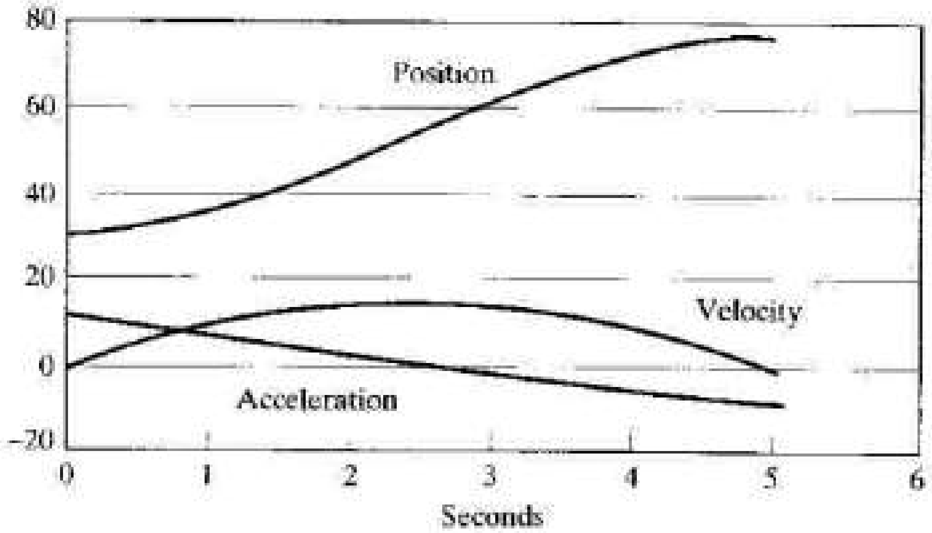


Figure (5.7). **Joint positions, velocities, and accelerations for Example 5.1**

Example 5.2

Suppose that the robot arm of Example 5.1 is to continue to the next point, where the joint is to reach 105° in another 3 seconds. Draw the position, velocity, and acceleration curves for the motion.

Solution

At the end of the first segment, the position and velocity of the joint are known. Using these values as initial conditions for the next segment:

$$\theta(t) = c_0 + c_1 t + c_2 t^2 + c_3 t^3,$$
$$\dot{\theta}(t) = c_1 + 2c_2 t + 3c_3 t^2,$$
$$\ddot{\theta}(t) = 2c_2 + 6c_3 t,$$

Where

$$t_i = 0, \quad \theta_i = 75, \quad \dot{\theta}_i = 0,$$
$$t_f = 3, \quad \theta_f = 105, \quad \dot{\theta}_f = 0,$$

which yields

$$c_0 = 75, \quad c_1 = 0, \quad c_2 = 10, \quad c_3 = -2.222,$$
$$\theta(t) = 75 + 10t^2 - 2.222t^3,$$
$$\dot{\theta}(t) = 20t - 6.666t^2,$$
$$\ddot{\theta}(t) = 20 - 13.332t.$$

Fig. (5.8) shows the position, velocity, and accelerations for the entire motion (Example 5.1 and 5.2). As can be seen, the boundary conditions are as expected.

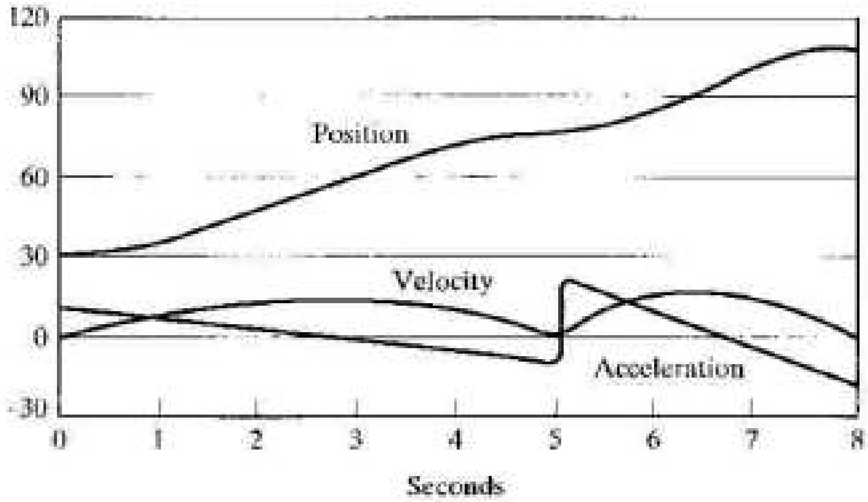


Figure (5.8). **Joint positions, velocities, and accelerations for Example 5.2**

In the figure above, the velocity curve is continuous, the slope of the velocity curve changes from negative to positive at the intermediate point, creating an instantaneous change in acceleration. Whether or not the robot is capable of creating such accelerations is a question that must be answered depending on the robot's capabilities. To ensure that the robot's accelerations will not exceed its capabilities, acceleration limits may be used to calculate the necessary time to reach the target. In that case, for $\dot{\theta}_i = 0$, and $\dot{\theta}_f = 0$, the maximum acceleration will be:

$$|\ddot{\theta}|_{max} = \left|\frac{6(\theta_f - \theta_i)}{(t_f - t_i)^2}\right|,$$

from which the time-to-target can be calculated. You should also notice that the velocity at the intermediate point does not have to be zero. In that case, the concluding velocity at the intermediate point will be the same as the initial velocity of the next segment. These values must be used in calculating the coefficients of the third-order polynomial.

4.2. Fifth-Order Polynomial Trajectory Planning

- ✓ In addition to specifying the initial and ending positions and velocities of a segment, it is also possible to specify initial and ending accelerations for a segment.
- ✓ In this case, the total number of boundary conditions is six, enabling us to use a fifth-order polynomial to plan a trajectory:

$$\theta(t) = c_0 + c_1 t + c_2 t^2 + c_3 t^3 + c_4 t^4 + c_5 t^5,$$
$$\dot{\theta}(t) = c_1 + 2c_2 t + 3c_3 t^2 + 4c_4 t^3 + 5c_5 t^4,$$
$$\ddot{\theta}(t) = 2c_2 + 6c_3 t + 12c_4 t^2 + 20c_5 t^3.$$

These equations allow us to calculate the coefficients of a fifth-order polynomial with position, velocity, and acceleration boundary conditions.

Example 5.3

Repeat Example 5.1, but this time assume that the initial acceleration and final deceleration will be 5 degrees/sec^2.

Solution

From Example 5.1 and the given accelerations, we have:

$$\theta_i = 30°, \quad \dot{\theta}_i = 0 \text{ degrees/sec}, \quad \ddot{\theta}_i = 5 \text{ degrees/sec}^2,$$
$$\theta_f = 75°, \quad \dot{\theta}_f = 0 \frac{\text{degrees}}{\text{sec}}, \quad \ddot{\theta}_f = -5 \text{ degrees/sec}^2.$$

Using Equations of fifth-order polynomial with the given initial and final boundary conditions, we get:

$$c_0 = 30, \quad c_1 = 0, \quad c_2 = 2.5,$$
$$c_3 = 1.6, \quad c_4 = -0.58, \quad c_5 = 0.0464.$$

This results in the following motion equations:

$$\theta(t) = 30 + 2.5t^2 + 1.6t^3 - 0.58t^4 + 0.0464t^5,$$
$$\dot{\theta}(t) = 5t + 4.8t^2 - 2.32t^3 + 0.232t^4,$$
$$\ddot{\theta}(t) = 5 + 9.6t - 6.96t^2 + 0.928t^3.$$

Fig. (5.9) shows the position, velocity, and acceleration graphs for the joint. The maximum acceleration is 8.7 degrees/sec^2.

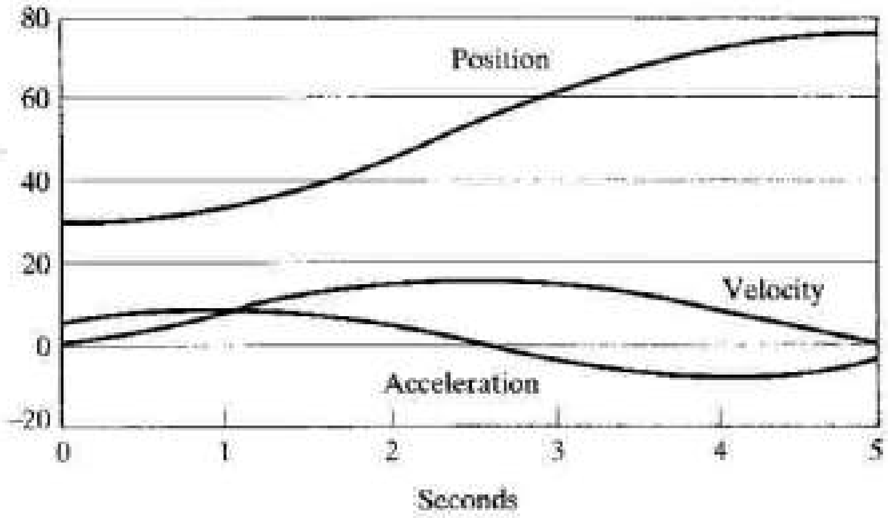


Figure 5.9. **Joint position, velocity, and acceleration for Example 5.3**

4.3. Linear Segments with Parabolic Blends

Another joint-space trajectory planning scheme to run the joints at constant speed between the initial and final locations. This is equivalent of a first order polynomial, where the velocity is constant and acceleration is zero. However, this also means that at the beginning and the end of the motion segment, accelerations must be infinite in order to create instantaneous velocities at the boundaries. To prevent this, the linear segment can be blended with parabolic sections at the beginning and the end of the motion segment, creating continuous position and velocity, as shown in Figure 5.10. Assuming that the initial and the final positions are θ_i and θ_f at time $t_i = 0$ and t_f, and that the parabolic segments are symmetrically blended with the linear section at blending times t_b and $t_f - t_b$ we can write:

$$\theta_i = \theta(t = 0) = \theta(0) = \text{initial position.}$$
$$\theta_f = \theta(t = t_f) = \theta(t_f) = \text{final position.}$$
$$t_i = 0(\text{starting position}).$$
$$t_f = \text{final time (ending time)}.$$

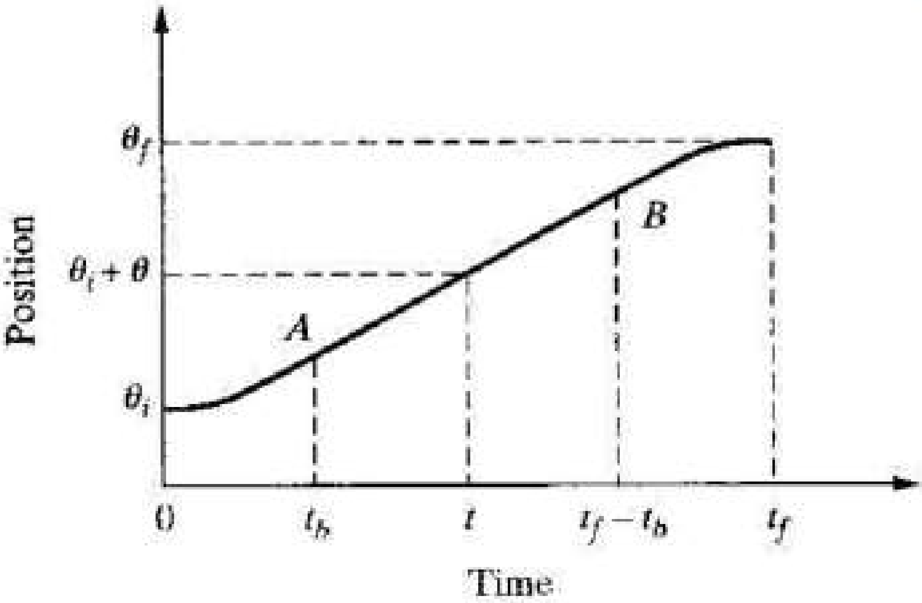


Figure (5.10). **Scheme for linear segments with parabolic blends**

For the first parabolic segment:

$$\theta(t) = c_0 + c_1 t + \frac{1}{2} c_2 t^2;$$
$$\dot{\theta}(t) = c_1 + c_2 t;$$
$$\ddot{\theta}(t) = c_2.$$

In this scenario, acceleration is constant for the parabolic sections, yielding a continuous velocity at the common points (called knot points) ***A*** and ***B***. Substituting the boundary conditions into the parabolic equation segment yields

At $t = 0 \rightarrow \quad \theta(0) = c_0 \rightarrow \quad c_0 = \theta_i;$

$$\dot{\theta}(0) = c_1 = 0 \text{ (starting velocity = 0)};$$
$$\ddot{\theta}(0) = c_2 \rightarrow \quad c_2 = \ddot{\theta} \; .$$

Substituting the initial conditions gives parabolic segments in the form:

$$\theta(t) = \theta_i + \frac{1}{2} c_2 t^2;$$
$$\dot{\theta}(t) = c_2 t;$$
$$\ddot{\theta}(t) = c_2.$$

For the linear segment:

Clearly, for the linear segment, the velocity will be constant and can be chosen based on the physical capabilities of the actuators. Substituting zero initial velocity, a constant known joint velocity ω in the linear portion, and zero final velocity, the joint positions and velocities for points ***A***, ***B*** and the final point as follows:

The general linear equation is,

$$\frac{y}{x}=\frac{y_2-y_1}{x_2-x_1} \quad \rightarrow \quad \frac{\theta}{t}=\frac{\theta_B-\theta_A}{t_f-t_b-t_b};$$
$$\frac{\theta}{t}=\omega \quad \rightarrow \quad \omega=\frac{\theta_B-\theta_A}{t_f-2t_b}$$
$$\rightarrow \theta_B=\theta_A+\omega(t_f-2t_b).$$

At $t=t_b$

Because of Point A = the end of the first parabolic segment = the start of the linear segment, then the Value of θ_A can be found from the end point of the first parabolic segement. So that:

$$\theta_A=\theta_i+\frac{1}{2}c_2t_b^2;$$
$$\dot{\theta}_A=c_2t_b=\omega \quad \text{(constant velocity at the linear segment).}$$

The necessary blending time t_b can be found as follows:

$$\theta_f=\theta_B+(\theta_A-\theta_i);$$
$$\theta_f=\theta_A+\omega(t_f-2t_b)+\theta_A-\theta_i; \quad \ldots\ldots\ldots \quad (3)$$
$$\theta_A=\theta_i+\frac{1}{2}c_2t_b^2; \quad \ldots\ldots\ldots \quad (4)$$
$$\theta_f=2\left(\theta_i+\frac{1}{2}c_2t_b^2\right)-\theta_i+\omega(t_f-2t_b);$$
$$\theta_f=\theta_i+c_2t_b^2+\omega(t_f-2t_b);$$
$$c_2=\frac{\omega}{t_b} \quad \ldots\ldots\ldots \quad (5)$$
$$\rightarrow \quad \theta_f=\theta_i+\left(\frac{\omega}{t_b}\right)t_b^2+\omega(t_f-2t_b);$$
$$\theta_f=\theta_i+\omega t_b+\omega t_f-2\omega t_b$$

Then calculating the blending time as:

$$t_b=\frac{\theta_i-\theta_f+\omega t_f}{\omega}.$$

- The time t_b cannot be bigger than half of the total time t_f which results in a parabolic speedup and a parabolic slowdown. With no linear segment, a corresponding maximum velocity:

$$\omega_{max}=2(\theta_f-\theta_i)/t_f.$$

- It should be mentioned here that if, for any segment, the initial time is not zero, but t_a, to simplify the mathematics, we can always shift the time axis by t_a to make the initial time zero.

For the final parabolic segment:

The final parabolic segment is symmetrical with the initial parabola, but with a negative acceleration, and thus can be expressed as follows:

$$\theta(t) = \theta_f - \frac{1}{2}c_2(t_f - t)^2, \quad \text{where} \quad c_2 = \frac{\omega}{t_b},$$

$$\rightarrow \begin{cases} \theta(t) = \theta_f - \dfrac{\omega}{2t_b}(t_f - t)^2, \\ \dot{\theta}(t) = \dfrac{\omega}{t_b}(t_f - t), \\ \ddot{\theta}(t) = -\dfrac{\omega}{t_b}. \end{cases}$$

Example 5.4

Joint 1 of the 6 DOF robot of Example 5.1 is to go from initial angle of $\theta_i = 30°$ to the final angle of $\theta_f = 70°$ in 5 seconds with a cruising velocity of $\omega_1 = 10$ degrees/second. Find the necessary time for blending, and plot the joint positions, velocities, and accelerations.

Solution:

Applying the equations of trajectory planning using Linear Segments with Parabolic Blends:

$$t_b = \frac{\theta_i - \theta_f + \omega_1 t_f}{\omega_1} = \frac{30 - 70 + 10(5)}{10} = 1 \text{ sec.}$$

For $\theta = \theta_i$ to θ_A

$$\begin{cases} \theta = 30 + 5t^2, \\ \dot{\theta} = 10t, \\ \ddot{\theta} = 10. \end{cases}$$

For $\theta = \theta_A$ to θ_B

$$\begin{cases} \theta = \theta_A + 10(t - t_b), \\ \dot{\theta} = 10, \\ \ddot{\theta} = 0. \end{cases}$$

For $\theta = \theta_B \ to \ \theta_f$

$$\begin{cases} \theta = 70 - 5(5 - t)^2, \\ \dot{\theta} = 10(5 - t), \\ \ddot{\theta} = -10. \end{cases}$$

Fig. (5.11) shows the position, velocity, and acceleration graphs for this joint.

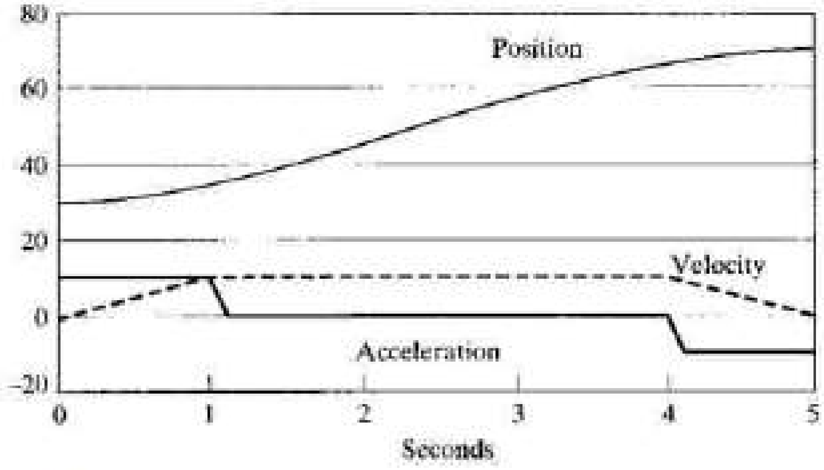


Figure (5.11). **Position, velocity, and acceleration graphs for Joint 1 of Example 5.4**

LECTURE 6. ROBOT FEATURES

1. REACH AND STROKE

The reach and the stroke are the measures of the dimensions of the work volume. They can be horizontal and vertical in the sense of movements. The respective reach and stroke are given in the Fig. (6.1) for a cartesian robot which has a cubical or parallelepiped work volume. A general relation between stroke and the reach is given by

$$\text{Reach} \geq \text{Stroke}$$

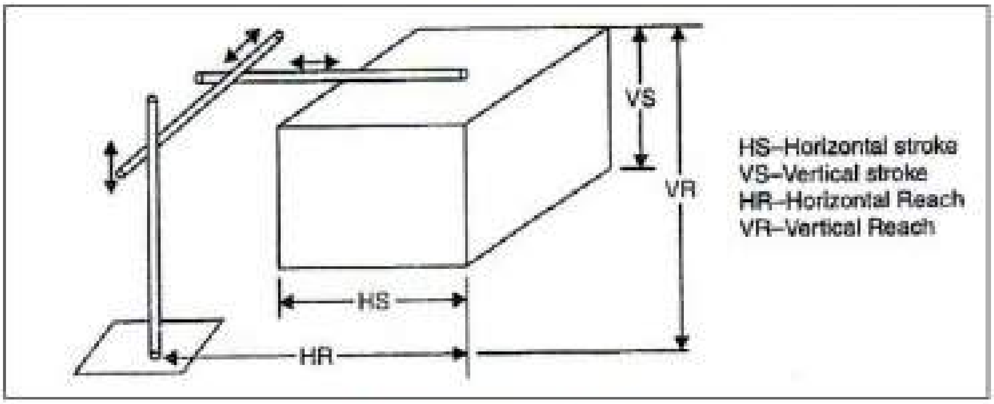


Figure (6.1). **Stroke and Reach**

2. OPERATING ENVIRONMENT

The nature of the work performing surrounding of a particular robot is specific to an application. The application of robot to a job can have following types of operating environments:

- ✓ dangerous to human beings;
- ✓ unhealthy in nature;
- ✓ harsh and difficult to access;
- ✓ complex and contaminated;
- ✓ extremely clean and dustless;
- ✓ ordinary and workable.

The examples of applications are movement of nuclear materials, spray coating or painting, welding (spot/continuous), loading and unloading, handling the electronic components and assembly of parts.

3. PERFORMANCE PARAMETERS

The manufacturing constraints and the design inevitability put some limitations on the performance of the robots. Such parameters are accounted by defining the following terminologies.

- Repeatability.
- Resolution or precision.
- Accurancy.

3.1. Repeatability

"**Repeatability** measures the ability of the robot to position the tool tip in the same place repeatedly". The position of the tool tip in the defined work space is programmed by understable commands. In the repeatitive work sequence the tip of the tool may or may not return back to the same point, leading to a repeatability error attributed to objective and subjective inaccuracies of the robot manipulator components.

The statistical distributions of the tip return points are conceptualised and illustrated in theFig. (6.2) The desired position of the point in space is denoted by 'D' and the achievable point through programming is 'A'. In the cycle of movement of the robot arm, the robot tries to return to a point nearest to 'A', denoted by R. The point 'R' can not be returned always to lead to the definition of the repeatability. This forms a cluster of repeated and non-repeated points distributed around the point A. *The difference in position of 'A' and the point 'R' is known as* **repeatability error**.

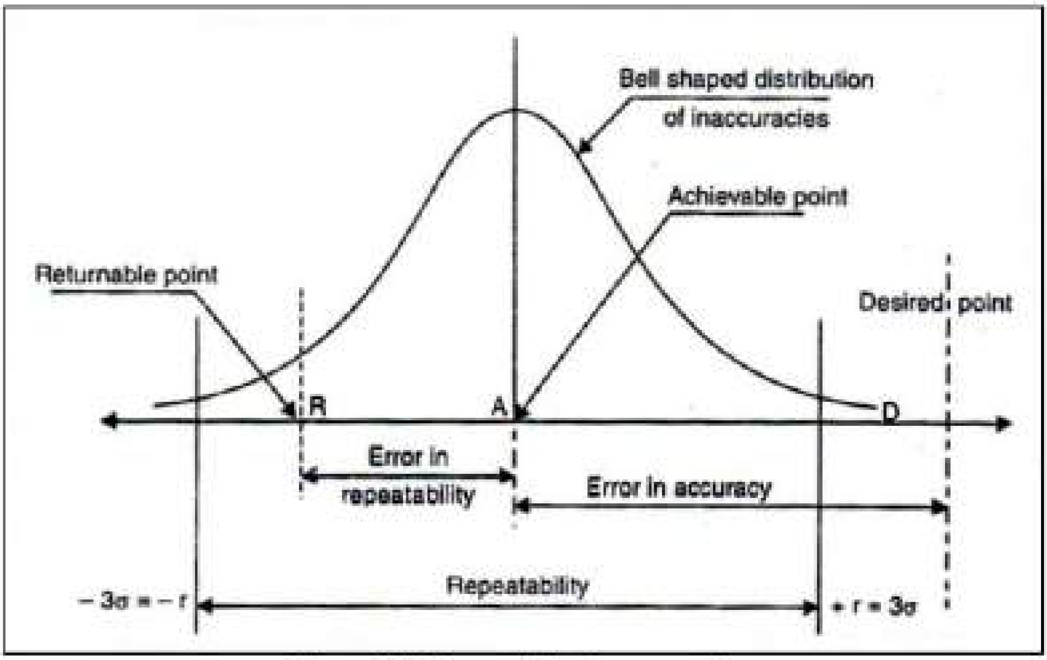


Figure (6.2). **Repeatability Representation**

The sphere of disrtibution in three dimensional space, of the point 'R' around the point 'A' gives the measure of repeatability. The manufacturer specifies the repeatability as the radius of the sphere on both sides of the centre of the sphere(point 'A'). If r_p is the radius of the sphere of distribution of point R from 'A' then,

$$\text{Repeatability} = \pm r_p.$$

3.2. Resolution

"Also known as spatial resolution is the least count of the movement into which the robot's work envelope can be divided to represent the incremental or decremental steps".

The spatial resolution can be contributed by two components:

- ✓ the control resolution;
- ✓ the mechanical resolution.

3.2.1. The Control Resolution

This component depends upon the type of position control system and its feedback control elements. The hardware capability of the controller determines the control resolution, the ability to divide the motion range into movement steps of the joints. The steps or the increments are also known as "Addressable points" depends on bit storage capacity of the control memory. The number of increments made equal by the designer, also called addressable point is,

$$i = 2^k$$

where k = number of bits in the control memory.

3.2.2. The Mechanical Resolution

The mechanical resolution: is the other component of resolution which puts limitation on the lower limit of the spatial resolution. The following are some of the factors that contribute to the limitations.

1. Inaccuracies in the dimentions of the links and joint components.
2. Elastic deflection of the structural members.
3. Backlash in the meshing gears of power transmission.
4. Streching of the transmission components.
5. Leakage of the fluids of hydraulic/pneumatic actuators.
6. In accuracy magnification due to scaling.
7. Load handled and speed to be achieved.

When the above said mechanical factors become dominant the increase in bit capacities of memory cease to benefit much on the improvement in the spatial resolution. The maintenance condition and the age of the robot also play significant role in the process of usage of robot and determining the resolution of the robot depreciation subject.

If 'S' is the stroke of a linear, prismatic link and 'i' the number of increments or the addressable points, the control resolution in total is given by

$$R_t = \frac{S}{i} = \frac{S}{2^k}.$$

The angular control resolution of a revolute joint can be expressed in terms of the 'n' number of slots arround the circumference of the circular disk in a reductionless drive as,

$$A_c = \frac{2\pi}{n2^k} \text{ radians/count.}$$

If the speed reduction occurs between driver shaft to load shaft through a gear train, then

$$A_c = \frac{2\pi}{nz2^k}$$

where gear reduction ratio is **z : 1**

For a partial rotation of angle ϕ degree

$$A_c = \frac{\phi}{2^k} \text{ degrees.}$$

The spatial resolution represented with the mechanical components due to inaccuracies is illustrated by a statistical distribution is shown in Fig. (6.3).

If σ is the standard deviation about the mean point of the programmed position of tool tip, the spatial resolution takes the form

$$R_s = R_t + 6\sigma.$$

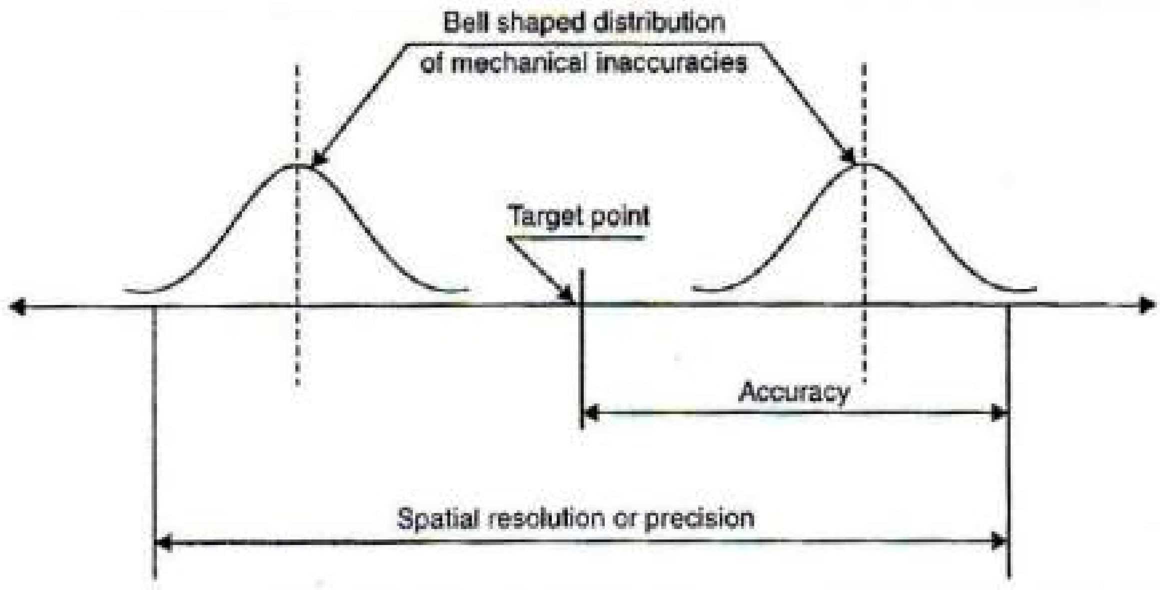


Figure (6.3). **Representation of Spatial Resolution and Accuracy**

The spatial resolution of a particular joint in a robot arm is due to the specifications of the drive system position sensors and power transmission mechanism like gears, spockets, chains, belts and cabeles. The programmed reference signal of the controlling computer generated inside is sent to the external analog feedback position control system through an 'k' bit digital to analog converter which decides the spatial resolution of the joint in focus.

Resolution for a Cylindrical Co-ordinated Robot

The work envelope of the cylindrical robot and elemental sweep is as shown in the Fig. (6.4a) The horizontal precision is lowest at the outer most radius or reach and highest at the inner most reach.

The grid element shown in Fig. (6.4b) is not a square but a sector. But for a very small division it almost looks like a square.

Hence,

The radial precision (resolution) = dr

Minimum angular resolution = $r d\phi$

Worst angular resolution = $R d\phi$.

Assuming grid element to be a square,

the horizontal resolution = $dh = \sqrt{(dr)^2 + (Rd\phi)^2}$

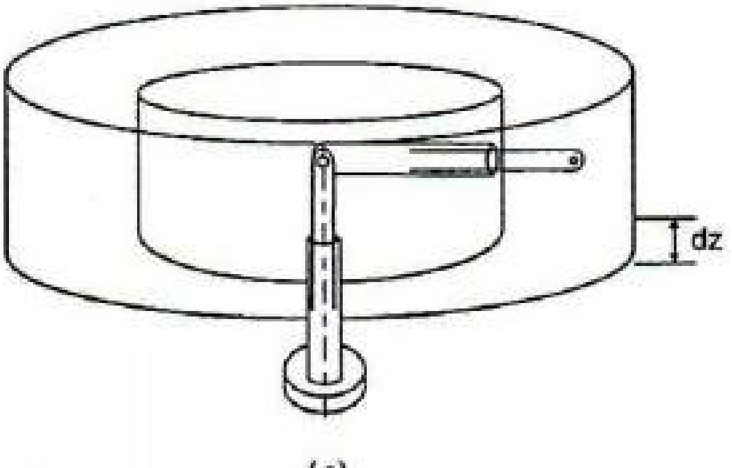
the vertical resolution = dz

Now,

the total control resolution is given by

$$dT = [(dr)^2 + (Rd\phi)^2 + (dz)^2]^{1/2}.$$

If a spherical robot is considered both the vertical and the horizontal precision are highest along the inside surface and decreases as the arm extends outward, and minimum at the outer most surface. If the robot with articulated joints is considered both the resolutions vary over the work space.



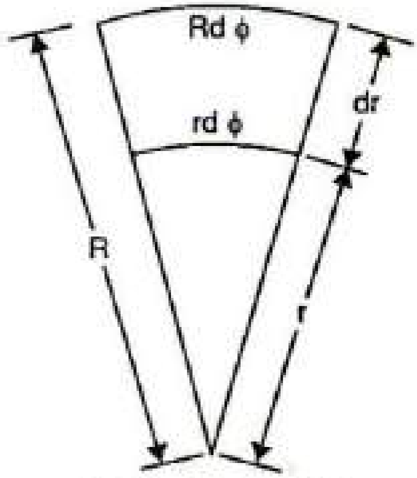


Figure (6.4). **work envelope of the cylindrical robot and elemental sweep**

3.3. Accuracy

"Accuracy is the measure of the robot's ability to orient and locate the tool tip at a desired target location in the prescribed work volume or envelop".

Accuracy is related to resolution because as the resolution value is less, the accuracy is more. So higher resolution gives better accuracy, the ability to achieve the prescribed target location. In a worst case the desired point may lie in between the two target points. The error in positioning is the other name to the inaccuracy given by the term,

$$\frac{\text{Control resolution}}{2} \leq \text{error},$$

where the machanical components of inaccuracies are neglected as they are more complicated to define and quantify. Hence, the precision related to the accuracy gives a picture of discrete grid nodes that can be visited by the wrist end or the tool tip within the work space. Hence, the best accuracy is half of the grid size as shown in the Fig. (6.5).

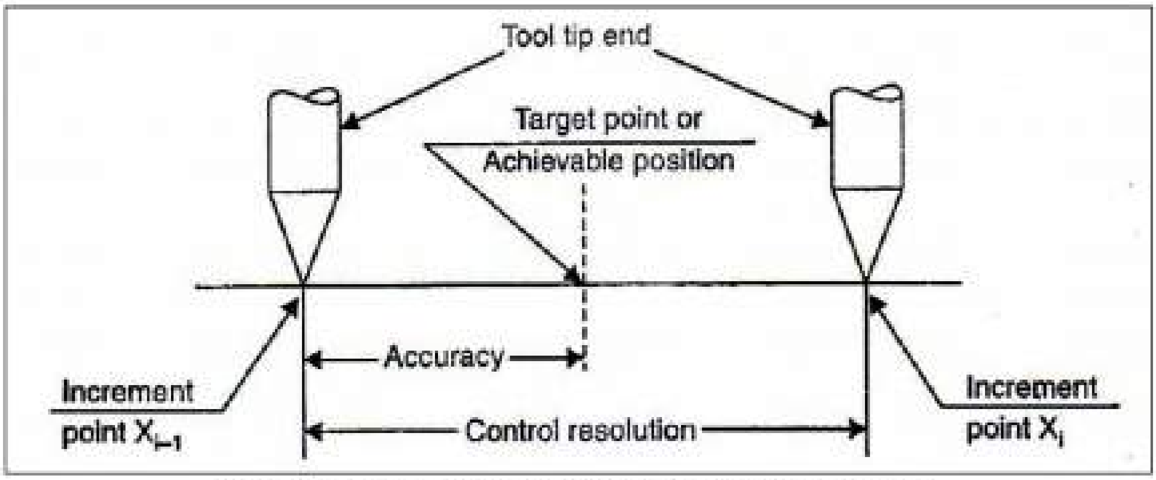


Figure (6.5). **Representation of Control Resolution and Accuracy**

After a periodic operation set the robot may to be calibrated to maintain the reasonable accuracy. The limit switch sensors of the robots are resert or zeroed during the periodic maintenance schedule. Futher intelligent algorithms with real time solutions are needed to define and re-define the control strategies to compensate for the uncertainity in environment and position.

LECTURE 7. **ROBOT PROGRAMMING**

For the robot user to get the task done from the robot manipulator there is the need for an effective and efficient programming method. Several subjective and objective programming techniques are developed to perform the task as determined to suit the application. The methods are: lead through teaching method, speech recognition and programming.

The lead through teaching which may be manual or powered is accomplished in the following steps.

- Leading the manipulator in slow motion controlled throughout the entire task operation.
- Storing the joint angles at needed path locations.
- Editing and play-back of the taught motion.
- Replay of the motion by the robot at a speed as specified by the user.

Speech recognition is voice technique where the user speaks out the discrete words of command. The stored commands are executed by the robot by word recognition process in real time leading to robot motion. The technique is primitive and requires large storage space of memory to record speech data. This also needs extensive training for the user in developing a speech template.

Programming is the most general and versatile subjective method of human-robot communication.

The considerations in robot programming are

- The three dimensional objects with different physical properties are to be manipulated.
- The environments of robot operations can be complex.
- Visualization of the object can be discrete.
- The processing and the analysis of the digital data from sensors and vision system has to be done in real time.

The high-level programming is oriented on tool and task basis, of operation sequence.

Robot Oriented Programming	Object Oriented Programming
• The sequence of robot motion is programmed and robot is made to follow. • The robot motions are explicitly defined. • The commands and actions are correspondingly related.	• The program developed on task basis as a sequence of goal positions. • The explicit definition of robot motion is not required. • No relation between command and action but the task is executed.

LEAD THROUGH PROGRAMMING

Lead through teaching

Manual lead through

- An operator grasps physically and moves the wrist end in the path of operation, which is recorded in the memory.
- He uses a teach button while in teach mode.
- During run mode the wrist end repeats the taught motion.
- When robot structure is complicated a geometrically similar model is used to ease the robot handling.
- More useful in continuous operations.

Applications:

- Machine loading and unloading.
- Spot welding.
- Point-to-point movements.

Powered lead through

- A control box with buttons known as teach pendent is used to control joint motors.
- The points of motion is recorded in the memory, and converted to motion programs.
- Subsequently the program is utilized to playback the motion during the cycle of operation.
- This is more useful in applications of point to point movements.

Applications:

- Spray painting.
- Full welding.
- Continuous operation.

REQUIREMENT OF GOOD PROGRAMMING LANGUAGE (ROBOT ORIENTED)

World Modeling

The setup of the work-space and the fixtures and feeders in which the parts are fixed and determined. Generally robots and the objects are confined to well defined world-space. The positional uncertainty is minimized by using restricted feeders and fixtures.

Position Specification

The position and orientation of the parts are defined in terms of co-ordinate frame. The feeder and beam locator and their features are described by using data structures provided by the language.

Motion Specification

The general **Pick and Place** activity is divided into the sequence of action such as movement of the arm from an initial configuration to grasp position, picking up an object and moving to the final configuration.

Sensory Control

To take care of the uncertainty in location and dimension of the objects in the work-envelope sensing is to be performed. The information gathered by the sensors acts as feedback from environment.

Programming Support

The programming supports like editing and debugging are provided by the sophisticated programming platform for the user to program.

VAL COMMANDS WITH DESCRIPTION

Definition	Command Statement	Explanation
1. Motion control	APPRO P1, Z1	Command to approach the point P1 in the z-direction by Z1 distance above the object.
	MOVE P1	Command to move the arm from the present position to P1.

Definition	Command Statement	Explanation
	MOVE P1 VIA P2	Asks the robot to move to P1 through point P2.
	DMOVE (J1, ΔX)	Moves the joint J1 by an increment of ΔX (linear)
	DMOVE (J1, J2, J3) ($d\alpha$, $d\beta$, $d\theta$)	Command to move joints J1, J2 and J3 by incremental angles of $d\alpha$, $d\beta$, $d\theta$ respectively.
2. Speed control	SPEED V IPS	The speed of the end effector is to be V inch per second at the time of program execution.
	SPEED R	Command to operate the arm end effector at R percent of the normal speed at the time of program execution.
3. Position control	HERE P1	Defining the name of a point as P1.
	DEFINE P1 = POINT (x, y, z, w_α, w_β, w_θ)	The command defines the point P1 with x, y, z co-ordinates and w_α, w_β, w_θ the wrist rotation angles.
	Path control: DEFINE PATH 1 = PATH (P1, P2, P3)	The path of the end effector is defined by the connection between points P1, P2 and P3 in series.
	MOVE PATH 1	Movement of the end effector along path 1.
	Frame definition: DEFINE FRAME 1 = FRAME(P1, P2, P3)	– Assigns variable name to FRAME 1 defined by points P1, P2 and P3. P1-origin, P2-point along x-axis and P3-point along xy plane.
	MOVE ROUTE: FRAME 1.	– Defines the movement in the path for frame 1.
4. End effector operation	OPEN	– Opens the gripper fingers.
	CLOSE 50 MM	– In forms gripper to close keeping 50 mm width between the fingers.

Definition	Command Statement	Explanation
	CLOSE 5 LB.	– Applies 5 Lb gripper force.
	CENTER	– Closes the gripper slowly till the establishment of contact with the object to be gripped.
	OPERATE TOOL (SPEED N RPM)	– Positioning and operating the powered tool. Here the EE is replaced by servo powered tool.
5. Operating of the sensors	SIGNAL 4, ON	The command actuates the output port 4 and turns on at certain stage of the program.
	SIGNAL 4, OFF. WAIT 13, ON	The output port 5 is turned off. The device gives a feedback signal indicating that it is on.
	REACT 16, SAFETY.	The change in signal (if any), in the input line 16, should be deviated to the sub-routine SAFETY.

DEFINITION AND STATEMENTS OF AML

Definition	AML-statement
• Definition of base frame	frame<<*x*, *y*, *z*>, EULER ROT(<ROT*x*, ROT*y*, ROT*z*>)>
• Definition of feature frame	DOT(base, << *x*, *y*, *z*>, EULER ROT(<ROT*x*, Rot*y*, Rot*z*>)>);
• Definition of motion statement	MOVE(<joint *i*, joint *j*>, <distance, angle>);
• Sensing control statement	MOVE(gripper, distance);
• Force sensing and compliance statement	Fmons = MONITOR(<SLP, SRP>, X, O, R); SLP and SRP are force sensors.

PROGRAMMING LANGUAGES FEATURES AND APPLICATION

Feature \ Language	AML	VAL
• Language base	LISP, Pascal	BASIC
• Level orientation	Robot	Robot
• Motion specification	Joints	Joints, Frame
• Sensing control	Position	Position, Force
• Robot multiplicity	NO	NO
• Robot application	IBM arm	PUMA

PROGRAM FOR PNP (PICK AND PLACE) ACTIVITY

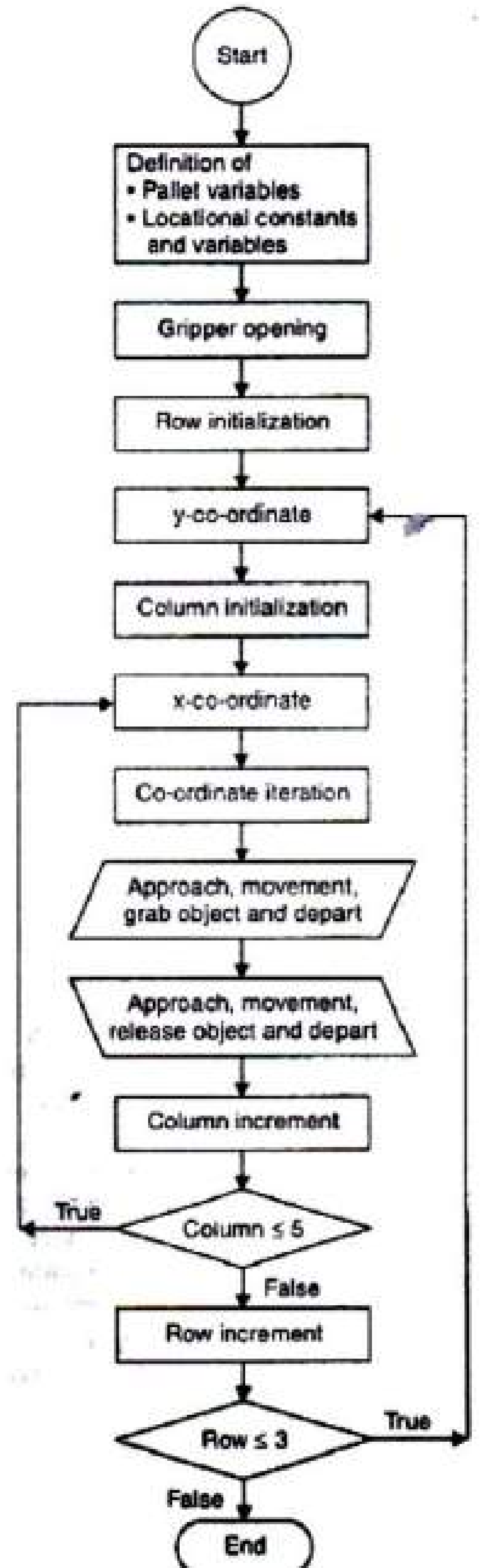
VAL STATEMENT	Statement Description
BRANCH PICK	– The branch of program indicating part pick.
MOVE INTER	– Move to a intermediate position above chute.
WAIT 12	– Wait for a incoming part to the chute.
SIGNAL 5	– Open gripper fingers (sensor control).
MOVE PICK-UP	–Move gripper and pick-up the object.
SIGNAL 6	– Close the gripper to grab the object
MOVE INTER	– Depart to intermediate position above chute.
END BRANCH	– End of pick-up activity.
BRANCH PLACE	– Start of placing activity.
MOVE Z (-50)	–Position part and gripper above the pallet.
SIGNAL 5	– Open gripper to release the part.
MOVE Z (50)	–Depart from the place (pallet) point.
END BRANCH	–End of place activity.

VAL = Variable Assembly Language

AML = A Manufacturing Language

LISP = LISt Processing (for data structure programming)

FLOW CHART FOR PALLETIZATION PROGRAM



LECTURE 8. Differential Motion

In this chapter, we are concerned not only with the final location of the end-effecter, but also with the ***velocity*** at which the end-effecter moves. In order to move the end-effecter in a specified direction at a specified speed, it is necessary to ***coordinate*** the motion of the individual joints. The focus of this chapter is the development of fundamental methods for achieving such coordinated motion in multiple-joint robotic systems. As discussed in the previous chapter, the end-effecter position and orientation are directly related to the joint displacements. Hence, in order to coordinate joint motions, we derive the ***differential*** relationship between the joint displacements and the end-effecter location, and then solve for the individual joint motions.

8.1 Differential Relationship

We begin by considering a two degree-of-freedom planar robot arm, as shown in Figure 8.1 The kinematic equations relating the end-effecter coordinates x_e and y_e to the joint displacements θ_1 and θ_2 are given by

$$x_e(\theta_1, \theta_2) = \ell_1 \cos\theta_1 + \ell_2 \cos(\theta_1 + \theta_2)$$
$$y_e(\theta_1, \theta_2) = \ell_1 \sin\theta_1 + \ell_2 \sin(\theta_1 + \theta_2)$$

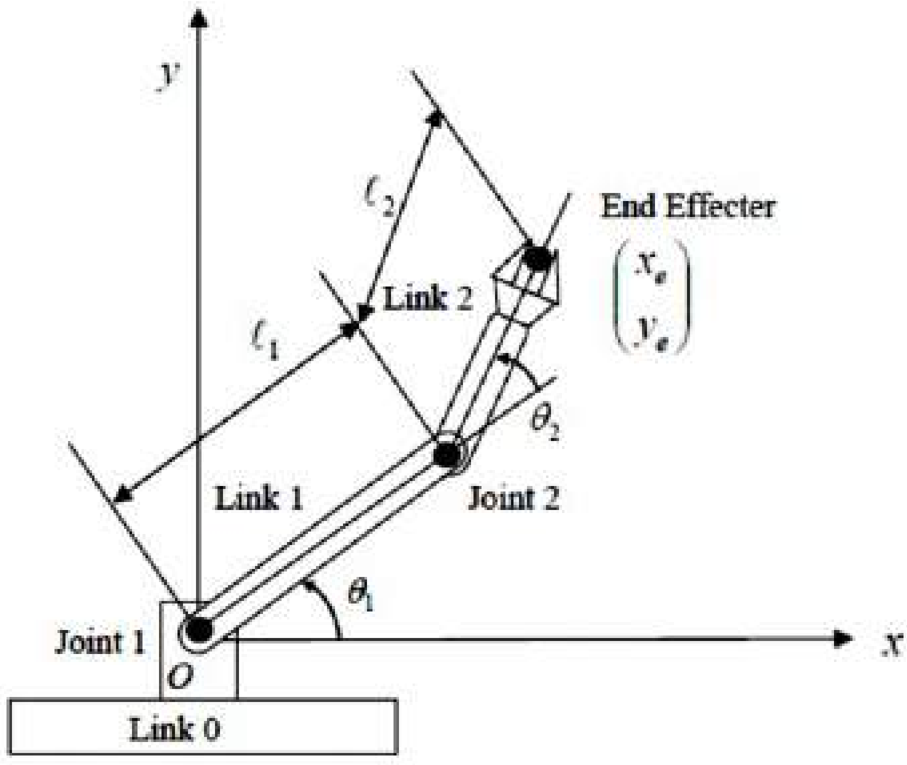


Figure 8.1 Two dof planar robot with two revolute joints

We are concerned with "small movements" of the individual joints at the current position, and we want to know the resultant motion of the end-effecter. This can be obtained by the total derivatives of the above kinematic equations:

$$dx_e = \frac{\partial x_e(\theta_1, \theta_2)}{\partial \theta_1} d\theta_1 + \frac{\partial x_e(\theta_1, \theta_2)}{\partial \theta_2} d\theta_2$$

$$dy_e = \frac{\partial y_e(\theta_1, \theta_2)}{\partial \theta_1} d\theta_1 + \frac{\partial y_e(\theta_1, \theta_2)}{\partial \theta_2} d\theta_2$$

where x_e, y_e are variables of both θ_1 and θ_2, hence two partial derivatives are involved in the total derivatives. In vector form the above equations reduce to

$$d\mathbf{x} = \mathbf{J} \cdot d\mathbf{q}$$

where

$$d\mathbf{x} = \begin{pmatrix} dx_e \\ dy_e \end{pmatrix}, \quad d\mathbf{q} = \begin{pmatrix} d\theta_1 \\ d\theta_2 \end{pmatrix}$$

and **J** is a 2 by 2 matrix given by

$$\mathbf{J} = \begin{pmatrix} \frac{\partial x_e(\theta_1, \theta_2)}{\partial \theta_1} & \frac{\partial x_e(\theta_1, \theta_2)}{\partial \theta_2} \\ \frac{\partial y_e(\theta_1, \theta_2)}{\partial \theta_1} & \frac{\partial y_e(\theta_1, \theta_2)}{\partial \theta_2} \end{pmatrix}$$

Homework: complete the Derivation of the Jacobian matrix (**J**)

Robot Dynamics

- While the kinematic equations describe the motion of the robot without consideration of the forces that produces the motion.
- The dynamics explicitly describe the relationship between force and motion
- The dynamic of the robot is necessary to consider in the design of robots, simulation and animation, and in the design of control algoritms
- We can derive the equations of motion for any *n*DOF system by using energy methods
  - All we need to know are the conservative (kinetic and potential) and non-conservative (dissipative) terms
- This is a shortcut to describing the motion of each particle in a rigid body along with the constraints that form rigid motions

Ex: 1DOF system

- To illustrate, we derive the equations of motion for a 1DOF system
  - Consider a particle of mass *m*
  - Using Newton's second law:

$$m\ddot{y} = f - mg$$

  - Now define the kinetic and potential energies:

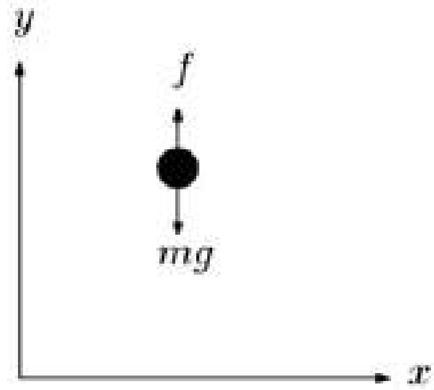
$$K = \frac{1}{2}m\dot{y}^2 \qquad P = mgy$$

  - Rewrite the above differential equation
    - Left side:

$$m\ddot{y} = \frac{d}{dt}(m\dot{y}) = \frac{d}{dt}\frac{\partial}{\partial \dot{y}}\left(\frac{1}{2}m\dot{y}^2\right) = \frac{d}{dt}\frac{\partial K}{\partial \dot{y}}$$

    - Right side:

$$mg = \frac{\partial}{\partial y}(mgy) = \frac{\partial P}{\partial y}$$



- Thus we can rewrite the initial equation:

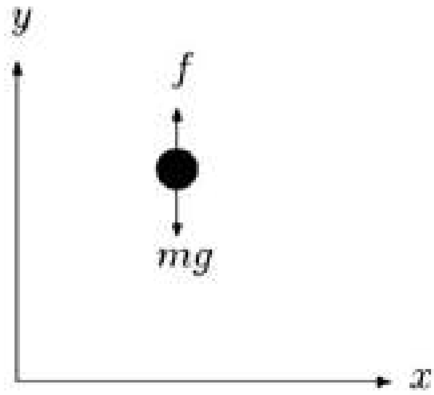
$$\frac{d}{dt}\frac{\partial K}{\partial \dot{y}} = f - \frac{\partial P}{\partial y}$$

- Now we make the following definition:

$$L = K - P$$

- *L* is called the *Lagrangian*
  - We can rewrite our equation of motion again:

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{y}} - \frac{\partial L}{\partial y} = f$$



- Thus, to define the equation of motion for this system, all we need is a description of the potential and kinetic energies

- If we represent the variables of the system as generalized coordinates, then we can write the equations of motion for an *n*DOF system as:

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} = \tau_i$$

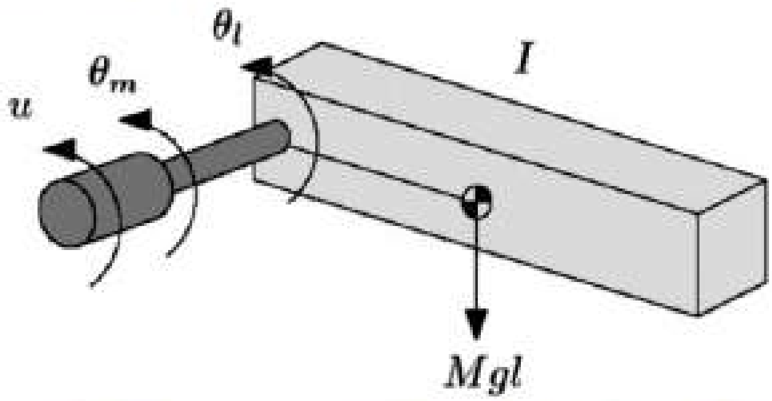
- Single link, single motor coupled by a drive shaft
  - θ_m and θ_l are the angular displacements of the shaft and the link respectively, related by a gear ratio, *r*:

$$\theta_m = r\theta_l$$

  - Start by determining the kinetic and potential energies:

$$K = \frac{1}{2}J_m\dot{\theta}_m^{\,2} + \frac{1}{2}J_l\dot{\theta}_l^{\,2} = \frac{1}{2}\left(r^2J_m + J_l\right)\dot{\theta}_l^{\,2}$$

$$P = \frac{MgL}{2}\left(1 - \cos\theta_l\right)$$



  - J_m and J_l are the motor/shaft and link inertias respectively and *M* and *L* are the mass and length of the link respectively

- **Let the total inertia, *J*, be defined by:**

$$J = r^2 J_m + J_l$$

- **Now write the Lagrangian:**

$$L = \frac{1}{2} J \dot{\theta}_l^2 - \frac{MgL}{2}(1 - \cos\theta_l)$$

- **Thus we can write the equation of motion for this 1DOF system as:**

$$J\ddot{\theta}_l + \frac{MgL}{2}\sin\theta_l = \tau_l$$

- **The right side contains the non-conservative terms such as:**
  - **The input motor torque:** $u = r\tau_m$
  - **Damping torques:** $B = rB_m + B_l$
- **Therefore we can rewrite the equation of motion:**

$$J\ddot{\theta}_l + B\dot{\theta}_l + \frac{MgL}{2}\sin\theta_l = u$$