Module 11 : Example study of robots

Lecture 39 : PUMA Robots- A Case Study

Objectives

In this course you will learn the following

- Geometric Configuration

- Kinematics of PUMA Robots:

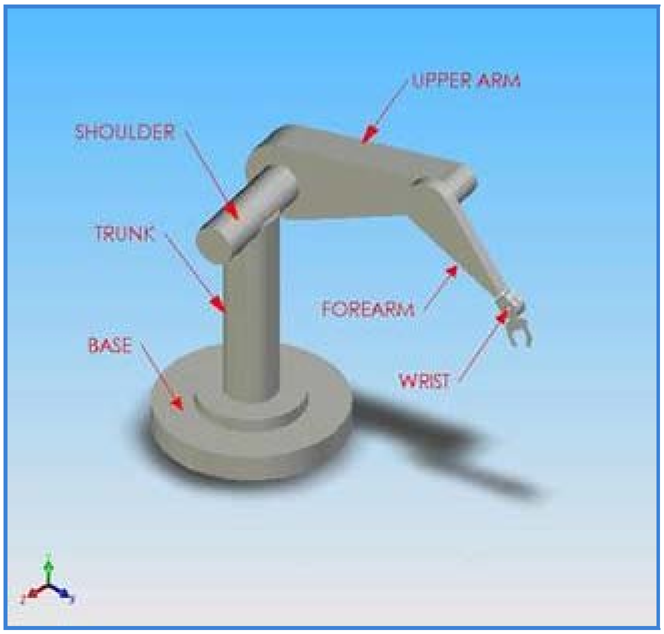
- Inverse Kinematics

- Dynamics of PUMA Robot:

- Task Planning With PUMA Robot

In 1969 Victor Scheinman developed the “Stanford arm” at Stanford University . This was 6-axis articulated robot using all electric actuators. The ability of this robot to follow predefined arbitrary paths accurately in space turned in to its use to more sophisticated applications like assembly and arc welding. Scheinman developed one more design called “MIT ARM”. Further “Unimation” developed it for commercial applications and it is then called as the *“Programmable Universal Machine for Assembly (PUMA) - Robots”* .

Following is the basic diagram for the PUMA robots. We will use this diagram for further references.



Click here for Video Clip

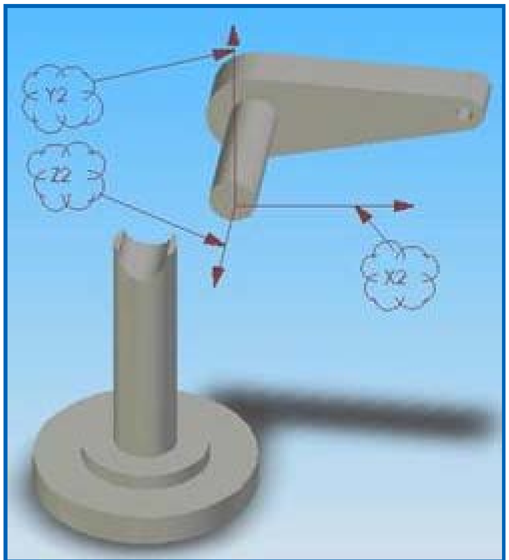
The general parameters that are used to define industrial robots are Number of Axes, Degrees of Freedom, Working Envolope, Payload, Speed-Acceleration of Joints, Kinematic Configuration, Accuracy, Control, Power Source and Drives.

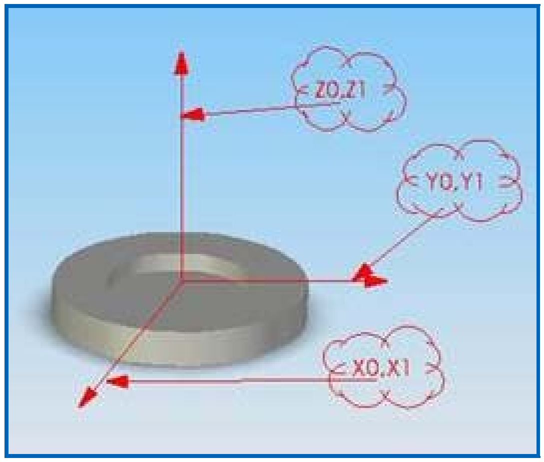
It is obvious to note that specifications of PUMA robot will change as per applications and sizes. But for getting generalized sense about the figures and values these specifications can be given as shown in following table.

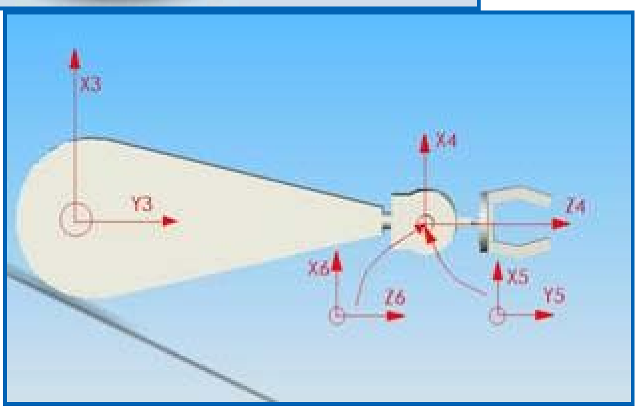
Kinematics	Axes/pairs	6 / Revolute
	Drives	DC Motors
	Control	Numerical
	Positional Control	Incremental Encoders
WORK Envolope and Angular Reach	Minimum Reach	0.125 M
	Maximum Reach	0.406 M
	Limit Joint 1	300-320 deg
	Limit Joint 2	250-300 deg
	Limit Joint 3	270-300 deg
	Limit Joint 4	270-300 deg
	Limit Joint 5	200-250 deg
	Limit Joint 6	300-360 deg
Load carrying capacity	Load under optimum Speed	This is specified as a load in kg to be held at a certain distance from given joint (Joint 5 is generally used) e.g. 5 kg at distance 0.2 m from joint 5
Performance parameters	Repeatability	generally in mm
	Maximum Speed	1-2 m/s
Surrounding Conditions	Temperature	e.g. 0- 50 deg C
	Power Supply	110/220/240 50-60 Hz 1kw -1 phase
Self Weight	Arm	10-100kg
	Cabinet / Controllers etc	50-200kg

Geometric Configuration

Now we will see the basic geometric configuration of the PUMA robots. Attaching the coordinates to each joint is based on the same principle as seen earlier. For revolute pair the Z axis is taken along the axis of rotation of the link. And other axes are attached using Right-Handed Coordinate system. We will start attaching the coordinate frames from base of robot. So it will be [X0, Y0, Z0] system. Also for simplicity of kinematic analysis we attach the coordinate system of first arm in particular fashion such that its origin and revolute axes (i.e. Z0 and Z1) are coincident. This is shown in the figure below.







Now similarly the coordinate frames can be attached to other links of the PUMA Robots. Coordinate frame [X2, Y2, Z2] attached to link arm 2 is shown in the following figure.

Click here for Video Clip

Frames 4, 5 and 6 are coincident at one origin and this is important construction that provides *Roll, Pitch and Yaw* motions. This particular configuration gives the “Wrist” like motion at the end.

Now following the Denavit- Hartenberg Nomenclature the table for PUMA robot can be written as:

Link (i)	α(i-1) deg	a(i-1)	d(i)	θ(i) deg
1	0	0	0	θ(1)
2	-90	0	0	θ(2)
3	0	a(2)	d(3)	θ(3)
4	-90	a(3)	d(4)	(4)

				θ
5	90	0	0	θ(5)
6	-90	0	0	θ(6)

Where α, a, d and θ are the link parameters that define geometry of the manipulator. Description for these parameters is given as;

α(i-1) : The angle from Z(i-1) to Z(i) measured about X(i-1)

a(i-1) : Distance from Z(i-1) to Z(i) measured along X(i-1)

d(i) : Distance from X(i-1) to X(i) measured along Z(i) and

θ(i) : The angle from X(i-1) to X(i) measured about Z(i).

Values of ‘a' and ‘d' are dependent on link lengths and minimum distance between corresponding X axes of two connecting links. As all the six joints are revolute joints, θ's are dependent on particular position of the end effecter.

As we have seen earlier that PUMA robot is used as a very preliminary robot for industrial and Laboratory applications, the links are designed in such a way that the robot will get Human Arm like motion. And hence the parts of PUMA are named accordingly. This Human Arm kind of motion is necessary to do basic jobs at workplace, e.g. Pick & Place, Painting, Assembly of Common components, Arc Welding. For most of the motions and assembly operations the compliance of end effecter is necessary. To achieve the wrist like motion at the end effecter, a special configuration of links 4, 5 and 6 is designed. Coordinate frames of these links are such that the all joint axes of these links intersect at common point which coincides with the origin of frames. Further these joint axes of 4, 5 and 6 are mutually orthogonal.

Kinematics of PUMA Robots:

Direct Kinematics

We have seen the method of attaching coordinate frames to each links. Generally universal coordinate frame is also taken coincident with the stationary base of the robot. Location of the work piece / target is given in terms of its universal coordinates. As we know the tool that is griped in the wrist of PUMA robot should operate on work piece, it is vital to find the relation between tool / wrist frame and coordinate frame attached to the work piece. So the direct kinematics is used to find out the final position of the wrist / tool in the workspace when the joint variables are provided. For PUMA robots the joint variables are joint angles because all the joints are revolute type.

Transformation of Coordinate Frames

The task becomes simpler with the knowledge of link parameters, transformation matrices giving relation of one link with the other. Generalized transformation matrix for transformation occurred from frame 1 to frame 2 with usual notations is given as:

$$ {}^{1}_{2}T = \begin{bmatrix} r11 & r12 & r13 & \Delta x \\ r21 & r22 & r23 & \Delta y \\ r31 & r32 & r33 & \Delta z \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

Here the part of rotation matrix i.e. R= [r(i , j)] ... i, j = 1:3; can be found out by using standard methods of *Z-Y-Z rotations* or *rotation about a vector* or *roll-pitch-yaw rotation* methods. Now for 6 link manipulator of PUMA robot will have six such transformation matrices involved in finding the final transformation matrix. This final transformation matrix will give us the relation between base frame and tool frame.

$$ {}^{0}_{6}T = {}^{0}_{1}T * {}^{1}_{2}T * {}^{2}_{3}T * {}^{3}_{4}T * {}^{4}_{5}T * {}^{5}_{6}T $$

Once we get the expression for each of these matrices which are functions of link parameters given in

Denavit- Hartenburg nomenclature. So we can write for two links as:

$$
{}^{i-1}_{i}T = \begin{bmatrix} Cos\theta_i & -Sin\,\theta_i & 0 & a_{i-1} \\ Sin\,\theta_i \cdot Cos\,\alpha_{i-1} & Cos\theta_i \cdot Cos\,\alpha_{i-1} & -Sin\,\alpha_{i-1} & -Sin\,\alpha_{i-1} \cdot d_i \\ Sin\,\theta_i \cdot Sin\,\alpha_{i-1} & Cos\theta_i \cdot Sin\,\alpha_{i-1} & Cos\,\alpha_{i-1} & Cos\,\alpha_{i-1} \cdot d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

If we observe the link parameters table carefully, it can be then clearly noted that the values for α(i-1) are either 0 / -90 / 90. So this gives sin α(i-1) or cos α(i-1) = 0 or -1 or 1. This provides huge simplicity in calculating the transformation matrices. Still the job of computations is not simpler. For illustration just some of the elemental values of final transformation matrix of PUMA Robot are given here, the expression for remaining can be either derived manually or found in any of the basic robotics textbook.

$$
\begin{aligned}
r_{11} &= C_1[C_{23}(C_4C_5C_6 - S_4S_5) - S_{23}S_5C_5] + S_1(S_4C_5C_6 + C_4S_6) \\
r_{21} &= S_1[C_{23}(C_4C_5C_6 - S_4S_6) - S_{23}S_5C_6] - C_1(S_4C_5C_6 + C_4S_6) \\
r_{31} &= -S_{23}(C_4C_5C_6 - S_4S_6) - C_{23}S_5C_6
\end{aligned}
$$

$$
\begin{aligned}
r_{12} &= C_1[C_{23}(-C_4C_5S_6 - S_4C_6) + S_{23}S_5S_6] + S_1(C_4C_6 - S_4C_5S_6) \\
r_{22} &= S_1[C_{23}(-C_4C_5S_6 - S_4C_6) + S_{23}S_5S_6] - C_1(C_4C_6 - S_4C_5S_6) \\
r_{32} &= -S_{23}(-C_4C_5S_6 - S_4C_6) + C_{23}S_5C_6
\end{aligned}
$$

$$
\begin{aligned}
r_{13} &= -C_1(C_{23}C_4S_5 + S_{23}C_5) - S_1S_4S_5 \\
r_{23} &= -S_1(C_{23}C_4S_5 + S_{23}C_5) - C_1S_4S_5 \\
r_{33} &= S_{23}C_4S_5 - C_{23}C_5
\end{aligned}
$$

$$
\begin{aligned}
\Delta X &= C_1[a_2C_2 + a_3C_{23} - d_4S_{23}] - d_3S_1 \\
\Delta Y &= S_1[a_2C_2 + a_3C_{23} - d_4S_{23}] + d_3C_1 \\
\Delta Z &= -a_3S_{23} - a_2S_2 - d_4C_{23}
\end{aligned}
$$

Where $C1 = Cos(\theta 1);\ S1 = Sin(\theta 1);\ S23 = Sin(\theta 2 + \theta 3)$ and so on.

Inverse Kinematics

As seen earlier for given set of joint angles the position of the tool / end-effecter can be found out by using Direct Kinematics. Similarly, for most of the applications it is important to know: if given a desired position and orientation of tool relative to the base, how to find out set of joint angles which will achieve this desired result.

For PUMA robots the solution for inverse kinematics problem can be found out by Algebraic or Geometric methods. Here we will only see the algebraic method of solution for PUMA robots. For derivation of Geometric Method the paper on “Geometric Approach in Solving Inverse Kinematics of PUMA Robots” by C.S.G. LEE & M. ZIEGLER, can be referred.

We have seen that ${}^0_1T = f(\theta 1) \cdots\cdots ; {}^5_6T = f(\theta 6)$. Also as we know the final location of the end effecter we know the numerical value of matrix ${}^0_6T$. We need to make the expression independent from θi at each step and find the values of θi. Now to start from $\theta 1$, first make the expression independent of ?1;

$$
[{}^0_1T(\theta 1)]^{-1} \cdot {}^0_6T = {}^1_2T(\theta 2) \cdots\cdots {}^5_6T(\theta 6)
$$

Solving for both the sides and comparing its elements we will get one of the expression as ($-S_1\Delta X + C_1\Delta Y = d_3$). Solving this equation we can get expression for $\theta 1$ as:

$$\theta_1 = \tan^{-1}(\Delta Y / \Delta X) - \tan^{-1}(d_3 / \pm\sqrt{\Delta X^2 + \Delta Y^2 + \Delta Z^2})$$

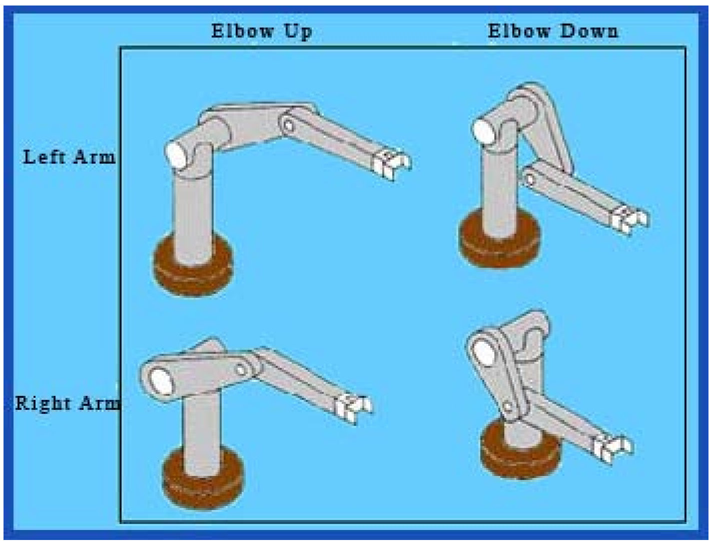
From above expression it is clear that for given endpoint there would be two possible solutions for $\theta 1$.

Similarly expressions for other joint angles can be found out e.g.

$$\theta_3 = \tan^{-1}\left(\frac{a_3}{d_4}\right) - \tan^{-1}\left(\frac{K}{\pm\sqrt{a_3^{\ 2} + d_4^{\ 2} - K^2}}\right)$$

$$where\ \ K = \left(\frac{\Delta X^2 + \Delta Y^2 + \Delta Z^2 - a_2^{\ 2} - a_3^{\ 2} - d_3^{\ 2} - d_4^{\ 2}}{2a_2}\right)$$

In general if we observe there are two possibilities for each joint angle for first three joints.



$$\theta 1 = \begin{cases} +1\ Right\ Arm \\ -1\ Left\ Arm \end{cases}$$

$$\theta 2 = \begin{cases} +1\ Elbow\ Up \\ -1\ Elbow\ Down \end{cases}$$

$$\theta 3 = \begin{cases} +1\ Wrist\ Down \\ -1\ Wrist\ Up \end{cases}$$

Flip of wrist joint angle gives two more possible combinations as

$$\theta 4' = \theta 4 + 180^0, \theta 5' = -\theta 5 - \theta 5, \theta 6' = \theta 6 + 180^0$$

From above description we can say that for any given point in workspace can be reached by PUMA robot by "Eight Possible Orientations". Four are as shown in figure and the other four achieved with flip position of wrist angles. If we look at this feature of PUMA robot from practical point of view, this ability stands useful in many tasks like assembly, fixing, painting and welding etc. Also if one of the orientations is forbidden due to physical obstacles then the end effecter can reach the same work piece location from some other orientation.

Dynamics of PUMA Robot:

By Newton 's formula the force acting on the link at center of its mass is given by $F = m\dot{v}_c$; and by using Euler's formula the moment acting on the body which causes the motion is given by $N = {}^cI\dot{\omega} + \omega \times {}^cI\omega$. For calculating force and moments on each links involved in the entire manipulator we need to use the rotation, translation and transformation matrices in the equations. In a general form for any manipulator we can write the dynamic equation as,

$$\tau = D(\Theta)\ddot{\Theta} + C(\Theta\dot{\Theta})\dot{\Theta} + G(\Theta)$$

Where $D(\Theta)$ is n x n mass matrix, $C(\Theta\dot{\Theta})$ is vector of centrifugal and coriolis terms. It is actually formed of two components. C1 which n x n(n-1)/2 matrix of coriolis terms that are function of [$\dot{\theta}_i \cdot \dot{\theta}_j$] where $i \neq j$. And C2 will be other component which is n x n matrix giving centrifugal terms that are function of [$\dot{\theta}_i \cdot \dot{\theta}_j$] where i=j., and $G(\Theta)$ is n x 1 vector of gravity terms.

The full dynamic equations of the PUMA robot model can be derived based on the Lagrange equation given by

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}_i}\right) - \frac{\partial L}{\partial \theta_i} = \tau_i \text{ Where L= KE – PE;}$$

KE is the kinetic energy and PE is the potential energy, and for first three links of PUMA Robot these entities can be given as,

$$\mathrm{KE} = \frac{1}{2}\sum_{i=1}^{3}\sum_{j=1}^{i}\sum_{k=1}^{i}\left\{(\mathrm{d}_{jk})^T \dot{\theta}_j \dot{\theta}_k\right\}$$

$$PE = \sum_{i=1}^{3}\left\{-m_i g\left(T_0^i r_i\right)\right\}$$

Here all the parameters are with their usual notations. ‘d' is function of jacobian matrix and inertia matrix. T is the transformation matrix and r is the distance of centre of mass from the origin of coordinate system for ‘i'th link.

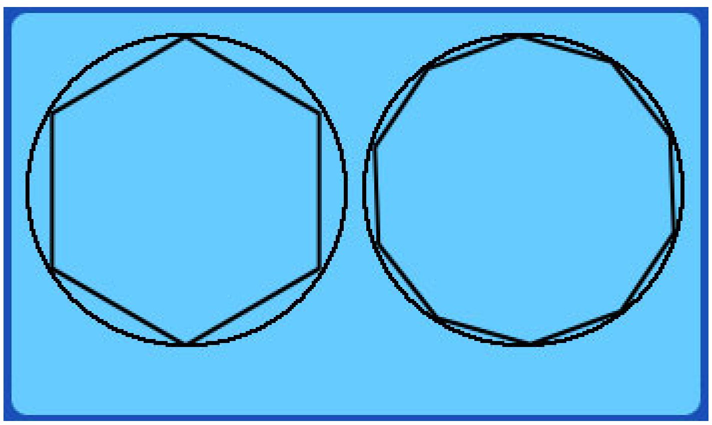
The term G in the dynamic equation can be directly given as, $G = \frac{\partial PE}{\partial \theta}$ hence here we can write

$$G_i = \sum_{j=i}^{3}(-mg^T \frac{\partial T_0^j}{\partial \theta_i} r_j)$$

For rest of the three joints the equations can also be derived by taking i=1:6. But from above discussion it is quite clear that calculation of such parameters is a tedious job. And hence if the external toques are provided the above equations can be solved by using numerical methods in MATLAB or similar softwares.

Task Planning With PUMA Robot

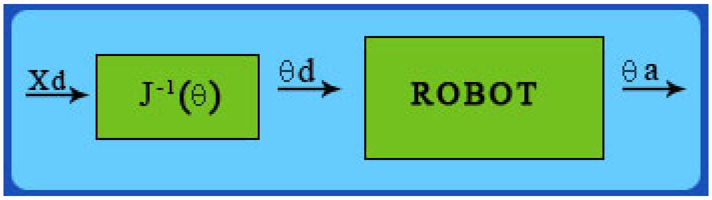
The discussion that we had till now was some preliminary basics that are totally related with some theoretical aspects of robots like PUMA. But being it an industrial robot the user would be more interested in knowing how to achieve the desired tasks by using PUMA robots. As we have seen earlier the general tasks for which the PUMA robots are being used in industry are Assembly, Fixing, Welding, Painting etc. Here for illustration we take a simple example of drawing a geometric figure say CIRCLE by using PUMA Robots.



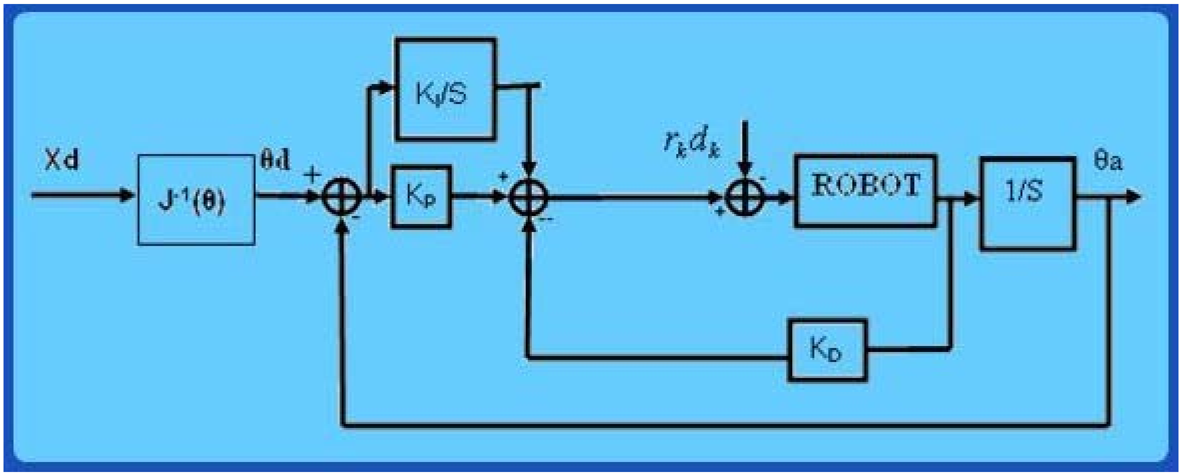
Now for defining a circle we should have an equation of particular circle. But it is difficult to describe the equation of contour to any of the hardware and its controller. Therefore better option is to define a circle through an array of points lying on its periphery. It is quite clear that more number of points on the periphery is provided more will be the accuracy in drawing circle. Suppose we are joining the points by straight lines then taking 10 equidistant points is more preferable than taking equidistant six points on periphery. This is shown in the figure above. Anyways we are free to choose the trajectory between the two points but in general the above statement regarding number of points and accuracy still stands true.

Now what we have is the array of points with their coordinates that are describing the figure, i.e. circle here. Now as discussed earlier we need to find the required joint variables' set that can achieve these points. One major aspect should be noted here is that we are not only dealing with whether the robot arm will reach the prescribed points or not, but we are interested in achieving these positions by robot arms as per given order of points at given instance of time. Hence it will also involve the dynamic part of analysis that is discussed above.

As we know the inverse jacobian operation converts the Cartesian coordinate's space into a joint variable space. The simple block diagram below shows this operation.



But as we are dealing here with a real mechanical system it is near to impossible to say ? d = ? a with just above drawn configuration. And hence we need to implement control strategies for getting the actual output nearer to desired output. In previous lectures we have already discussed dynamics, varies control strategies and their effects on the performance of robotic systems. Hence based on these strategies we can develop a similar model for PUMA robot. The feedback is taken from various sensors and encoders. The actual position is identified which in turn reflects the error between desired and actual values. Using these errors in different control strategies the compensation is done.



In this way for given set of points the desired output is achieved by obtaining the correct values of joint variables. The other parts of controller and actuators are not shown here but they are similar to those discussed in earlier chapters.

The above discussion is useful for getting the feel about how the task planning for particular example can be done. Actually the calculation part is left on the microprocessor/ microcontrollers of the PUMA robots. And some programming languages are used for the ease of users. VAL, TRC are some of the famous interfaces used for programming for PUMA Robots. Now a day some MATLAB models are also available for PUMA robots.

The motivation of PUMA Robot Case study is just to give some practical insight of the aspects that we have seen in earlier lectures. Although designing each and every part of PUMA Robot will be a too long process, it is expected that this will lay some basic footwork. This Case study will give some basic steps when the user wants to design their own Robots.

Recap

In this course you have learnt the following

- Geometric Configuration
- Kinematics of PUMA Robots:
- Inverse Kinematics
- Dynamics of PUMA Robot:
- Task Planning With PUMA Robot