Dr. Norbert Cheung's
Lecture Series

Level 5 Topic no: 11

Multi-Axes Issues

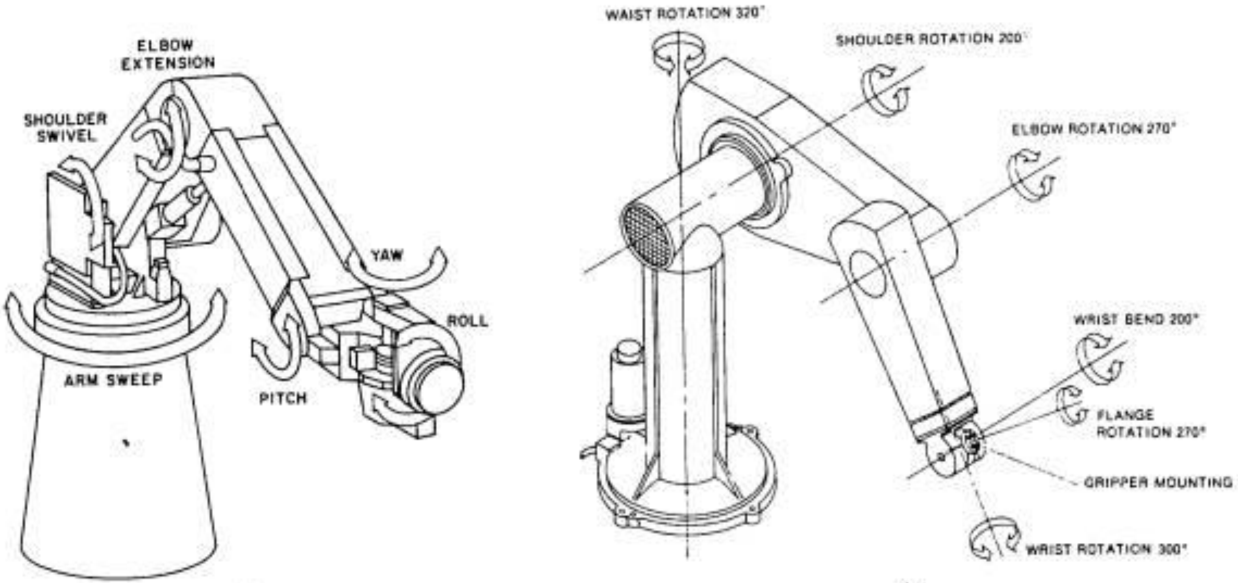
Contents

1. Introduction
2. Relation between Cartesian Positions and Rotary Angles
3. Robot Tasks Classification
4. Implementing Force Control in Robots
5. Work Method Analysis
6. Robot Arm Kinematics
7. Robot Trajectory Planning
8. Controller Design for Robots

Email: norbertcheung@szu.edu.hk
Web Site: norbert.hk

1. Introduction

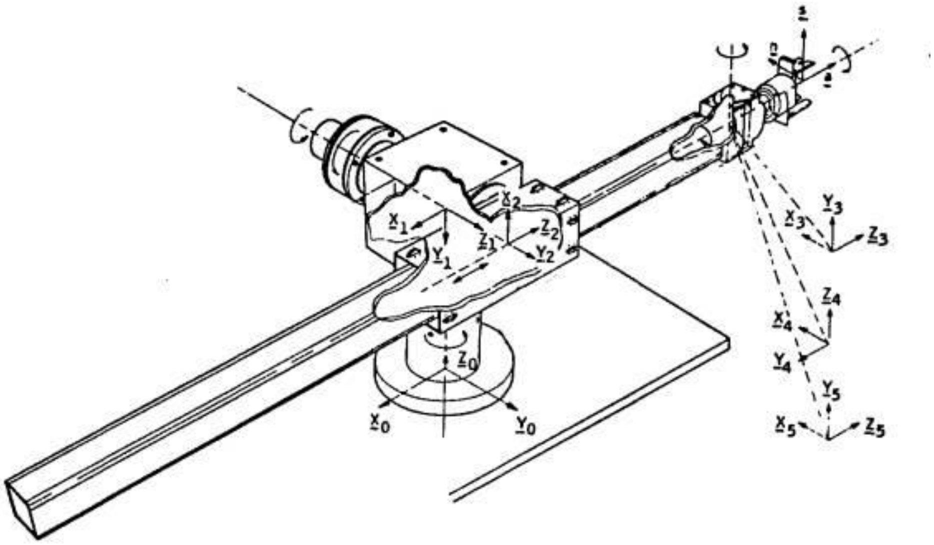
Robotic manipulators are typical application of multi-axis motion control system. It can be further divided into rotary type and Cartesian type.



(a) (b)

Fig. 5. Examples of industrial robots. (a) Cincinnati Milacron T3. Courtesy of Cincinnati Milacron. (b) Unimation PUMA 600. Courtesy of UNIMATION®, Inc.

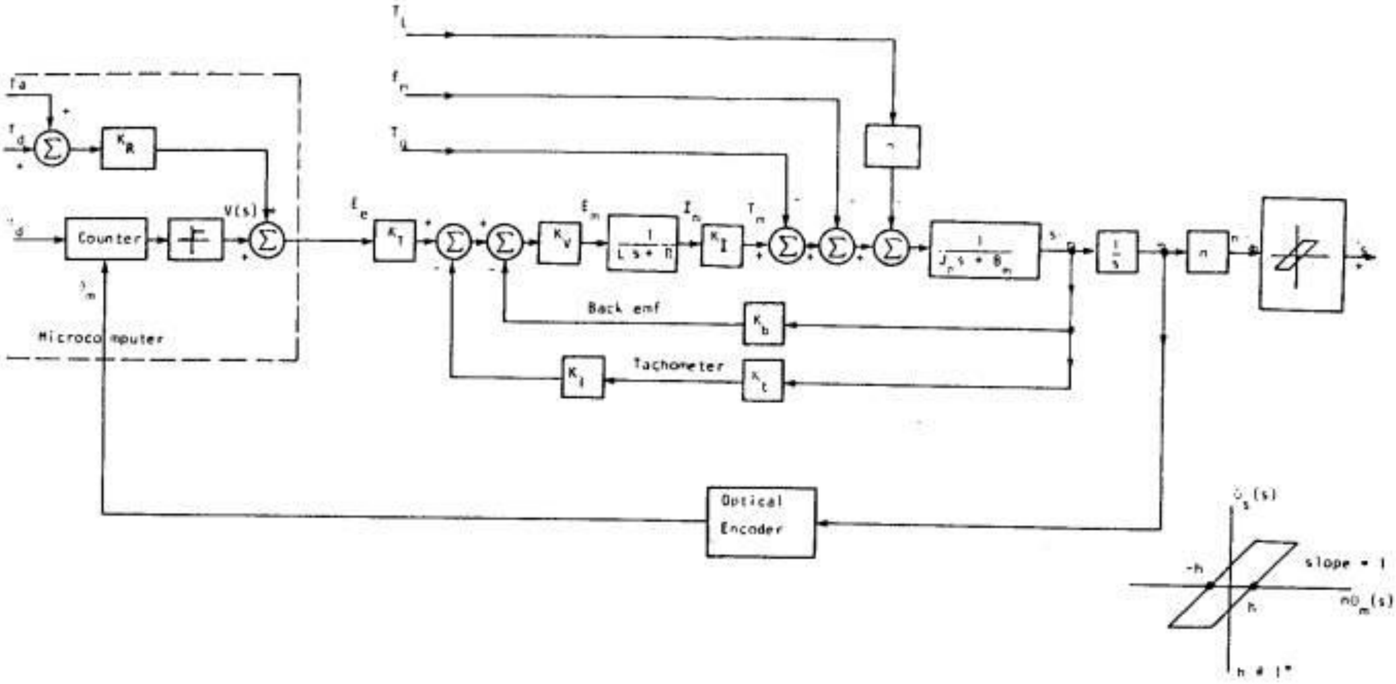
Rotary type of robotic arms



(a)

Cartesian type of robotic arm, the Stanford arm

This is a typical position control block diagram of a Stanford arm. Note the function of the individual blocks. Feed forward algorithm is used extensively. Can you identify one of them?



(b)

Fig. 6. (a) The Stanford Arm. (b) Block diagram of a positional control system.

The feed forward mechanism

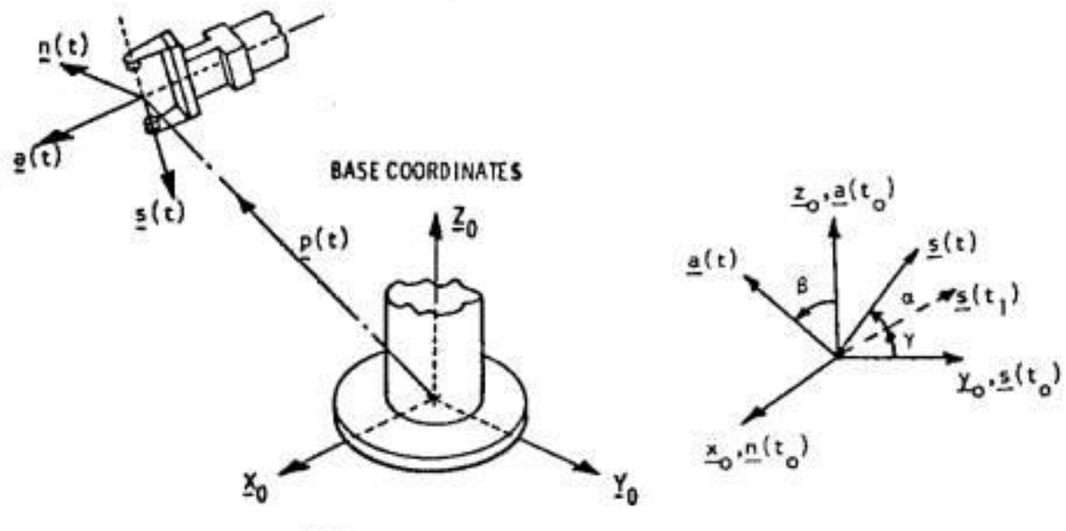
2. Relation between Cartesian Positions and Rotary Angles

- 6 joints, 6 degrees of freedoms

- Defines by a Cartesian coordinates: described by a position vector p(t), an orientation vector (or approach vector) a(t), and a unit sliding vector s(t). All these are referenced to base co-ordinates.

- For convenience, it is sometimes expressed as a unit normal coordinate n(t), with a(t) and s(t) based on this vector.

- Hence n(t)=s(t) x a(t)



(a) (b)

Fig. 7. (a) Position and orientation vectors of the hand. (b) Euler angles of orientation.

$$
\begin{aligned}
\Re &= \boldsymbol{R}(z_0,\gamma)\boldsymbol{R}(y_0,\beta)\boldsymbol{R}(z_0,\alpha)\\
&= \begin{bmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}
\begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix}\\
&\quad \cdot \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}.
\end{aligned}
\tag{1}
$$

Since $[\boldsymbol{n}(t_0)\ \ \boldsymbol{s}(t_0)\ \ \boldsymbol{a}(t_0)]$ aligns with $[x_0\ \ y_0\ \ z_0]$ originally then

$$
\begin{cases}
\boldsymbol{n}(t) = \Re \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \\
\boldsymbol{s}(t) = \Re \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \\
\boldsymbol{a}(t) = \Re \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}.
\end{cases}
\tag{2}
$$

$$
\begin{aligned}
&[\boldsymbol{n}(t)\ \ \boldsymbol{s}(t)\ \ \boldsymbol{a}(t)\ \ \boldsymbol{p}(t)]\\
&\qquad = \Re \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} + [\boldsymbol{0}\ \ \boldsymbol{0}\ \ \boldsymbol{0}\ \ \boldsymbol{p}].
\end{aligned}
\tag{3}
$$

To simplify the representation of the hand and the mathematical transformation of its orientation and position, the above operations described by (3) may be written as

$$
\begin{aligned}
\boldsymbol{H}(t) &= \begin{bmatrix} \boldsymbol{n}(t) & \boldsymbol{s}(t) & \boldsymbol{a}(t) & \boldsymbol{p}(t) \\ 0 & 0 & 0 & 1 \end{bmatrix}\\
&= \begin{bmatrix} & \Re & & \boldsymbol{p} \\ 0 & 0 & 0 & 1 \end{bmatrix}
\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
\end{aligned}
\tag{4}
$$

3. Robot Tasks Classification

Simple tasks for robots - 4 classes of problems

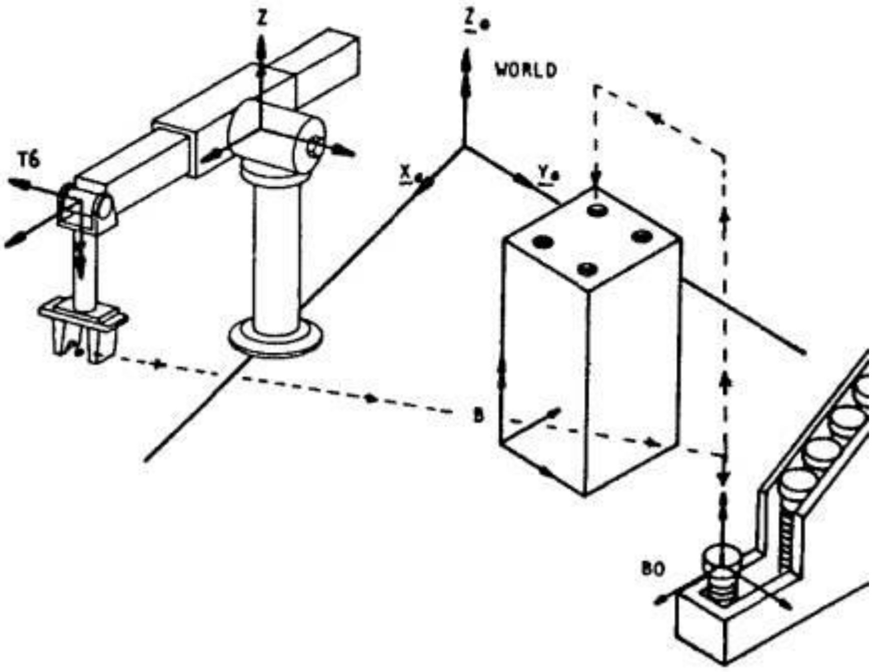


Fig. 8. Simple robot task for illustration.

TABLE I

		Is the work space free from obstacles?	
		Yes	No
Must the hand follow the specified path?	No	Class 1. Positional Control Problem	Class 4. Positional Control Plus On-line Collision Avoidance Travelling
	Yes	Class 2. Path Tracking Problem	Class 3. Off-line Collision-free Path Planning Plus On-line Path Tracking

4 distinct types of motion problems

- position control
- path tracking
- collision free path planning
- collision avoidance traveling

Common high-level programming language for robots

- WAVE programming
- POINTY
- AUTOPASS
- AML
- RAPT
- No uniform language

Optimization Control - extremely difficult, because

- Nonlinear dynamic behavior, centrifugal forces
- Strict constraints, restricted area
- Strict performance index

Complicated Task – example 1

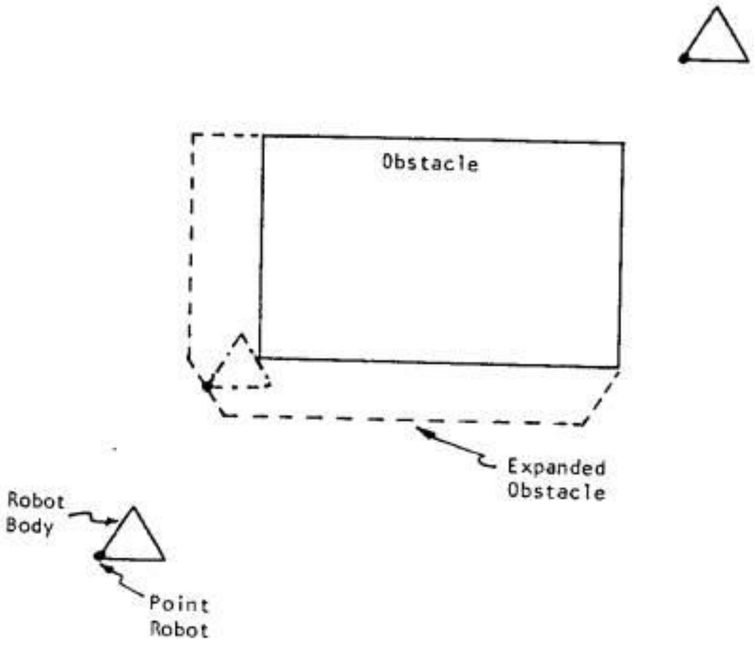
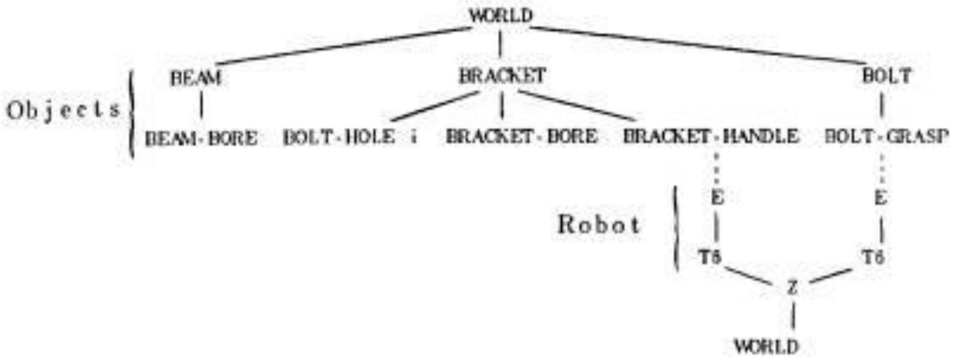


Fig. 9. Robot sliding around the obstacle with no rotation.

Complicated Task – example 2

TABLE II



The “world” as known by the robot

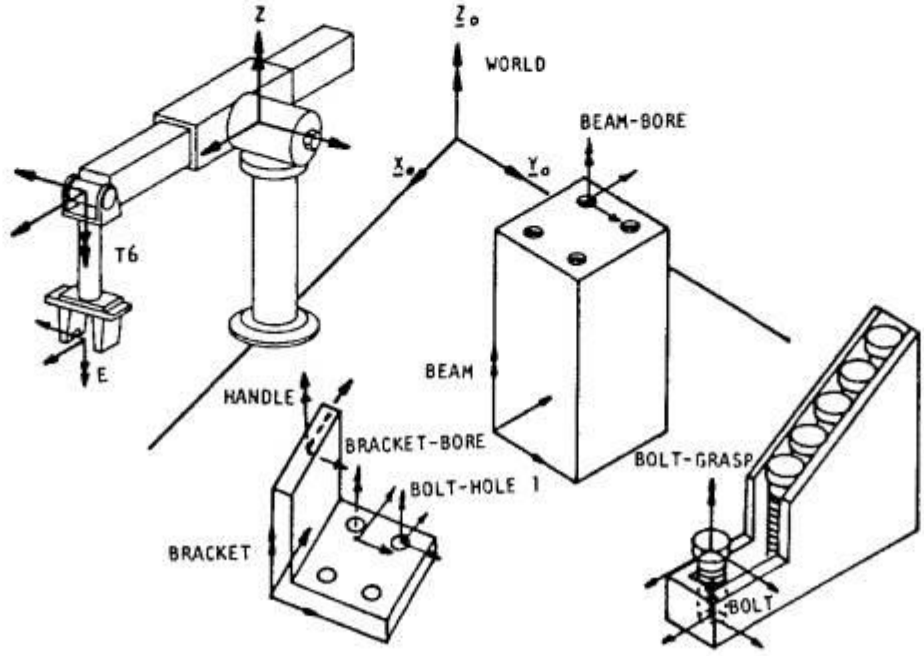


Fig. 10. Bolting a bracket task.

The robot assembly layout

4. Implementing Force Control in Robots

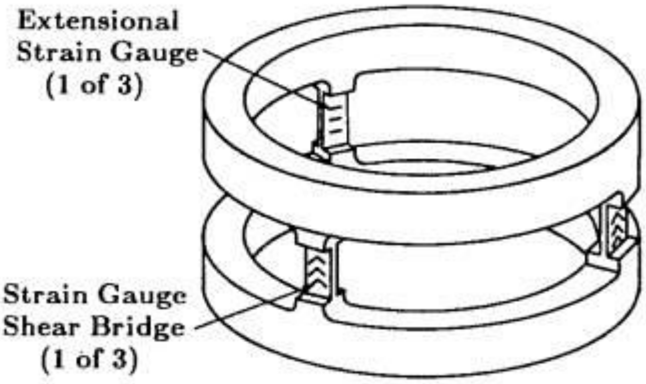
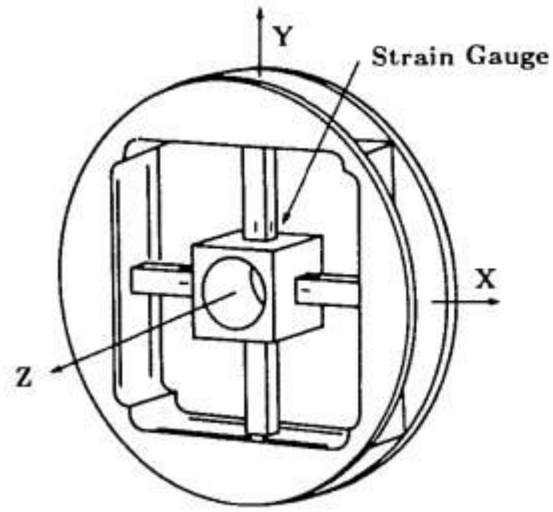


Fig. 17. Draper Laboratory force-sensing wrist.

The force sensing wrist



$$\begin{bmatrix} F_X \\ F_Y \\ F_Z \\ M_X \\ M_Y \\ M_Z \end{bmatrix} = \begin{bmatrix} 0 & 0 & c_{13} & 0 & 0 & 0 & c_{17} & 0 \\ c_{21} & 0 & 0 & 0 & c_{25} & 0 & 0 & 0 \\ 0 & c_{32} & 0 & c_{34} & 0 & c_{36} & 0 & c_{38} \\ 0 & c_{42} & 0 & 0 & 0 & c_{46} & 0 & 0 \\ 0 & 0 & 0 & c_{54} & 0 & 0 & 0 & c_{58} \\ c_{61} & 0 & c_{63} & 0 & c_{65} & 0 & c_{67} & 0 \end{bmatrix} \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ \epsilon_4 \\ \epsilon_5 \\ \epsilon_6 \\ \epsilon_7 \\ \epsilon_8 \end{bmatrix}$$

Fig. 16. Scheinman force-sensing wrist.

The 6 degrees of freedom force sensor

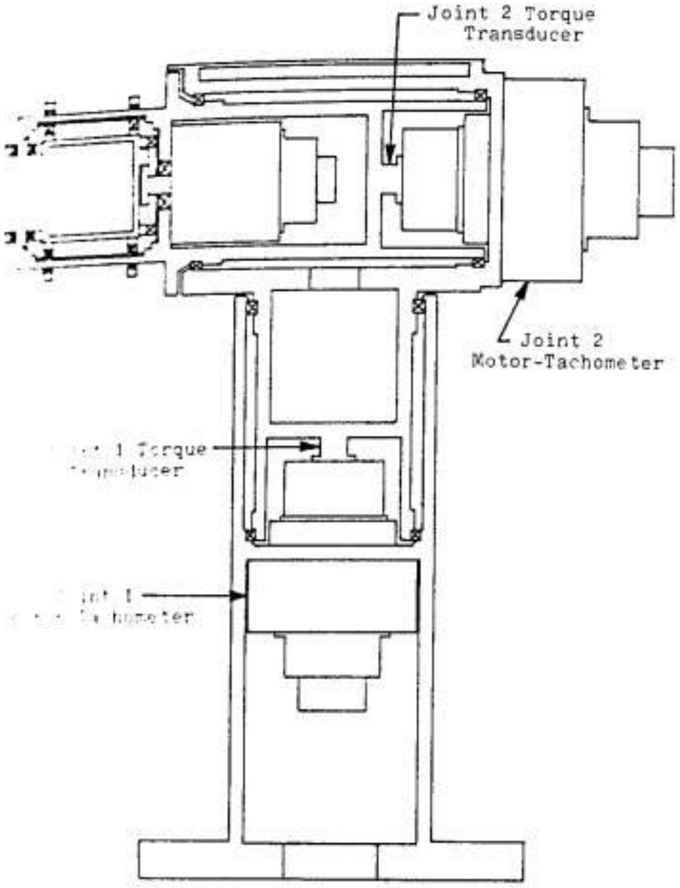


Fig. 18. Joints 1 and 2 assembly of Stanford Arm.

Installing the sensors in the robot arm

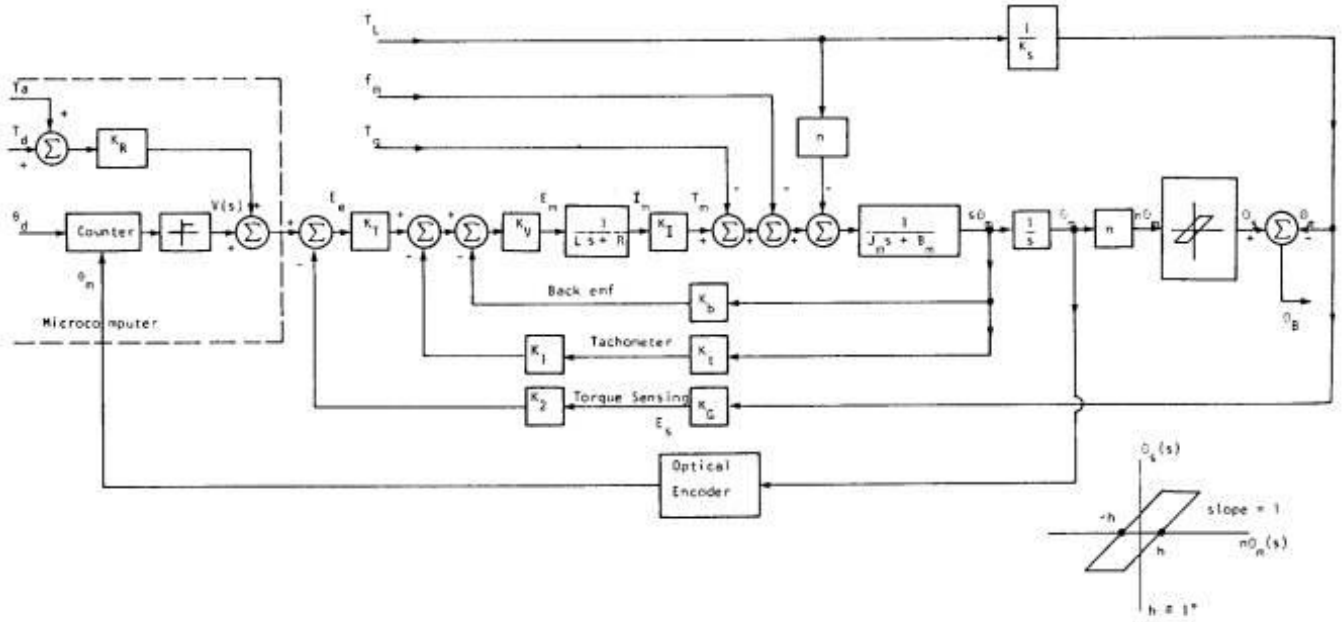


Fig. 20. Block diagram of a positional control system for a torque-sensing joint.

Position Control with torque sensing

5. Work Method Analysis

WMA is a method of commanding the robot to do certain tasks, by learning from the human operator.

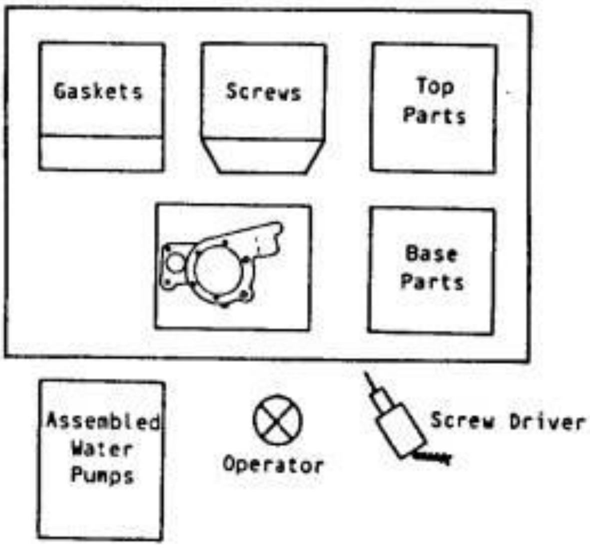


Fig. 22. Workplace for human operator pump assembly task.

Left hand			Right hand	
Element no.	Element	Time (TMU)	Time (TMU)	Element
	Take 2 screws			Take the top
1	Reach to a screw	12.9	10.5	Reach hand to top
2	Grasp screw 1	9.1	2.0	Grasp top
3	Grasp screw 2	9.1	5.6	Regrasp (since it is an odd shape)
	Move to base, support right hand			Position top to base
4	Move to base	11.3	16.9	Move top to base (No weight allowance)
5	Contact grasp top	0	2.0	Adjustment to align parts
6			2.0	Release top
	Screws			Screws
7			5.3	Reach to take screw 1
8	Release screw 1	2.0	2.0	Grasp screw 1
9	Move screw 2 to hole 2	5.7	6.7	Move screw 1 to hole 1
10	Move object to exact location.	2.0		
11	Position screw 2	5.6	5.6	Position screw 1
12	Turn screw 2 two turns	2.5	2.5	Turn screw 1 two turns

Repeat the cycle for each pair of the remaining screws.

Fig. 23. MTM details of pump assembly task by human operator.

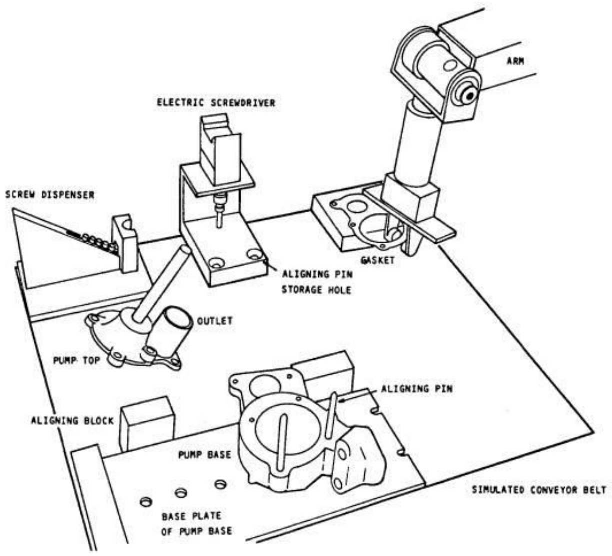
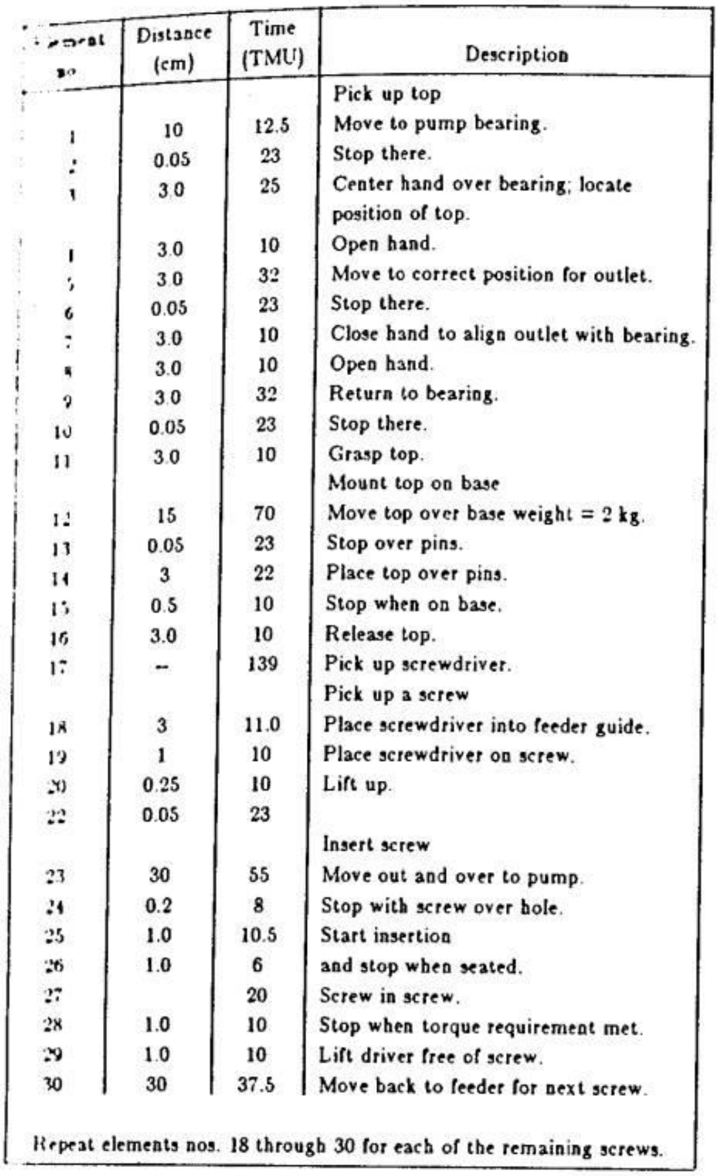


Fig. 24. Workplace for robot operator pump assembly task.



Element no.	Distance (cm)	Time (TMU)	Description
			Pick up top
1	10	12.5	Move to pump bearing.
2	0.05	23	Stop there.
3	3.0	25	Center hand over bearing; locate position of top.
4	3.0	10	Open hand.
5	3.0	32	Move to correct position for outlet.
6	0.05	23	Stop there.
7	3.0	10	Close hand to align outlet with bearing.
8	3.0	10	Open hand.
9	3.0	32	Return to bearing.
10	0.05	23	Stop there.
11	3.0	10	Grasp top.
			Mount top on base
12	15	70	Move top over base weight = 2 kg.
13	0.05	23	Stop over pins.
14	3	22	Place top over pins.
15	0.5	10	Stop when on base.
16	3.0	10	Release top.
17	--	139	Pick up screwdriver.
			Pick up a screw
18	3	11.0	Place screwdriver into feeder guide.
19	1	10	Place screwdriver on screw.
20	0.25	10	Lift up.
22	0.05	23	
			Insert screw
23	30	55	Move out and over to pump.
24	0.2	8	Stop with screw over hole.
25	1.0	10.5	Start insertion
26	1.0	6	and stop when seated.
27		20	Screw in screw.
28	1.0	10	Stop when torque requirement met.
29	1.0	10	Lift driver free of screw.
30	30	37.5	Move back to feeder for next screw.

Repeat elements nos. 18 through 30 for each of the remaining screws.

Fig. 25. RTM for top assembly.

6. Robot Arm Kinematics

- The study of the geometry of motion of robot arms with respect to a fixed reference coordinate system, without regard to the forces/moments that cause the motion
- Direct Kinematics: forward solution; from joint angles to spatial coordinates
- Reverse Kinematics: arm solution, from spatial coordinates to arm angles

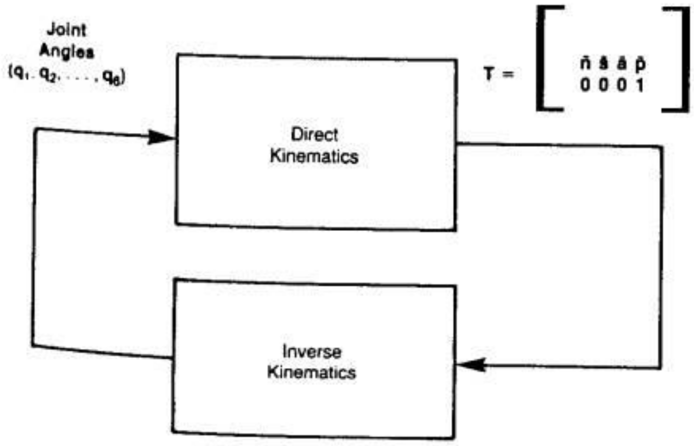


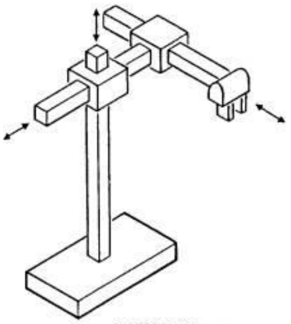
Figure 2.1. The Direct and Inverse Kinematics Problems

Robot Arms Problems: Kinematics, Dynamics, and Control

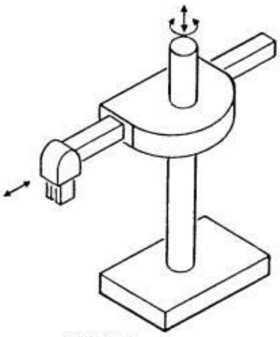
Kinematics - spatial transform between the robot joints and the work space

Dynamics - formulation of robotics arms' equations of motions

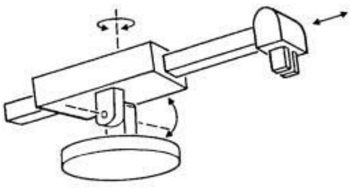
Control - maintain the dynamic response of a robot according to prescribed goals



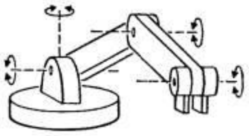
CARTESIAN OR *x-y-z*



CYLINDRICAL



SPHERICAL



REVOLUTE

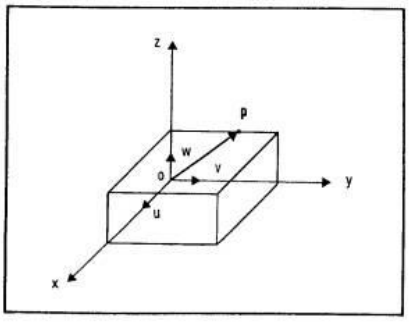


Figure 2. Coordinate systems for a rigid body.

The homogeneous transformation matrix is a 4×4 matrix that maps a position vector expressed in homogeneous coordinates from one coordinate system to another coordinate system. We can consider a homogeneous transformation matrix as comprising four submatrices:

$$\mathbf{T} = \left[\begin{array}{c|c} \mathbf{R}_{3\times 3} & \mathbf{p}_{3\times 1} \\ \text{—} & \text{—} \\ \mathbf{f}_{1\times 3} & 1\times 1 \end{array}\right] = \left[\begin{array}{c|c} \text{Rotation} & \text{Position} \\ \text{Matrix} & \text{Vector} \\ \text{—} & \text{—} \\ \text{Perspective} & \text{Scaling} \\ \text{Transf.} & \text{Factor} \end{array}\right] \quad (14)$$

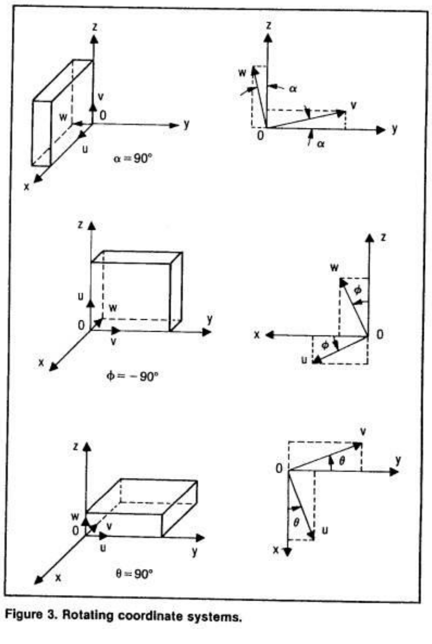
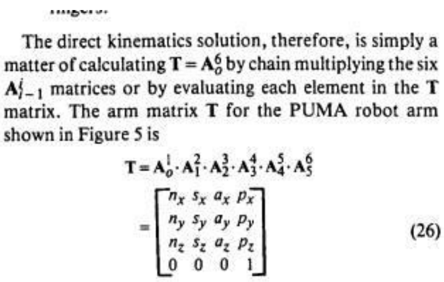


Figure 3. Rotating coordinate systems.

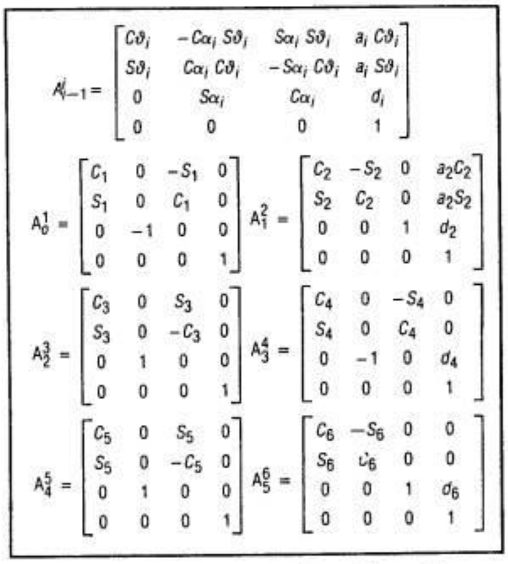
Example of Kinematics Translation – Puma Manipulator Arm



fingers.

The direct kinematics solution, therefore, is simply a matter of calculating $\mathbf{T} = \mathbf{A}_o^6$ by chain multiplying the six $\mathbf{A}_{i-1}^i$ matrices or by evaluating each element in the $\mathbf{T}$ matrix. The arm matrix $\mathbf{T}$ for the PUMA robot arm shown in Figure 5 is

$$\mathbf{T} = \mathbf{A}_o^1 \cdot \mathbf{A}_1^2 \cdot \mathbf{A}_2^3 \cdot \mathbf{A}_3^4 \cdot \mathbf{A}_4^5 \cdot \mathbf{A}_5^6 = \begin{bmatrix} n_x & s_x & a_x & p_x \\ n_y & s_y & a_y & p_y \\ n_z & s_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (26)$$



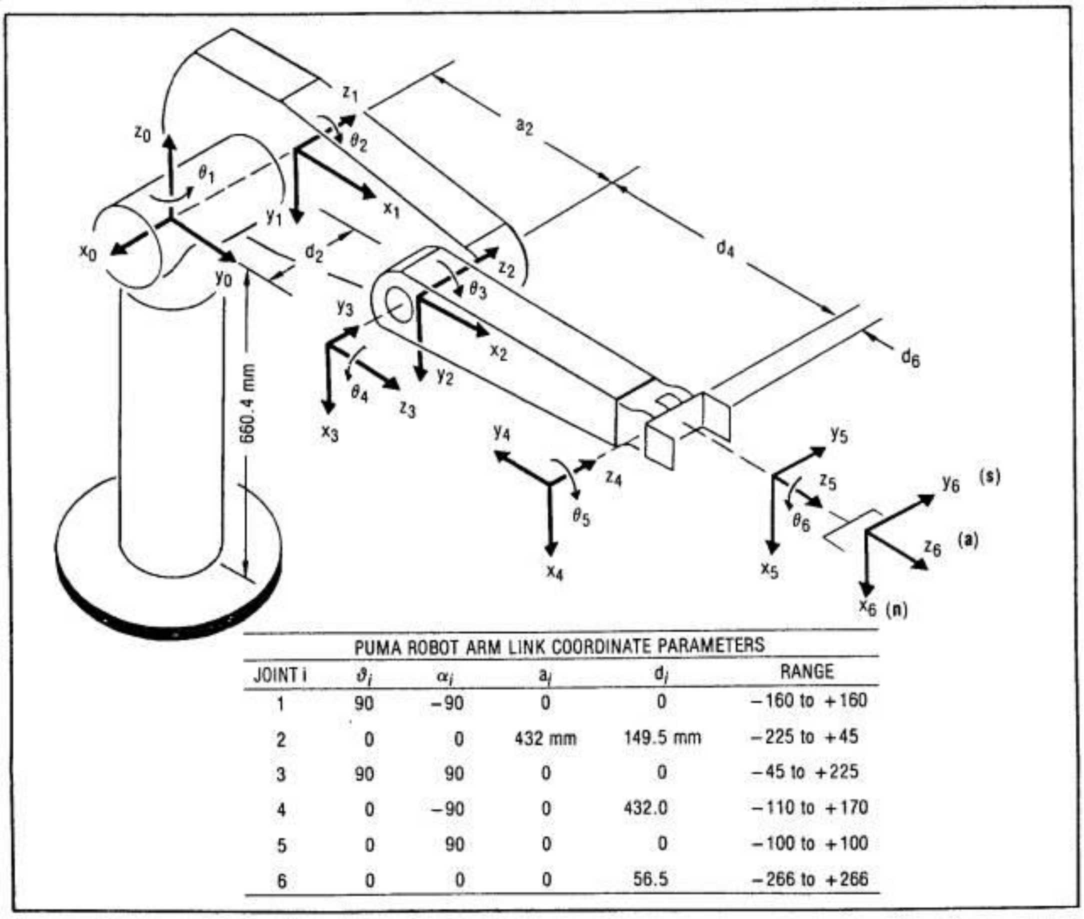
$$A_{i-1}^i = \begin{bmatrix} C\vartheta_i & -C\alpha_i\, S\vartheta_i & S\alpha_i\, S\vartheta_i & a_i\, C\vartheta_i \\ S\vartheta_i & C\alpha_i\, C\vartheta_i & -S\alpha_i\, C\vartheta_i & a_i\, S\vartheta_i \\ 0 & S\alpha_i & C\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_o^1 = \begin{bmatrix} C_1 & 0 & -S_1 & 0 \\ S_1 & 0 & C_1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad A_1^2 = \begin{bmatrix} C_2 & -S_2 & 0 & a_2C_2 \\ S_2 & C_2 & 0 & a_2S_2 \\ 0 & 0 & 1 & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_2^3 = \begin{bmatrix} C_3 & 0 & S_3 & 0 \\ S_3 & 0 & -C_3 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad A_3^4 = \begin{bmatrix} C_4 & 0 & -S_4 & 0 \\ S_4 & 0 & C_4 & 0 \\ 0 & -1 & 0 & d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_4^5 = \begin{bmatrix} C_5 & 0 & S_5 & 0 \\ S_5 & 0 & -C_5 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad A_5^6 = \begin{bmatrix} C_6 & -S_6 & 0 & 0 \\ S_6 & C_6 & 0 & 0 \\ 0 & 0 & 1 & d_6 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Figure 6. PUMA coordinate transformation matrices.



PUMA ROBOT ARM LINK COORDINATE PARAMETERS

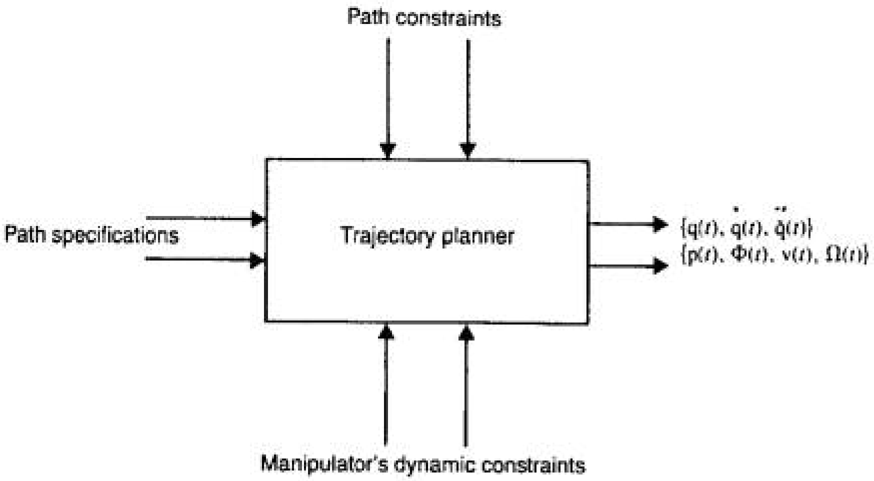
JOINT i	ϑ_i	α_i	a_i	d_i	RANGE
1	90	−90	0	0	−160 to +160
2	0	0	432 mm	149.5 mm	−225 to +45
3	90	90	0	0	−45 to +225
4	0	−90	0	432.0	−110 to +170
5	0	90	0	0	−100 to +100
6	0	0	0	56.5	−266 to +266

Figure 5. Establishing link coordinate systems for a PUMA robot.

7. Robot Trajectory Planning

		Obstacle Constraint	
		Yes	No
Path Constraint	Yes	Off-line, collision-free-path planning plus on-line path tracking	Off-line path planning plus on-line path tracking
	No	Positional control plus on-line obstacle detection and avoidance	Positional control

Table 4.1. Control Modes of Manipulator



(a) When picking up an object, the motion of the hand must be directed away from an object; otherwise the hand may crash into the supporting surface of the object.

(b) If we specify a departure position (lift-off point) along the normal vector to the surface out from the initial position and if we require the hand (i.e., the origin of the hand coordinate frame) to pass through this position, we then have an admissible departure motion. If we further specify the time required to reach this position, we can then control the speed at which the object is to be lifted.

(c) The same set of lift-off requirements for the arm motion is also true for the set-down point of the final position motion (i.e., we must move to a normal point out from the surface and then slow down to the final position) so that the correct approach direction can be obtained.

(d) From the above, we have four positions for each arm motion: initial, lift-off, set-down, and final (see Figure 4.2).

(e) Position constraints:
    (i) Initial position: velocity and acceleration are given (normally zero).
    (ii) Lift-off position: continuous motion for intermediate points.
    (iii) Set-down position: same as lift-off position.
    (iv) Final position: velocity and acceleration are given (normally zero).

(f) In addition to these constraints, the extrema of all the joint trajectories must be within the physical and geometric limits of each joint.

(g) Time considerations:

(i) Initial and final trajectory segments: time is based on the rate of approach of the hand to and from the surface and is some fixed constant based on the joint motor's characteristics.

(ii) Intermediate points or midtrajectory segment: time is based on maximum velocity and acceleration of the joints, and the maximum of these times is used (i.e., the maximum time of the slowest joint is used for normalization).

The constraints of a typical trajectory are listed in Table 4.2. Based on these constraints, we are concerned with selecting a class of polynomial functions of degree n or less such that the required joint position, velocity, and acceleration at these knot points (initial, lift-off, set-down, and final positions) are satisfied, and the joint position, velocity, and acceleration are continuous on the entire time interval $[t_0, t_f]$.

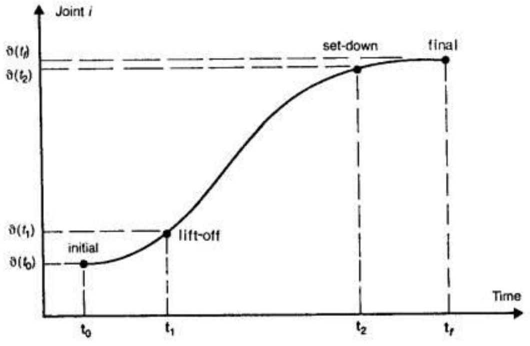


Figure 4.2. Position Conditions for a Joint Trajectory

Table 4.2
Constraints for Planning Joint-Interpolated Trajectory

Initial Position:

(1) position (given)
(2) velocity (given, normally zero)
(3) acceleration (given, normally zero)

Intermediate Positions:

(4) lift-off position (given)
(5) lift-off position (continuous with previous trajectory segment)
(6) velocity (continuous with previous trajectory segment)
(7) acceleration (continuous with previous trajectory segment)
(8) set-down position (given)
(9) set-down position (continuous with next trajectory segment)
(10) velocity (continuous with next trajectory segment)
(11) acceleration (continuous with next trajectory segment)

Final Position:

(12) position (given)
(13) velocity (given, normally zero)
(14) acceleration (given, normally zero)

8. Controller Design for Robots

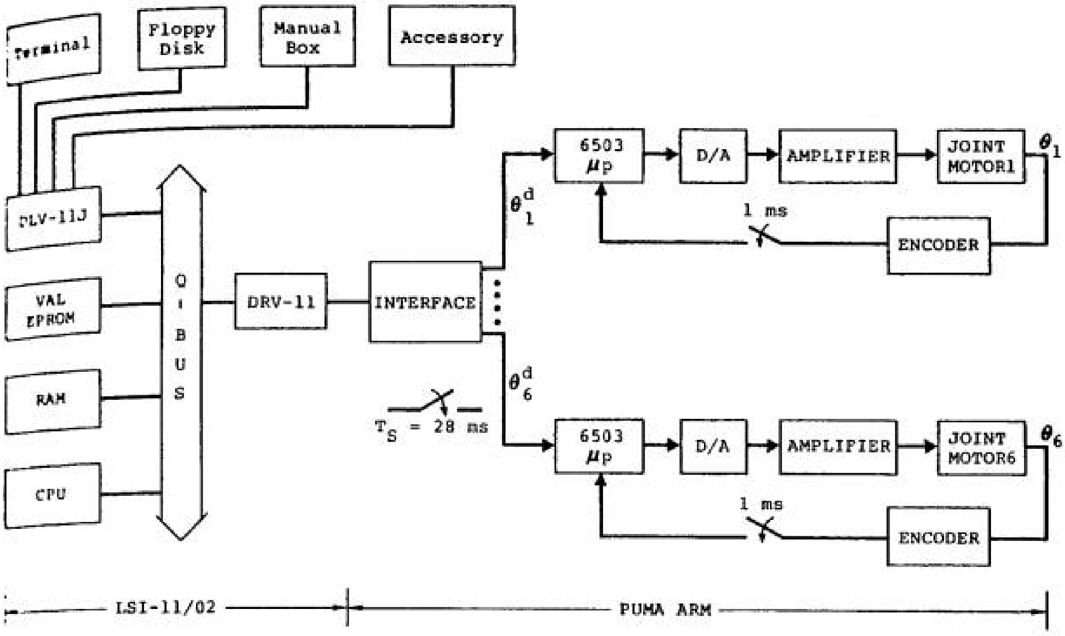
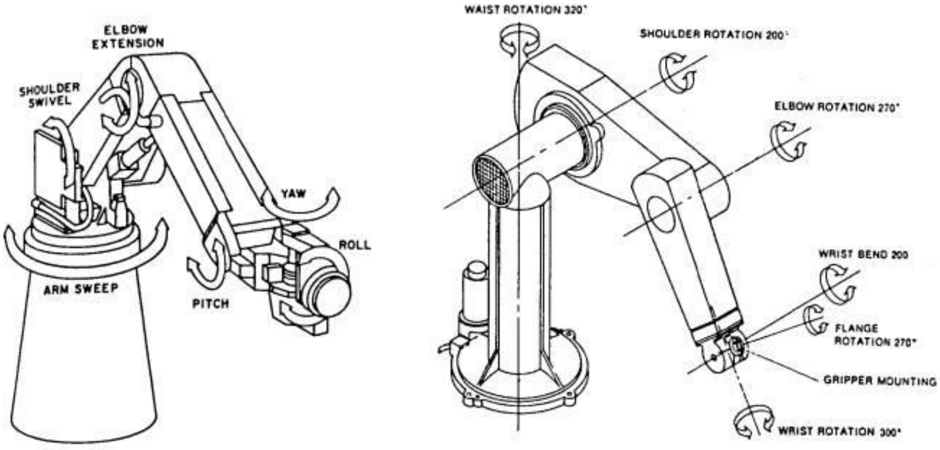


Figure 1. PUMA Robot Arm Control Structure Diagram.



(a) (b)

Fig. 1. Examples of industrial robots. (a) Cincinnati Milacron T3. (b) Unimation PUMA 600.

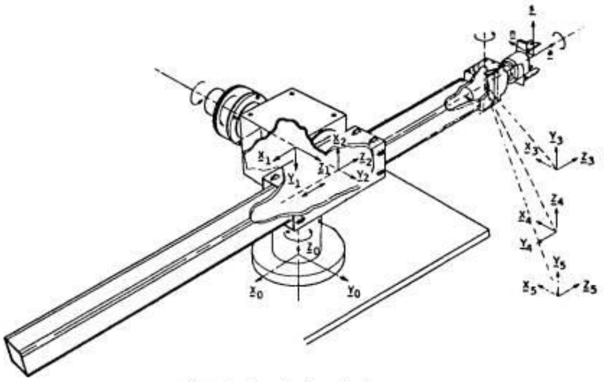


Fig. 2. Stanford manipulator.

Position Control and Hand Orientation

- Fig 3 - a full blown position control system of one axis

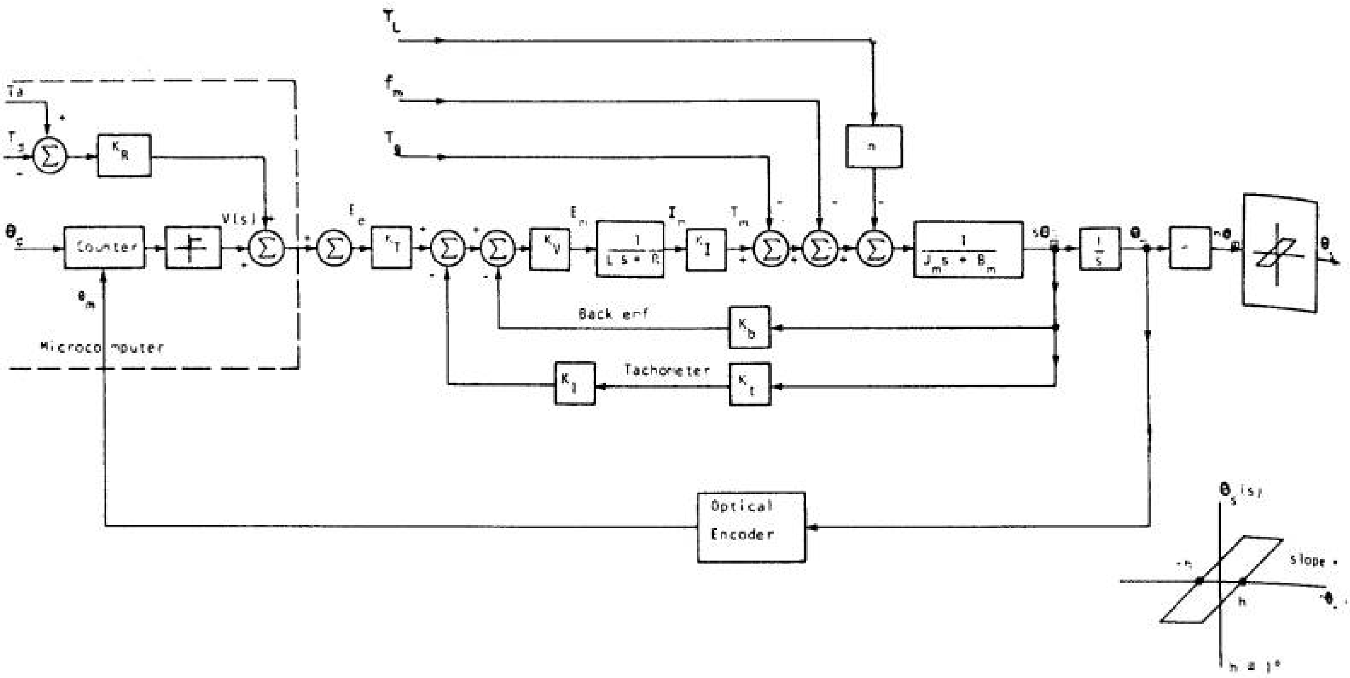
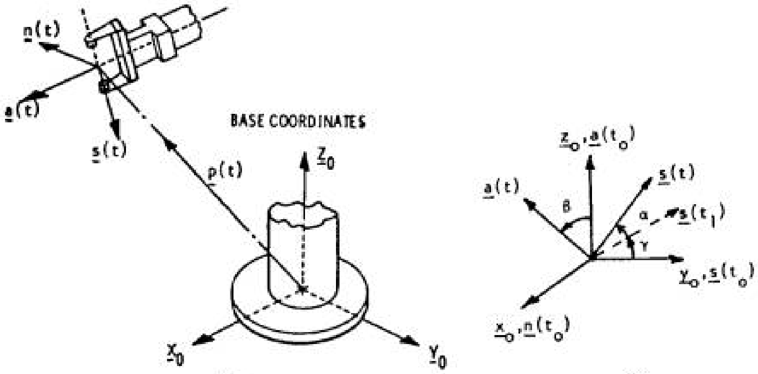


Fig. 3. Positional control system.



(a) (b)

Fig. 4. (a) Position and orientation vectors of hand. (b) Euler angles of orientation.

- Fig 8a - the basic position controller with position feedback
- Fig 8b - the basic position controller with velocity and position feedback
- Fig 8c - the revised and simplified block diagram of 8b, with loading disturbance added

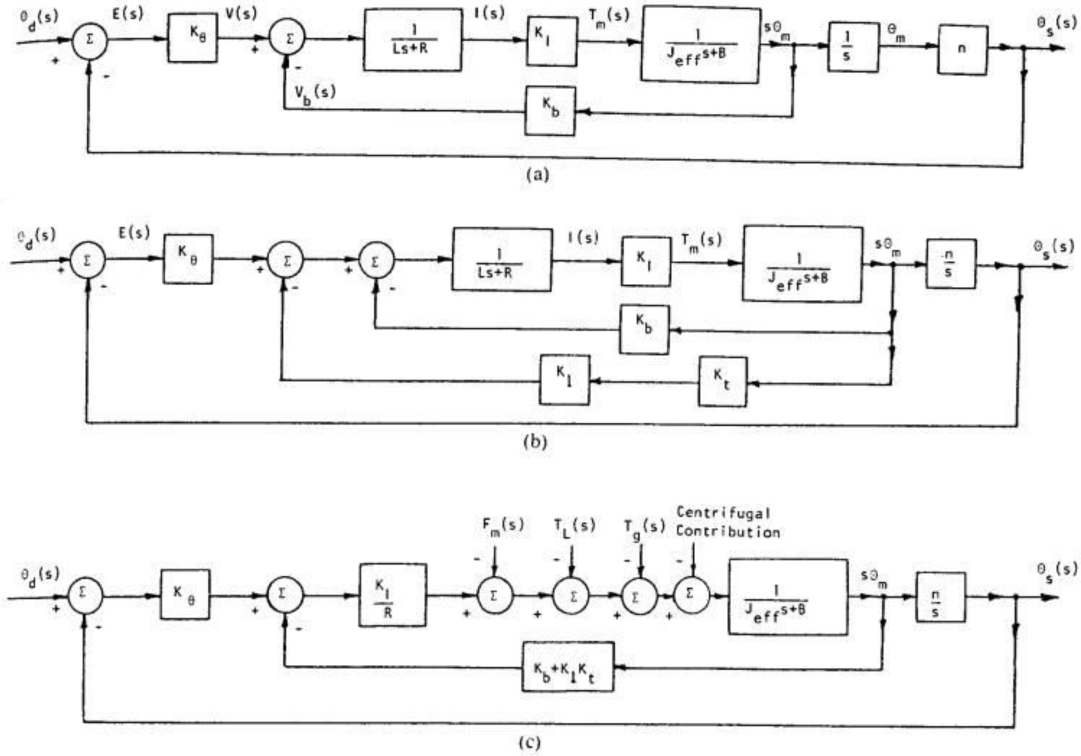


Fig. 8. Positional controller.

Feedforward Compensation

- Fig 10a - anticipated gravitational torque signal Ta +Td is added
- Fig 10b - to reduce steady state velocity error, additional velocity feedforward Vd is added during ramping
- Fig 10c - obtaining Vd from the input signal Xs
- Fig 10d - schematic arrangement of feedforward compensation for the centrifugal force
- Fig 10e - acceleration error feedforward compensation
- Fig 10f - obtaining accelaretion from the command signal

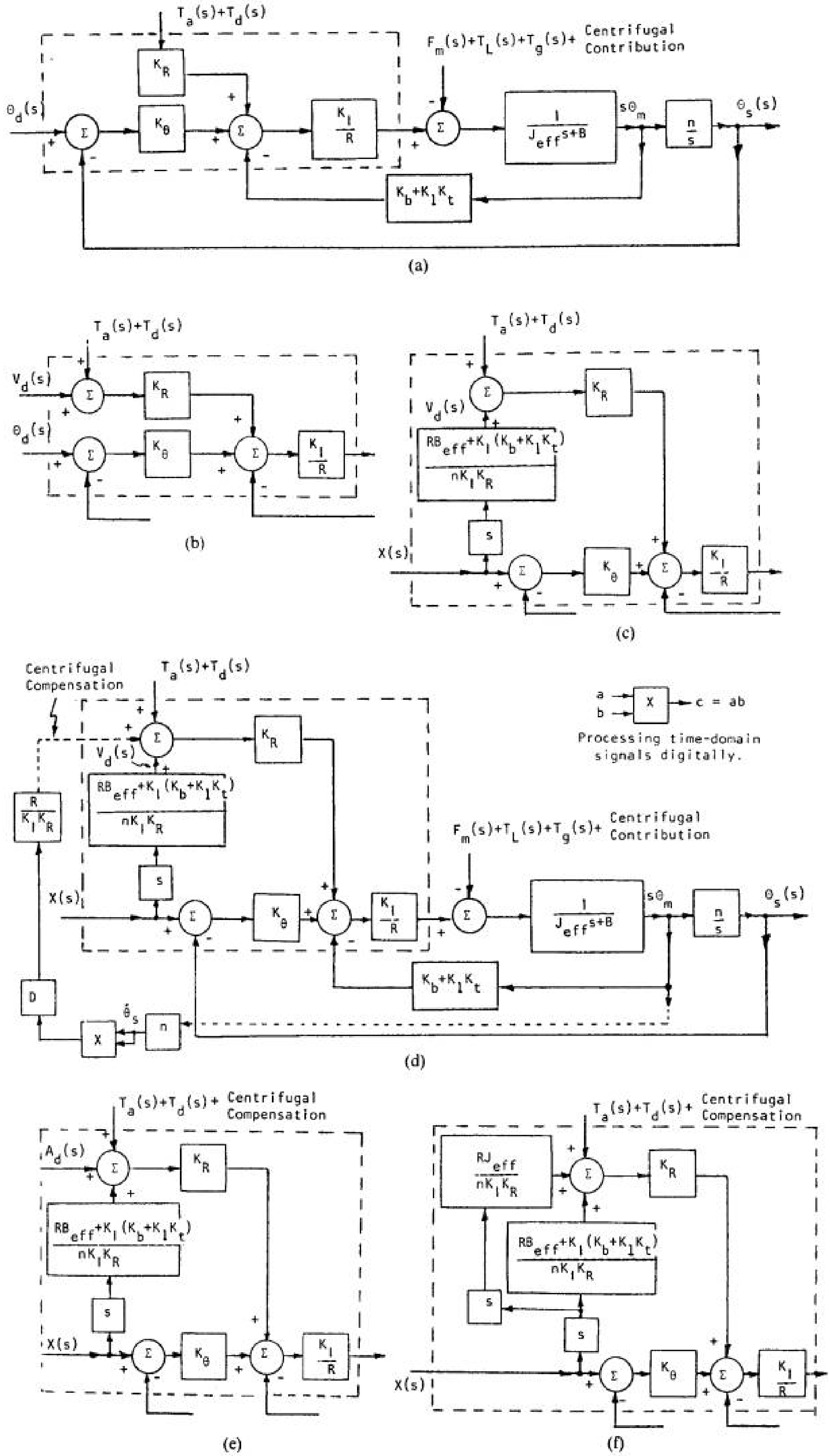


Fig. 10. Controller with anticipated burden and feedforward compensation.

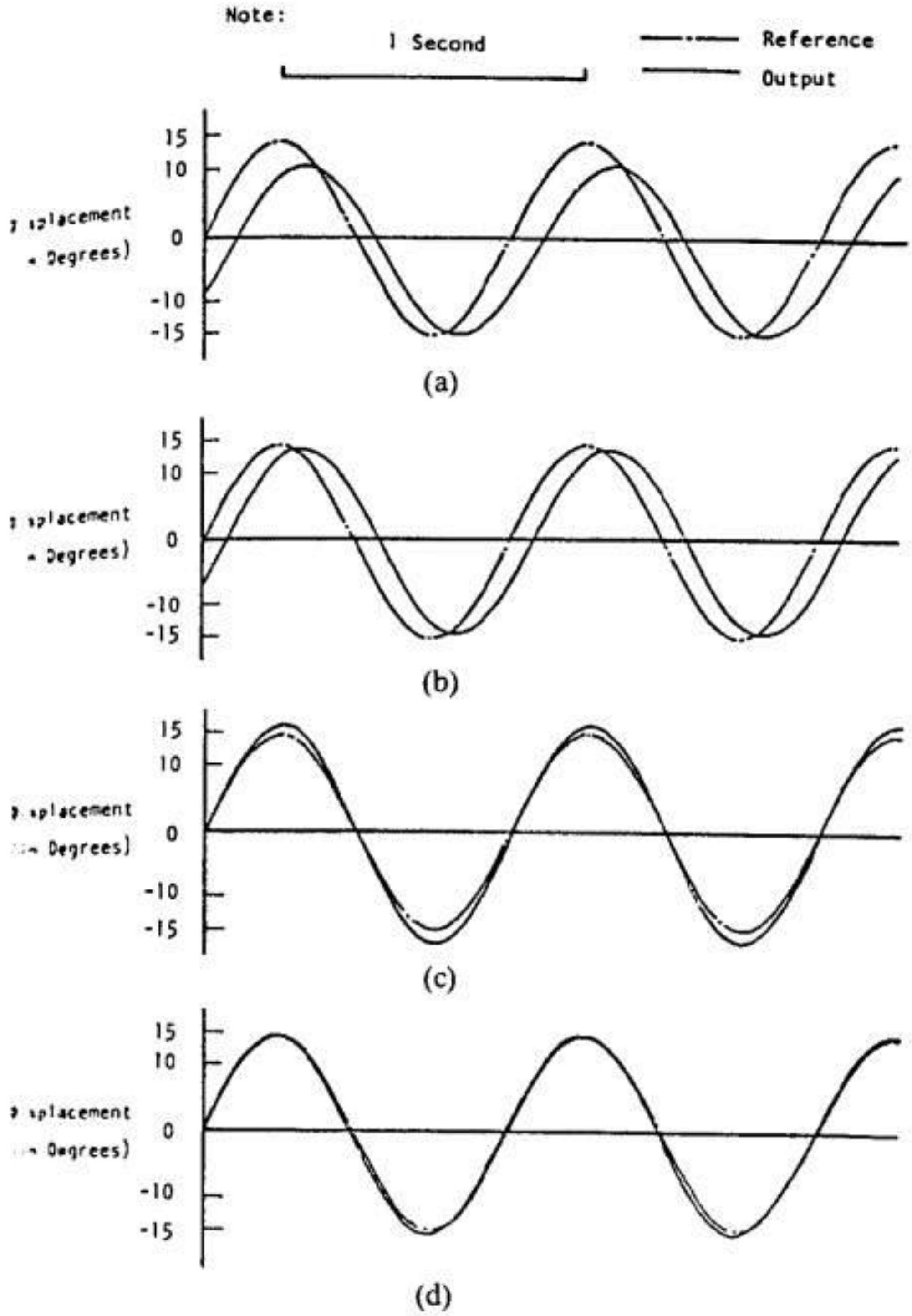


Fig. 11. Effect of feedforward compensation, case of single joint (Joint 4) (a) Without compensation. (b) With compensation for gravity. (c) With compensation for gravity and friction. (d) With compensation for gravity, friction, and inertia.

Coupling between joints and compensation

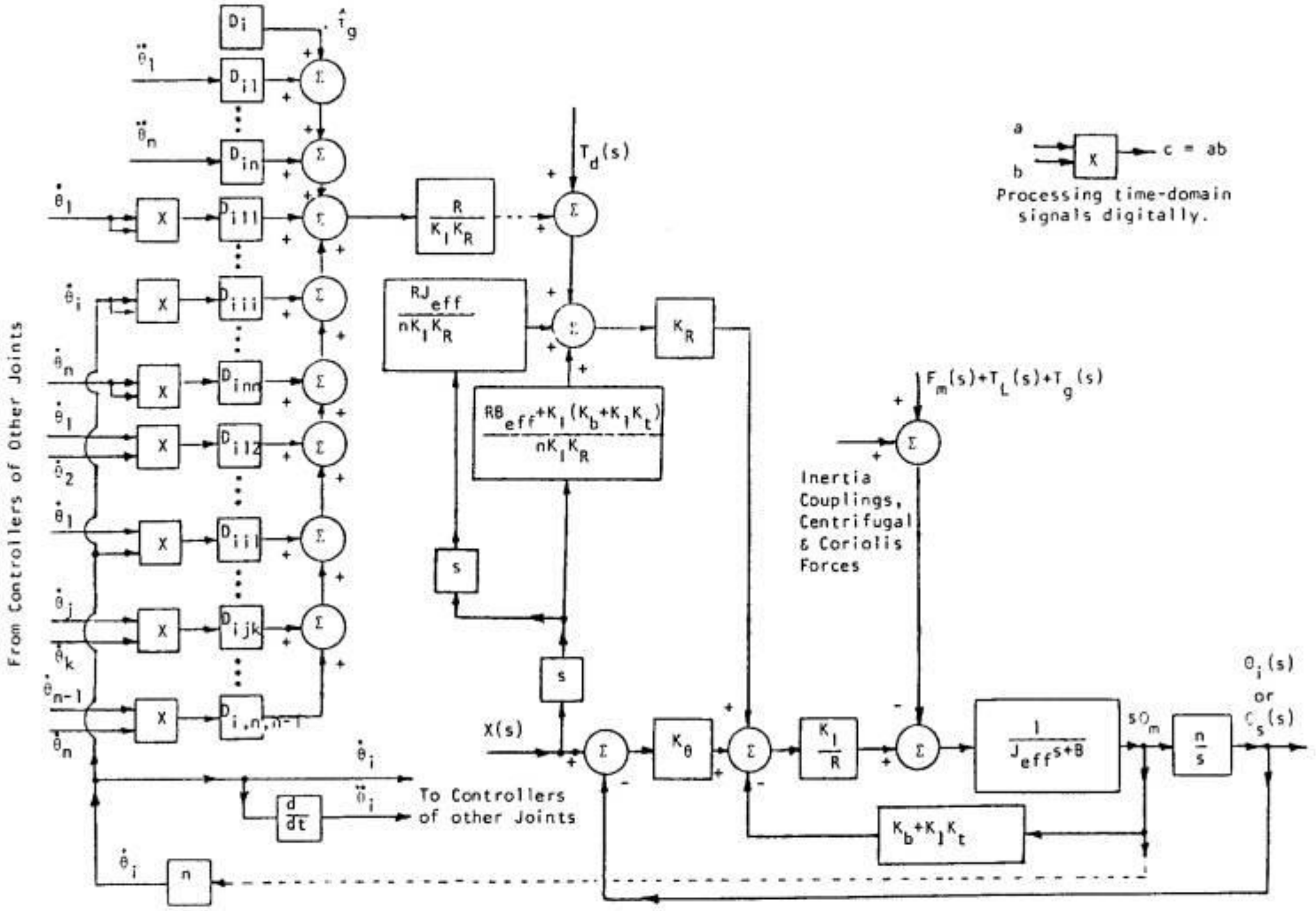


Fig. 12. Complete controller for Joint *i* of robot having *n* joints.

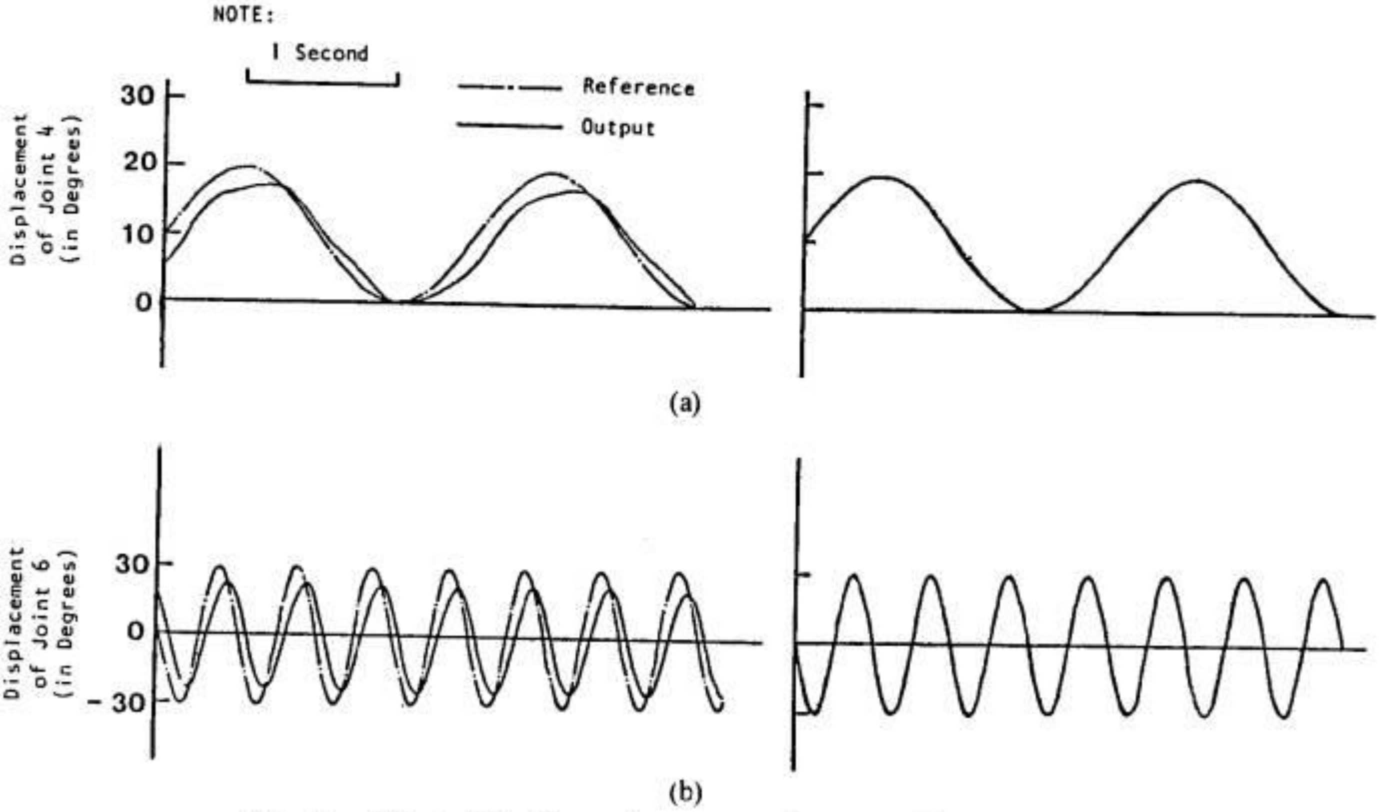


Fig. 16. Effect of feedforward compensation, case of multiple joints with interaction. (a) Without compensation. (b) With compensation.

--- END ---