NOTICE

The quality of this microform is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Reproduction in full or in part of this microform is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30, and subsequent amendments.

AVIS

La qualité de cette microforme dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

La reproduction, même partielle, de cette microforme est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30, et ses amendements subséquents.

Artificial Impedance Approach to the Collision Avoidance of Robots with Obstacles.

By

Rahim Jassemi-Zargani

A thesis presented to the
University of Ottawa
in partial of fulfilment of the
requirements for the degree of

Master of Applied Science

in

Mechanical Engineering

Department of Mechanical Engineering
University of Ottawa

NOTICE

The quality of this microform is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

AVIS

La qualité de cette microforme dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

ISBN 0-315-60611-8

UNIVERSITÉ D'OTTAWA
UNIVERSITY OF OTTAWA

TO MY WIFE AND DAUGHTER

SHIRLEY & SARA

ABSTRACT

Among the desired features for robot manipulation in future applications are autonomous trajectory generation and automatic collision avoidance.

In this thesis, the Impedance control method is used for the trajectory generation of a revolute manipulator in free motion, and for obstacle avoidance.

In the Impedance control method, a desired mechanical impedance between the robot end-effector and the target is generated by modulating the force or torque output of the robot actuators.

An approach using Impedance control, in conjunction with 'Coastal and Surface Navigation' concept, is developed for generating trajectories in non-convex spaces utilizing proximity-sensor data.

ACKNOWLEDGEMENT

I wish to acknowledge my appreciation to all those who helped me in this research. I am particularly grateful to Dr. Necsulescu whose participation in the present research was both valuable and perceptive. He showed that he was very dependable, thorough and responsive. I am also grateful to Mr. Bill Graham, Space Mechanics Directorate, Canadian Space Agency, for his support and access to their facilities. I want to thank also Gorge Dinardo, software specialist, Canadian Space Agency, for his advices in using the computer facilities.

I am also very grateful to my wife Shirley and my daughter Sara for their continuing support throughout my education.

CONTENTS

List of Figures

NOMENCLATURE

B	virtual damper constant
D	desired position of end-effector
D_o	starting point of moving target
D_i	interception point of moving target
E	equilibrium point
f	force-torque vector acting on the end-effector
F_C	coastal navigation force
F_o	repulsive force
F_{o_x}	repulsive force, x-axis component
F_{o_y}	repulsive force, y-axis component
F_S	surface navigation force
$F(\theta,\dot{\theta})$	friction term in joint space
$F_X(\theta,\dot{\theta})$	friction term in cartesian space
F_{x_D}	attractive force
$G(\theta)$	gravitation term in joint space
$G_X(\theta)$	gravitation term in Cartesian space
I	identity matrix
$J(\theta)$	Jacobian matrix
$J^T(\theta)$	transposed Jacobian matrix
$\dot{J}$	derivative of the Jacobian matrix
K_C	coastal navigation force constant
K_o	repulsive force constant
K_Q	joint limits virtual damping constant

K_S	surface navigation constant
L_1	length of the link number one(1)
L_2	length of the link number two(2)
L_3	length of the link number three(3)
m_1	mass of the link number one
m_2	mass of the link number two
m_3	mass of the link number three
$M(\theta)$	mass(inertia) matrix in joint space
$M_X(\theta)$	mass(inertia) matrix in Cartesian space
θ	joint angular position
$\dot{\theta}$	joint angular velocity
R	the radius of effectiveness of repulsive force
$V(\theta,\dot{\theta})$	centrifugal and Coriolis forces term in joint space
$V_X(\theta,\dot{\theta})$	centrifugal and Coriolis forces term in Cartesian space
ω_n	natural frequency
$(X,Y,Z)_m$ OR X_m,Y_m,Z_m	manipulator end-effector current position
$(X,Y,Z)_d$ OR X_d,Y_d,Z_d	manipulator end-effector desired position
$(X,Y,Z)_o$ OR X_o,Y_o,Z_o	position of closest point of the obstacle to the end-effector

1.0 INTRODUCTION

One of the important topics in robotics research today is the generation of trajectories and avoidance of obstacles.

Most of the commercial robots today are controlled by PID position controllers in joint space. Approaches for controlling robots in Cartesian space are still in research stage.

A major problem in coordinating robot motion is the avoidance of obstacles which can enter in the robot's workvolume. One solution for solving this problem is to equip the robot with complex vision systems, in order to find the full geometric description of obstacles and apply a collision avoidance algorithm. In this case, the robot uses off-line trajectory planning in order to be able to avoid obstacles.

Recently, new control techniques were developed for robots in order to achieve autonomous robot controllers.

Impedance control is one of the new control techniques

which can generate trajectories and correct trajectories in real time, in order to avoid any obstacle in the work volume without knowing the full geometry of obstacles. In this case, there is no need for trajectory pre-planning. Also, being a Cartesian space control,inverse kinematics computations are not needed.

This thesis covers impedance control systems for the case of moving targets, avoidance of obstacles and coastal or surface navigation schemes.

The simulations are done for two and three Degree Of Freedom(DOF) robots.

2.0 LITERATURE SURVEY

Most of the research in creating safe working environment for robots and operators has the goal of stopping the robot from imminent collision with any obstacles or humans. Which would result in damaging the robot itself, the objects in the workvolume or would injure humans.

Most of the effort so far was to use complex vision systems to detect full geometry of any obstacle in workvolume and combine them with conventional PID control of the robot system [29]. In order to avoid obstacles, a robot equipped with PID position controllers needs full geometry information of any obstacles in its workvolume, obtained from a complex vision system.

"Artificial potential field" and "artificial impedance" are two of the most recent approches, developed in the last decade. These control approches can create trajectories and avoid any obstacle in the workvolume. Artificial impedance approach is used in this thesis.

Khatib is the first to develop an Artificial Potential Field (APF) approach, which later proved very similar to impedance control systems developed independently. In the APF approach, attractive and repulsive potential fields are created in order to generate robot trajectory. Khatib introduces no limits for the resulting repulsive forces, assumed non-linear functions of distance (figure 2.1). Khatib describes the APF as: "The manipulator moves in a field of forces. The position to be reached is an attractive pole for the end-effector, and obstacles are repulsive surfaces for the manipulator parts". [18]

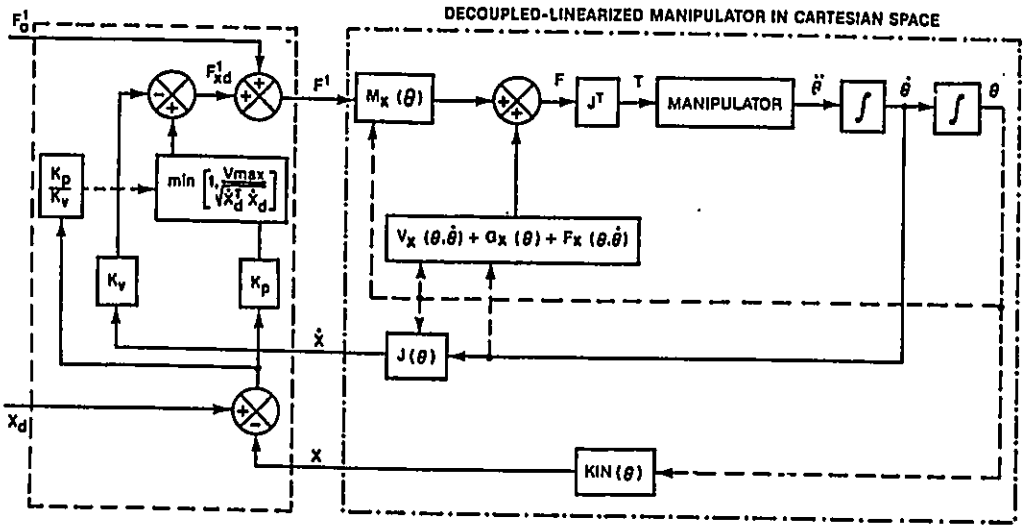


Figure 2.1 Khatib's control scheme for collision avoidance using APF

Hogan is one of the first contributors in the development of impedance control approach (figure 2.2) and he developed control schemes both for free and for contact motion. Free motion control refers to motion in which no physical contact with the environment occurs. In this case, the controller generates an artificial mechanical impedance between end-effector and target. In contact motion, the control system uses an extra feedback from the contact force between the end-effector and the environment. [10-13]

In developing the control system, Hogan introduces two force fields, an attractive force field, which is created between the end-effector and the target, and a repulsive force field, which is created around the obstacles contained in the workvolume. This control scheme operates in Cartesian space and all dynamic terms of the robot model have to be decoupled for the Cartesian space(inertial, centrifugal and Coriolis, gravitational and frictional terms).

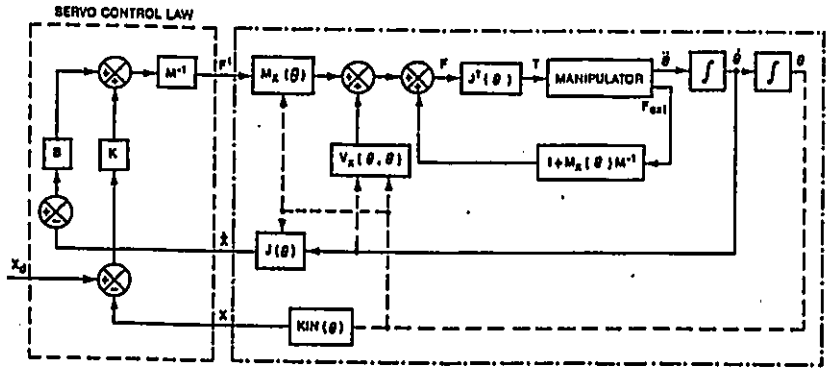


Figure 2.2 Hogan's impedance control scheme with force measurement

Kazerooni introduces an impedance controller (figure 2.3) for robots equipped with position controllers. Kazerooni's impedance control system controls the robot in joint space, therefore decoupling the joint space motion. Also, Kazerooni introduces a negative force feedback from the interaction between the end-effector and the environment. The resulting compensator, H, modulates the contact generized force f_{ext} as desired. [14-17]

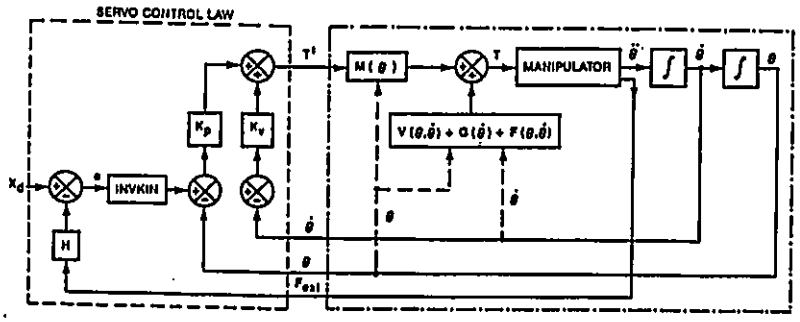


Figure 2.3 Kazerooni's impedance control scheme applied to a joint space controlled manipulator

The impedance control scheme used in this thesis is[26]:

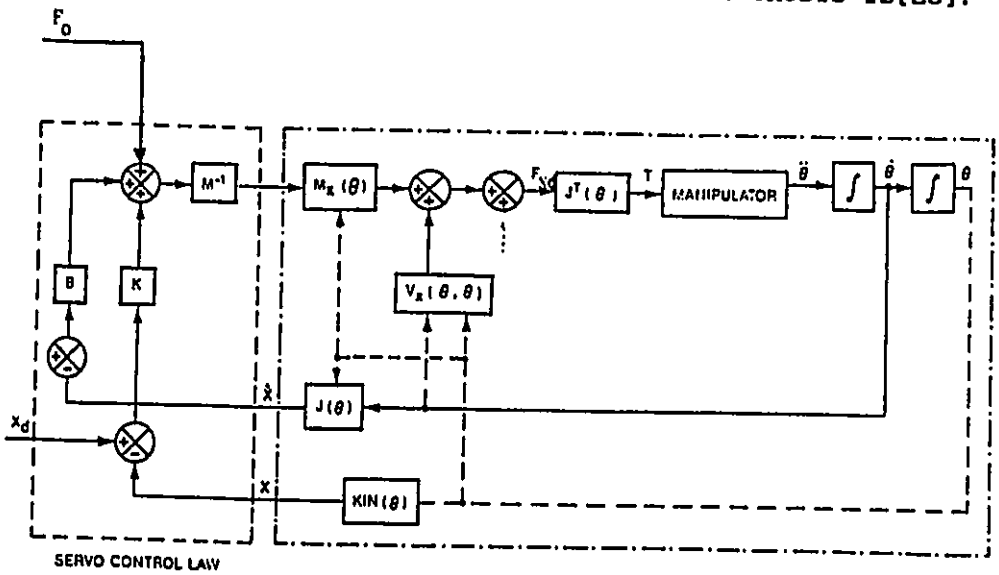


Figure 2.4 Impedance control scheme used for simulations

3.0 IMPEDANCE CONTROL OF ROBOT MOTION

3.1 INTRODUCTORY REMARKS

The concept of Impedance Control

Impedance control is based on creating an artificial mechanical impedance between the robot end-effector and the target point. This impedance is intended to direct the end-effector to the target point. In order to do this, we consider that the mechanical system has an artificial link, formed by a spring and a damper, between the robot end-effector and the target. This mechanical system has the spring component defined at rest in the target point and will produce an attractive force in Cartesian space in the direction of a straight line from the end-effector to the target. The Cartesian force (f), applied to the end-effector, can be generated by using its dependence on the joint torque vector(τ):

$$\tau = J^T f \qquad (3.1.1)$$

To direct the robot to the target on a straight line path, the robot dynamics have to be decoupled in Cartesian space, by compensating the inertial,centrifugal, Coriolis, gravity and frictional terms:

$$\tau=J^T \left[M_X(\theta) \ddot{X} + V_X(\theta,\dot{\theta}) + G_X(\theta) + F_X(\theta,\dot{\theta}) \right] \quad (3.1.2)$$

Complete linear dynamics equations are given in Appendix A. In this thesis the simulations were performed for two Degree of Freedom(DOF) planer robot and for a 3 DOF revolute robot. The parameters used in the simulation of these robots motion are given in Appendix B.

Illustrative Simulations of the Application of Impedance Control to two simple robots.

Figure 3.1.1 shows the simulation results of the trajectory of 2 DOF planar robot from its initial position(I) to the target position(D). As the results show, the Impedance Controller generates a straight line trajectory. Figure 3.1.2 shows the time behavior of end-effector position (a), joints positions (b) and actuator torques (c) for the case shown in figure 3.1.1 . The end-effector of the robot reached the target within 3.5 seconds. There was no oscillations in the end-effector and joints positions or in torques.

Figure 3.1.3 shows examples of simulation results for a 3 DOF robot equipped with an Impedance Controller. The

results show again a straight line three dimensional space trajectory (marked with larger circles), and its projections (marked with smaller circles).

Figure 3.1.4 shows the time behavior of end-effector positions (a), joint positions (b) and actuator torques (c) for the case shown in figure 3.1.3 . Robot reached the target within 3.5 seconds and there was no sign of undesired behavior in end-effector and joint positions, or in the torques of the robot joints.

Figure 3.1.5 shows a trajectory of a 2 DOF planar robot's end-effector which is controlled by a PD controller in joint space. The PD controller could not produce a straight line trajectory as the impedance controller did in figure 3.1.1, because each of the robot's joints tried to reach it's desired position independently.

Figure 3.1.6 shows the time behavior of end-effector positions (a), joint position (b) and actuator torques (c) for the case shown in figure 3.1.3. The robot reached the target
within 3.5 seconds with torque in the joints slightly higher than in the impedance controller case.

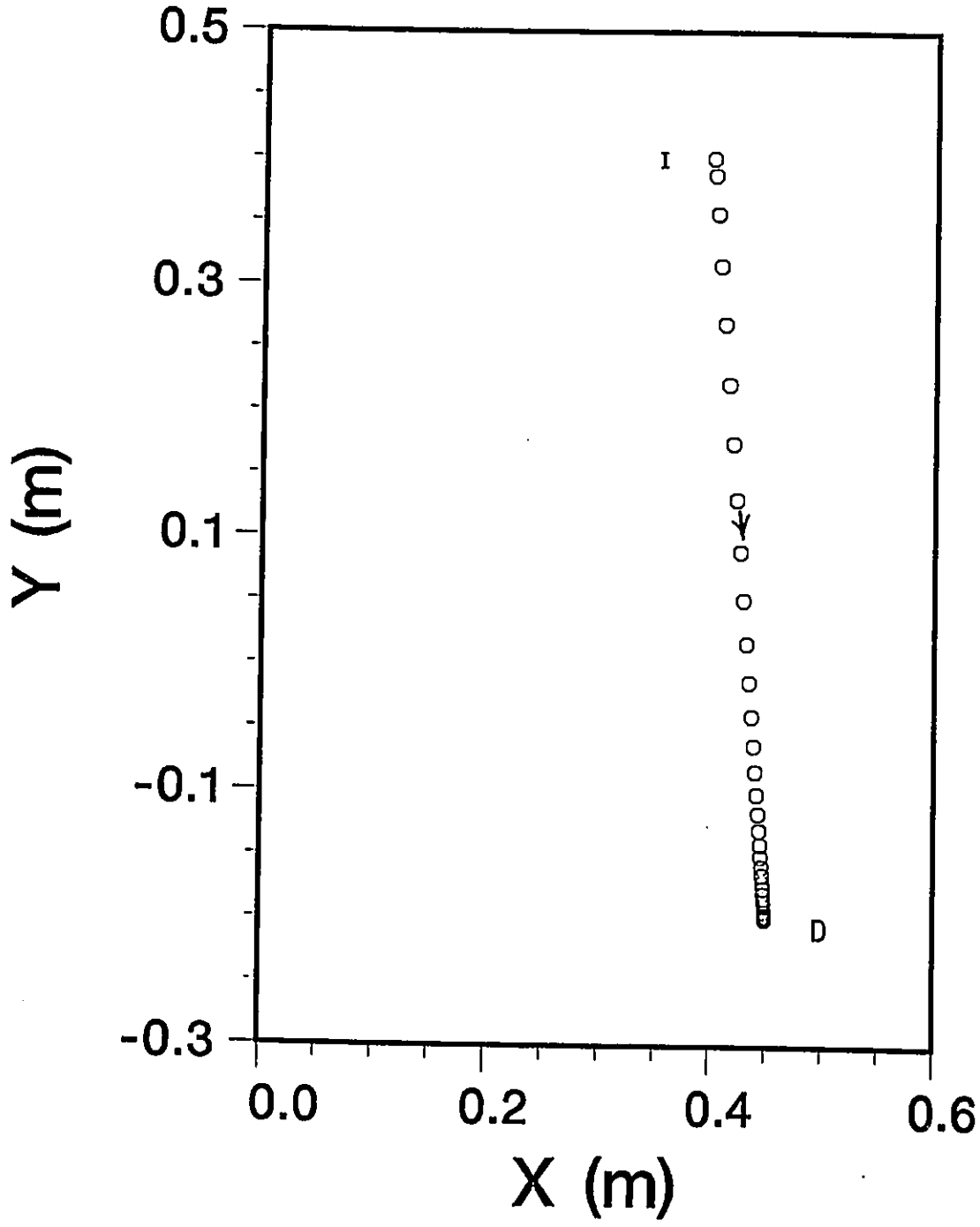


Figure 3.1.1 End-effector trajectory for a 2 DOF robot (K=4 N/m, B=4 N/(m/s), K_c=0.0, K_Q=0.0)

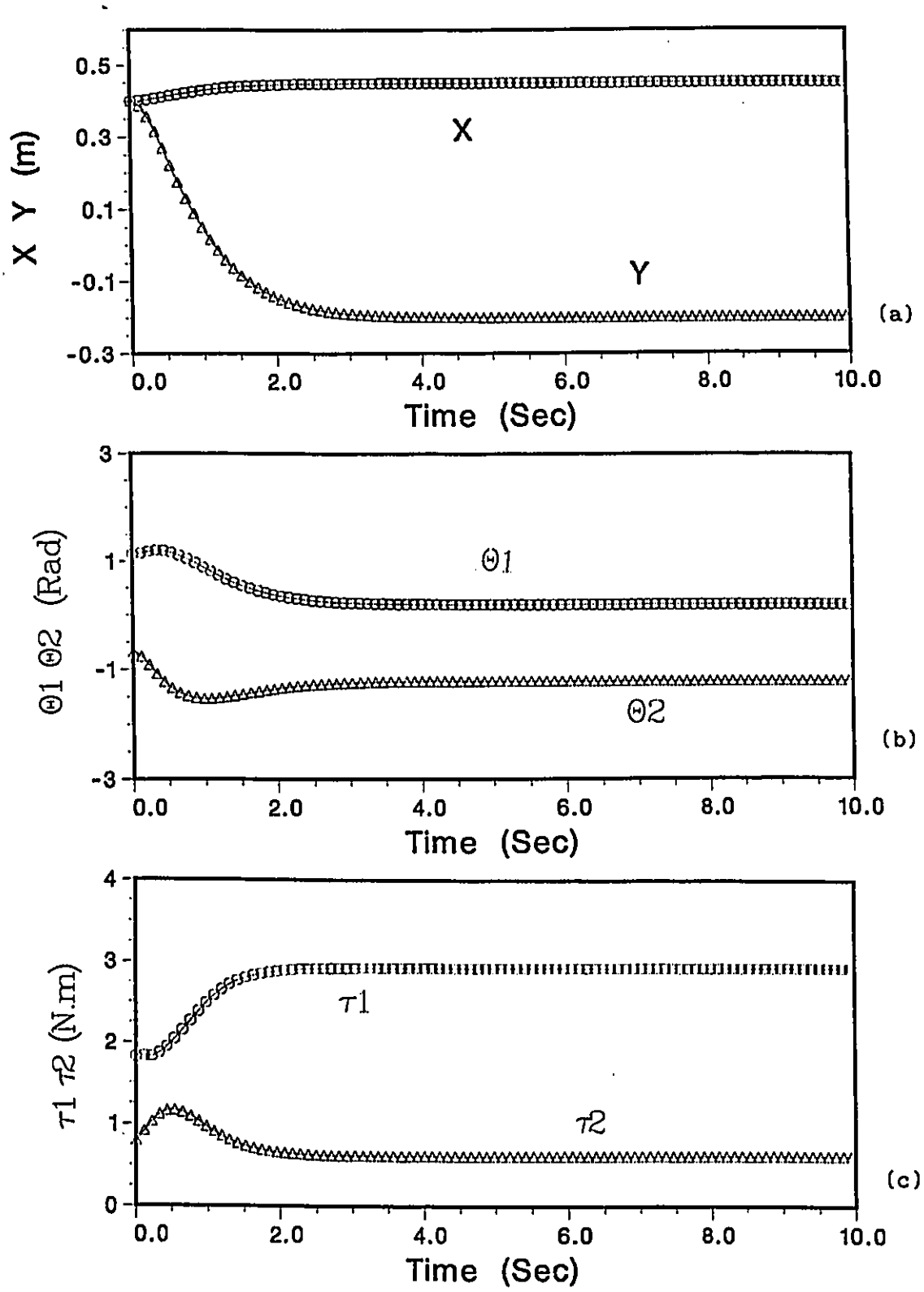


Figure 3.1.2 Simulation results of the robot shown in figure 3.1.1
(a) End-effector position
(b) Joints angular positions
(c) Joint actuator torque

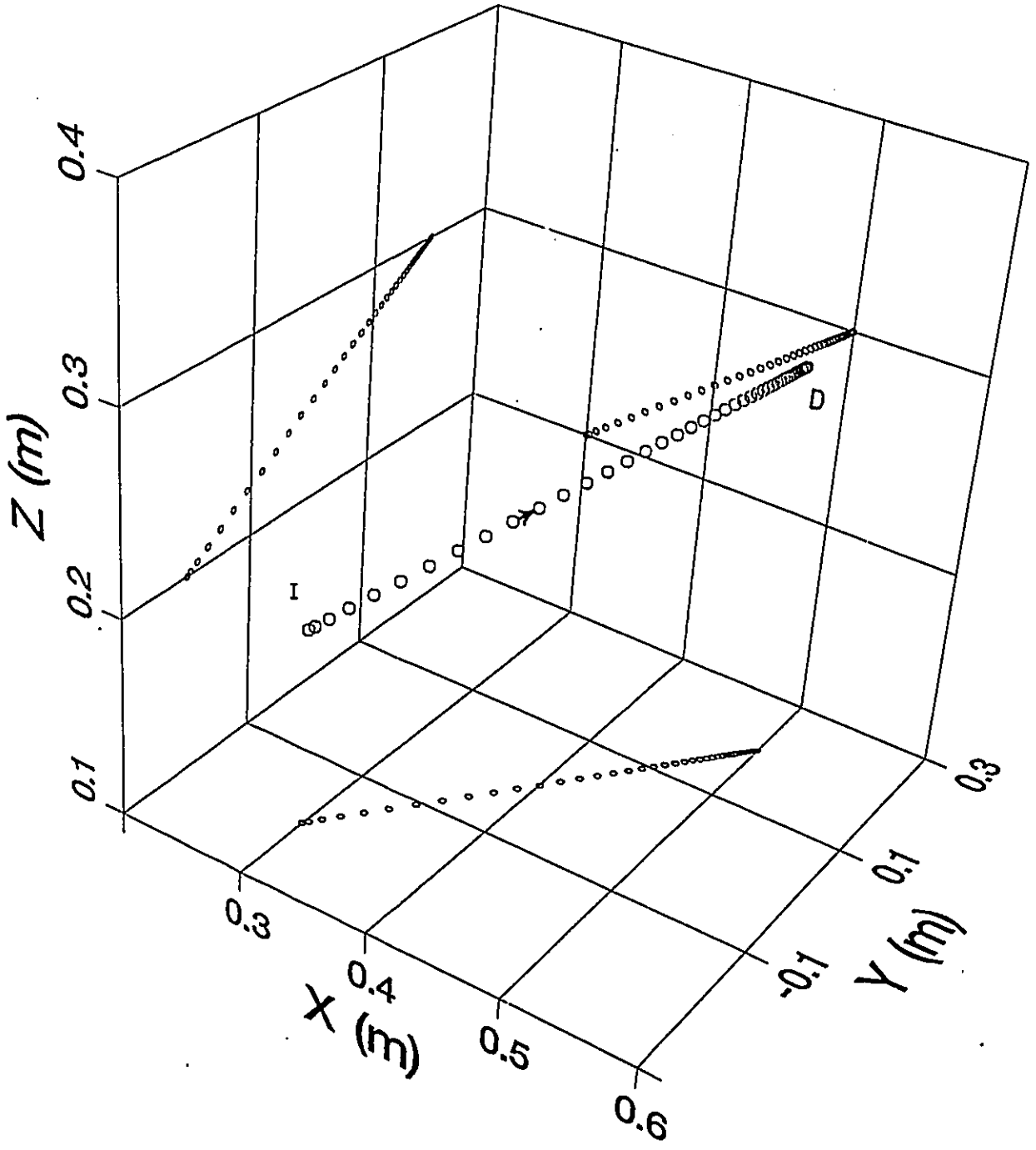


Figure 3.1.3 End-effector trajectory for a 3 DOF robot. (K=4 N/m, B=4 N/(m/s), W=0.0, K_Q=10.0 N/(rad/s))

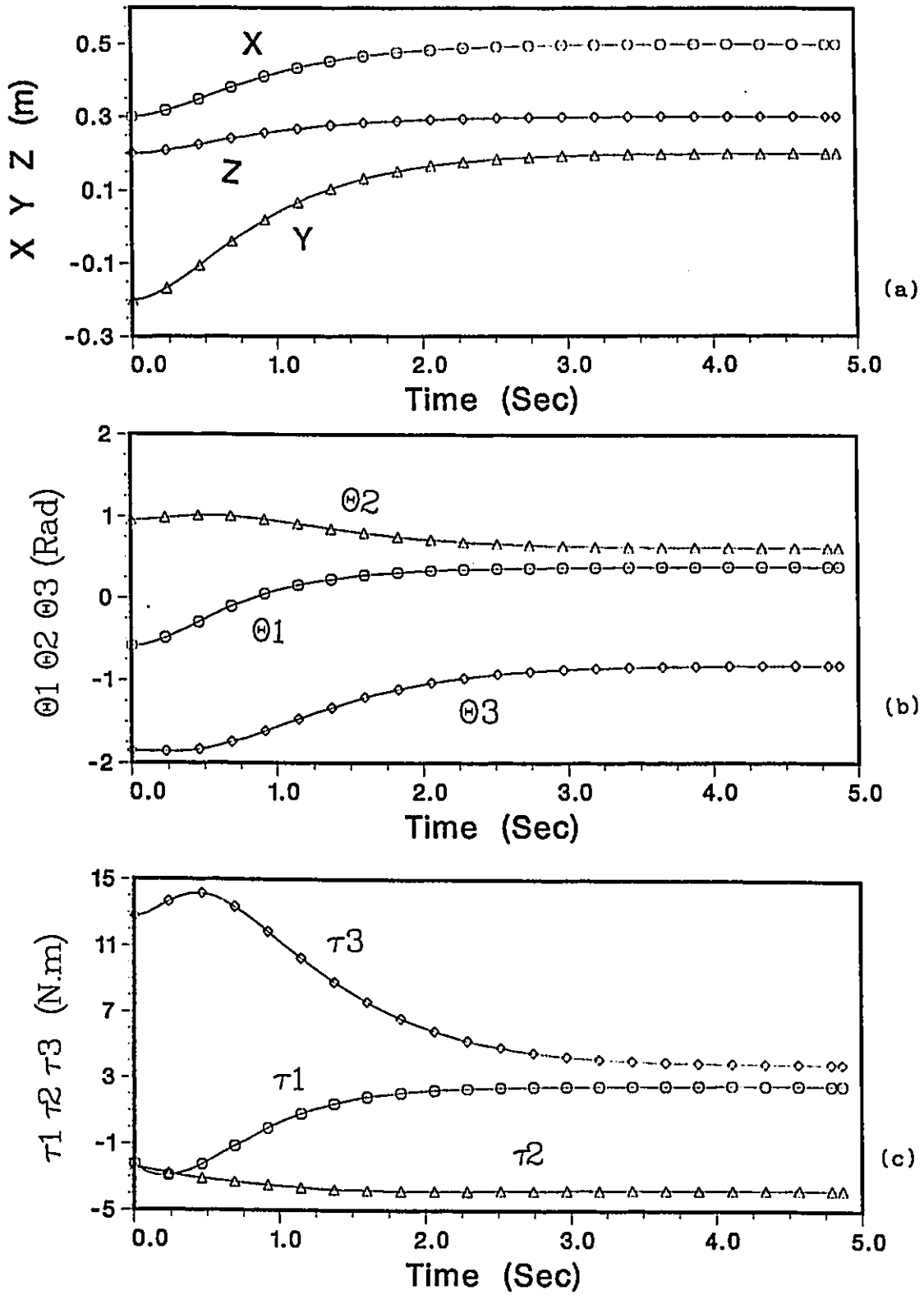


Figure 3.1.4 Simulation results of the robot in figure 3.1.3
(a) End-effector position
(b) Joints angular positions
(c) Actuator torques

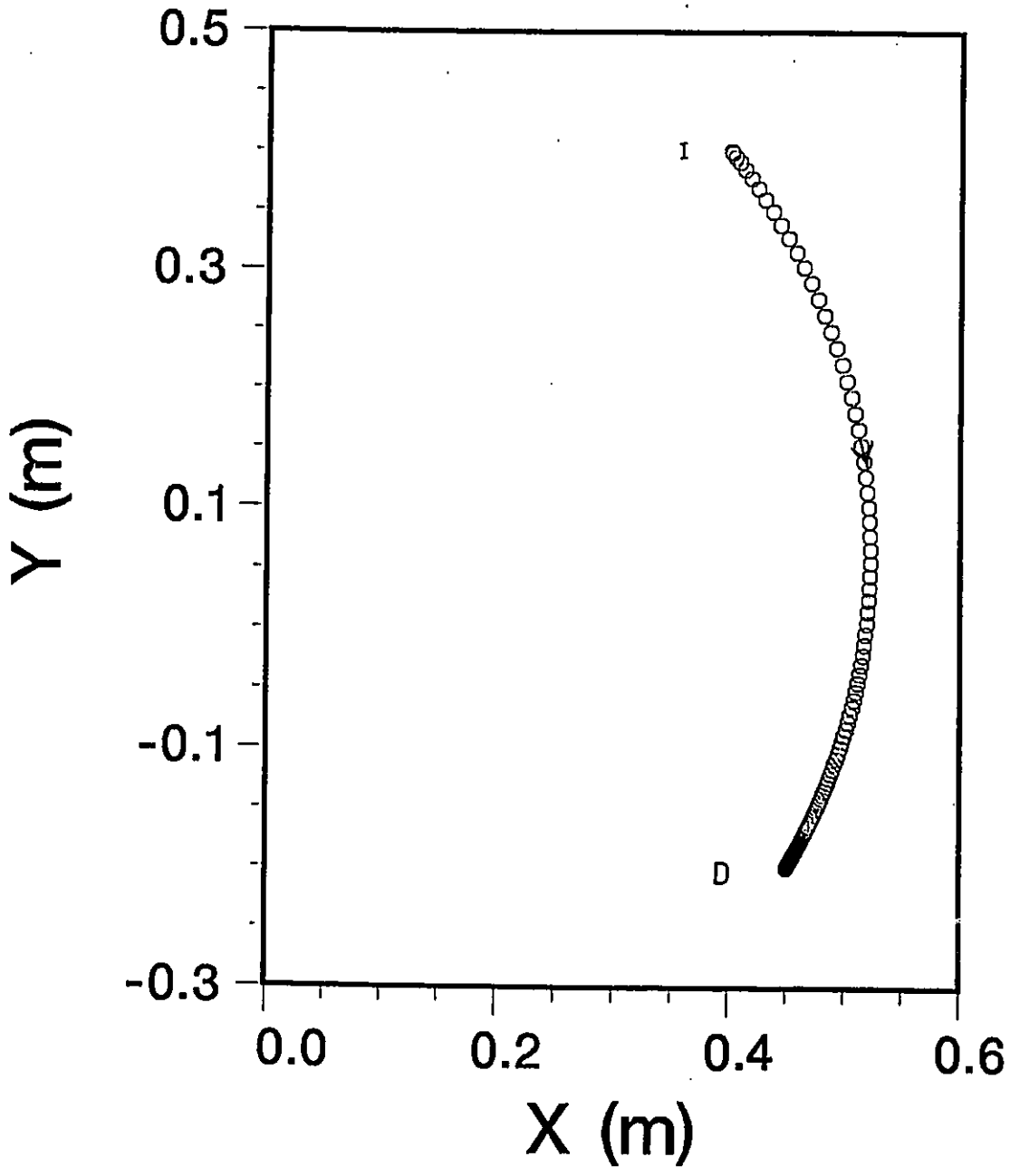


Figure 3.1.5 End-effector trajectory for 2 DOF planar robot with PD controller (K_P=4 N/m, K_V=4 N/(m/s))

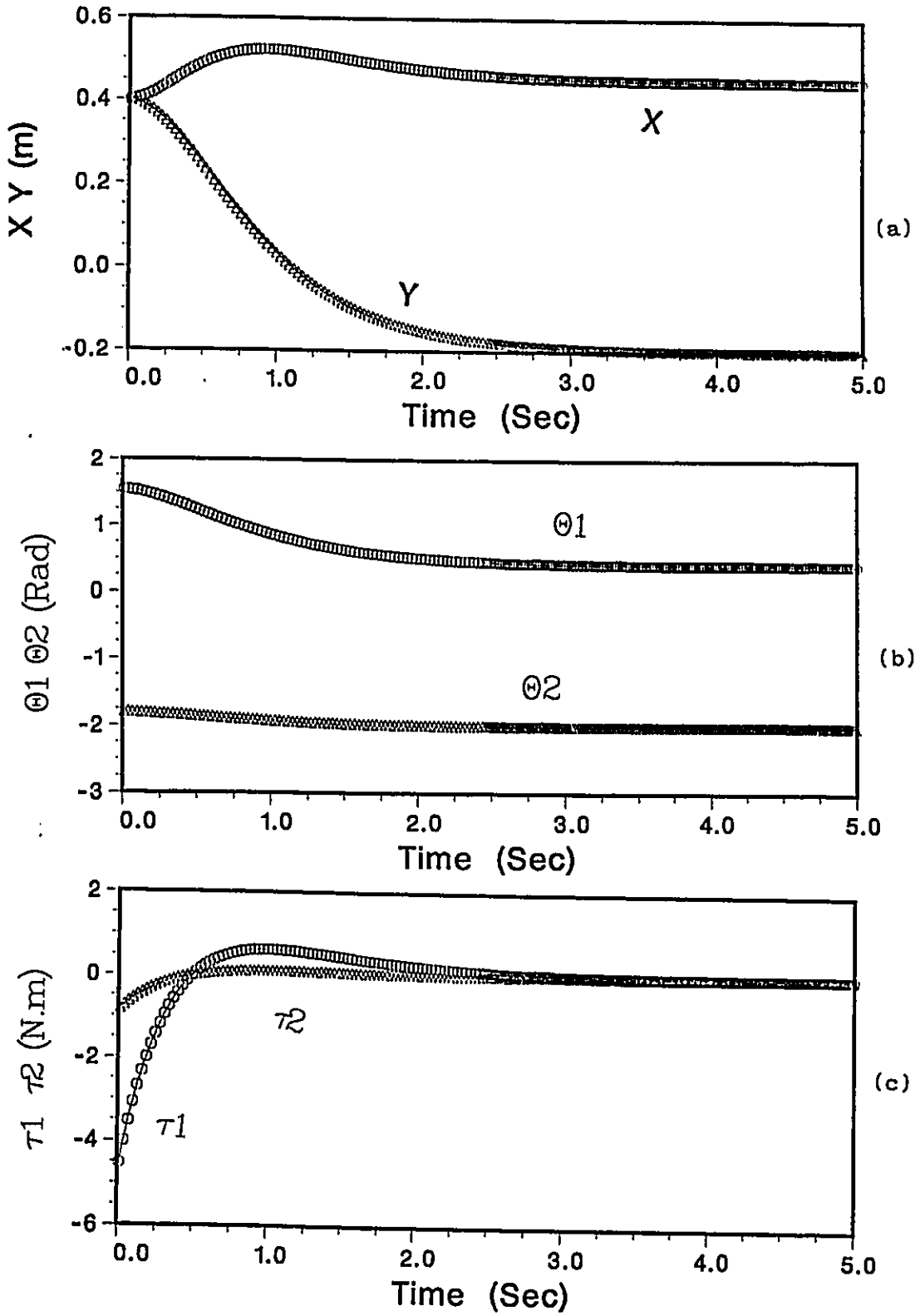


Figure 3.1.6 Simulation results of the robot shown in figure 3.1.5 :
(a) End-effector trajectory
(b) Joints angular positions
(c) Joint actuator torque

3.2 ATTRACTIVE FORCE FIELD

The attractive forces, generated in impedance control, can be explained using the simple artificial or virtual mechanical system shown in figure 3.2.1.

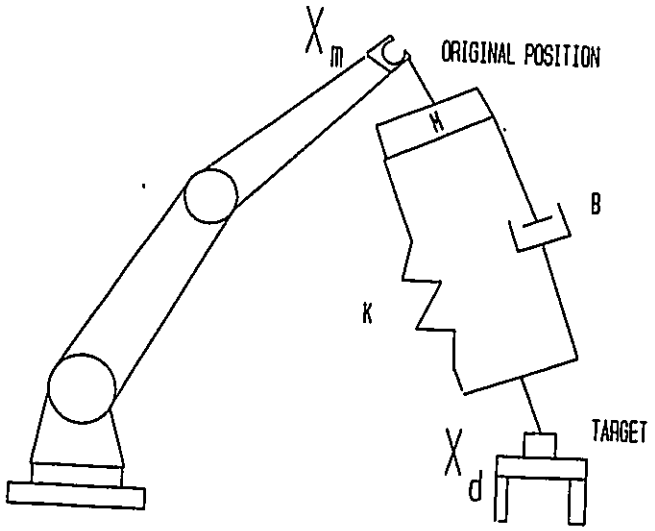


Figure 3.2.1 Artificial Mechanical Impedance B-K System

The virtual spring results from the relation between the force and the position error $\bar{X}_d - \bar{X}_m$. The virtual damper results from a relation between the force and the velocity $\dot{\bar{X}}_m$ producing a dissipation term, where the target is stationary. The control law expressed in Cartesian space for the case of no other external forces is:

$$\bar{F} = K\,(\bar{X}_d - \bar{X}_m) - B\,\dot{\bar{X}}_m \qquad (3.2.1)$$

where:

$$\dot{X} = J(\theta)\ \dot{\theta} \qquad (3.2.2)$$

$$X = KIN\ \{\theta\} \qquad (3.2.3)$$

where $J(\theta)$ is the Jacobian of the robot

Joint actuator torques are given by Eq. 3.1.1:

$$\tau = J^T(\theta)\ f$$

The control law, Eq. 3.2.1, will accomplish Cartesian end-point position control. Here, the inverse kinematics problem is completely eliminated, and only the forward kinematics equation 3.2.3 is needed. This fact is particularly important for those manipulators for which no closed-form solution to the inverse kinematics problem exists.[3] The virtual spring constant(K) and virtual

damping constant(B) can be obtained using a desired natural frequency, ω_n, a desired damping coefficient of the system:

$$\omega_n = \sqrt{\frac{K}{M}} \qquad (3.2.4)$$

$$\zeta = \frac{B}{2\sqrt{K}} \qquad (3.2.5)$$

For critical damping(ζ=1) we obtain:

$$B = 2\sqrt{K} \qquad (3.2.6)$$

For a virtual spring constant chosen to be K=4 . Equation 3.2.6 gives B=4.

3.3 ROBOT WORKVOLUME

The workvolume is defined by the robot joint limits. In order to stop the robot from trying to reach points near end at the limits of workvolume, artificial dampers for joint limits are considered. They bring the robot to a stop at joint limits and smoothly direct the robot to the target position within workvolume. From equation 3.1.1 :

$$\begin{cases} \tau = J^T f & \text{MINQ} < \theta < \text{MAXQ} \\ \tau = J^T f - \dot{\theta} K_Q & \text{MINQ} \geq \theta \geq \text{MAXQ} \end{cases} \qquad (3.3.1)$$

where: θ = vector of joint angles

In section (4.6), simulation results showing the effect of the work volume limits are presented.

3.4 JOINT TORQUE LIMITS

Each joint actuator has limit torques and can produce forces in Cartesian space up to a maximum value specific to each robot state. In some cases, the actuator can reach the saturation point if the error inputs and controller gains are high.

In this thesis, a "Cartesian Rescaling Scheme" was developed to guarantee that the forces in the direction from the end-effector to the target are the result of joint torques within joint torque limits. In this scheme, each of the joint torques are compared continuously with their limits, and if any violation occurs then all joints will have the torques rescaled until all torques are within limits.

Therefor, from equation 3.1.2 :

$$\ddot{X} = \left[J^T(\theta)\, M_x(\theta) \right]^{-1} \left[\tau - J^T(\theta)\, (V_x + G_x) \right]$$

from figure 2.4:

$$\ddot{X} = M^{-1} \left[K\,(X_d - X) - B\,\dot{X} + F_o \right]$$

from equation 3.2.1 the resultant force is:

$$F = K\,(X_d - X) - B\,\dot{X} + F_o$$

In this thesis, M^{-1} is considered an identity matrix are:

$$\ddot{X} = M^{-1} F = F$$

There are three conditions for 3 DOF robot. First condition is when first joint reaches the saturation point,

therefore:

$$\tau_1 = \text{T1MAX}$$

the resultant force is:

$$F = \left[J^T(\theta)\ M_x(\theta)\right]^{-1} \left[\tau_1 - J^T(\theta)\ (V_x + G_x)\right]$$

τ_2 and $\tau_3$can be recalculated by equation 3.1.1 . Second and third conditions are for the case when joints 2 and 3 reaches their respective saturation points and the same procedure as shown above can be used to calculate other joints torques. The two subroutines TORQUE and TORFX in Appendix B are based on the above equations.

As a result of rescaling, the cartesian force is reduced up to the level at which no joint torque limit is violated.

Figure 3.4.1(curve a) shows the simulations results where one actuator reaches its limits and no correction of the command torque is performed. By applying the Cartesian Rescaling Scheme straight line trajectory is obtained again, as in figure 3.4.1(curve b) .

In figure 3.4.2, joints torques in both cases are compared and the maximum torques each actuator can produce are: for joint one 3.0 N.m and for joint two 1.4 N.m .

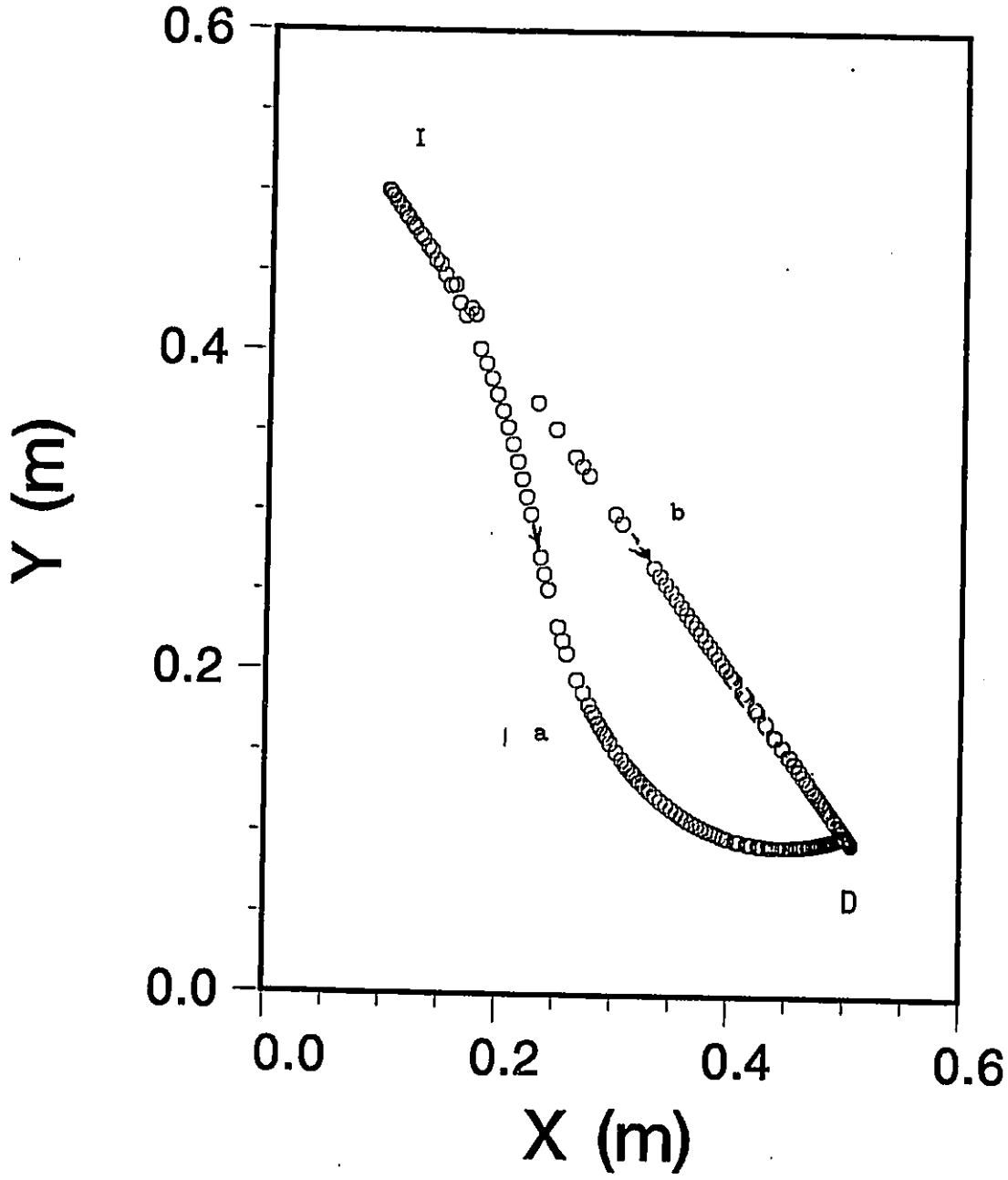
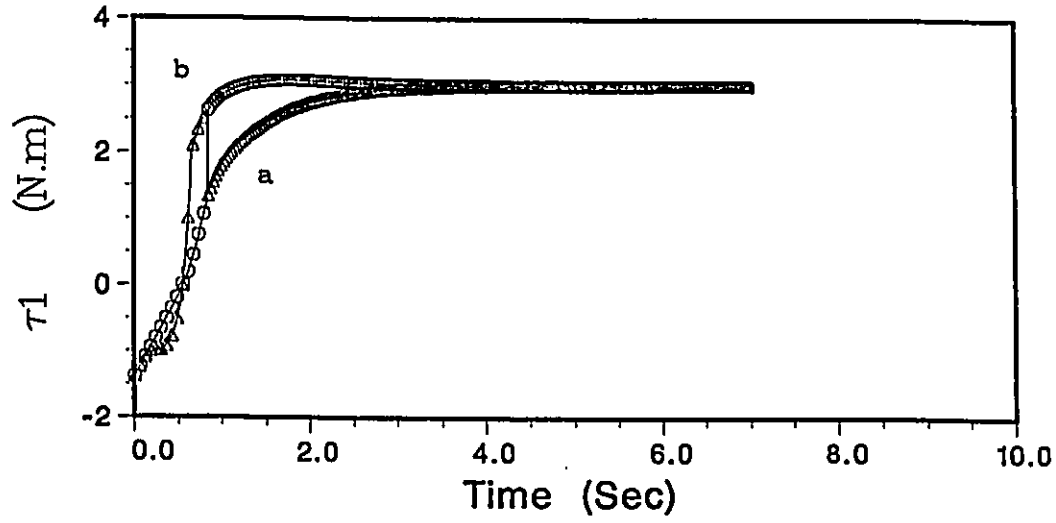


Figure 3.4.1 End-effector trajectory when one of the joint's torque reaches the saturation point.
(a) without using any scheme
(b) with using the Cartesian Rescalling Scheme
(K=4 N/m, B=4 N/(m/s), K_c=0.0, K_Q=0.0, TMAX1=3.0 N.m, TMAX2= 1.4 N.m.)



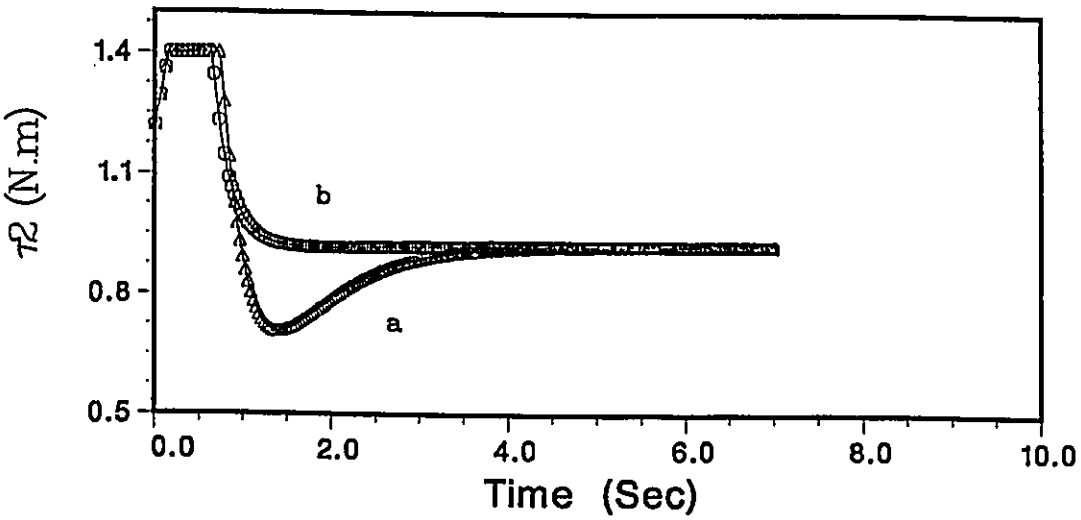


Figure 3.4.2 Joint torques for the case (a) and (b) of figure 3.4.1

3.5 MOVING TARGETS

In this case, the impedance control system will be based on the instantaneous position and velocity of the target. Therefore, the control law will be:

$$\bar{F} = K\,(\bar{X}_d - \bar{X}_m) + B\,(\dot{\bar{X}}_d - \dot{\bar{X}}_m) \qquad (3.5.1)$$

The velocity of the target is a very important factor. If the target is moving with low velocity, the manipulator can reach the target. If the target is moving relatively fast, then the robot might not be fast enough to reach the moving target. For simulations, two targets are illustrated, one moving on a straight line and second moving in a circle.

Figure 3.5.1 shows a target moving on a straight line trajectory, starting at D_o. The target is intercepted at D_i.

Figure 3.5.2 shows a target moving in a circular motion with an angular velocity 0.3 rad/s. The target is intercepted at D_i.

In figure 3.5.3, the target moving with 3.0 rad/s angular velocity is never intercepted. In fact, artificial impedance parameters, M=1, B=4 and K=4 correspond to natural frequency of 2 rad/sec, lower than the target angular velocity.

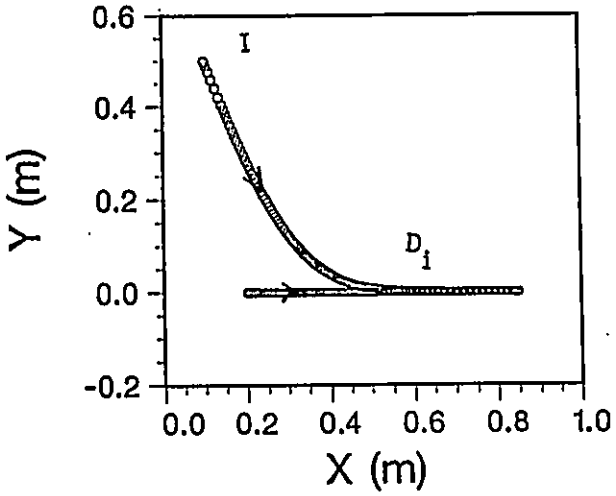


Figure 3.5.1 Interception of a target moving on a stright line (K=4 N/m, B=4 N/(m/s))

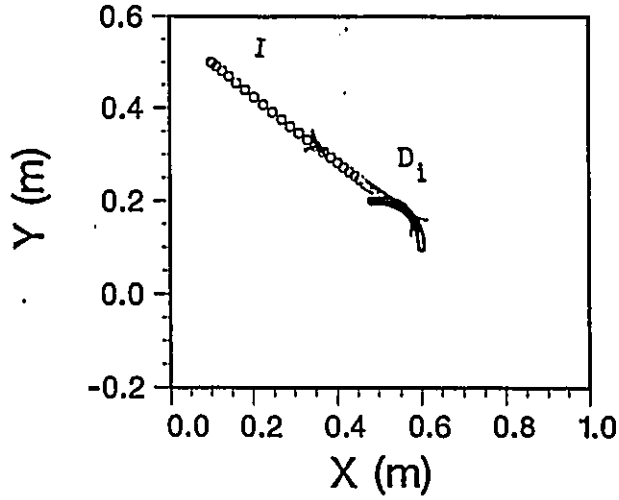


Figure 3.5.2 Interception of a target moving on circular path at low angular speed (K=4 N/m, B=4 N/(m/s), ω=0.3 rad/sec)

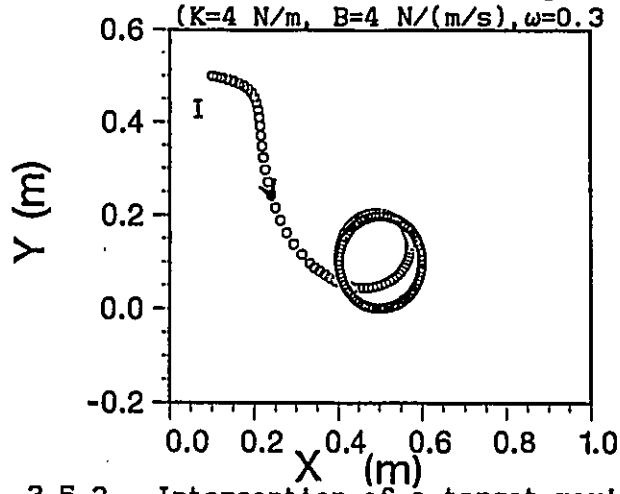


Figure 3.5.3 Interception of a target moving on circular path at high angular speed (K=4 N/m, B=4 N/(m/s), ω=3 rad/sec)

4.0 COLLISION AVOIDANCE OF THE ROBOT ARM WITH OBSTACLES

4.1 INTRODUCTORY REMARKS

Avoiding obstacles in the work volume has always represented a major difficulty for robots trajectory planning. A complex vision system was considered for detecting the obstacles and off line trajectory planning was used to avoid them. [22]

Obstacle avoidance capability is one of the important features which distinguishes an impedance controller from a PID controller's capabilities. The impedance controller can generate the trajectory on-line and, to avoid an obstacle, the only information needed is the distance to the closest point on the obstacle with reference to the end-effector. Therefore, the robot does not have, in this case, to be equipped with complex vision systems or to be preprogramed for any new trajectory. Impedance control system corrects the present trajectory of the end-effector into a new trajectory which avoids obstacles; only in critical

situations the robot will stop at a safe distance from obstacles. In order to do this, a repulsive force field is introduced by the impedance controller figure 4.1.1 . The repulsive field is created around all obstacles up to a distance of effectiveness. The virtual repulsive forces increase as the robot gets closer to the centre of the field or to the obstacle. Beyond the limit of effectiveness the repulsive forces are zero and have no effect on robot trajectory. Within this limit, the resultant of the attractive and repulsive forces direct the robot to a new feasible trajectory.

For the operation of this impedance controller, the robot has to be equipped with a proximity sensor which gives the information needed for the calculation of the shortest distance between the end-effector and obstacles. Often, the "Polaroi" Ultersonic Proximity Sensor" is used for this purpose.

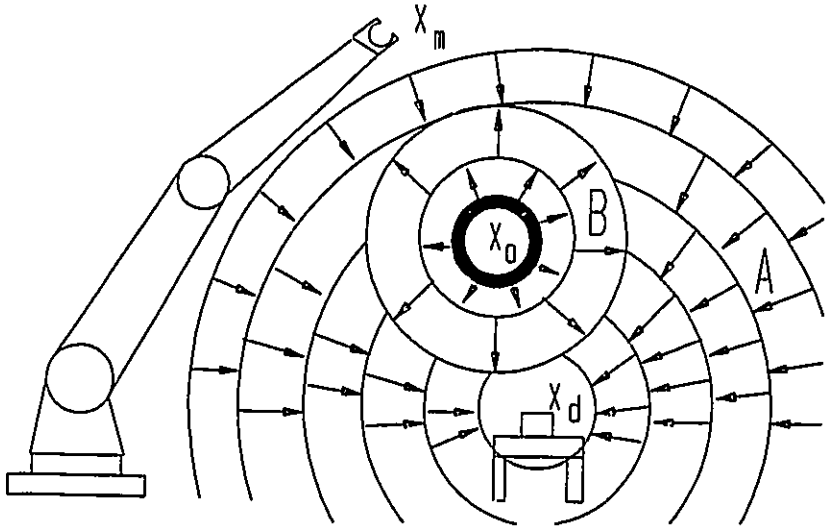


Figure 4.1.1 Attractive and Repulsive force field
A= attractive force field
B= repulsive force field

4.2 Repulsive field of forces:

A proximity sensor can provide data concerning the shortest distance between the end-effector and the obstacle. The forces in the repulsive field are generated with a direction from the obstacle toward the end-effector. For a 2 DOF robot the calculation of the repulsive field is described by the following equations using vector notation: [30]

- general equation of the repulsive force

$$\begin{cases} \bar{F}_o = (F_{o_x})\,\bar{i} + (F_{o_y})\,\bar{j} & R_{om} < R_o \\ \bar{F}_o = 0 & R_{om} > R_o \end{cases} \qquad (4.2.1)$$

$$X_{om} = X_m - X_o$$
$$Y_{om} = Y_m - Y_o$$

$$F_{o_x} = K_o\,(R_{om}-R)^2\,\frac{X_{om}}{R_{om}} \qquad (4.2.2)$$

$$F_{o_y} = K_o\,(R_{om}-R)^2\,\frac{Y_{om}}{R_{om}} \qquad (4.2.3)$$

where:

$$X_{om} = X_m - X_o$$
$$Y_{om} = Y_m - Y_o$$
$$\bar{R}_{om} = (X_m-X_o)\,\bar{i} + (Y_m-Y_o)\,\bar{j} \qquad (4.2.4)$$

$$\bar{R}_{om} = X_{om}\,\bar{i} + Y_{om}\,\bar{j} \qquad (4.2.5)$$

$$R_{om} = \sqrt{X_{om}^{2} + Y_{om}^{2}} \qquad (4.2.6)$$

where $\bar{i}$ and $\bar{j}$ are the unit vectors in the Cartesian plane and the bars denote vectors.

Repulsive forces are designed to have no effect on the control system beyond the limit of effectiveness. Within the limits of effectiveness of the repulsive field, the repulsive forces become stronger as the robot end-effector gets closer to the obstacle and push away the robot, eventually directing it to a new feasable path.

Figure 4.2.1 shows the resultant F of the repulsive forces F_o and the attractive force F_{x_d} which direct the robot to a new trajectory.

A first set of simulations were performed for a 2 DOF planar robot arm. Figure 4.2.2 shows the simulation results of the trajectory ID of a robot avoiding an obstacle with a circular shape located on the path between robot current position and the target. The robot in this case has a feasible trajectory which avoids the obstacle and reaches the target. As shown on figure 4.2.2, the robot controller needs only the value of the shortest distance between the obstacle and the end-effector at any time. Therefore, rather than the complete shape of the obstacle being shown in

figure 4.2.2, only the part of the whole obstacle, marked by small triangles, that the robot could "see" during its motion, is shown.

Figure 4.2.3 shows the simulation results of the end-effector position, joint positions and actuator torques for the case shown in figure 4.2.2. The robot reached the target within 11 seconds without any oscillations in end-effector and joints positions. Furthermore, the joints did not exceed their limiting a angular positions.

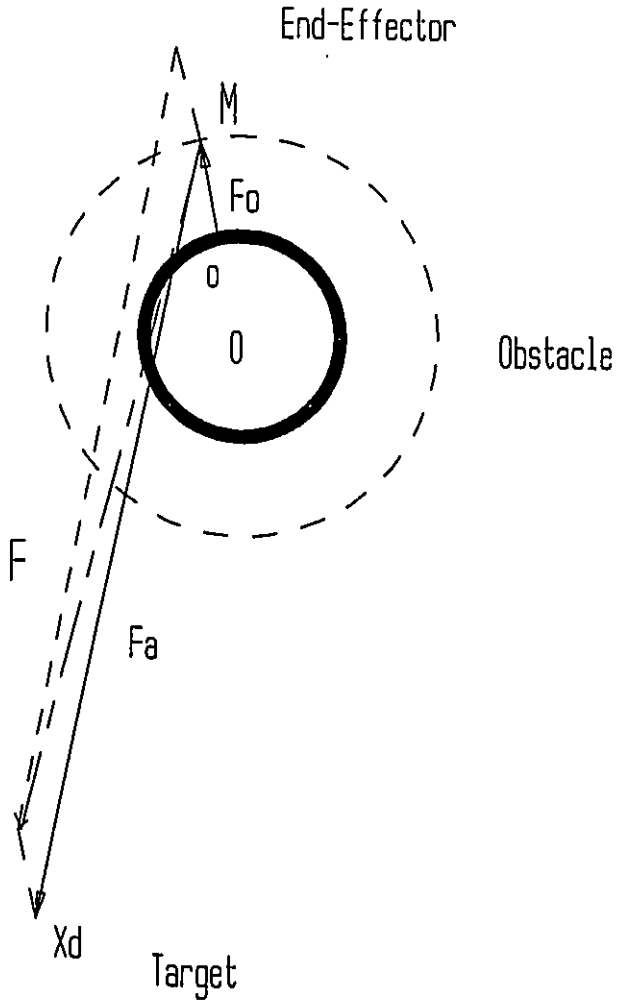


Figure 4.2.1 Repulsive and attractive forces

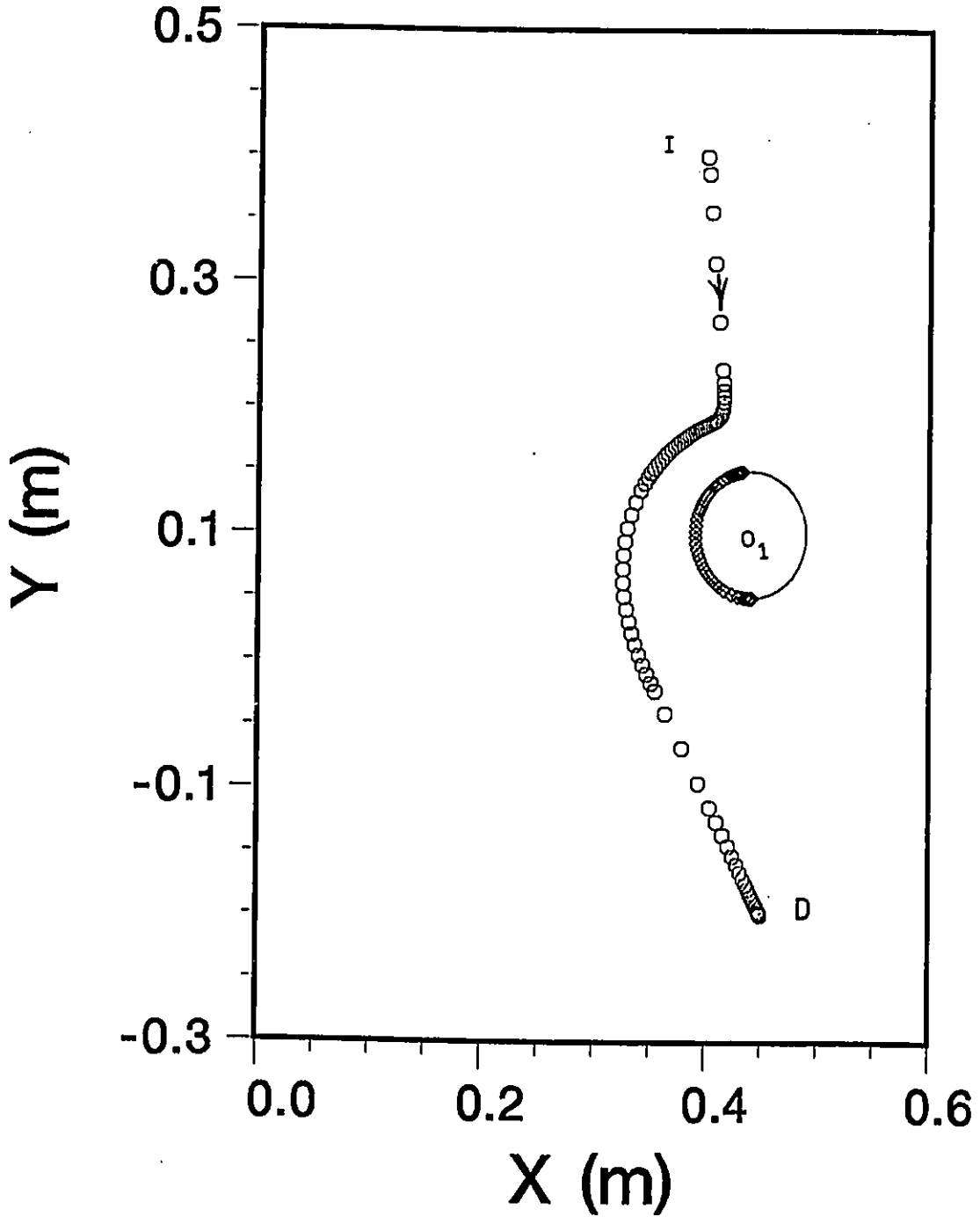


Figure 4.2.2 End-effector trajectory of a robot which avoids an obstacle. (K=4 N/m, B=4 N/(m/s), R=0.1 m, K_o=500 N/m, B_o=20 N/(m/s), K_c=0.0, K_q=0.0)

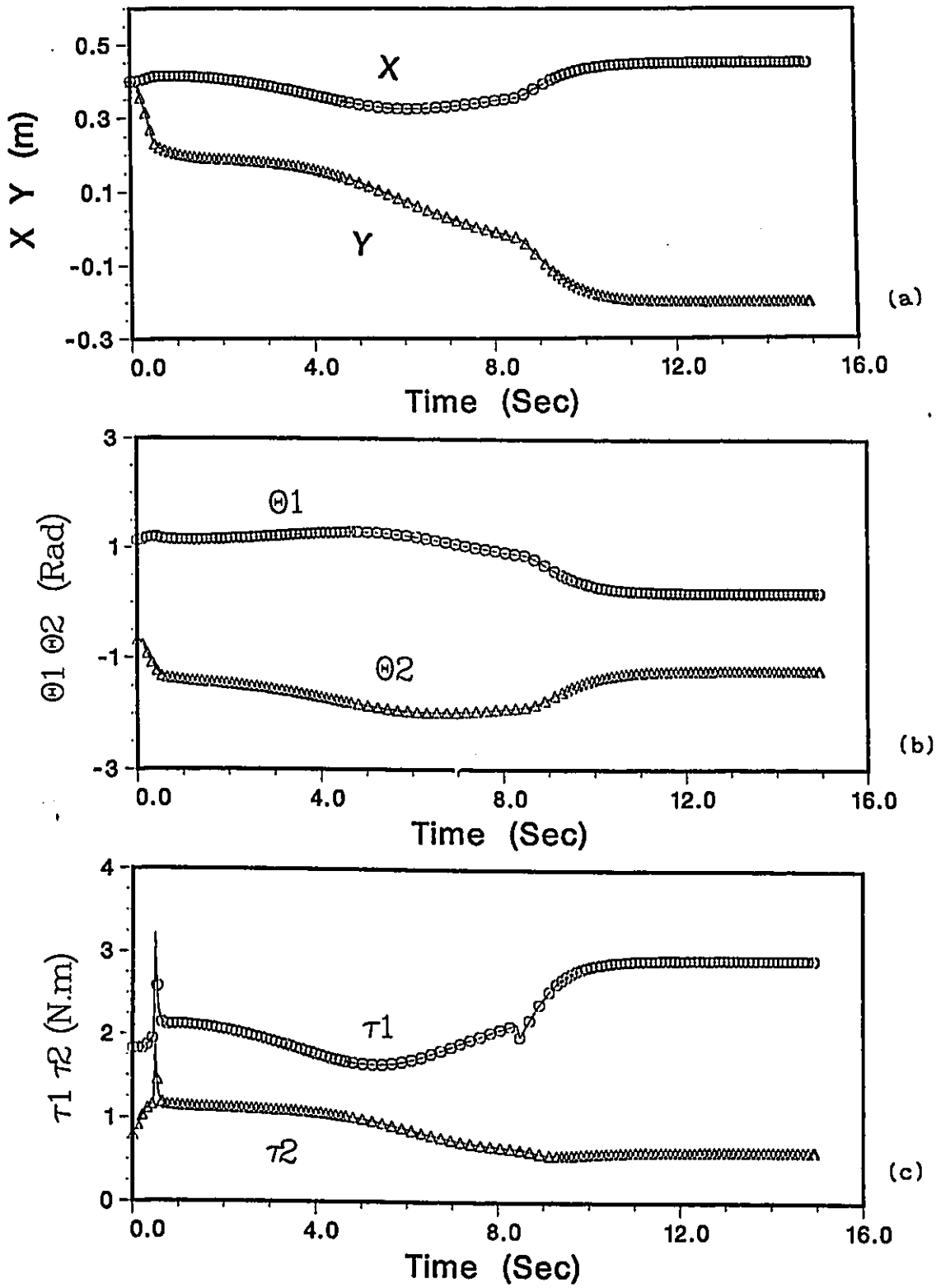


Figure 4.2.3 Simulation results of the robot shown in figure 4.2.2
(a) End-effector position
(b) Joints angular positions
(c) Actuator torques

A second set of simulations were performed for a three DOF revolute robot. Figure 4.2.4 shows obstacle avoidance in 3-dimensional space. In this case, the obstacle has a spherical shape, but the robot sees only the closest points of the sphere as it moves and only these points are shown in the graph (marked by triangles). Figure 4.2.5 shows the time behavior of the end-effector position, joints positions and actuators torques of the robot when no damping is present in the repulsive forces. The robot reached the target within 8 seconds and there is no undesired transient behavior at the joints and end-effector. Some minor oscillations occur just before the robot starts avoiding the obstacle. The joints did not exceed their limits.

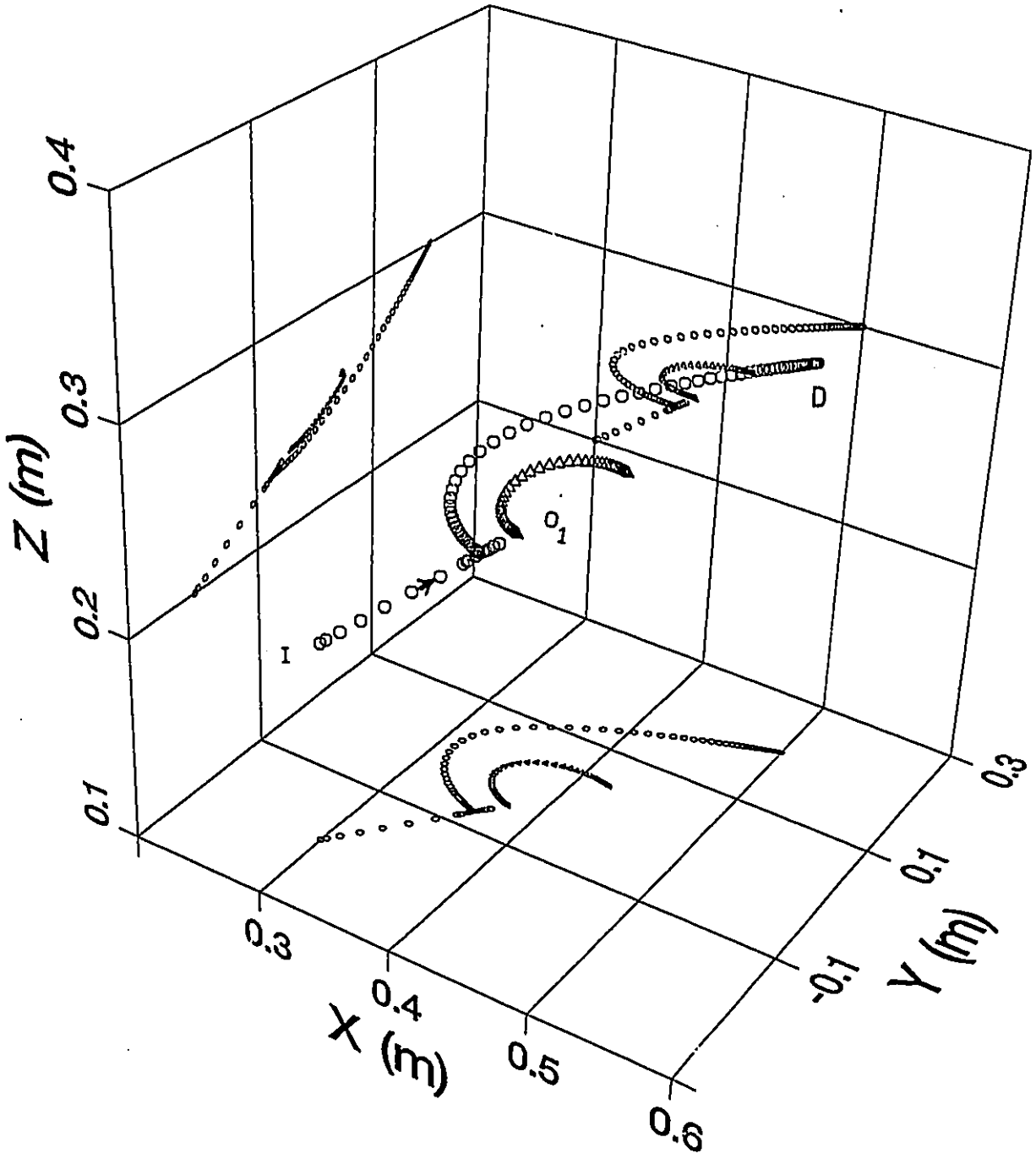


Figure 4.2.4 End-effector trajectory ID of a robot avoiding an obstacle O.
(K=4 N/m, B=4 N/(m/s), R=0.05 m, W=0.0, K_o=5000 N/m, B_o=0.0, K_q=10 N/(m/s), R_s=0.05 m, $(X,Y,Z)_s$=(0.4,0.0,0.25) m)

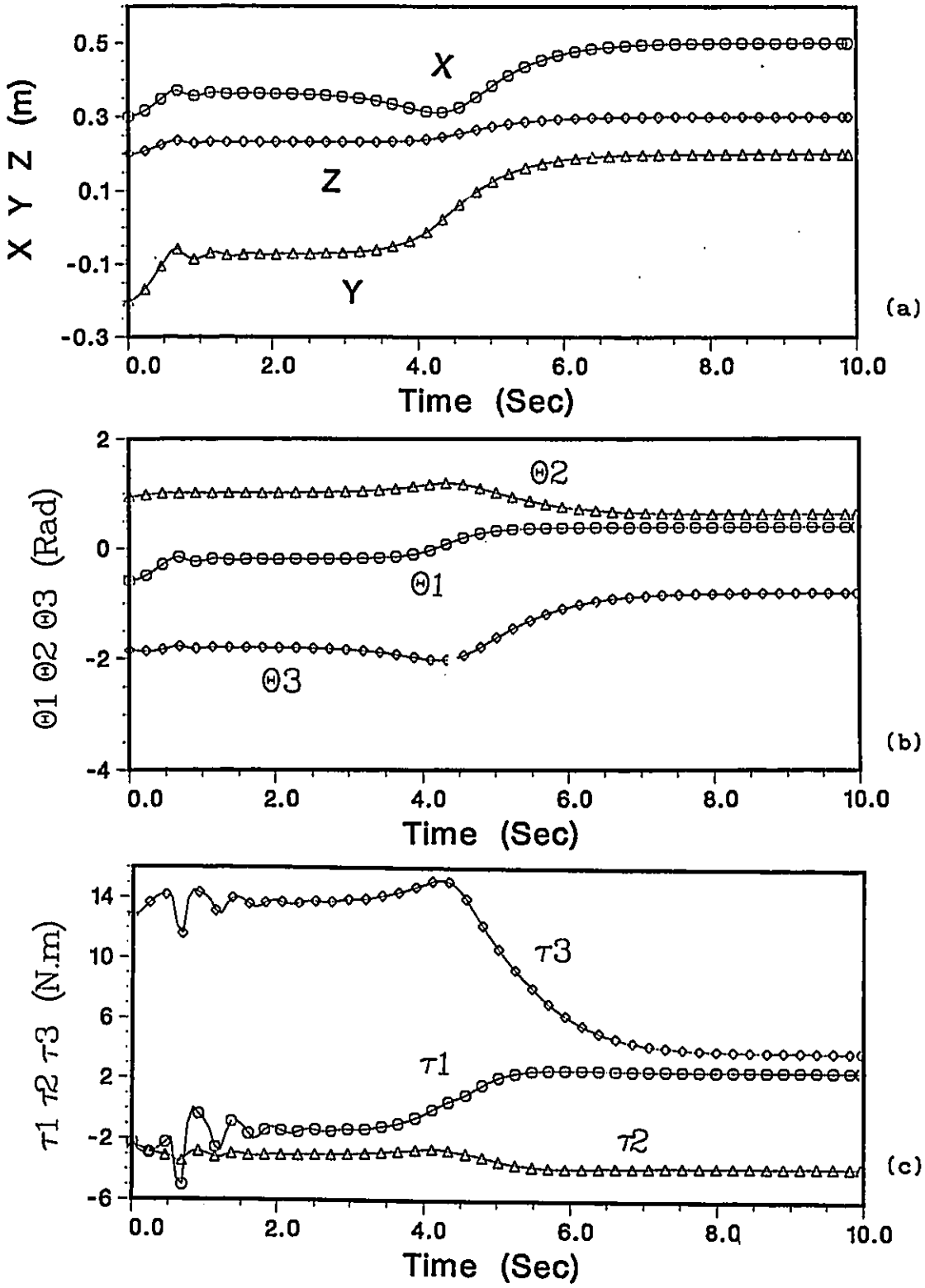


Figure 4.2.5 Simulation results of the robot shown in figure 4.2.4
(a) End-effector position
(b) Joints angular positions
(c) Actuator torques

4.3 Contribution of Damping in Repulsive Force Field:

Damping in repulsive force field contributes to generating a smoother robot path. Damping reduces the manipulator speed, preventing the manipulator from moving too quickly around an obstacle. It also contributes to achieving a safe distance between the robot and the obstacle.

In some special cases the robot might be trapped between two obstacles or within a cavity of complex obstacles. This might cause an oscillatory motion of the end-effector and in this situation the workpiece might slip in the gripper or, in the case of moving a container with liquid, it might lead to spilling the liquid. The avoidance of these end-effector oscillations can be achieved by choosing a suitable damping constant.

The selection of the damping constant (B_o) is based on taking into account that high B_o slows down the robot by a large factor.

Equations used for the repulsive force field, when damping is present, are:

$$F_{o_x} = K_o \, (R_{om}-R)^2 \, \frac{X_{om}}{R_{om}} \; - \dot{X}_m \, B_o \qquad (4.3.1)$$

$$F_{o_y} = K_o \, (R_{om}-R)^2 \, \frac{Y_{om}}{R_{om}} \; - \dot{Y}_m \, B_o \qquad (4.3.2)$$

Figures 4.3.1 & 4.3.2 show the results of a 2 DOF robot simulation for a case in which the resultant of the attractive and repulsive forces brings the end-effector to an equilibrium point(E), after entering the concave potential trap formed by the potential fields associated with the obstacles O_1,O_2 and the target point D.

In figure 4.3.1, there is no damping in repulsive forces and this results in an oscillatory motion. In Figure 4.3.2 damping was added to repulsive force(B=20) and this results in a trajectory without overshooting the equilibrium point.

In figure 4.3.3 a comparison of X(t) in both situations, B=0(curve a) and B=20(curve b) is presented.

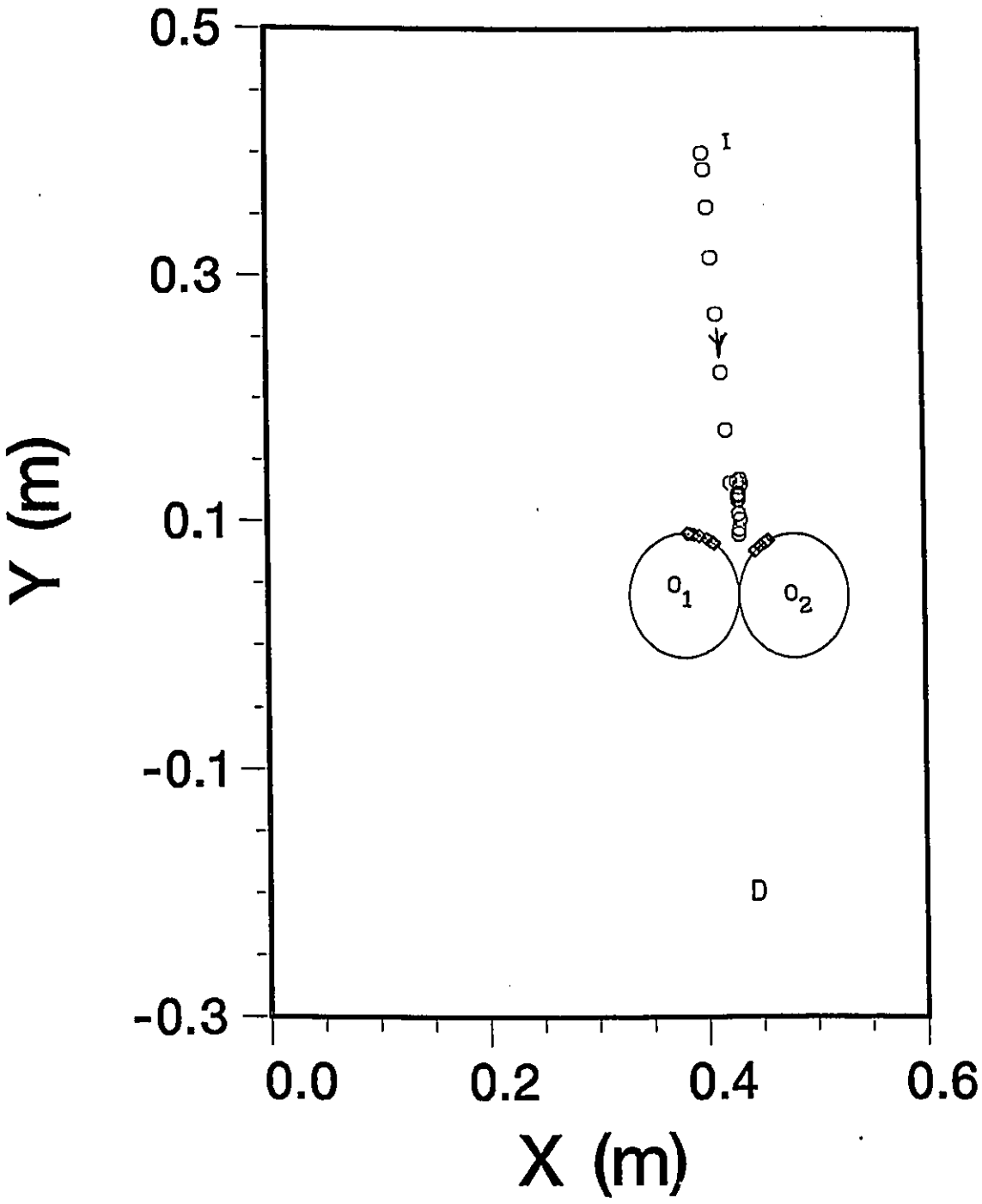


Figure 4.3.1 Robot trajectory without damping in the repulsive field
(K=4 N/m, B=4 N/(m/s), R=.1 m,
K_o=500 N/m, B_o=0.0, K_q=0, R_{o1}=R_{o2}=0.05 m,
$(X,Y)_{o1}$=(0.38,0.04), $(X,Y)_{o2}$=(0.48,0.04) m

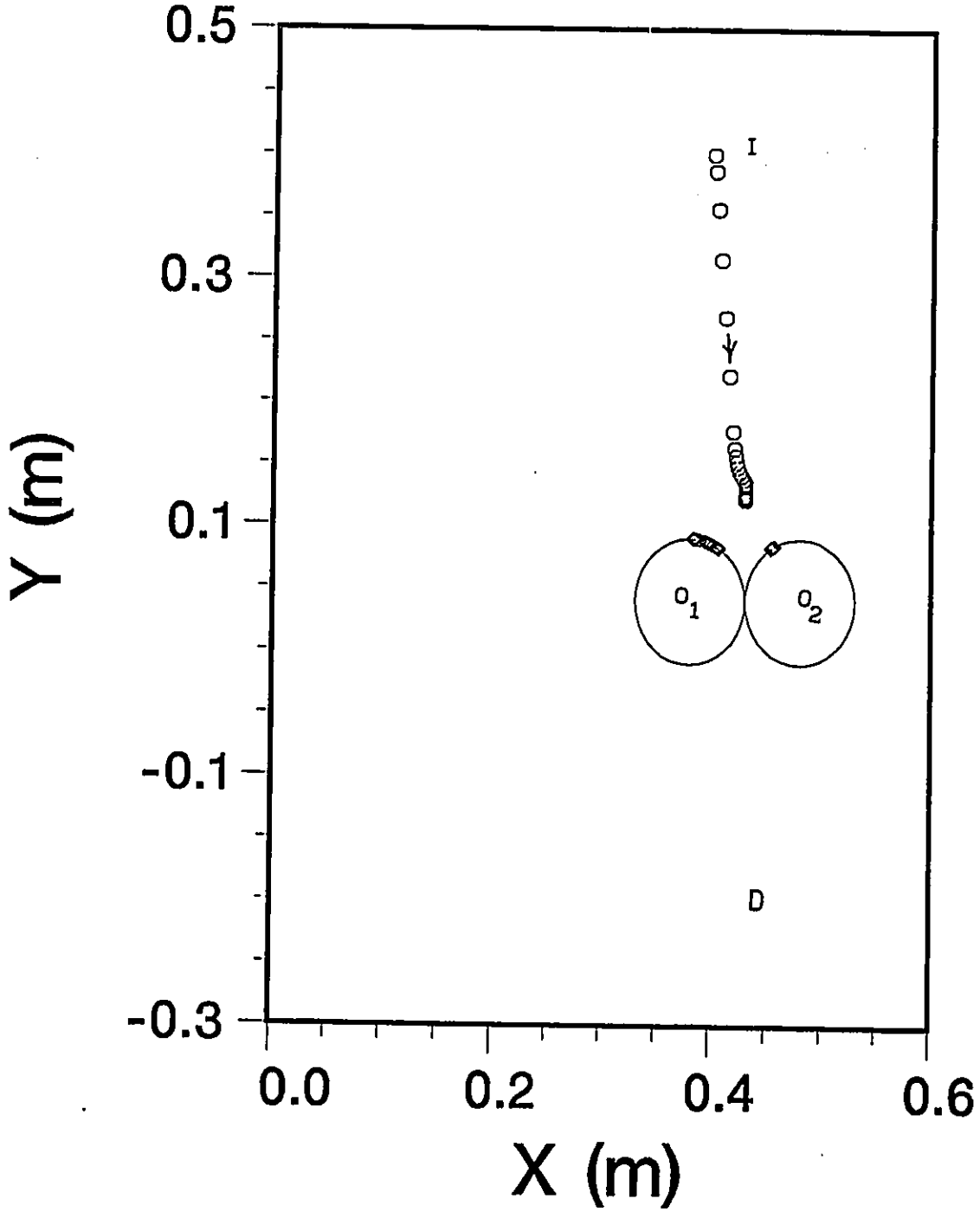


Figure 4.3.2 Robot trajectory with damping in in the repulsive field (same parameters as figure 4.3.1 except B_o=20.0 N/(m/s))

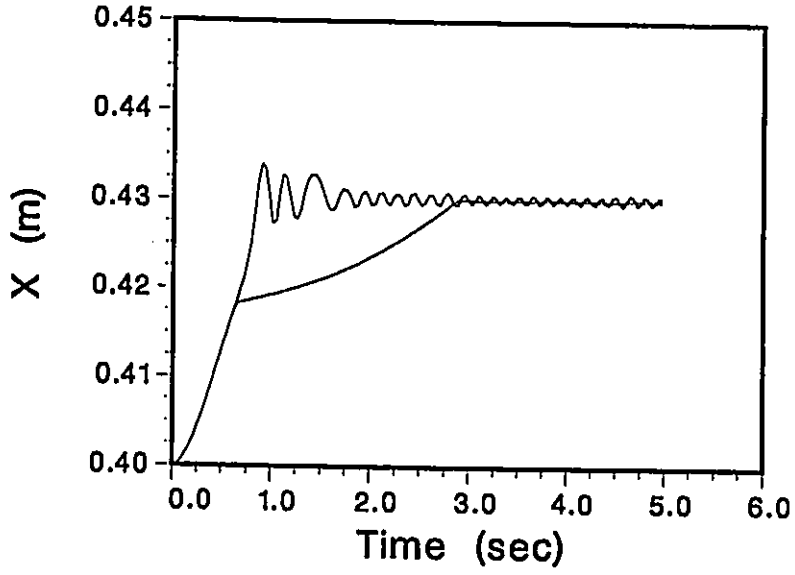


Figure 4.3.3 Comparison the effect of damping in the robot trajectory shown for the cases shown in figures 4.3.1 & 4.3.2

Simulation results for a 3 DOF robot are presented in figures 4.3.4 and 4.3.5. In figure 4.3.4, the trajectory of the robot avoids the obstacle as a result of repulsive forces with a damping component. Figure 4.3.5 shows the resulting end-effector position, joint position and actuators torques of the robot for the case shown in figure 4.3.4 . The robot reached the target within 15 seconds but there is no position oscillations in the end-effector or of the joints during obstacle avoidance.

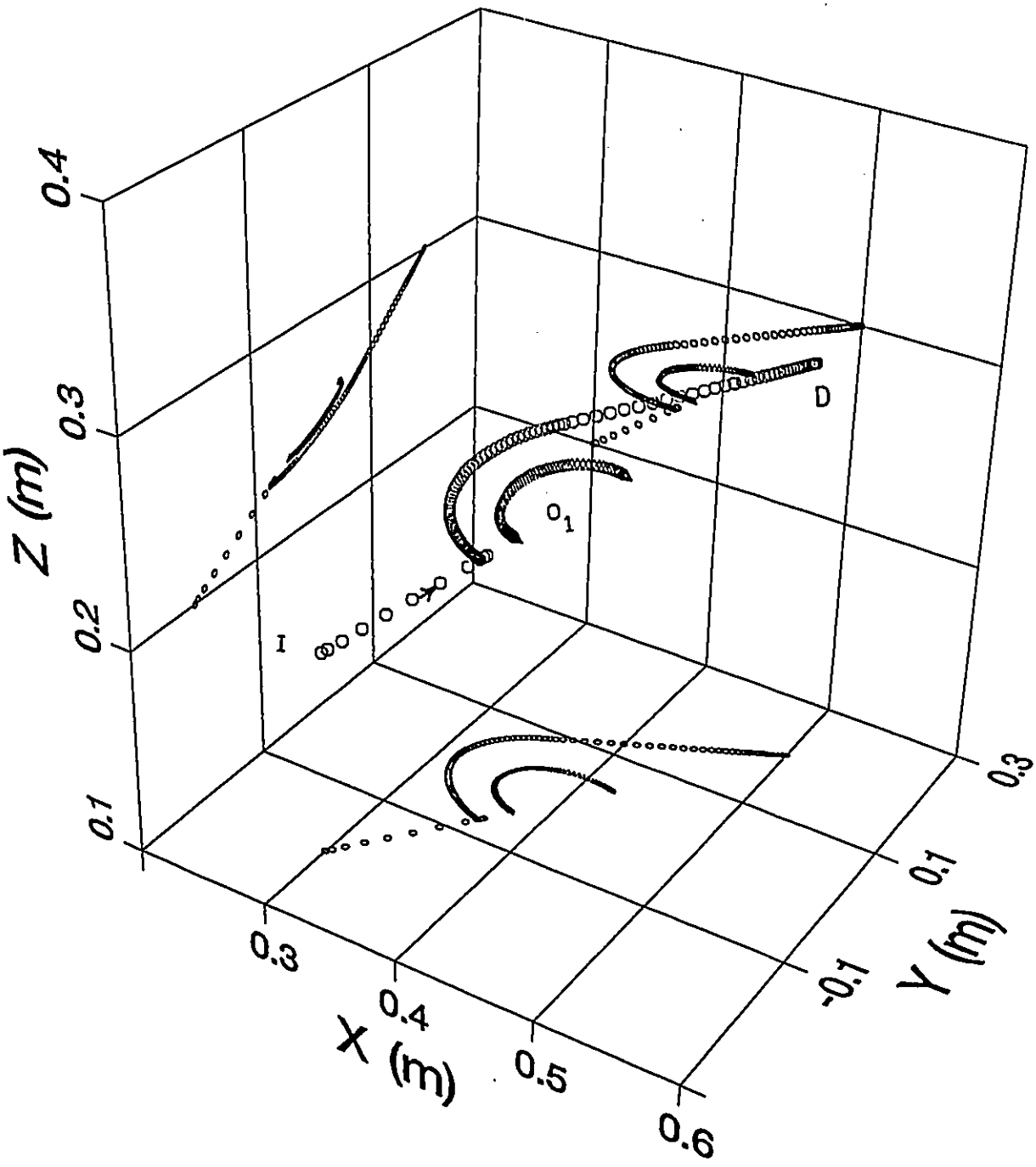


Figure 4.3.4 The trajectory ID of a 3 DOF robot in the presence of an obstacle O, when the repulsive forces have a damping component (K=4 N/m, B=4 N/(m/s), R=0.05 m, W=0.0, K_q=10 N/(m/s), K_o=5000 N/m, B_o=10 N/(m/s) $(X,Y,Z)_s$=(0.4,0.0,0.25) m, R_s=0.05 m)

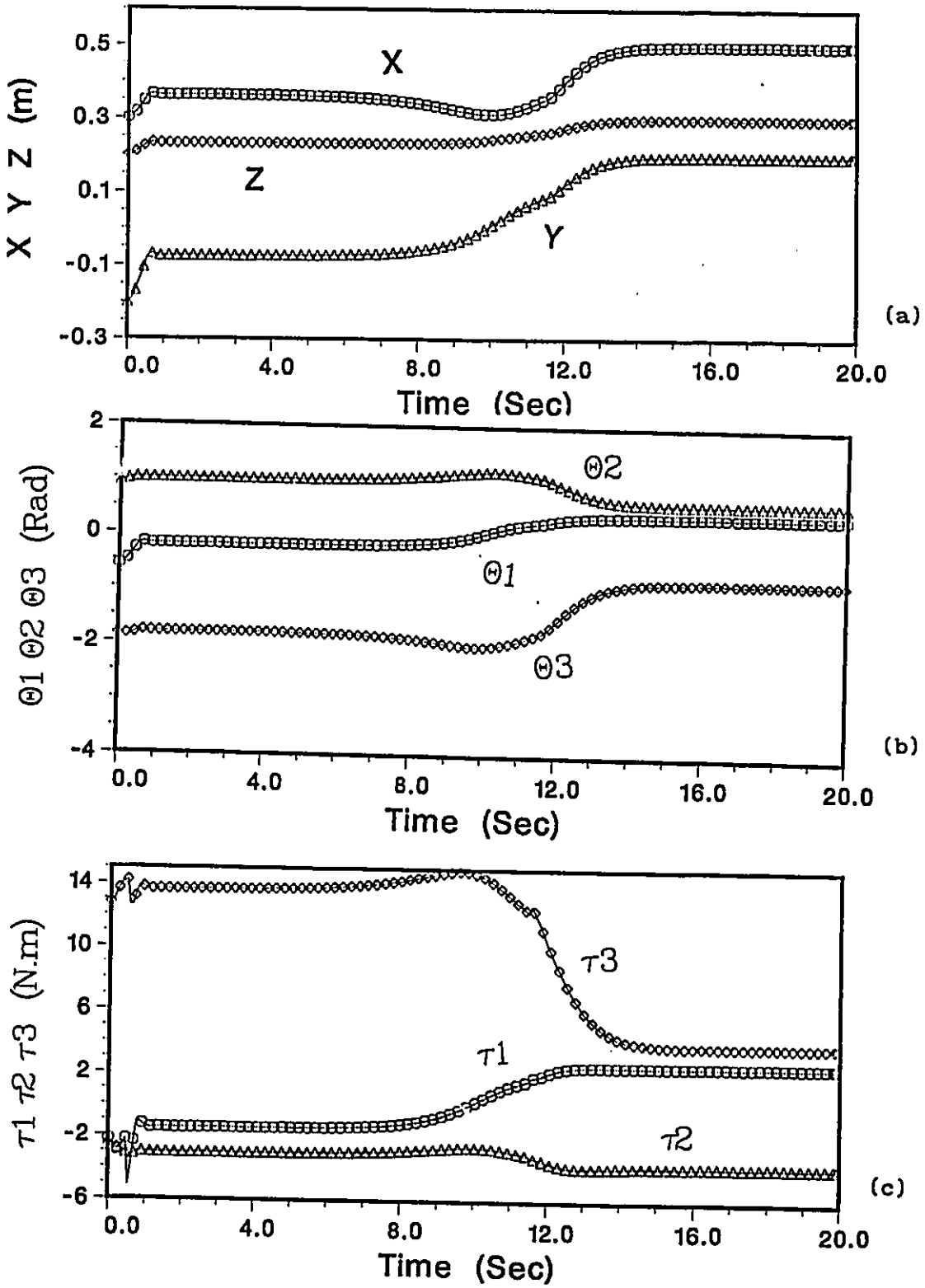


Figure 4.3.5 Simulation results of the robot shown in figure 4.3.4
(a) End-effector position
(b) Joints angular positions
(c) Actuator torques

4.4 TORQUE SATURATION LIMITS:

Simulations of a 3 DOF robot were performed for the case of set limits for torque saturation. After reaching the saturation point, the rescaling procedure described in section 3.4 led to a straight line trajectory presented in figure 4.4.1 .

Figure 4.4.2 shows the behavior of end-effector position, joints positions and actuators torques of the robot for the case in figure 4.4.1.

The robot reached the target within 4 seconds and there is no oscillations in end-effector and joints. As the results show, joint 3 reached the saturation point and the other two joint torques are recalculated. Also, joints did not exceed their limits.

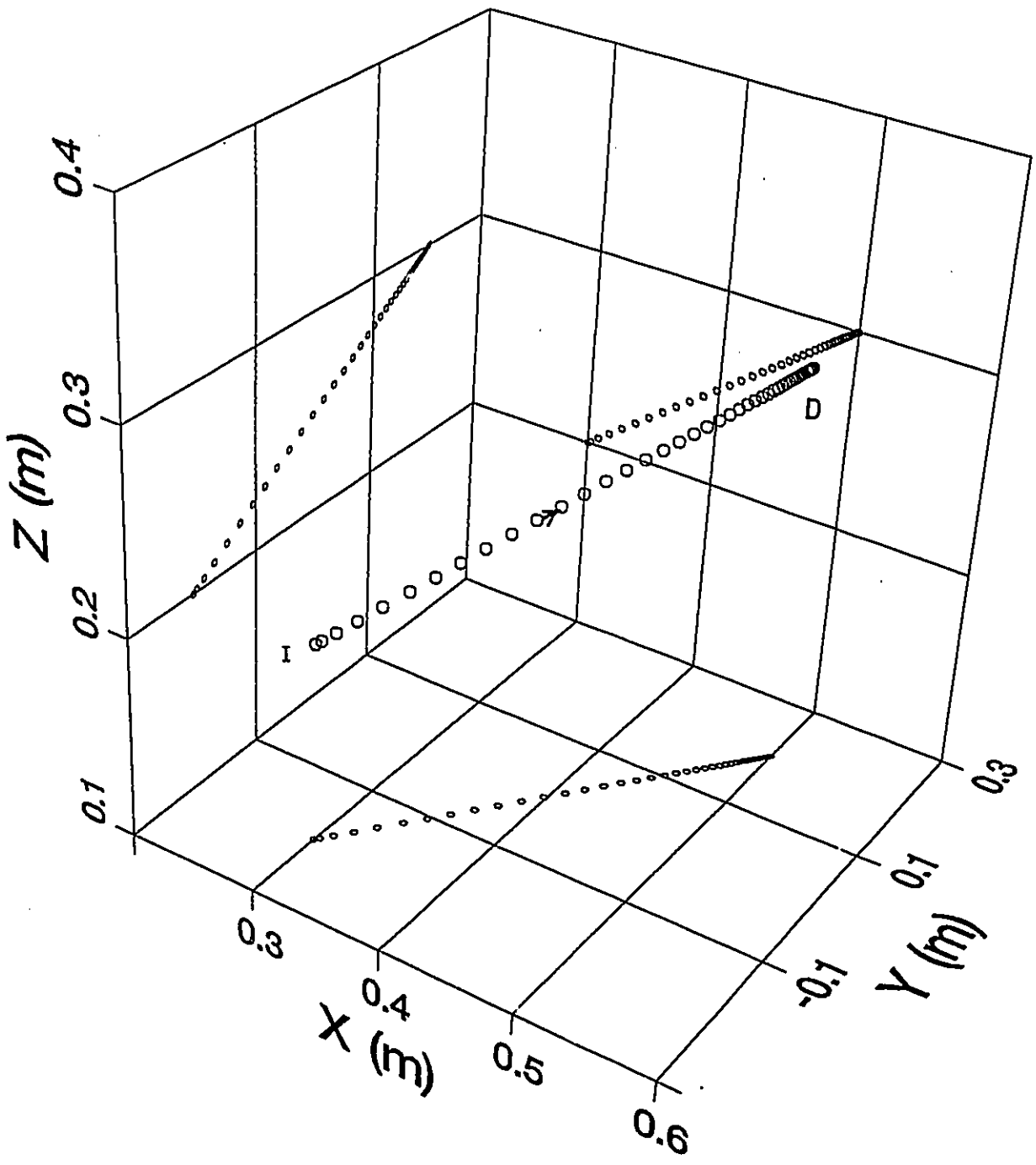


Figure 4.4.1 Robot trajectory with in saturation limits of joint
(K=4 N/m, B=4 N/(m/s), W=0.0, T1MAX= T2MAX=T3MAX= 14 N.m, K_Q=10 N/(rad/s))

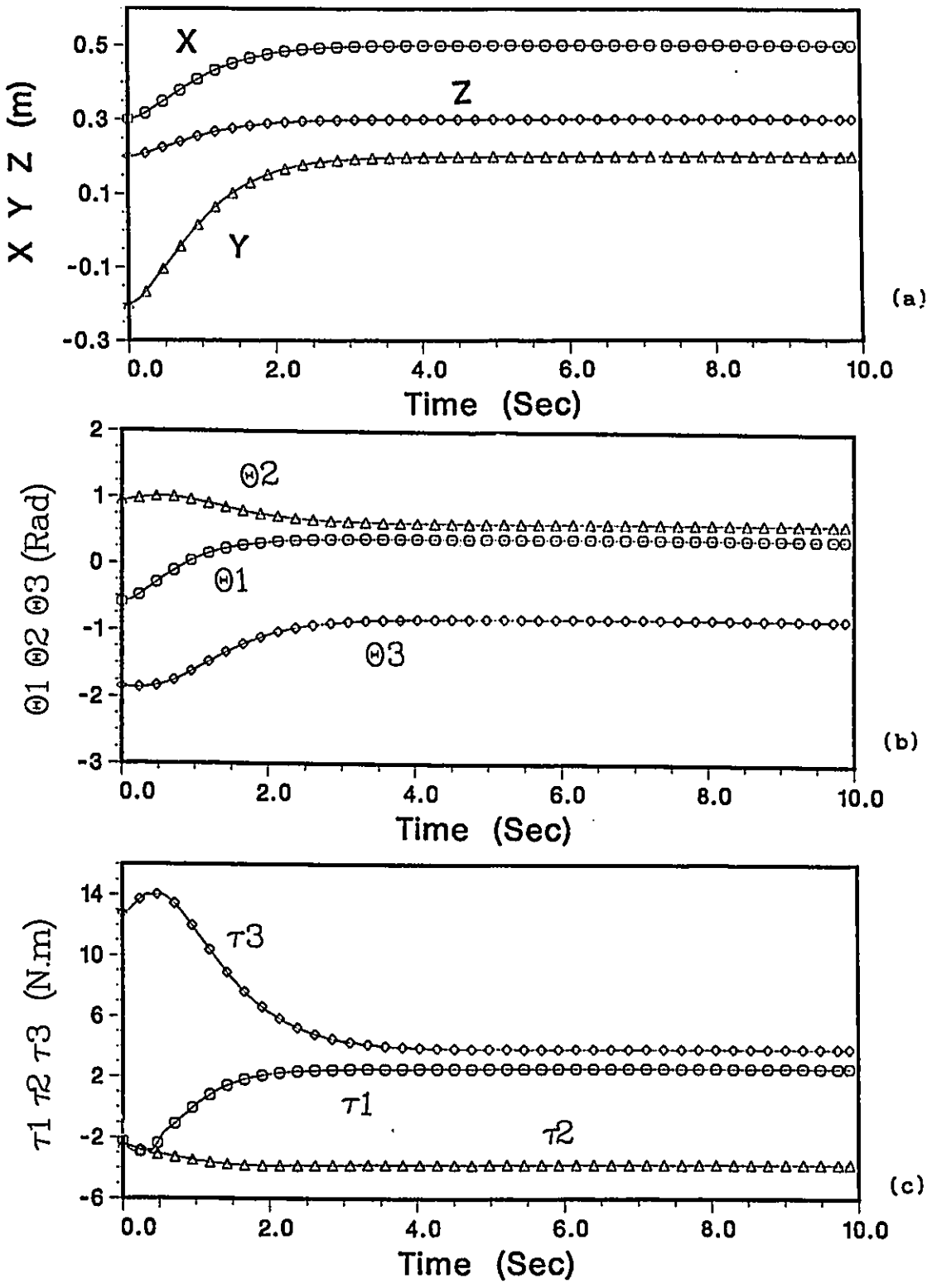


Figure 4.4.2 Simulation results of the robot shown in figure 4.4.1
(a) end-effector position
(b) joints angular positions
(c) actuator torques

4.5 COASTAL(CNS) AND SURFACE(SNS) NAVIGATION SCHEMES:

The net effect of attractive and repulsive forces applied to a manipulator end effector can lead to concave potential traps, which are local equilibrium points away from the target. Often, feasible trajectories toward the target still exist. A local equilibrium point may be identified by simply testing if the attractive force is zero while $X \neq Xd$. CNS and SNS were proposed for moving the robot end effector from a local equilibrium point around obstacles until a line-of-sight trajectory to the target is available.[25,26]

When a robot reaches an equilibrium point, the controller in a navigation scheme temporarily switches off the attractive and repulsive forces and the switches on CNS force for a 2D motion or SNS force for 3D motion until the line-of-sight to target is reached.

CNS applies to planar robots in two dimensional space and for the case represented in figure 4.5.1 the force equation is given by:

$$\bar{F}_c = K_c \frac{|\bar{R}_{om}|}{|r|} \bar{P} \qquad (4.5.1)$$

where:

$$|\bar{P}| = 1$$

$$\bar{P} \cdot \bar{R}_{om} = 0 \qquad (4.5.2)$$

K_c = coastal navigation constant

$\bar{F}_c$ = coastal navigation force vector

r = limit of repulsive field

$\bar{R}_{om} = \bar{R}_m - \bar{R}_o$

- The vectors are denoted by bars

Figure 4.5.2 shows how CNS finds the path for the robot away from the trap position and directs it toward the target.

Figure 4.5.3 shows the end-effector position, joint position and actuator torque for the case shown in the figure 4.5.2 . The robot reached the target with in 8 second. There are some position oscillations of the joints and the end-effector when the robot switches to coastal navigation scheme. The robot succesfully avoids the obstacle and reaches the target.

SNS applies for a robot which moves in a three dimensional space. The force given following equations (figure 4.5.4):

$$\bar{F}_s = K_s \frac{\bar{V}_s}{|V_s|} \qquad (4.5.3)$$

where:

$$\bar{N} = \bar{V}_2 \times \bar{V}_1 \qquad (4.5.4)$$

$$\bar{V}_s = \bar{V}_1 \times \bar{N} \qquad (4.5.5)$$

$$\bar{V}_1 = \bar{R}_{om} = \bar{X}_m - \bar{X}_o$$

$\bar{V}_2 = \bar{R}_{md} = \bar{X}_d - \bar{X}_m$

$\bar{V}_3 = \bar{R}_{ms} = \bar{X}_s - \bar{X}_m$

$\bar{F}_s$ = surface navigation force

K_s = surface navigation constant

Figure 4.5.5 shows how the SNS forces moves the end-effector from the trap and direct it to the target.

Figure 4.5.6 shows the end-effector position, joint position and actuator torque for the case shown in figure 4.5.5.

Figure 4.5.7 shows another successful test of SNS, in which the end-effector bypasses an obstacle and reaches the target. Also figure 4.5.8 shows the end-effector position, joint angles and actuator torques of the robot.

When, as a result of the application of CNS or SNS, a singular point is reached, forces will have their direction reversed to seek a feasable path on the other side of the obstacle.

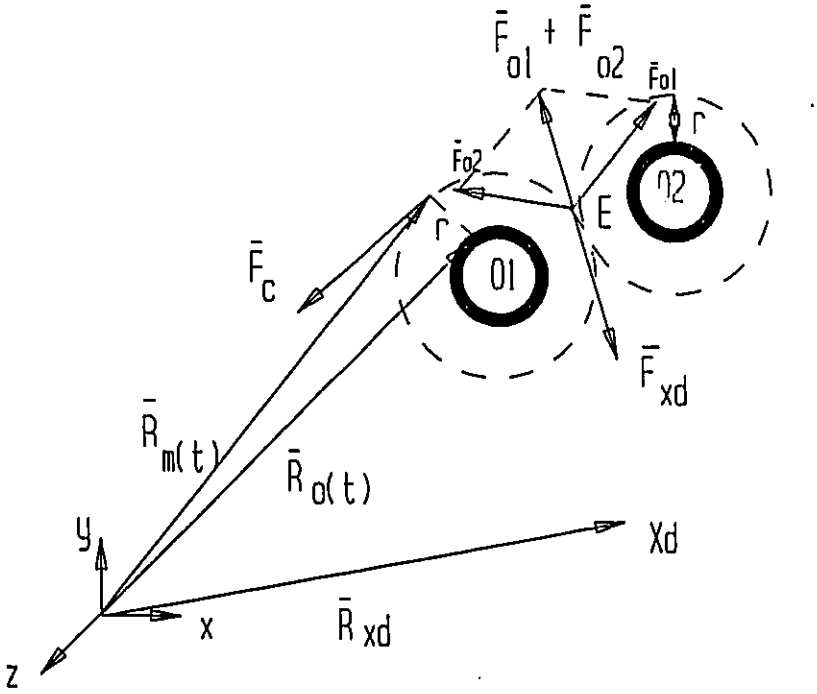


Figure 4.5.1 Coastal Navigation Scheme (CNS)

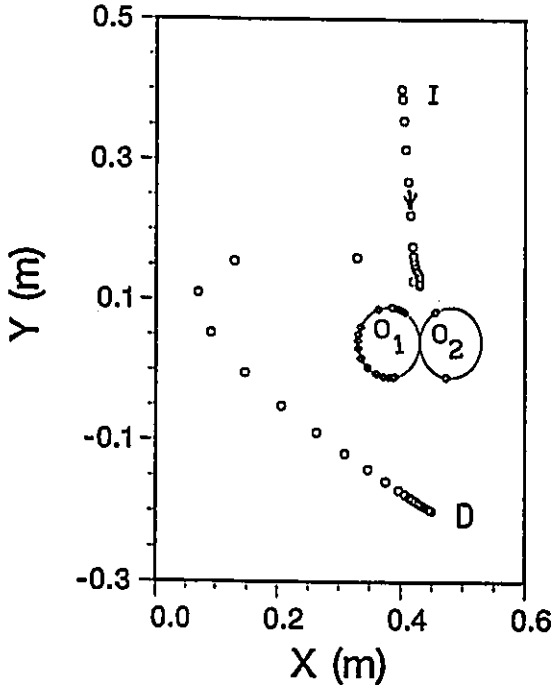


Figure 4.5.2 End-effector trajectory of a robot avoiding two obstacles (same parameters as figure 4.3.2 except K_c=4 N/m)

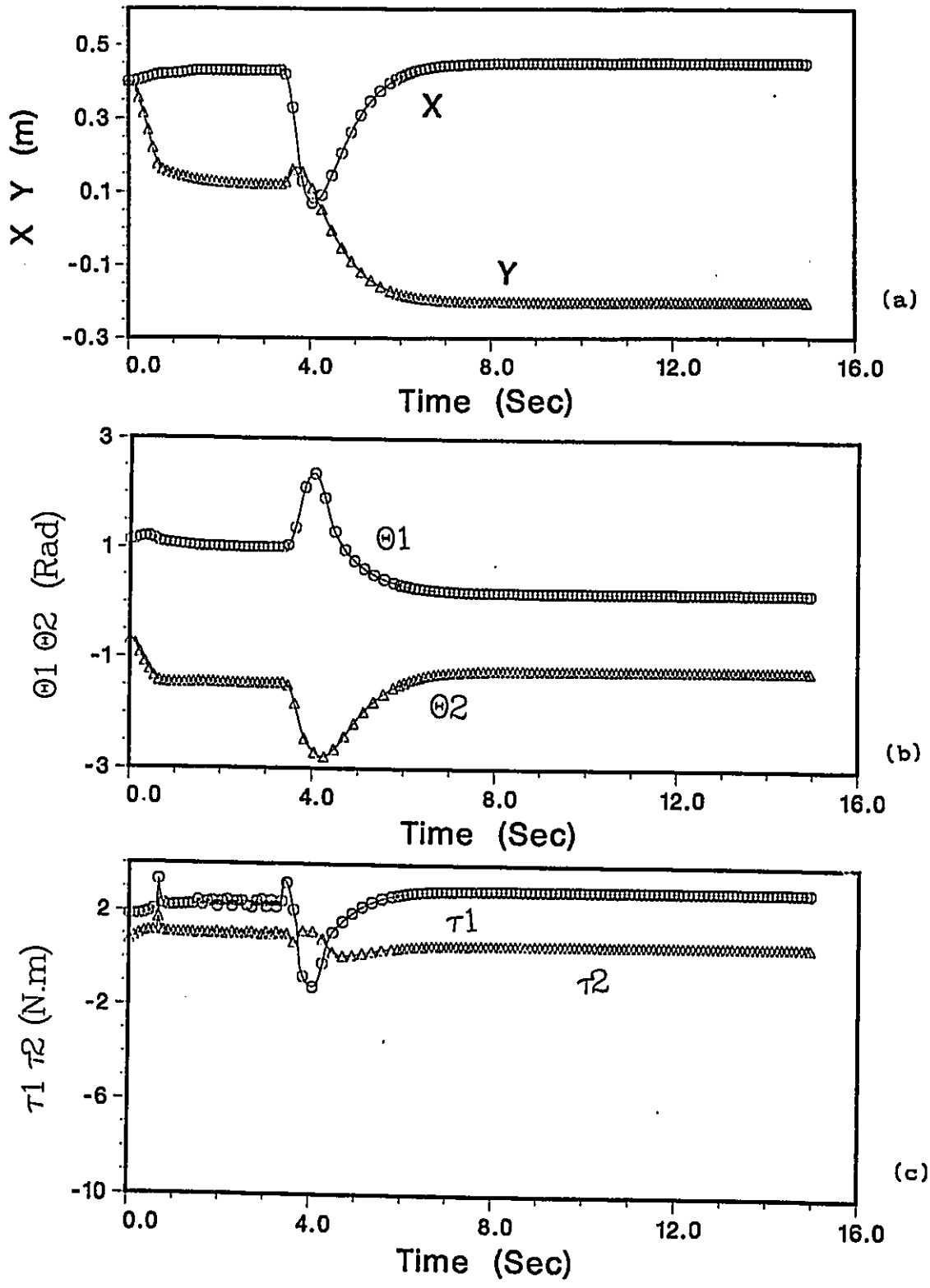


Figure 4.5.3 Simulation results of the robot shown in figure 4.5.2
(a) End-effector position
(b) Joints angular positions
(c) Actuator torques

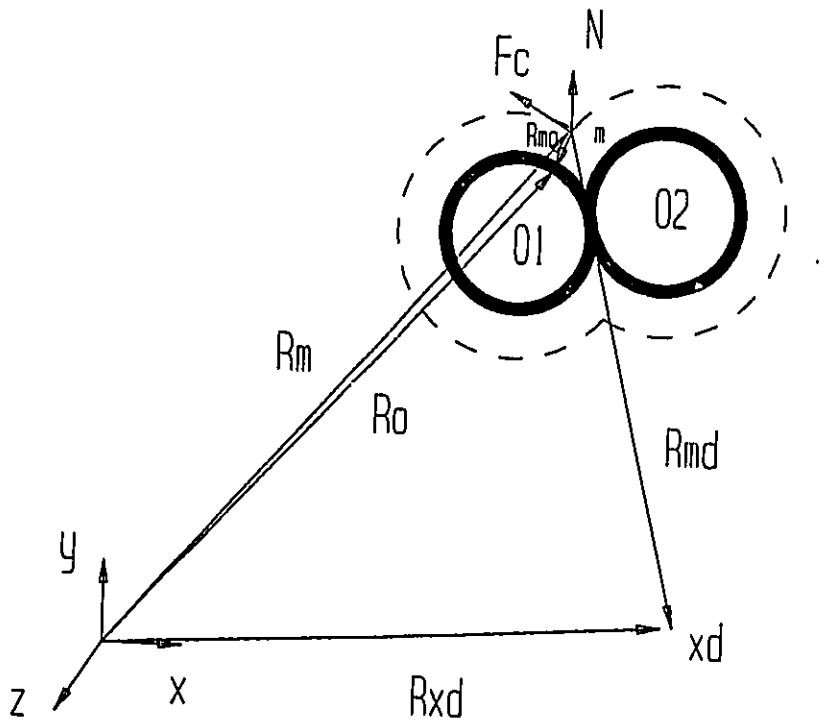


Figure 4.5.4 Surface Navigation Scheme

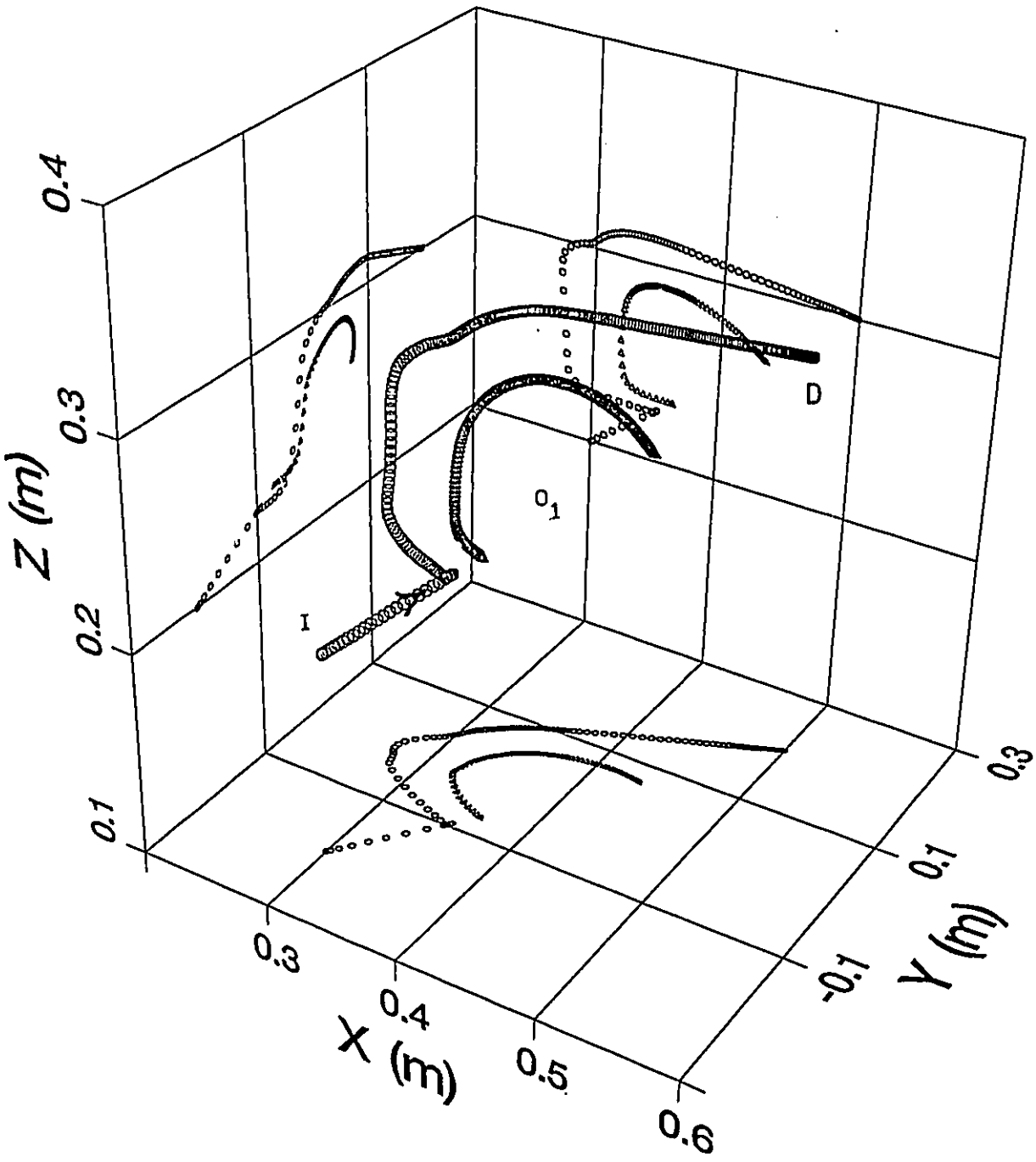


Figure 4.5.5 A first example of the Surface Navigation Scheme
(K=4 N/m, B=4 N/(m/s), R=0.05 m, W=0.0,
K_o=5000 N/m, B_o=10 N/(m/s), K_c=1 N/m
$(X,Y,Z)_s$=(0.4,0.0,0.25) m, R_s=0.08 m)

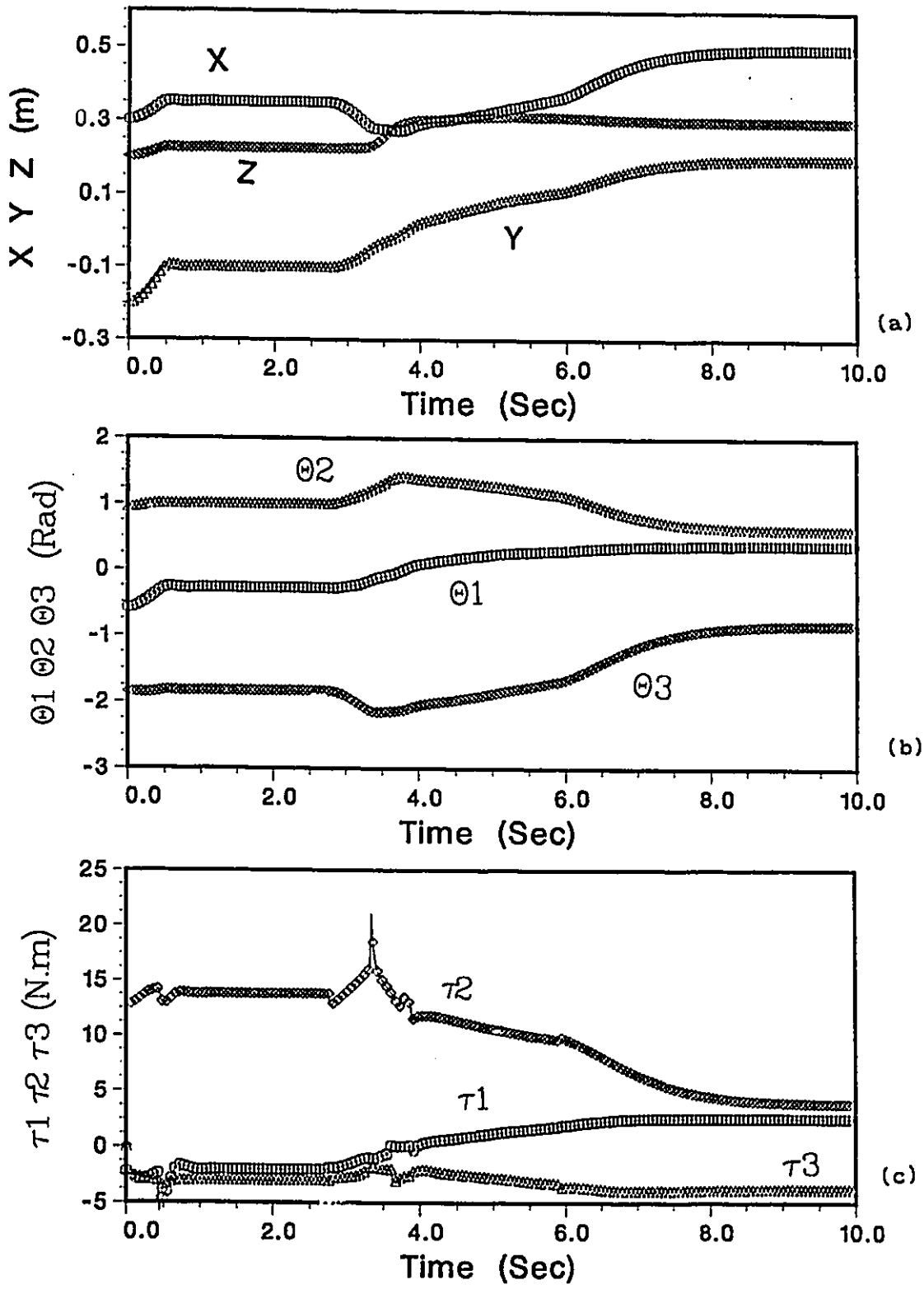


Figure 4.5.6 Simulation results of the robot shown in figure 4.5.5
(a) End-effector position
(b) Joints angular positions
(c) Actuator torques

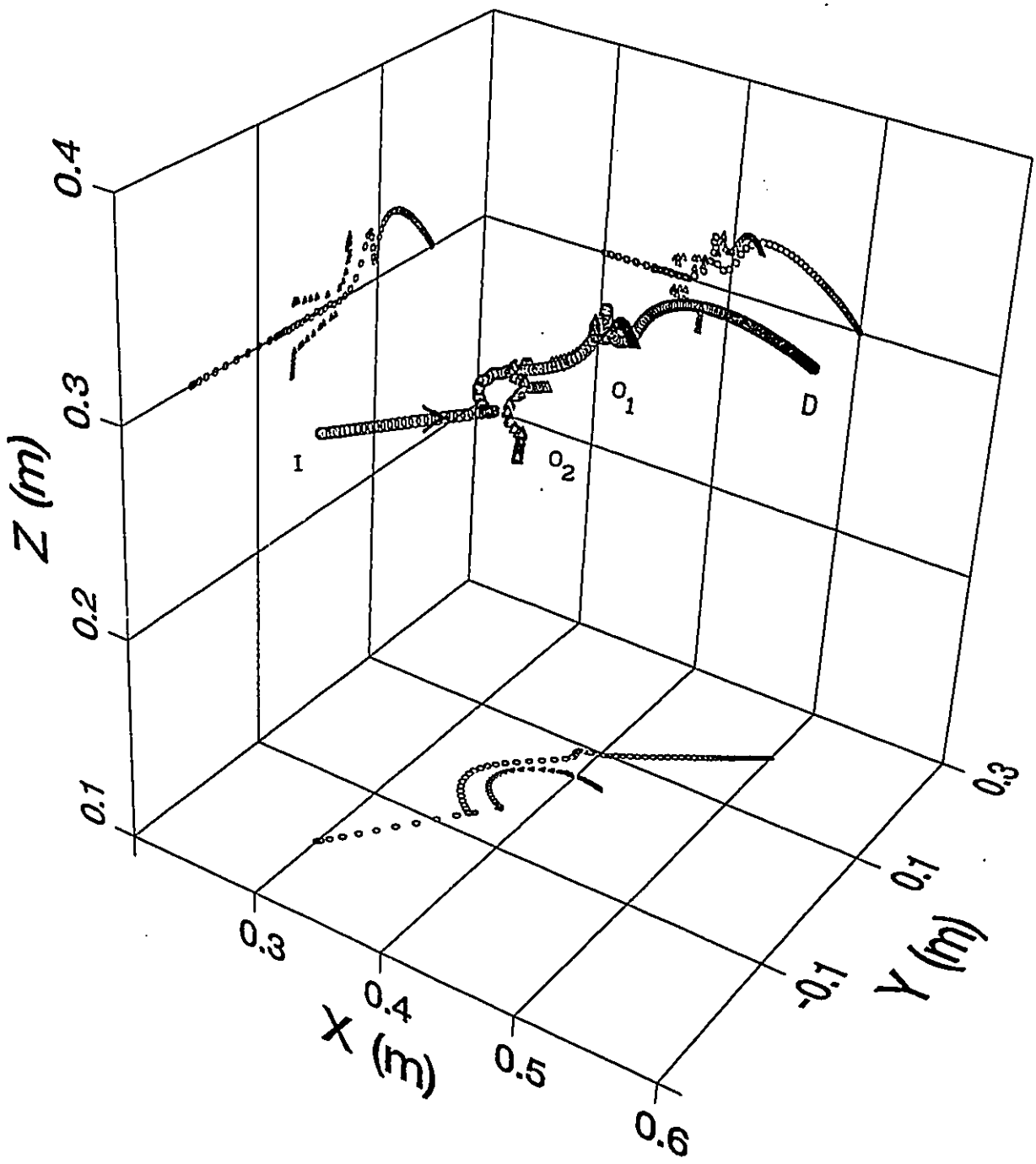


Figure 4.5.7 A second example of Surface Navigation Scheme
(K=4 N/m, B=4 N/(m/s), R=0.05 m, W=0.0,
K_o=5000 N/m, B_o=10 N/(m/s), K_c=1 N/m,
K_Q=10 N/(rad/s), $(X,Y,Z)_{s1}$=(0.4,0.0,0.33)m
$(X,Y,Z)_{s2}$=(0.4,0.0,0.27) m, $R_{s1}=R_{s2}$=0.05m)

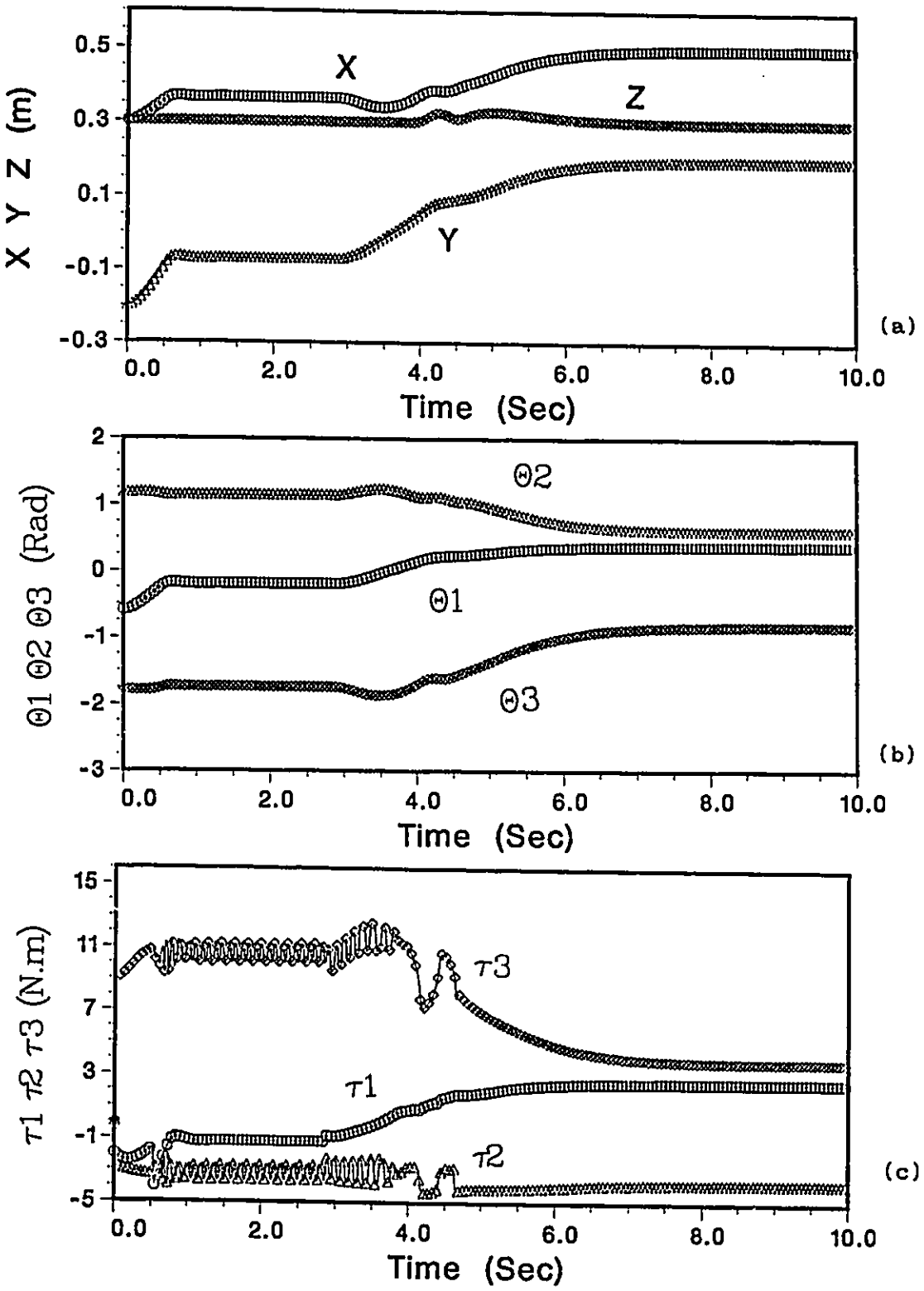


Figure 4.5.8 Simulation results of the robot shown in figure 4.5.7
(a) End-effector position
(b) Joints angular positions
(c) Actuator torques

4.6 WORK VOLUME LIMITS

In some cases, the resultant of the repulsive and attractive forces could direct the robot toward a point located outside of the work volume in order to avoid an obstacle. Figure 4.5.3 shows that the robot would have to exceed the limits of the joints in order to avoid obstacles. By introducing artificial damping at joint limits, the controller leads to the avoidance of such a result and results in reaching the target, located in the workvolume, as shown in the figure 4.6.1 .

Figure 4.6.2 shows the end-effector position (a), joint position (b) and joint actuator torques (c) for the case shown in figure 4.6.1 .

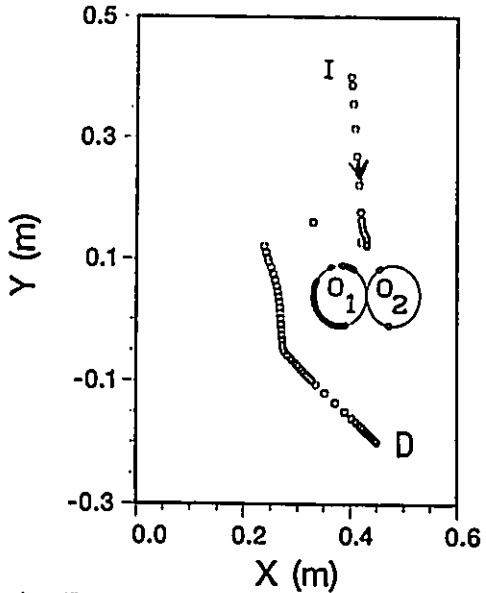


Figure 4.6.1 End-effector trajectory in work volume limits
(same parameters as figure 4.5.2, except K_Q=2 N/(rad/s))

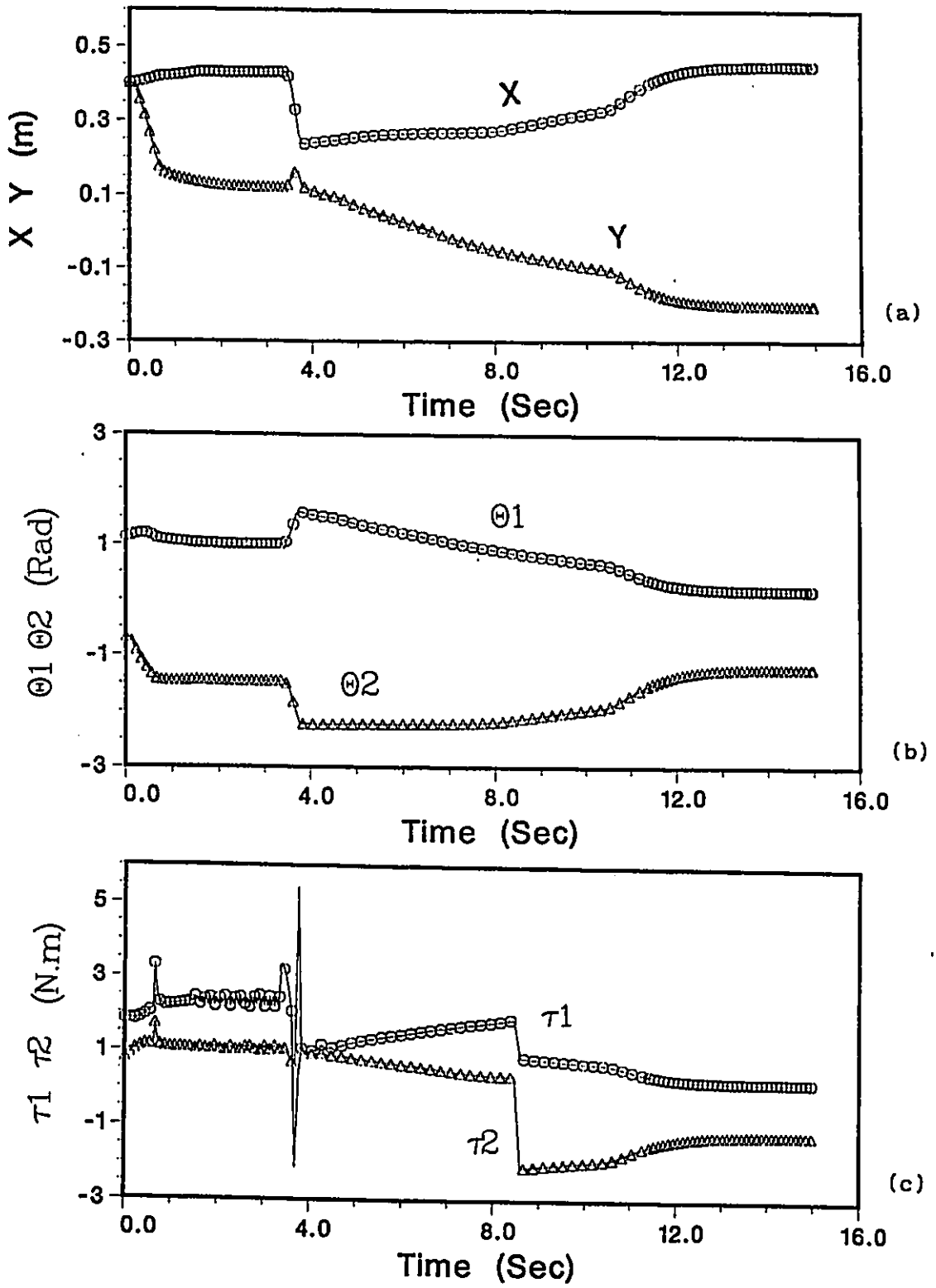


Figure 4.6.2 Simulation results of the robot shown in figure 4.6.1
(a) End-effector position
(b) Joints angular positions
(c) Actuator torques

5.0 BEHAVIOR OF DYNAMIC TERMS IN IMPEDANCE CONTROL

In the impedance control systems, the dynamic parameters should be known in order to create a straight line trajectory and inertial, centrifugal, Coriolis, gravity and frictional terms should be decoupled in Cartesian space. Each of these terms were individually tested to see how the robot would behave in case of the absence of their compensation.

First, the actual mass(inertia) term has been replaced by 50% of actual value $\left[M_x(\theta)=50\% \text{ of actual } M_x(\theta)\right]$. The results in Figure 5.0.1 show that the robot comes to a stop at target point after a slight overshoot, still following a quasi-straight line trajectory. For an over estimation of mass properties, for example 150% of actual value, the results from figure 5.0.3 show that the robot would stop at target point without overshooting the target. Also the robot followed a quasi-straight line trajectory. These results indicate that impedance control is robust for those cases.

Shown in figure 5.0.5 are shown the simulation results when centrifugal and Coriolis compensation terms are

missing ($V_x(\theta,\dot{\theta})=0$). As results show, the robot will reach target but not on a straight line path. As the speed of the end-effector increases the centrifugal and Coriolis terms gets more important to the controller. As the speed decreases the controller would generate a trajectory closer to a straight line. The tests proved that centrifugal and Coriolis terms are important, however less important than the inertia term, and the robot can still reach the target even when the compensation of $V_x(\theta,\dot{\theta})$ is not present. Figure 5.0.6 shows the behavior of end-effector position, joint angles and actuator torques of the robot for case shown in figure 5.0.5.

Figure 5.0.7 shows the end-effector trajectory in the absence of gravity term compensator. Results show undesired behavior of the robot as the robot never reaches the target.

Figure 5.0.8 shows the time behavior of the end-effector position, joint angles and actuator torques for the case shown in figure 5.0.7.

Frictional term ($F_x(\theta,\dot{\theta})$) is generally difficult to compensate given that it is less known. Figure 5.0.9 shows the robot trajectory when the compensation term for friction is missing ($F_x(\theta,\dot{\theta}) = 0$). The results show that the target is reached but not in a straight line path. Figure 5.0.10 shows the end-effector position, joint angles and actuator torques for the case shown in figure 5.0.9.

Figure 5.0.11 shows how the robot avoids an obstacle when compensation term $F_x(\theta,\dot{\theta})= 0$. The trajectory is not

smooth but the robot is able to avoid the obstacle and reach the target. By compensating the frictional term, the impedance controller can produce, however, a smoother trajectory, without oscillations.

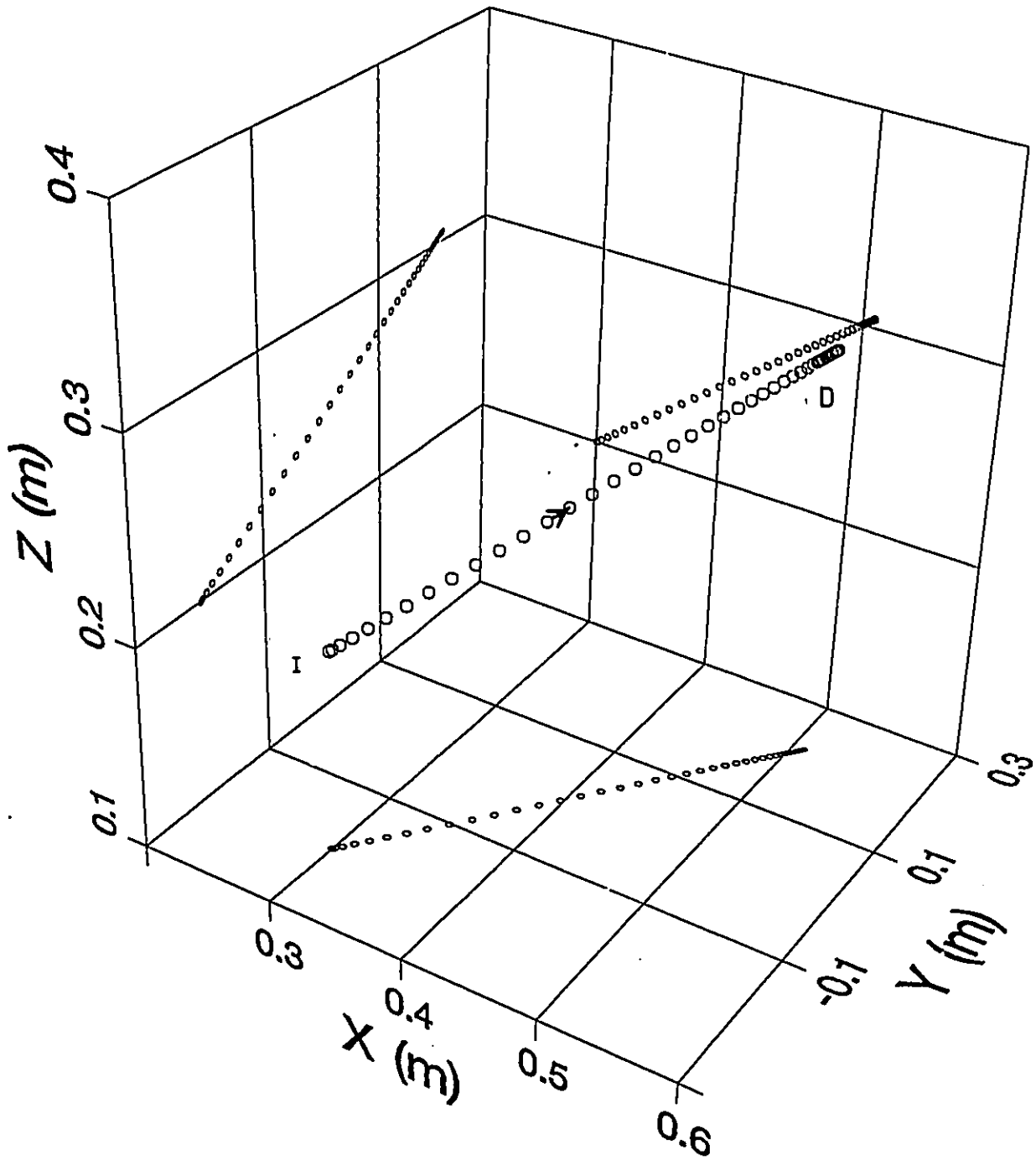


Figure 5.0.1 End-effector trajectory when ($M_x(\theta)$=50% of actual $M_x(\theta)$)
(K=4 N/m, B=4 N/(m/s), W=0.0)

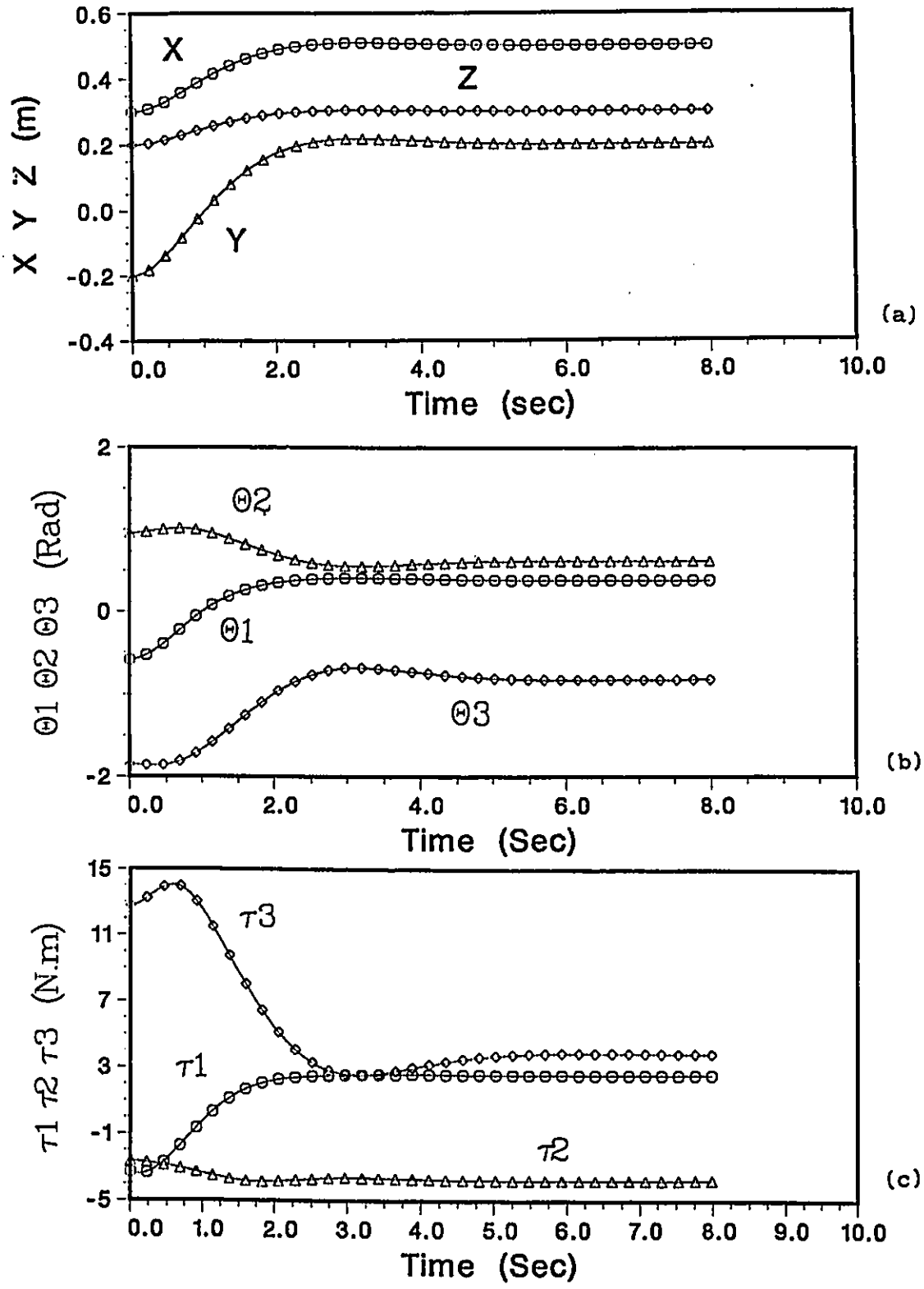


Figure 5.0.2 Simulation results of the robot shown in figure 5.0.1
(a) End-effector position
(b) Joints angular positions
(c) Actuator torques

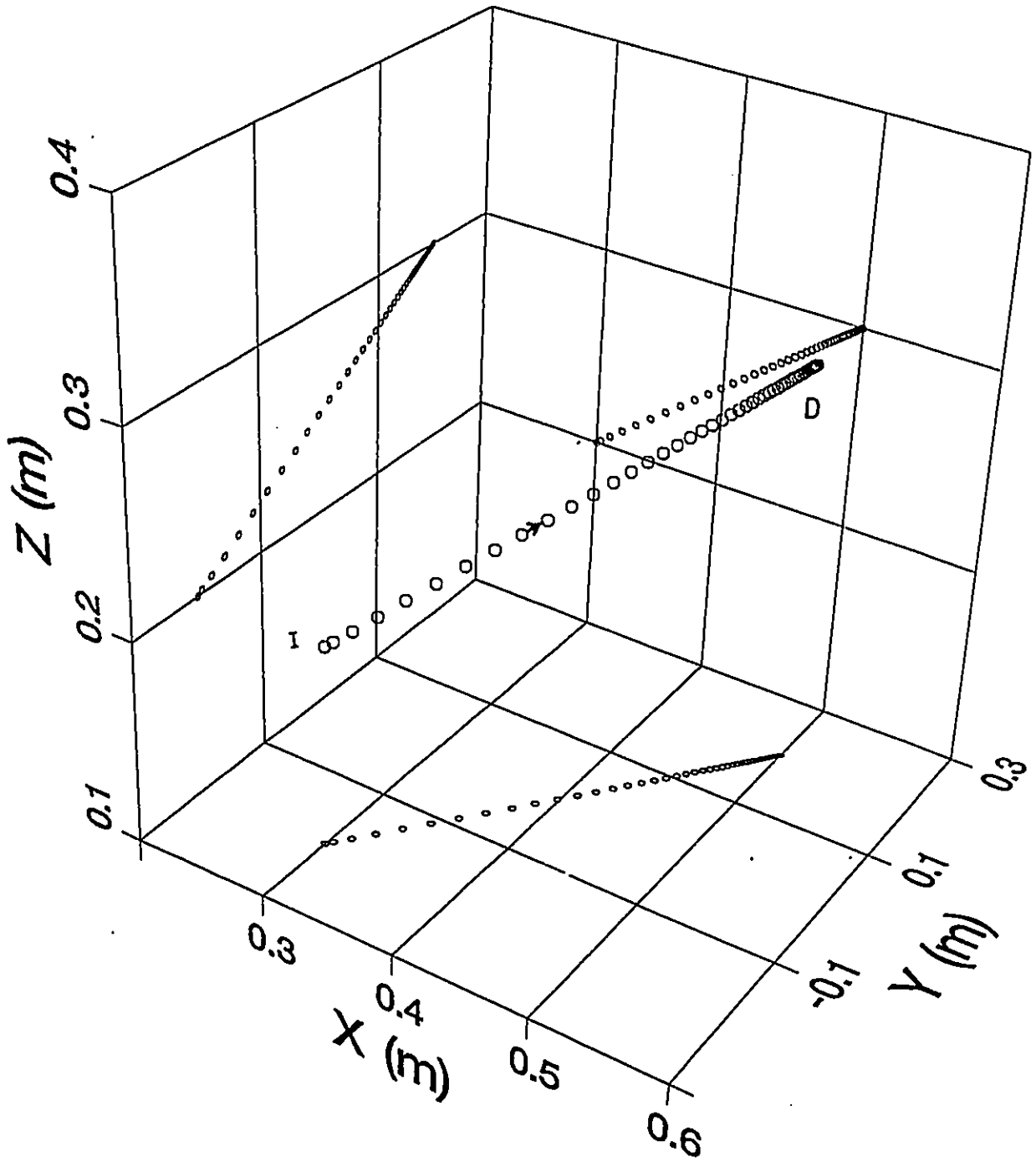


Figure 5.0.3 End-effector trajectory when
($M_x(\theta)$=150% of actual $M_x(\theta)$)
(K=4 N/m, B=4 N/(m/s), W=0.0)

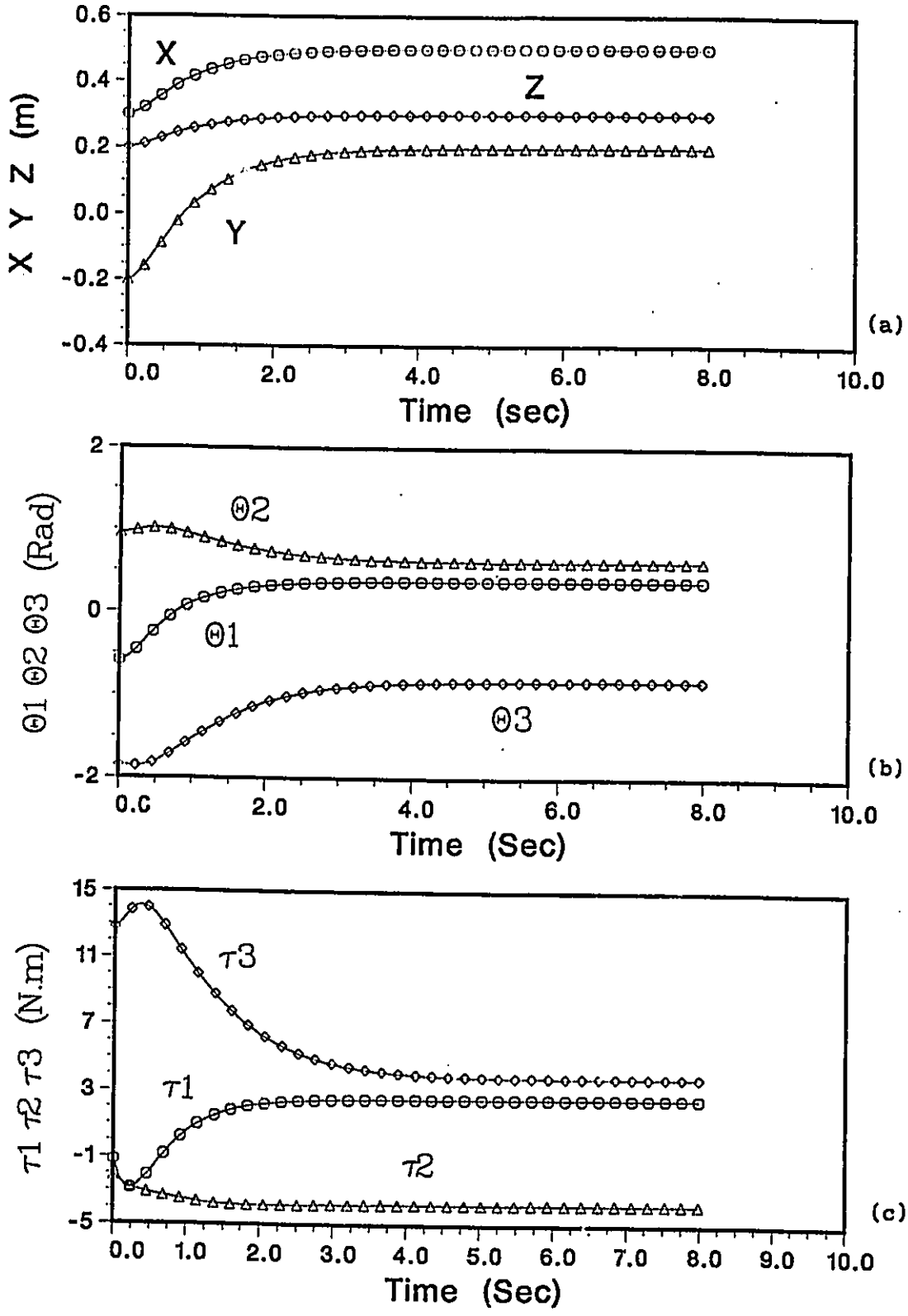


Figure 5.0.4 Simulation results of the robot shown in figure 5.0.3
(a) End-effector position
(b) Joints angular positions
(c) Actuator torques

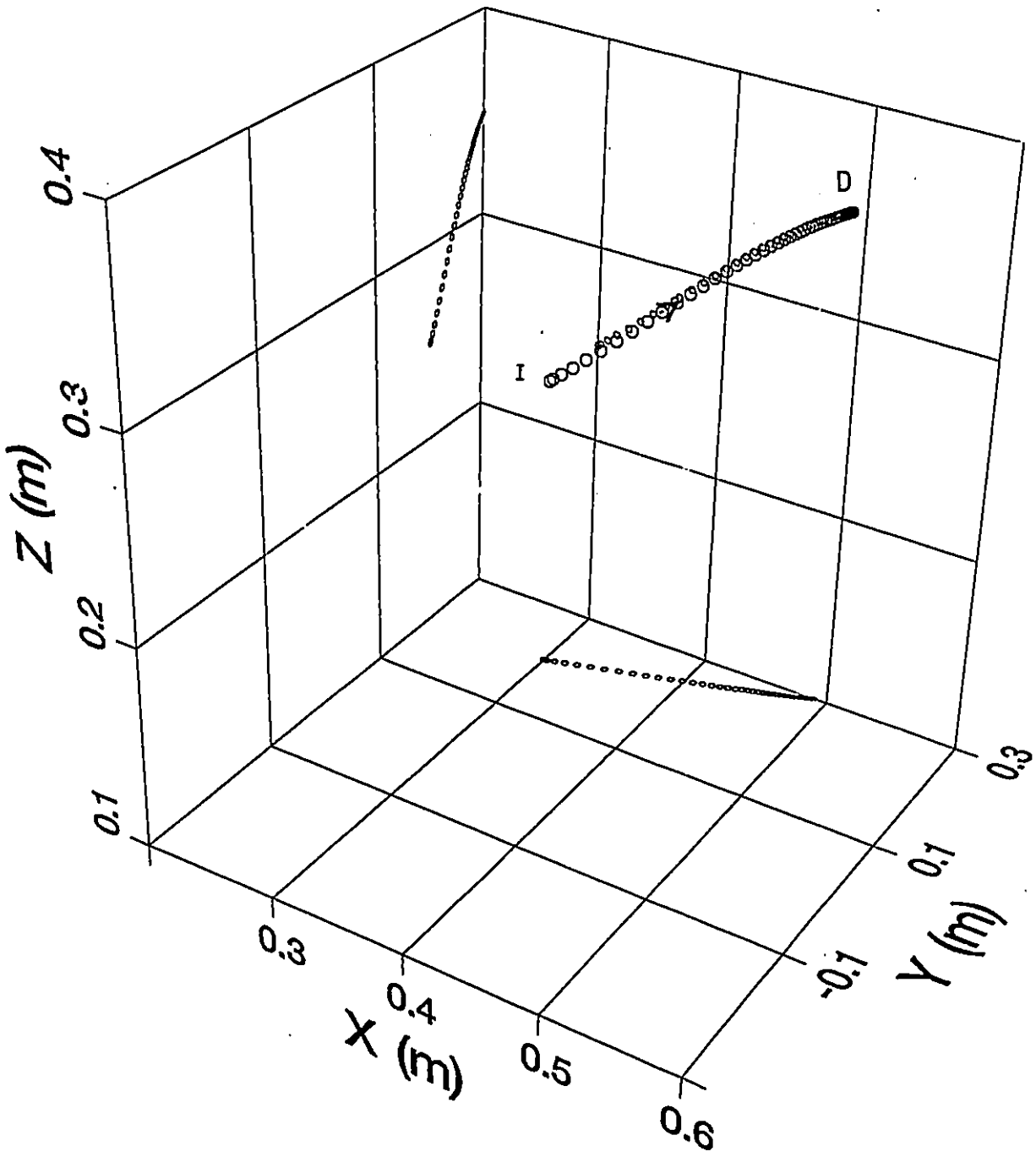


Figure 5.0.5 End-effector trajectory of robot when $V_x(\theta,\dot{\theta}) = 0$
(K=4 N/m, B=4 N/(m/s), W=0.0, K_Q=5 N/(rad/s))

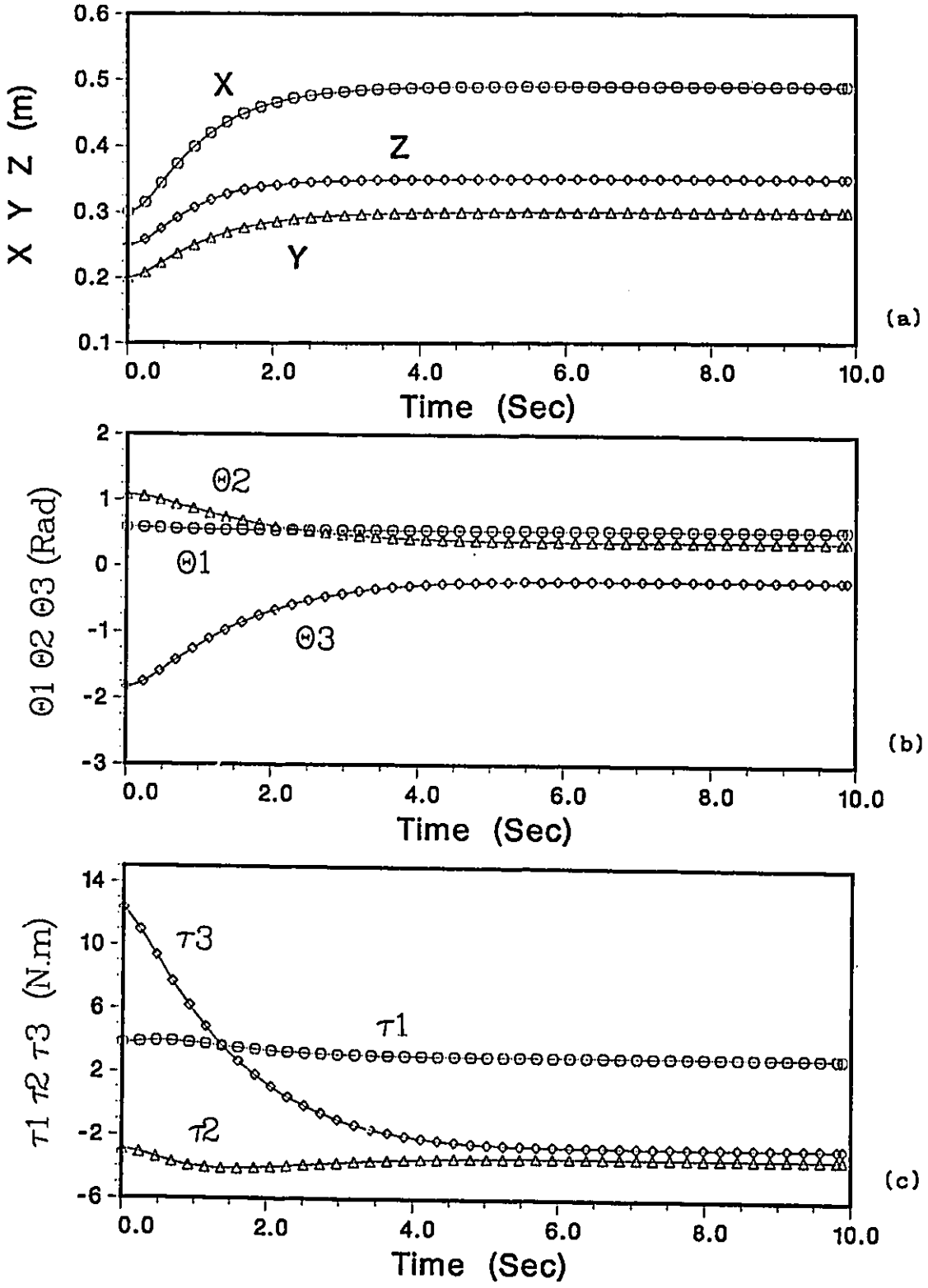


Figure 5.0.6 Simulation results of the robot shown in figure 5.0.5
(a) End-effector position
(b) Joints angular positions
(c) Actuator torques

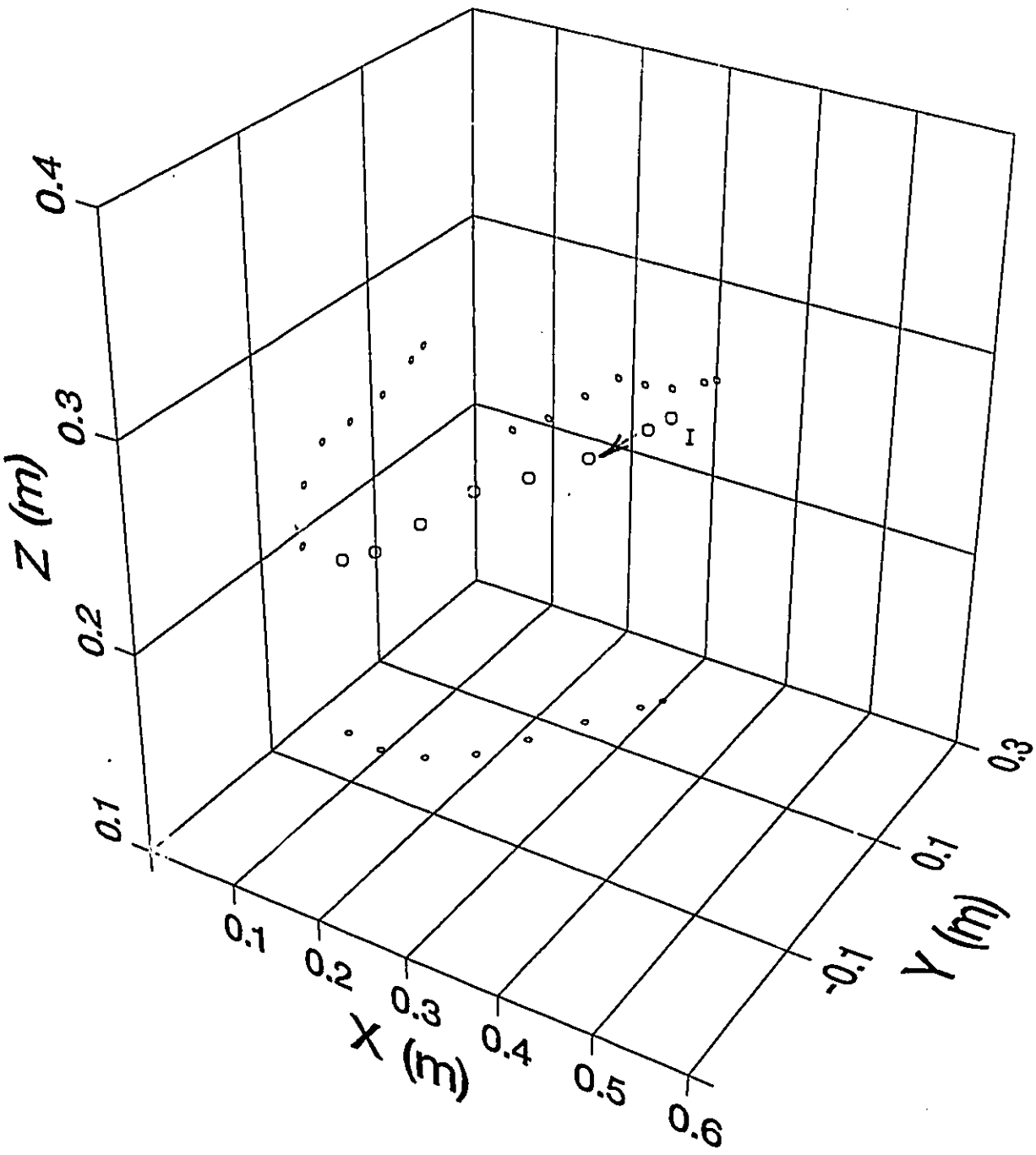


Figure 5.0.7 End-effector trajectory of the robot when $G_x(\theta) = 0$
(K=4 N/m, B=4 N/(m/s), K_Q=5 N/(rad/s)
W=0, $(X,Y,Z)_D$= (0.5, 0.3, 0.35) m,
$(X,Y,Z)_M$=(0.3, 0.2, 0.25) m)

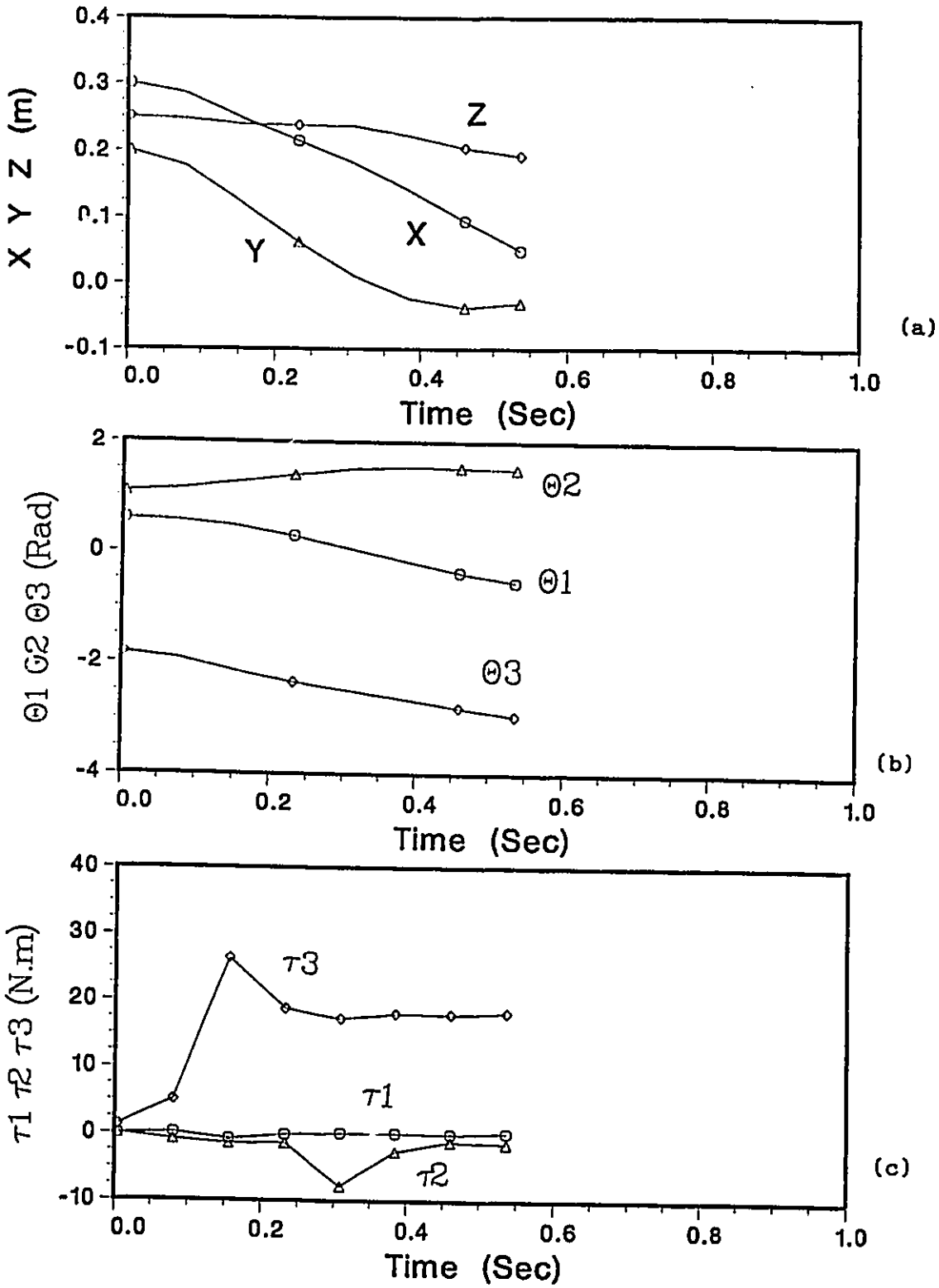


Figure 5.0.8 Simulation results of the robot shown in figure 5.0.7
(a) End-effector position
(b) Joints angular positions
(c) Actuator torques

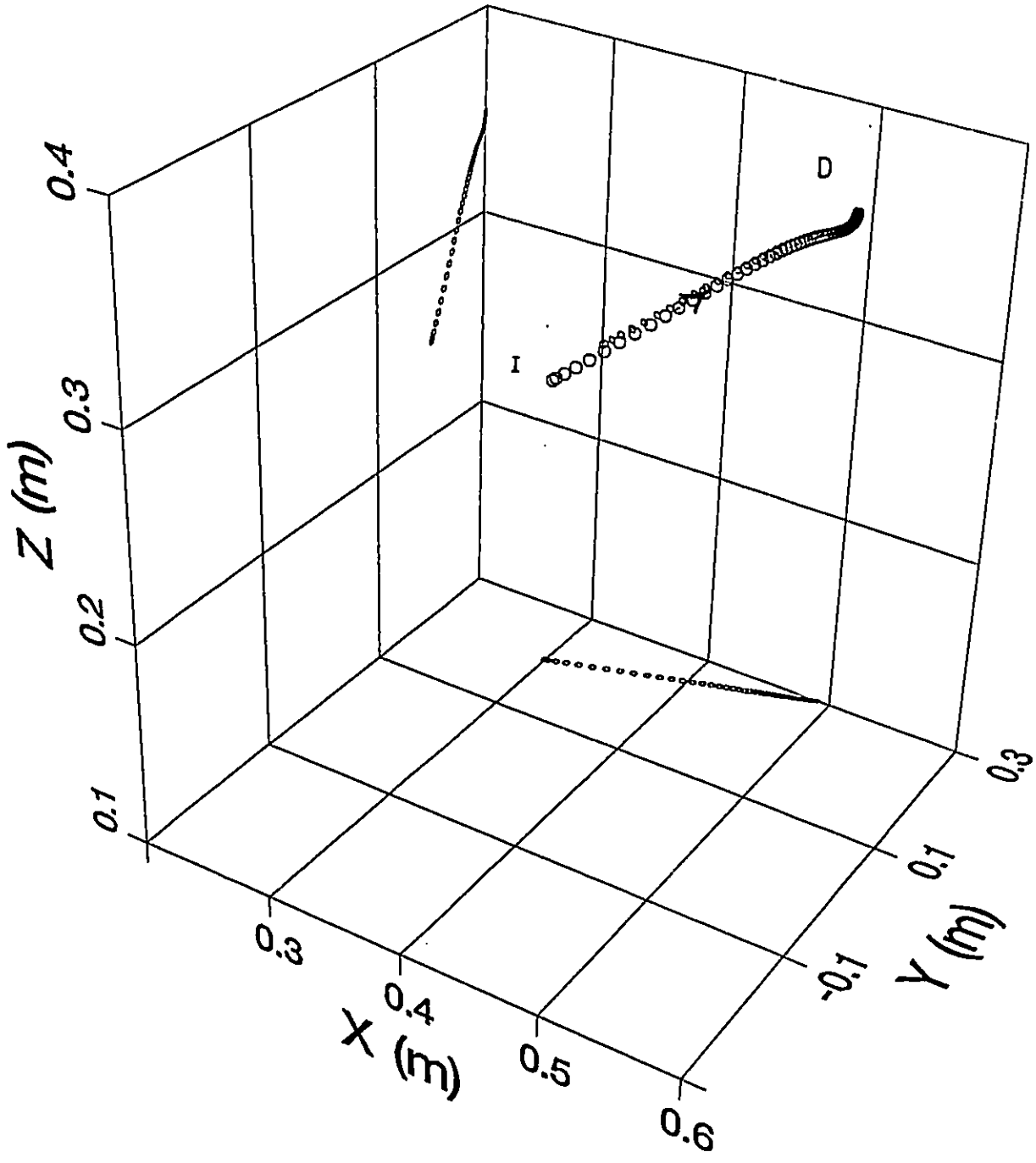


Figure 5.0.9 End-effector trajectory of the robot when $F_x(\theta,\dot{\theta}) = 0$

(K=4 N/m, B=4 N/(m/s), W=0.0, c=0.0, v=0.1 N/(rad/s), K_0=5 N/(rad/s))

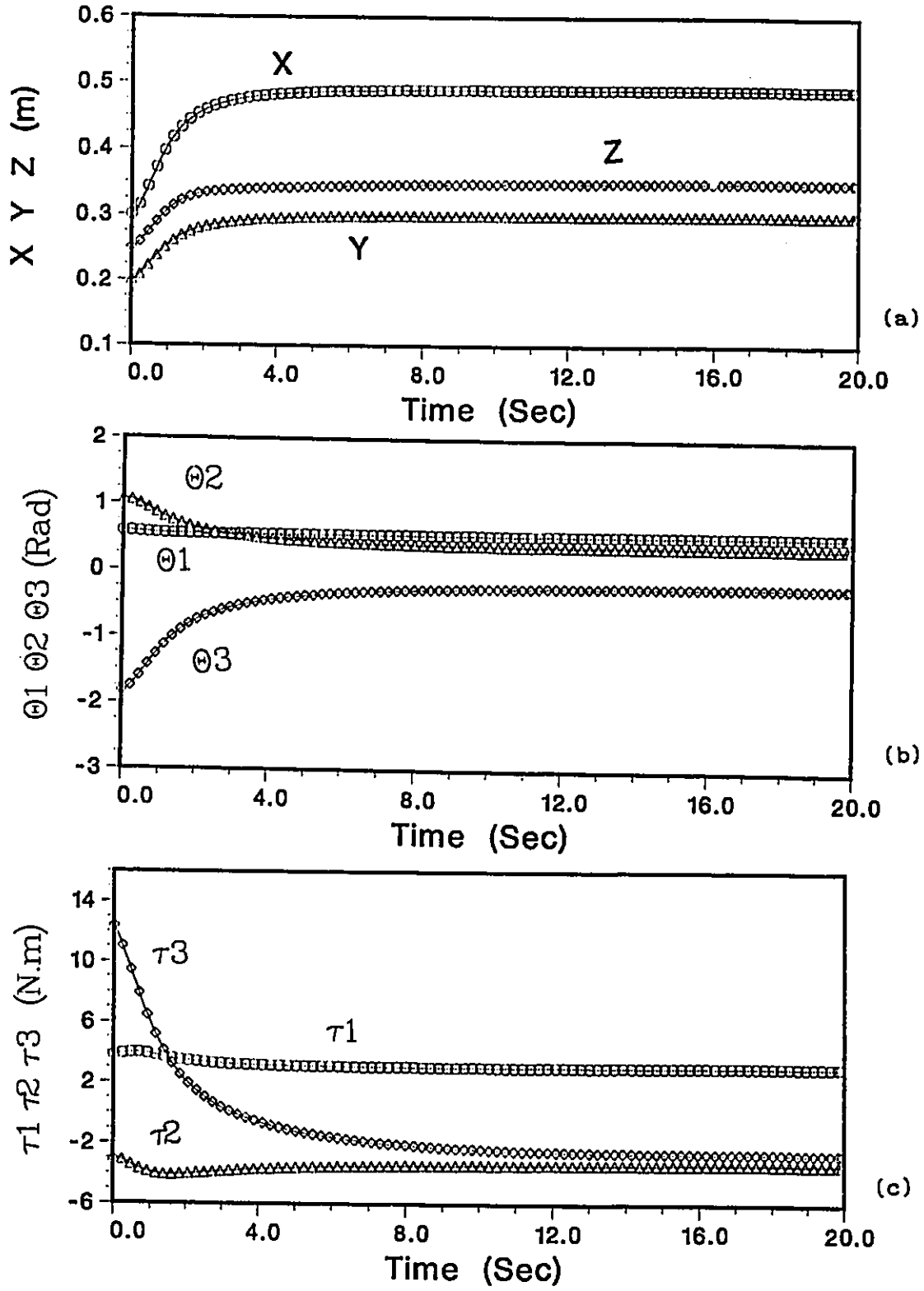


Figure 5.0.10 Simulation results of the robot shown in figure 5.0.9
(a) End-effector position
(b) Joints angular positions
(c) Actuator torques

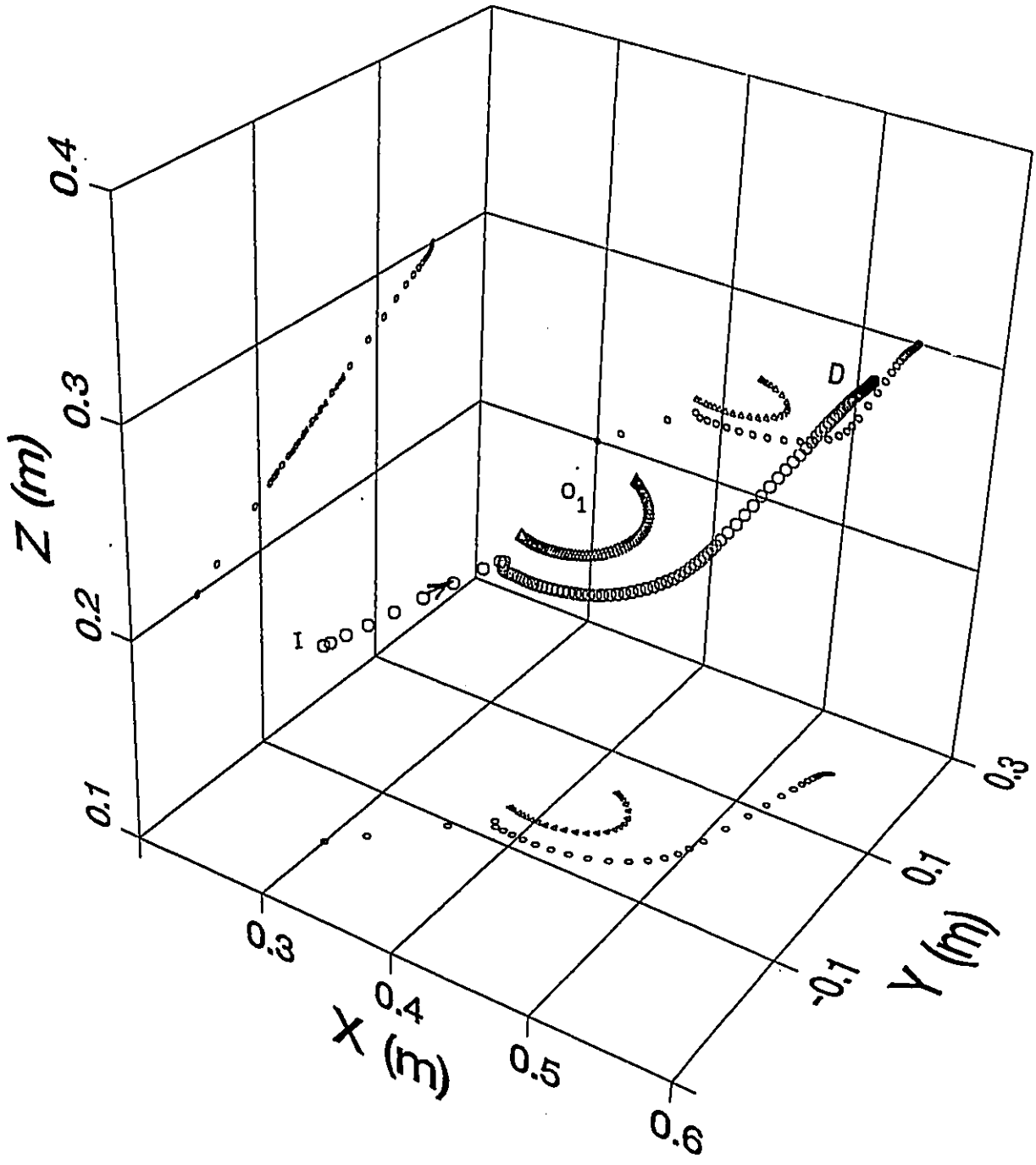


Figure 5.0.11 End-effector trajectory of a robot avoiding an obstacle when F_X=0

(same parameters as figure 4.3.4 plus c=0.05 N and v=0.05 N/(rad/s))

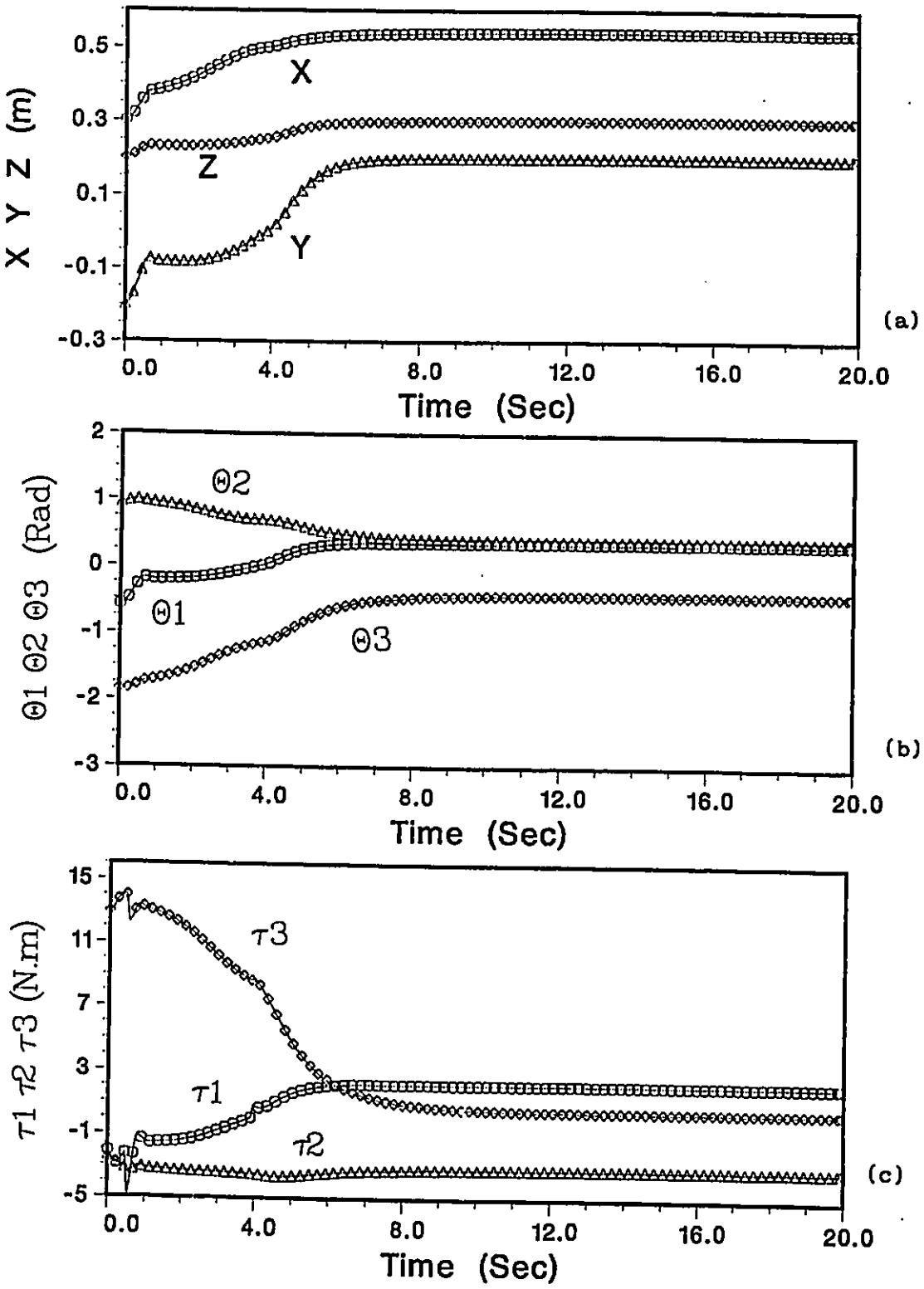


Figure 5.0.12 Simulation results of the robot shown in figure 5.0.11 :
(a) End-effector position
(b) Joints angular positions
(c) Actuator torques

6.0 CONCLUSIONS

This thesis analyses the utilization of impedance control in the generation of robot trajectories and their correction for avoiding collisions with obstacles. The results of the analyses in this thesis prove that impedance control is an important candidate for controlling complex manipulative tasks. Also an impedance based controller proved to be easily expandable for implementing features dealing with collision avoidance, actuator limits and moving target interception.

An Impedance controller can generate on-line trajectories without using inverse kinematics calculations, and can create straight line paths at the expense of increasing real time computational demand.

Impedance control, being a Cartesian based control, can generate on-line trajectory corrections for avoiding obstacles without preplanning or stopping robot motion. The avoidance of obstacles is performed without needing full geometric description of the obstacles which normally

require a complex vision system. The only information needed is the current closest point from the obstacle to the end-effector, provided by a simple proximity sensor.

In this thesis two schemes (Coastal Navigation Scheme(CNS), Surface Navigation Scheme(SNS)) are introduced for finding a path out of local equilibrium traps between obstacles or inside the concavity of an obstacle.

Also a new approach (Cartesian Rescaling Scheme) is used for readjusting the torques of the actuators, when violated, to within the saturation limits. The analysis of the effect of dynamic parameters on the operation of a robot equipped with an impedance controller has shown the relative sensitivity on various parameters and justifies the need for accurate evaluation of the inertial parameters in particular.

For generating a straight line trajectory, all the terms (inertial, centrifugal, Coriolis, gravitational, friction) should be compensated for. As shown by the simulation results, in the absence of centrifugal,Coriolis and friction compensator terms, the controller cannot generate a straight line path, but still the target is reached. In the absence of a gravitational compensator term the system shows undesired behavior. In case of wrong estimation in dynamic parameters the impedance control still directs the robot to the target point. If these estimations are less than actual values the robot would stop at the target after a slight overshoot. If these estimations are

higher than actual values then the robot would reach the target without any overshoot. Further stability studies are required for testing the effect of parametric uncertainty on robot stability.

Sampling time proved to have an important effect on the behavior of the robot system. For the simulations performed, a shorter sampling time led to a better performance, for cases where some terms (centrifugal, Coriolis, frictional) were not compensated for. In general, shorter sampling times. produced trajectories closer to a straight line.

Given the promising results obtained in simulations my recommendation is to continue the research with experimental implementation of an impedance controller on a robot system.

REFERENCES

[1] An,C.H., Hollerbach, J.M.,"Dynamic Stability Issues In Force Control of Manipulators",The Proceedings of the IEEE International Conference on Robotics and Automation,1987,pp 890-896.

[2] Anderson,R.J., Spong,M.W.,"Hybrid Impedance Control of Robotics Manipulators", The Proceedings of the IEEE International Conference on Robotics and Automation , 1987, pp.1073-1080.

[3] Andrews, J.R. and Hogan, N."Impedance Control as a Framework for Implementating Obstacle Avoidance in a Manipulator",Manufacturing Process and Robotics Systems,Eds. Hardt,D. Book, W. ,ASME, 1983,pp 243-251.

[4] Andrews, J.R., "Impedance Control as a Framework for Implementing Obstacle Avoidance in a Manipulator", S.M. Thesis, MIT, February, 1983.

[5] Asada, H.,Slotine, J.J.E.,"Robotic Analysis and Control", John Wiley&Sons,1986, pp 135.

[6] Asada, H.,West, H.,"A Method for the Design of Hybrid Position/Force Controllers for Manipulators Constrained by Contact with the Environment", The Proceedings of the IEEE International Conference on Robotics and Automation , 1985, pp. 251-259.

[7] Craig,J.J.,"Introduction to Robotics", Addison-Wesley, 1986.

[8] Goldenberg, A.A.,"Implementation of Force and Impedance Control in Robot manipulators", The Proceedings of the IEEE International Conference on Robotics and Automation , 1988, pp 1626-1632.

[9] Hill, S.D., Vaccaro, R.J.,"Cartesian Control of Robotic Manipulators with Joint Compliance,"Proceedings of the American Control Conference", Seattle, Wa., June 1986, pp.114-120.

[10] Hogan, N.,"Mechanical Impedance Control in Assistive and Manipulators.", Proceedings of the Joint Automatic Control Conference, San Francisco,vol. 1.,August 1980.

[11] Hogan, N.," Impedance Control: An Approach to

Manipulation", Proceedings, American Control Conference, San Diego, California, June 6-8, 1984, pp. 304-313.

[12] Hogan, N.," Stable Execution of Contact Tasks Using Impedance Control", The Proceedings of the IEEE International Conference on Robotics and Automation Automation, vol. 1, 1987, pp. 1047-1054.

[13] Hogan, N.,"Impedance Control: An Approach to Manipulation: Part. I-Theory, Part. II- Implementation, Part. III- Applications", Journal of Dynamic Systems, Measurement and Control, Trans. of ASME, vol. 107, March 1985.

[14] Kazerooni, H.," Robust Non-Linear Impedance Control for Robot Manipulators", The Proceedings of the IEEE International Conference on Robotics and Automation , 1987, pp. 741-750.

[15] Kazerooni, H.," Automated Roboting Deburing Using Electronic Compliancy: Impedance Control", The Proceedings of the IEEE International Conference on Robotics and Automation , 1987, pp. 1025-1032.

[16] Kazeronni, H., Tsay, T. I., Stability Criteria for Robot Compliant Maneuvers", The Proceedings of the IEEE International Conference on Robotics and Automation, 1988, pp. 1166-1172.

[17] Kazerooni, H., Guo, J.," Direct-Drive, Active Compliant End-Effector", The Proceedings of the IEEE International Conference on Robotics and Automation, 1987, pp. 758-766.

[18] Khatib, O.,"Real-Time Obstacle Avoidance for Robotics Manipulators and Mobile Robots", The Proceedings of the IEEE International Conference on Robotics and Automation, 1985, pp 500-505.

[19] Khatib, O.,"A Unified Approach for Motion and Force Control of Robot Manipulators: The Operational Space Formulation", IEEE Journal of Robotics and Automation", VOL. RA-3, No. 1, Feb. 1987.

[20] Krogh, B.,"A Generalized Potential Field Approach to Obstacle Avoidance Control", SME Conf. Robotics Research: The Next Five Years and Beyond, Bethlehem, Pennsylvania, August, 1984.

[21] Lawrence, D. A.,"Impedance Control Stability Properties in Common Implementations", The Proceeding of the IEEE International Conference on Robotics and Automation, 1988, pp. 1185-1190.

[22] Lozano-Perez, T.,"Spatial Planning: A Configuration Space Approach", IEEE Transactions on Computers",VOL C-32, Feb. 1983,pp 108-120.

[23] Meirovitch,L.,"Methods of Analytical Dynamics", McGrew-Hill,1970.

[24] Necsulescu, D.S., "Critical Review of the Method of Artificial Potential fields and Impedance Control in Robotics", Technical Memorandom, Directorate of Space Mechanics,April 1989.

[25] Necsulescu D.S., Jassemi-Zargani R., Graham W.B.," Impedance Control for Robotic Manipulation", Second workshop on Military Robotics Applications, August 8-11,1989,at RMC, Kingston,Ontario.

[26] Necsulescu D.S., Jassemi-Zargani R., Graham W.B.," The Methods of Artificial Potential Fields and Impedance Control in Robotics", CASI Symposium on Space Station, Nov.8-9,1989,Ottawa,Canada.

[27] Newman, W.S. and Hogan, N.," High Speed Control and Obstacle Avoidance using Dynamic Potential Functions", The Proceedings of the IEEE International Conference on Robotics and Automation 1987, pp. 14-24.

[28] Salisbury, J.K.,"Active Stiffness Control of a Manipulator in Cartesian Coordinates",IEEE Conference on Decision and Control, New Mexico, 1980.

[29] Takegaki,M.,Arimoto,S., A New Feedback Method for Dynamic Control of Manipulators,Journal of Dynamic Systems, Measurement and Control, VOL 102,June 1989,pp 119-125.

[30] Whitney,D.E.,"Historical Perspective and State of the Art in Robot Force Control", The Proceedings of the IEEE International Conference on Robotics and Automation , 1985, pp.262-268.

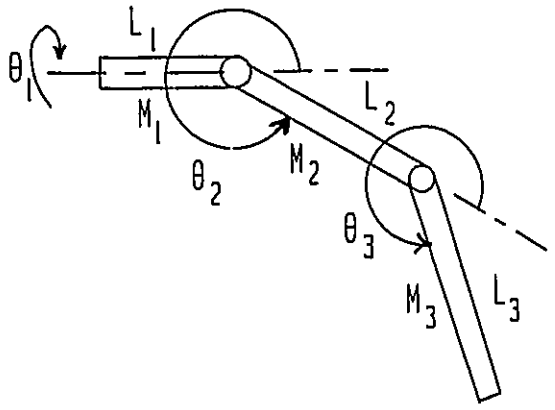
APPENDIX A

APPENDIX (A)

DYNAMICS AND KINEMATICS EQUATIONS OF ROBOT

The simulations in this thesis were performed for a two Degree of Freedom(DOF) planar robot and a 3 DOF revolute robot. The Kinematics, Inverse Kinematics and Dynamics equations of each robot are documented in this appendix.

A.1 THREE DEGREE OF FREEDOM ROBOT (3 DOF)



A.1.1 KINEMATICS:

To find the general form of the transformation which relates the frames attached to neighbouring links, first we should defined the link parameters in terms of link frames. The definition of link parameters follows Denavit-Hartenberg convention

a_i = the distance from z_i to z_{i+1} measured along x_i

α_i = the angle between z_i and z_{i+1} measured about x_i

d_i = the distance from x_{i-1} to x_i measured along z_i

θ_i = the angle between x_{i-1} to x_i measured about z_i

$C_1 = \cos(\theta_1)$

$S_1 = \sin(\theta_1)$

$C_{12} = \cos(\theta_1+\theta_2)$

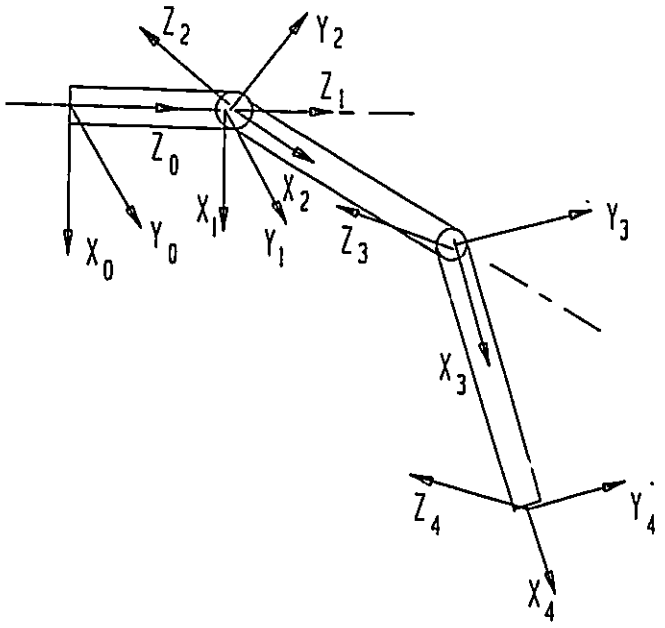
$S_{12} = \sin(\theta_1+\theta_2)$

The complete transformation of link i with respect to link i-1 or joint i with respect to joint i-1 is given by:

$$ {}^{i-1}_{i}T = \begin{bmatrix} C\theta_i & -S\theta_i & 0 & a_i \\ S\theta_i C\alpha_{i-1} & C\theta_i C\alpha_{i-1} & -S\alpha_{i-1} & -S\alpha_{i-1} d_i \\ S\theta_i S\alpha_{i-1} & C\theta_i S\alpha_{i-1} & C\alpha_{i-1} & C\alpha_{i-1} d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (A.1.1.1) $$

The combined transformation that relates directly the frame {N} to the frame {0} is given by:

$$ {}^{0}_{N}T = {}^{0}_{1}T \; {}^{1}_{2}T \; \ldots\ldots \; {}^{N-1}_{N}T \qquad (A.1.1.2) $$



The link parameters of 3 DOF robot are:

i	α_{i-1}	a_i	d_i	θ_i
1	0	0	L_1	θ_1
2	90	0	0	θ_2
3	0	L_2	0	θ_3
4	0	L_3	0	0

The transformation matrices are given by:

$$ {}^{0}_{1}T = \begin{bmatrix} C_1 & -S_1 & 0 & 0 \\ S_1 & S_1 & 0 & 0 \\ 0 & 0 & 1 & L_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad {}^{1}_{2}T = \begin{bmatrix} C_2 & -S_2 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ S_2 & C_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$ {}^{2}_{3}T = \begin{bmatrix} C_3 & -S_3 & 0 & L_2 \\ S_3 & C_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad {}^{3}_{4}T = \begin{bmatrix} 1 & 0 & 0 & L_3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

The resulting transformation matrix between frame 4 and frame 0 is:

$$ {}^{0}_{4}T = \begin{bmatrix} C_1C_{23} & -C_1S_{23} & S_1 & L_3C_1C_{23}+L_2C_1C_2 \\ S_1C_{23} & -S_1S_{23} & -C_1 & L_3S_1C_{23}+L_2S_1C_2 \\ S_{23} & C_{23} & 0 & L_3S_{23}+L_2S_2+L_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (A.1.1.3) $$

The kinematics equations are:

$$ \begin{cases} X = L_3C_1C_{23}+L_2C_1C_2 & (A.1.1.4) \\ Y = L_3S_1C_{23}+L_2S_1C_2 & (A.1.1.5) \\ Z = L_3S_{23}+L_2S_2+L_1 & (A.1.1.6) \end{cases} $$

A.1.2 INVERSE KINEMATICS:

The algebraic solution method used for inverse kinematics is obtained [7]:

from equation A.1.1.4 :

$$C_1 = X\ (\ L_3C_{23}+L_2C_2)^{-1}$$

$$S_1 = Y\ (\ L_3C_{23}+L_2C_2)^{-1}$$

$$\theta_1 = \text{ATAN2}\ (S_1, C_i)$$

OR

$$\theta_1 = \text{ATAN2}\ (Y,X) \qquad \text{(A.1.2.1)}$$

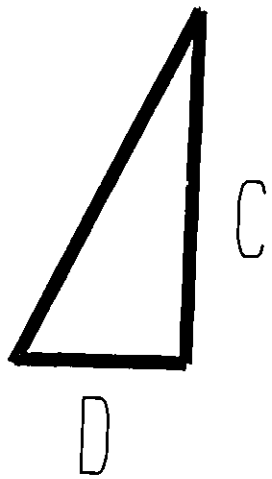
In order to calculate angle θ_3, we square both sides of equations A.1.1.5 and A.1.1.6 and add them:

$$K = \left(\frac{Y}{S_1}\right)^2 + (Z - L_1)^2$$

$$C = \sqrt{(2L_2L_3)^2 - (K-L_2^2-L_3^2)^2}$$

$$D = K - L_3^2 - L_2^2$$

$$\theta_3 = \text{ATAN2}\ (C,D) \qquad \text{(A.1.2.2)}$$



In order to calculate θ_2, from equations A.1.1.536 and A.1.2.2 we used

$$A = (Z-L_1)\, L_3S_3 + \frac{Y}{S_1}\,(L_3C_3+L_2)$$

$$B = (L_3S_3)^2 + (L_3C_3+L_2)^2$$

$$E = \sqrt{B^2 - A^2}$$

$$\theta_2 = \text{ATAN2}\,(E,A) \qquad \text{(A.1.2.3)}$$

A.1.3 DYNAMICS

To compute the torque of each actuator that corresponds to a given trajectory of manipulator, the recursive Newton-Euler algorithm was used. We assume known position, velocity and acceleration of joints($\theta, \dot{\theta}, \ddot{\theta}$), and known the kinematics and dynamics parameters.[7]

Forward dynamic equations are the following: (from frame 1 to frame 4)

- for angular velocity

$$ {}^{i+1}\omega_{i+1} = {}^{i+1}_{i}R \; {}^{i}\omega_{i} + \dot{\theta}_{i+1} \; {}^{i+1}Z_{i+1} \qquad (A.1.3.1) $$

- for angular acceleration

$$ {}^{i+1}\omega_{i+1} = {}^{i+1}_{i}R \; {}^{i}\omega_{i} + {}^{i+1}_{i}R \; {}^{i}\omega_{i} \times \dot{\theta}_{i+1} \; {}^{i+1}Z_{i+1} + \ddot{\theta} \; {}^{i+1}Z_{i+1} \qquad (A.1.3.2) $$

- for linear acceleration of each link

$$ {}^{i+1}\dot{V}_{i+1} = {}^{i+1}_{i}R \; (\; {}^{i}\omega_{i} \times {}^{i}P_{i+1} + {}^{i}\omega_{i} \times (\; {}^{i}\omega_{i} \times {}^{i}P_{i+1} \;) + {}^{i}\dot{V}_{i}) \qquad (A.1.3.3) $$

- for linear acceleration of center of mass

$$ {}^{i+1}Vc_{i+1} = {}^{i+1}\omega_{i+1} \times {}^{i+1}Pc_{i+1} + {}^{i+1}\omega_{i+1} \times ({}^{i+1}\omega_{i+1} \times {}^{i+1}Pc_{i+1}) + {}^{i+1}V_{i+1} \qquad (A.1.3.4) $$

- for effective forces

$$ {}^{i+1}F_{i+1} = m_{i+1} \; {}^{i+1}\dot{Vc}_{i+1} \qquad (A.1.3.5) $$

- for effective couple

$$ {}^{i+1}N_{i+1} = {}^{Ci+1}I_{i+1} \; {}^{i+1}\omega_{i+1} + {}^{i+1}\omega_{i+1} \times {}^{Ci+1}I_{i+1} \; {}^{i+1}\omega_{i+1} \qquad (A.1.3.6) $$

Inverse dynamic equations are given by: (from frame 1 to frame 4)

- for force exerted on link i by link i-1 :

$$ {}^{i}f_{i} = {}^{i}_{i+1}R \; {}^{i+1}f_{i+1} + {}^{i}F_{i} \qquad \text{(A.1.3.7)} $$

- for torque exerted on link i by link i-1

$$ {}^{i}n_{i} = {}^{i}N_{i} + {}^{i}_{i+1}R \; {}^{i+1}n_{i+1} + {}^{i}Pc_{i} \times {}^{i}F_{i} + {}^{i}P_{i+1} \times {}^{i}_{i+1}R \; {}^{i+1}f_{i+1} \qquad \text{(A.1.3.8)} $$

- for torque applied by each actuator:

$$ \tau_{i} = {}^{i}n_{i}^{T} \; {}^{i}Z_{i} \qquad \text{(A.1.3.9)} $$

The following initial conditions were considered:

$$ {}^{1}\omega_{1} = \begin{Bmatrix} 0 \\ 0 \\ \dot{\theta}_{1} \end{Bmatrix} \qquad \dot{\omega}_{o} = 0 $$

$$ V_{o} = 0 \qquad \dot{V}_{o} = \begin{Bmatrix} g \\ 0 \\ 0 \end{Bmatrix} $$

$$ f = \begin{Bmatrix} W \\ 0 \\ 0 \end{Bmatrix} \qquad W = \text{payload} $$

Torque equations for each actuator are:

TORQUE OF JOINT 1:

$$\tau_1 = \ddot{\theta}_1 \left\{ \begin{array}{l} Izz_1 + S_2^2\ Ixx_2 + C_2^2\ Iyy_2 + S_{23}^2\ Ixx_3 - C_{23}^2\ Iyy_3 \\ + \dfrac{L_3^2}{4}\ C_{23}^2\ m_3 + L_2 C_2^2\ \left(\dfrac{m_2}{4} + m_3\right) + L_2 L_3 C_2 C_{23} m_3 \end{array} \right\}$$

$$+\dot{\theta}_1\dot{\theta}_2 \left\{ \begin{array}{l} 2\ C_2 S_2\ (Ixx_2 - Iyy_2) + S_{2233}\ (Ixx_3 - Iyy_3) \\ - \dfrac{L_3^2}{2}\ S_{23} C_{23} m_3 - L_2 L_3 S_{223} m_3 - S_2 C_2 L_2^2 \left(\dfrac{m_2}{2} + 2\ m_2\right) \end{array} \right\}$$

$$+\dot{\theta}_1\ddot{\theta}_3 \left\{ S_{2233}(Ixx_3 - Iyy_3) + \frac{L_3^2}{2} S_{23}(S_2 S_3 + C_2 C_3) - L_2 L_3 S_{23} C_2 m_3 \right\}$$

$$+g \left\{ - \frac{L_3}{2}\ S_1\ m_3\ (S_2 S_3 + C_2 C_3) + L_2 S_1 C_2 \left(\frac{m_2}{2} + m_3\right) \right\}$$

(A.1.3.10)

TORQUE OF JOINT 2:

$$\tau_2 = \ddot{\theta}_2 \left\{ I_{ZZ} + I_{ZZ} + m_3 \left(\frac{L_3^2}{4} + L_2 L_3 C_3 + L_2^2 \right) \right\}$$

$$+ \ddot{\theta}_3 \left\{ I_{ZZ_3} + m_3 \left(\frac{L_3^2}{4} + \frac{L_2 L_3}{2} C_3 \right) \right\}$$

$$+ \dot{\theta}_1^2 \left\{ \begin{array}{l} S_2 C_2 (I_{YY_2} - I_{XX_2}) + S_{23} C_{23} (I_{YY_3} - I_{XX_3}) + L_2^2 C_2 S_2 m_3 \\ + \frac{L_3^2}{4} C_{23} S_{23} m_3 + \frac{L_2^2}{4} S_2 C_2 m_2 + \frac{L_2 L_3}{2} S_{223} m_3 \end{array} \right\}$$

$$+ \dot{\theta}_2 \left(\frac{L_2^2}{4} m_2 \right) + \left(\dot{\theta}_2^2 - (\dot{\theta}_2 + \dot{\theta}_3)^2 \right) \left(\frac{L_2 L_3}{2} S_3 m_3 \right)$$

$$+ g \left[- \frac{L_3}{2} C_1 S_{23} m_3 - L_2 S_2 C_1 \left(m_3 + \frac{m_2}{2} \right) \right] + W L_2 S_3$$

(A.1.3.11)

TORQUE OF JOINT 3:

$$\tau_3 = \ddot{\theta}_2 \left[I_{ZZ_3} + \left(\frac{L_3}{2} + L_2 C_3 \right) m_3 \right] + \ddot{\theta}_3 \left(I_{ZZ_3} + \frac{L_3}{2} m_3 \right)$$

$$+ \dot{\theta}_1^2 \left(S_{23} C_{23} (I_{YY_3} - I_{XX_3}) + \frac{L_3^2}{4} C_{23} S_{23} m_3 + \frac{L_2 L_3}{2} C_2 S_{23} m_3 \right)$$

$$+ \dot{\theta}_2^2 \left(L_2 S_3 m_3 \right) + g \left(- C_1 S_{23} m_3 \right)$$

(A.1.3.12)

The general form of the torque equation is:

$$\tau = M(\theta)\ddot{\theta} + V(\theta,\dot{\theta}) + G(\theta) \qquad \text{(A.1.3.13)}$$

where:

INERTIA TERM (M(θ)) IS:

$$M(\theta)=\begin{bmatrix} Izz_1+S_2^2 Ixx_2+C_2^2 Iyy_2 & & \\ +S_{23}^2 Ixx_3-C_{23}^2 Iyy_3+\frac{L_3^2}{4}C_{23}^2 m_3 & 0 & 0 \\ +L_2C_2^2(\frac{m_2}{4}+m_3)+L_2L_3C_2C_{23}m_3 & & \\ 0 & Izz_2+Izz_3+\frac{L_3^2}{4}m_3+L_2L_3C_3m_3 & Izz_3+\frac{L_3^2}{4}m_3+\frac{L_2L_3}{2}C_3m_3 \\ 0 & Izz_3+(\frac{L_3}{2}+L_2C_3)m_3 & Izz_3+\frac{L_3}{2}m_3 \end{bmatrix}$$

(A.1.3.14)

THE CENTRIFUGAL AND CORIOLIS TERM:

$V(\theta,\dot{\theta}) =$

$$\begin{bmatrix} \begin{bmatrix} \dot{\theta}_1\dot{\theta}_2\left\{2C_2S_2(I_{XX_2}-I_{YY_2})+S_{2233}(I_{XX_3}-I_{YY_3})-\frac{L_3^2}{2}S_{23}C_{23}m_3\right. \\ \left. -L_2L_3S_{223}m_3-S_2C_2L_2^2\left(\frac{m_2}{2}+2m_3\right)\right\} + \\ \dot{\theta}_1\ddot{\theta}_3\left\{S_{2233}(I_{XX_3}-I_{YY_3})+\frac{L_3^2}{2}S_{23}(S_2S_3+C_2C_3)-L_2L_3S_{23}C_2m_3\right\} \end{bmatrix} \\ \begin{bmatrix} \dot{\theta}_1^2\left\{S_2C_2(I_{YY_2}-I_{XX_2})+S_{23}C_{23}(I_{YY_3}-I_{XX_3})+\frac{L_3^2}{4}C_{23}S_{23}m_3+\right. \\ \left. \frac{L_2^2}{4}S_2C_2m_2+\frac{L_2L_3}{2}S_{223}m_3+L_2C_2S_2m_3\right\}+\dot{\theta}_2\left(\frac{L_2^2}{4}m_2\right)+ \\ (\dot{\theta}_2^2-(\dot{\theta}_2+\dot{\theta}_3)^2)\left(\frac{L_2L_3}{2}S_3m_3\right) \end{bmatrix} \\ \begin{bmatrix} \dot{\theta}_1^2\left\{S_{23}C_{23}(I_{YY_3}-I_{XX_3})+\frac{L_3^2}{4}C_{23}S_{23}m_3+\frac{L_2L_3}{2}C_2S_{23}m_3\right\} \\ +\dot{\theta}_2^2(L_2S_3m_3) \end{bmatrix} \end{bmatrix}$$

(A.1.3.15)

THE GRAVITY TERM IS:

$$G(\theta)=\begin{Bmatrix} g\left[-\frac{L_3}{2} S_1 m_3 (S_2 S_3 + C_2 C_3) + L_2 S_1 C_2 \left(\frac{m_2}{2} + m_3\right)\right] \\ g\left[-\frac{L_3}{2} C_1 S_{23} m_3 - \frac{L_2}{2} S_2 C_1 m_2 - L_2 C_2 S_2 m_3\right] + w\, L_2 S_3 \\ g\left[-C_1\, S_{23}\, m_3\right] \end{Bmatrix} \qquad \text{(A.1.3.16)}$$

THE FRICTION TERM IS:

$$F(\theta,\dot{\theta}) = V\,\dot{\theta} + C\,\mathrm{sgn}(\dot{\theta}) \qquad \text{(A.1.3.17)}$$

where:

V = Viscous friction constant
C = Coulomb friction constant

A.1.4 THE JACOBIAN

The Jacobian matrix was calculated in order to convert end-effector linear position and velocity to angular position and velocity of the joints or vice versa. Transposed Jacobian is used to transform applied forces at end-effector level to joint torques.

The Jacobian matrix is given by:

$$J = \begin{Bmatrix} \frac{\delta X}{\delta \theta_1} & \frac{\delta X}{\delta \theta_2} & \frac{\delta X}{\delta \theta_3} \\ \frac{\delta Y}{\delta \theta_1} & \frac{\delta Y}{\delta \theta_2} & \frac{\delta Y}{\delta \theta_3} \\ \frac{\delta Z}{\delta \theta_1} & \frac{\delta Z}{\delta \theta_2} & \frac{\delta Z}{\delta \theta_3} \end{Bmatrix} \qquad \text{(A.1.4.1)}$$

From the equations A.1.1.4&5&6:

$$X = L_3 C_1 C_{23} + L_2 C_1 C_2$$

$$Y = L_3 S_1 C_{23} + L_2 S_1 C_2$$

$$Z = L_3 S_{23} + L_2 S_2 + L_1$$

we can obtain:

$$J = \begin{Bmatrix} -L_3S_1C_{23}-L_2S_1C_2 & -L_3C_1S_{23}-L_2C_1S_2 & -L_3C_1S_{23} \\ L_3C_{23}C_1+L_2C_1C_2 & -L_3S_1S_{23}-L_2S_1S_2 & -L_3S_1S_{23} \\ 0 & L_3C_{23}+L_2C_2 & L_3C_{23} \end{Bmatrix}$$

(A.1.4.2)

where the relations between positions and velocities are:

$$\bar{X} = KIN\{\theta\} \qquad (A.1.4.3)$$

$$\bar{V} = J(\theta)\,\dot{\theta} \qquad (A.1.4.4)$$

The equation for torque and force is:

$$\tau = J^T(\theta)\ f \qquad (A.1.4.5)$$

A.1.5 DYNAMIC EQUATION TERMS IN CARTESIAN SPACE:

All dynamics terms should be decoupled in Cartesian space in order obtain straight line trajectories when using impedance control. Mass(inertia), centrifugal, Coriolis, gravity and friction terms are computed in Cartesian space [20] as follows:

$$M_x(\theta) = J^{-T}(\theta)\ M(\theta)\ J^{-1}(\theta) \qquad \text{(A.1.5.1)}$$

$$V_x(\theta,\dot{\theta}) = J^{-T}(\theta)\ [\ V(\theta,\dot{\theta}) - M(\theta)\ J^{-1}(\theta)\ \dot{J}(\theta)\ \dot{\theta}\ ] \qquad \text{(A.1.5.2)}$$

$$G_x(\theta) = J^{-T}(\theta)\ G(\theta) \qquad \text{(A.1.5.3)}$$

$$F_x(\theta,\dot{\theta}) = J^{-T}(\theta)\ [\ F(\theta,\dot{\theta}) - M(\theta)\ J^{-1}(\theta)\ \dot{J}(\theta)\ \dot{\theta}\ ] \qquad \text{(A.1.5.4)}$$

where:

$M_x(\theta)$ = inertia term in Cartesian space

$V_x(\theta,\dot{\theta})$ = centrifugal and Coriolis term in Cartesian space

$G_x(\theta)$ = gravity term in Cartesian space

$F_x(\theta,\dot{\theta})$ = friction term in Cartesian space

$J^T(\theta)$ = transposed Jacobian

$J^{-T}(\theta)$ = inverse of the Jacobian

$\dot{J}(\theta)$ = derivative of the Jacobian

The force acting on end-effector is:

$$f = M_x(\theta)\,\ddot{X} + V_x(\theta,\dot{\theta}) + G_x(\theta) + F_x(\dot{\theta}) \quad \text{(A.1.5.5)}$$

The torque equation is given by:

$$\tau = J^T(\theta)\; f$$

A.1.6 DYNAMIC SIMULATIONS:

The simulations were performed by calculating the acceleration from the dynamics model of the manipulator.:

$$\ddot{\theta} = M^{-1}(\theta) [\tau - V(\theta,\dot{\theta}) - G(\theta) - F(\theta,\dot{\theta})] \qquad (A.1.6.1)$$

By applying a numerical integration techniques the equations were integrated. Euler integration scheme was used to integrate forward in time by steps of size Δt.
The initial conditions(t=0) were:

$$\theta(0) = \theta_0$$
$$\dot{\theta}(0) = 0$$
$$\ddot{\theta}(0) = 0$$

$$\dot{\theta}(t+\Delta t) = \dot{\theta}(t) + \ddot{\theta}(t)\ \Delta t \qquad (A.1.6.2)$$

$$\theta(t+\Delta t) = \theta(t) + \dot{\theta}(t)\ \Delta t + \frac{1}{2}\ \ddot{\theta}(t)\ \Delta t^2 \qquad (A.1.6.3)$$

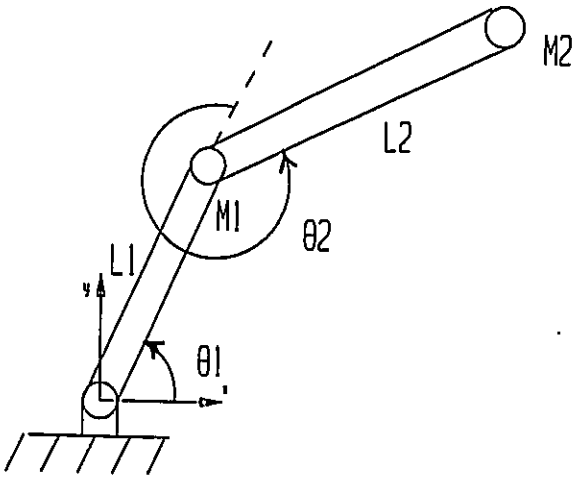
The selection the size of Δt depends on:

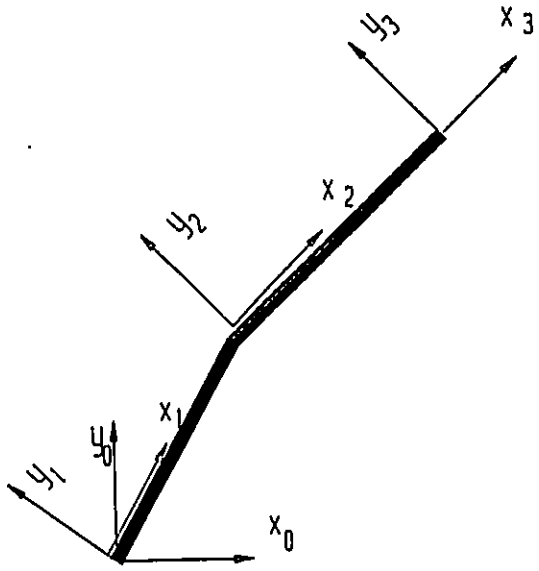
- computing time of computer for each step
- sensor sampling time
- desired computation accuracy

A.2 TWO DEGREE OF FREEDOM PLANNER ROBOT (2 DOF)

A.2.1 The kinematics, inverse kinematics and dynamics equations are the same as for 3-DOF robot calculation.

They are summarized in what follows:





LINK PARAMETERS:

i	α_{i-1}	a_{i-1}	d_{i-1}	θ_i
1	0	0	0	θ_1
2	0	L_1	0	θ_2
3	0	L_2	0	0

TRANSFORMATION MATRIX:

$$T = \begin{Bmatrix} C_{12} & -S_{12} & 0 & L_1C_1 + L_2C_{12} \\ S_{12} & C_{12} & 0 & L_1S_1 + L_2S_{12} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{Bmatrix}$$

(A.2.1.1)

KINEMATICS EQUATIONS:

$$X_1 = L_1 C_1 + L_2 C_{12} \qquad \text{(A.2.1.2)}$$

$$Y_1 = L_1 S_1 + L_2 S_{12} \qquad \text{(A.2.1.3)}$$

INVERSE KINEMATICS:

$$D = X^2 + Y^2 - L_1^2 - L_2^2$$

$$C = \sqrt{4 L_1^2 L_2^2 - D^2}$$

$$\theta_2 = \text{ATAN2 (C,D)} \qquad \text{(A.2.1.4)}$$

$$\theta_1 = \text{ATAN2(Y,X)} - \text{ATAN2(A,B)} \qquad \text{(A.2.1.5)}$$

A.2.2 THE JACOBIAN :

$$J(\theta) = \begin{bmatrix} -L_1 S_1 - L_2 S_{12} & -L_2 S_{12} \\ L_1 C_1 + L_2 C_{12} & L_2 C_{12} \end{bmatrix}$$

(A.2.2.1)

$$J^{-1}(\theta) = \frac{1}{L_1 L_2 S_2} \begin{bmatrix} L_2 C_{12} & L_2 S_{12} \\ -L_1 C_1 - L_2 C_{12} & -L_1 S_1 - L_2 S_{12} \end{bmatrix}$$

(A.2.2.2)

$$J^{T}(\theta) = \begin{bmatrix} -L_1 S_1 - L_2 S_{12} & L_1 C_1 + L_2 C_{12} \\ -L_2 S_{12} & L_2 C_{12} \end{bmatrix}$$

(A.2.2.3)

$$J^{-T}(\theta) = \frac{1}{L_1 L_2 S_2} \begin{bmatrix} L_2 C_{12} & -L_1 C_1 - L_2 C_{12} \\ L_2 S_{12} & -L_1 S_1 - L_2 S_{12} \end{bmatrix}$$

(A.2.2.4)

$$\dot{J}(\theta) = \begin{bmatrix} -L_1 C_1 \dot{\theta}_1 - L_2 C_{12}(\dot{\theta}_1 + \dot{\theta}_2) & -L_2 C_{12}(\dot{\theta}_1 + \dot{\theta}_2) \\ -L_1 S_1 \dot{\theta}_1 - L_2 S_{12}(\dot{\theta}_1 + \dot{\theta}_2) & -L_2 S_{12}(\dot{\theta}_1 + \dot{\theta}_2) \end{bmatrix}$$

(A.2.2.5)

A.2.3 DYNAMICS:

Initial conditions:

$$^{1}\omega_1 = 0 \qquad \omega_0 = 0 \qquad V_0 = 0$$

$$V_0 = \begin{Bmatrix} 0 \\ g \end{Bmatrix} \qquad f = \begin{Bmatrix} 0 \\ \omega \end{Bmatrix}$$

TORQUE EQUATIONS:

$$\tau = M(\theta)\,\ddot{\theta} + V(\theta,\dot{\theta}) + G(\theta)$$

MASS(INERTIA) TERM:

$$M(\theta) = \begin{Bmatrix} L_2^2 m_2 + 2 L_1 L_2 m_2 C_2 + L_1^2 (m_1+m_2) & L_2^2 m_2 + L_1 L_2 m_2 C_2 \\ L_2^2 m_2 + L_1 L_2 m_2 C_2 & L_2^2 m_2 \end{Bmatrix}$$

(A.2.3.1)

CENTRIFUGAL AND CORIOLIS TERM:

$$V(\theta,\dot{\theta}) = \begin{Bmatrix} - m_2 L_1 L_2 S_2 \dot{\theta}_2^2 - 2 m_2 L_1 L_2 S_2 \theta_1 \theta_2 \\ m_2 L_1 L_2 S_2 \dot{\theta}_1^2 \end{Bmatrix}$$

(A.2.3.2)

GRAVITY TERM:

$$G(\theta) = \begin{Bmatrix} m_2 L_2 C_{12} g + (m_1+m_2) L_1 C_1 g \\ m_2 L_2 C_{12} g \end{Bmatrix}$$

(A.2.3.3)

Dynamic terms in Cartesian space:

INERTIA TERM:

$$M_x(\theta) = \begin{Bmatrix} m_2+m_1 \dfrac{C_{12}^2}{S_2^2} & m_1 \dfrac{C_{12}S_{12}}{S_2^2} \\ m_1 \dfrac{S_{12}C_{12}}{S_2^2} & m_2+ m_1 \dfrac{S_{12}^2}{S_2^2} \end{Bmatrix} \qquad \text{(A.2.3.4)}$$

CENTRIFUGAL & CORIOLIS TERM:

$$V_x(\theta,\dot{\theta}) = \frac{1}{S_2^2} \begin{bmatrix} L_1 m_1 C_2 C_{12}\dot{\theta}_1^2 + L_2 m_1 C_{12}(\dot{\theta}_1+\dot{\theta}_2)^2 \\ L_1 m_1 C_2 S_{12}\dot{\theta}_1^2 + L_2 m_1 S_{12}(\dot{\theta}_1+\dot{\theta}_2)^2 \end{bmatrix} \qquad \text{(A.2.3.5)}$$

GRAVITY TERM:

$$G_x(\theta) = \frac{1}{S_2^2} \begin{bmatrix} m_1 g C_1 C_{12} \\ m_1 g C_1 S_{12}+ m_2 g S_2 \end{bmatrix} \qquad \text{(A.2.3.6)}$$

APPENDIX B

APPENDIX (B)

Computer code: Impedance controller for robots

```
C*****************************************
C*            R3DIMP.FOR                 *
C*****************************************
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                         C
C IMPEDANCE CONTROL FOR 3 DOF ROBOT IN    C
C 3 DIMENSIONAL SPACE                     C
C                                         C
C ALSO HAS:  - JOINT LIMITS SCHEME        C
C            - DECOUPLING TERMS           C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
      REAL*4 XD,YD,ZD,QD1,QD2,QD3,XM,YM,ZM,QM1,QM2,QM3,Q1(20000),EZV
      REAL*4 Q2(20000),Q3(20000),DQ1(20000),DQ2(20000),DQ3(20000),KO
      REAL*4 EX,EY,EZ,JA(3,3),XVD,YVD,ZVD,FXP,FYP,FZP,KP,KV,MX(3,3),R
      REAL*4 VX(3),GX(3),FX,FY,FZ,T1(20000),T2(20000),T3(20000),EYV
      REAL*4 TT,W,TMAX,DQM1,DQM2,DQM3,ANS,KQ,KQ1,KQ2,KQ3,YVDD(20000)
      REAL*4 Q1MAX,Q1MIN,Q2MAX,Q2MIN,Q3MAX,Q3MIN,B,FO,XVDD(20000),EXV
      REAL*4 RO,ROF(10),RS(10),XO,YO,ZO,XOF(10),YOF(10),ZVDD(20000),DM
      REAL*4 ZOF(10),XS(10),YS(10),ZS(10),COM(10),MIND,FXO,FYO,FZO,TV
      REAL*4 EXO,EYO,NOBS,F(3),C,V,MV,XDD(20000),YDD(20000),ZDD(20000)
C
      INTEGER NB,IP
C
C
C
      IP=1
      Q1MAX=1.92
      Q1MIN=-1.92
      Q2MAX=1.484
      Q2MIN=-.47
      Q3MAX=0.0
      Q3MIN=-2.13
C
      L1=.19
      L2=.3
      L3=.3
      M1=2.
      M2=2.
      M3=2.
C
C INPUT DATA
C
      TT=0.0
```

```
      WRITE(6,754)
  754 FORMAT(2X,' WHAT IS THE SPRING CONSTANT?')
      READ(*,753) KP
  753 FORMAT(F12.3)
      WRITE(6,752)
  752 FORMAT(2X,' WHAT IS THE DAMPER CONSTANT?')
      READ(*,756) KV
  756 FORMAT(F12.3)
C
      II=1
C
C
  113 WRITE(6,200)
  200 FORMAT(2X,' XD=')
C
      READ(*,210) XD
  210 FORMAT(F12.4)
C
      WRITE(6,220)
  220 FORMAT(2X,' YD=')
C
      READ(*,230) YD
  230 FORMAT(F12.4)
C
      WRITE(6,240)
  240 FORMAT(2X,' ZD=')
C
      READ(*,250) ZD
  250 FORMAT(F12.4)
C
      WRITE(6,260)
  260 FORMAT(2X,'XM=')
C
      READ(*,270) XM
  270 FORMAT(F12.4)
C
      WRITE(6,280)
  280 FORMAT(2X,'YM=')
C
      READ(*,290) YM
  290 FORMAT(F12.4)
C
      WRITE(6,300)
  300 FORMAT(2X,'ZM=')
C
      READ(*,310) ZM
  310 FORMAT(F12.4)
C
      WRITE(6,411)
  411 FORMAT(2X,' WHAT IS MAXIMUM TIME? ')
C
      READ(*,412) TMAX
  412 FORMAT(F10.3)
C
      WRITE(6,413)
```

```
  413 FORMAT(2X,' WHAT IS PAYLOAD WEIGHT? ')
C
      READ(*,414) W
  414 FORMAT(F10.3)
C
      WRITE(6,843)
  843 FORMAT(2X,' WHAT IS TARGET VELOCITY')
C
      READ(*,832) TV
  832 FORMAT(F10.3)
C
      WRITE(6,342)
  342 FORMAT(2X,' WHAT IS ANGLE CONSTANT (KQ)? ')
C
      READ(*,343) KQ
  343 FORMAT(F10.3)
C
      WRITE(6,622)
  622 FORMAT(2X,'IS THERE ANY OBSTACLE IN WORKVOLUME? YES(1.)')
C
      READ(*,722) NOBS
  722 FORMAT(F4.2)
C
      IF (NOBS .NE. 1.) GO TO 723
C
      WRITE(6,601)
  601 FORMAT(2X,' HOW MANY OBSTACLES? ')
C
      READ(*,602) NB
  602 FORMAT(I2)
C
      DO 222 I=1,NB
      WRITE(6,223) I
  223 FORMAT(2X,'XS(',I2,')= ')
C
      READ(*,224) XS(I)
  224 FORMAT(F10.3)
C
      WRITE(6,225) I
  225 FORMAT(2X,'YS(',I2,')=')
C
      READ(*,226) YS(I)
  226 FORMAT(F10.3)
C
      WRITE(6,227) I
  227 FORMAT(2X,'ZS(',I2,')=')
C
      READ(*,228) ZS(I)
  228 FORMAT(F10.3)
C
      WRITE(6,229) I
  229 FORMAT(2X,'RS(',I2,')=')
C
      READ(*,241) RS(I)
  241 FORMAT(F10.3)
```

```
C
  222 CONTINUE
C
      WRITE(6,233)
  233 FORMAT(2X,' WHAT IS REPULSIVE FORCE CONSTANT (KO)?')
C
C
      READ(*,234) KO
  234 FORMAT(F10.3)
C
      WRITE(6,242)
  242 FORMAT(2X,' WHAT IS REPULSIVE FORCE DAMPING CONSTANT (B)?')
C
      READ(*,243) B
  243 FORMAT(F10.3)
C
      WRITE(6,235)
  235 FORMAT(2X,' WHAT IS RADIUS OF REPULSIVE FORCE (R)?')
C
      READ(*,236) R
  236 FORMAT(F10.3)
C
C
  723 CALL INVKIN(XD,YD,ZD,QD1,QD2,QD3,ANS)
      IF (ANS .EQ. 1.0) GO TO 114
      CALL INVKIN(XM,YM,ZM,QM1,QM2,QM3,ANS)
C
  114 IF (ANS .EQ. 1.0) THEN
        WRITE(6,51)
   51   FORMAT(2X,' ROBOT CAN NOT REACH ORIGINAL OR FINAL POSITION
     1  ,PLEASE TRY AGAIN. ')
        GO TO 113
      ENDIF
C
      IF(QD1 .GT. Q1MAX .OR. QD1 .LT. Q1MIN)GO TO 330
      IF(QD2 .GT. Q2MAX .OR. QD2 .LT. Q2MIN)GO TO 330
      IF(QD3 .GT. Q3MAX .OR. QD3 .LT. Q3MIN)GO TO 330
      GO TO 340
  330 WRITE(6,332)
  332 FORMAT(2X,'THE TARGET POSITION IS NOT IN WORK VOLUME. TRY AGAIN')
      GO TO 113
C
  340 IF(QM1 .GT. Q1MAX .OR. QM1 .LT. Q1MIN) GO TO 333
      IF(QM2 .GT. Q2MAX .OR. QM2 .LT. Q2MIN) GO TO 333
      IF(QM3 .GT. Q3MAX .OR. QM3 .LT. Q3MIN) GO TO 333
      GO TO 341
  333 WRITE(6,334)
  334 FORMAT(2X,'ORIGINAL POSITION OF ROBOT IS NOT IN WORK VOLUME.
     1 TRY AGAIN! ')
      GO TO 113
C
  341 WRITE(6,44) QD1,QD2,QD3,QM1,QM2,QM3
   44 FORMAT(2X,6F10.4)
C
```

```
      Q1(1)=QM1
      Q2(1)=QM2
      Q3(1)=QM3
C
      DQ1(1)=0.0
      DQ2(1)=0.0
      DQ3(1)=0.0
C
      WRITE(6,354)
  354 FORMAT(2X,' WHAT IS VISCOUS FRICTION CONSTANT (V)? ')
C
      READ(*,355) V
  355 FORMAT(F10.3)
C
      WRITE(6,356)
  356 FORMAT(2X,' WHAT IS COULOMB FRICTION CONSTANT (C)? ')
C
      READ(*,357) C
  357 FORMAT(F10.3)
C
      DM=0.0
C
      WRITE(6,804)
  804 FORMAT(2X,'WHICH TERM SHOULD NOT BE COMPENSATED INERTIAL(1.),
     1 CENTRIFUGAL(2.), GRAVITY(3.), FRICTION(4.)')
      READ(*,877) DM
  877 FORMAT(F10.3)
C
  999 XDD(II)=XD+.04*COS(TV*TT)
      YDD(II)=YD+.04*SIN(TV*TT)
      ZDD(II)=ZD
C
      XVDD(II)=-.04*TV*SIN(TV*TT)
      YVDD(II)=.04*TV*COS(TV*TT)
      ZVDD(II)=0.0
C
      EX=XDD(II)-XM
      EY=YDD(II)-YM
      EZ=ZDD(II)-ZM
C
      IF (NOBS .NE. 1.) GO TO 824
      DO 231 I=1,NB
      COM(I)=RS(I)/((XM-XS(I))**2.+(YM-YS(I))**2.+(ZM-ZS(I))**2.)**.5
        XOF(I) = (XM-XS(I))*COM(I) + XS(I)
        YOF(I) = (YM-YS(I))*COM(I) + YS(I)
        ZOF(I) = (ZM-ZS(I))*COM(I) + ZS(I)
      ROF(I) = ((XM-XOF(I))**2.+(YM-YOF(I))**2.+(ZM-ZOF(I))**2.)**.5
  231 CONTINUE
C
      MIND=100.
      DO 232 I=1,NB
         IF(ROF(I) .LE. MIND) THEN
            XO=XOF(I)
            YO=YOF(I)
            ZO=ZOF(I)
```

```
            RO=ROF(I)
            MIND=ROF(I)
         ENDIF
  232 CONTINUE
C
      EXO=XM-XO
      EYO=YM-YO
      EZO=ZM-ZO
C
      IF (RO .GT. R) THEN
      FXO=0.0
      FYO=0.0
      FZO=0.0
      ENDIF
C
      IF (RO .LT. R) THEN
        FO=KO*((RO-R)**2.)
        FXO=FO*EXO/RO -XVD*B
        FYO=FO*EYO/RO -YVD*B
        FZO=FO*EZO/RO -ZVD*B
      ENDIF
C
  824 CALL JACOBIAN(Q1,Q2,Q3,II,JA)
C
      XVD=JA(1,1) * DQ1(II) +JA(1,2)*DQ2(II)+JA(1,3)*DQ3(II)
      YVD=JA(2,1) * DQ1(II) +JA(2,2)*DQ2(II)+JA(2,3)*DQ3(II)
      ZVD=JA(3,1) * DQ1(II) +JA(3,2)*DQ2(II)+JA(3,3)*DQ3(II)
C
C     MV=(XVD*XVD+YVD*YVD+ZVD*ZVD)**.5
C
      IF (NOBS .NE. 1.) THEN
         FXO=0.0
         FYO=0.0
         FZO=0.0
      ENDIF
C
      EXV=XVDD(II)-XVD
      EYV=YVDD(II)-YVD
      EZV=ZVDD(II)-ZVD
C
      FXP=EX*KP+EXV*KV+FXO
      FYP=EY*KP+EYV*KV+FYO
      FZP=EZ*KP+EZV*KV+FZO
C
      CALL MVGX(W,TT,DQ1,DQ2,DQ3,Q1,Q2,Q3,II,MX,VX,GX)
C
      IF (DM .EQ. 2.) THEN
      VX(1)=0.0
      VX(2)=0.0
      VX(3)=0.0
      ENDIF
C
      IF (DM .EQ. 1.) THEN
      MX(1,1)=1.0
      MX(1,2)=0.0
```

```
      MX(1,3)=0.0
      MX(2,1)=0.0
      MX(2,2)=1.0
      MX(2,3)=0.0
      MX(3,1)=0.0
      MX(3,2)=0.0
      MX(3,3)=1.0
      ENDIF
C
      IF (DM .EQ. 3.) THEN
      GX(1)=0.0
      GX(2)=0.0
      GX(3)=0.0
      ENDIF
C
  888 FX=FXP*MX(1,1)+FYP*MX(1,2)+FZP*MX(1,3)+VX(1)+GX(1)
      FY=FXP*MX(2,1)+FYP*MX(2,2)+FZP*MX(2,3)+VX(2)+GX(2)
      FZ=FXP*MX(3,1)+FYP*MX(3,2)+FZP*MX(3,3)+VX(3)+GX(3)
C
C
      KQ1=0.0
      KQ2=0.0
      KQ3=0.0
      IF(Q1(II) .GT. Q1MAX .OR. Q1(II) .LT. Q1MIN) KQ1=KQ
      IF(Q2(II) .GT. Q2MAX .OR. Q2(II) .LT. Q2MIN) KQ2=KQ
      IF(Q3(II) .GT. Q3MAX .OR. Q3(II) .LT. Q3MIN) KQ3=KQ
C
      T1(II)=FX*JA(1,1)+FY*JA(2,1)+FZ*JA(3,1)-DQ1(II)*KQ1
      T2(II)=FX*JA(1,2)+FY*JA(2,2)+FZ*JA(3,2)-DQ2(II)*KQ2
      T3(II)=FX*JA(1,3)+FY*JA(2,3)+FZ*JA(3,3)-DQ3(II)*KQ3
C
C
C
      CALL ARMDYN(C,V,W,T1,T2,T3,Q1,Q2,Q3,DQ1,DQ2,DQ3,II,QM1,QM2,QM3,
     1 DQM1,DQM2,DQM3,TT,F)
C
      II=II+1
C
      Q1(II)=QM1
      Q2(II)=QM2
      Q3(II)=QM3
C
      DQ1(II)=DQM1
      DQ2(II)=DQM2
      DQ3(II)=DQM3
C
      CALL KIN(Q1,Q2,Q3,II,XM,YM,ZM)
C
      IF (XM .EQ. XDD(II-1) .AND. YM .EQ. YDD(II-1)) THEN
      IF (ZM .EQ. ZDD(II-1)) GO TO 111
      ENDIF
C
C
C CREATE DATA FILES FOR ALL OUTPUTS
C
```

```
      IF (IP .EQ. 1) THEN
      WRITE(4,66) XM,YM
   66 FORMAT(2F14.5)
C
      WRITE(4,676) XDD(II-1),YDD(II-1)
  676 FORMAT(2F14.5)
C
      IF (NOBS .EQ. 1.) THEN
      WRITE(4,67) XO,YO
   67 FORMAT(2F14.5)
      ENDIF
C
      WRITE(3,77) XM,ZM
   77 FORMAT(2F14.5)
C
      IF (NOBS .EQ. 1.) THEN
      WRITE(3,75) XO,ZO
   75 FORMAT(2F14.5)
      ENDIF
C
      WRITE(2,91) TT,Q1(II)
   91 FORMAT(2F14.5)
C
      WRITE(2,396) TT,Q2(II)
  396 FORMAT(2F14.5)
C
      WRITE(2,397) TT,Q3(II)
  397 FORMAT(2F14.5)
C
C
      WRITE(1,88) TT,T1(II-1)
   88 FORMAT(2F14.5)
C
      WRITE(1,78) TT,T2(II-1)
   78 FORMAT(2F14.5)
C
      WRITE(1,79) TT,T3(II-1)
   79 FORMAT(2F14.5)
C
      WRITE(7,93) TT,XM
   93 FORMAT(2F14.5)
C
      WRITE(7,94) TT,YM
   94 FORMAT(2F14.5)
C
      WRITE(7,95) TT,ZM
   95 FORMAT(2F14.5)
C
      WRITE(10,811) XM,YM,ZM
  811 FORMAT(3F14.5)
C
      IF (NOBS .EQ. 1.) THEN
      WRITE(10,812) XO,YO,ZO
  812 FORMAT(3F14.5)
      ENDIF
```

```
      WRITE(8,876) TT,F(1)
  876 FORMAT(2F14.5)
C
      WRITE(8,875) TT,F(2)
  875 FORMAT(2F14.5)
C
      WRITE(8,874) TT,F(3)
  874 FORMAT(2F14.5)
C
      WRITE(9,534) TT,MV
  534 FORMAT(2F14.5)
C
      ENDIF
C
      IP=IP+1
      IF(IP .EQ. 20) IP=1
C
      IF (TT .GT. TMAX) GO TO 111
      GO TO 999
C
C
  111 STOP
      END
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C MULTIPLICATION OF TWO MATRICES     C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      SUBROUTINE  MULTMAT(A,B,AB)
C
      REAL*4 AB(3,3),A(3,3),B(3,3)
C
C
      DO 5 K=1,3
       DO 10 I=1,3
       AB(K,I)=0.0
         DO 20 J=1,3
           AB(K,I) = A(K,J) * B(J,I) + AB(K,I)
   20    CONTINUE
   10  CONTINUE
    5 CONTINUE
C
      RETURN
      END
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C FORWARD KINEMATICS CALCULATIONS       C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
      SUBROUTINE KIN(Q1,Q2,Q3,II,XM,YM,ZM)
C
      REAL*4 Q1(20000),Q2(20000),Q3(20000),XM,YM,ZM
      REAL*4 L3,L2,L1,C1,S1,S23,C23,S2,C2
C
```

```
      L1=.19
      L2=.3
      L3=.3
      C1=COS(Q1(II))
      S1=SIN(Q1(II))
      C2=COS(Q2(II))
      S2=SIN(Q2(II))
      S23=SIN(Q2(II)+Q3(II))
      C23=COS(Q2(II)+Q3(II))
C
      XM=L3*C1*C23+L2*C1*C2
      YM=L3*S1*C23+L2*S1*C2
      ZM=L3*S23+L2*S2+L1
C
      RETURN
      END
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C INVERSE KINEMATICS CALCULATIONS          C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      SUBROUTINE INVKIN(X,Y,Z,Q11,Q22,Q33,ANS)
C
      REAL*4 X,Y,Z,Q11,Q22,Q33,K,D,C,ANS
      REAL*4 L1,L2,L3,A,B,E,S1,C3,S3,KK
C
      L1=.19
      L2=.3
      L3=.3
C
      ANS=0.0
C
C
      Q11 = ATAN2(Y,X)
      S1=SIN(Q11)
      IF (Q11 .EQ. 0) THEN
      KK=0.0
      GO TO 666
      ENDIF
      KK=Y/S1
  666 K=(KK)**2 + (Z-L1)**2
      D=K-L3*L3-L2*L2
      CC=(2.*L2*L3)**2-D*D
      IF (CC .LT. 0.0) THEN
        ANS=1.0
        GO TO 112
      ENDIF
C
      C=CC**.5
      Q33=ATAN2(-C,D)
      C3=COS(Q33)
      S3=SIN(Q33)
      A=(Z-L1)*L3*S3+KK*(L3*C3+L2)
      B=(L3*S3)**2+(L3*C3+L2)**2
```

```
      AB=B*B-A*A
      IF (AB .LT. 0.0) THEN
         ANS=1.0
         GO TO 112
      ENDIF
C
      E=AB**.5
      Q22= ATAN2(E,A)
C
C
  112 RETURN
      END
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C JACOBIAN MATRIX CALCULATIONS              C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      SUBROUTINE JACOBIAN(Q1,Q2,Q3,II,JA)
C
      REAL*4 Q1(20000),Q2(20000),Q3(20000),JA(3,3)
      REAL*4 L1,L2,L3,S1,C1,S2,C2,C23,S23
C
      L1=.19
      L2=.3
      L3=.3
C
      S1=SIN(Q1(II))
      C1=COS(Q1(II))
      S2=SIN(Q2(II))
      C2=COS(Q2(II))
      C23=COS(Q2(II)+Q3(II))
      S23=SIN(Q2(II)+Q3(II))
C
      JA(1,1)=-L3*S1*C23-L2*S1*C2
      JA(1,2)=-L3*C1*S23-L2*C1*S2
      JA(1,3)=-L3*C1*S23
      JA(2,1)=L3*C23*C1+L2*C1*C2
      JA(2,2)=-L3*S1*S23-L2*S1*S2
      JA(2,3)=-L3*S1*S23
      JA(3,1)=0.0
      JA(3,2)=L3*C23+L2*C2
      JA(3,3)=L3*C23
C
C
      RETURN
      END
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C TRANSPOSE OF JACOBIAN MATRIX              C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      SUBROUTINE TRANSPJA(JA,JT)
C
      REAL*4 JA(3,3),JT(3,3)
C
```

```
      JT(1,1)=JA(1,1)
      JT(1,2)=JA(2,1)
      JT(1,3)=JA(3,1)
C
      JT(2,1)=JA(1,2)
      JT(2,2)=JA(2,2)
      JT(2,3)=JA(3,2)
C
      JT(3,1)=JA(1,3)
      JT(3,2)=JA(2,3)
      JT(3,3)=JA(3,3)
C
C
      RETURN
      END
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C CALCULATIONS OF INERTIAL, CENTRIFUGAL, CORIOLIS,GRAVITY   C
C AND FRICTION TERMS                                        C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      SUBROUTINE MVG(W,Q1,Q2,Q3,II,M,V,G)
C
      REAL*4 Q1(20000),Q2(20000),Q3(20000),M(3,3),V(3),G(3),IZZ1,IYY1
      REAL*4 IXX1,IZZ2,IYY2,IXX2,IXX3,IYY3,IZZ3,L3,L2,L1
      REAL*4 S1,S2,S3,S23,C1,C2,C3,C23,M3,M2,M1,W,S223,S2233
      REAL*4 DQ1(20000),DQ2(20000),DQ3(20000),GG
C
      GG=9.81
      L1=.19
      L2=.3
      L3=.3
      M1=2.
      M2=2.
      M3=2.
      IXX1=0.0
      IYY1=0.0
      IXX2=0.0
      IYY2=0.0
      IXX3=0.0
      IYY3=0.0
      IZZ1=0.1
      IZZ2=0.1
      IZZ3=0.1
      S1=SIN(Q1(II))
      S2=SIN(Q2(II))
      S3=SIN(Q3(II))
      S23=SIN(Q2(II)+Q3(II))
      C1=COS(Q1(II))
      C2=COS(Q2(II))
      C3=COS(Q3(II))
      C23=COS(Q2(II)+Q3(II))
      S223=SIN(2.*Q2(II)+Q3(II))
      S2233=SIN(2.*Q2(II)+2.*Q3(II))
```

```
C
      M(1,1)=IZZ1+S2*S2*IXX2+C2*C2*IYY2+S23*S23*IXX3-C23*C23*IYY3
     1 +L3*L3*C23*C23*M3/4.+L2*C2*C2*(M2/4.+M3)+L2*L3*C2*C23*M3
      M(1,2)=0.0
      M(1,3)=0.0
      M(2,1)=0.0
      M(2,2)=IZZ2+IZZ3+L3*L3*M3/4.+L2*L3*C3*M3+M3*L2*L2
      M(2,3)=IZZ3+L3*L3*M3/4.+L2*L3*C3*M3/2.
      M(3,1)=0.0
      M(3,2)=IZZ3+(L3/2.+L2*C3)*M3
      M(3,3)=IZZ3+L3*M3/2.
C
C
      V(1)=DQ1(II)*DQ2(II)*(2.*C2*S2*(IXX2-IYY2)+S2233*(IXX3-IYY3)
     1   -L3*L3*S23*C23*M3/2.-L2*L3*S223*M3-S2*C2*L2*L2*(M2/2.+2.*M3))
     2   +DQ1(II)*DQ3(II)*(S2233*(IXX3-IYY3)+L3*L3*S23*(S2*S3+C2*C3)/2.
     3   -L2*L3*S23*C2*M3)
C
C
      V(2)=DQ1(II)**2 *(S2*C2*(IYY2-IXX2)+S23*C23*(IYY3-IXX3)+L3*L3
     1    *C23*S23*M3/4.+L2*L3*C2*S23*M3/2.+L2*L2*S2*C2*M2/4.+
     2    L2*L3*C23*S2*M3/2.+L2*L2*C2*S2*M3) + DQ2(II)*L2*L2*M2/4.+
     3    DQ2(II)**2 *(L2*L3*S3*M3/2.)-(DQ2(II)+DQ3(II))**2*L2*L3*
     4    S3*M3/2.
C
C
      V(3)=DQ1(II)**2 *(S23*C23*(IYY3-IXX3)+L3*L3*C23*S23*M3/4.+
     1     L2*L3*C2*S23*M3/2.)+DQ2(II)**2 *L2*S3*M3
C
C
      G(1)=GG*(-L3*S1*M3*(S2*S3+C2*C3)/2. + L2*S1*C2*(M2/2. +M3))
      G(2)=GG*(-L3*C1*S23*M3/2.-L2*S2*C1*M2/2.-L2*C2*S2*M3)+W*L2*S3
      G(3)=GG*(-C1*S23*M3)
C
      RETURN
      END
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C CALCULATIONS OF JACOBIAN DERIVATIVE MATRIX  C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      SUBROUTINE JACOBIAND(Q1,Q2,Q3,DQ1,DQ2,DQ3,II,JD)
C
C
      REAL*4 Q1(20000),Q2(20000),Q3(20000),DQ1(20000),DQ2(20000)
      REAL*4 DQ3(20000),JD(3,3),L1,L2,L3,C1,S1,C2,S2,C23,S23
C
      L1=.25
      L2=.3
      L3=.3
      C1=COS(Q1(II))
      S1=SIN(Q1(II))
      C2=COS(Q2(II))
      S2=SIN(Q2(II))
```

```
      C23=COS(Q2(II)+Q3(II))
      S23=SIN(Q2(II)+Q3(II))
C
      JD(1,1)=-L3*C1*C23*DQ1(II)+L3*S1*S23*(DQ2(II)+DQ3(II))
     1        -L2*C1*DQ1(II)*C2+L2*S1*S2*DQ2(II)
C
      JD(1,2)=L3*S1*S23*DQ1(II)-L3*C1*C23*(DQ2(II)+DQ3(II))
     1        +L2*S1*DQ1(II)*S2-L2*C1*C2*DQ2(II)
C
      JD(1,3)=L3*S1*S23*DQ1(II)-L3*C1*C23*(DQ2(II)+DQ3(II))
C
      JD(2,1)=-L3*C23*S1*DQ1(II)-L3*C1*S23*(DQ2(II)+DQ3(II))
     1        -L2*S1*C2*DQ1(II)-L2*C1*S2*DQ2(II)
C
      JD(2,2)=-L3*C1*S23*DQ1(II)-L3*S1*C23*(DQ2(II)+DQ3(II))
     1        -L2*C1*DQ1(II)*S2-L2*S1*C2*DQ2(II)
C
      JD(2,3)=-L3*C1*S23*DQ1(II)-L3*S1*C23*(DQ2(II)+DQ3(II))
C
      JD(3,1)=0
      JD(3,2)=-L3*S23*(DQ2(II)+DQ3(II))-L2*S2*DQ2(II)
      JD(3,3)=-L3*S23*(DQ2(II)+DQ3(II))
C
      RETURN
      END
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C CENTRIFUGAL, CORIOLIS AND GRAVITY TERMS CALCULATION   C
C IN CARTESIAN SPACE                                    C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      SUBROUTINE VGX(JI,JTI,M,V,G,JD,DQ1,DQ2,DQ3,II,VX,GX)
C
      REAL*4 JI(3,3),JTI(3,3),M(3,3),V(3),G(3),DQ1(20000)
      REAL*4 DQ2(20000),DQ3(20000),JD(3,3),JIDQ(3),JDQ(3)
      REAL*4 MJIDQ(3),VM(3),VX(3),GX(3)
C
         JDQ(1)=JD(1,1)*DQ1(II)+JD(1,2)*DQ2(II)+JD(1,3)*DQ3(II)
         JDQ(2)=JD(2,1)*DQ1(II)+JD(2,2)*DQ2(II)+JD(2,3)*DQ3(II)
         JDQ(3)=JD(3,1)*DQ1(II)+JD(3,2)*DQ2(II)+JD(3,3)*DQ3(II)
C
         DO 30 I=1,3
            JIDQ(I)=0.0
            DO 40 J=1,3
               JIDQ(I)=JI(I,J)*JDQ(J)+JIDQ(I)
   40       CONTINUE
   30    CONTINUE
C
      DO 50 I=1,3
         MJIDQ(I)=0.0
         DO 60 J=1,3
            MJIDQ(I)=M(I,J)*JIDQ(J)+MJIDQ(I)
   60    CONTINUE
   50 CONTINUE
```

```
C
      DO 70 I=1,3
         VM(I)=V(I)-MJIDQ(I)
   70 CONTINUE
C
      DO 80 I=1,3
         VX(I)=0.0
         DO 90 J=1,3
            VX(I)=JTI(I,J)*VM(J)+VX(I)
   90    CONTINUE
   80 CONTINUE
C
      DO 100 I=1,3
          GX(I)=0.0
          DO 110 J=1,3
            GX(I)=JTI(I,J)*G(J)+GX(I)
  110     CONTINUE
  100 CONTINUE
C
C
C
      RETURN
      END
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C INERTIAL, CENTRIFUGAL, CORIOLIS AND GRAVITY TERMS   C
C IN CARTESIAN SPACE                                  C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      SUBROUTINE MVGX(W,TT,DQ1,DQ2,DQ3,Q1,Q2,Q3,II,MX,VX,GX)
C
      REAL*4 JT(3,3),M(3,3),V(3),G(3),JA(3,3),JI(3,3),JTI(3,3)
      REAL*4 JD(3,3),JTIM(3,3),MX(3,3),DQ1(20000),DQ2(20000)
      REAL*4 DQ3(20000),W,VX(3),GX(3),TT,INDEX(3)
      INTEGER NP
C
      CALL JACOBIAN(Q1,Q2,Q3,II,JA)
C
      CALL TRANSPJA(JA,JT)
C
      CALL MVG(W,Q1,Q2,Q3,II,M,V,G)
      DO 10 I=1,3
         DO 20 J=1,3
           JI(I,J)=JA(I,J)
          JTI(I,J)=JT(I,J)
   20    CONTINUE
   10 CONTINUE
C
      NP=3
C     WRITE(6,100) TT
C 100 FORMAT(2X,' INVERS OF JI AT TIME  ',F12.5)
      CALL INVERS(JI,NP,INDEX)
C     WRITE(6,110) TT
C 110 FORMAT(2X,' INVERS OF JTI AT TIME ',F12.5)
```

```
C
      CALL INVERS(JTI,NP,INDEX)
      CALL JACOBIAND(Q1,Q2,Q3,DQ1,DQ2,DQ3,II,JD)
      CALL MULTMAT(JTI,M,JTIM)
      CALL MULTMAT(JTIM,JI,MX)
      CALL VGX(JI,JTI,M,V,G,JD,DQ1,DQ2,DQ3,II,VX,GX)
C
C
      RETURN
      END
C
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C ARM DYNAMICS CALCULATIONS          C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      SUBROUTINE ARMDYN(C,VV,W,T1,T2,T3,Q1,Q2,Q3,DQ1,DQ2,DQ3,II,QM1,QM2,
     1 QM3,DQM1,DQM2,DQM3,TT,F)
C
C
      REAL*4 Q1(20000),Q2(20000),Q3(20000),DQ1(20000),DQ2(20000)
      REAL*4 DQ3(20000),DDQ1(20000),DDQ2(20000),DDQ3(20000),W
      REAL*4 T1(20000),T2(20000),T3(20000),T(3),V(3),G(3),DT
      REAL*4 TVG(3),MI(3,3),TT,M(3,3),QM1,QM2,QM3,DQM1,DQM2,DQM3
      REAL*4 INDEX(3),F(3),C,VV
C
      INTEGER NP
C
      CALL MVG(W,Q1,Q2,Q3,II,M,V,G)
      DO 33 I=1,3
          DO 34 J=1,3
               MI(I,J)=M(I,J)
   34     CONTINUE
   33 CONTINUE
C
      DT=.004
      IF (II .EQ. 1) TT=0.0
      TT=TT+DT
      NP=3
      CALL INVERS(MI,NP,INDEX)
C
      T(1)=T1(II)
      T(2)=T2(II)
      T(3)=T3(II)
C
C
      IF (DQ1(II) .EQ. 0) F(1)=0
      IF (DQ1(II) .NE. 0) F(1)=DQ1(II)*VV + C* DQ1(II)/ABS(DQ1(II))
      IF (DQ2(II) .EQ. 0) F(2)=0
      IF (DQ2(II) .NE. 0) F(2)=DQ2(II)*VV + C* DQ2(II)/ABS(DQ2(II))
      IF (DQ3(II) .EQ. 0) F(3)=0
      IF (DQ3(II) .NE. 0) F(3)=DQ3(II)*VV + C* DQ3(II)/ABS(DQ3(II))
      DO 10 I=1,3
         TVG(I)=T(I)-V(I)-G(I)-F(I)
```

```
   10 CONTINUE
C
      DDQ1(II)=0.0
      DDQ2(II)=0.0
      DDQ3(II)=0.0
C
      DO 20 I=1,3
         DDQ1(II)=MI(1,I)*TVG(I)+DDQ1(II)
         DDQ2(II)=MI(2,I)*TVG(I)+DDQ2(II)
         DDQ3(II)=MI(3,I)*TVG(I)+DDQ3(II)
   20 CONTINUE
C
      DQM1=DQ1(II)+DDQ1(II)*DT
      DQM2=DQ2(II)+DDQ2(II)*DT
      DQM3=DQ3(II)+DDQ3(II)*DT
C
      QM1=Q1(II)+DQ1(II)*DT+.5*DDQ1(II)*DT*DT
      QM2=Q2(II)+DQ2(II)*DT+.5*DDQ2(II)*DT*DT
      QM3=Q3(II)+DQ3(II)*DT+.5*DDQ3(II)*DT*DT
C
C
      RETURN
      END
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C INVERSE OF MATRIX CALCULATIONS         C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      SUBROUTINE INVERS(ASAT,NP,INDEX)
C
C
C
      DIMENSION ASAT(3,3),INDEX(3)
      DO 113 I=1,NP
  113 INDEX(I)=0
  114 AMAX=-1.
      DO 115 I=1,NP
      IF(INDEX(I)) 115,116,115
  116 TEMP=ABS(ASAT(I,I))
      IF(TEMP-AMAX)115,115,117
  117 ICOL=I
      AMAX=TEMP
  115 CONTINUE
      IF(AMAX)118,124,119
  119 INDEX(ICOL)=1
      PIVOT=ASAT(ICOL,ICOL)
      ASAT(ICOL,ICOL)=1.0
      PIVOT=1./PIVOT
      DO 120 J=1,NP
      IF(ASAT(ICOL,J).EQ.0.0)GO TO 120
      ASAT(ICOL,J)=ASAT(ICOL,J)*PIVOT
  120 CONTINUE
      DO 121 I=1,NP
      IF(I-ICOL)122,121,122
```

```
  122 TEMP=ASAT(I,ICOL)
      ASAT(I,ICOL)=0.0
      DO 123 J=1,NP
      IF(ASAT(ICOL,J).EQ.0.0)GO TO 123
      ASAT(I,J)=ASAT(I,J)-ASAT(ICOL,J)*TEMP
  123 CONTINUE
  121 CONTINUE
      GO TO 114
  124 WRITE(6,125)
  125 FORMAT(14HOPIVOT IS ZERO)
      STOP
  118 CONTINUE
      RETURN
      END
C
```

```
C
C*****************************************
C*            R3DTOR.FOR                 *
C*****************************************
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C IMPEDANCE CONTROL FOR 3 DOF ROBOT IN  C
C 3 DIMENSIONAL SPACE                   C
C                                       C
C ALSO INCLUDS: - CARTESIAN RESCALING   C
C                 SCHEME FOR SATURATION C
C                 POINT OF ANY JOINT    C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
      REAL*4 XD,YD,ZD,QD1,QD2,QD3,XM,YM,ZM,QM1,QM2,QM3,Q1(20000)
      REAL*4 Q2(20000),Q3(20000),DQ1(20000),DQ2(20000),DQ3(20000)
      REAL*4 EX,EY,EZ,JA(3,3),XVD,YVD,ZVD,FXP,FYP,FZP,KP,KV,MX(3,3)
      REAL*4 VX(3),GX(3),FX,FY,FZ,T1(20000),T2(20000),T3(20000)
      REAL*4 TT,W,TMAX,DQM1,DQM2,DQM3,ANS,KQ,KQ1,KQ2,KQ3
      REAL*4 Q1MAX,Q1MIN,Q2MAX,Q2MIN,Q3MAX,Q3MIN,B,FO
      REAL*4 KO,R,RO,ROF(10),RS(10),XO,YO,ZO,XOF(10),YOF(10)
      REAL*4 ZOF(10),XS(10),YS(10),ZS(10),COM(10),MIND,FXO,FYO,FZO
      REAL*4 EXO,EYO,NOBS,F(3),C,V,MV,T1MAX,T2MAX,T3MAX
      REAL*4 A1,A2,TT1,TT2,TT3,AT1(20000),AT2(20000),AT3(20000)
      REAL*4 DIFT1,DIFT2,DIFT3,DIFT
C
      INTEGER NB,IP
C
C
C
      IP=1
      Q1MAX=1.92
      Q1MIN=-1.92
      Q2MAX=1.484
      Q2MIN=-.47
      Q3MAX=0.0
      Q3MIN=-2.13
C
      L1=.19
      L2=.3
      L3=.3
      M1=2.
      M2=2.
      M3=2.
C
      TT=0.0
      KV=4.
      KP=4.
      II=1
C
```

```
      WRITE(6,941)
  941 FORMAT(2X,'WHAT IS THE SPRING CONSTANT(KP)?')
      READ(*,942) KP
  942 FORMAT(F10.3)
C
      WRITE(6,943)
  943 FORMAT(2X,'WHAT IS THE DAMPING CONSTANT(KV)?')
      READ(*,944) KV
  944 FORMAT(F10.3)
C
  113 WRITE(6,200)
  200 FORMAT(2X,' XD=')
C
      READ(*,210) XD
  210 FORMAT(F12.4)
C
      WRITE(6,220)
  220 FORMAT(2X,' YD=')
C
      READ(*,230) YD
  230 FORMAT(F12.4)
C
      WRITE(6,240)
  240 FORMAT(2X,' ZD=')
C
      READ(*,250) ZD
  250 FORMAT(F12.4)
C
      WRITE(6,260)
  260 FORMAT(2X,'XM=')
C
      READ(*,270) XM
  270 FORMAT(F12.4)
C
      WRITE(6,280)
  280 FORMAT(2X,'YM=')
C
      READ(*,290) YM
  290 FORMAT(F12.4)
C
      WRITE(6,300)
  300 FORMAT(2X,'ZM=')
C
      READ(*,310) ZM
  310 FORMAT(F12.4)
C
      WRITE(6,411)
  411 FORMAT(2X,' WHAT IS MAXIMUM TIME? ')
C
      READ(*,412) TMAX
  412 FORMAT(F10.3)
C
      WRITE(6,413)
  413 FORMAT(2X,' WHAT IS PAYLOAD WEIGHT? ')
```

```
C
      READ(*,414) W
  414 FORMAT(F10.3)
C
      WRITE(6,342)
  342 FORMAT(2X,' WHAT IS ANGLE CONSTANT (KQ)? ')
C
      READ(*,343) KQ
  343 FORMAT(F10.3)
C
      WRITE(6,622)
  622 FORMAT(2X,'IS THERE ANY OBSTACLE IN WORKVOLUME? YES(1.)')
C
      READ(*,722) NOBS
  722 FORMAT(F4.2)
C
      IF (NOBS .NE. 1.) GO TO 723
C
      WRITE(6,601)
  601 FORMAT(2X,' HOW MANY OBSTACLES? ')
C
      READ(*,602) NB
  602 FORMAT(I2)
C
      DO 222 I=1,NB
      WRITE(6,223) I
  223 FORMAT(2X,'XS(',I2,')= ')
C
      READ(*,224) XS(I)
  224 FORMAT(F10.3)
C
      WRITE(6,225) I
  225 FORMAT(2X,'YS(',I2,')=')
C
      READ(*,226) YS(I)
  226 FORMAT(F10.3)
C
      WRITE(6,227) I
  227 FORMAT(2X,'ZS(',I2,')=')
C
      READ(*,228) ZS(I)
  228 FORMAT(F10.3)
C
      WRITE(6,229) I
  229 FORMAT(2X,'RS(',I2,')=')
C
      READ(*,241) RS(I)
  241 FORMAT(F10.3)
C
  222 CONTINUE
C
      WRITE(6,233)
  233 FORMAT(2X,' WHAT IS REPULSIVE FORCE CONSTANT (KO)?')
C
```

```
C
      READ(*,234) KO
  234 FORMAT(F10.3)
C
      WRITE(6,242)
  242 FORMAT(2X,' WHAT IS REPULSIVE FORCE DAMPING CONSTANT (B)?')
C
      READ(*,243) B
  243 FORMAT(F10.3)
C
      WRITE(6,235)
  235 FORMAT(2X,' WHAT IS RADIUS OF REPULSIVE FORCE (R)?')
C
      READ(*,236) R
  236 FORMAT(F10.3)
C
C
  723 CALL INVKIN(XD,YD,ZD,QD1,QD2,QD3,ANS)
      IF (ANS .EQ. 1.0) GO TO 114
      CALL INVKIN(XM,YM,ZM,QM1,QM2,QM3,ANS)
C
  114 IF (ANS .EQ. 1.0) THEN
        WRITE(6,51)
   51   FORMAT(2X,' ROBOT CAN NOT REACH ORIGINAL OR FINAL POSITION
     1  ,PLEASE TRY AGAIN. ')
        GO TO 113
      ENDIF
C
      IF(QD1 .GT. Q1MAX .OR. QD1 .LT. Q1MIN)GO TO 330
      IF(QD2 .GT. Q2MAX .OR. QD2 .LT. Q2MIN)GO TO 330
      IF(QD3 .GT. Q3MAX .OR. QD3 .LT. Q3MIN)GO TO 330
      GO TO 340
  330 WRITE(6,332)
  332 FORMAT(2X,'THE TARGET POSITION IS NOT IN WORK VOLUME. TRY AGAIN')
      GO TO 113
C
  340 IF(QM1 .GT. Q1MAX .OR. QM1 .LT. Q1MIN) GO TO 333
      IF(QM2 .GT. Q2MAX .OR. QM2 .LT. Q2MIN) GO TO 333
      IF(QM3 .GT. Q3MAX .OR. QM3 .LT. Q3MIN) GO TO 333
      GO TO 341
  333 WRITE(6,334)
  334 FORMAT(2X,'ORIGINAL POSITION OF ROBOT IS NOT IN WORK VOLUME.
     1 TRY AGAIN! ')
      GO TO 113
C
  341 WRITE(6,44) QD1,QD2,QD3,QM1,QM2,QM3
   44 FORMAT(2X,6F10.4)
C
      Q1(1)=QM1
      Q2(1)=QM2
      Q3(1)=QM3
C
C
```

```
      DQ1(1)=0.0
      DQ2(1)=0.0
      DQ3(1)=0.0
C
      WRITE(6,354)
  354 FORMAT(2X,' WHAT IS VISCOUS FRICTION CONSTANT (V)? ')
C
      READ(*,355) V
  355 FORMAT(F10.3)
C
      WRITE(6,356)
  356 FORMAT(2X,' WHAT IS COULOMB FRICTION CONSTANT (C)? ')
C
      READ(*,357) C
  357 FORMAT(F10.3)
C
      WRITE(6,358)
  358 FORMAT(2X,' MAXIMUM TORQUE FOR JOINT ONE IS ... ')
C
      READ(*,359) T1MAX
  359 FORMAT(F10.3)
C
      WRITE(6,401)
  401 FORMAT(2X,' MAXIMUM TORQUFOR JOINT TWO IS ... ')
C
      READ(*,402) T2MAX
  402 FORMAT(F10.3)
C
      WRITE(6,403)
  403 FORMAT(2X,' MAXIMUM TORQUE FOR JOINT THREE IS ...')
C
      READ(*,404) T3MAX
  404 FORMAT(F10.3)
C
  999 EX=XD-XM
      EY=YD-YM
      EZ=ZD-ZM
C
C
      IF (NOBS .NE. 1.) GO TO 824
      DO 231 I=1,NB
      COM(I)=RS(I)/((XM-XS(I))**2.+(YM-YS(I))**2.+(ZM-ZS(I))**2.)**.5
        XOF(I) = (XM-XS(I))*COM(I) + XS(I)
        YOF(I) = (YM-YS(I))*COM(I) + YS(I)
        ZOF(I) = (ZM-ZS(I))*COM(I) + ZS(I)
      ROF(I) = ((XM-XOF(I))**2.+(YM-YOF(I))**2.+(ZM-ZOF(I))**2.)**.5
  231 CONTINUE
C
      MIND=100.
      DO 232 I=1,NB
         IF(ROF(I) .LE. MIND) THEN
            XO=XOF(I)
            YO=YOF(I)
            ZO=ZOF(I)
```

```
            RO=ROF(I)
            MIND=ROF(I)
          ENDIF
  232 CONTINUE
C
      EXO=XM-XO
      EYO=YM-YO
      EZO=ZM-ZO
C
      IF (RO .GT. R) THEN
      FXO=0.0
      FYO=0.0
      FZO=0.0
      ENDIF
C
      IF (RO .LT. R) THEN
        FO=KO*((RO-R)**2.)
        FXO=FO*EXO/RO -XVD*B
        FYO=FO*EYO/RO -YVD*B
        FZO=FO*EZO/RO -ZVD*B
      ENDIF
C
C
  824 CALL JACOBIAN(Q1,Q2,Q3,II,JA)
C
C
      XVD=JA(1,1) * DQ1(II) +JA(1,2)*DQ2(II)+JA(1,3)*DQ3(II)
      YVD=JA(2,1) * DQ1(II) +JA(2,2)*DQ2(II)+JA(2,3)*DQ3(II)
      ZVD=JA(3,1) * DQ1(II) +JA(3,2)*DQ2(II)+JA(3,3)*DQ3(II)
C
      MV=(XVD*XVD+YVD*YVD+ZVD*ZVD)**.5
C
      IF (NOBS .NE. 1.) THEN
         FXO=0.0
         FYO=0.0
         FZO=0.0
      ENDIF
C
      FXP=EX*KP-XVD*KV+FXO
      FYP=EY*KP-YVD*KV+FYO
      FZP=EZ*KP-ZVD*KV+FZO
C
C
      CALL MVGX(W,TT,DQ1,DQ2,DQ3,Q1,Q2,Q3,II,MX,VX,GX)
C
  888 FX=FXP*MX(1,1)+FYP*MX(1,2)+FZP*MX(1,3)+VX(1)+GX(1)
      FY=FXP*MX(2,1)+FYP*MX(2,2)+FZP*MX(2,3)+VX(2)+GX(2)
      FZ=FXP*MX(3,1)+FYP*MX(3,2)+FZP*MX(3,3)+VX(3)+GX(3)
C
      IF (DQ1(II) .EQ. 0) F(1)=0.0
      IF (DQ1(II) .NE. 0) F(1)=DQ1(II)*V+C*DQ1(II)/ABS(DQ1(II))
      IF (DQ2(II) .EQ. 0) F(2)=0.0
      IF (DQ2(II) .NE. 0) F(2)=DQ2(II)*V+C*DQ2(II)/ABS(DQ2(II))
      IF (DQ3(II) .EQ. 0) F(3)=0.0
```

```
      IF (DQ3(II) .NE. 0) F(3)=DQ3(II)*V+C*DQ3(II)/ABS(DQ3(II))
C
      KQ1=0.0
      KQ2=0.0
      KQ3=0.0
      IF(Q1(II) .GT. Q1MAX .OR. Q1(II) .LT. Q1MIN) KQ1=KQ
      IF(Q2(II) .GT. Q2MAX .OR. Q2(II) .LT. Q2MIN) KQ2=KQ
      IF(Q3(II) .GT. Q3MAX .OR. Q3(II) .LT. Q3MIN) KQ3=KQ
C
      T1(II)=FX*JA(1,1)+FY*JA(2,1)+FZ*JA(3,1)+F(1)-DQ1(II)*KQ1
      T2(II)=FX*JA(1,2)+FY*JA(2,2)+FZ*JA(3,2)+F(2)-DQ2(II)*KQ2
      T3(II)=FX*JA(1,3)+FY*JA(2,3)+FZ*JA(3,3)+F(3)-DQ3(II)*KQ3
C
      IF (II .NE. 1) THEN
         DIFT1=AT1(II-1) - T1(II-1)
         DIFT2=AT2(II-1) - T2(II-1)
         DIFT3=AT3(II-1) - T3(II-1)
         IF (DIFT1 .NE. 0) DIFT1=1.+DIFT1/ABS(T1(II-1))
         IF (DIFT2 .NE. 0) DIFT2=1.+DIFT2/ABS(T2(II-1))
         IF (DIFT3 .NE. 0) DIFT3=1.+DIFT3/ABS(T3(II-1))
         DIFT=(DIFT1+DIFT2+DIFT3)/3.
         ENDIF
C
C CHECK THE MAXIMUM TORQUES FOR EACH JOINT.
C
C
      A1=FXP/FYP
      A2=FZP/FYP
      CALL TORQUE(A1,A2,T1,T2,T3,II,T1MAX,T2MAX,T3MAX,
     1 JA,MX,VX,GX,DQ1,DQ2,DQ3,KQ1,KQ2,KQ3,TT1,TT2,TT3)
C
      T1(II)=TT1
      T2(II)=TT2
      T3(II)=TT3
C
      CALL ARMDYN(C,V,W,T1,T2,T3,Q1,Q2,Q3,DQ1,DQ2,DQ3,II,QM1,QM2,QM3,
     1 DQM1,DQM2,DQM3,TT,F,AT1,AT2,AT3)
C
      II=II+1
C
      Q1(II)=QM1
      Q2(II)=QM2
      Q3(II)=QM3
C
      DQ1(II)=DQM1
      DQ2(II)=DQM2
      DQ3(II)=DQM3
C
      CALL KIN(Q1,Q2,Q3,II,XM,YM,ZM)
C
      IF (IP .EQ. 1) THEN
      WRITE(4,66) XM,YM
   66 FORMAT(2F14.5)
C
```

```
      IF (NOBS .EQ. 1.) THEN
      WRITE(4,67) XO,YO
   67 FORMAT(2F14.5)
      ENDIF
C
      WRITE(3,77) XM,ZM
   77 FORMAT(2F14.5)
C
      IF (NOBS .EQ. 1.) THEN
      WRITE(3,75) XO,ZO
   75 FORMAT(2F14.5)
      ENDIF
C
      WRITE(2,91) TT,Q1(II)
   91 FORMAT(2F14.5)
C
      WRITE(2,963) TT,Q2(II)
  963 FORMAT(2F14.5)
C
      WRITE(2,964) TT,Q3(II)
  964 FORMAT(2F14.5)
C
      IF (NOBS .EQ. 1.) THEN
      WRITE(2,92) ZO,YO
   92 FORMAT(2F14.5)
      ENDIF
C
      WRITE(1,88) TT,T1(II-1)
   88 FORMAT(2F14.5)
C
      WRITE(1,78) TT,T2(II-1)
   78 FORMAT(2F14.5)
C
      WRITE(1,79) TT,T3(II-1)
   79 FORMAT(2F14.5)
C
      WRITE(7,93) TT,XM
   93 FORMAT(2F14.5)
C
      WRITE(7,94) TT,YM
   94 FORMAT(2F14.5)
C
      WRITE(7,95) TT,ZM
   95 FORMAT(2F14.5)
C
      WRITE(10,811) XM,YM,ZM
  811 FORMAT(3F14.5)
C
      IF (NOBS .EQ. 1.) THEN
      WRITE(10,812) XO,YO,ZO
  812 FORMAT(3F14.5)
      ENDIF
C
      IF (II .NE. 1) THEN
```

```
      WRITE(8,876) TT,AT1(II-1)
  876 FORMAT(2F14.5)
C
      WRITE(8,875) TT,AT2(II-1)
  875 FORMAT(2F14.5)
C
      WRITE(8,874) TT,AT3(II-1)
  874 FORMAT(2F14.5)
C
      ENDIF
C
      WRITE(9,534) TT,MV
  534 FORMAT(2F14.5)
C
      ENDIF
C
      IP=IP+1
      IF(IP .EQ. 80) IP=1
C
      IF (TT .GT. TMAX) GO TO 111
      GO TO 999
C
C
  111 STOP
      END
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C  ALL SUBROUTINES ARE SAME AS R3DIMP.FOR PROGRAM          C
C  THEREFOR, THERE IS NO NEED TO REPEAT THEM. ONLY TWO     C
C  NEW SUBROUTINES ARE ADDED                               C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
      SUBROUTINE TORQUE(A1,A2,T1,T2,T3,II,T1MAX,T2MAX,T3MAX,
     1                  JA,MX,VX,GX,DQ1,DQ2,DQ3,KQ1,KQ2,KQ3,TT1,TT2,TT3)
C
C
      REAL*4 T1(20000),T2(20000),T3(20000),MX(3,3),VX(3),GX(3)
      REAL*4 A1,A2,A3,A4,A5,A6,TT1,TT2,TT3,JA(3,3),T1MAX,T2MAX,T3MAX
      REAL*4 FX,FY,FZ,DQ1(20000),DQ2(20000),DQ3(20000),KQ1,KQ2,KQ3
C
      TT1=T1(II)
      TT2=T2(II)
      TT3=T3(II)
C
   77 IF (ABS(TT1) .GT. T1MAX) THEN
         A3=JA(1,1)
         A4=JA(2,1)
         A5=JA(3,1)
         A6=DQ1(II)*KQ1
         TT1=T1MAX * TT1/ABS(TT1)
         CALL TORFIX(TT1,A1,A2,A3,A4,A5,A6,MX,VX,GX,II,FX,FY,FZ)
         TT2=FX*JA(1,2)+FY*JA(2,2)+FZ*JA(3,2)-DQ2(II)*KQ2
```

```
         TT3=FX*JA(1,3)+FY*JA(2,3)+FZ*JA(3,3)-DQ3(II)*KQ3
      ENDIF
C
   88 IF (ABS(TT2) .GT. T2MAX) THEN
         A3=JA(1,2)
         A4=JA(2,2)
         A5=JA(3,2)
         A6=DQ2(II)*KQ2
         TT2=T2MAX *TT2/ABS(TT2)
         CALL TORFIX(TT2,A1,A2,A3,A4,A5,A6,MX,VX,GX,II,FX,FY,FZ)
         TT1=FX*JA(1,1)+FY*JA(2,1)+FZ*JA(3,1)-DQ1(II)*KQ1
         TT3=FX*JA(1,3)+FY*JA(2,3)+FZ*JA(3,3)-DQ3(II)*KQ3
      ENDIF
C
   99 IF (ABS(TT3) .GT. T3MAX) THEN
         A3=JA(1,3)
         A4=JA(2,3)
         A5=JA(3,3)
         A6=DQ3(II)*KQ3
         TT3=T3MAX * TT3/ABS(TT3)
         CALL TORFIX(TT3,A1,A2,A3,A4,A5,A6,MX,VX,GX,II,FX,FY,FZ)
         TT1=FX*JA(1,1)+FY*JA(2,1)+FZ*JA(3,1)-DQ1(II)*KQ1
         TT2=FX*JA(1,2)+FY*JA(2,2)+FZ*JA(3,2)-DQ2(II)*KQ2
      ENDIF
C
      IF (ABS(TT1) .GT. T1MAX .AND. ABS(TT1) .GT. T2MAX) THEN
      IF (ABS(TT3) .GT. T3MAX) THEN
      TT1=T1MAX * TT1/ABS(TT1)
      TT2=T2MAX * TT2/ABS(TT2)
      TT3=T3MAX * TT3/ABS(TT3)
      ENDIF
      ENDIF
      IF (TT1 .GT. T1MAX) GO TO 77
      IF (TT2 .GT. T2MAX) GO TO 88
      IF (TT3 .GT. T3MAX) GO TO 99
C
      RETURN
      END
CC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
      SUBROUTINE TORFIX(T,A1,A2,A3,A4,A5,A6,MX,VX,GX,II,FX,FY,FZ)
C
      REAL*4 T,A1,A2,A3,MX(3,3),VX(3),GX(3),FX,FY,FZ
      REAL*4 A4,A5,A6,B1,B2,B3,B,C1,C2,C3,C,FXP,FYP,FZP
C
      B1=(VX(1)+GX(1))*A3
      B2=(VX(2)+GX(2))*A4
      B3=(VX(3)+GX(3))*A5
      B=T-B1-B2-B3+A6
C
      C1=(A1*MX(1,1)+MX(1,2)+A2*MX(1,3))*A3
      C2=(A1*MX(2,1)+MX(2,2)+A2*MX(2,3))*A4
      C3=(A1*MX(3,1)+MX(3,2)+A2*MX(3,3))*A5
```

```
      C=C1+C2+C3
C
      FYP=B/C
      FXP=A1*FYP
      FZP=A2*FYP
C
      FX=FXP*MX(1,1)+FYP*MX(1,2)+FZP*MX(1,3)+VX(1)+GX(1)
      FY=FXP*MX(2,1)+FYP*MX(2,2)+FZP*MX(2,3)+VX(2)+GX(2)
      FZ=FXP*MX(3,1)+FYP*MX(3,2)+FZP*MX(3,3)+VX(3)+GX(3)
C
      RETURN
      END
C
C
```

```
C*****************************************
C*                R3DC3.FOR              *
C*****************************************
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C IMPEDANCE CONTROL FOR 3 DOF ROBOT IN  C
C 3 DIMENSIONAL SPACE                   C
C                                       C
C ALSO INCLUDS: - SURFACE NAVIGATION    C
C                 SCHEME                C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
      REAL*4 XD,YD,ZD,QD1,QD2,QD3,XM,YM,ZM,QM1,QM2,QM3,Q1(20000)
      REAL*4 Q2(20000),Q3(20000),DQ1(20000),DQ2(20000),DQ3(20000)
      REAL*4 EX,EY,EZ,JA(3,3),XVD,YVD,ZVD,FXP,FYP,FZP,KP,KV,MX(3,3)
      REAL*4 VX(3),GX(3),FX,FY,FZ,T1(20000),T2(20000),T3(20000)
      REAL*4 TT,W,TMAX,DQM1,DQM2,DQM3,ANS,KQ,KQ1,KQ2,KQ3
      REAL*4 Q1MAX,Q1MIN,Q2MAX,Q2MIN,Q3MAX,Q3MIN,B,FO
      REAL*4 KO,R,RO,ROF(10),RS(10),XO,YO,ZO,XOF(10),YOF(10)
      REAL*4 ZOF(10),XS(10),YS(10),ZS(10),COM(10),MIND,FXO,FYO,FZO
      REAL*4 EXO,EYO,EZO,KC,ANS2,XMM,YMM,ZMM,XDD,YDD,ZDD,AA,BB,CC
      REAL*4 AAA,BBB,GGG,MM,NN,P,ZC,XC,YC,TC,EXX,EYY,EZZ
      REAL*4 XDM,XCM,YCM,ZCM,YDM,ZDM,XOM,YOM,ZOM,N1,N2,N3,P1,P2,R1
C
      INTEGER NB,IP
C
      ANS2=0.0
      IP=1
      Q1MAX=1.92
      Q1MIN=-1.92
      Q2MAX=1.484
      Q2MIN=-.47
      Q3MAX=0.0
      Q3MIN=-2.13
C
      L1=.19
      L2=.3
      L3=.3
      M1=2.
      M2=2.
      M3=2.
C
      TT=0.0
      KV=4.
      KP=4.
      II=1
C
C
  113 WRITE(6,200)
  200 FORMAT(2X,' XD=')
```

```
C
      READ(*,210) XD
  210 FORMAT(F12.4)
C
      WRITE(6,220)
  220 FORMAT(2X,' YD=')
C
      READ(*,230) YD
  230 FORMAT(F12.4)
C
      WRITE(6,240)
  240 FORMAT(2X,' ZD=')
C
      READ(*,250) ZD
  250 FORMAT(F12.4)
C
      WRITE(6,260)
  260 FORMAT(2X,'XM=')
C
      READ(*,270) XM
  270 FORMAT(F12.4)
C
      WRITE(6,280)
  280 FORMAT(2X,'YM=')
C
      READ(*,290) YM
  290 FORMAT(F12.4)
C
      WRITE(6,300)
  300 FORMAT(2X,'ZM=')
C
      READ(*,310) ZM
  310 FORMAT(F12.4)
C
      WRITE(6,411)
  411 FORMAT(2X,' WHAT IS MAXIMUM TIME? ')
C
      READ(*,412) TMAX
  412 FORMAT(F10.3)
C
      WRITE(6,413)
  413 FORMAT(2X,' WHAT IS PAYLOAD WEIGHT? ')
C
      READ(*,414) W
  414 FORMAT(F10.3)
C
      WRITE(6,342)
  342 FORMAT(2X,' WHAT IS ANGLE CONSTANT (KQ)? ')
C
      READ(*,343) KQ
  343 FORMAT(F10.3)
C
      WRITE(6,601)
  601 FORMAT(2X,' HOW MANY OBSTACLES? ')
```

```
C
      READ(*,602) NB
  602 FORMAT(I2)
C
      DO 222 I=1,NB
      WRITE(6,223) I
  223 FORMAT(2X,'XS(',I2,')= ')
C
      READ(*,224) XS(I)
  224 FORMAT(F10.3)
C
      WRITE(6,225) I
  225 FORMAT(2X,'YS(',I2,')=')
C
      READ(*,226) YS(I)
  226 FORMAT(F10.3)
C
      WRITE(6,227) I
  227 FORMAT(2X,'ZS(',I2,')=')
C
      READ(*,228) ZS(I)
  228 FORMAT(F10.3)
C
      WRITE(6,229) I
  229 FORMAT(2X,'RS(',I2,')=')
C
      READ(*,241) RS(I)
  241 FORMAT(F10.3)
C
  222 CONTINUE
C
      WRITE(6,233)
  233 FORMAT(2X,' WHAT IS REPULSIVE FORCE CONSTANT (KO)?')
C
C
      READ(*,234) KO
  234 FORMAT(F10.3)
C
      WRITE(6,242)
  242 FORMAT(2X,' WHAT IS REPULSIVE FORCE DAMPING CONSTANT (B)?')
C
      READ(*,243) B
  243 FORMAT(F10.3)
C
      WRITE(6,235)
  235 FORMAT(2X,' WHAT IS RADIUS OF REPULSIVE FORCE (R)?')
C
      READ(*,236) R
  236 FORMAT(F10.3)
C
      WRITE(6,345)
  345 FORMAT(2X,' WHAT IS COSTAL NAVIGATION CONSTANT (KC)? ')
C
```

```
      READ(*,346) KC
  346 FORMAT(F10.3)
C
      CALL INVKIN(XD,YD,ZD,QD1,QD2,QD3,ANS)
      IF (ANS .EQ. 1.0) GO TO 114
      CALL INVKIN(XM,YM,ZM,QM1,QM2,QM3,ANS)
C
  114 IF (ANS .EQ. 1.0) THEN
        WRITE(6,51)
   51   FORMAT(2X,' ROBOT CAN NOT REACH ORIGINAL OR FINAL POSITION
     1  ,PLEASE TRY AGAIN. ')
        GO TO 113
      ENDIF
C
      IF(QD1 .GT. Q1MAX .OR. QD1 .LT. Q1MIN)GO TO 330
      IF(QD2 .GT. Q2MAX .OR. QD2 .LT. Q2MIN)GO TO 330
      IF(QD3 .GT. Q3MAX .OR. QD3 .LT. Q3MIN)GO TO 330
      GO TO 340
  330 WRITE(6,332)
  332 FORMAT(2X,'THE TARGET POSITION IS NOT IN WORK VOLUME. TRY AGAIN')
      GO TO 113
C
  340 IF(QM1 .GT. Q1MAX .OR. QM1 .LT. Q1MIN) GO TO 333
      IF(QM2 .GT. Q2MAX .OR. QM2 .LT. Q2MIN) GO TO 333
      IF(QM3 .GT. Q3MAX .OR. QM3 .LT. Q3MIN) GO TO 333
      GO TO 341
  333 WRITE(6,334)
  334 FORMAT(2X,'ORIGINAL POSITION OF ROBOT IS NOT IN WORK VOLUME.
     1 TRY AGAIN! ')
      GO TO 113
C
  341 WRITE(6,44) QD1,QD2,QD3,QM1,QM2,QM3
   44 FORMAT(2X,6F10.4)
C
      Q1(1)=QM1
      Q2(1)=QM2
      Q3(1)=QM3
C
C
      DQ1(1)=0.0
      DQ2(1)=0.0
      DQ3(1)=0.0
C
C
  999 EX=XD-XM
      EY=YD-YM
      EZ=ZD-ZM
C
C
      DO 231 I=1,NB
      COM(I)=RS(I)/((XM-XS(I))**2.+(YM-YS(I))**2.+(ZM-ZS(I))**2.)**.5
        XOF(I) = (XM-XS(I))*COM(I) + XS(I)
        YOF(I) = (YM-YS(I))*COM(I) + YS(I)
        ZOF(I) = (ZM-ZS(I))*COM(I) + ZS(I)
```

```
      ROF(I) = ((XM-XOF(I))**2.+(YM-YOF(I))**2.+(ZM-ZOF(I))**2.)**.5
  231 CONTINUE
C
      MIND=100.
      DO 232 I=1,NB
         IF(ROF(I) .LE. MIND) THEN
            XO=XOF(I)
            YO=YOF(I)
            ZO=ZOF(I)
            IJ=I
            RO=ROF(I)
            MIND=ROF(I)
         ENDIF
  232 CONTINUE
C
C
      EXO=XM-XO
      EYO=YM-YO
      EZO=ZM-ZO
C
C
      IF (RO .GT. R) THEN
      FXO=0.0
      FYO=0.0
      FZO=0.0
      ANS2=0.0
      ENDIF
C
      IF (RO .LT. R) THEN
        FO=KO*((RO-R)**2.)
        FXO=FO*EXO/RO -XVD*B
        FYO=FO*EYO/RO -YVD*B
        FZO=FO*EZO/RO -ZVD*B
C
C
        IF (ANS2 .EQ. 1.) THEN
C
          XDM=XD-XM
          YDM=YD-YM
          ZDM=ZD-ZM
          XOM=XO-XM
          YOM=YO-YM
          ZOM=ZO-ZM
C
          N1=(YDM*ZOM-YOM*ZDM)
          N2=(XOM*ZDM-XDM*ZOM)
          N3=(XDM*YOM-YOM*YDM)
C
C
               XCM=YOM*N3-N2*ZOM
               YCM=N1*ZOM-XOM*N3
               ZCM=N2*XOM-N1*YOM
C
          R1=((XCM)**2 + (YCM)**2 + (ZCM)**2 )**.5
```

```
C
          FXO=(XCM)*KC/R1
          FYO=(YCM)*KC/R1
          FZO=(ZCM)*KC/R1
          WRITE(6,956)
  956     FORMAT(' ','SURFACE NAVIGATION')
C
C
        ENDIF
      ENDIF
C
C
      CALL JACOBIAN(Q1,Q2,Q3,II,JA)
C
C
      XVD=JA(1,1) * DQ1(II) +JA(1,2)*DQ2(II)+JA(1,3)*DQ3(II)
      YVD=JA(2,1) * DQ1(II) +JA(2,2)*DQ2(II)+JA(2,3)*DQ3(II)
      ZVD=JA(3,1) * DQ1(II) +JA(3,2)*DQ2(II)+JA(3,3)*DQ3(II)
C
C
C
      FXP=EX*KP-XVD*KV+FXO
      FYP=EY*KP-YVD*KV+FYO
      FZP=EZ*KP-ZVD*KV+FZO
C
      IF (ANS2 .EQ. 1.) THEN
          FXP=FXO-XVD*KV
          FYP=FYO-YVD*KV
          FZP=FZO-ZVD*KV
      ENDIF
C
C
      CALL MVGX(W,TT,DQ1,DQ2,DQ3,Q1,Q2,Q3,II,MX,VX,GX)
C
C
  888 FX=FXP*MX(1,1)+FYP*MX(1,2)+FZP*MX(1,3)+VX(1)+GX(1)
      FY=FXP*MX(2,1)+FYP*MX(2,2)+FZP*MX(2,3)+VX(2)+GX(2)
      FZ=FXP*MX(3,1)+FYP*MX(3,2)+FZP*MX(3,3)+VX(3)+GX(3)
C
C
      KQ1=0.0
      KQ2=0.0
      KQ3=0.0
      IF(Q1(II) .GT. Q1MAX .OR. Q1(II) .LT. Q1MIN) KQ1=KQ
      IF(Q2(II) .GT. Q2MAX .OR. Q2(II) .LT. Q2MIN) KQ2=KQ
      IF(Q3(II) .GT. Q3MAX .OR. Q3(II) .LT. Q3MIN) KQ3=KQ
C
      T1(II)=FX*JA(1,1)+FY*JA(2,1)+FZ*JA(3,1)-DQ1(II)*KQ1
      T2(II)=FX*JA(1,2)+FY*JA(2,2)+FZ*JA(3,2)-DQ2(II)*KQ2
      T3(II)=FX*JA(1,3)+FY*JA(2,3)+FZ*JA(3,3)-DQ3(II)*KQ3
C
C
C
```

```
      CALL ARMDYN(W,T1,T2,T3,Q1,Q2,Q3,DQ1,DQ2,DQ3,II,QM1,QM2,QM3,
     1 DQM1,DQM2,DQM3,TT)
C
      II=II+1
C
      Q1(II)=QM1
      Q2(II)=QM2
      Q3(II)=QM3
C
      DQ1(II)=DQM1
      DQ2(II)=DQM2
      DQ3(II)=DQM3
C
      XM1=XM
      YM1=YM
      ZM1=ZM
C
      CALL KIN(Q1,Q2,Q3,II,XM,YM,ZM)
C
      EXX=ABS(XM1-XM)
      EYY=ABS(YM1-YM)
      EZZ=ABS(ZM1-ZM)
C
      IF(EXX .GE. .0001 .OR. EYY .GE. .0001) TC=0.0
         IF(EZZ .GE. .0001) TC=0.0
C
      IF(EXX .LT. .0001 .AND. EYY .LT. .0001) THEN
      IF (EZZ .LT. .0001) THEN
         TC=TC+.005
      ENDIF
      ENDIF
      IF (TC .GE. 2.) ANS2=1.
C
      IF (IP .EQ. 1) THEN
      WRITE(4,66) XM,YM
   66 FORMAT(2F14.5)
C
      WRITE(4,67) XO,YO
   67 FORMAT(2F14.5)
C
      WRITE(3,77) XM,ZM
   77 FORMAT(2F14.5)
C
      WRITE(2,75) TT,T1(II-1)
   75 FORMAT(2F14.5)
C
      WRITE(2,91) TT,T2(II-2)
   91 FORMAT(2F14.5)
C
      WRITE(2,92) TT,T3(II-3)
   92 FORMAT(2F14.5)
C
      WRITE(1,88) TT,Q1(II)
   88 FORMAT(2F14.5)
```

```
C
      WRITE(1,78) TT,Q2(II)
   78 FORMAT(2F14.5)
C
      WRITE(1,79) TT,Q3(II)
   79 FORMAT(2F14.5)
C
      WRITE(7,93) TT,XM
   93 FORMAT(2F14.5)
C
      WRITE(7,94) TT,YM
   94 FORMAT(2F14.5)
C
      WRITE(7,95) TT,ZM
   95 FORMAT(2F14.5)
C
      WRITE(8,828) XM,YM,ZM
  828 FORMAT(3F14.5)
C
      WRITE(8,812) XO,YO,ZO
  812 FORMAT(3F14.5)
C
      ENDIF
C
      IP=IP+1
      IF(IP .EQ. 5) IP=1
C
  919 IF (TT .GT. TMAX) GO TO 111
      GO TO 999
C
C
  111 STOP
      END
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C  ALL SUBROUTINES ARE SAME AS R3DIMP.FOR PROGRAM    C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
```