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 dactylogravées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopy de qualité inférieure.

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 fulfillment of the
requirements for the degree of
Master of Applied Science

in

Mechanical Engineering

Department of Mechanical Engineering
University of Ottawa

C Rahim Jassemi-Zargani, Ottawa, Canada, 1990
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.
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. Neculescu 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

Abstract ................................................................. iii
Acknowledgement ...................................................... iv
List of Figures ........................................................... vii
Nomenclature ............................................................. x
1. Introduction ........................................................... 1
2. Literature Survey ................................................... 3
3. Impedance Control of Robot Motion .............................. 7
   (3.1) Introductory Remarks ....................................... 7
   (3.2) Attractive Force Field ....................................... 16
   (3.3) Robot WorkVolume .......................................... 19
   (3.4) Joint Torque Limits ......................................... 20
   (3.5) Moving Targets ............................................. 24
4. Collision Avoidance of the Robot Arm with Obstacles 26
   (4.1) Introductory Remarks ....................................... 26
   (4.2) Repulsive Field of Forces .................................. 29
   (4.3) Contribution of Damping in Repulsive Fields ........... 37
   (4.4) Torque Saturation Limits ................................... 45
   (4.5) Coastal(CNS) and Surface(SNS) Navigation Scheme 48
   (4.6) Work Volume Limits ......................................... 58
5. Behavior of Dynamic Terms in Impedance Control .......... 60
6. Conclusions .......................................................... 75
References ............................................................... 78
Appendix A  Dynamics and Kinematics equations of robot  81

Appendix B  Computer code: Impedance controller for .. 105 robots
LIST OF FIGURES

2.1 .... Khatib's control scheme for collision avoidance using APF .......... 4

2.2 .... Hogan's impedance control scheme with force measurement ..... 5

2.3 .... Kazerooni's impedance control scheme applied to a joint space controlled manipulator 6

2.4 .... Impedance control scheme used for simulations 6

3.1.1 .. End-effector trajectory of a 2 DOF robot .......... 10

3.1.2 .. Simulation results of the robot in figure 3.1.1 11

3.1.3 .. End-effector trajectory of a 3 DOF robot .......... 12

3.1.4 .. Simulation results of the robot shown in figure 3.1.3 13

3.1.5 .. End-effector trajectory for 2 DOF planar robot with PD controller 14

3.1.6 .. Simulation results of the robot shown in figure 3.1.5 15

3.2.1 .. Artificial mechanical impedance B-K system ...... 16

3.4.1 .. End-effector trajectory when one joint torque reaches the saturation point 22

3.4.2 .. Joint torques for the case (a) and (b) of figure 3.4.1 23

3.5.1 .. Interception of a target moving on a straight line 25

3.5.2 .. Interception of a target moving on a circular path at low angular speed 25

3.5.3 .. Interception of a target moving on a circular path at high angular speed 25

4.1.1 .. Attractive and Repulsive force field ............... 28

4.2.1 .. Repulsive and attractive forces ..................... 31

4.2.2 .. End-effector trajectory of a robot which avoids an obstacle 32
4.2.3 Simulation results of the robot shown in figure 4.2.2
4.2.4 End-effector trajectory ID of a robot avoiding an obstacle 0.
4.2.5 Simulation results of the robot shown in figure 4.2.4

4.3.1 Robot trajectory without damping in the repulsive field
4.3.2 Robot trajectory with damping in the repulsive field
4.3.3 Comparison of the effect of damping in the robot trajectory for the cases shown in figures 4.3.1 and 4.3.2.
4.3.4 The trajectory ID of a 3 DOF robot in the presence of an obstacle 0, when the repulsive forces have a damping component

4.3.5 Simulation results for the case shown in figure 4.3.4
4.4.1 Robot trajectory with in saturation limits of joint

4.4.2 Simulation results for the case shown in figure 4.4.1
4.5.1 Coastal Navigation Scheme (CNS) ...................... 51
4.5.2 End-Effector trajectory of a robot avoiding two obstacles
4.5.3 Simulation results of the robot shown in figure 4.5.2
4.5.4 Surface Navigation Scheme (SNS) ...................... 53
4.5.5 A first example of the surface navigation scheme 54
4.5.6 Simulation results of the robot shown in figure 4.5.5
4.5.7 A second example of surface navigation scheme 56
4.5.8 Simulation results of the robot shown in figure 4.5.7

4.6.1 End-effector trajectory in work volume limits

4.6.2 Simulation results of the robot shown in figure 4.6.1

5.0.1 End-effector trajectory when \( M_x(\theta)\) = 50% of actual \( M_x(\theta)\)

5.0.2 Simulation results of the robot shown in figure 5.0.1

5.0.3 End-effector trajectory when \( M_x(\theta)\) = 150% of actual \( M_x(\theta)\)

5.0.4 Simulation results of the robot shown in figure 5.0.3

5.0.4 End-effector trajectory of robot when \( V_x(\theta, \theta)\) = 0

5.0.5 Simulation results of the robot shown in figure 5.0.4

5.0.6 End-effector trajectory of a robot when \( G_x = 0\)

5.0.7 Simulation results of the robot shown in figure 5.0.6

5.0.8 End-effector trajectory of a robot when \( F_x = 0\)

5.0.9 Simulation results of the robot shown in figure 5.0.8

5.0.10 End-effector trajectory of a robot avoiding an obstacle when \( F_x = 0\)

5.0.11 Simulation results of the robot shown in figure 5.0.10
**NOMENCLATURE**

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>virtual damper constant</td>
</tr>
<tr>
<td>D</td>
<td>desired position of end-effector</td>
</tr>
<tr>
<td>D₀</td>
<td>starting point of moving target</td>
</tr>
<tr>
<td>D₁</td>
<td>interception point of moving target</td>
</tr>
<tr>
<td>E</td>
<td>equilibrium point</td>
</tr>
<tr>
<td>f</td>
<td>force-torque vector acting on the end-effector</td>
</tr>
<tr>
<td>F&lt;sub&gt;C&lt;/sub&gt;</td>
<td>coastal navigation force</td>
</tr>
<tr>
<td>F₀</td>
<td>repulsive force</td>
</tr>
<tr>
<td>F&lt;sub&gt;₀x&lt;/sub&gt;</td>
<td>repulsive force, x-axis component</td>
</tr>
<tr>
<td>F&lt;sub&gt;₀y&lt;/sub&gt;</td>
<td>repulsive force, y-axis component</td>
</tr>
<tr>
<td>F&lt;sub&gt;S&lt;/sub&gt;</td>
<td>surface navigation force</td>
</tr>
<tr>
<td>F(θ,θ̇)</td>
<td>friction term in joint space</td>
</tr>
<tr>
<td>F&lt;sub&gt;ₓ(θ,θ̇)&lt;/sub&gt;</td>
<td>friction term in cartesian space</td>
</tr>
<tr>
<td>F&lt;sub&gt;ₓD&lt;/sub&gt;</td>
<td>attractive force</td>
</tr>
<tr>
<td>G(θ)</td>
<td>gravitation term in joint space</td>
</tr>
<tr>
<td>G&lt;sub&gt;ₓ(θ)&lt;/sub&gt;</td>
<td>gravitation term in Cartesian space</td>
</tr>
<tr>
<td>I</td>
<td>identity matrix</td>
</tr>
<tr>
<td>J(θ)</td>
<td>Jacobian matrix</td>
</tr>
<tr>
<td>J&lt;sup&gt;T(θ)&lt;/sup&gt;</td>
<td>transposed Jacobian matrix</td>
</tr>
<tr>
<td>J̇</td>
<td>derivative of the Jacobian matrix</td>
</tr>
<tr>
<td>K&lt;sub&gt;C&lt;/sub&gt;</td>
<td>coastal navigation force constant</td>
</tr>
<tr>
<td>K₀</td>
<td>repulsive force constant</td>
</tr>
<tr>
<td>K&lt;sub&gt;q&lt;/sub&gt;</td>
<td>joint limits virtual damping constant</td>
</tr>
<tr>
<td>x</td>
<td></td>
</tr>
</tbody>
</table>
\( K_s \)  
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

\( \theta \)  
joint angular position

\( \dot{\theta} \)  
joint angular velocity

\( R \)  
the radius of effectiveness of repulsive force

\( \mathcal{V}(\theta, \dot{\theta}) \)  
centrifugal and Coriolis forces term in joint space

\( \mathcal{V}_X(\theta, \dot{\theta}) \)  
centrifugal and Coriolis forces term in Cartesian space

\( \omega_n \)  
natural frequency

\((X,Y,Z)_m\)  
manipulator end-effector current position

OR \( X_m, Y_m, Z_m \)

\((X,Y,Z)_d\)  
manipulator end-effector desired position

OR \( X_d, Y_d, Z_d \)

\((X,Y,Z)_o\)  
purpose of closest point of the obstacle to the end-effector

OR \( X_o, Y_o, Z_o \)
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 [23]. 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 approaches, developed in the last decade. These control approaches 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).

![Diagram](image)

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 generalized force $f_{\text{ext}}$ as desired. [14-17]

![Figure 2.3 Kazerooni's impedance control scheme applied to a joint space controlled manipulator](image)

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

![Figure 2.4 Impedance control scheme used for simulations](image)
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\( (\tau) \):

\[
\tau = J^T f
\]

(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:

\[ \tau = J^T \left[ M_x(\theta) \ddot{x} + V_x(\theta, \dot{\theta}) + G_x(\theta) + f_x(\theta, \dot{\theta}) \right] \] (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. \((K=4 \text{ N/m, } B=4 \text{ N/(m/s)}, \ W=0.0, \ K_q=10.0 \text{ N/(rad/s)})\)
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 $\ddot{X}_d - \ddot{X}_m$. The virtual damper results from a relation between the force and the velocity $\dot{X}_m$ 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:

$$\ddot{F} = K (\ddot{X}_d - \ddot{X}_m) - B \dddot{X}_m$$

where:

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

$$X = KIN \{\theta\}$$

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, \( \omega_n \), a desired damping coefficient of the system:

\[
\omega_n = \sqrt{\frac{K}{M}} \tag{3.2.4}
\]

\[
\zeta = \frac{B}{2\sqrt{K}} \tag{3.2.5}
\]

For critical damping \( \zeta = 1 \) we obtain:

\[
B = 2\sqrt{K} \tag{3.2.6}
\]

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:

\[
\begin{align*}
\tau &= J^T f & \text{MINQ} < \theta < \text{MAXQ} \\
\tau &= J^T f - \theta K_q & \text{MINQ} \geq \theta \geq \text{MAXQ}
\end{align*}
\]

where: \( \theta \) = 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.

Therefore, from equation 3.1.2:

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

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:

\[ \tau_1 = T1\text{MAX} \]

the resultant force is:

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

\( \tau_2 \) 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 Rescaling 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:

$$\vec{F} = K (\vec{X}_d - \vec{X}_m) + B (\vec{X}_d' - \vec{X}_m')$$  \hspace{1cm} (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 \(D_0\). The target is intercepted at \(D_1\).

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 \(D_1\).

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 straight 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), \(\omega=0.3\) 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), \(\omega=3\) 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 preprogrammed 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" Ultrasound 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

\[
\begin{align*}
\vec{F}_0 &= (F_{0x}) \vec{i} + (F_{0y}) \vec{j} & & R_{om} < R_o \\
\vec{F}_0 &= 0 & & R_{om} > R_o
\end{align*}
\]  

(4.2.1)

\[
X_{om} = X_m - X_o \\
Y_{om} = Y_m - Y_o
\]

\[
F_{0x} = K_o \left( \frac{2}{(R_{om} - R)} \right) \frac{X_{om}}{R_{om}}
\]  

(4.2.2)

\[
F_{0y} = K_o \left( \frac{2}{(R_{om} - R)} \right) \frac{Y_{om}}{R_{om}}
\]  

(4.2.3)

where:

\[
X_{om} = X_m - X_o \\
Y_{om} = Y_m - Y_o \\
\vec{R}_{om} = (X_m - X_o) \vec{i} + (Y_m - Y_o) \vec{j}
\]  

(4.2.4)
\[ \bar{R}_{om} = x_{om} \bar{i} + y_{om} \bar{j} \]  
(4.2.5)

\[ R_{om} = \sqrt{x_{om}^2 + y_{om}^2} \]  
(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 feasible 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 angular positions.

![Diagram showing repulsive and attractive forces](image)

**Figure 4.2.1 Repulsive and attractive forces**
Figure 4.2.2  End-effector trajectory of a robot which avoids an obstacle. 
\((K=4 \text{ N/m, } B=4 \text{ N/(m/s)}, R=0.1 \text{ m, } K_o=500 \text{ N/m, } B_o=20 \text{ N/(m/s)}, K_c=0.0, K_q=0.0)^5\)
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:

\[
F_{ox} = K_o (R_{om} - R)^2 \frac{X_{om}}{R_{om}} - \dot{X}_m B_o
\]

\[
F_{oy} = K_o (R_{om} - R)^2 \frac{Y_{om}}{R_{om}} - \dot{Y}_m B_o
\]
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=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 the repulsive field (same parameters as figure 4.3.1 except $B_0=20.0 \text{ N/(m/s)}$)
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, $W=0.0$, $K_0=10$ 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=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 $X \neq X_d$. 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:

$$
\hat{F}_c = K_c \frac{|\hat{R}_m|}{|r|} \hat{P}
$$

(4.5.1)

where:

$$
|\hat{P}| = 1
$$
\[ \vec{P} \cdot \vec{R}_{om} = 0 \]  \hspace{1cm} (4.5.2)

\( K_c \) = coastal navigation constant

\( \vec{F}_c \) = coastal navigation force vector

\( r \) = limit of repulsive field

\[ \vec{R}_{om} = \vec{R}_m - \vec{R}_o \]

- 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 successfully 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):

\[ \vec{F}_s = K_s \frac{\vec{V}_s}{|\vec{V}_s|} \]  \hspace{1cm} (4.5.3)

where:

\[ \vec{N} = \vec{V}_2 \times \vec{V}_1 \]  \hspace{1cm} (4.5.4)

\[ \vec{V}_s = \vec{V}_1 \times \vec{N} \]  \hspace{1cm} (4.5.5)

\[ \vec{V}_1 = \vec{R}_{om} = \vec{X}_m - \vec{X}_o \]
\[ V_2 = \vec{R}_{md} = \vec{x}_d - \vec{x}_m \]
\[ V_3 = \vec{R}_{ms} = \vec{x}_s - \vec{x}_m \]
\[ F_s = \text{surface navigation force} \]
\[ K_s = \text{surface navigation constant} \]

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 feasible 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 $K_c = 4 \text{ N/m}$)
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
\((X,Y,Z)_s=(0.4,0.0,0.25) \text{ m}, R_s=0.08 \text{ m})\)
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_0=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 $K_q=2 \text{ N/(rad/s)}$)

58
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 \( M_x(\theta) = 50\% \) of actual \( M_x(\theta) \). 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 $M_x(\theta)$)
($K=4 \text{ N/m}, B=4 \text{ N/(m/s)}, W=0.0$)
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
\(M_x(\theta) = 150\% \text{ of actual } M_x(\theta)\)

\(K=4 \text{ N/m, } B=4 \text{ N/(m/s), } W=0.0\)
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 $V_x(\theta,\dot{\theta}) = 0$

($K=4 \text{ N/m}, B=4 \text{ N/(m/s)}, \omega=0.0, K_q=5 \text{ N/(rad/s)}$)
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 $G_x(\theta) = 0$

$K=4 \text{ N/m, } B=4 \text{ N/(m/s), } k_q=5 \text{ N/(rad/s)}$

$v=0, (X,Y,Z)_d = (0.5, 0.3, 0.35) \text{ m,}$

$(X,Y,Z)_n = (0.3, 0.2, 0.25) \text{ m}$
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 $F_x(\theta,\phi) = 0$

$(K=4 \text{ N/m}, B=4 \text{ N/(m/s)}, \bar{w}=0.0, c=0.0, \nu=0.1 \text{ N/(rad/s)}, K_0=5 \text{ N/(rad/s)})$
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) Joint 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


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 define the link parameters in terms of link frames. The definition of link parameters follows:

Denavit-Hartenberg convention

\[ a_1 = \text{the distance from } z_1 \text{ to } z_{1+1} \text{ measured along } x_1 \]
\[ \alpha_1 = \text{the angle between } z_1 \text{ and } z_{1+1} \text{ measured about } x_1 \]
\[ d_1 = \text{the distance from } x_{1-1} \text{ to } x_1 \text{ measured along } z_1 \]
\[ \theta_1 = \text{the angle between } x_{1-1} \text{ to } x_1 \text{ measured about } z_1 \]
\[ C_1 = \cos(\theta_1) \]
\[ S_1 = \sin(\theta_1) \]
\[ C_{12} = \cos(\theta_1 + \theta_2) \]
\[ S_{12} = \sin(\theta_1 + \theta_2) \]

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

\[
{}^{1-1}_1 T = \begin{bmatrix}
C_\theta_1 & -S_\theta_1 & 0 & a_1 \\
S_\theta_1 C_\alpha_{1-1} & C_\theta_1 C_\alpha_{1-1} & -S_\alpha_{1-1} & -S_\alpha_{1-1} d_1 \\
S_\theta_1 S_\alpha_{1-1} & C_\theta_1 S_\alpha_{1-1} & C_\alpha_{1-1} & C_\alpha_{1-1} d_1 \\
0 & 0 & 0 & 1
\end{bmatrix}
\]

(A.1.1.1)
The combined transformation that relates directly the frame \{N\} to the frame \{0\} is given by:

\[
^0 T_N = ^0 T_1 T_2 \ldots \ldots T_{N-1} T_N \quad (A.1.1.2)
\]
The link parameters of 3 DOF robot are:

<table>
<thead>
<tr>
<th></th>
<th>$\alpha_{1-1}$</th>
<th>$a_1$</th>
<th>$d_1$</th>
<th>$\theta_1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>$L_1$</td>
<td>$\theta_1$</td>
</tr>
<tr>
<td>2</td>
<td>90</td>
<td>0</td>
<td>0</td>
<td>$\theta_2$</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>$L_2$</td>
<td>0</td>
<td>$\theta_3$</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>$L_3$</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

The transformation matrices are given by:

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

$$
^{2}_{3}T = \begin{bmatrix} C_3 & -S_3 & 0 & L_2 \\ S_3 & C_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad ^{3}_{4}T = \begin{bmatrix} 1 & 0 & 0 & L_3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$
The resulting transformation matrix between frame 4 and frame 0 is:

\[
0^4_T = \begin{bmatrix}
C_1C_{23} & -C_1S_{23} & S_1 & L_3C_1C_{23} + L_2C_1C_2 \\
S_1C_{23} & -S_1S_{23} & -C_1 & L_3S_1C_{23} + L_2S_1C_2 \\
S_{23} & C_{23} & 0 & L_3S_{23} + L_2S_2 + L_1 \\
0 & 0 & 0 & 1
\end{bmatrix}
\]

(A.1.1.3)

The kinematics equations are:

\[
\begin{align*}
X &= L_3C_1C_{23} + L_2C_1C_2 \\
Y &= L_3S_1C_{23} + L_2S_1C_2 \\
Z &= L_3S_{23} + L_2S_2 + L_1
\end{align*}
\]

(A.1.1.4)  
(A.1.1.5)  
(A.1.1.6)

A.1.2 INVERSE KINEMATICS:

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

from equation A.1.1.4 :
\[ C_1 = X \left( L_3 C_{23} + L_2 C_2 \right)^{-1} \]
\[ S_1 = Y \left( L_3 C_{23} + L_2 C_2 \right)^{-1} \]
\[ \theta_1 = \text{ATAN2} \left( S_1, C_1 \right) \]

OR

\[ \theta_1 = \text{ATAN2} \left( Y, X \right) \quad (A.1.2.1) \]

In order to calculate angle \( \theta_3 \), we square both sides of equations A.1.1.5 and A.1.1.6 and add them:

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

\[ C = \sqrt{\left( 2L_2 L_3 \right)^2 - \left( K - L_2^2 - L_3^2 \right)^2} \]

\[ D = K - L_3^2 - L_2^2 \]

\[ \theta_3 = \text{ATAN2} \left( C, D \right) \quad (A.1.2.2) \]
In order to calculate $\theta_2$, from equations A.1.1.535 and A.1.2.2 we used

$$A = (Z-L_1) L_3 S_3 + S_1 \left( L_3 C_3 + L_2 \right)$$

$$B = (L_3 S_3)^2 + (L_3 C_3 + L_2)^2$$

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

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

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

\[
1^+ 1\omega_{1+1} = 1^+ 1R_{1} 1\omega_{1} + \theta_{1+1} 1^+ 1Z_{1+1} \tag{A.1.3.1}
\]

- for angular acceleration

\[
1^+ 1\omega_{1+1} = 1^+ 1R_{1} \dot{1}\omega_{1} + \theta_{1+1} 1^+ 1\omega_{1} \times \theta_{1+1} 1^+ 1Z_{1+1} + \theta_{1+1} \cdot 1^+ 1Z_{1+1} \tag{A.1.3.2}
\]

- for linear acceleration of each link

\[
1^+ 1V_{1+1} = 1^+ 1R_{1} (1\omega_{1} \times 1P_{1+1} + 1\omega_{1} \times (1\omega_{1} \times 1P_{1+1}) + 1V_{1}) \tag{A.1.3.3}
\]

- for linear acceleration of center of mass

\[
1^+ 1V_{c1+1} = 1^+ 1\omega_{1+1} \times 1^+ 1P_{c1+1} + 1^+ 1\omega_{1+1} \times (1^+ 1\omega_{1+1} \times 1^+ 1P_{c1+1}) + 1^+ 1V_{1+1} \tag{A.1.3.4}
\]

- for effective forces

\[
1^+ 1F_{1+1} = m_{1+1} 1^+ 1V_{c1+1} \tag{A.1.3.5}
\]

- for effective couple

\[
1^+ 1N_{1+1} = C 1^+ 1I_{1+1} 1^+ 1\omega_{1+1} + 1^+ 1\omega_{1+1} \times C 1^+ 1I_{1+1} 1^+ 1\omega_{1+1} \tag{A.1.3.6}
\]
Inverse dynamic equations are given by: (from frame 1 to frame 4)
- for force exerted on link 1 by link 1-1:

\[ {^1}{f}_1 = {^1}{R} {^1}{r}_{1+1} + {^1}{F}_1 \]  \hspace{1cm} (A.1.3.7)

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

\[ {^1}{n}_1 = {^1}{N}_1 + {^1}{R} {^1}{r}_{1+1} + {^1}{Pc}_1 \times {^1}{F}_1 + {^1}{P}_1 \times {^1}{R} {^1}{r}_{1+1} \]  \hspace{1cm} (A.1.3.8)

- for torque applied by each actuator:

\[ {^1}{\tau}_1 = {^1}{n}_1^T {^1}{Z}_1 \]  \hspace{1cm} (A.1.3.9)

The following initial conditions were considered:

\[ {^1}{\omega}_1 = \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_1 \end{bmatrix}, \quad \dot{\omega}_o = 0 \]

\[ {^1}{V}_o = 0, \quad \dot{V}_o = \begin{bmatrix} g \\ 0 \\ 0 \end{bmatrix} \]

\[ f = \begin{bmatrix} w \\ 0 \\ 0 \end{bmatrix}, \quad w = \text{payload} \]
Torque equations for each actuator are:

TORQUE OF JOINT 1:

\[
\tau_1 = \dot{\theta}_1 \left\{ \frac{1}{2} I_{zz} + S_2^2 I_{xx} + C_2^2 I_{yy} + S_{23}^2 I_{xx} - C_{23}^2 I_{yy} \right\} \\
+ \frac{L_3^2}{4} C_{23} m_3 + L_2 C_2 \left( \frac{m_2}{4} + m_3 \right) + L_2 L_3 C_2 C_{23} m_3 \\
+ \ddot{\theta}_1 \dot{\theta}_2 \left\{ 2 C_2 S_2 (I_{xx} - I_{yy}) + S_{2233} (I_{xx} - I_{yy}) \right\} \\
- \frac{L_3^2}{2} S_{23} C_{23} m_3 - L_2 L_3 S_{2233} m_3 - S_2 C_2 L_2 \left( \frac{m_2}{2} + 2 m_2 \right) \\
+ \ddot{\theta}_1 \dot{\theta}_3 \left\{ L_3^2 S_{2233} (I_{xx} - I_{yy}) + \frac{L_3^2}{2} S_{23} (S_2 S_3 + C_2 C_3) - L_2 L_3 S_{23} C_2 m_3 \right\} \\
+ g \left\{ - \frac{L_3}{2} I_1 m_3 (S_2 S_3 + C_2 C_3) + L_2 S_1 C_2 \left( \frac{m_2}{2} + m_3 \right) \right\}
\]

(A.1.3.10)
TORQUE OF JOINT 2:

\[
\tau_2 = \ddot{\theta}_2 \left( lzz \right) + lzz \frac{m_3}{4} \left( \frac{L_3^2}{L_2 L_3 C_3} + L_2 \right) + \ddot{\theta}_3 \left( lzz \right) + \frac{m_3}{2} \left( \frac{L_3^2}{L_2} + \frac{L_2 L_3}{C_3} \right) + \theta_1 \left( S_2 C_2 \left( lyy - lxx \right) + S_2 C_2 \left( lyy - lxx \right) + \frac{L_2^2}{C_2 S_2 m_3} \right) + \frac{L_3}{4} \left( C_{23} S_{23} m_3^2 + \frac{L_2^2}{4} \left( S_{23}^2 m_2^2 + \frac{L_2 L_3}{2} \right) \right) + \theta_2 \left( \frac{m_2}{4} \right) + \left( \ddot{\theta}_2 - \dot{\theta}_2 \dot{\theta}_3 \right) \left( \frac{L_2 L_3}{2} \right) \frac{C_{23} S_{23} m_3}{S_2} + g \left( - \frac{L_3}{2} C_{123} m_3 - L_2 S_2 C_1 \left( \frac{m_3}{2} + \frac{m_2}{4} \right) \right) + w L_2 S_3
\]

(A.1.3.11)

TORQUE OF JOINT 3:

\[
\tau_3 = \ddot{\theta}_2 \left( lzz \right) + lzz \frac{m_3}{2} \left( \frac{L_3}{L_2 C_3} \right) + \ddot{\theta}_3 \left( lzz \right) \frac{L_3}{2} + \theta_1 \left( S_{23} C_{23} \left( lyy - lxx \right) + \frac{L_2^2}{4} \left( C_{23} S_{23} m_3^2 + \frac{L_2 L_3}{2} \right) \right) + \theta_2 \left( L_2 S_3 m_3 \right) + g \left( - C_{123} S_{23} m_3 \right)
\]

(A.1.3.12)
The general form of the torque equation is:

\[ \tau = M(\theta) \ddot{\theta} + V(\theta, \ddot{\theta}) + G(\theta) \quad (A.1.3.13) \]

where:

**Inertia Term (M(\theta)) is:**

\[
M(\theta)=
\begin{pmatrix}
IZZ_1^2S_2xx_2^2 + C_2yy_2^2 \\
\frac{L_3^2}{2} + \frac{S_3xx_3^2 - C_3y_3^2}{4} C_2 m_3 \\
+ L_2 C_2 \left( \frac{m_2}{4} + m_3 \right) + L_2 L_3 C_2 C_3 m_3 \\
0 & 0 & 0 \end{pmatrix}
\]

\[
(A.1.3.14)
\]
THE CENTRIFUGAL AND CORIOLIS TERM:

\[ V(\theta, \dot{\theta}) = \]

\[
\begin{align*}
\dot{\theta}_1 \dot{\theta}_2 \left\{ 2C_2 \frac{2C_2 S_2 (1x^2 - 1y^2) + S_{2233} (1x_3 - 1y_3)}{2} - \frac{L_3^2}{2} S_{23} C_{23} \right\} + \\
- L_2 L_3 S_{2233} m_3 - S_2 C_2 \frac{m_2}{2} + 2 m_3 \right\} + \\
\dot{\theta}_1 \dot{\theta}_3 \left\{ S_{2233} (1x^3 - 1y^3) + \frac{L_3^2}{2} S_{23} (S_2 S_3 + C_2 C_3) - L_2 L_3 S_{23} C_{23} m_3 \right\} \\
2 \left\{ \dot{\theta}_1 \left( S_2 C_2 (1y_2^2 - 1x_2) + S_{23} C_{23} (1y_3^3 - 1x_3) + \frac{L_3^2}{4} C_{23} S_{23} m_3 \right) - \frac{L_2 L_3}{4} S_2 C_2 m_2 + \frac{L_2 L_3}{2} S_{2233} m_3 + L_2 C_2 S_2 m_3 \right\} + \dot{\theta}_2 \left( \frac{L_2}{4} m_2 \right) \\
(\dot{\theta}_2 + (\dot{\theta}_2 + \dot{\theta}_3)^2) \left( \frac{L_2 L_3}{2} S_3 m_3 \right) \\
2 \left\{ \dot{\theta}_1 \left( S_{23} C_{23} (1y_3^3 - 1x_3) + \frac{L_3^2}{4} C_{23} S_{23} m_3 + \frac{L_2 L_3}{2} C_2 S_{23} m_3 \right) \right\} \\
+ \dot{\theta}_2 \left( L_2 S_3 m_3 \right)
\end{align*}
\]

(A.1.3.15)
THE GRAVITY TERM IS:

\[
G(\theta) = \left\{ \begin{array}{l}
g \left[ - \frac{L_3}{2} S_{13} m_3 (S_2 S_3 + C_2 C_3) + L_2 S_{12} \left( \frac{m_2}{2} + m_3 \right) \right] \\
g \left[ - \frac{L_3}{2} C_{12} S_{23} m_3 - \frac{L_2}{2} S_{21} m_2 - L_2 C_{23} S_{23} m_3 \right] + w L_3 S_3 \\
g \left[ - C_1 S_{23} m_3 \right]
\end{array} \right. 
\]

THE FRICTION TERM IS:

\[
F(\theta, \dot{\theta}) = V \dot{\theta} + C \text{ sgn}(\dot{\theta}) \quad (A.1.3.17)
\]

where:

\[
\begin{align*}
V &= \text{Viscous friction constant} \\
C &= \text{Coulomb friction constant}
\end{align*}
\]
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:

\[
J = \begin{bmatrix}
\delta X & \delta X & \delta X \\
\delta \theta_1 & \delta \theta_2 & \delta \theta_3 \\
\delta Y & \delta Y & \delta Y \\
\delta \theta_1 & \delta \theta_2 & \delta \theta_3 \\
\delta Z & \delta Z & \delta Z \\
\delta \theta_1 & \delta \theta_2 & \delta \theta_3
\end{bmatrix}
\] (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:

\[
J = \begin{bmatrix}
- L_3 S_1 S_2 - L_2 S_1 C_2 & - L_3 C_1 S_2 - L_2 C_1 S_2 & - L_3 C_1 S_23 \\
L_3 C_23 C_1 + L_2 C_1 C_2 & - L_3 S_1 S_23 - L_2 S_1 S_2 & - L_3 S_1 S_23 \\
0 & L_3 C_23 + L_2 C_2 & L_3 C_23 \\
\end{bmatrix}
\]  
(A.1.4.2)

where the relations between positions and velocities are:

\[
\dot{X} = \text{KIN} \{\theta\} \quad (A.1.4.3)
\]

\[
\dot{V} = J(\theta) \dot{\theta} \quad (A.1.4.4)
\]

The equation for torque and force is:

\[
\tau = J^T(\theta) f \quad (A.1.4.5)
\]
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:

\[
M_x(\theta) = J^{-T}(\theta) \cdot M(\theta) \cdot J^{-1}(\theta)
\]  \hspace{1cm} (A.1.5.1)

\[
V_x(\theta, \dot{\theta}) = J^{-T}(\theta) \left[ V(\theta, \dot{\theta}) - M(\theta) \cdot J^{-1}(\theta) \cdot J(\theta) \cdot \dot{\theta} \right]
\]  \hspace{1cm} (A.1.5.2)

\[
G_x(\theta) = J^{-T}(\theta) \cdot G(\theta)
\]  \hspace{1cm} (A.1.5.3)

\[
F_x(\theta, \dot{\theta}) = J^{-T}(\theta) \left[ F(\theta, \dot{\theta}) - M(\theta) \cdot J^{-1}(\theta) \cdot J(\theta) \cdot \dot{\theta} \right]
\]  \hspace{1cm} (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
- \( J(\theta) \) = derivative of the Jacobian
The force acting on end-effector is:

\[ f = M_x(\theta) \ddot{x} + V_x(\theta, \dot{\theta}) + G_x(\theta) + F_x(\theta, \dot{\theta}) \]  \hspace{1cm} (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:

\[ \ddot{\theta} = M^{-1}(\theta) \left[ \tau - V(\theta, \dot{\theta}) - G(\theta) - F(\theta, \dot{\theta}) \right] \]  
(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 \( \Delta t \).

The initial conditions \( t=0 \) were:

\[ \begin{align*}
\theta(0) &= \theta_0 \\
\dot{\theta}(0) &= 0 \\
\ddot{\theta}(0) &= 0 \\
\end{align*} \]

\[ \dot{\theta}(t+\Delta t) = \dot{\theta}(t) + \ddot{\theta}(t) \Delta t \]  
(A.1.6.2)

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

The selection the size of \( \Delta t \) 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:

<table>
<thead>
<tr>
<th></th>
<th>(a_{i-1})</th>
<th>(a_{i-1})</th>
<th>(d_{i-1})</th>
<th>(\theta_i)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>(\theta_1)</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>(L_1)</td>
<td>0</td>
<td>(\theta_2)</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>(L_2)</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
TRANSFORMATION MATRIX:

\[ T = \begin{pmatrix} 
  c_{12} & -s_{12} & 0 & l_1 c_1 + l_2 c_{12} \\
  s_{12} & c_{12} & 0 & l_1 s_1 + l_2 s_{12} \\
  0 & 0 & 1 & 0 \\
  0 & 0 & 0 & 1 
\end{pmatrix} \]

(A.2.1.1)

KINEMATICS EQUATIONS:

\[ x_1 = l_1 c_1 + l_2 c_{12} \]  

(A.2.1.2)

\[ y_1 = l_1 s_1 + l_2 s_{12} \]  

(A.2.1.3)

INVERSE KINEMATICS:

\[ d = x^2 + y^2 - l_1^2 - l_2^2 \]

\[ c = \sqrt{4 l_1^2 l_2^2 - d^2} \]

\[ \theta_2 = \text{ATAN2} (c,d) \]  

(A.2.1.4)

\[ \theta_1 = \text{ATAN2}(y,x) - \text{ATAN2}(a,b) \]  

(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)

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

(A.2.2.2)

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

(A.2.2.3)

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

(A.2.2.4)

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

(A.2.2.5)
A.2.3 DYNAMICS:

Initial conditions:

\[ 1\omega_1 = 0 \quad \omega_0 = 0 \quad V_0 = 0 \]

\[ V_0 = \begin{bmatrix} 0 \\ g \end{bmatrix} \quad f = \begin{bmatrix} 0 \\ \omega \end{bmatrix} \]

TORQUE EQUATIONS:

\[ \tau = M(\theta) \dot{\theta} + V(\theta, \theta) + G(\theta) \]

MASS (INERTIA) TERM:

\[ M(\theta) = \begin{bmatrix} \frac{2}{L_2^2 m_2 + 2 L_1 L_2 m_2 C_2 + L_1 (m_1 + m_2)} & \frac{2}{L_2^2 m_2 + L_1 L_2 m_2 C_2} \\ \frac{2}{L_2^2 m_2 + L_1 L_2 m_2 C_2} & \frac{2}{L_2^2 m_2} \end{bmatrix} \]

CENTRIFUGAL AND CORIOLIS TERM:

\[ V(\theta, \theta) = \begin{bmatrix} -m_2 L_1 L_2 S_2 \dot{\theta}_2^2 - 2 m_2 L_1 L_2 S_2 \theta_1 \theta_2 \\ m_2 L_1 L_2 S_2 \theta_1^2 \end{bmatrix} \]

GRAVITY TERM:

\[ G(\theta) = \begin{bmatrix} m_2 L_2 C_{12} g + (m_1 + m_2) L_1 C_1 g \\ m_2 L_2 C_{12} g \end{bmatrix} \]
Dynamic terms in Cartesian space:

**INERTIA TERM:**

\[
M_x(\theta) = \begin{bmatrix}
\frac{m_2 + m_1}{S_2^2} & \frac{C_{12} S_{12}}{S_2^2} & \frac{m_1}{S_2^2} \\
\frac{S_{12} C_{12}}{S_2^2} & \frac{C_{12} S_{12}}{S_2^2} & \frac{m_2 + m_1}{S_2^2}
\end{bmatrix}
\]  

(A.2.3.4)

**CENTRIFUGAL & CORIOLIS TERM:**

\[
V_x(\dot{\theta}_1, \dot{\theta}_2) = \frac{1}{S_2^2} \begin{bmatrix}
L_1 m_1 C_2 C_{12} \dot{\theta}_1^2 + L_2 m_1 C_{12} (\dot{\theta}_1 + \dot{\theta}_2)^2 \\
L_1 m_1 C_2 S_{12} \dot{\theta}_1^2 + L_2 m_1 S_{12} (\dot{\theta}_1 + \dot{\theta}_2)^2
\end{bmatrix}
\]  

(A.2.3.5)

**GRAVITY TERM:**

\[
G_x(\theta) = \frac{1}{S_2^2} \begin{bmatrix}
m_1 g C_1 C_{12} \\
m_1 g C_1 S_{12} + m_2 g S_2
\end{bmatrix}
\]  

(A.2.3.6)
APPENDIX (B)

Computer code: Impedance controller for robots

C*-----------------------------------------------
C  R3DIMP.FOR
C*-----------------------------------------------
C
C
C CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C IMPEDANCE CONTROL FOR 3 DOF ROBOT IN C
C 3 DIMENSIONAL SPACE C
C
C ALSO HAS: - JOINT LIMITS SCHEME C
C - DECOUPLING TERMS C
C
C CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
REAL*4 XD, YD, ZD, QD1, QD2, QD3, XM, YM, ZM, QM1, QM2, QM3, Q1(20000), E2V
REAL*4 Q2(20000), Q3(20000), DQ1(20000), DQ2(20000), DQ3(20000), K0
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, XO(10), YO(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
C
C
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
C L1=.19
L2=.3
L3=.3
M1=2.
M2=2.
M3=2.
C
C INPUT DATA
C
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
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
C READ(*,243) B
243 FORMAT(F10.3)
C WRITE(6,235)
235 FORMAT(2X,'WHAT IS RADIUS OF REPULSIVE FORCE (R)?)'
C
C READ(*,236) R
236 FORMAT(F10.3)
C
C 723 CALL INVVIN(XD,YD,ZD,QD1,QD2,QD3,ANS)
   IF (ANS .EQ. 1.0) GO TO 114
   CALL INVVIN(XM,YM,ZM,QM1,QM2,QM3,ANS)
C
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)
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(I)=XD+.04*COS(TV*TT)
YDD(I)=YD+.04*SIN(TV*TT)
ZDD(I)=ZD
C
XVDD(I)=-.04*TV*SIN(TV*TT)
YVDD(I)=.04*TV*COS(TV*TT)
ZVDD(I)=0.0
C
EX=XDD(I)-XM
EY=YDD(I)-YM
EZ=ZDD(I)-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=XMO
EYO=YM-YO
EZNEZM-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*BO
FYO=FO*EYO/RO -YVD*B
FZO=FO*EZO/RO -ZVD*BO
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
MVD=(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 MVGIX(W,TT,DQ1,DQ2,DQ3,Q1,Q2,Q3,II,UX,UX,UX)
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
MXX(1,1)=1.0
MXX(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
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
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
CALL ARMDYN(C, V, W, T1, T2, T3, Q1, Q2, Q3, DQ1, DQ2, DQ3, II, QM1, QM2, QM3, 1, DQM1, DQM2, DQM3, IT, 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. YDD(II-1)) GO TO 111
ENDIF
C
C CREATE DATA FILES FOR ALL OUTPUTS
C
111
IF (IP .EQ. 1) THEN
   WRITE(4,66) XM, YM
   66 FORMAT(2F14.5)
   WRITE(4,676) XDD(II-1), YDD(II-1)
   676 FORMAT(2F14.5)
   IF (NOBS .EQ. 1.) THEN
      WRITE(4,67) XO, YO
      67 FORMAT(2F14.5)
      ENDIF
   WRITE(3,77) XM, ZM
   77 FORMAT(2F14.5)
   IF (NOBS .EQ. 1.) THEN
      WRITE(3,75) XO, ZO
      75 FORMAT(2F14.5)
      ENDIF
   WRITE(2,91) TT, Q1(II)
   91 FORMAT(2F14.5)
   WRITE(2,396) TT, Q2(II)
   396 FORMAT(2F14.5)
   WRITE(2,397) TT, Q3(II)
   397 FORMAT(2F14.5)
   WRITE(1,88) TT, T1(II-1)
   88 FORMAT(2F14.5)
   WRITE(1,78) TT, T2(II-1)
   78 FORMAT(2F14.5)
   WRITE(1,79) TT, T3(II-1)
   79 FORMAT(2F14.5)
   WRITE(7,93) TT, XM
   93 FORMAT(2F14.5)
   WRITE(7,94) TT, YM
   94 FORMAT(2F14.5)
   WRITE(7,95) TT, ZM
   95 FORMAT(2F14.5)
   WRITE(10,811) XM, YM, ZM
   811 FORMAT(3F14.5)
   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
111 STOP
END
C
MULTIPLICATION OF TWO MATRICES    C
SUBROUTINE MULTMAT(A,B,AB)
C
REAL*4 AB(3,3),A(3,3),B(3,3)
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
FORWARD KINEMATICS CALCULATIONS    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

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))

XM=L3*C1*C23+L2*C1*C2
YM=L3*S1*C23+L2*S1*C2
ZM=L3*S23+L2*S2+L1

RETURN
END

C

C

CCC

C INVERSE KINEMATICS CALCULATIONS C

SUBROUTINE INVkin(X,Y,Z,Q11,Q22,Q33,ANS)

REAL*4 X, Y, Z, Q11, Q22, Q33, K, D, C, ANS
REAL*4 L1, L2, L3, A, B, E, S1, C3, S3, KK

L1=.19
L2=.3
L3=.3

ANS=0.0

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=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

E=AB**.5
Q22= ATAN2(E,A)

112 RETURN
END

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C JACOBIAN MATRIX CALCULATIONS  C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
SUBROUTINE JACOBIAN(Q1,Q2,Q3,II,JA)

REAL*4 Q1(20000),Q2(20000),Q3(20000),JA(3,3)
REAL*4 L1,L2,L3,S1,C1,S2,C2,C23,S23

L1=.19
L2=.3
L3=.3

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))

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

RETURN
END

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C TRANPOSE OF JACOBIAN MATRIX  C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
SUBROUTINE TRANSPJA(JA,JT)

REAL*4 JA(3,3),JT(3,3)
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
RETURN
END

C

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C CALCULATIONS OF INERTIAL, CENTRIFUGAL, CORIOLIS, GRAVITY
C AND FRICTION TERMS
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
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,IFY1
REAL*4 IXX1,IZZ2,IFY2,IXX2,IXX3,IFY3,IZZ3,L3,L2,L1
REAL*4 S1,S2,S3,S23,C1,C2,C3,C23,M1,M2,M1,W,S223,S233
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
IFY1=0.0
IXX2=0.0
IFY2=0.0
IXX3=0.0
IFY3=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))
M(1,1)=IZZ1+S2*S2*IXX2+C2*C2*YY2+S23*S23*IXX3-C23*C23*IYY3
M(1,2)=0.0
M(1,3)=0.0
M(2,1)=0.0
M(3,1)=0.0
M(3,2)=IZZ3+(L3/2.+L2*C3)*M3
M(3,3)=IZZ3+L3*M3/2.

V(1)=DQ1(II)**DQ2(II)**(2.*C2*S2*(IXX2-1*YY2)+S2333*(IXX3-1*YY3)
2  +DQ1(II)**DQ3(II)**(S2233*(IXX3-1*YY3)+L3*L3*S23*(S2*S3+C2*C3)/2.
3  -L2*L3*S23*C2*M3

V(2)=DQ1(II)**2 *(S2*C2*(IYX2-IXX2)+S23*C2*(IYX3-IXX3)+L3*L3
4  S3*M3/2.


G(1)=GG*(-L3*S1*M3*(S2*S3+C2*C3)/2. + L2*S1*C2*(M2/2. +M3))
G(3)=GG*(-C1*S23*M3)

RETURN
END

C
CCCCCC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C CALCULATIONS OF JACOBIAN DERIVATIVE MATRIX  C
CCCC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
SUBROUTINE JACOBIAND(Q1,Q2,Q3,DQ1,DQ2,DQ3,II,JD)

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

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

CENTRIFUGAL, CORIOLIS AND GRAVITY TERMS CALCULATION  C
IN CARTESIAN SPACE  C
SUBROUTINE VGX(JI, JTI, M, V, G, JD, DQ1, DQ2, DQ3, II, VX, GX)

C
REAL*4 J1(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) * JDQ(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 80 J=1,3
     VX(I)=JI(T(I,J))*VM(J)+VX(I)
    80 CONTINUE
C
     DO 100 I=1,3
     GX(I)=0.0
     DO 110 J=1,3
     GX(I)=JI(T(I,J))*G(J)+GX(I)
    110 CONTINUE
C
     RETURN
END
C
CCC
CCC INERTIAL, CENTRIFUGAL, CORIOLIS AND GRAVITY TERMS C
CCC IN CARTESIAN SPACE C
CCC 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),JI(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 TRANSPIA(JA,TT)
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)=JI(I,J)
   20 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)
CALL INVER(JTI, NP, INDEX)
CALL JACOBIAND(Q1, Q2, Q3, DQ1, DQ2, DQ3, II, JD)
CALL MULTMAT(JTI, M, JTIM)
CALL MULTMAT(JTIM, JI, MK)
CALL VGX(JI, JTI, M, V, G, JD, DQ1, DQ2, DQ3, II, VX, GX)

RETURN
END

CC
CC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CC ARMDYNAMICS CALCULATIONS C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
SUBROUTINE ARMDYN(C, VV, W, T1, T2, T3, Q1, Q2, Q3, DQ1, DQ2, DQ3, II, QM1, QM2,
1 QM3, DQM1, DQM2, DQM3, TT, F)

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), M1(3, 3), TT, M1(3, 3), QM1, QM2, QM3, DQM1, DQM2, DQM3
REAL*4 INDEX(3), F(3), C, VV

INTEGER NP

CALL MVE(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

DT=.004
IF (II .EQ. 1) TT=0.0
TT=TT+DT
NP=3
CALL INVER(M1, NP, INDEX)

T(1)=T1(II)
T(2)=T2(II)
T(3)=T3(II)

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
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,II)*TVG(I)+DDQ1(II)
      DDQ2(II)=MI(2,II)*TVG(I)+DDQ2(II)
      DDQ3(II)=MI(3,II)*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
      RETURN
      END
      
      C
      CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      C INVERSE OF MATRIX CALCULATIONS      C
      CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      SUBROUTINE INVERS(ASAT, NP, INDEX)
      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
         TEMP=ABS(ASAT(I, I))
         IF(TEMP-AMAX)115, 115, 117
      116    I=I
      117   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
      121
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
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), NIND, 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

INTEGER NB, IP

IP=1
Q1MAX=1.92
Q1MIN=-1.92
Q2MAX=1.484
Q2MIN=-.47
Q3MAX=0.0
Q3MIN=-2.13

L1=.19
L2=.3
L3=.3
M1=2.
M2=2.
M3=2.

TT=0.0
KV=4.
KP=4.
II=1
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
C       414 FORMAT(F10.3)
C
C    WRITE(6,342)
C       342 FORMAT(2X,'WHAT IS ANGLE CONSTANT (KQ)? ')
C
C    READ(*,343) KQ
C       343 FORMAT(F10.3)
C
C    WRITE(6,622)
C       622 FORMAT(2X,'IS THERE ANY OBSTACLE IN WORKVOLUME? YES(1.)')
C
C    READ(*,722) NOBS
C       722 FORMAT(F4.2)
C
C    IF (NOBS .NE. 1.) GO TO 723
C
C    WRITE(6,601)
C       601 FORMAT(2X,'HOW MANY OBSTACLES? ')
C
C    READ(*,602) NB
C       602 FORMAT(I2)
C
C    DO 222 I=1,NB
C    WRITE(6,223) I
C       223 FORMAT(2X,'XS(',I2,')'= ')
C
C    READ(*,224) XS(I)
C       224 FORMAT(F10.3)
C
C    WRITE(6,225) I
C       225 FORMAT(2X,'YS(',I2,')'=')
C
C    READ(*,226) YS(I)
C       226 FORMAT(F10.3)
C
C    WRITE(6,227) I
C       227 FORMAT(2X,'ZS(',I2,')'=')
C
C    READ(*,228) ZS(I)
C       228 FORMAT(F10.3)
C
C    WRITE(6,229) I
C       229 FORMAT(2X,'RS(',I2,')'=')
C
C    READ(*,241) RS(I)
C       241 FORMAT(F10.3)
C
C    222 CONTINUE
C
C    WRITE(6,233)
C       233 FORMAT(2X,'WHAT IS REPULSIVE FORCE CONSTANT (KO)?')
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
723 CALL INVVIN(XD,YD,ZD,QD1,QD2,QD3,ANS)
   IF (ANS .EQ. 1.0) GO TO 114
   CALL INVVIN(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
            ,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 VISCOS 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 TORQUE FOR 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
EX=XD-XM
EY=YM-YM
EZ=ZD-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)
232 CONTINUE

127
RO=ROF(I)
MIND=ROF(I)
ENDIF

232 CONTINUE

C
EXO=XM-XO
EYO=YM-YO
EZ0=ZM-ZO
C
IF (RO .GT. R) THEN
FXO=0.0
FY0=0.0
FZO=0.0
ENDIF
C
IF (RO .LT. R) THEN
FO=KO*((RO-R)**2.)
FXO=FO*EXO/RO -XVD*B
FY0=FO*EYO/RO -YVD*B
FZO=FO*EZ0/RO -ZVD*B
ENDIF
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
FX0=0.0
FY0=0.0
FZO=0.0
ENDIF
C
FXP=EX*KP-XVD*KV+FX0
FYP=FY*KP-YVD*KV+FY0
FZP=EZ*KP-ZVD*KV+FZO
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)+CX(1)
FY=FXP*MX(2,1)+FYP*MX(2,2)+FZP*MX(2,3)+VX(2)+CX(2)
FZ=FXP*MX(3,1)+FYP*MX(3,2)+FZP*MX(3,3)+VX(3)+CX(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
128
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
CHECK THE MAXIMUM TORQUES FOR EACH JOINT.

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)
IF (NOBS .EQ. 1.) THEN
WRITE(4,67) X0, Y0
67 FORMAT(2F14.5)
ENDIF

WRITE(3,77) XM, ZM
77 FORMAT(2F14.5)

IF (NOBS .EQ. 1.) THEN
WRITE(3,75) X0, Z0
75 FORMAT(2F14.5)
ENDIF

WRITE(2,91) TT, Q1(I1)
91 FORMAT(2F14.5)

WRITE(2,963) TT, Q2(I1)
963 FORMAT(2F14.5)

WRITE(2,964) TT, Q3(I1)
964 FORMAT(2F14.5)

IF (NOBS .EQ. 1.) THEN
WRITE(2,92) Z0, Y0
92 FORMAT(2F14.5)
ENDIF

WRITE(1,88) TT, T1(I1-1)
88 FORMAT(2F14.5)

WRITE(1,78) TT, T2(I1-1)
78 FORMAT(2F14.5)

WRITE(1,79) TT, T3(I1-1)
79 FORMAT(2F14.5)

WRITE(7,93) TT, XM
93 FORMAT(2F14.5)

WRITE(7,94) TT, YM
94 FORMAT(2F14.5)

WRITE(7,95) TT, ZM
95 FORMAT(2F14.5)

WRITE(10,811) XM, YM, ZM
811 FORMAT(3F14.5)

IF (NOBS .EQ. 1.) THEN
WRITE(10,812) X0, Y0, Z0
812 FORMAT(3F14.5)
ENDIF

IF (II .NE. 1) THEN
WRITE(8,875) TT,AT1(I-1)
876 FORMAT(2F14.5)
C
WRITE(8,875) TT,AT2(I-1)
875 FORMAT(2F14.5)
C
WRITE(8,874) TT,AT3(I-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
111 STOP
END
C

ALL SUBROUTINES ARE SAME AS R3DIMP. FOR PROGRAM
C
THEREFOR, THERE IS NO NEED TO REPEAT THEM. ONLY TWO
C
NEW SUBROUTINES ARE ADDED
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
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

131
TT3=FX*JA(1,3)+FY*JA(2,3)+FZ*JA(3,3)-DQ3(I1)*KQ3
ENDIF

C
88 IF (ABS(TT2).GT. T2MAX) THEN
A3=JA(1,2)
A4=JA(2,2)
A5=JA(3,2)
A6=DQ2(I1)*KQ2
TT2=T2MAX * TT2/ABS(TT2)
CALL TORFIX(TT2, A1, A2, A3, A4, A5, A6, MX, VX, GX, I1, FX, FY, FZ)
TT1=FX*JA(1,1)+FY*JA(2,1)+FZ*JA(3,1)-DQ1(I1)*KQ1
TT3=FX*JA(1,3)+FY*JA(2,3)+FZ*JA(3,3)-DQ3(I1)*KQ3
ENDIF

C
99 IF (ABS(TT3).GT. T3MAX) THEN
A3=JA(1,3)
A4=JA(2,3)
A5=JA(3,3)
A6=DQ3(I1)*KQ3
TT3=T3MAX * TT3/ABS(TT3)
CALL TORFIX(TT3, A1, A2, A3, A4, A5, A6, MX, VX, GX, I1, FX, FY, FZ)
TT1=FX*JA(1,1)+FY*JA(2,1)+FZ*JA(3,1)-DQ1(I1)*KQ1
TT2=FX*JA(1,2)+FY*JA(2,2)+FZ*JA(3,2)-DQ2(I1)*KQ2
ENDIF

C
IF (ABS(TT1).GT. T1MAX .AND. ABS(TT1).GT. T2MAX) THEN
IF (ABS(TT3).GT. T3MAX) THEN
TT1=T1MA* 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
CCC CCC CCC CCC CCC CCC CCC CCC CCC CCC CCC CCC CCC CCC CCC CCC CCC CCC CCC
C
SUBROUTINE TORFIX(T, A1, A2, A3, A4, A5, A6, MX, VX, GX, I1, 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

FYP = B/C
FXP = A1 * FYP
FZP = A2 * FYP

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)

RETURN
END
C******************************************************************************
C* R3DC3.FOR
C******************************************************************************
C
C
CCC

C IMPEDEANCE CONTROL FOR 3 DOF ROBOT IN C
C 3 DIMENSIONAL SPACE
C
C ALSO INCLUDES: - SURFACE NAVIGATION C
C SCHEME
C
CCC

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), XV, YV, ZV, FX, FY, FZ, PX, PY, F1(20000)
REAL*4 VX(3), GX(3), FX, FY, FZ, T1(20000), T2(20000), T3(20000)
REAL*4 TT, T, TMAX, TQ1, TQ2, TQ3, 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
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=')
134
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(6,236)
  236 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 INVVIN(XD,YD,ZD,QD1,QD2,QD3,ANS)
IF (ANS .EQ. 1.0) GO TO 114
CALL INVVIN(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
   ,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.
   ,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
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)
231
ROF(I) = ((XM-XOF(I))**2 + (YM-YOF(I))**2 + (ZM-ZOF(I))**2)**0.5

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

EXO=XM-XO
EYO=YM-YO
EZ0=ZM-ZO
C

IF (RO .GT. R) THEN
   FXO=0.0
   FYO=0.0
   FZD=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
   FZ0=FO*EZ0/RO -ZVD*B
C

IF (ANS2 .EQ. 1.) THEN

XDM=XD-XM
YDM=YM-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

XCM=YOM*N3-N2*ZOM
YCM=N1*ZOM-XOM*N3
ZCM=N2*XOM-N1*YOM
C

R1=((XCM)**2 + (YCM)**2 + (ZCM)**2)**0.5

138
FXO=(XCM)*KC/R1
FYO=(YCM)*KC/R1
FZO=(ZCM)*KC/R1
WRITE(6,956)
956
FORMAT(’,SURFACE NAVIGATION’)

ENDIF
ENDIF

CALL JACOBIAN(Q1,Q2,Q3,II,JA)

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)

FXP=EX*KP-XVD*KV+FXO
FYP=EY*KP-YVD*KV+FYO
FZP=EZ*KP-ZVD*KV+FZO

IF (ANS2 .EQ. 1.) THEN
  FXP=FXO-XVD*KV
  FYP=FYO-YVD*KV
  FZP=FZO-ZVD*KV
ENDIF

CALL MVGX(W,TT,DQ1,DQ2,DQ3,Q1,Q2,Q3,II,MX,VX,GX)

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)

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

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

139
CALL ARMDYN(W, T1, T2, T3, Q1, Q2, Q3, DQ1, DQ2, DQ3, II, QM1, QM2, QM3,
  DQM1, DQM2, DQM3, TT)

II = II + 1
Q1(II) = QM1
Q2(II) = QM2
Q3(II) = QM3
DQ1(II) = DQM1
DQ2(II) = DQM2
DQ3(II) = DQM3
XM1 = XM
YM1 = YM
ZM1 = ZM

CALL KIN(Q1, Q2, Q3, II, XM, YM, ZM)

EXX = ABS(XM1 - XM)
EYY = ABS(YM1 - YM)
EZZ = ABS(ZM1 - ZM)

IF (EXX .GE. .0001 .OR. EYY .GE. .0001) TC = 0.0
  IF (EZZ .GE. .0001) TC = 0.0
  IF (EXX .LT. .0001 .AND. EYY .LT. .0001) THEN
    IF (EZZ .LT. .0001) THEN
      TC = TC + .005
    ENDIF
  ENDF
ENDIF
IF (TC .GE. 2.) ANS2 = 1.

IF (IP .EQ. 1) THEN
  WRITE(4, 66) XM, YM
  S6 FORMAT(2F14.5)
ENDIF
WRITE(4, 67) XO, YO
67 FORMAT(2F14.5)
WRITE(3, 77) XM, ZM
77 FORMAT(2F14.5)
WRITE(2, 75) TT, T1(II-1)
75 FORMAT(2F14.5)
WRITE(2, 91) TT, T2(II-2)
91 FORMAT(2F14.5)
WRITE(2, 92) TT, T3(II-3)
92 FORMAT(2F14.5)
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 All subroutines are same as R3DIMP. For program C

C