

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/257941469>

# Handling Interaction Forces Between an Object and Cartesian Robot Arm in Planar Motion

Thesis · August 2002

---

CITATIONS

5

READS

5,180

1 author:



A. Emre Cetin

Olgun Celik A.S.

41 PUBLICATIONS 172 CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:



Permutations of idempotent maps and their applications [View project](#)



# CONTENTS

<b>ACKNOWLEDGEMENTS.....</b>	<b>I</b>
<b>CONTENTS.....</b>	<b>II</b>
<b>ÖZET.....</b>	<b>V</b>
<b>ABSTRACT.....</b>	<b>VII</b>
<b>LIST OF SYMBOL.....</b>	<b>IX</b>
<b>LIST OF FIGURES.....</b>	<b>XII</b>
<b>LIST OF TABLES.....</b>	<b>XVI</b>
<b>PART I. INTRODUCTION AND OBJECTIVES .....</b>	<b>1</b>
<b>I.1 AUTOMATION AND MANIPULATORS.....</b>	<b>1</b>
<b>I.2 MANIPULATOR AND CONTROLLER.....</b>	<b>4</b>
<b>I.2.1 Mechanisms of Manipulators.....</b>	<b>4</b>
<b>I.2.2 Controllers.....</b>	<b>6</b>
<b>I.3 SYSTEM IDENTIFICATION.....</b>	<b>8</b>
<b>PART II. GENERAL BACKGROUND .....</b>	<b>10</b>
<b>II.1 MECHANICS AND CONTROL OF MANIPULATORS.....</b>	<b>10</b>
<b>II.1.1 Description of Position and Orientation.....</b>	<b>10</b>
<b>II.1.2 Forward Kinematics of Manipulators.....</b>	<b>11</b>
<b>II.1.2.1 Denavit-Hartenberg Convention .....</b>	<b>14</b>
<b>II.1.3 Inverse Kinematics of Manipulators .....</b>	<b>14</b>
<b>II.1.4 Velocities, Static Forces, Singularities.....</b>	<b>15</b>
<b>II.1.5 Dynamics.....</b>	<b>16</b>
<b>II.1.6 Trajectory Generation .....</b>	<b>19</b>
<b>II.1.7 Position Control.....</b>	<b>20</b>

II.1.7.1	PID Control.....	21
II.1.7.2	Feedforward and Computed Torque Control.....	22
II.1.7.3	Cartesian-Based Position Control .....	24
<b>II.1.8</b>	<b>Nonlinear Position Control.....</b>	<b>25</b>
<b>II.1.9</b>	<b>Interaction Control .....</b>	<b>25</b>
II.1.9.1	Indirect vs. Direct Control .....	28
II.1.9.2	Force control .....	29
II.1.9.3	Hybrid Position/Force Control.....	32
II.1.9.4	Stiffness Control .....	34
II.1.9.5	Impedance Method .....	35
<b>II.1.10</b>	<b>IDENTIFICATION METHODS .....</b>	<b>40</b>
<b>PART III.</b>	<b>THESIS STUDIES.....</b>	<b>44</b>
<b>III.1</b>	<b>THREE AXIS CARTESIAN ROBOT ARM.....</b>	<b>44</b>
<b>III.2</b>	<b>EXPERIMENTAL SETUP.....</b>	<b>46</b>
<b>III.2.1</b>	<b>Baldor BSM 63A-275 Three Phase Brushless Servo Motor .....</b>	<b>46</b>
<b>III.2.2</b>	<b>Baldor BSC 1100 AC Three Phase Brushless Servo Motor Controller</b>	<b>46</b>
<b>III.2.3</b>	<b>PLS 70 Servo-Gear-Head .....</b>	<b>46</b>
<b>III.2.4</b>	<b>Axes.....</b>	<b>47</b>
III.2.4.1	FESTO DGE 25-ZR Toothed (Timing) Belt Drive .....	47
III.2.4.2	FESTO DGE 40-SR Spindle Drive .....	47
<b>III.2.5</b>	<b>Force Sensors .....</b>	<b>47</b>
<b>III.2.6</b>	<b>Motion sensor .....</b>	<b>47</b>
<b>III.2.7</b>	<b>Data Acquisition Cards.....</b>	<b>48</b>
III.2.7.1	Advantech PCL 812PG MultiLab Analog and Digital I/O Card... 48	
III.2.7.2	Advantech PCI-1720 4-channel Isolated D/A Output Card ....., 48	
III.2.7.3	Advantech PCL-833 3-axis Quadrature Encoder and Counter Card ....., 49	

<b>III.2.8 Client/Host PC's .....</b>	<b>49</b>
<b>III.2.9 Software .....</b>	<b>50</b>
III.2.9.1 MATLAB .....	50
III.2.9.2 xPC Target.....	50
<b>III.3 KINEMATICS OF THE THREE-AXIS CARTESIAN ROBOT ARM..</b>	<b>52</b>
<b>III.4 DYNAMICS OF THE THREE-AXIS CARTESIAN ROBOT ARM..</b>	<b>56</b>
<b>III.5 TRAJECTORY GENERATION .....</b>	<b>58</b>
<b>III.6 IDENTIFICATION OF THE THREE-AXIS CARTESIAN ROBOT ARM .....</b>	<b>59</b>
III.6.1 Step-By-Step Identification Procedure.....	60
III.6.2 Identification in X-Axis.....	61
III.6.3 Identification in Y-Axis.....	67
<b>III.7 IDENTIFICATION OF THE AMPLIFIER CONSTANTS .....</b>	<b>74</b>
<b>III.8 IMPEDANCE CONTROL OF A MANIPULATOR COOPERATING WITH A HUMAN .....</b>	<b>75</b>
<b>III.9 IMPEDANCE CONTROL OF A MANIPULATOR PUSHING AN OBJECT TO THE DESIRED POSITION.....</b>	<b>84</b>
III.9.1 Pushing with Position Controller.....	86
III.9.2 Pushing with Impedance Controller.....	90
<b>PART IV. CONCLUSION .....</b>	<b>103</b>
<b>REFERENCES.....</b>	<b>107</b>
<b>APPENDIX A .....</b>	<b>112</b>
<b>APPENDIX B .....</b>	<b>122</b>

## ÖZET

Modelleme ve tanımlama mühendislere bir sistem tasarlarken gerekli olur. Modelleme ve tanımlama ile bir sistemin ve kontrolörün performansı test edilebilir. Model, bir sistemin matematiksel bir şekilde izah edilmesidir. Modeller, sınırlamaların ve sistemi oluşturan alt sistemler ve parçaların doğaları ile oluşturulur. Modeller bu sınırlamaların ve doğanın parametrelerini içerir. Bu parametreler, tanımlanana dek bilinmemektedir. Modelden yola çıkarak gerçek sistemin davranışları görmek istenirse, bu parametreler tanımlanmak zorundadır. Bundan dolayı tanımlama, gerçek sistemi oluşturmadan, model ile çeşitli analizlerin yapılması gerekliliğinde önem kazanır. Bunun yanında çevresi ile teması bulunması gereken robotların kontrol yöntemleri robotun dinamik parametreleri üzerine kurulmuştur, dolayısı ile sistemin tanımlanması kontrolün yeterli hassasiyetle gerçekleştirilebilmesi için hayatı öneme sahiptir.

Makinalar ve robotlar iş ve operasyon yapan sistemlerdir. Makinalara ve robotlara bu iş ve operasyonları yaptıran unsur Kontrol Mekanizmasıdır. Kontrol edilmek istenen parametrelerin arzu edilen değerleri kontrol mekanizmasına girilir ve çıktı da makina veya robotun yapmış olduğu iştir. PID Kontrol, İleribesleme Kontrol, Hesaplanmış Tork Kontrol, Çözünmüş İvme Kontrol, Kuvvet Kontrol, Hibrid Pozisyon/Kuvvet Kontrol ve İmpedans Kontrol, hata algoritmasına dayalı bu kontrol mekanizmalarına bazı örneklerdir. Bu yöntemlerden son üçü, istenen görevin yerine getirilebilmesi için robotun çevresi ile teması geçmesinin kaçınılmaz olduğu durumlarda kullanılmak üzere tasarlanmıştır. Buna örnek olarak parlatma, taşlama, işleme, montaj ve insan ile birlikte çalışma verilebilir. Örneklerin çokluğuna bakıldığından bu tür bir robot görevi sınıflandırmasına gitmek, ve bu sınıflandırmanın çevresi ile teması geçen robotlar için genel bir strateji geliştirilmesine yardımcı olacağını düşünmek çokta mantıklı değildir.

Proje aşamasında, ilk olarak "otomasyon ve robotlar", "robot mekanizmaları ve kontrolörler", ve "sistem tanımlama" konularında araştırma ve analizler yapıldı.

Araştırma ve analizlerin sonunda, elde edilen bilgilerin gerçekte de uygulanabilirliğini görmek üzere deneylere başlandı. Rapor üç kısımda incelenebilir. İlk bölüm, otomasyon ve robotlar, robot mekanizmaları ve kontrolörler, ve sistem

tanımlama konularında genel bilgiler içermektedir. İkinci kısımda bu bilgiler detaylandırılmış ve temel kurallar ve prensipler incelenmiştir. Son kısımda ise bilgiler, labaratuarda kurulmuş olan üç eksenli kartezyen robot üzerinde pratiğe dönüştürülmüştür.

## **ABSTRACT**

Modelling and Identification is essential when engineers design a system. By modelling and identification, we can test the performance of the system and controller. A model is the definition of the system in a mathematical form. They are builded by examining the constraints and the nature of the components and subsystems. The model of a system consists the parameters of these constraints and nature of the components and subsystems. These parameters are unknown until identified. If it is desired to execute the model to see the behaviour of the actual system, identification of the unknown parameters must be caried. So, identification is essential to simulate the model and observe the behaviour of the system. Furthermore, interaction control strategies in robotics are based on the model of the manipulator, where the identification process has crucial importance.

Machines and robots are systems which do tasks and operations. A control mechanism enables machines and robots do these tasks. Control authority is exerted to ensure that some variable of interest is kept close (in some appropriately defined sense) to a desired value despite uncertainties and hardware limitations. The input for the desired value is given to the control mechanism and the output is the work done by the machine or robot for the desired task. PID Control, Feedforward Control, Computed Torque Control, Resolved Acceleration Control, Force Control, Hybrid Force/Position Control and Impedance Control are some examples to the control strategies used today which take a parameter needed to control and control it by an error algorithm. The last three of these strategies are intended to be used for the manipulators where interaction with the environment is crucial for successful execution of the task. Typical examples include polishing, deburring, machining, assembly or cooperation with a human. A complete classification of possible robot tasks is practically infeasible in view of the large variety of cases that may occur, nor would such a classification be really useful to find a general strategy to control interaction with environment.

In this study, initially a literature survey on "automation and manipulators", "manipulator mechanisms and controllers", and "system identification" is made. The

theory of these topics is explored and the studies are summarized.

After the completion of the analysis and the experimental setup, the experiments are started to see the theory in real life. This report can be evaluated in three main parts. Firstly, general information about the Automation and Manipulators, Manipulator Mechanisms and Controller and System Identification are given. Secondly, the principles and laws of these topics are covered. The last part is the implementation of the topics on a 3-Axis Cartesian Robot Arm.

## LIST OF SYMBOLS

<b>q</b>	The displacement vector of joint $i$
<b>r</b>	The position vector of the end-effector
$f_r$	The relation between $r$ and $q$
<b>L</b>	The Lagrangian
<b>K</b>	Kinetic energy
<b>U</b>	Potential energy
$\xi_i$	Generalized force at joint
$\tau_{ci}$	Driving torque at the joint
$\tau_{fi}$	Torque due to joint friction
$\tau_{ei}$	Torque caused by the external force and moment exerted by the end-effector
<b>f<sub>e</sub></b>	Vector of external end-effector force
<b><math>\mu_e</math></b>	Vector of external end-effector moment
<b>Y</b>	The matrix called the Regressor of the dynamic model
<b>I</b>	Moment of inertia of a rigid body
<b>K<sub>p</sub></b>	Proportional gain
<b>K<sub>i</sub></b>	Integral gain
<b>K<sub>d</sub></b>	Derivative gain
<b>e</b>	The tracking error
<b>u</b>	Feedback Signal
<b>B(q)</b>	Inertia matrix of the robot manipulator with respect to joint space

$\mathbf{C}(q, \dot{q})$	Vector of Coriolis and centrifugal torques with respect to joint space
$\mathbf{F}_d$	Matrix of damping effects with respect to joint space
$\mathbf{g}(q)$	Vector of gravity torques with respect to joint space
$\mathbf{F}_s$	Matrix of static friction with respect to joint space
$\boldsymbol{\tau}_c$	Vector of joints torques
$\mathbf{h}$	Vector of end-effector force and moments
$\mathbf{J}$	Jacobian matrix which transforms end-effector force and moments to joint space
$\mathbf{q}_i$	Vector of joint variables
$\mathbf{K}_f$	Feed constant matrix of the axes
$\mathbf{K}_r$	Reduction ratio matrix of the servo-gear-heads
$\mathbf{K}_t$	Matrix of torque constants
$\mathbf{K}_a$	Matrix of amplifier constants
$\mathbf{V}$	Vector of servomotors read voltage
$\mathbf{B}$	Mass matrix with respect to task space
$\mathbf{D}$	Damping coefficient with respect to task space
$m_i$	Mass of axis $i$
$d_i$	Damping coefficient of axis $i$
$\mathbf{f}_c$	Vector of joint forces
$\mathbf{f}$	Vector of joint forces with respect to the task space
$\mathbf{f}_y$	Vector of joint forces with respect to the task space for y-axis
$\mathbf{f}_e$	Vector of external end-effector force
$\mathbf{f}_{ey}$	Vector of y-axis external end-effector force
$\mathbf{F}_{sr}$	Matrix of static friction with respect to task space

$\mathbf{F}_{sy}$	Static friction with respect to task space for y-axis
$x, y, z$	Displacement of x, y, z axis respectively
$\dot{x}, \dot{y}, \dot{z}$	Velocity of x, y, z axis respectively
$\ddot{x}, \ddot{y}, \ddot{z}$	Acceleration of x, y, z axis respectively
$m_o$	Mass of the estimated slave characteristics
$b_o$	Mass of the estimated slave characteristics
$\mathbf{M}_d$	Desired mass matrix of the impedance
$\mathbf{D}_d$	Desired damping coefficient matrix of the impedance
$\mathbf{K}_d$	Desired spring constant matrix of the impedance
$m_d$	Desired mass of the impedance
$d_d$	Desired damping coefficient of the impedance
$k_d$	Desired spring constant of the impedance
$m_{dy}$	Desired mass of the impedance for y-axis
$d_{dy}$	Desired damping coefficient of the impedance for y-axis
$k_{dy}$	Desired spring constant of the impedance for y-axis
$\mathbf{r}_e$	Difference between the position vector $\mathbf{r}$ and its desired value $\mathbf{r}_d$
$m_a$	Actual mass of the body
$d_a$	Desired damping coefficient of the body
$k_a$	Desired spring constant of the body

## LIST OF FIGURES

<b>Figure I.1</b>	Robot system [2].....	4
<b>Figure I.2</b>	Symbols of joints (arrows show direction of motion). (a) Prismatic joint (b) Revolute Joint. (c) Revolute Joint (c1) Up-and-down rotation (c2) Back-and-forth rotation [2].....	5
<b>Figure I.3</b>	Arm mechanisms. (a) Orthogonal-coordinate type. (b) Cylindrical-coordinate type. (c) polar coordinate type (d) Vertical multi joint type (e) Horizontal multi joint type [2] .....	6
<b>Figure I.4</b>	Wrist mechanisms [2] .....	6
<b>Figure I.5</b>	Rough sketch of position-control system [2].....	7
<b>Figure I.6</b>	Position-velocity feedback servosystem. [2].....	8
<b>Figure II.1</b>	Coordinate systems or "frames" are attached to the manipulator and objects in the environment. [3].....	11
<b>Figure II.2</b>	Kinematic equation describes the tool frame relative to the base frame as a function of the joint variables. [3].....	12
<b>Figure II.3</b>	Description of the position and orientation of the end-effector frame [1].....	13
<b>Figure II.4</b>	The geometrical relationship between joint rates and velocity of the end-effector can be described in a matrix called the Jacobian [3].....	16
<b>Figure II.5</b>	In order to move the end-effector through space from point A to point B, we must compute a trajectory for each joint to follow. [3] .....	20
<b>Figure II.6</b>	Feedback control system .....	21
<b>Figure II.7</b>	In order for a manipulator to slide across a surface while applying a constant force, a hybrid position-force control system must be used. [3].....	27
<b>Figure II.8</b>	A simple model of the robot in contact with the environment .....	29
<b>Figure II.9</b>	Hybrid position/force control.....	32
<b>Figure II.10</b>	Resolved acceleration force control.....	33
<b>Figure II.11</b>	Impedance control of one degree of freedom .....	35
<b>Figure II.12</b>	Impedance method (a) Feedback control system. (b) Mechanical impedance realized by (a).....	36
<b>Figure II.13</b>	Block diagram of impedance control .....	38
<b>Figure II.14</b>	Block Diagram of an unknown system.....	40
<b>Figure III.1</b>	Three-Axis Cartesian Robot Arm .....	44
<b>Figure III.2</b>	Data Flow of a Three-Axis Cartesian Robot Arm.....	45
<b>Figure III.3</b>	Kinematic Model of the Three Axis Cartesian Robot Arm.....	52
<b>Figure III.4</b>	Displacement, Velocity and Acceleration vs Time Graphs of X-Axis.....	58
<b>Figure III.5</b>	Displacement, Velocity and Acceleration vs Time Graphs of Y-Axis.....	58
<b>Figure III.6</b>	Trajectory Generation Result, X-Axis vs Y-Axis plot .....	59
<b>Figure III.7</b>	X-axis, Displacement versus time graph with ramp input for torque .....	61

<b>Figure III.8</b>	X-axis, Force versus time graph with ramp input for torque .....	62
<b>Figure III.9</b>	X-axis, Displacement versus time graph with constant torque .....	62
<b>Figure III.10</b>	X-axis, velocity versus time graph with constant torque.....	63
<b>Figure III.11</b>	X-axis, acceleration versus time graph with constant torque .....	63
<b>Figure III.12</b>	X-axis, force versus time graph with constant torque .....	64
<b>Figure III.13</b>	X-axis, Velocity versus time graph .....	64
<b>Figure III.14</b>	X-axis, Acceleration versus time graph.....	65
<b>Figure III.15</b>	X-axis, Force versus time graph .....	66
<b>Figure III.16</b>	X-axis, Response of the system versus Model response .....	66
<b>Figure III.17</b>	Y-axis, Displacement versus time graph with ramp input for torque .....	67
<b>Figure III.18</b>	Y-axis, Force versus time graph with ramp input for torque .....	68
<b>Figure III.19</b>	Y-axis, Displacement versus time graph with constant torque .....	68
<b>Figure III.20</b>	Y-axis, velocity versus time graph with constant torque.....	69
<b>Figure III.21</b>	Y-axis, acceleration versus time graph with constant torque .....	69
<b>Figure III.22</b>	Y-axis, force versus time graph with constant torque .....	70
<b>Figure III.23</b>	Y-axis, Velocity versus time graph .....	70
<b>Figure III.24</b>	Y-axis, Acceleration versus time graph.....	71
<b>Figure III.25</b>	Y-axis, Force versus time graph .....	72
<b>Figure III.26</b>	Y-axis, Response of the system versus Model response .....	72
<b>Figure III.27</b>	X-Axis amplifier constant error.....	74
<b>Figure III.28</b>	Y-Axis amplifier constant error.....	75
<b>Figure III.29</b>	Z-Axis amplifier constant error.....	75
<b>Figure III.30</b>	A human pushing a cart on a frictionless plane in one direction.....	76
<b>Figure III.31</b>	Two human positioning a cart on a frictionless plane in one direction.....	76
<b>Figure III.32</b>	Model of the slave .....	76
<b>Figure III.33</b>	Experimental setup for estimating the slave's model .....	77
<b>Figure III.34</b>	First data set from experiments.....	77
<b>Figure III.35</b>	Second data set from experiments.....	78
<b>Figure III.36</b>	Experimental and simulated trajectories for first and second data set.....	79
<b>Figure III.37</b>	Impedance control scheme of the manipulator behaving as the slave.....	80
<b>Figure III.38</b>	Trajectory and Force Data using first and second slave model with Feedforward PD Control parameters $K_p=1000$ , $Kv=200$ .....	81
<b>Figure III.39</b>	Trajectory and Force Data using first and second slave model with Feedforward PD Control parameters $K_p =10000$ , $Kv =300$ .....	82
<b>Figure III.40</b>	Trajectory and Force data using second model ( $m_o=2.23$ , $b_o=19.52$ ) with Feedforward PD Control parameters $K_p =30000$ , $Kv =2000$ .....	82
<b>Figure III.41</b>	Trajectory and Force Data using second model ( $m_o=2.23$ , $b_o=19.52$ , $k_o=0.3$ ) with Feedforward PD Control parameters $K_p =30000$ , $Kv =2000$ .....	83
<b>Figure III.42</b>	Trajectory and Force Data using $m_o=4$ and $b_o=32$ with Feedforward PD Control	

	parameters $K_p = 30000$ , $K_v = 2000$ .....	84
<b>Figure III.43</b>	Pushing an object on a frictional plane .....	84
<b>Figure III.44</b>	Position and External Force Data using Feedforward PID Control parameters; $K_p = 50000$ , $K_v = 8000$ , $K_i = 10000$ . Approach Distance=0.01 m.....	87
<b>Figure III.45</b>	Position and External Force Data using Feedforward PID Control parameters; $K_p = 50000$ , $K_v = 8000$ , $K_i = 10000$ . Approach Distance=0.02 m.....	87
<b>Figure III.46</b>	Position and External Force Data using Feedforward PID Control parameters; $K_p = 50000$ , $K_v = 8000$ , $K_i = 10000$ . Approach Distance=0.03 m.....	87
<b>Figure III.47</b>	Position and External Force Data using Feedforward PID Control parameters; $K_p = 50000$ , $K_v = 8000$ , $K_i = 10000$ . Approach Distance=0.04 m.....	88
<b>Figure III.48</b>	Position and External Force Data using Feedforward PID Control parameters; $K_p = 50000$ , $K_v = 8000$ , $K_i = 10000$ . Approach Distance=0.05 m.....	88
<b>Figure III.49</b>	Pushing an object on a frictional plane with obstruction .....	89
<b>Figure III.50</b>	Trajectory and External Force Data using Feedforward PID Control parameters $K_p = 50000$ , $K_v = 8000$ , $K_i = 10000$ . Approach Distance=0.05 m. Obstruction in opposite direction exists.....	89
<b>Figure III.51</b>	Commanded force to the manipulator with and without obstruction using Feedforward PID Control parameters; $K_p = 50000$ , $K_v = 8000$ , $K_i = 10000$ . Approach Distance=0.05 m .....	90
<b>Figure III.52</b>	Position and External Force Data using Impedance Control parameters; $m_d = 3$ , $d_d = 16$ , $k_d = 150$ . Approach Distance=0.01 m .....	92
<b>Figure III.53</b>	Position and External Force Data using Impedance Control parameters; $m_d = 3$ , $d_d = 50$ , $k_d = 300$ . Approach Distance=0.01 m .....	93
<b>Figure III.54</b>	Position and External Force Data using Impedance Control parameters; $m_d = 5$ , $d_d = 100$ , $k_d = 1000$ . Approach Distance=0.01 m .....	93
<b>Figure III.55</b>	Position and External Force Data using Impedance Control parameters; $m_d = 0.9$ , $d_d = 30$ , $k_d = 100$ . Approach Distance=0.01 m .....	94
<b>Figure III.56</b>	Position and External Force Data using Impedance Control parameters; $m_d = 0.8$ , $d_d = 16$ , $k_d = 250$ . Approach Distance=0.01 m .....	94
<b>Figure III.57</b>	Position and External Force Data using Impedance Control parameters; $m_d = 1.2$ , $d_d = 100$ , $k_d = 1000$ . Approach Distance=0.01 m .....	95
<b>Figure III.58</b>	Position and External Force Data using Impedance Control parameters; $m_d = 15$ , $d_d = 50$ , $k_d = 500$ . Approach Distance=0.01 m .....	95
<b>Figure III.59</b>	Position and External Force Data using Impedance Control parameters; $m_d = 10$ , $d_d = 50$ , $k_d = 1000$ . Approach Distance=0.01 m .....	96
<b>Figure III.60</b>	Position and External Force Data using Impedance Control parameters; $m_d = 48$ , $d_d = 30$ , $k_d = 800$ . Approach Distance=0.02 m .....	96
<b>Figure III.61</b>	Position and External Force Data using Impedance Control parameters; $m_d = 48$ , $d_d = 30$ , $k_d = 800$ . Approach Distance=0.04 m .....	97

<b>Figure III.62</b>	Position and External Force Data using Impedance Control parameters; $m_d = 100$ , $d_d = 30$ , $k_d = 800$ . Approach Distance=0.05 m .....	97
<b>Figure III.63</b>	Position and External Force Data using Impedance Control parameters; $m_d = 20$ , $d_d = 30$ , $k_d = 1500$ . Approach Distance=0.05 m .....	97
<b>Figure III.64</b>	Position and External Force Data using Impedance Control parameters; $m_d = 5$ , $d_d = 4$ , $k_d = 600$ . Approach Distance=0.01 m. Step Input used instead of desired Trajectory.....	98
<b>Figure III.65</b>	Position Data using Impedance Control parameters; $m_d = 5$ , $d_d = 4$ , $k_d = 600$ . Approach Distance=0.01 m. Step Input used instead of desired Trajectory. Obstruction and support exists.....	99
<b>Figure III.66</b>	External Force Data using Impedance Control parameters; $m_d = 5$ , $d_d = 4$ , $k_d = 600$ . Approach Distance=0.01 m. Step Input used instead of desired trajectory. Obstruction and support exists.....	99
<b>Figure III.67</b>	Position Data using Impedance Control parameters; $m_d = 1$ , $d_d = 4$ , $k_d = 300$ . Approach Distance=0.03 m. Obstruction and support exists.....	100
<b>Figure III.68</b>	External Force Data using Impedance Control parameters; $m_d = 1$ , $d_d = 4$ , $k_d = 300$ . Approach Distance=0.03 m. Obstruction and support exists.....	100
<b>Figure III.69</b>	Position Data using Impedance Control parameters; $m_d = 48$ , $d_d = 30$ , $k_d = 800$ . Approach Distance=0.04 m. Obstruction exists.....	101
<b>Figure III.70</b>	External Force Data using Impedance Control parameters; $m_d = 48$ , $d_d = 30$ , $k_d = 800$ . Approach Distance=0.04 m. Obstruction exists.....	101

## **LIST OF TABLES**

<b>Table III.1</b>	<b>Denavit-Hartenberg notation parameters .....</b>	<b>52</b>
<b>Table III.2</b>	<b>Identification results .....</b>	<b>73</b>
<b>Table III.3</b>	<b>Estimated results of the slave models (First) .....</b>	<b>79</b>
<b>Table III.4</b>	<b>Estimated results of the slave models (Used).....</b>	<b>79</b>

# PART I

## I. INTRODUCTION AND OBJECTIVES

This project involves the studies about the identification and interaction control of a 3-Axis Cartesian Robot Arm. The two topics "*identification*" and "*interaction control*" are examined separately. First chapter covers general view that includes the definitions of *Automation and Manipulators*, *Mechanisms of Manipulators and Manipulator Controllers* and for the last topic *Identification*. In the second part of the report two sections exist; *Mechanics and Control of Manipulators* that includes, kinematics, dynamics, control types of the manipulators, and *Identification Methods* that gives a general view to the system identification. At the last chapter thesis studies are mentioned, which includes *Experimental Setup*, *Kinematic and Dynamic Modeling*, *Identification of Three-Axis Cartesian Robot Arm* and *Interaction Control of the Three-Axis Cartesian Robot Arm*.

### I.1 AUTOMATION AND MANIPULATORS

By its usual meaning, the term *automation* denotes a technology aimed at replacing human beings with machines in a manufacturing process, as regards not only the execution of physical operations, but also the intelligent processing of information on the status of the process. Automation is then the synthesis of industrial technologies typical of the manufacturing process and computer technology allowing information management.

The *industrial robot* is a machine with significant characteristics of versatility and flexibility. According to the widely accepted definition of the Robot Institute of America, a *robot* is a *reprogrammable multifunctional manipulator designed to move materials, parts, tools or specialized devices through variable programmed motions for the performance of a variety of tasks*. Such a definition, dating to 1980, reflects the status of robotics technology.

An industrial robot is constituted by:

- A mechanical structure or *manipulator* that consists of a sequence of rigid bodies (*links*) connected by means of articulations (*joints*). A manipulator is characterized by an *arm* that ensures mobility, a *wrist* that confers dexterity, and an *end-effector* that performs the task required for the robot.
- *Actuators* that set the manipulator in motion through actuation of the joints; the motors employed are typically electric and hydraulic, and occasionally pneumatic
- *Sensors* that measure the status of the manipulator (proprioceptive sensors) and, if necessary, the status of the environment (heteroceptive sensors).
- A *control system* (computer) that enables control and supervision of manipulator motion.

Industrial robots present three fundamental capacities that make them useful for a manufacturing process: *material handling*, *manipulation*, and *measurement*.

In a manufacturing process, each object has to be transferred from one location of the factory to another in order to be stored, manufactured, assembled, and packed. During transfer, the physical characteristics of the object do not undergo any alteration. The robot's capacity to pick up an object, move it in space on predefined paths and release it makes the robot itself an ideal candidate for material handling operations. Typical applications include:

- Palletizing (placing objects on a pallet in an ordered way),
- Warehouse loading and unloading,
- Mill and machine tool loading and unloading,

- Part sorting,
- Packaging.

Manufacturing consists of transforming objects from raw material into finished products: during this process, the part either changes its own physical characteristics as a result of machining or loses its identity as a result of an assembly of more parts. The robot's capacity to manipulate both objects and tools make it suitable to be employed for manufacturing. Typical applications include:

- Arc and spot welding,
- Spray painting,
- Milling and drilling,
- Gluing and scaling,
- Laser and water jet cutting,
- Deburring and grinding,
- Assembly of mechanical and electrical groups,
- Assembly of electronic boards,
- Screwing,
- Wiring.

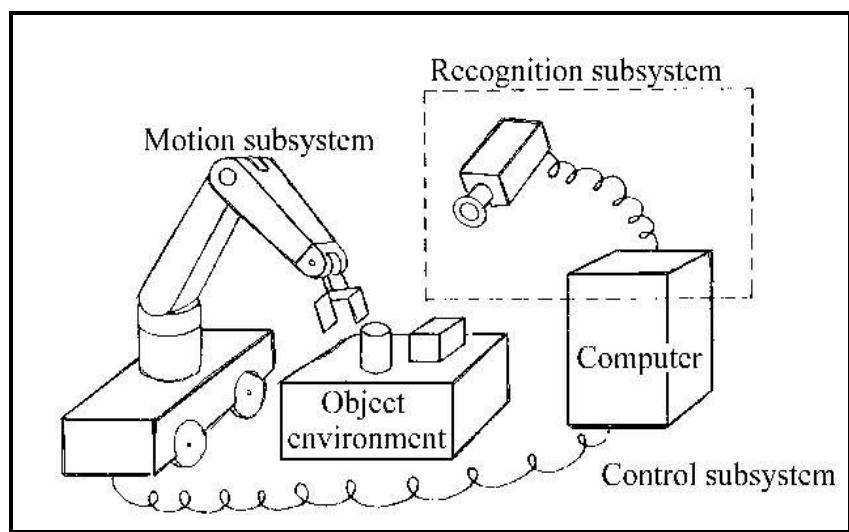
Besides material handling and manipulation, in a manufacturing process it is necessary to perform measurements to test product quality. The robot's capacity to explore the three-dimensional space together with the availability of measurements on the manipulator's status allows using a robot as a measuring device. Typical applications include:

- Object inspection,
- Contour finding,
- Detection of manufacturing imperfections.

The listed applications describe the current employment of robots as components of industrial automation systems. They all refer to strongly structured working environments and thus they do not exhaust all the possible utilizations of robots for industrial applications. The fall-outs of advanced robotics products may be of concern for industrial robotics whenever one attempts to solve problems regarding the adaptation of the robot to a changeable working environment.

## I.2 MANIPULATOR AND CONTROLLER

A robot system generally consists of three subsystems: a motion subsystem, a recognition subsystem, and a control subsystem (Figure I.1). The motion subsystem is the physical structure that carries out desired motions, corresponding to human arms or legs. The recognition subsystem uses various sensors to gather information about any objects being acted upon, about the robot itself, and about the environment; it recognizes the robot's state, the objects, and the environment from the gathered information. The control subsystem influences the motion subsystem to achieve a given task using the information from the recognition subsystem.

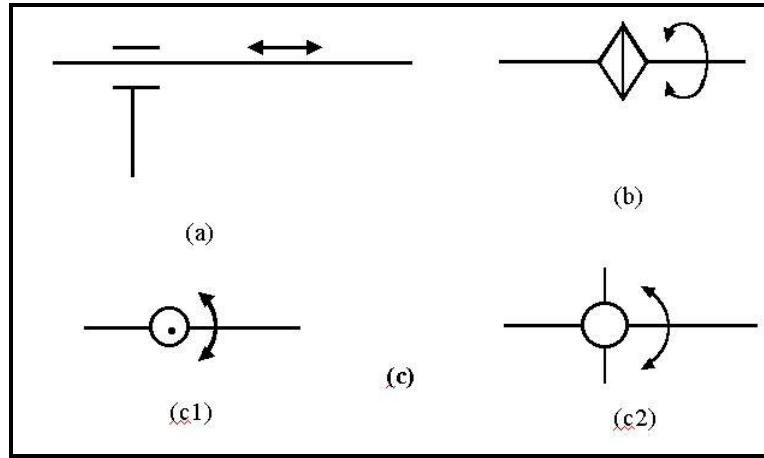


**Figure I.1**      **Robot system [2]**

### I.2.1 Mechanisms of Manipulators

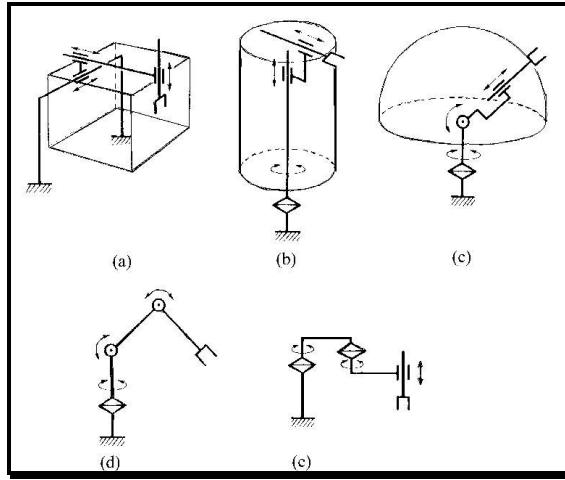
Robot manipulators can be regarded as open-loop link mechanisms consisting of several links connected together by joints. Typical joints are *revolute joints* and

*prismatic joints*, which are represented by the symbols shown in Figure I.2. Joint c in this figure is sometimes called the *pivot joint* to distinguish it from joint b. The endpoint of the mechanism is moved by driving these joints with appropriate actuators. A manipulator can usually be divided into an arm portion, a wrist portion, and a hand portion. Several typical mechanisms of the arm and wrist portions will be shown. The hand is not addressed here, since its mechanism depends on the task to be performed.



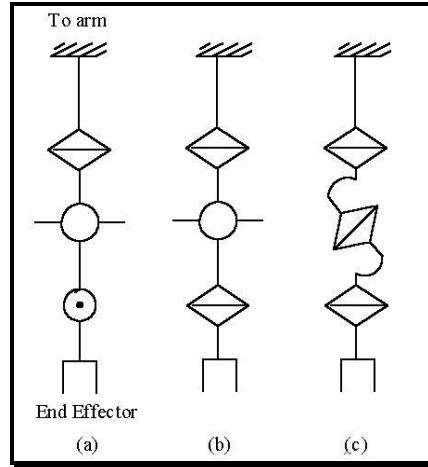
**Figure I.2** Symbols of joints (arrows show direction of motion). (a) Prismatic joint (b) Revolute Joint. (c) Revolute Joint (c1) Up-and-down rotation (c2) Back-and-forth rotation [2]

Figure I.3 shows several types of arm mechanisms: (a) the orthogonal-coordinate type, (b) the cylindrical-coordinate type, (c) the polar-coordinate type, (d) the vertical multi-joint type, and (e) the horizontal multi-joint type. Type a is structurally simple and rigid, and so its positioning accuracy is high. Types b – e are inferior to type a in positioning accuracy; however, they need less floor area for a base, and they have broader reach. Every mechanism in Figure I.3 has three degrees of freedom, which is the minimum number of degrees of freedom needed for placing the endpoint of the arm at an arbitrary point in three-dimensional space. Here the degree of freedom is defined as the minimal number of position variables necessary for completely specifying the configuration of a mechanism.



**Figure I.3** Arm mechanisms. (a) Orthogonal-coordinate type. (b) Cylindrical-coordinate type. (c) polar coordinate type (d) Vertical multi joint type  
(e) Horizontal multi joint type [2]

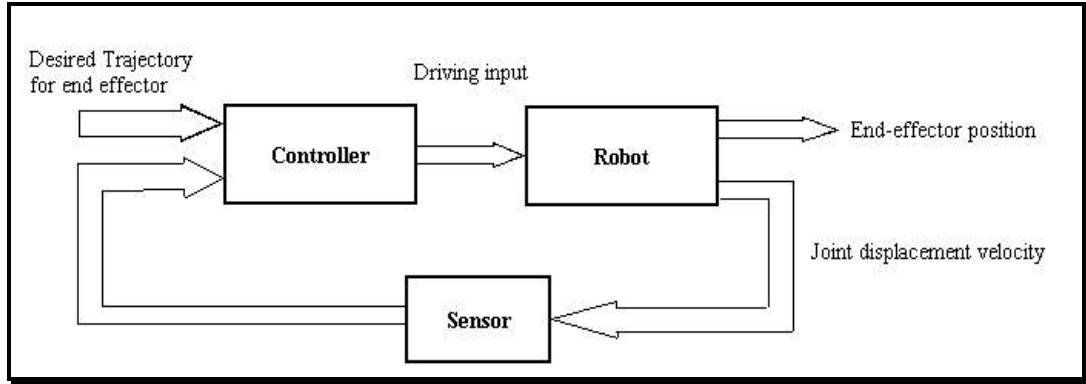
The wrist is connected to the end of the arm portion. The main role of the wrist is to change the orientation of the hand. Examples of wrist mechanism are shown in Figure I.4.



**Figure I.4** Wrist mechanisms [2]

## I.2.2 Controllers

The fundamental elements of tasks performed by robot manipulators are to move the end-effector, with or without a load, along a desired trajectory and to exert a desired force on an object when the end-effector is in contact with it. The former is called *position control* (or *trajectory control*) and the latter *force control*.



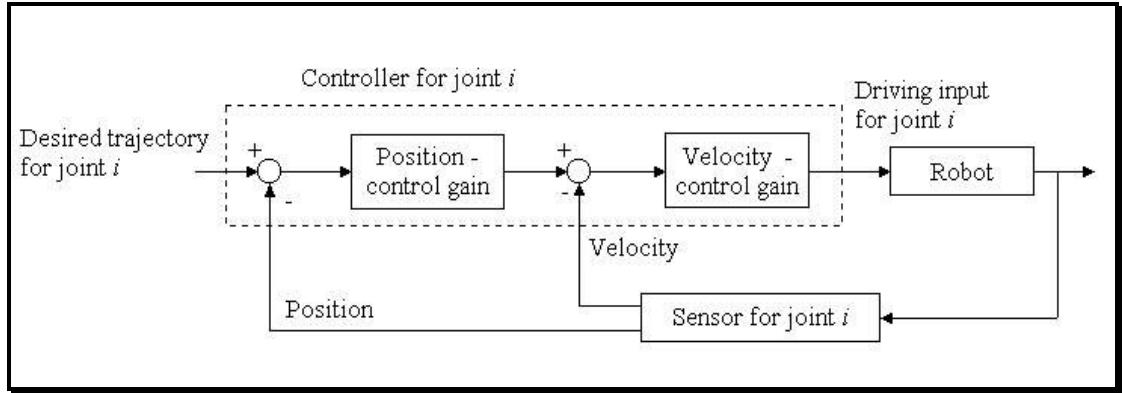
**Figure I.5      Rough sketch of position-control system [2]**

Joint positions and velocities are generally measured by joint sensors, such as potentiometers, tachometer-generators, encoders, and/or resolvers. Using these data, the controller determines the inputs to the joint actuators so that the end-effector follows the desired trajectory (position and/or force) as closely as possible (Figure I.5).

Figure I.6 shows a simple servomechanism that uses the position and velocity feedback with constant feedback gains. This kind of controller has often been used for general industrial manipulators. Generally, changes in dynamics are due to changes in the manipulator configuration; there is also interaction among the joints.

Controller design in Figure I.6 implicitly assumes the possibility of coping with changes in manipulator dynamics and joint interactions by regarding them as disturbances. However, when there are severe demands for fast and accurate positioning, the tracking performance of this type of controller is no longer adequate. Eventually it would be necessary to return to Figure I.5 and to design the controller taking into consideration the interaction among joints and the change of dynamics.

In force control, it is generally necessary to measure the forces driving the joints or the contact force between the end-effector and the object by force sensors, and to feed these signals back to the controller.



**Figure I.6 Position-velocity feedback servosystem. [2]**

### I.3 SYSTEM IDENTIFICATION

The system modeling and identification has become an important problem because of a large number of applications in diverse fields like chemical processes, biomedical systems, social and economic systems, transportation, ecology, electric power systems, aeronautics and engineering. In each of these cases, a model consists of basic mathematical equations, which can be used for understanding the behavior of the system, and wherever possible, for prediction and control. [8-9]

Two basic types of modeling problems arise. In the first type, one can associate with each physical phenomenon, a small number of measurable causes (inputs) and a small number of measurable effects (outputs). The outputs and the inputs can generally be related through a set of mathematical equations, in most cases nonlinear partial differential equations. The determination of these equations is the problem in such cases. These can be obtained either by writing a set of equilibrium equations based on mass and energy balance and other physical laws, or by using the “*black-box*” approach which consists of determining the equations from past records of the inputs and the outputs.

Modeling problems of this type appear quite often in engineering practice. Some typical problems are modeling of (i) a stirred-tank chemical reactor, (ii) a multi-machine electrical power system, (iii) a synchronous-orbital communications satellite, and (iv) the control mechanism of a nuclear power reactor (v) dynamic parameters of a mechanism. In each of these examples one can easily identify certain

input, output quantities, and then obtain the mathematical model relating them.

Another type of modeling problem arises in those situations where although we can identify a certain quantity as a definite measurable output or effect the causes are not so well defined. Some typical examples are (1) the annual population of a country, (2) the annual rainfall in a certain country, (3) the average annual flow in a river, and (4) the daily value of a certain stock in the stock market. In all these cases, we have available sequence of outputs, which will be called a time series, but the inputs or causes are numerous and not quite known in addition to often being unobservable. Nevertheless, it is important to develop a model in order that one may have some understanding of the process, which may be used for planning. The models in such cases are called stochastic models, due to a certain amount of uncertainty, which is unavoidable.

Modeling problems in engineering is important subject because the internal parameters that means the mass, the location of center of mass, and the moments of inertia of each rigid body link of a robot are usually not known even to the manufacturers of the robots [4]. Robots are usually designed to satisfy the kinematic specifications, but the inertial parameters are incidental attributes. Since commercial robots are invariably controlled by simple independent-joint PID control, there is no impetus by the manufacturer to determine the inertial parameters accurately since a dynamic model is not used.

Link materials can be determined experimentally by disassembling the robot, and weighting the pieces for mass, counterbalancing for center of mass, and swinging for moments of inertia. However this procedures introduces huge amount of measurement difficulties. Another method is CAD modeling of the parts, which requires intensive human involvement. CAD modeling is also subjected to modeling errors.

Controlling a robot involves both position and force variables. The role parameter identification in position control has been extensively elaborated, but less so in force control. Therefore parameter identification of the robot will be considerably helpful for both position and force control.

## **PART II**

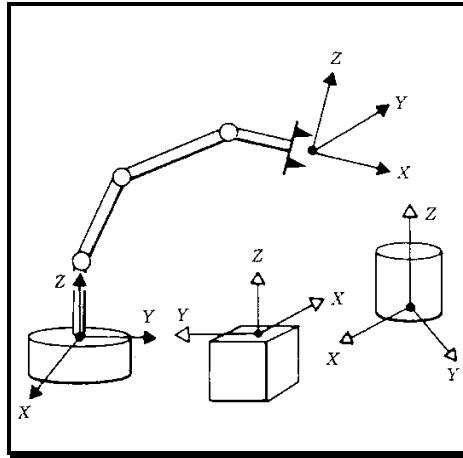
## **II GENERAL BACKGROUND**

### **II.1 MECHANICS AND CONTROL OF MANIPULATORS**

The following sections introduce some terminology and briefly preview each of the topics, which will be covered in the report.

#### **II.1.1 Description of Position and Orientation**

In the study of robotics, we are constantly concerned with the location of objects in three-dimensional space. These objects are the links of the manipulator, the parts and tools with which it deals, and other objects in the manipulators environment. At a crude but important level, these objects are described by just two attributes: their position and their orientation. Naturally, one topic of immediate interest is the manner in which we represent these quantities and manipulate them mathematically.



**Figure II.1 Coordinate systems or "frames" are attached to the manipulator and objects in the environment. [3]**

In order to describe the position and orientation of a body in space we will always attach a coordinate system, or *frame*, rigidly to the object. We then proceed to describe the position and orientation of this frame with respect to some reference coordinate system (Figure II.1).

Since any frame can serve as a reference system within which to express the position and orientation of a body, we often think of *transforming* or *changing the description* of these attributes of a body from one frame to another.

## II.1.2 Forward Kinematics of Manipulators

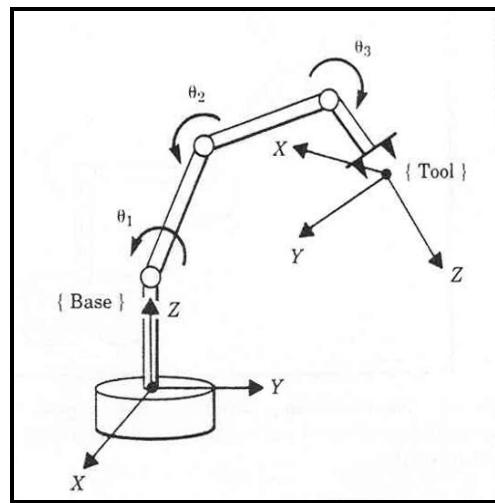
*Kinematics* is the science of motion, which treats motion without regard to the forces, which cause it. Within the science of kinematics, one studies the position, velocity, acceleration, and all higher order derivatives of the position variables (with respect to time or any other variable(s)). Hence, the study of the kinematics of manipulators refers to all the geometrical and time-based properties of the motion.

Manipulators consist of nearly rigid links, which are connected with joints, which allow relative motion of neighboring links. These joints are usually instrumented with position sensors, which allow the relative position of neighboring links to be measured. In the case of rotary or revolute joints, these displacements are called joint angles. Some manipulators contain sliding, or prismatic joints in which the relative displacement between links is a translation, sometimes called the joint

offset.

The number of degrees of freedom that a manipulator possesses is the number of independent position variables, which would have to be specified in order to locate all parts of the mechanism. This is a general term used for any mechanism. For example, a four-bar linkage has only one degree of freedom (even though there are three moving members). In the case of typical industrial robots, because a manipulator is usually an open kinematic chain, and because each joint position is usually defined with a single variable, the number of joints equals the number of degrees of freedom.

At the free end of the chain of links, which make up the manipulator is the end-effector. Depending on the intended application of the robot, the end-effector may be a gripper, welding torch, electromagnet, or other device. We generally describe the position of the manipulator by giving a description of the tool frame, which is attached to the end-effector, relative to the base frame, which is attached to the nonmoving base of the manipulator (Figure II.2).



**Figure II.2 Kinematic equation describes the tool frame relative to the base frame as a function of the joint variables. [3]**

A very basic problem in the study of mechanical manipulation is that of *forward kinematics* [1-3]. This is the static geometrical problem of computing the position and orientation of the end-effector of the manipulator. Specifically, given a set of joint angles, the forward kinematics problem is to compute the position and orientation of the tool frame relative to the base frame. Sometimes we think of this as

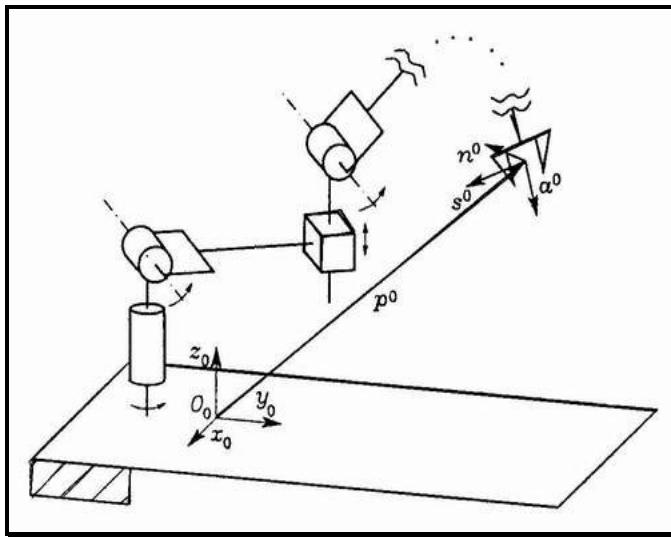
changing the representation of manipulator position from a joint space description into a Cartesian space description. By Cartesian space, we mean the space in which the position of a point is given with three numbers and in which the orientation of a body is given with three numbers. It is sometimes called task space or operational space.

A manipulator consists of series of rigid bodies (links) connected by joints. Each joint provides the mechanical structure a single degree of mobility, corresponding to the *joint variable* in joint space. Derivation of the direct kinematics equation allows expressing the end-effector position and orientation in operation space as a function of the joint variables of the mechanical structure with respect to the reference frame.

Consider a manipulator constituted by  $n + 1$  links connected by  $n$  joints. The aim of the direct kinematics is to determine the end-effector position and orientation as a function of the joint variables. In Figure II.3 the joints are numbered  $1, 2, \dots, n$ , starting from the base of the manipulator. The displacement of joint  $i$  is denoted  $q_i$  and called the joint variable. The collection of joint variables

$$\mathbf{q} = [q_1, q_2, \dots, q_n]^T \quad (\text{II.1})$$

is called the joint vector.



**Figure II.3 Description of the position and orientation of the end-effector frame [1]**

The position of the end-effector  $\mathbf{r}$  denoted by the  $m$ -dimensional vector

$$\mathbf{r} = [r_1, r_2, \dots, r_m]^T \quad (\text{II.2})$$

where  $m \leq n$ . For a general case where the end-effector can take an arbitrary position and orientation in three-dimensional Euclidean space, we have  $m = 6$ .

The relation between  $\mathbf{r}$  and  $\mathbf{q}$ , determined by the manipulator mechanism, generally is nonlinear. We assume that this relation is given by

$$\mathbf{r} = f_r(\mathbf{q}) \quad (\text{II.3})$$

This equation is called the *kinematic equation* of the manipulator.

It is well known from *Homogeneous Transformations* [1] that the position and orientation of a body with respect to a reference frame are described by the position vector of the origin and the unit vectors of a frame attached to the body. Hence, with respect to reference frame  $O_o-x_o y_o z_o$ , the direct kinematics function is expressed by the homogeneous transformation matrix

$$T^o(\mathbf{q}) = \begin{bmatrix} n^o(\mathbf{q}) & s^o(\mathbf{q}) & a^o(\mathbf{q}) & p^o(\mathbf{q}) \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{II.4})$$

### II.1.2.1 Denavit-Hartenberg Convention

The construction of an operating procedure for the computation of direct kinematics is naturally derived from the typical open kinematic chain of the manipulator structure. In fact, since each joint connects two and only two consecutive links, it is reasonable to consider first description of kinematic relationship between consecutive links and then to obtain the overall description of kinematic manipulator kinematics in a recursive fashion. To this purpose, the material presented in the previous sections for describing position and orientation of a rigid body is useful for obtaining composition of coordinate transformations between consecutive frames [1-3].

## II.1.3 Inverse Kinematics of Manipulators

If we return to the Forward Kinematics, the relation between  $r$  and  $q$ , determined by the manipulator mechanism, generally is nonlinear and assume that this relation is given by

$$\mathbf{r} = f_r(\mathbf{q}) \quad (\text{II.5})$$

This equation is called the *kinematic equation* of the manipulator [1-3]. When the joint vector  $\mathbf{q}$  is given, the corresponding  $\mathbf{r}$  is determined uniquely and the calculation is rather simple. However, when some task is assigned to the manipulator, what is given first is usually its end-effector position  $\mathbf{r}$  or a trajectory of  $\mathbf{r}$ . Thus we have to calculate a joint vector  $\mathbf{q}$ , which will realise the required end-effector position  $\mathbf{r}$  – that is, we have to obtain the vector  $\mathbf{q}$  satisfying equation (II.7). This solution can be written formally as

$$\mathbf{q} = f_r^{-1}(\mathbf{r}) \quad (\text{II.6})$$

Note, however, that  $\mathbf{q}$  does not necessarily exist, and even when it does exist it may not be unique. The problem of obtaining  $\mathbf{r}$  for a given  $\mathbf{q}$  is called the *inverse kinematics problem*. As we can see from the above argument, the inverse kinematics problem is usually the more difficult of the two.

In the problem of *inverse kinematics*, with the given position and orientation of the end-effector of the manipulator, we will calculate all possible sets of joint angles which could be used to attain this given position and orientation. This is a fundamental problem in the practical use of manipulators.

The inverse kinematic problem is not as simple as the forward kinematics. Because the kinematic equations are nonlinear, their solution is not always easy or even possible in a closed form. In addition, the questions of existence of a solution, and of multiple solutions, arise.

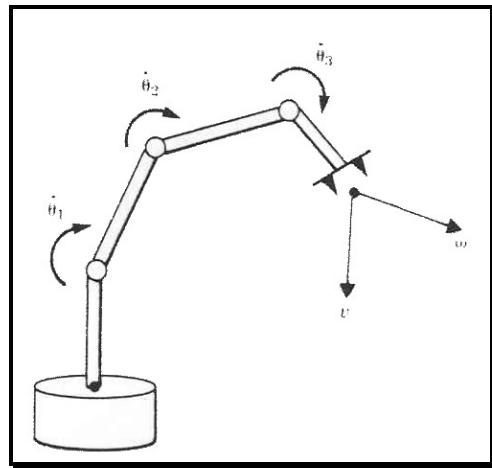
The existence or nonexistence of a kinematic solution defines the *workspace* of a given manipulator. The lack of a solution means that the manipulator cannot attain the desired position and orientation because it lies outside of the manipulator's workspace.

## II.1.4 Velocities, Static Forces, Singularities

In addition to dealing with static positioning problems, we may wish to analyze manipulators in motion. Often in performing velocity analysis of a mechanism, it is convenient to define a matrix quantity called the *Jacobian* of the

manipulator.

The Jacobian specifies a *mapping* from velocities in joint space to velocities in Cartesian space (Figure II.4). The nature of this mapping changes as the configuration of the manipulator varies. At certain points, called *singularities*, this mapping is not invertible. An understanding of the phenomenon is important to designers and users of manipulators. Manipulators do not always move through space: sometimes they are also required to contact a workpiece or work surface and apply a static force. In this case, the problem arises: Given a desired contact force and moment, what set of *joint torques* are required to generate them? Again, the Jacobian matrix of the manipulator arises quite naturally in the solution of this problem [3].



**Figure II.4** The geometrical relationship between joint rates and velocity of the end-effector can be described in a matrix called the Jacobian [3]

## II.1.5 Dynamics

*Dynamics* is a huge field of study devoted to studying the forces required to cause motion. In order to accelerate a manipulator from rest, glide at a constant end-effector velocity, and finally decelerate to a stop, a complex set of torque functions must be applied by the joint actuators. The exact form of the required functions of actuator torque depend on the spatial and temporal attributes of the path taken by the end-effector as well as the mass properties of the links and payload, friction in the joints, etc. One method of controlling a manipulator to follow a desired path involves calculating these actuator torque functions using the dynamic equations of motion of

the manipulator.

A second use of the dynamic equations of motion is in *simulation*. By reformulating the dynamic equations so that acceleration is computed as a function of actuator torque, it is possible to simulate how a manipulator would move under application of a set of actuator torques.

Dynamic modeling of a robot manipulator consists of finding the relationship between the forces exerted on the structure and the joint positions, velocities and accelerations. The *dynamic model* can be derived using the Lagrange formulation [1-5]. Since the joint variables  $q_i$  constitute a set of generalized coordinates for the mechanical system, the Lagrange equations can be written as

$$\frac{d}{dt} \left( \frac{\partial L}{\partial \dot{q}_i} \right) - \frac{\partial L}{\partial q_i} = \xi_i \quad i=1,\dots,n \quad (\text{II.7})$$

where

$$L = K - U \quad (\text{II.8})$$

is the Lagrangian given by the difference between *kinetic energy* ( $K$ ) and *potential energy* ( $U$ ) and  $\xi_i$  is the generalized force at joint  $i$ ; a torque for a revolute joint and a force for a prismatic joint, respectively. Typically, the generalized forces are shortly referred to as torques, since most joints of a manipulator are revolute.

The kinetic energy is a quadratic form of the joint velocities, i.e.

$$K = \frac{1}{2} \dot{q}^T \mathbf{B}(q) \dot{q} \quad (\text{II.9})$$

where the ( $n \times n$ ) matrix  $\mathbf{B}(q)$  is the inertia matrix of the robot manipulator which is symmetric and positive definite. Substituting II.9 in II.8 and taking the derivatives needed by II.7 leads the equation of motion

$$\mathbf{B}(q) \ddot{q} + \mathbf{C}(q, \dot{q}) \dot{q} + \mathbf{g}(q) = \xi \quad (\text{II.10})$$

where  $\xi$  is the ( $n \times 1$ ) vector of joints torques,  $\mathbf{g}(q)$  is the ( $n \times 1$ ) vector of gravity torques with

$$\mathbf{g}_i(q) = \frac{\partial U}{\partial q_i} \quad (\text{II.11})$$

and

$$\mathbf{C}(q, \dot{q})\dot{q} = \dot{\mathbf{B}}(q, \dot{q})\dot{q} - \frac{1}{2} \left( \frac{\partial}{\partial q} (\dot{q}^T \mathbf{B}(q) \dot{q}) \right)^T \quad (\text{II.12})$$

is the ( $n \times 1$ ) vector of Coriolis and centrifugal torques. This term is quadratic in the joint velocities, and thus its generic element can be written as

$$\mathbf{C}_{ij} = \sum_{k=1}^n c_{ijk} \dot{q}_j \dot{q}_k \quad (\text{II.13})$$

There exist several choices of the elements  $\mathbf{C}_{ij}$  of the matrix  $\mathbf{C}$  satisfying (II.13) corresponding to different factorizations of the term  $\mathbf{C}(q, \dot{q})\dot{q}$ . The choice

$$c_{ijk} = \frac{1}{2} \left( \frac{\partial B_{ij}}{\partial q_k} + \frac{\partial B_{ik}}{\partial q_j} + \frac{\partial B_{jk}}{\partial q_i} \right) \quad (\text{II.14})$$

where the  $c_{ijk}$ 's are termed *Christoffel symbols* of the first type makes the matrix  $\dot{\mathbf{B}} - 2\mathbf{C}$  skew-symmetric; this property is very useful for control design purposes.

Regarding the joint torque, each joint is driven by an actuator (direct drive or gear drive); in general, the following torque contributions appear

$$\xi_i = \tau_{ci} - \tau_{fi} - \tau_{ei} \quad (\text{II.15})$$

where  $\tau_{ci}$  is the driving (control) torque at the joint,  $\tau_{fi}$  is the torque due to joint friction, and  $\tau_{ei}$  is the torque caused by the external force and moment exerted by the end-effector when in contact with the environment. Note that the actuators (electric or hydraulic) have been assumed as ideal torque generators.

Joint friction is difficult to model accurately; as a simplified model, only viscous friction is considered, i.e.

$$\tau_f = \mathbf{F}_d \dot{q} \quad (\text{II.16})$$

where  $\mathbf{F}_d$  is a positive definite ( $n \times n$ )(diagonal) matrix of viscous friction coefficients at the joints. Models that are more complex would include nonlinear phenomena such as Coulomb friction.

Finally, let  $\mathbf{f}_e$  denote the ( $3 \times 1$ ) vector of external *end-effector force* and  $\boldsymbol{\mu}_e$  the ( $3 \times 1$ ) vector of external *end-effector moment*. By applying the principle of virtual work, the resulting joint torque is

$$\boldsymbol{\tau}_e = \mathbf{J}^T(q)\mathbf{h} \quad (\text{II.17})$$

where

$$\mathbf{h} = \begin{bmatrix} \mathbf{f}_e \\ \boldsymbol{\mu}_e \end{bmatrix} \quad (\text{II.18})$$

and  $\mathbf{J}$  is defined  $\dot{\mathbf{r}} = \mathbf{J}(q)\dot{\mathbf{q}}$

In the view of (II.15), (II.16) and (II.17), the dynamic model can be written in the form

$$\mathbf{B}(q)\ddot{\mathbf{q}} + \mathbf{C}(q, \dot{q})\dot{\mathbf{q}} + \mathbf{F}_d\dot{\mathbf{q}} + \mathbf{g}(q) = \boldsymbol{\tau}_e - \mathbf{J}^T(q)\mathbf{h} \quad (\text{II.19})$$

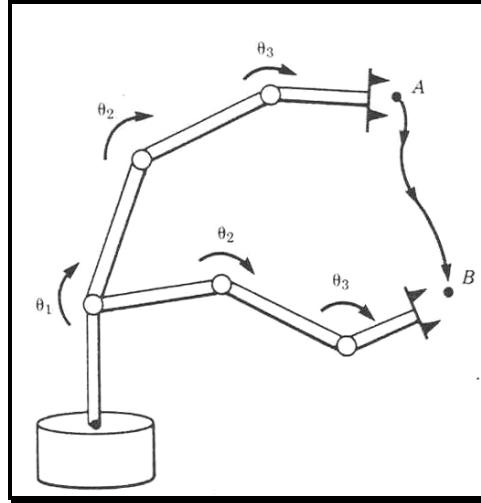
It can be shown that equation (II.19) can be cast in a linear form with respect to a suitable ( $p \times 1$ ) vector  $\boldsymbol{\pi}$  of dynamic parameters as

$$\mathbf{Y}(q, \dot{q}, \ddot{q})\boldsymbol{\pi} = \boldsymbol{\tau}_e - \mathbf{J}^T(q)\mathbf{h} \quad (\text{II.20})$$

where the ( $n \times p$ ) matrix  $\mathbf{Y}$  is termed *regressor* of the dynamic model. In general, the dynamic parameters depend on the mass, first moment of inertia and inertia tensor of each link, and the friction coefficient of each joint.

## II.1.6 Trajectory Generation

A common way of causing a manipulator to move from here to there in a smooth, controlled fashion is to cause each joint to move as specified by a smooth function of time. Commonly, each joint starts and ends its motion at the same time, so that the manipulator motion appears coordinated. Exactly how to compute these motion functions is the problem of *trajectory generation* (Figure II.5) [1-3].



**Figure II.5** In order to move the end-effector through space from point A to point B, we must compute a trajectory for each joint to follow. [3]

The goal of *trajectory planning* is to generate the reference inputs to the motion control system, which ensures that the manipulator executes the planned trajectories. The user typically specifies a number of parameters to describe the desired trajectory. Planning consists of generating a time sequence of the values attained by a polynomial function interpolating the desired trajectory. Two techniques exist for trajectory generation, first one is the case when the initial and final point of the path are assigned (*point-to-point motion*), and the second one is the case when a finite sequence of points are assigned along the path (*path motion*).

## II.1.7 Position Control

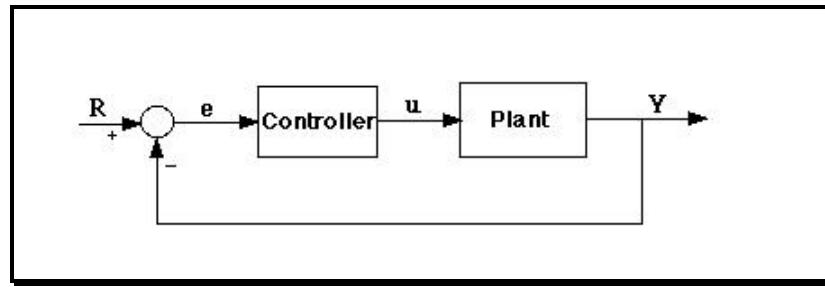
Some manipulators are equipped with stepper motors or other actuators which can directly execute a desired trajectory. However, the vast majority of manipulators are driven by actuators, which supply a force or a torque to cause motion of the links. In this case, an algorithm is needed to compute torques, which will cause the desired motion. The problem of dynamics is central to the design of such algorithms but does not constitute a solution.

A primary concern of a *position control system* is to automatically compensate for errors in knowledge of the parameters of a system, and to suppress disturbances, which tend to perturb the system from the desired trajectory [1-3]. To accomplish this, position and velocity *sensors* are monitored by the *control*

algorithm, which computes torque commands for the actuators.

### II.1.7.1 PID Control

A feedback controller is designed to generate an output that causes some corrective effort to be applied to a process so as to drive a measurable process variable towards a desired value known as the *setpoint*. The controller uses an actuator to affect the process and a sensor to measure the results.



**Figure II.6      Feedback control system**

Virtually all feedback controllers determine their output by observing the *error* between the set point and a measurement of the process variable. Errors occur when an operator changes the set point intentionally or when a disturbance or a load on the process changes the process variable accidentally. The controller's mission is to eliminate the error automatically and to control the overall *system* behavior.

In the Figure II.6 the variable ( $e$ ) represents the tracking error, the difference between the desired input value ( $R$ ) and the actual output ( $Y$ ). This error signal ( $e$ ) will be sent to the PID controller, and the controller computes both the derivative and the integral of this error signal. The signal ( $u$ ) just past the controller is now equal to the proportional gain ( $K_P$ ) times the magnitude of the error plus the integral gain ( $K_I$ ) times the integral of the error plus the derivative gain ( $K_D$ ) times the derivative of the error.

$$u = K_P e + K_I \int e dt + K_D \frac{de}{dt} \quad (\text{II.21})$$

This signal ( $u$ ) will be sent to the plant, and the new output ( $Y$ ) will be obtained. This new output ( $Y$ ) will be sent back to the sensor again to find the new error signal ( $e$ ). The controller takes this new error signal and computes its derivative

and its integral again. This process goes on and on.

If we again consider the general manipulator dynamic equation (II.19) assuming no interaction with the environment,

$$\mathbf{B}(q)\ddot{\mathbf{q}} + \mathbf{C}(q, \dot{q})\dot{\mathbf{q}} + \mathbf{F}_d\dot{\mathbf{q}} + \mathbf{g}(q) = \boldsymbol{\tau}_c \quad (\text{II.22})$$

then the control torque that will be applied to the joints of the manipulator will be in the following form with PD type controller:

$$\boldsymbol{\tau}_c = \mathbf{K}_v(\dot{\mathbf{q}}_d - \dot{\mathbf{q}}) + \mathbf{K}_p(\mathbf{q}_d - \mathbf{q}) \quad (\text{II.23})$$

where  $\mathbf{K}_v$  and  $\mathbf{K}_p$  are (nxn) diagonal matrices of velocity and position gains,  $\mathbf{q}_d$  and  $\dot{\mathbf{q}}_d$  are the (nx1) desired position and velocity vectors,  $\mathbf{q}$  and  $\dot{\mathbf{q}}$  are (nx1) actual position and velocity vectors of the manipulator respectively.

### II.1.7.2 Feedforward and Computed Torque Control

The accuracy of the manipulator dynamic model impinges on the performance of feedforward and computed torque control. Since friction is negligible for direct drive arms, and presuming that one has good control of joint torques, the issue of accuracy reduces to how well the inertia parameters of the rigid links are known. These parameters can be obtained by the use of cad models of the manipulators or can be estimated by identification methods described in section II.2.

#### Controller Algorithms

The simplest and most common form of robot control is independent joint PD control mentioned above is described by

$$\boldsymbol{\tau}_c = \mathbf{K}_v(\dot{\mathbf{q}}_d - \dot{\mathbf{q}}) + \mathbf{K}_p(\mathbf{q}_d - \mathbf{q}) \quad (\text{II.24})$$

where  $\mathbf{K}_v$  and  $\mathbf{K}_p$  are (nxn) diagonal matrices of velocity and position gains.

The common feedforward controller is the PD controller with gravity compensation, described by

$$\boldsymbol{\tau}_c = \mathbf{K}_v(\dot{\mathbf{q}}_d - \dot{\mathbf{q}}) + \mathbf{K}_p(\mathbf{q}_d - \mathbf{q}) + \mathbf{g}(q) \quad (\text{II.25})$$

where  $\mathbf{g}(q)$  represents the gravity term of the manipulator dynamics.

The feedforward controller augments the basic PD controller by providing a set of nominal torques  $\tau_{ff}$ :

$$\tau_{ff}(q_d, \dot{q}_d, \ddot{q}_d) = \mathbf{B}(q_d)\ddot{q}_d + \mathbf{C}(q_d, \dot{q}_d)\dot{q}_d + \mathbf{F}_d\dot{q}_d + \mathbf{g}(q_d) \quad (\text{II.26})$$

where  $\mathbf{B}$ ,  $\mathbf{C}$ ,  $\mathbf{F}_d$  and  $\mathbf{g}$  refers to the dynamic model values. When this equation is combined with (II.24), the feedforward controller results:

$$\tau_c = \tau_{ff}(q_d, \dot{q}_d, \ddot{q}_d) + \mathbf{K}_v(\dot{q}_d - \dot{q}) + \mathbf{K}_p(q_d - q) \quad (\text{II.27})$$

The feedforward term  $\tau_{ff}$  can be thought of as a set of nominal torques which allow the dynamics (II.22) to be linearized about the operating points  $q_d$ ,  $\dot{q}_d$  and  $\ddot{q}_d$ . Therefore, it is reasonable to add the linear feedback terms

$$\mathbf{K}_v(\dot{q}_d - \dot{q}) + \mathbf{K}_p(q_d - q)$$

as the control for the linearized system. These feedforward terms can be computed off-line, since they are functions of the parameters of the desired trajectory only.

On the other hand, the computed torque controller computes the dynamics on-line, using the sampled joint position and velocity data. The control equation is:

$$\tau_c = \mathbf{B}(q)\ddot{q}^* + \mathbf{C}(q, \dot{q})\dot{q} + \mathbf{F}_d\dot{q} + \mathbf{g}(q) \quad (\text{II.28})$$

where  $\ddot{q}^*$ , the new input to the system is given by:

$$\ddot{q}^* = \ddot{q}_d + \mathbf{K}_v(\dot{q}_d - \dot{q}) + \mathbf{K}_p(q_d - q) \quad (\text{II.29})$$

If the robot model is exact, then each link of the robot is decoupled, and the trajectory error goes to zero. The computed torque method has been considered by several investigators. Although simulation results have been present, there has been very few published reports on the actual implementation of this controller, mainly due to the lack of an appropriate manipulator or on-line computational facility. Also, an enhanced computed torque control algorithm is proposed in [56].

### II.1.7.3 Cartesian-Based Position Control

The position controllers discussed earlier (independent-joint PD control, the feedforward controller and the computed torque control) are based on joint coordinates. If the trajectory is initially specified in terms of Cartesian coordinates (task space) of the endpoint, then inverse kinematic transformations are required to convert to joint angles.

It is also possible to specify the control law in terms of Cartesian coordinates rather than in terms of joint coordinates. The reason that Cartesian-based position control has been proposed is that the control law is often best cast into the task variables according to which the trajectory is planned. This is particularly true of force control, where variables are partitioned into those that can be controlled for position versus those that are controlled for force. As an example, a *Cartesian-based PD position controller* could be defined by:

$$\mathbf{f} = \mathbf{K}_v(\dot{\mathbf{r}}_d - \dot{\mathbf{r}}) + \mathbf{K}_p(\mathbf{r}_d - \mathbf{r}) \quad (\text{II.30})$$

where  $\mathbf{f}$  is the endpoint force that the manipulator generates in response to a perturbation. Here,  $\mathbf{K}_v$  and  $\mathbf{K}_p$  are the velocity and position gains, respectively.  $\mathbf{r}$  and  $\mathbf{r}_d$  are the actual and desired position vectors of the end-effector, respectively. That is to say, the endpoint of the manipulator acts like a spring plus damper.

To evaluate this control law, joint positions and velocities must be converted to endpoint positions and velocities. The endpoint force can be converted to joint torques in several ways. One way is directly convert to joint torques by the relation  $\boldsymbol{\tau}_c = \mathbf{J}^T(q)\mathbf{f}$ :

$$\boldsymbol{\tau}_c = \mathbf{J}^T(q)(\mathbf{K}_v(\dot{\mathbf{r}}_d - \dot{\mathbf{r}}) + \mathbf{K}_p(\mathbf{r}_d - \mathbf{r})) \quad (\text{II.31})$$

This implementation of the Cartesian-based PD position controller is pure feedback controller and do not incorporate a dynamic model of the robot. One could define Cartesian-based position controllers analogous to both the feedforward controller and computed torque controller. In actual implementations and the literature, only the *Cartesian-based computed torque controller* has been proposed and is called *resolved acceleration position control*.

*Resolved acceleration position control* is quite similar to *computed torque control* (II.29), except that the desired trajectory and the feedback law are expressed in terms of task coordinates  $\mathbf{r}$ .

$$\ddot{\mathbf{r}}^* = \ddot{\mathbf{r}}_d + \mathbf{K}_v(\dot{\mathbf{r}}_d - \dot{\mathbf{r}}) + \mathbf{K}_p(\mathbf{r}_d - \mathbf{r}) \quad (\text{II.32})$$

In this equation direct kinematic transformations are required to compute the actual positions and velocities from the joint positions and velocities, and an inverse kinematic transformation is required to convert the nominal endpoint acceleration to a nominal joint acceleration. Differentiating the kinematic relation  $\dot{\mathbf{r}} = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}}$  and after some modification

$$\ddot{\mathbf{q}}^* = \mathbf{J}^{-1}(\mathbf{q})(\ddot{\mathbf{r}}^* - \dot{\mathbf{J}}\dot{\mathbf{q}}) \quad (\text{II.33})$$

This nominal joint acceleration is then substituted into the inverse dynamics to yield the joint torques,

$$\boldsymbol{\tau}_c = \mathbf{B}(\mathbf{q})\ddot{\mathbf{q}}^* + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{F}_d\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) \quad (\text{II.34})$$

## II.1.8 Nonlinear Position Control

Although control systems based on approximate linear models are popular in current industrial robots, it is important to consider the complete nonlinear dynamics of the manipulator when synthesizing control algorithms. Some industrial robots are now being introduced which make use of *nonlinear control* algorithms in their controllers. These nonlinear techniques of controlling a manipulator promise better performance than do simpler linear schemes.

## II.1.9 Interaction Control

Control of interaction between a robot manipulator and the environment is crucial for successful execution of a number of practical tasks where the robot end-effector has to manipulate an object or perform some operation on a surface. Typical examples include polishing, deburring, machining or assembly. A complete classification of possible robot tasks is practically infeasible in view of the large variety of cases that may occur, nor would such a classification be really useful to

find a general strategy to control *interaction with environment*.

During interaction, the environment sets constraints on the geometric paths that can be followed by the end-effector. This situation is generally referred to as *constraint motion* [7]. In such a case, the use of a purely *motion control* strategy for controlling interaction is a candidate to fail.

Successful execution of an interaction task with the environment by using motion control could be obtained only if the task were accurately planned. This would in turn require an accurate model of both the robot manipulator (kinematics and dynamics) and the environment (geometry and mechanical features). Manipulator modeling can be known with enough precision, but a detailed description of the environment is difficult to obtain. An estimation method for the environment properties is proposed in [4].

To understand the importance of task planning accuracy, it is sufficient to observe that to perform a mechanical part mating with a positional approach, the relative positioning of the parts should be guaranteed with an accuracy of an order of magnitude greater than part mechanical tolerance. Once the absolute position of one part is exactly known, the manipulator should guide the motion of the other with the same accuracy.

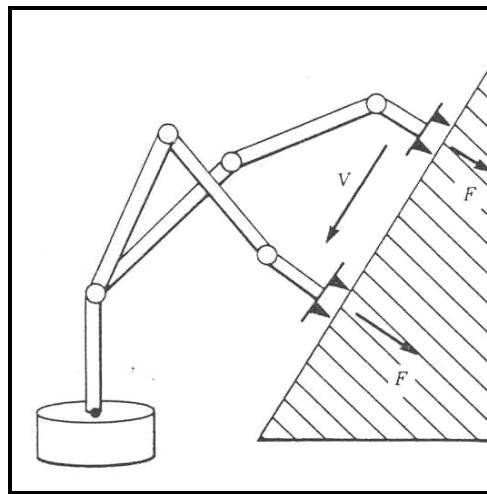
In practice, the planning errors may give rise to a *contact force* causing a deviation of the end-effector from the desired trajectory. On the other hand, the control system reacts to reduce such deviation, which leads to an increase of the contact force until saturation of the joint actuators or breakage of the parts in contact.

The higher the environment stiffness and position control accuracy are, the easier a situation like the one just described can occur. This drawback can be overcome if a *compliant behavior* is ensured during the interaction. This can be achieved either in a *passive* fashion by using a suitable compliant mechanical device between the end-effector and the environment, or in an *active* fashion by devising a *interaction control* strategy.

The contact force is the quantity describing the state of interaction in the most complete fashion. Therefore, it is expected that enhanced performance can be

achieved with an interaction control provided that force measurements are available. For this purpose, a force/torque sensor can be mounted on a robot manipulator, typically between the wrist and the end-effector, and its reading shall be passed to the robot control unit via a suitable interface.

The ability for a manipulator to control forces of contact when it touches parts, tools, or work-surfaces seems to be of great importance in applying manipulators to many real-world tasks. *Force control [1-6]* is complementary to position control in that we usually think of one or the other as applicable in a certain situation.



**Figure II.7** In order for a manipulator to slide across a surface while applying a constant force, a hybrid position-force control system must be used.  
[3]

When a manipulator is moving in free space, only position control makes sense, since there is no surface to react against. When a manipulator is touching a rigid surface however, position control schemes can cause excessive forces to build up at the contact or may cause contact to be lost with the surface when it was desired for some application. Since manipulators are rarely constrained by reaction surfaces in all directions simultaneously, using a mixed or *hybrid control* [10] is required with some directions controlled by a *position control law* and remaining directions controlled by a *force control law* (Figure II.7).

Many researchers investigate the control of the manipulators, interacting with their environments. The problem of control design of a robot to perform compliant

manipulation on dynamic environments is studied in [10], and many compliant motion control strategies of manipulators are reviewed in [11,12]. In [13], controlling the interaction of a robot manipulator with a compliant curved surface is considered. In [14], a new method for controlling the contact compliance via internal forces on objects held by dual-arm robots is proposed. And the compliance control of redundant manipulators is investigated in [15-17].

Researchers pay attention to the control of the manipulators, interacting with their environments, but the *stability analysis* of compliant motion control [18] and the physical interaction [19] is neglected by many of them.

The collision phenomena is an important concept in physical contact and dynamic interaction. [20] Presents an impact controller, which is robust to the uncertainties in environment dynamics and location of the collision surface, and then compares it with impedance control strategy via simulation. [21] Examined a little-studied method of manipulation – manipulation by striking an object and letting it to slide, which is an impact problem. Also a methodology to analyze the stability of a multi-link manipulator system with collision phenomena is proposed in [22,23].

### **II.1.9.1 Indirect vs. Direct Control**

Interaction control strategies can be grouped in two categories, those performing *indirect force control* and those performing *direct force control*. The main difference between the two categories is that the former achieves force control via motion control, without explicit closure of force feedback loop; the latter, instead, offer the possibility of controlling the contact force to desired value, thanks to the closure of force feedback loop.

The first category belong *compliance (or stiffness) control* [1-3,5] and *impedance control* [1-2,4-5], where the position error is related to the contact force through a mechanical stiffness or impedance of adjustable parameters. A robot manipulator under impedance control is described by an equivalent mass-spring-damper system with the contact force as input. The resulting impedance in the various task space directions is typically nonlinear and coupled. If a force/torque sensor is available, then force measurements can be used in the control law so as to

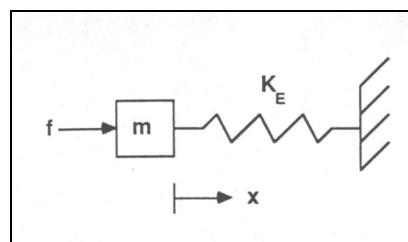
achieve a linear and decoupled impedance.

If a detailed model of the environment is available, a widely adopted strategy belonging to the second category is the *hybrid position/force control* [1-5], which aims at controlling position along the unconstrained task directions. A selection matrix acting on both desired and feedback quantities serves this purpose for typically planar contact surface, whereas the explicit constraint equations have to be taken into account for general curved contact surfaces.

In most practical situations, a detailed model of the environment is not available. In such a case, an effective strategy still in the second category is the *inner/outer motion/force control* where an outer force control loop is closed around the inner motion control loop, which is typically available in a robot manipulator. In order to embed the possibility of controlling motion along the unconstrained task directions, the desired end-effector motion can be input to the inner loop of an inner/outer motion/force control scheme. The resulting *parallel control* is composed of a force control action and a motion control action, where the former is designed so as to dominate the latter in order to ensure force control along the constrained task directions.

### II.1.9.2 Force control

Force control is the most general form of trajectory control, because the manipulator is allowed to contact the environment as it executes a trajectory. Instead of just position variables to plan and control, there are now additionally force variables to plan and control. When we use the term *force control*, we mean the simultaneous control of both force and position.



**Figure II.8** A simple model of the robot in contact with the environment

Many robot force controllers go unstable during hard contact, such as against

metal. The robot chatters uncontrollably, bouncing back and forth against the surface. To illustrate the problem, figure II.8 shows a simple model of the robot and its environment. The robot is modeled as a pure mass  $m$ , and the environment is modeled as a pure stiffness  $K_E$ . We presume there is a stiff and massless force sensor between the robot and environment that measures the contact force  $f_e = K_E x$ , where  $x$  is the displacement. Basic force control is the simple proportional controller, multiplying the force error  $f_d - f_e$  by a gain  $K_f$ , where  $f_d$  is the desired force giving the applied force from the force control law:

$$f = K_f(f_d - K_E x) \quad (\text{II.35})$$

The pertinent feature of this equation is that *force control* is essentially *high gain position control*. The stiffness of the environment multiplies the force control gain, yielding a large effective position gain of  $K_f K_E$ . Systems with such large feedback gains in general exhibit unstable behavior. Sources of instability include unmodeled dynamics, such as flexibility in the manipulator joints or links. Since flexibility is present in all manipulators, the chattering behavior mentioned earlier is seen in virtually all force controllers.

There are number of ways in which the dynamic instability can be overcome. One way is to dominate the stiffness of the environment with a soft skin or covering or attaching the force sensor to the robot end-effector via a soft spring. Disadvantages of this approach include loss of position resolution and a reduction in the speed of response. The damping of the controller can also be elevated to match the high position gain, but again the response speed would be slowed.

So, as in position control, if the end-effector force measurements are available, it is possible to regulate the contact force to a desired value. Force control can be entrusted to the adoption of a P action on the force error plus desired force feedforward. Let  $\mathbf{f}_d$  denotes the desired contact force, and  $\mathbf{f}_e$  denotes the external force acting to the end-effector, the driving (control) torques can be chosen as

$$\boldsymbol{\tau}_c = \mathbf{J}^T(\mathbf{q})(\mathbf{K}_p(\mathbf{K}_f(\mathbf{f}_d - \mathbf{f}_e) - \mathbf{r}) + \mathbf{f}_d) - \mathbf{K}_v \dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) \quad (\text{II.36})$$

Here  $\mathbf{K}_f$  is the proportional gain of the force error between the desired and the actual end-effector force error. Also, an inner loop on the end-effector position with the

gain  $\mathbf{K}_p$  is used in (II.36), which in turn correspond to leaving the proportional action for motion control in task space; this is in accordance with the fact that a position feedback is usually available in an industrial robot controller. No derivative gain is used on the force feedback, as it is highly corrupted by noise, and a derivative action on it cannot be implemented in practice. As an alternative, a damping action can be provided by joint velocity feedback, which in turn corresponds to using a inner velocity loop with the gain  $\mathbf{K}_v$ . The calculated force that must act to the end-effector in task space is converted to the joint space with the Jacobian matrix like in II.31.

This control scheme can be thought as a *static model based force control* as it only concerns the gravity torques and Jacobian model-based compensation requirements[24]. On the other hand, computing the dynamics on-line, using the sampled joint position and velocity data like in *resolved acceleration control*, the driving torque is

$$\tau_c = \mathbf{B}(q)\ddot{\mathbf{q}}^* + \mathbf{C}(q, \dot{q})\dot{\mathbf{q}} + \mathbf{F}_d\dot{\mathbf{q}} + \mathbf{g}(q) \quad (\text{II.37})$$

here  $\ddot{\mathbf{q}}^*$ , the new input to the system in joint space is related to the new input  $\ddot{\mathbf{r}}^*$  in task space by the Jacobian as

$$\ddot{\mathbf{q}}^* = \mathbf{J}^{-1}(q)(\ddot{\mathbf{r}}^* - \dot{\mathbf{J}}\dot{\mathbf{q}})$$

Now, the new input  $\ddot{\mathbf{r}}^*$  can be selected as

$$\ddot{\mathbf{r}}^* = \mathbf{K}_f(\mathbf{f}_d - \mathbf{f}_e) - \mathbf{K}_v\dot{\mathbf{q}} \quad (\text{II.38})$$

for *force control with inner velocity loop*, and as

$$\ddot{\mathbf{r}}^* = \mathbf{K}_p(\mathbf{K}_f(\mathbf{f}_d - \mathbf{f}_e) - \mathbf{r}) - \mathbf{K}_v\dot{\mathbf{q}} \quad (\text{II.39})$$

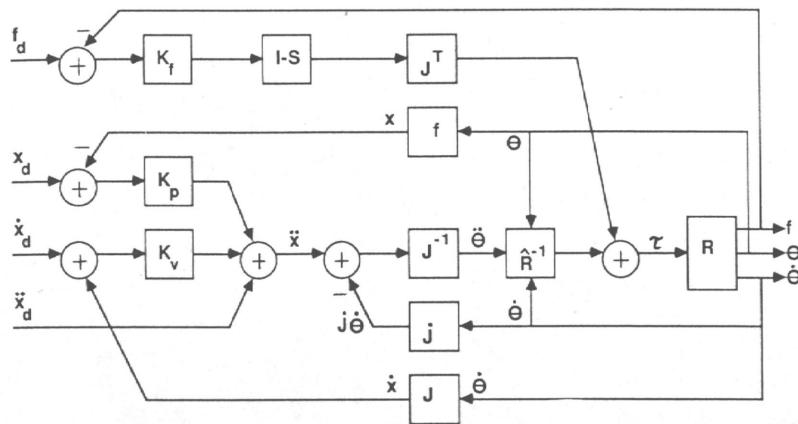
for *force control with inner position control* in order to eliminate the disturbance at steady state by adding an inner loop on the end-effector position, as already done in (II.36). If the robot model is exact, then each link of the robot is decoupled, and the error goes to zero for the two selections.

The theory of force control based on formal models of the manipulator and the task geometry is presented in [25]. The force control strategies are applied to an

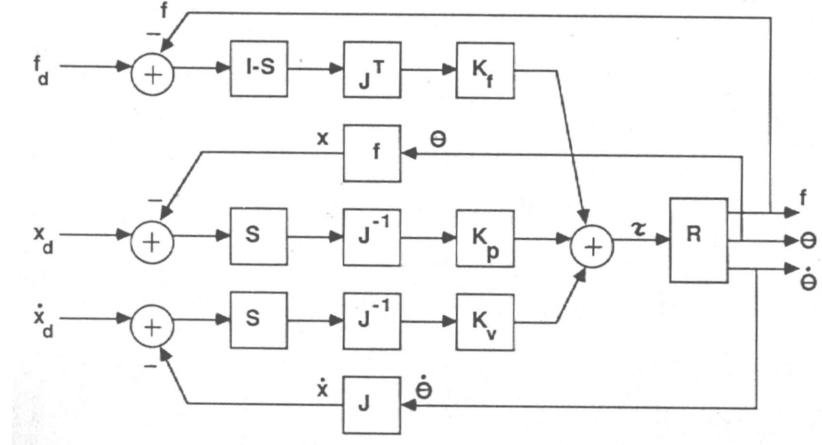
experimentally derived model for analysis in [26]. The problem of concurrent motion and force control for manipulator tasks with continuous or transient contact between end-effector and environment is addressed in [27]. A new sensor-referenced control method for robot impact control and force regulation with environmental uncertainties is developed in [28]. An adaptive model-based force control is proposed in [29], with experiments, which offers performance superior to that of its non-model-based counterpart over a wide variety of operation conditions. Two adaptive PID and PI force compensators, and two adaptive lag-plus-feedforward compliance compensators are proposed in [30]. And the force control algorithms are applied to real applications in [31-33] for grinding, chamfering, polishing and milling.

### II.1.9.3 Hybrid Position/Force Control

When the tip of the manipulator contacts the environment, it will be able to generate positions in certain directions and forces in other directions. Thus the geometry of the environment provides the best coordinate system to partition variables into position-controlled versus force-controlled and to plan the movement.



**Figure II.9**      **Hybrid position/force control**



**Figure II.10      Resolved acceleration force control**

Just as in position control, there are issues of feedback versus feedforward control, and of the role of a model in accurate trajectory tracking. In addition, there is an issue of instability as well; that is to say, not using a dynamic model can make force control unstable as well as inaccurate. Figures II.9 and II.10 illustrates two alternatives of using or not using a model in force control.

The controller in Figure II.9 is known as *hybrid position/force control*, and does not use a model of the robot.

$$\tau_c = \mathbf{K}_v \mathbf{J}^{-1}(q) \mathbf{S}(\dot{\mathbf{r}}_d - \dot{\mathbf{r}}) + \mathbf{K}_p \mathbf{J}^{-1}(q) \mathbf{S}(\mathbf{r}_d - \mathbf{r}) + \mathbf{K}_f \mathbf{J}^T(q) (\mathbf{I} - \mathbf{S})(\mathbf{f}_d - \mathbf{f}_e) \quad (\text{II.40})$$

The force error  $\mathbf{f}_d - \mathbf{f}_e$  is transformed to joint coordinates by the Jacobian matrix  $\mathbf{J}^T$  and then multiplied by the force gain  $\mathbf{K}_f$ . The force and position feedbacks (with the gains  $\mathbf{K}_p$  and  $\mathbf{K}_v$ ) are summed to provide torques to the manipulator joints.

The external variables are presumed to have been partitioned into position-controlled  $\mathbf{r}$  versus force-controlled  $\mathbf{f}_e$ , and desired trajectories  $\mathbf{r}_d$  and  $\mathbf{f}_d$  have been specified for each. In the figure, the partitioning is indicated by the projection matrices  $\mathbf{S}$  and  $\mathbf{I}-\mathbf{S}$ , where  $\mathbf{S}$  selects the variables to be position-controlled, and  $\mathbf{I}-\mathbf{S}$  selects the complementary force variables where  $\mathbf{I}$  is the identity matrix.

The results of an experimental investigation of hybrid/force control algorithms for a robot manipulator, whose end-effector is in contact with a nearly rigid spherical surface can be found in [34].

*Resolved acceleration force control* (Figure II.10) is a simply extension to

resolved acceleration position control, by adding a force loop:

$$\boldsymbol{\tau}_c = \boldsymbol{\tau}_{ff}(\boldsymbol{q}_d, \dot{\boldsymbol{q}}_d, \ddot{\boldsymbol{q}}_d) + \mathbf{K}_v(\dot{\boldsymbol{q}}_d - \dot{\boldsymbol{q}}) + \mathbf{K}_p(\boldsymbol{q}_d - \boldsymbol{q}) + \mathbf{J}^T(\mathbf{I} - \mathbf{S})\mathbf{K}_f(\mathbf{f}_d - \mathbf{f}_e) \quad (\text{II.41})$$

where dynamic computation  $\boldsymbol{\tau}_{ff}$  is defined in (II.39) as,

$$\boldsymbol{\tau}_{ff}(\boldsymbol{q}_d, \dot{\boldsymbol{q}}_d, \ddot{\boldsymbol{q}}_d) = \mathbf{B}(\boldsymbol{q}_d)\ddot{\boldsymbol{q}}_d + \mathbf{C}(\boldsymbol{q}_d, \dot{\boldsymbol{q}}_d)\dot{\boldsymbol{q}}_d + \mathbf{F}_d\dot{\boldsymbol{q}}_d + \mathbf{g}(\boldsymbol{q}_d) \quad (\text{II.42})$$

In (II.50) a force error  $\mathbf{f}_d - \mathbf{f}_e$  is multiplied first by a force gain  $\mathbf{K}_f$  and then by a selection matrix  $\mathbf{I} - \mathbf{S}$  to ensure the correct partitioning between position variables and force variables. The transformation to joint torques is then accomplished with the transpose Jacobian matrix  $\mathbf{J}^T$ . Finally, the outputs from the position ad force controllers are summed to produce the joint torques.

Thus the dynamic model is applied to the position control loop (analogous to computed torque control or resolved acceleration position control) and the force control loop is added separately. Resolved acceleration control is virtually identical to other Cartesian-based force controllers, such as impedance control and the operational space method.

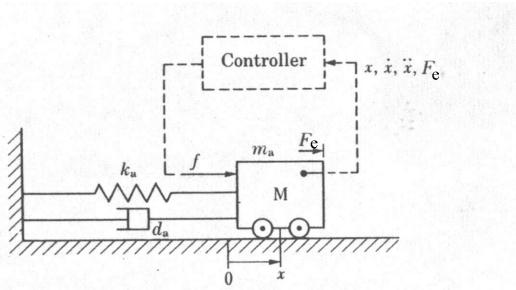
#### II.1.9.4 Stiffness Control

Stiffness control [35] derives from a position control scheme of PD type with gravity compensation. The driving torques are chosen as

$$\boldsymbol{\tau}_c = \mathbf{J}^T(\boldsymbol{q})(\mathbf{K}_p(\mathbf{K}_f(\mathbf{f}_d - \mathbf{f}_e) - \mathbf{r}) + \mathbf{f}_d) - \mathbf{K}_v\dot{\boldsymbol{q}} + \mathbf{g}(\boldsymbol{q}) \quad (\text{II.43})$$

where  $\mathbf{K}_p$  is the gain of active stiffness on the end-effector position error, and  $\mathbf{K}_v$  is the gain of a joint damping action. The purpose of this control is to make the end-effector compliant with respect to contact forces by acting on  $\mathbf{K}_p$ . For such a reason, this strategy is also referred in the literature as (active) compliance control, and, since damping is controlled besides stiffness, the control law can be regarded as impedance control with static model-based compensation. Notice that no force measurement is required. Also, it is possible to convert this controller to a model-based controller compensating the model dynamics.

### II.1.9.5 Impedance Method



**Figure II.11 Impedance control of one degree of freedom**

A simple one degree of freedom system will serve to illustrate the impedance method proposed by Neville Hogan [36-38]. We assume that the dynamic equation of the mechanical system shown in Figure II.11 is given by

$$m_a \ddot{x} + b_a \dot{x} + k_a x = f - f_e \quad (\text{II.44})$$

where \$m\_a\$ is the mass of the body, \$f\_e\$ is the external force, \$f\$ is the driving force which we can apply, \$x\$, \$\dot{x}\$ and \$\ddot{x}\$ are the displacement, velocity and the acceleration of the object respectively, \$k\_a\$ is the spring constant and \$b\_a\$ is the damping coefficient. We also define a desired impedance of the body to the external force as follows:

$$m_d \ddot{x} + b_d \dot{x} + k_d x = -f_e \quad (\text{II.45})$$

where \$m\_d\$, \$b\_d\$ and \$k\_d\$ are the desired mass, damping coefficient and spring constant, and \$x\_d\$, \$\dot{x}\_d\$ is the desired position and velocity trajectory, respectively.

When \$x\$, \$\dot{x}\$ and \$\ddot{x}\$ are measurable, we can use the control law

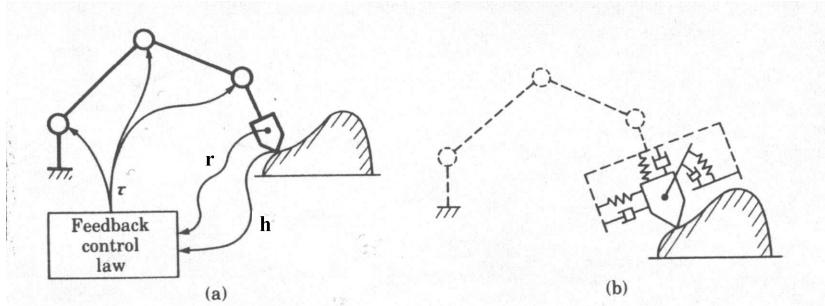
$$f = (m_a - m_d) \ddot{x} + (b_a - b_d) \dot{x} + (k_a - k_d) x + b_d \dot{x}_d + k_d x_d \quad (\text{II.46})$$

substituting equation (II.46) into equation (II.44) yields equation (II.45) showing that the closed-loop system has the desired impedance. When the external force is measurable, the control law represented by II.54 can be replaced by

$$\begin{aligned} f = & (b_a - m_a m_d^{-1} b_d) \dot{x} + (k_a - m_a m_d^{-1} k_d) x + (1 - m_a m_d^{-1}) f_e \\ & + m_a m_d^{-1} (b_d \dot{x}_d + k_d x_d) \end{aligned} \quad (\text{II.47})$$

and if it is allowable to have the original mass  $m_a$  as the desired mass  $m_d$ , then the equations (II.46) and (II.47) reduce to a simple position and velocity feedback law:

$$f = (b_a - b_d) \dot{x} + (k_a - k_d)x + b_d \ddot{x}_d + k_d \dot{x}_d \quad (\text{II.48})$$



**Figure II.12 Impedance method (a) Feedback control system. (b) Mechanical impedance realized by (a)**

Now, the approach introduced above can be easily generalized to manipulators with multiple degrees of freedom. As shown in figure II.12a measurements of the manipulation vector  $\mathbf{r}$  and the external force and moment  $\mathbf{h}$  acting on the end-effector are used to drive the actuators at the joints through a feedback control law. By selecting the feedback law properly, we wish to develop a system that behaves like an end-effector with a desired mechanical impedance supported by an ideal arm that perfectly follows the desired trajectory without being effected by any external force. This system is shown pictorially in figure II.12b.

Consider a six link manipulator, and assume that the desired mechanical impedance for its end-effector is described by

$$\mathbf{M}_d \ddot{\mathbf{r}}_d + \mathbf{B}_d \dot{\mathbf{r}}_d + \mathbf{K}_d \mathbf{r}_d = -\mathbf{h} \quad (\text{II.49})$$

where  $\mathbf{r}_d$  is the difference between the current value of an appropriate six dimensional manipulation vector  $\mathbf{r}$  and its desired value  $\mathbf{r}_d$ :

$$\mathbf{r}_e = \mathbf{r} - \mathbf{r}_d \quad (\text{II.50})$$

In equation (II.49)  $\mathbf{h}$  represents the six dimensional external force and moment exerted on the end-effector by its environment. The  $(6 \times 6)$  matrices  $\mathbf{M}_d$ ,  $\mathbf{B}_d$  and  $\mathbf{K}_d$  are, respectively, the inertia matrix, the damping coefficient matrix and the stiffness

matrix, all which are symmetric and nonnegative definite. A simple way of choosing  $\mathbf{M}_d$ ,  $\mathbf{B}_d$  and  $\mathbf{K}_d$  is to make them diagonal and take into account the directions in which large impedance is desirable and the directions in which small impedance is desirable.

The next step is to develop a control law that achieves the desired impedance (II.49). The general dynamic equation of the manipulator including the force and moment exerted by its environment is

$$\mathbf{B}(q)\ddot{\mathbf{q}} + \mathbf{C}(q, \dot{q})\dot{\mathbf{q}} + \mathbf{F}_d(q) + \mathbf{g}(q) = \boldsymbol{\tau}_c - \mathbf{J}^T(q)\mathbf{h} \quad (\text{II.51})$$

which can further be simplified to

$$\mathbf{B}(q)\ddot{\mathbf{q}} + \mathbf{N}(q, \dot{q}) = \boldsymbol{\tau}_c - \mathbf{J}^T(q)\mathbf{h} \quad (\text{II.52})$$

where

$$\mathbf{N}(q, \dot{q}) = \mathbf{C}(q, \dot{q})\dot{\mathbf{q}} + \mathbf{F}_d(q) + \mathbf{g}(q) \quad (\text{II.53})$$

and from direct kinematics we know that the relation between the task space variable  $\mathbf{q}$  and the joint space variable  $\mathbf{r}$  is

$$\dot{\mathbf{r}} = \mathbf{J}(q)\dot{\mathbf{q}} \quad (\text{II.54})$$

Differentiating equation (II.54), and solving for  $\ddot{\mathbf{q}}$  we get

$$\ddot{\mathbf{q}} = \mathbf{J}^{-1}(q)(\ddot{\mathbf{r}} - \dot{\mathbf{J}}(q)\dot{\mathbf{q}}) \quad (\text{II.55})$$

Substituting  $\ddot{\mathbf{q}}$  and  $\dot{\mathbf{q}}$  from equations (II.54) and (II.55) into dynamic equation of the manipulator (II.52), and after some modification, we get the equation with respect to task space:

$$\mathbf{M}(q)\ddot{\mathbf{r}} + \mathbf{N}_r(q, \dot{q}) = \mathbf{J}^{-T}(q)\boldsymbol{\tau}_c - \mathbf{h} \quad (\text{II.56})$$

where

$$\mathbf{M}(q) = \mathbf{J}^{-T}(q)\mathbf{B}(q)\mathbf{J}^{-1}(q) \quad (\text{II.57})$$

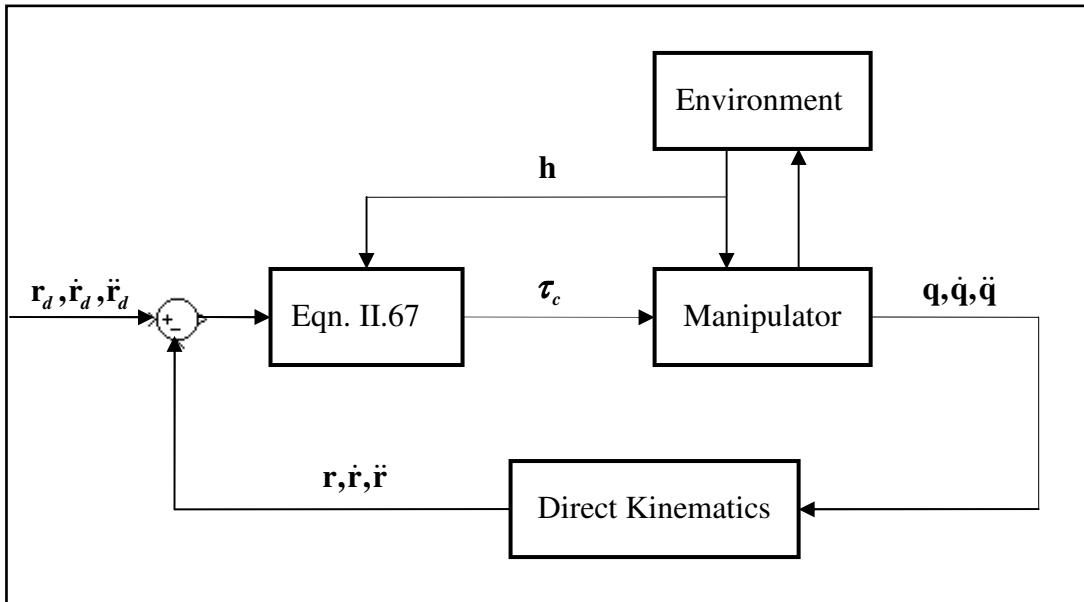
$$\mathbf{N}_r = \mathbf{J}^T(q)\mathbf{N}(q, \dot{q}) - \mathbf{M}(q)\dot{\mathbf{J}}(q)\dot{q} \quad (\text{II.58})$$

Now we apply a nonlinear feedback law of the form

$$\begin{aligned} \boldsymbol{\tau}_c = & \mathbf{J}^T(q)\{\mathbf{N}_r(q, \dot{q}) - \mathbf{M}(q)\mathbf{M}_d^{-1}(\mathbf{D}_d \dot{\mathbf{r}}_e + \mathbf{K}_d \mathbf{r}_e) \\ & + [\mathbf{I} - \mathbf{M}(q)\mathbf{M}_d^{-1}]\mathbf{h}\} \end{aligned} \quad (\text{II.59})$$

to the dynamic equation (II.56) which shows the closed loop system becomes equal to equation (II.49).

The block diagram of the control strategy can be seen in Figure II.13.



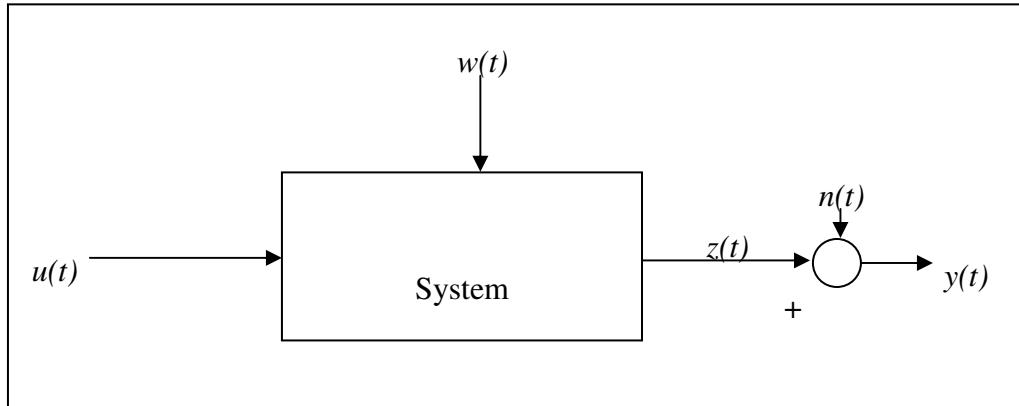
**Figure II.13 Block diagram of impedance control**

Many investigations have been made about impedance control method by researchers. [39] Presented a quaternion-based impedance control framework, which ensures geometric consistency for the execution of six degree of freedom interaction tasks. [40] Proposed and realized a method, not only to regulate the interaction between manipulator generalized force and position, but additionally control the generalized contact force and position. [41] Discussed the essential equivalence of the second-order impedance control with force feedback and proportional gain explicit force control with force feedback. [42] Proposed force-impedance controller, which enables two kind of behavior: force limited impedance control and position limited force control. [43] Presented and realized a new approach to six-dof

impedance control, which leads to a physically meaningful definition of the rotational part of the impedance equation. [44] characterized the mechanical impedance by a translational part and a rotational part (nondiagonal stiffness). Robust implementations of impedance control against modelling errors and disturbances are proposed and realized in [45]. A learning impedance control problem for robotic manipulators is formulated, solved and simulated in [46]. [47] Proposed and simulate a new method for implementing impedance control, which is designed to take the advantage of the error-correction capabilities of position controllers, while maintaining good impedance tracking performance. [48] Presented and realized a new approach of transporting a flexible beam handled by two manipulators to a desired position/orientation while suppressing its vibration, and simultaneously controlling the internal forces between the manipulators and the beam to avoid any damage on the system. In [57], the control of a 2-dof Cartesian manipulator interacting a curved surface with varying stiffness is simulated. Then, the impedance control algorithms are extended to redundant manipulators in [49,50].

## II.1.10 IDENTIFICATION METHODS

In system identification, we are concerned with the determination of system models from records of system operation. The problem can be represented diagrammatically as shown in Figure II.14.



**Figure II.14 Block Diagram of an unknown system**

Where

$u(t)$  is the known input vector of dimension m

$z(t)$  is the output vector of dimension p

$w(t)$  is the input disturbance vector

$n(t)$  is the observation noise vector

$y(t)$  is the measured output vector of dimension p

Thus, the problem of system identification is the determination of the system model from records of  $u(t)$  and  $y(t)$ .

At this point it is important to distinguish between *system* and *its model*. A *system* is defined as “a collection of objects arranged in an ordered form, which is, in some sense, purpose or goal directed.” What constitutes a ‘system’ depends upon the viewpoint of the analyst or designer. For instance, an electronic amplifier consisting of a large number of components may be regarded as a system by the electronic engineer. On the other hand, the same amplifier may be one of the main parts of a “feedback control system.” Furthermore, this feedback control system may be a part of a chemical process (or system) containing many loops of this type. Finally, we

may have a plant containing many such units.

A *model* may be defined as “a representation of the essential aspects of a system, which presents knowledge of that system in a useable form. A model, in order to be useful, must not be so complicated that it cannot be understood and thereby be unsuitable for predicting the behaviour of the system. At the same time it must not be trivial to the extent that predictions of the behaviour of the system based on the model are grossly inaccurate.

A fundamental problem in system identification is the choice of the nature of the model, which should be used for the system. The model may be one of the following types:

- linear time-invariant (lumped parameter) – ordinary linear differential equations
- linear time-variant (lumped parameter) – ordinary linear differential equations
- linear but with distributed parameters – partial differential equations
- nonlinear-nonlinear differential equations

Although, in practice, most systems are nonlinear with distributed parameters, linear models for such systems are often used because of their simplicity. In a large number of cases, “incremental,” or “piecewise” linear models can be conveniently used for approximate understanding of the system. In using such models, one must be careful and should have an idea of limits of their validity. Nevertheless, a great deal of work has been done on obtaining linear models for systems; so much that often system identification one understands the determination of the parameters of “suitable” linear model for the system.

- Determining the order of the linear model
- Selection of a suitable criterion for determining the “accuracy” of the model
- Designing an input signal, which will maximize the accuracy of the estimates of the parameters of the model.

Although most systems are of the “continuous-time” type, the application of the digital computer for identification makes it desirable to use “discrete-time” model

is more straightforward. Furthermore, provided that the sampling interval satisfies certain conditions, the determination of the continuous-time model from the discrete-time model is fairly straightforward.

Many applications require “*on-line*” identification instead of “*off-line*”. An identification method is said to be of the “*off-line*” type when collects a large amount of input and output data for system which may be stored in a computer or recorded in some manner. These data are then processed in a batch to estimate the parameters of the model and obtain the best fit according to a prescribed cost function. In off-line identification, one may often select the type of input most suitable. Also, there is a greater flexibility in selecting computational methods without any restriction on computing time. As a result the accuracy of the estimates can be made fairly high.

In a number of control applications, especially adaptive control, it is necessary to identify the system in a fairly short time. An identification scheme is said to be of the “*offline*” type if it satisfies the following conditions:

- it does not require a special input
- all the data need to be stored
- a recursive algorithm is used for adjusting the estimates of the parameter after each sampling instant
- the amount of computation required for “model adjustment” is a fraction of the sampling period

It may be added that, in general, on-line methods will not lead to as accurate models as possible with off-line methods, which can use a much larger amount of data. But in many practical situations one cannot afford to wait for the time required to collect all the data. As a matter of fact, it will be recognized that life is the art of reaching sufficient conclusions from insufficient data. Some typical examples of situations where one must make an important decision on the basis of insufficient information are: (i) getting married, (ii) accepting a job, (iii) hiring a new employee, and (iv) investing in the stock market.

A large variety of the methods have been applied to system identification,

both off-line and on-line. The methods can be classified in many ways; one scheme for classification is given below.

1. Classical Methods: (mostly off-line)

- Frequency Response Identification
- Impulse response identification
- Step response identification
- Identification from correlation functions

2. Equation-Error Approach: (batch-processing)

- Least-squares
- Generalised least squares
- Maximum likelihood
- Minimum variance
- Gradient Methods

3. Model Adjustment Techniques:

- Least-squares (recursive)
- Generalised least squares (recursive)
- Instrumental variables
- Bootstrap
- Maximum likelihood (recursive)
- Correlation (recursive)
- Stochastic approximation

# **PART III**

## **III THESIS STUDIES**

In this part of the report, the thesis studies will be examined. Experimental setup, kinematics, dynamics, interaction control and identification of Three-Axis Cartesian Robot Arm are the titles that will be introduced consequently.

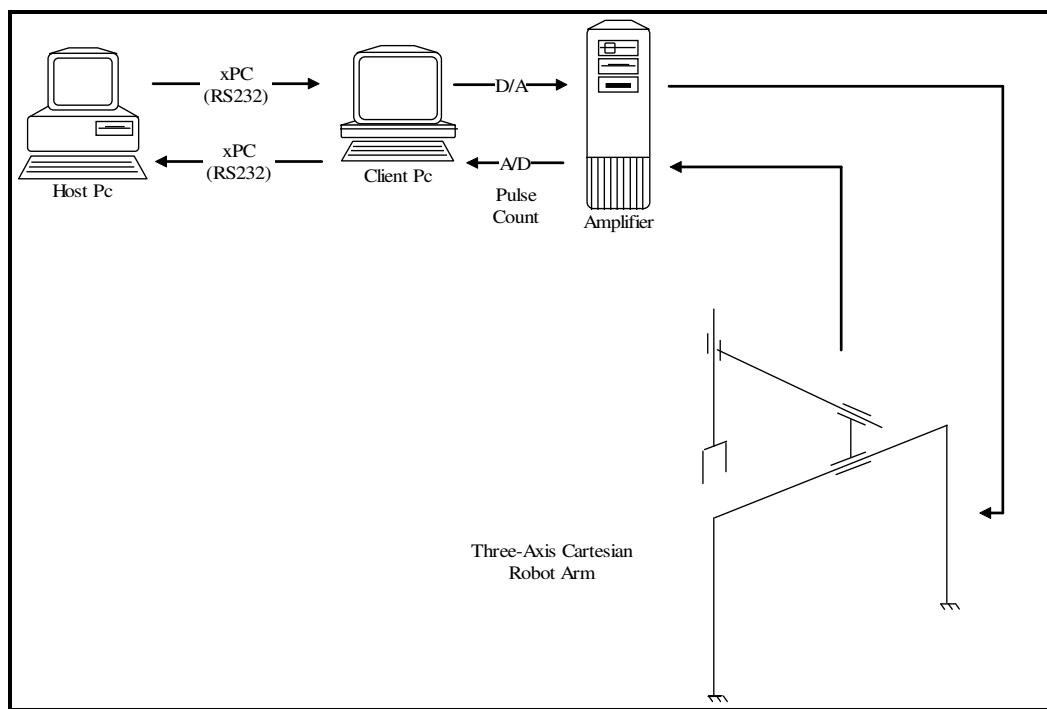
### **III.1 THREE AXIS CARTESIAN ROBOT ARM**



**Figure III.1      Three-Axis Cartesian Robot Arm**

Figure III.1 shows the Three-Axis Cartesian Robot Arm. The thesis study concerns the dynamic parameter identification and the interaction control of the Three-Axis Cartesian Robot Arm.

The thesis study is based on two concepts “*identification*” and “*interaction control*”. Also “*position control*” techniques are used. This study will be real application with simulation support, and help us to understand the dynamics of manipulators interacting with the environment.



**Figure III.2      Data Flow of a Three-Axis Cartesian Robot Arm**

## **III.2 EXPERIMENTAL SETUP**

### **III.2.1 Baldor BSM 63A-275 Three Phase Brushless Servo Motor**

The A-Series of Brushless servomotor [60] provides a very responsive acceleration in a durable design using Samarium Cobalt magnetics. This series provides continuous stall torque capability ranging from 6.8 lb-in (0.77 N-m) to 366 lb-in (41.3 N-m). Peak torques are typically 4 times continuous. For Further information, product specifications and performance graph of the servomotor, see Appendix A-1.

### **III.2.2 Baldor BSC 1100 AC Three Phase Brushless Servo Motor Controller**

BSC 1100 [60] is electrical equipment for industrial power installations. It is designed for machine applications, which need variable speed controlled three-phase A.C. motors.

The BSC 1100 is a resolver based servo control designed to be used with brushless servomotors. The control accepts a standard  $\pm 10$  VDC to control Brushless servomotors in either a velocity or current (torque) loop. The experiments has been done in current loop as all the control algorithms are based on torque input. Hence, the amplification constants of the servo motor controllers are identified, which will be explained later in section III.7.

### **III.2.3 PLS 70 Servo-Gear-Head**

Gears are the most essential parts of a precision gearbox. All gears in the PLS Series Servo-Gearboxes [61] use straight tooth geometry which eliminates the generation of unbalanced axial forces. Helical tooth geometry creates axial forces. As a result, the planetary systems using straight teeth is inherently balanced. Despite of the straight teeth, a low noise level is achieved by maintaining a very high gear

precision and utilising gear honing as the final machining step. Honing produces an exceptional high surface finish, which is not only beneficial for the noise reduction but for better lubrication and it eliminates the increase of backlash due to "wear in".

For further information and product specifications see Appendix A-2.

## **III.2.4 Axis**

### **III.2.4.1 FESTO DGE 25-ZR Toothed (Timing) Belt Drive**

In the three-axis Cartesian robot arm,  $z$ -axis is FESTO DGE 25-ZR [62] type. This type of axis has toothed belts inside and drive shafts are rotating the belt. The driver is connected to the belt. The specifications and the further information about DGE 25-ZR is in the Appendix A-3.

### **III.2.4.2 FESTO DGE 40-SR Spindle Drive**

In the three axis Cartesian robot arm,  $x$  and  $y$  axes are FESTO DGE 40-SR [62] type. This type of axis has spindle shaft and drive shaft. The driver is connected to the spindle shaft. The specifications and the further information about DGE 40-SP is in the Appendix A-4.

## **III.2.5 Force Sensors**

The force sensors are one axis  $+/-50$  Newton PASCO [63] scientific model CI-6537. It has an output between  $-8$  Volts and  $+8$  Volts and a range between  $-50$  Newton and  $+50$  Newton. A push is considered to be positive, and a pull is considered to be negative. The sensor has strain gauges mounted on a specially designed "binocular beam" which deflects less than 1 millimeter, and has built-in over-limit protection against forces greater than  $+/-50$  Newton.

The sensor has only the ability to measure the forces in one direction, so, it is attached to the end-effector of the manipulator to measure the forces in  $y$  direction.

## **III.2.6 Motion sensor**

The PASCO [63] scientific model CI-6529 motion sensor is a sonar ranging

device with 15°. With the proper interface (PASCO *Science Workshop* program) it can emit ultrasonic pulses and measure the time it takes the pulses to return a echos from the target. It has a maximum range of 10 meters.

### **III.2.7 Data Acquisition Cards**

#### **III.2.7.1 Advantech PCL 812PG MultiLab Analog and Digital I/O Card**

The PCL-812PG [64] is a multifunction analog and digital I/O card that features the five most desired measurement and control functions for PC/AT and compatible systems: A/D conversion, D/A conversion, digital input, digital output and counter/timer. This half-size card neatly packages 16 12-bit analog input channels, two 12-bit analog output channels, 16 digital input channels, 16 digital output channels and a programmable counter/timer.

The card is supported by Matlab to be used in Simulink and real time applications by xPC.

More information and product specifications are in Appendix A-5.

#### **III.2.7.2 Advantech PCI-1720 4-channel Isolated D/A Output Card**

The PCI-1720 [64] provides four 12-bit isolated digital-to-analog outputs for the PCI bus. With isolation protection of 2500 VDC between the outputs and the PCI bus, the PCI-1720 is ideal for industrial applications where high-voltage protection is required.

##### **Applications**

- Process control
- Programmable voltage source
- Programmable current sink
- Servo control

More information and product specifications are in Appendix A-6. The card is not supported by Matlab to be used in Simulink and real time applications by xPC.

So the driver code of the card is written by C language with the Simulink S-Function application programming interface (Appendix B-3).

### **III.2.7.3 Advantech PCL-833 3-axis Quadrature Encoder and Counter Card**

The PCL-833 [64] is a three-axis quadrature encoder and counter add-on card for the IBM PC/AT and compatible (ISA bus). This card lets your PC perform position monitoring for motion control systems.

#### **Encoder Interface**

Each input includes a decoding circuit for incremental quadrature encoding. Inputs accept either single-ended or differential signals. Quadrature input works with or without an index, allowing linear or rotary encoder feedback.

#### **Counters**

The PCL-833 has three independent 24-bit counters. The maximum quadrature input rate is 1.0 MHz, and the maximum input rate in counter mode is 2.4 MHz. You can individually configure each counter for quadrature decoding, pulse/direction counting or up/ down counting.

More information and product specifications are in Appendix A-7. The card is not supported by Matlab to be used in Simulink and real time applications by xPC. So the driver code of the card is written by C language with the Simulink S-Function application programming interface (Appendix B-2).

## **III.2.8 Client/Host PC's**

Client and Host personal computers (PC) are desktop computers and data acquisition cards are attached to the client PC. Host is connected to the client by RS232 connection. The real time code that is to be executed is downloaded to the client PC through this connection after building with a C compiler. The target PC does not have a hard disk, because the build program, downloaded from the host is directly copied to the ram. The target PC's processor is Pentium 133 Mhz, and it has

32 Mb memory. The host PC's processor is Pentium 200 Mhz, and it is build with 96 Mb memory.

### **III.2.9 Software**

#### **III.2.9.1 Matlab**

MATLAB integrates mathematical computing, visualisation, and a powerful language to provide a flexible environment for technical computing. The open architecture makes it easy to use MATLAB and its companion products to explore data, create algorithms, and create custom tools that provide early insights and competitive advantages.

MATLAB includes tools for:

- Data acquisition
- Data analysis and exploration
- Visualisation and image processing
- Algorithm prototyping and development
- Modelling and simulation
- Programming and application development

Additional information and MATLAB features can be found at [59].

The 3-axis Cartesian manipulator is modeled in Simulink (Appendix B-9) in order to simplify the block diagrams.

#### **III.2.9.2 xPC Target**

xPC Target is a solution for prototyping, testing, and deploying real-time systems using standard PC hardware. It is an environment that uses a target PC, separate from the host PC, for running real-time applications.

In this environment you use the desktop computer as a host PC with MATLAB, Simulink, C Compiler and Stateflow (optional) to create model using Simulink blocks and Stateflow diagrams. After creating your model, you can build

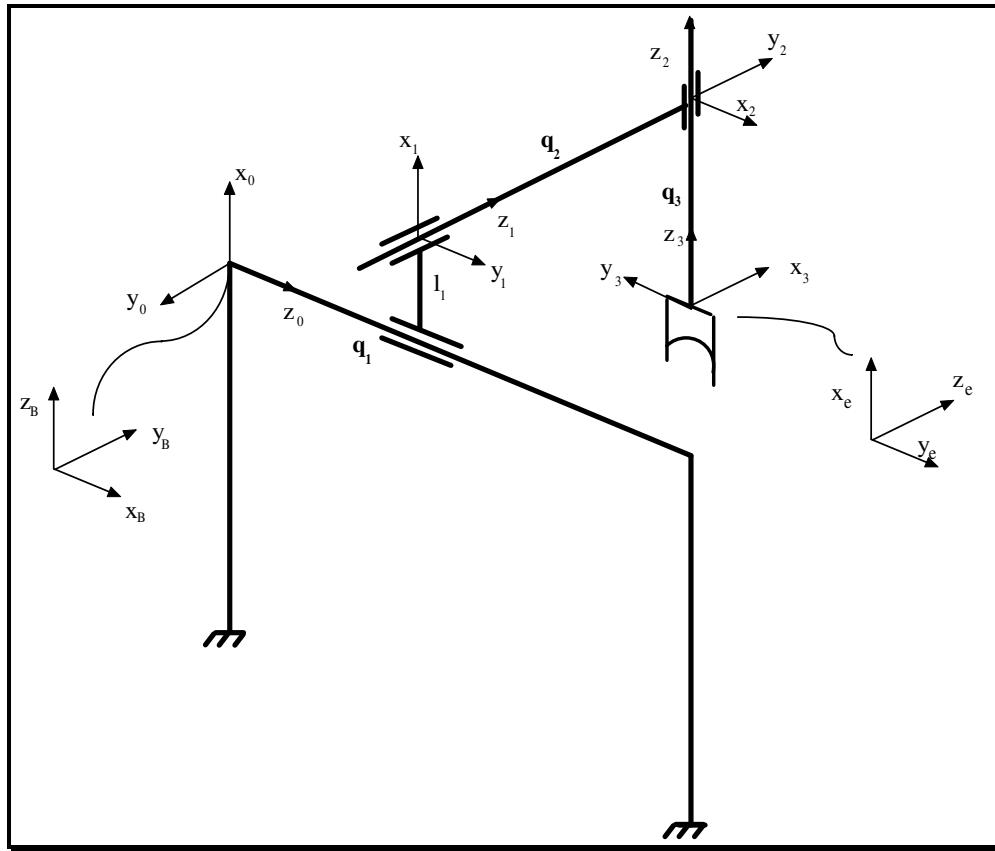
the model and download to the target PC in order to run simulations in real-time. xPC Target allows you to add I/O blocks (data acquisition card blocks) to your model.

The executable code is downloaded from the host PC to the target PC running the xPC Target real-time kernel with a serial connection (RS232) or a Network connection (TCP/IP). After downloading the executable code, you can run and test your target application in real-time.

- **Special hardware requirements** — The xPC Target software requires a host PC, target PC, and for I/O, the target PC must also have I/O boards supported by xPC Target. However, the target PC can be a desktop PC, Industrial PC, PC/104, PC/104+, or Compact PCI computer.
- **Special software requirements** — The xPC Target software requires either a Microsoft Visual C/C++ compiler (version 5.0 or 6.0) or a Watcom C/C++ compiler (version 10.6 or 11.0). In addition, xPC Target requires MATLAB, Simulink, and Real-Time Workshop.

More information and features of xPC Target can be found at [59].

### III.3 KINEMATICS OF THE THREE-AXIS CARTESIAN ROBOT ARM



**Figure III.3      Kinematic Model of the Three Axis Cartesian Robot Arm**

Denavit-Hartenberg notation matrix for the Three-Axis Cartesian Robot Arm is

**Table III.1 Denavit-Hartenberg notation parameters**

$K$	$\alpha$	$Q$	$d$	$\theta$
1	$\pi/2$	$l_1$	$q_1$	0
2	$\pi/2$	0	$q_2$	$\pi/2$
3	0	$l_e$	$-q_3$	$\pi/2$

The general form of the transformation matrix of the link  $k$  to the link  $k-1$  is

$${}_{k-1}\mathbf{T}_k = \begin{bmatrix} \cos(\theta_k) & -\cos(\alpha_k)\cos(\theta_k) & \sin(\alpha_k)\sin(\theta_k) & a_k \cos(\theta_k) \\ \sin(\theta_k) & \cos(\alpha_k)\cos(\theta_k) & -\sin(\alpha_k)\cos(\theta_k) & a_k \sin(\theta_k) \\ 0 & \sin(\alpha_k) & \cos(\alpha_k) & a_k \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{III.1})$$

The following matrices are the transformation matrices (Equation II.5) for the three-axis cartesian robot arm.

$${}^b\mathbf{T}_0 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{III.2})$$

$${}^0\mathbf{T}_1 = \begin{bmatrix} 1 & 0 & 1 & l_1 \\ 0 & 0 & -1 & 0 \\ 0 & q_1 & 0 & q_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{III.3})$$

$${}^1\mathbf{T}_2 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & q_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{III.4})$$

$${}^2\mathbf{T}_3 = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -q_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{III.5})$$

$${}^3\mathbf{T}_e = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{III.6})$$

The resulting transformation matrix (Equation II.6) between the end-effector and base coordinate is as follows;

$${}^b\mathbf{T}_e = \begin{bmatrix} \mathbf{0} & \mathbf{1} & \mathbf{0} & q_1 \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & q_2 \\ \mathbf{1} & \mathbf{0} & \mathbf{0} & -q_3 + l_1 \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & 1 \end{bmatrix} \quad (\text{III.7})$$

The upper-left (3x3) part of this matrix represents the orientation of the end-effector with respect to the base frame, which is constant because of the structure of the Cartesian manipulators, and the upper-right (3x1) part represents the position vector  $\mathbf{r}=f(\mathbf{q})$  of the end-effector with respect to the base frame. Hence, the Jacobian matrix can be found with the kinematic relation between the joint space variables and the task space variables by

$$\mathbf{J}_{kj} = \frac{\partial \mathbf{r}_k(\mathbf{q})}{\partial q_j} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \quad (\text{III.8})$$

which is configuration dependent because of the structure of the Cartesian manipulators.

With the reduction ratio and feed constant of the driver shaft, the relation between the joint variable and the servo motor variable is

$$\mathbf{q} = \mathbf{K}\boldsymbol{\theta}_m \quad (\text{III.9})$$

where  $\boldsymbol{\theta}_m$  is the (3x3) motor variable vector in radians. And  $\mathbf{K}$  is defines as

$$\mathbf{K} = \frac{\mathbf{K}_f \mathbf{K}_r}{2\pi} \quad (\text{III.10})$$

where (3x3) matrix  $\mathbf{K}_f$  is the feed constant of the axis, and (3x3) matrix  $\mathbf{K}_r$  is the reduction ratio of the servo-gear-head.

$$\mathbf{K}_f = \begin{bmatrix} \mathbf{0.02} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0.02} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0.063} \end{bmatrix} \text{ m/rev} \quad \mathbf{K}_r = \begin{bmatrix} 1/4 & 0 & 0 \\ 0 & 1/4 & 0 \\ 0 & 0 & 1/4 \end{bmatrix}$$

The current of servomotor is obtained from amplifier as voltage between 0~10 Volts. With the torque constant of the servo motors the relation between the voltage and the motor torque is

$$\tau_m = \mathbf{K}_t \mathbf{V} \quad (\text{III.11})$$

where  $\mathbf{K}_t$  is the (3x3) matrix of torque constant,  $\tau_m$  is the (3x1) vector of servomotor torques, and  $\mathbf{V}$  is the (3x1) vector of voltages read from servomotor amplifiers.

$$\tau = \begin{bmatrix} \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} \quad \mathbf{K}_t = \begin{bmatrix} 0.31 & 0 & 0 \\ 0 & 0.31 & 0 \\ 0 & 0 & 0.31 \end{bmatrix} \quad \mathbf{V} = \begin{bmatrix} V_x \\ V_y \\ V_z \end{bmatrix}$$

From Virtual Work principle,

$$\tau_m^T d\theta_m = \mathbf{f}_c^T dq \quad (\text{III.12})$$

and from derivation of equation (III.9) we get

$$d\mathbf{q} = \frac{\mathbf{K}_f \mathbf{K}_r}{2\pi} d\theta_m \quad (\text{III.13})$$

If we substitute (III.13) into (III.12), we obtain the relation between the servomotor torques and the forces generated in joint space as

$$\tau_m = \frac{\mathbf{K}_f \mathbf{K}_r}{2\pi} \mathbf{f}_c^T \quad (\text{III.14})$$

where  $\mathbf{f}_c$  represents the joint driving (control) forces, as no torque is present because of the structure of the Cartesian robot arm.

## III.4 DYNAMICS OF THE THREE-AXIS CARTESIAN ROBOT ARM

The general dynamic model for a three axis manipulator is in the form given in the Equation II.19

$$\mathbf{B}(q)\ddot{\mathbf{q}} + \mathbf{C}(q, \dot{q})\dot{\mathbf{q}} + \mathbf{F}_d\dot{\mathbf{q}} + \mathbf{g}(q) = \boldsymbol{\tau}_c - \mathbf{J}^T(q)\mathbf{h}$$

The frictional force can be added to this dynamic model as follows;

$$\mathbf{B}(q)\ddot{\mathbf{q}} + \mathbf{C}(q, \dot{q})\dot{\mathbf{q}} + \mathbf{F}_d\dot{\mathbf{q}} + \mathbf{g}(q) + \mathbf{F}_s \operatorname{sgn}(\dot{\mathbf{q}}) = \boldsymbol{\tau}_c - \mathbf{J}^T(q)\mathbf{h} \quad (\text{III.15})$$

Where  $\mathbf{B}(q)$  is the (3x3) inertia matrix of the robot manipulator which is symmetric and positive definite,  $\mathbf{C}(q, \dot{q})$  is the (3x3) vector of Coriolis and centrifugal torques,  $\mathbf{F}_d$  is the (3x3) matrix of damping effects,  $\mathbf{g}(q)$  is the (3x3) vector of gravity torques,  $\mathbf{F}_s$  (3x3) vector of static friction forces,  $\boldsymbol{\tau}_c$  is the (3x1) vector of joints torques,  $\mathbf{h}$  is the (3x1) vector of end-effector force and moments,  $\mathbf{J}$  is the (3x3) Jacobian matrix which transforms end-effector force and moments to joint space and  $\mathbf{q}$  is the (3x1) joint variables.

This dynamic model is simplified for the three-axis Cartesian robot arm kinematically modeled in Figure III.3 in joint space is:

$$\mathbf{B}\ddot{\mathbf{q}} + \mathbf{F}_d\dot{\mathbf{q}} + \mathbf{F}_s \operatorname{sgn}(\dot{\mathbf{q}}) + \mathbf{B}\mathbf{g} = \mathbf{f}_c - \mathbf{J}^T\mathbf{f}_e \quad (\text{III.16})$$

Here (3x3) diagonal mass matrix  $\mathbf{B}$  and (3x3) diagonal matrix  $\mathbf{J}$  are independent of the joint variables  $\mathbf{q}$  because of the structure of the Cartesian robot arm. For the same reason no Coriolis, centrifugal, joint and external torques exists in the system. Thus joint torque vector  $\boldsymbol{\tau}$  and external force and torque vector  $\mathbf{h}$  are represented as  $\mathbf{f}_c$  and  $\mathbf{f}_e$  respectively.

This dynamic model can be converted to the task space easily with the help of  $\dot{\mathbf{r}} = \mathbf{J}\dot{\mathbf{q}}$  and  $\ddot{\mathbf{r}} = \mathbf{J}\ddot{\mathbf{q}}$  kinematic relations as

$$\mathbf{M}\ddot{\mathbf{r}} + \mathbf{D}\dot{\mathbf{r}} + \mathbf{F}_{sr} \operatorname{sgn}(\dot{\mathbf{r}}) + \mathbf{M}\mathbf{J}\mathbf{g} = \mathbf{f} - \mathbf{f}_e \quad (\text{III.17})$$

where

$$\mathbf{M} = \mathbf{J}^{-T} \mathbf{B} \mathbf{J}^{-1} \quad (\text{III.18})$$

$$\mathbf{D} = \mathbf{J}^{-T} \mathbf{F}_d \mathbf{J}^{-1} \quad (\text{III.19})$$

$$\mathbf{F}_{sr} = \mathbf{J}^{-T} \mathbf{F}_s \mathbf{J}^{-1} \quad (\text{III.20})$$

$$\mathbf{f} = \mathbf{J}^{-T} \mathbf{f}_c \quad (\text{III.21})$$

$\mathbf{M}$ ,  $\mathbf{D}$  and  $\mathbf{F}_{sr}$  are (3x3) matrix

$$\mathbf{M} = \begin{bmatrix} m_x & 0 & 0 \\ 0 & m_y & 0 \\ 0 & 0 & m_z \end{bmatrix} kg \quad (\text{III.22})$$

$$\mathbf{D} = \begin{bmatrix} d_x & 0 & 0 \\ 0 & d_y & 0 \\ 0 & 0 & d_z \end{bmatrix} \frac{Ns}{m} \quad (\text{III.23})$$

$$\mathbf{F}_{sr} = \begin{bmatrix} f_{sx} & 0 & 0 \\ 0 & f_{sy} & 0 \\ 0 & 0 & f_{sz} \end{bmatrix} N \quad (\text{III.24})$$

$\ddot{\mathbf{r}}$ ,  $\dot{\mathbf{r}}$  and  $\text{sgn}(\dot{\mathbf{r}})$  are (3x1) vectors

$$\ddot{\mathbf{r}} = \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} \frac{m}{sn^2} \quad \dot{\mathbf{r}} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} \frac{m}{sn} \quad \text{sgn}(\dot{\mathbf{r}}) = \begin{bmatrix} \text{sgn}(\dot{x}) \\ \text{sgn}(\dot{y}) \\ \text{sgn}(\dot{z}) \end{bmatrix}$$

$\mathbf{f}$ ,  $\mathbf{f}_e$  and  $\mathbf{g}$  are (3x1) vectors

$$\mathbf{f} = \begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix} N \quad \mathbf{f}_e = \begin{bmatrix} f_{ex} \\ f_{ey} \\ f_{ez} \end{bmatrix} N \quad \mathbf{g} = \begin{bmatrix} 0 \\ 0 \\ -9.81 \end{bmatrix} \frac{m}{sn^2}$$

Then the equation of motions for each axis of the 3-axis Cartesian robot arm are

$$m_x \ddot{x} + d_x \dot{x} + f_{sx} \text{sgn}(x) = f_x - f_{ex} \quad (\text{III.25})$$

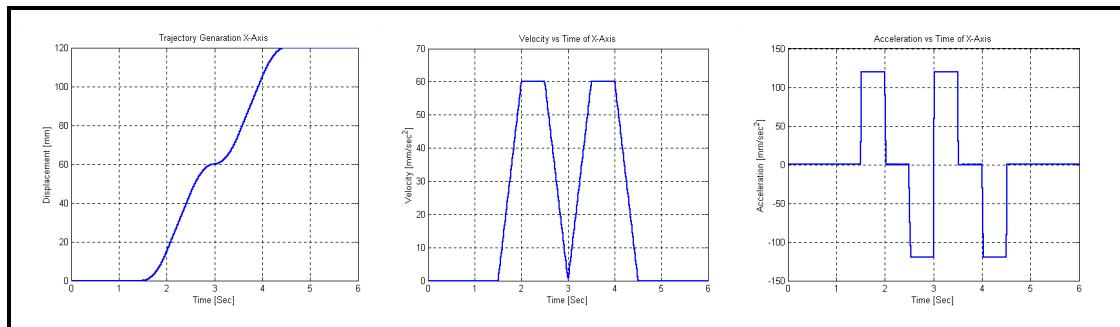
$$m_y \ddot{y} + d_y \dot{y} + f_{sy} \text{sgn}(y) = f_y - f_{ey} \quad (\text{III.26})$$

$$m_z \ddot{z} + d_z \dot{z} + f_{sz} \operatorname{sgn}(\dot{z}) + 9.81m_z = f_z - f_{ez} \quad (\text{III.27})$$

### III.5 TRAJECTORY GENERATION

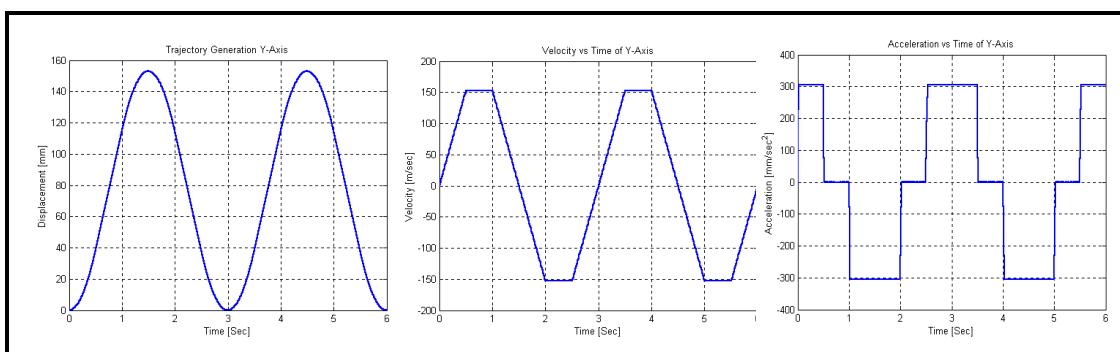
Trajectory generation is used in all experiments including the identification process with the position control and interaction control strategies.

A MATLAB Script Code is written for the trajectory generation with the principles told in [1]. User defines initial time and position, and final time and position and the code calculates the trajectory. The code for the generation of the trajectory for the word “M” for x axis is given in the Appendix B-1.

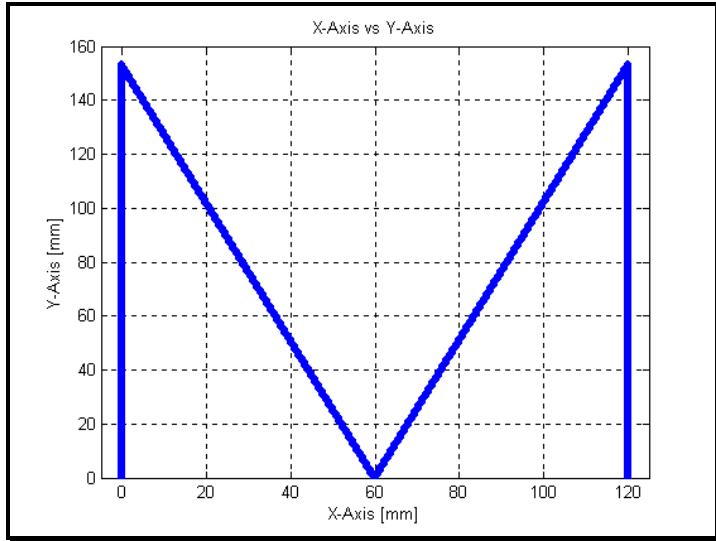


**Figure III.4 Displacement, Velocity and Acceleration vs Time Graphs of X-Axis**

With the code file, the trajectory is determined for both x-axis and y-axis. The displacement, velocity and the acceleration graphs with respect to time is given in the Figure III.4 for x-axis and Figure III.5 for y-axis.



**Figure III.5 Displacement, Velocity and Acceleration vs Time Graphs of Y-Axis**



**Figure III.6 Trajectory Generation Result, X-Axis vs Y-Axis plot**

With the trajectory generated for both axis the capital letter "M" is plotted as shown in the Figure III.6.

The same methodology is applied for the other capital letters, and at the last the trajectory of the word "MARMARA" is generated and used for writing on a paper in [58].

## III.6 IDENTIFICATION OF THE THREE-AXIS CARTESIAN ROBOT ARM

Identification follows the general strategy used in Robotics and adopted by several researchers [51-53]. The main idea is to use the inverse dynamic model, which calculates the control input (torque) as a function of the state. This model is linear in relation to the dynamic parameters, which can be identified using *least square techniques* and *step by step techniques*. Torque data are sampled while the robot is tracking an exciting trajectory, which means a closed loop identification, and the output is converted to the position, velocity and acceleration consequently by taking the derivative with respect to sample time. This approach is quite different from those based on the use of the direct state space model or the transfer function model for linear systems.

The identification strategies are not only used for identifying the manipulator

parameters, but also used to estimate the state and parameter that serve to distinguish the task geometry to dynamically adapt subsequent motions in [54], and environment estimation for enhanced impedance control to prevent the shortcoming of improper target impedance selection in [55].

### III.6.1 Step-By-Step Identification Procedure

In the identification part of the study, step-by-step identification method is used in order to determine the dynamic parameters of Three-Axis Cartesian Robot Arm. The other methods that are mentioned in section II.2 could also be carried on the process. However during the experimental study the least square and boot strap were insufficient when applied alone to the process because of the high static friction and nearly quasi-static nature of the structure. Therefore there became a need to identify some of the parameters first with the knowledge of the dynamics of the structure and then determine the rest by the suitable methods. With the pre-determined parameter it is more convenient to identify the other dynamic parameters. The process can be summarized as follows;

- In order to identify the friction force  $f_{sr}$ , a ramp torque is applied. As the manipulator starts to move, the data sampled and the torque that is exerted to overcome the static friction force is found, at the point where the velocity and the acceleration is zero.
- For the identification of the damping coefficient  $\mathbf{D}$ , a step torque is applied, and data is sampled for a while that corresponds a constant velocity motion.
- An arbitrary path, generated with the trajectory generation code is tracked by the robot arm in order to identify the equivalent mass of the system. With the identified parameters at the previous stages, the dynamic model is calculated with Least-Square technique, and the equivalent mass is identified.

It can be shown that the dynamic equation of the manipulator (II.19) can be cast in a linear form with respect to a suitable ( $p \times 1$ ) vector  $\boldsymbol{\pi}$  of dynamic parameters recall as equation (II.20);

$$\mathbf{Y}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})\boldsymbol{\pi} = \boldsymbol{\tau} - \mathbf{J}^T(\mathbf{q})\mathbf{h}$$

Where the  $(n \times p)$  matrix  $Y$  is termed *regressor* of the dynamic model. In general, the dynamic parameters depend on the mass, first moment of inertia and inertia tensor of each link, and the friction coefficient of each joint.

### III.6.2 Identification in X-Axis

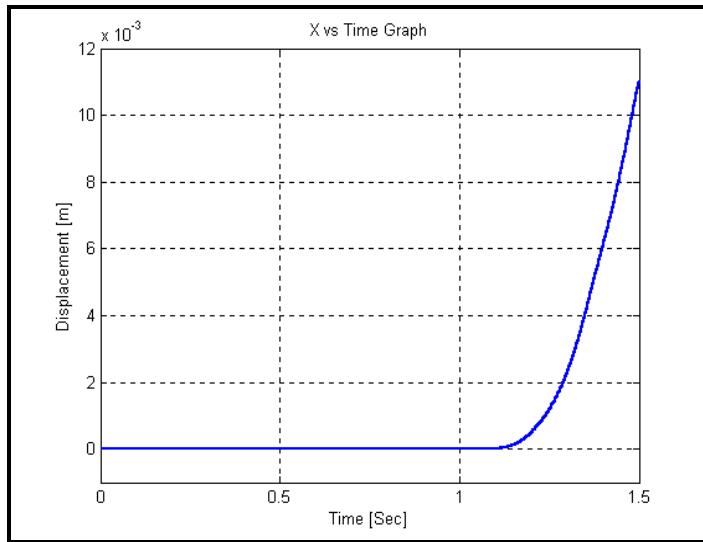
The model of x-axis is

$$m_x \ddot{x} + d_x \dot{x} + f_{sx} \operatorname{sgn}(x) = f_x \quad (\text{III.28})$$

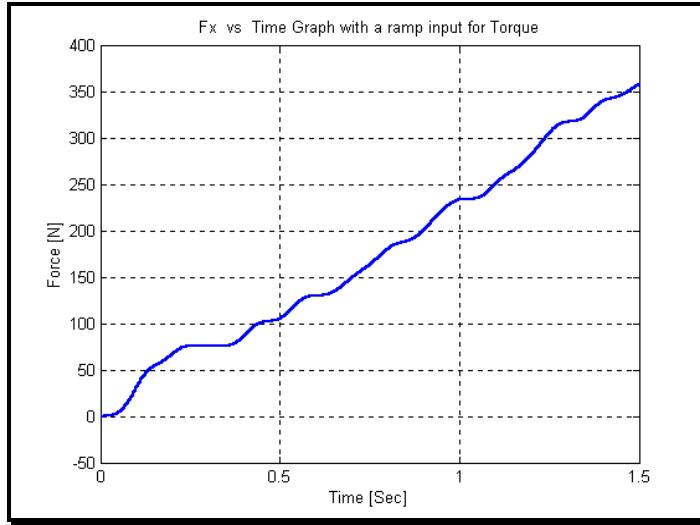
The step-by-step procedure is carried out and at the first stage ramp input is given to the system and  $f_{sx}$  is identified by defining the force when the axis is to move (Figure III.7). At this time

$\dot{x} = 0$  and  $\ddot{x} = 0$ , so the Equation (III.28) becomes

$$f_{sx} = f_x \quad (\text{III.29})$$



**Figure III.7** X-axis, Displacement versus time graph with ramp input for torque

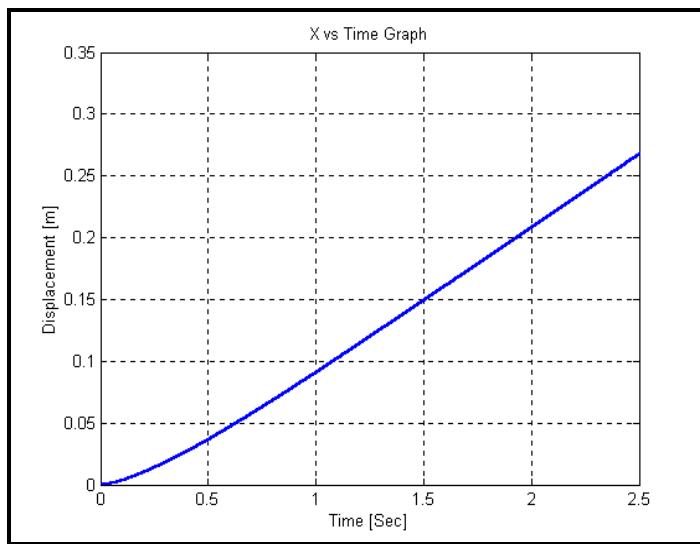


**Figure III.8     X-axis, Force versus time graph with ramp input for torque**

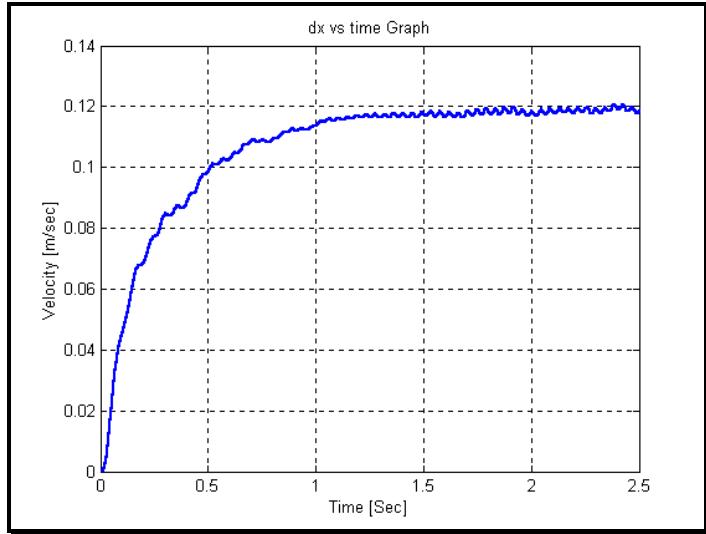
Torque is obtained from the D/A Converter and after filtering (Appendix B-4) the data it is converted to force (Figure III.8) by using equations (III.11), (III.14) and (III.21). This operation is repeated for several times and an average value for  $f_{sx}$  is obtained as;

$$f_{sx} = 215 \text{ N}$$

At the second stage of identification, step input is given for torque. With the constant torque the axis moves with a constant velocity, which means the acceleration is zero (Figure III.11). Figure III.9 shows the axis displacement with constant velocity.



**Figure III.9     X-axis, Displacement versus time graph with constant torque**



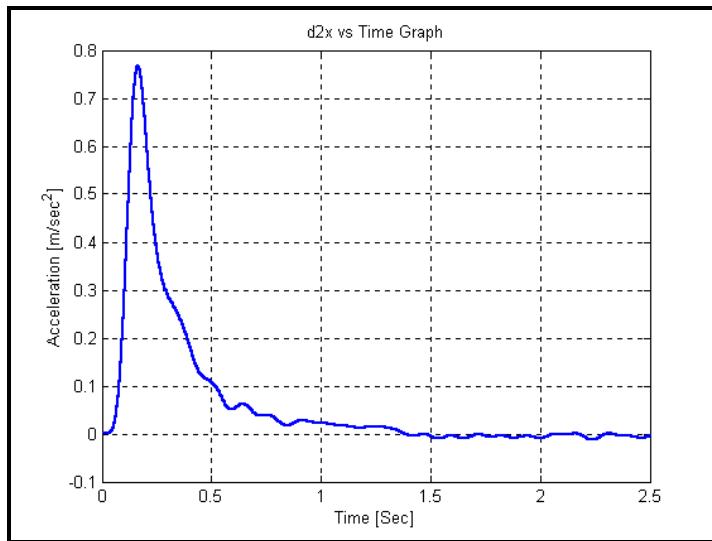
**Figure III.10 X-axis, velocity versus time graph with constant torque**

In this motion,  $\ddot{x} = 0$ , so the equation (III.28) becomes an equation of one unknown  $d_x$ .

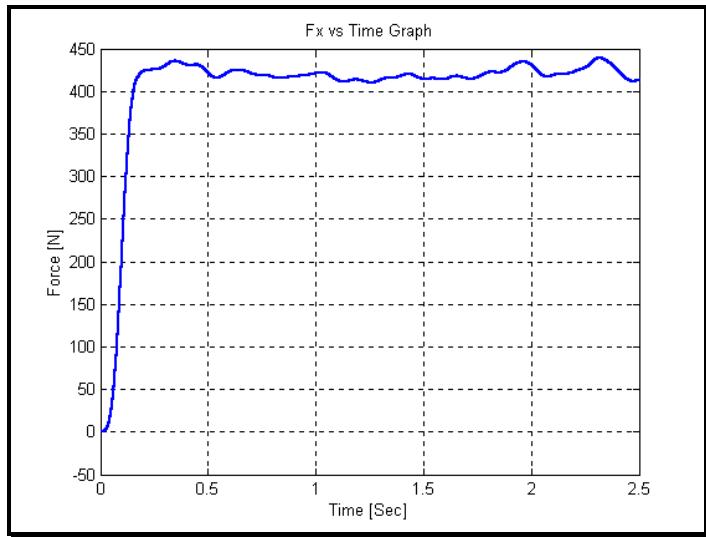
$$d_x \dot{x} + f_{sx} \operatorname{sgn}(\dot{x}) = f_x \quad (\text{III.30})$$

Collecting the constant part of the velocity  $\dot{x}$  (Figure III.10), and corresponding force  $f_x$  (Figure III.12) values in separate vectors with the known friction force  $f_{sx}$ , the damping coefficient is determined solving equation (III.30) using least square technique as

$$d_x = 1670 \text{ N s/m}$$

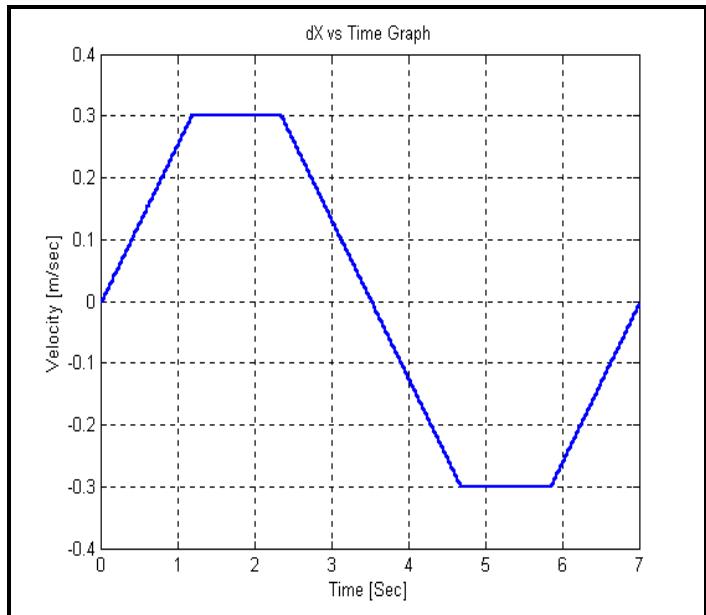


**Figure III.11 X-axis, acceleration versus time graph with constant torque**

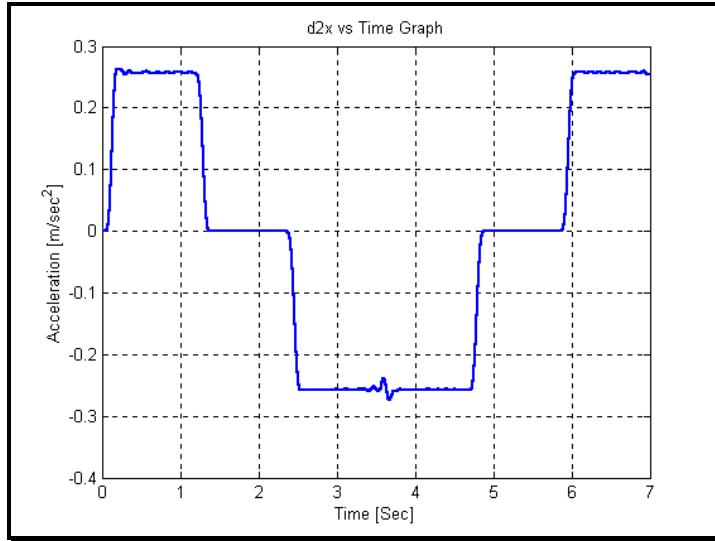


**Figure III.12 X-axis, force versus time graph with constant torque**

For the third and last stage an arbitrary trajectory with PID control is used for determining  $m_x$ . The velocity profile is given in Figure III.13.



**Figure III.13 X-axis, Velocity versus time graph**



**Figure III.14 X-axis, Acceleration versus time graph**

In this motion, the equation (III.28) becomes

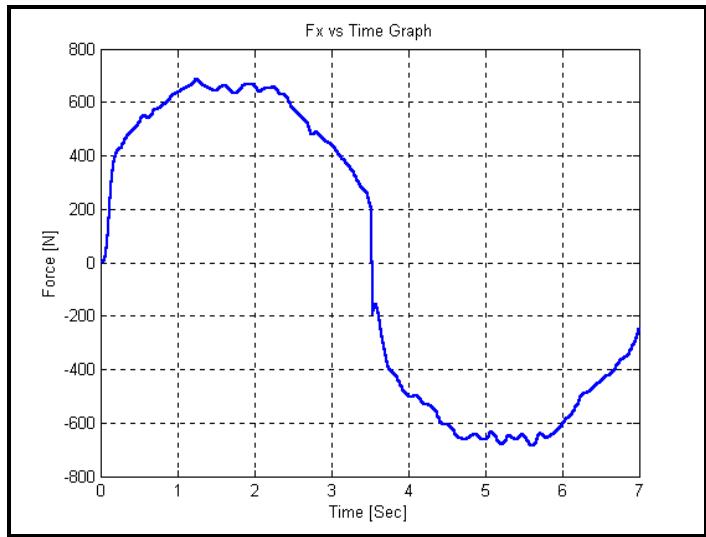
$$m_x \ddot{x} + d_x \dot{x} + f_{sx} \operatorname{sgn}(x) = f_x \quad (\text{III.31})$$

where  $d_x = 1670 \text{ N s/m}$ , and  $f_{sx} = 215 \text{ N}$

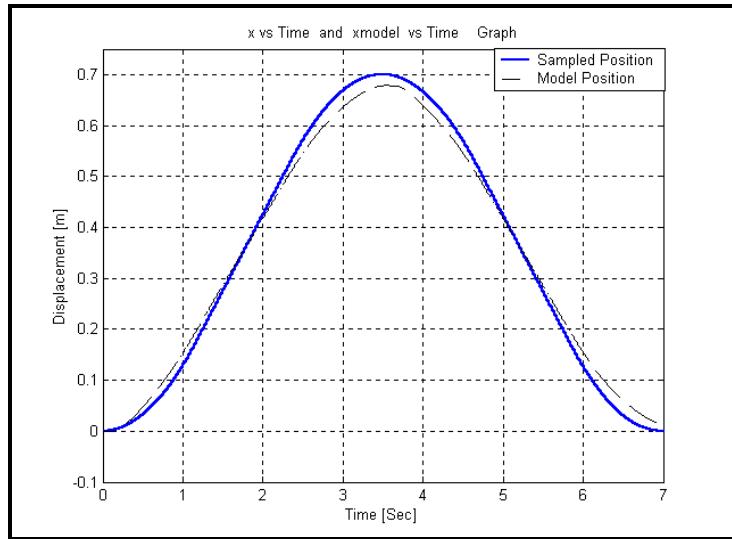
Collecting the velocity  $\dot{x}$  (Figure III.13), acceleration  $\ddot{x}$  (Figure III.14), and corresponding force  $f_x$  (Figure III.15) values in separate vectors in suitable time interval with the known value of friction force  $f_{sx}$  and damping coefficient  $d_x$  the mass  $m_x$  of the x-axis is determined solving equation (III.31) with the MATLAB using Least Square technique as

$$m_x = 38 \text{ kg}$$

It must be stated that, because of the structure of the 3-axis Cartesian manipulator the model has high static friction, and the maximum allowable acceleration is not so much which decrease the effect of the mass to the model. This results to a nearly quasi-static model resulting an inaccurate mass identification. The mass is changed from 25 kg to 40 kg to test its effect, and all the simulated model results were nearly identical.



**Figure III.15 X-axis, Force versus time graph**



**Figure III.16 X-axis, Response of the system versus Model response**

With the dynamic parameters that are identified, an error model is built (Appendix B-5) and simulated. The difference between the sampled position and the model position can be seen from the Figure III.16.

### III.6.3 Identification in Y-Axis

The model of y-axis is

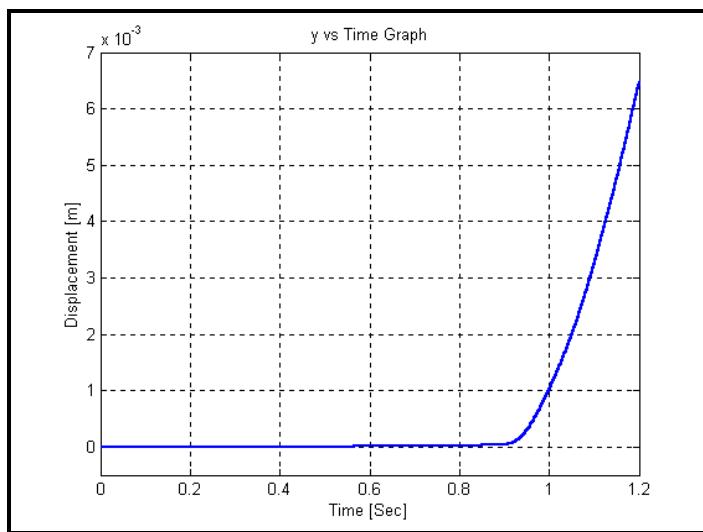
$$m_y \ddot{y} + d_y \dot{y} + f_{sy} \operatorname{sgn}(\dot{y}) = f_y \quad (\text{III.32})$$

The step by step procedure is carried out and at the first stage ramp input is given to the system and  $f_{sy}$  is identified by defining the force when the axis is to move (Figure III.17). At this time

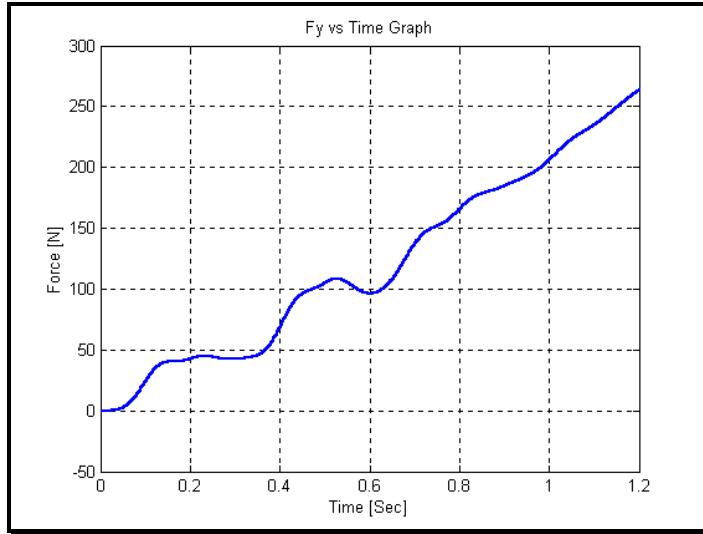
$\dot{y} = 0$  and  $\ddot{y} = 0$ , so the equation (III.32) becomes

$$f_{sy} = f_y \quad (\text{III.33})$$

Torque is obtained from the D/A Converter and after filtering (Appendix B-4) the data it is converted to force (Figure III.18) by using equations (III.11), (III.14) and (III.21).



**Figure III.17** Y-axis, Displacement versus time graph with ramp input for torque

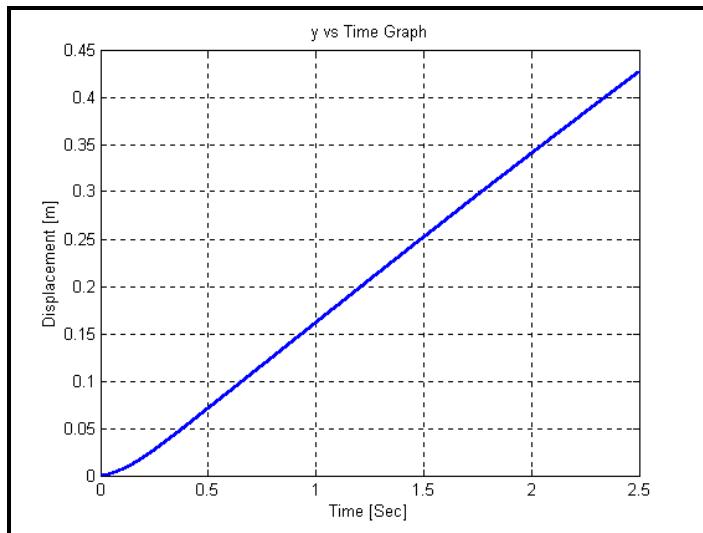


**Figure III.18 Y-axis, Force versus time graph with ramp input for torque**

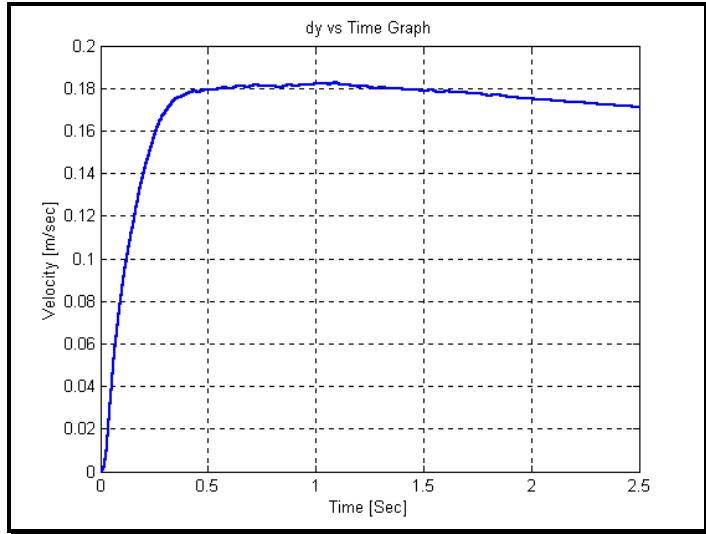
From Figure III.17 the time where the axis is to move is determined and from Figure III.18 the force that corresponds to that time is equalized to frictional force. This operation is repeated for several times and an average value for  $f_{sy}$  is obtained as;

$$f_{sy} = 184 \text{ N}$$

At the second stage of identification, step input is given for torque. With the constant torque the axis moves with a constant velocity, which means the acceleration is zero (Figure III.21). Figure III.19 shows the axis displacement with constant velocity.



**Figure III.19 Y-axis, Displacement versus time graph with constant torque**



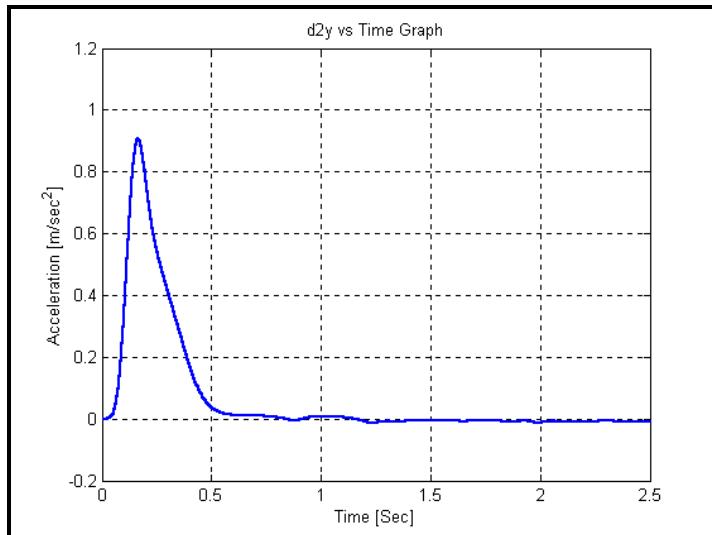
**Figure III.20 Y-axis, velocity versus time graph with constant torque**

In this motion,  $\ddot{y} = 0$ , so the equation (III.32) becomes an equation of one unknown  $d_y$ .

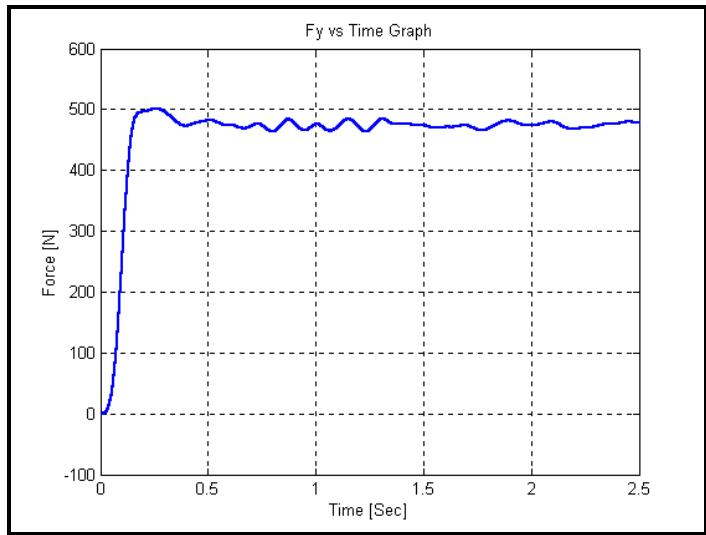
$$d_y \dot{y} + f_{sy} \operatorname{sgn}(\dot{y}) = f_y \quad (\text{III.34})$$

Collecting the constant part of the velocity  $\dot{y}$  (Figure III.20), and corresponding force  $f_y$  (Figure III.22) values in separate vectors with the known friction force  $f_{sy}$ , the damping coefficient is determined solving equation (III.34) using least square technique as

$$d_y = 1620 \text{ N s/m}$$

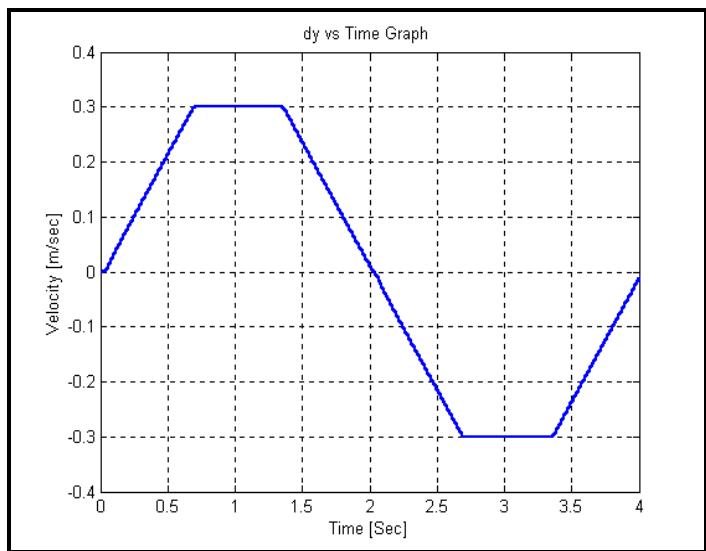


**Figure III.21 Y-axis, acceleration versus time graph with constant torque**

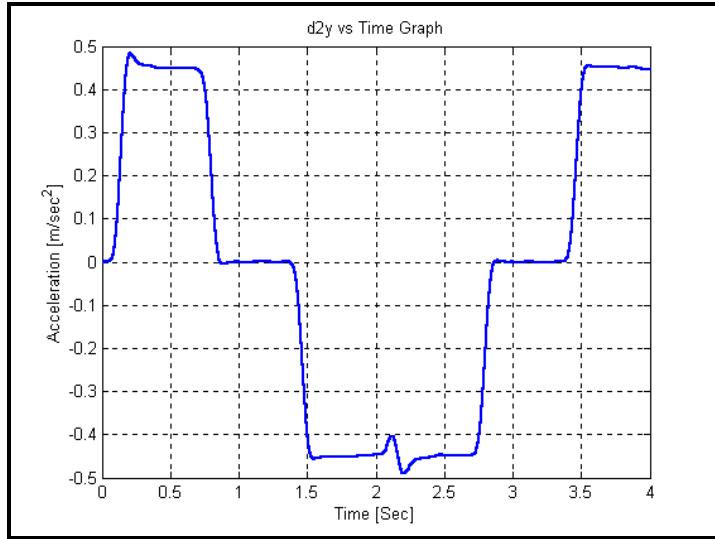


**Figure III.22 Y-axis, force versus time graph with constant torque**

For the third and last stage an arbitrary trajectory with PID control is used for determining  $m_y$ . The velocity profile is given in Figure III.23.



**Figure III.23 Y-axis, Velocity versus time graph**



**Figure III.24 Y-axis, Acceleration versus time graph**

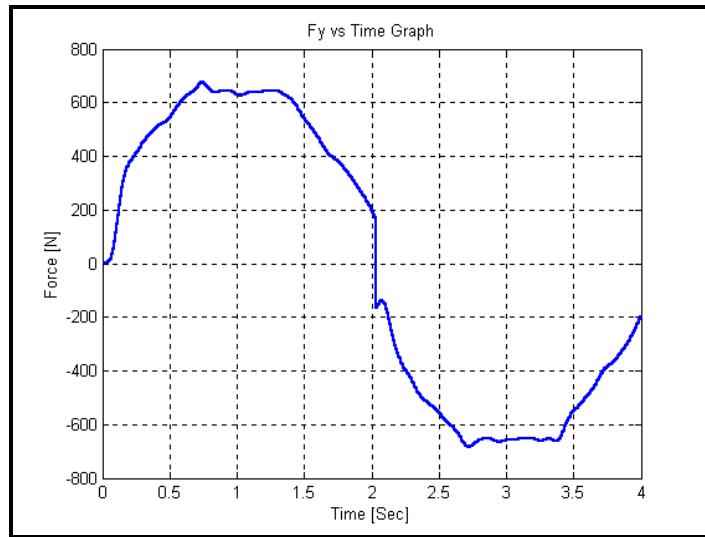
In this motion, the equation (III.32) becomes

$$m_y \ddot{y} + d_y \dot{y} + f_{sy} \operatorname{sgn}(\dot{y}) = f_y \quad (\text{III.35})$$

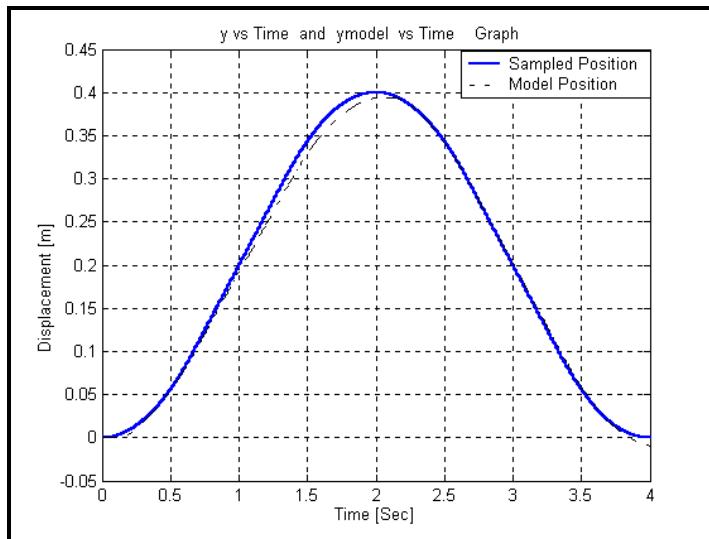
where  $d_y = 1620 \text{ N s/m}$ ,  $f_{sy} = 184 \text{ N}$

Collecting the velocity  $\dot{y}$  (Figure III.23), acceleration  $\ddot{y}$  (Figure III.24), and corresponding force  $f_y$  (Figure III.25) values in separate vectors in suitable time interval with the known value of friction force  $f_{sy}$ , and  $d_y$  the mass  $m_y$  of the y-axis is determined solving equation (III.35) with the MATLAB Least Square code as

$$m_y = 30 \text{ kg}$$



**Figure III.25 Y-axis, Force versus time graph**



**Figure III.26 Y-axis, Response of the system versus Model response**

With the dynamic parameters that are identified for y-axis, a model is built (Appendix B-5) and simulated. The difference between the sampled position and the model position can be seen from the Figure III.26. However it is not easy to determine the difference from the figure.

**Table III.2 Identification results**

Axis	<b><i>m</i></b>	<b><i>d</i></b>	<b><i>f<sub>s</sub></i></b>
x-axis	38 kg	1670 Ns/m	215 N
y-axis	30 kg	1620 Ns/m	184 N

X-Axis and Y-Axis are found to be suitable for the step-by-step identification. However during the experiments, it was noticed that the identification of Z-Axis is not suitable to be determined by step-by-step identification method, since it's friction force is not so much to be calculated from the read voltage where the motion starts, especially to the gravity direction.

The identification of the 3-axis Cartesian robot arm can also be carried by transforming the dynamic equations III.25, III.26 and III.27 to discrete form

$$a_1x_{i+1} + b_1x_i + c_1x_{i-1} + d_1f_{sx} \operatorname{sgn}(x_{i+1} - x_{i-1}) = f_{x_i} \quad (\text{III.36})$$

$$a_2y_{i+1} + b_2y_i + c_2y_{i-1} + d_2f_{sy} \operatorname{sgn}(y_{i+1} - y_{i-1}) = f_{y_i} \quad (\text{III.37})$$

$$a_3z_{i+1} + b_3z_i + c_3z_{i-1} + d_3f_{sz} \operatorname{sgn}(z_{i+1} - z_{i-1}) + 9.81m_z = f_{z_i} \quad (\text{III.38})$$

which can be written as

$$Ar_{i+1} + Br_i + Cr_{i-1} + Df_{sr} \operatorname{sgn}(r_{i+1} - r_{i-1}) + MJg = f_i \quad (\text{III.39})$$

and in the matrix form

$$\begin{bmatrix} r_{i+1} & r_i & r_{i-1} & \operatorname{sgn}(r_{i+1} - r_{i-1}) & Jg \end{bmatrix} \begin{bmatrix} A \\ B \\ C \\ D \\ M \end{bmatrix} = f_i \quad (\text{III.40})$$

which gives a least square sense to find the parameters **A**, **B**, **C**, **D** and **M** which are directly related to the dynamic parameters.

## III.7 IDENTIFICATION OF THE AMPLIFIER CONSTANTS

The amplifier accepts input voltage between +10 and -10 Volts and amplifies the input voltage level to the motor signal level which gives the required torque. Thus, there exists a relation between the input voltage and the output torque

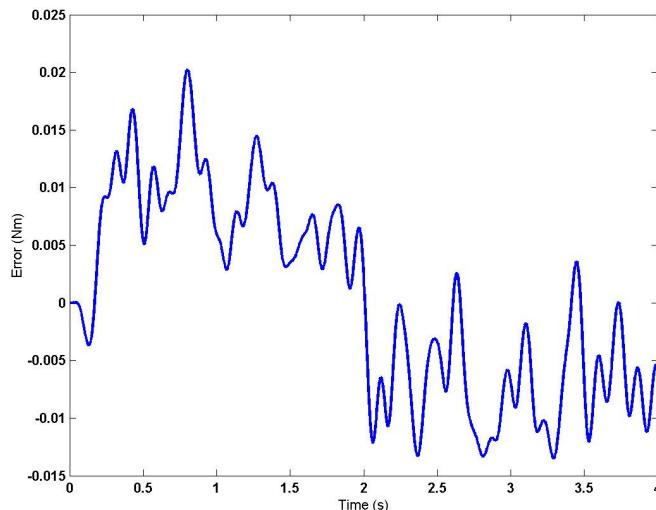
$$\tau_m = \mathbf{K}_a \mathbf{V}_i \quad (\text{III.41})$$

where  $\mathbf{K}_a$  is the (3x3) amplifier constant and  $\mathbf{V}_i$  is the (3x1) input voltages.

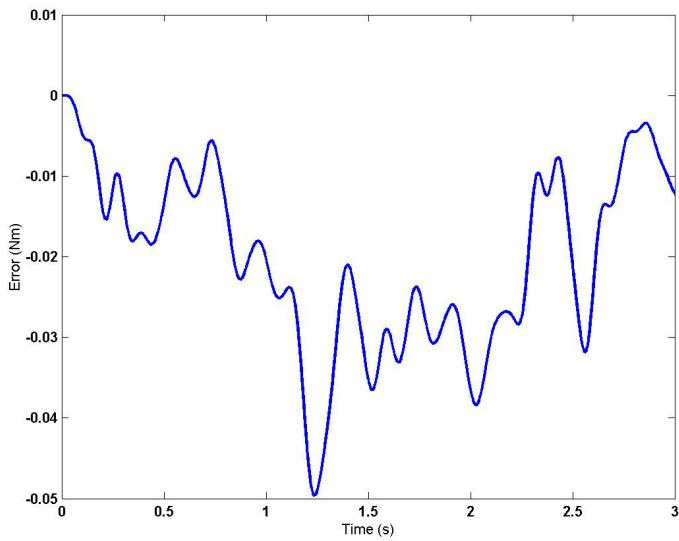
In order to identify the amplifier constant  $\mathbf{K}_a$  an arbitrary trajectory with PID control is used. The input voltage and the voltage that corresponds to the motor torque read from the amplifier are recorded. The motor torque is calculated from the voltage read from the amplifier using equation III.11. Then equation III.41 is solved in a least square sense and the amplifier constants are found as

$$\mathbf{K}_a = \begin{bmatrix} 0.0662 & 0 & 0 \\ 0 & 0.06844 & 0 \\ 0 & 0 & 0.11246 \end{bmatrix} \text{Nm/V} \quad (\text{III.42})$$

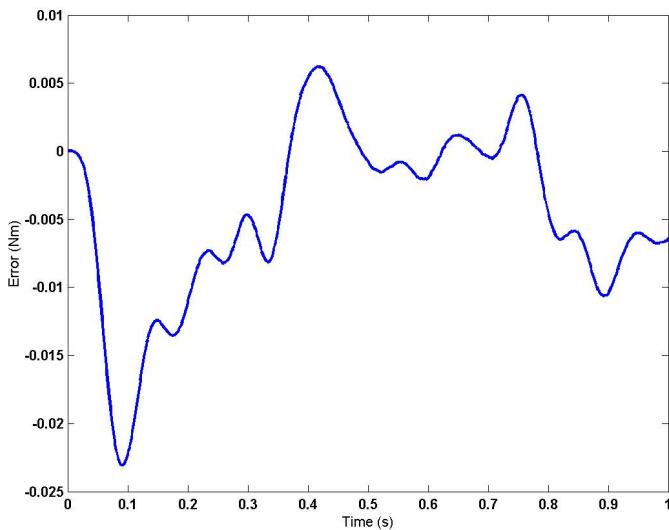
The error between the motor torque read from the amplifier and the calculated motor torque with III.41 are plotted with respect to time in Figure III.27, Figure III.28 and Figure III.29.



**Figure III.27 X-Axis amplifier constant error**



**Figure III.28 Y-Axis amplifier constant error**

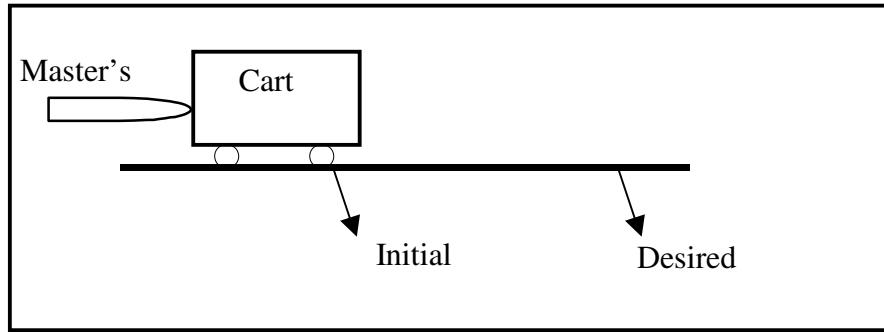


**Figure III.29 Z-Axis amplifier constant error**

## III.8 IMPEDANCE CONTROL OF A MANIPULATOR COOPERATING WITH A HUMAN

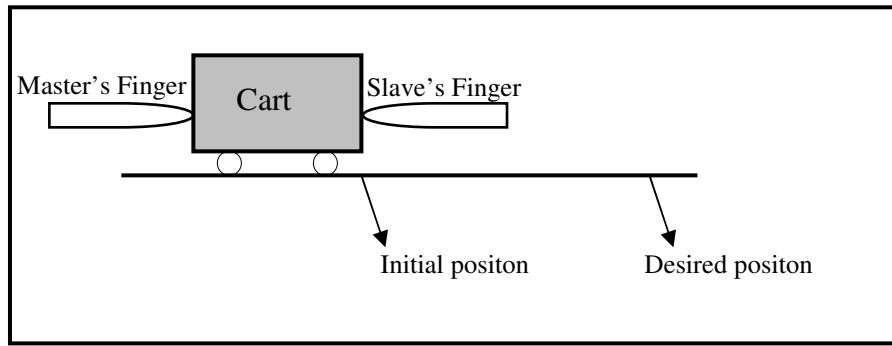
Robots are expected to execute tasks in cooperation with humans. Control systems for such robots should be designed so as to adapt to human characteristics. Pushing a cart on a frictionless plane in cooperation with a human is a basic example to such tasks. As seen in Figure III.30, it is impossible to control the motion of an object from one point to another only by pushing in one direction (master) without

supporting in the opposite direction (slave). As soon as the force is applied the object will move.



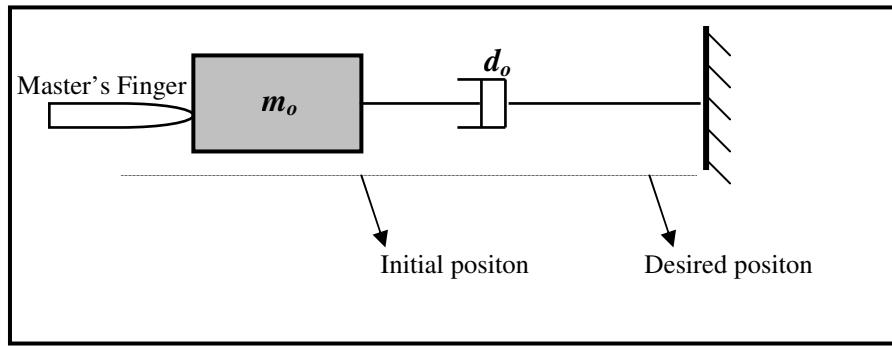
**Figure III.30** Pushing a cart on a frictionless plane by a unilateral force

However with the help of another force (slave) the cart can be pushed to the desired position with enough accuracy as seen in Figure III.31.



**Figure III.31** Positioning a cart on a frictionless plane by a bilateral force

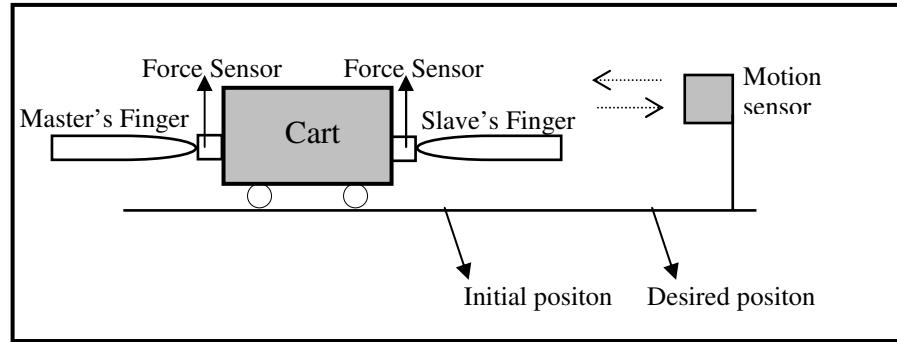
If position and force acting to the master's finger can be measured it will be possible to model the slave's behavior with a mass and a damper as seen in Figure III.32.



**Figure III.32** Model of the slave

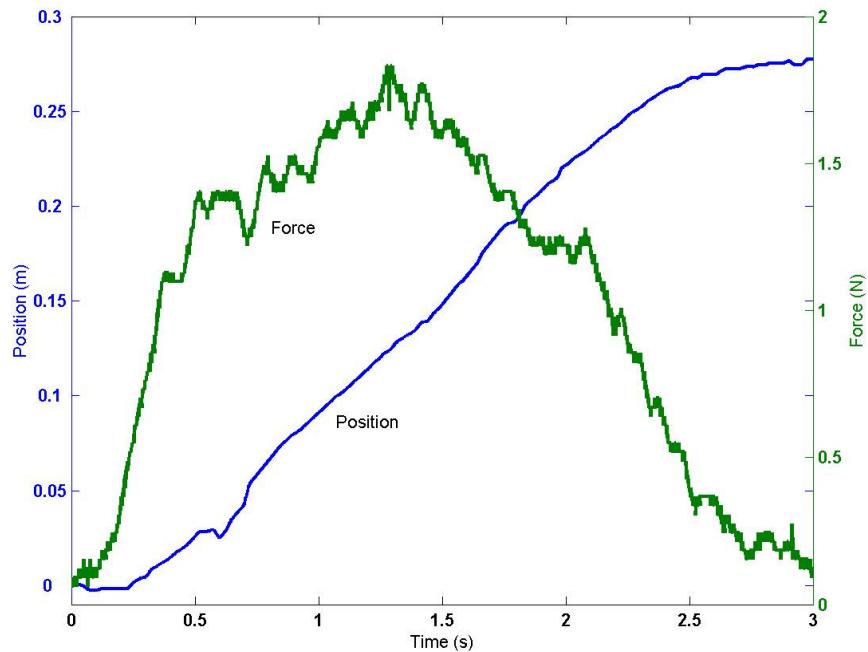
In order to model the behaviour of the slave the experiment shown in Figure III.33 is implemented. Two force sensors are attached on a cart for measuring the forces at the

master and slave's fingers, and a motion sensor is positioned at the end of the plane to measure the position of the cart.



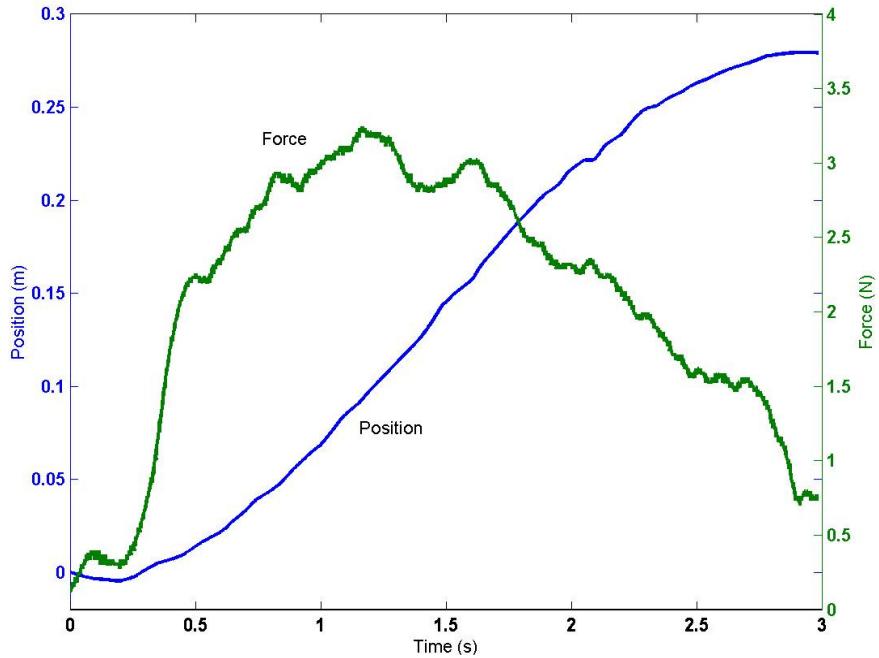
**Figure III.33** Experimental setup for estimating the slave's model

The process is repeated several times with different slave behaviors.



**Figure III.34** First data set from experiments

Figures III.34 and III.35 show the change in position of the cart and the force acting by the master's finger on the cart, while the cart is moved from initial position to the desired position, for two different trials.



**Figure III.35 Second data set from experiments**

The position of the cart is measured by motion sensor, which is a sonar-ranging device, so the data is highly affected by noise, as seen in Figure III.34.

The slave's behavior is modeled as a mass-damper system with the following equation,

$$f_e = m_o \ddot{x} + d_o \dot{x} \quad (\text{III.43})$$

If the acceleration, velocity and force data are collected into separate vectors, one can write the equation in matrix form as

$$\mathbf{f}_e = [\ddot{x} \quad \dot{x}] \begin{bmatrix} m_o \\ d_o \end{bmatrix} \quad (\text{III.44})$$

Note that,  $\mathbf{f}_e$ ,  $\dot{x}$  and  $\ddot{x}$  are all vectors with  $(n \times 1)$  dimension, where  $n$  is number of data recorded in 0.001 second sample time. After modification, the cart's mass and damping coefficient can be obtained as

$$\begin{bmatrix} m_o \\ d_o \end{bmatrix} = [\ddot{x} \quad \dot{x}]^+ \mathbf{f}_e \quad (\text{III.45})$$

where  $[\ddot{x} \quad \dot{x}]^+ \in \mathbb{R}^{(n \times 2)}$  is the pseudoinverse[8] of the matrix. Equation III.45 can be solved with the least square method to obtain mass  $m_o$  and damping coefficient  $d_o$  of

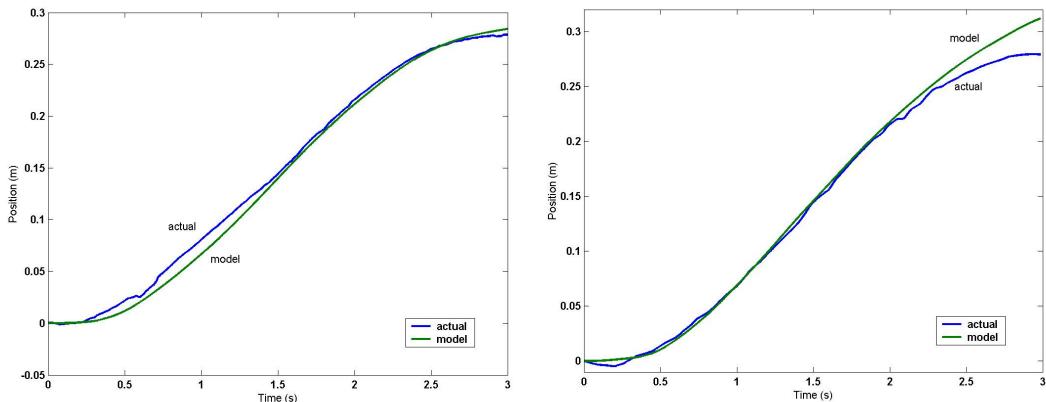
the slave model. This is done using Matlab, after deriving the velocity and the acceleration from the position read from the motion sensor and filtering with a lowpass filter of 15Hz frequency. Because of the high noise in position data, and low effect of the mass to the system, the solution of eqn. III.45 gave the results listed in Table III.3, which make the robot end-effector bounce when came in contact with the cart. By trial and error the parameters, which approximates a better model, are listed in Table III.4. Using these parameters, slave models are simulated with the force acting on the master's finger and compared with the experimental results to see the model errors (Figure III.36).

**Table III.3      Estimated results of the slave models (First)**

Model	$m_o$	$d_o$
1	0.58 kg	10.74 Ns/m
2	0.14 kg	19.52 Ns/m

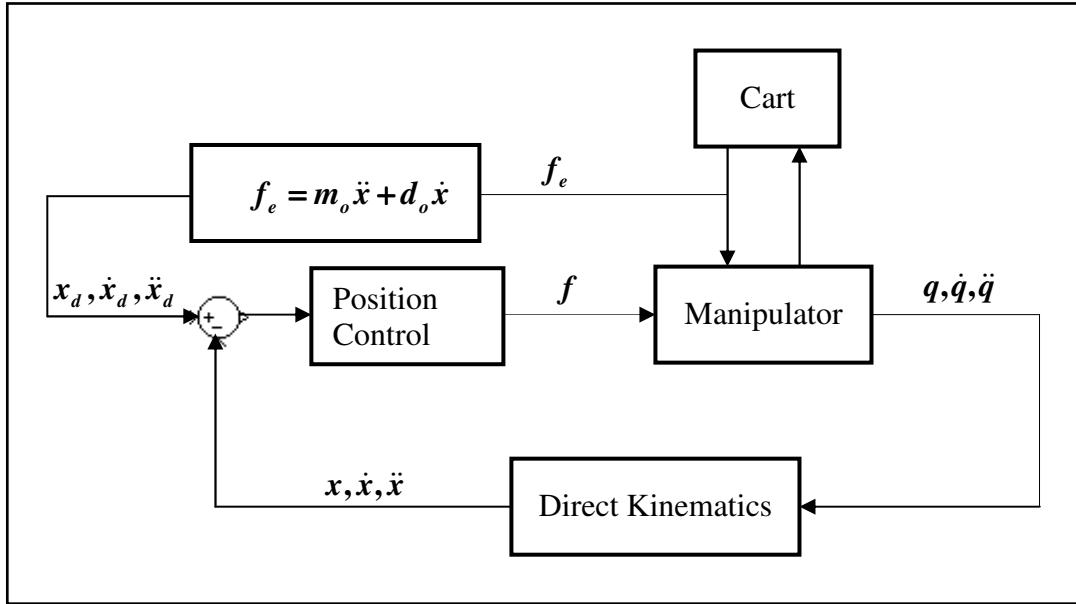
**Table III.4      Estimated results of the slave models (Used)**

Model	$m_o$	$d_o$
1	1.92 kg	10.74 Ns/m
2	2.23 kg	19.52 Ns/m



**Figure III.36      Experimental and simulated trajectories for first and second data set**

In order to command the manipulator to behave like the slave modeled with Eqn. III.42, the impedance control scheme seen in Figure III.37 is used (Appendix B-6).



**Figure III.37 Impedance control scheme of the manipulator behaving as the slave**

This control scheme will be applied to only the  $y$ -axis of the 3-Axis Cartesian manipulator shown in Fig. III.1. The equation of motion for the  $y$  axis can be written as

$$m_y \ddot{y} + d_y \dot{y} + f_{sy} \operatorname{sgn}(\dot{y}) = f_y - f_{ey} \quad (\text{III.46})$$

where  $m_y$  is the mass,  $d_y$  is the damping coefficient,  $f_{sy}$  is the frictional force,  $f_y$  is the commanded force (control input) to the manipulator,  $f_{ey}$  is the external force and  $y$  is the position of the axis. The manipulator force  $f$ , the external force  $f_e$ , and the actual position  $x$  in the control block diagram (Figure III.37), correspond to the commanded force  $f_y$ , the external force  $f_{ey}$ , and the actual position  $y$  in the equation of motion (Eqn. III.46), respectively.

The position control, which commands the manipulator to the position of the slave model can be any type, such as PD control, feedforward PD control, computed torque control or resolved acceleration control. In the experiments, feedforward PD control is used (Eqns. II.26 and II.27), because the model dynamic parameters are known with enough accuracy from the identification results. If we rewrite these equations for the  $y$  axis of the 3-Axis Cartesian robot arm, they become

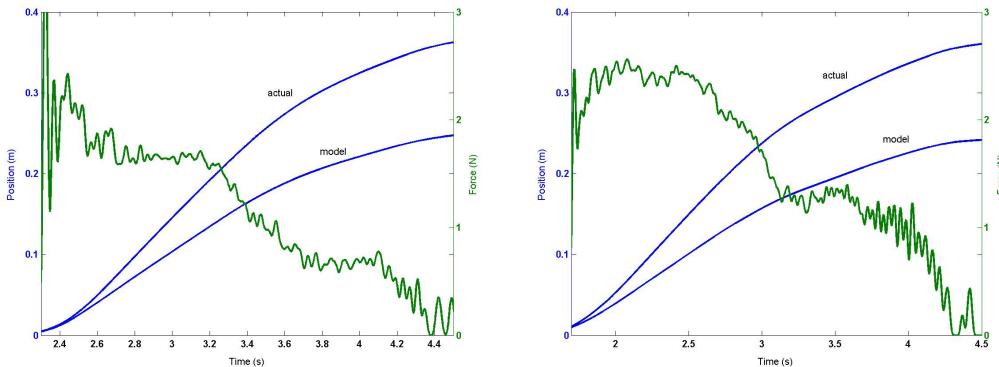
$$f_{ff} = m_y \ddot{y}_d + d_y \dot{y}_d + f_{sy} \operatorname{sgn}(\dot{y}_d) \quad (\text{III.47})$$

for the feedforward compensation term, and

$$f_y = f_{ff} + \mathbf{K}_v(\dot{y}_d - \dot{y}) + \mathbf{K}_p(y_d - y) \quad (\text{III.48})$$

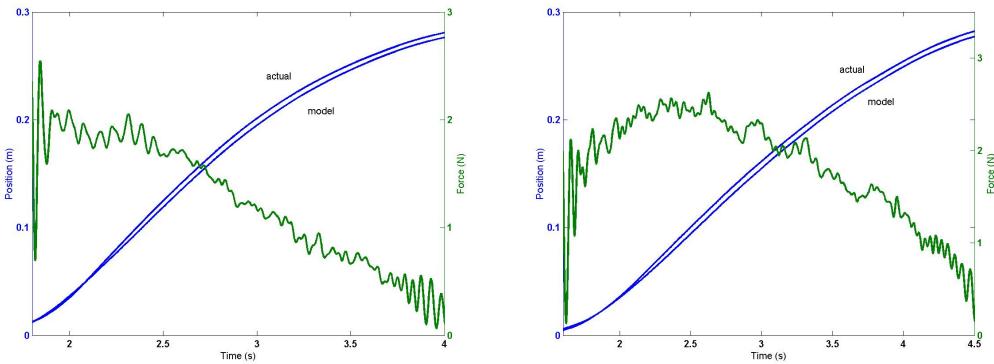
for the commanded force (control input) to the system described by Eqn. III.46, where  $\mathbf{K}_p$  and  $\mathbf{K}_v$  are the position and velocity gains of the controller. The simulink model of feedforward PD control and the slave's model can be seen in Appendix B-7 and Appendix B-8, respectively.

Several experiments have been done, and the force and position data of the end-effector, cooperating with a human to position the cart to the desired point are given in following figures. The time scales are different in all figures, due to the different times that the master starts to push the cart. In all figures, the slave model response to the external force, in other words the desired input to the position controller (Figure III.37) is also plotted.



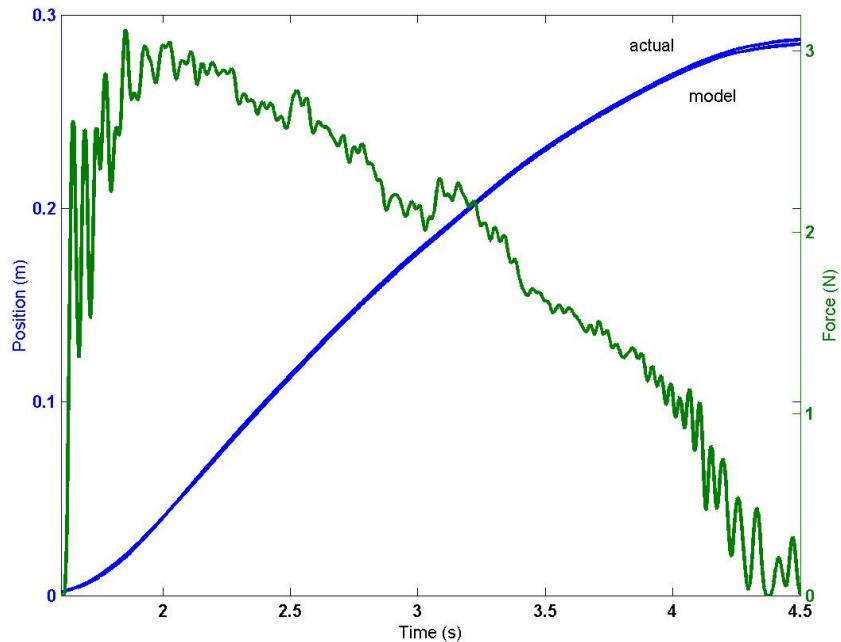
**Figure III.38 Trajectory and Force Data using first and second slave model with Feedforward PD Control [ $K_p=1000$ ,  $K_v=200$ .]**

As seen in Figure III.38 the position tracking accuracy is not good because of the position and velocity gains of feedforward PD control (Eqn. III.48). Increasing these gains gives better results (Figure III.39).



**Figure III.39 Trajectory and Force Data using first and second slave model with Feedforward PD Control [ $K_p = 10000$ ,  $K_v = 300$ .]**

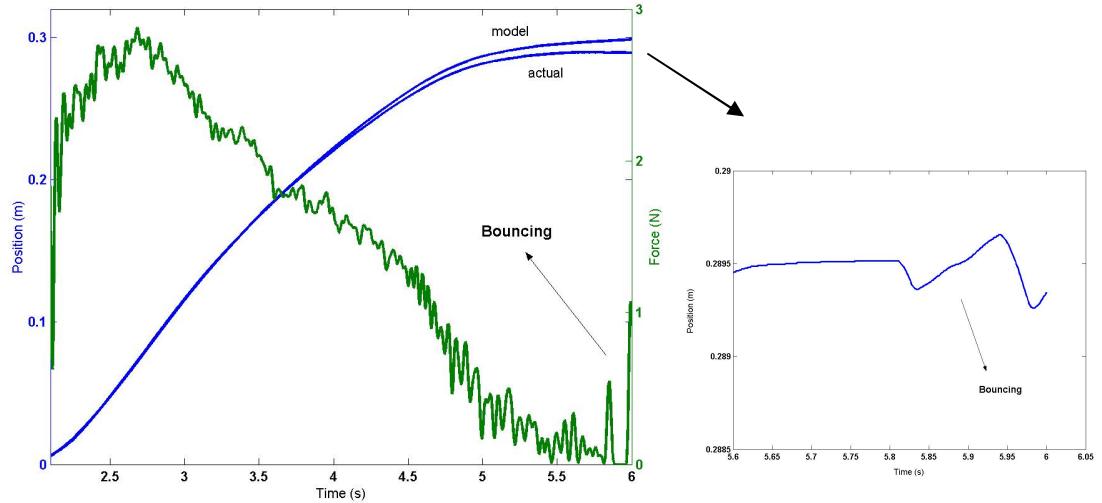
Second model ( $m_o=2.23\text{ kg}$ ,  $b_o=19.52\text{ Ns/m}$ ) gives better results for both positioning and ease of work from the master's perspective. Thus, that model is also tested for  $K_p=30000$  and  $K_v=2000$  control parameters to decrease the error, and it is very difficult to distinguish the actual and the model responses as seen in Figure III.40.



**Figure III.40 Trajectory and Force data using second model ( $m_o=2.23$ ,  $b_o=19.52$ ) with Feedforward PD Control [ $K_p = 30000$ ,  $K_v = 2000$ .]**

In practice there always exists a spring coefficient in slave's model (Eqn. III.43). But this results to a necessity for the master to apply a resultant force to keep the cart at the desired point, which is impossible in practice. Nevertheless, a small spring coefficient of  $3\text{ N/m}$  is used in slave's model to see its effects, and it has been seen

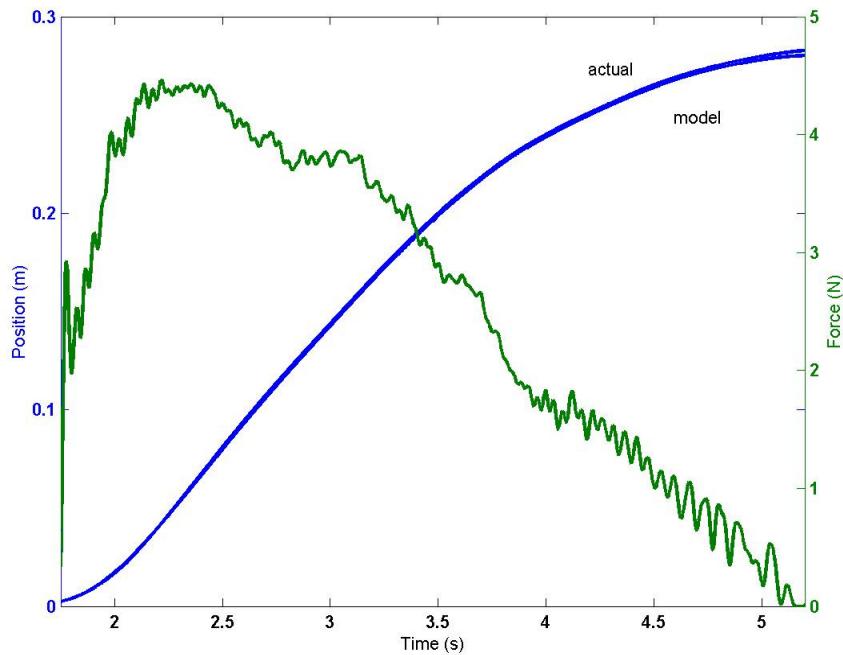
that the manipulator starts bouncing (Figure III.41) from the cart pushed by the master when the master tries to keep the cart at the desired point. This is due to the high stiffness of both the end-effector and the cart. This problem can be eliminated by using a soft skin or cover on either the end-effector or the cart. However, this will slow down the response of the system to the force feedback and not desired in real applications.



**Figure III.41** Trajectory and Force Data using second model ( $m_o=2.23$ ,  $b_o=19.52$ ,  $k_o=0.3$ ) with Feedforward PD Control [ $K_p = 30000$ ,  $K_v = 2000$ .]

Also, as seen in Figure III.41, the error between the end-effector position and the model position starts to increase at the end of the positioning process, which may be due to the uncertainties in the manipulator dynamics, and due to the spring force which constitutes a regulation problem for the master to balance the applied force to the spring force in order to keep the cart at the desired point.

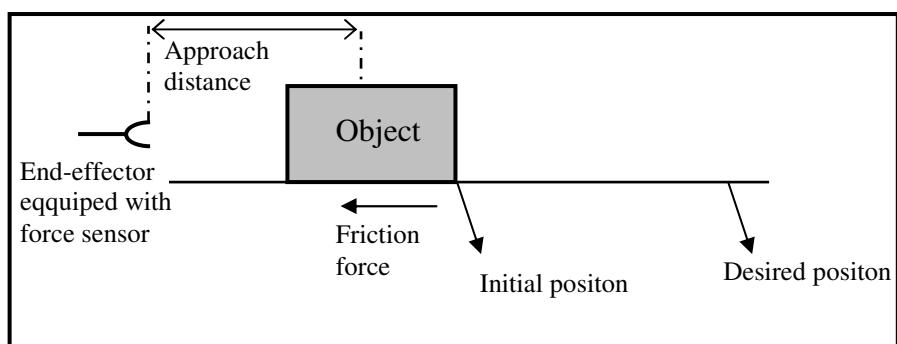
In order to see the effects, the mass and damping coefficients of the slave models are increased to 4 kg and 32 Ns/m, respectively. As seen in Figure III.42, the decrease in the error between the model and the manipulator is not so much, while the pushing force increases from 3 Newtons to 5 Newtons, which may be acceptable as a high value when the varying properties of the objects that are going to be pushed are considered. Thus, the external force will dominate the selection of the model.



**Figure III.42** Trajectory and Force Data using  $m_o=4$  and  $b_o=32$  with Feedforward PD Control [ $K_p = 30000$ ,  $K_v = 2000$ .]

### III.9 IMPEDANCE CONTROL OF A MANIPULATOR PUSHING AN OBJECT TO THE DESIRED POSITION

Another example to interaction schemes is the pushing operation of the manipulator of an object on a frictional plane to the desired point as illustrated in Figure III.43.



**Figure III.43** Pushing an object on a frictional plane

The hybrid control methods are not appropriate, because the task is one dimensional,

as they intend to control the force in constrained directions and position in unconstrained directions. The best choice for the task is the impedance control method (Section II.1.9.5), which constitutes a relation between the external force and the position of the end-effector and command the manipulator to behave as a predefined mass-damper-spring (impedance) system and succeed, even in case of disturbing effects. But in order to understand the advantages of this control method, the pushing process is also implemented using pure position control strategy. These two control strategies are designed on the same Simulink model (Appendix B-10). If the dynamics of the pushed object was known with an enough accuracy, it would be possible to calculate the force to position it to the desired point with the known trajectory, and this force can be used as the desired force input to a pure force controller. But this is practically impossible when we consider the varying dynamics of the object that is going to be positioned. At the same time if the stiffness of the object is high, this results to a high gain position feedback, which causes instability problems.

The task is to push an object of  $5.9 \text{ kg}$  mass on a board to the desired point. The desired position is selected as  $0.2 \text{ m}$  away from the end-effector, and the object is positioned in front of the end-effector with a varying approach distance (Figure III.43) from  $0$  to  $0.05 \text{ m}$ . The distance between the end-effector and the object varies in this range, because its thought that the control strategy used must handle the approach distance, which may not be known in practice.

When two objects contact each other, high contact forces occur in small time intervals. It can be said that, this contact force is mainly due to the velocities of the objects that contact each other. Therefore, it is possible to generate the trajectory, which makes the manipulator come in contact with the object with the appropriate velocity, if the distance of the object to the end-effector is known. Thus, even a pure position controller can handle the situation properly. But this distance is not known exactly in practice.

The appropriate controller for pushing process must also handle the disturbing effects like the one, when the object gets jammed between the end-effector and an obstacle. So, the need for the force feedback becomes important in the

controller selection.

In pushing experiments, a trapezoidal trajectory is generated which enables the manipulator move 0.2 m in 2 seconds. Also, the varying approach distance between the end-effector and the object (0~0.05 m) gave us the chance to analyze the different velocities of the manipulator when come in contact with the object. The experiments are done using only the y-axis of the 3-Axis Cartesian robot arm.

### III.9.1 Pushing with Position Controller

Rewriting the equation of motion for the y-axis

$$m_y \ddot{y} + d_y \dot{y} + f_{sy} \operatorname{sgn}(\dot{y}) = f_y - f_{ey} \quad (\text{III.49})$$

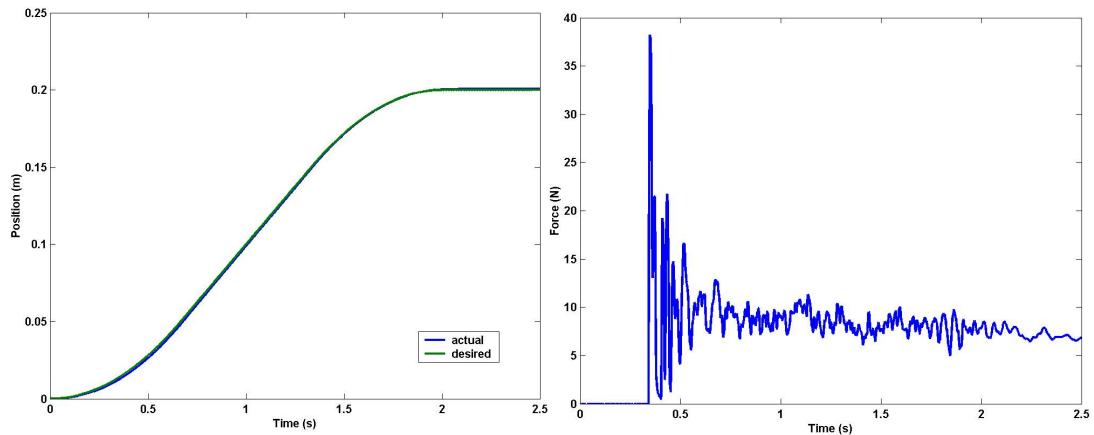
where  $m_y$  is the mass,  $d_y$  is the damping coefficient  $f_{sy}$  is the frictional force,  $f_y$  is the commanded force (control input) to the manipulator,  $f_{ey}$  is the external force (pushing force) and  $y, \dot{y}$  and  $\ddot{y}$  are the position, velocity and acceleration of the axis, respectively. First, the manipulator is commanded to move using the feedforward PID controller with the approach distance of 0.01 m. Feedforward PID control is used (Eqns. II.26 and II.27), because the model dynamic parameters are known with enough accuracy from the identification results. If we rewrite these equations for the y-axis of the 3-Axis Cartesian robot arm, they become

$$f_{ff} = m_y \ddot{y}_d + d_y \dot{y}_d + f_{sy} \operatorname{sgn}(\dot{y}_d) \quad (\text{III.50})$$

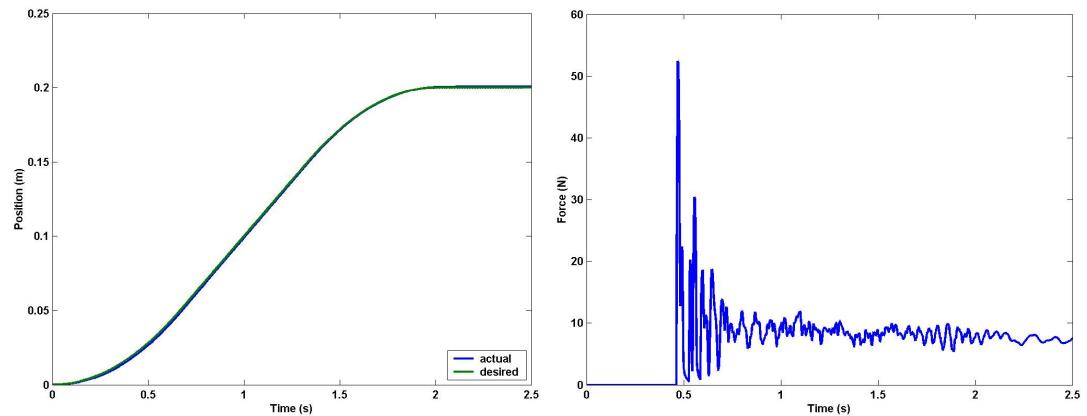
for the feedforward compensation term, and

$$f_y = f_{ff} + \mathbf{K}_v(\dot{y}_d - \dot{y}) + \mathbf{K}_p(y_d - y) + \mathbf{K}_i \int (y_d - y) dt \quad (\text{III.51})$$

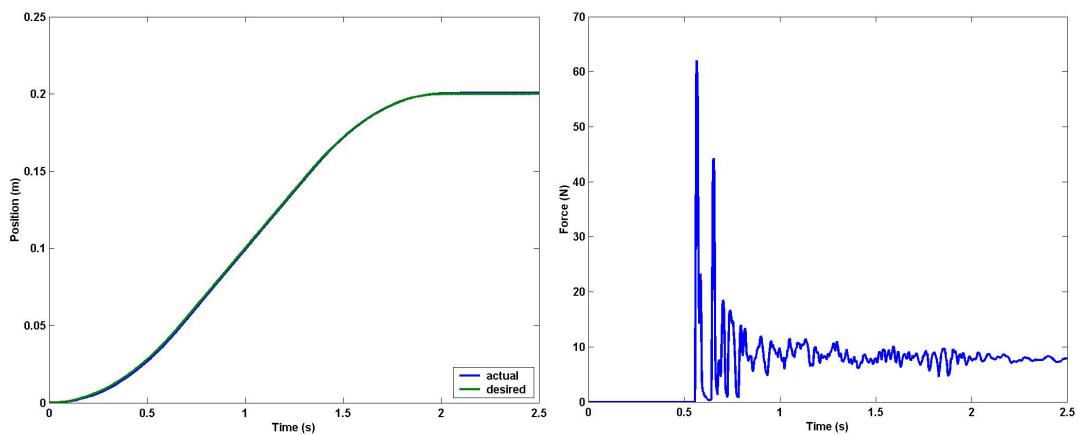
for the commanded force (control input) to the system described by Eqn. III.49 with the added integral term to decrease the steady state error. Then, the experiment is repeated five times with an approach distance changing from 0.01 m to 0.05 m with an increment of 0.01 m. The external (pushing) force and the position of the end-effector with the desired trajectory are plotted in following figures. The contact time can be distinguished from the pushing force plots.



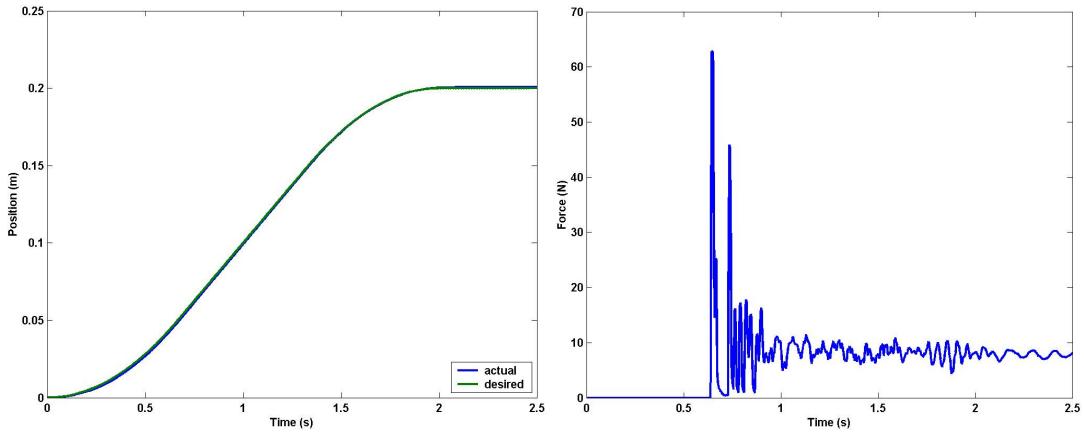
**Figure III.44 Position and External Force Data using Feedforward PID Control**  
 $[K_p = 50000, K_v = 8000, K_i = 10000]$ . Approach Distance=0.01 m]



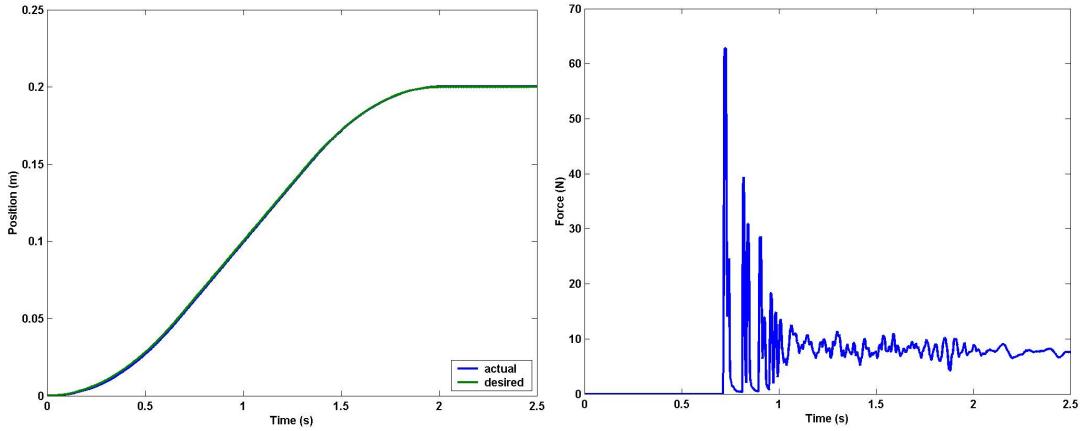
**Figure III.45 Position and External Force Data using Feedforward PID Control**  
 $[K_p = 50000, K_v = 8000, K_i = 10000]$ . Approach Distance=0.02 m]



**Figure III.46 Position and External Force Data using Feedforward PID Control**  
 $[K_p = 50000, K_v = 8000, K_i = 10000]$ . Approach Distance=0.03 m]



**Figure III.47 Position and External Force Data using Feedforward PID Control**  
 $[K_p = 50000, K_v = 8000, K_i = 10000]$ . Approach Distance=0.04 m]

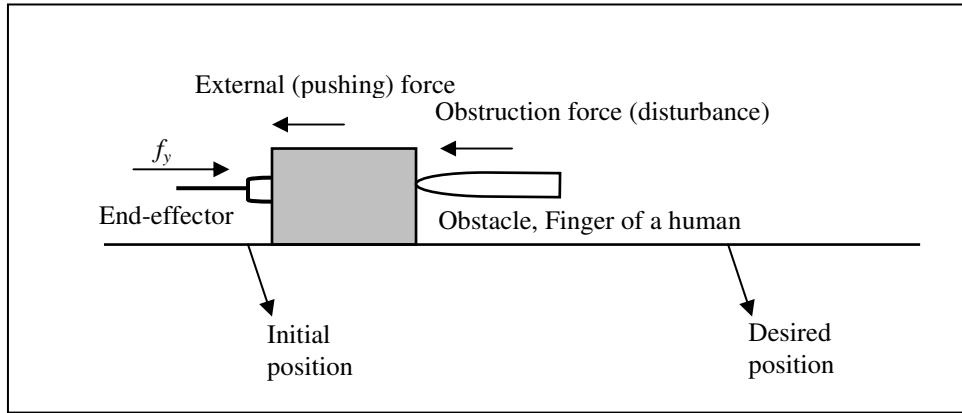


**Figure III.48 Position and External Force Data using Feedforward PID Control**  
 $[K_p = 50000, K_v = 8000, K_i = 10000]$ . Approach Distance=0.05 m]

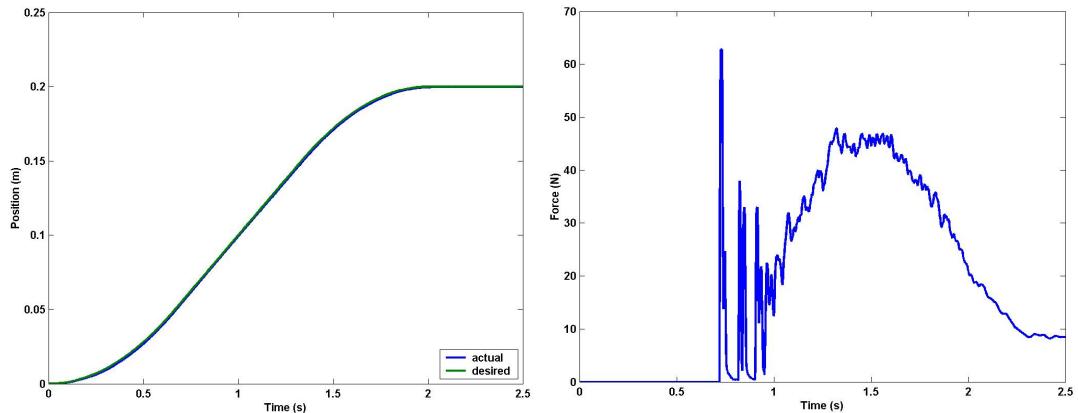
As expected, an increase in the approach distance results to a higher impact force. It must be noted that the force sensor can only measure forces from  $-50$  N to  $+50$  N. Thus, in Figures III.46, III.47 and III.48 the impact forces are not the true values since they exceed the force sensor limits. When the approach distance increases, the impact velocity of the end-effector increases according to the trajectory, but this does not result to a contact loose between the end-effector and the pushed object. The loss of contact may be expected when the approach distance increases more than 0.05 m, because the impact happens two times in Figure III.48.

The pure position controller does not take the obstructions in consideration because of the absence of force feedback; it only tries to reject the obstruction as a disturbance. Thus, when an obstacle tries to stop the object against the motion of the

end-effector as illustrated in Figure III.49, the controller increases the commanded force  $f_y$  to the manipulator, according to the Eqn. III.51, because of the increase in error. This results to an increase in external (pushing) force depending on the resistance of the obstruction, which can damage either the end-effector or the object, or both. In order to see the effects, experiment done with 0.05 m approach distance (Fig. III.48) is repeated with a random force by a human finger is applied to the object as illustrated in Figure III.49 in opposite direction to the motion of the end-effector. The response is plotted in Figure III.50.



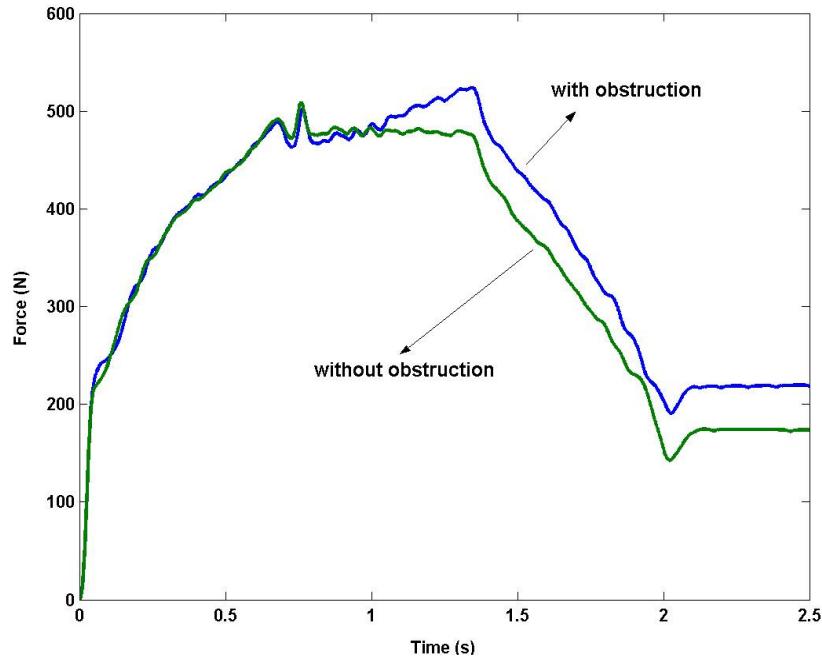
**Figure III.49** Pushing an object on a frictional plane with obstruction



**Figure III.50** Trajectory and External Force Data using Feedforward PID Control  
[ $K_p = 50000$ ,  $K_v = 8000$ ,  $K_i = 10000$ . Approach Distance=0.05 m Obstruction in opposite direction exists.]

The time, obstruction starts can be distinguished by comparing the external (pushing) forces in Figs. III.50 and III.48. The obstruction starts approximately at the first second. The commanded force  $f_y$  to the manipulator in Eqn. III.49 when the approach distance is 0.05 m with and without obstruction, is plotted in Figure

III.51. As expected the applied force starts to increase according to the opposite force applied by the obstacle approximately at first second. But, this does not effect the position tracking of the manipulator because of the disturbance rejection structure of the feedforward PID type controller, which can damage either the end-effector or the object, or both.



**Figure III.51** Commanded force to the manipulator with and without obstruction using Feedforward PID Control [ $K_p = 50000$ ,  $K_v = 8000$ ,  $K_i = 10000$ . Approach Distance=0.05 m.]

### III.9.2 Pushing with Impedance Controller

In this control strategy, the manipulator is commanded to move using the impedance controller with the approach distance of 0.01 m. When the external force is measurable, the impedance control law can be written for the y-axis of the 3-Axis Cartesian robot arm as

$$f_y = (d_y - m_y m_{dy}^{-1} d_{dy}) \dot{y} + (-m_y m_{dy}^{-1} k_{dy}) y + (1 - m_y m_{dy}^{-1}) f_{ey} + m_y m_{dy}^{-1} (d_d \dot{y}_d + k_d y_d) + f_{sy} \text{sgn}(\dot{y}) \quad (\text{III.52})$$

where  $m_y$  is the actual mass,  $f_{ey}$  is the external force,  $f_y$  is the commanded (driving) force which is applied to the y-axis of the manipulator,  $y$  and  $\dot{y}$  are the displacement and velocity of the manipulator,  $d_y$  is the actual damping coefficient,  $f_{sy}$  is the friction

force,  $m_{dy}$ ,  $b_{dy}$  and  $k_{dy}$  are the desired mass, damping coefficient and spring constant for the axis, and  $y_d$ ,  $\dot{y}_d$  are the desired position and velocity, respectively. If we substitute this equation into the equation of motion (Eqn. III.49) of the  $y$ -axis of the 3-Axis Cartesian manipulator we obtain the desired impedance

$$m_{dy} \ddot{y} + d_{dy} (\dot{y} - \dot{y}_d) + k_{dy} (y - y_d) = -f_{ey} \quad (\text{III.53})$$

between the external force and the position of the manipulator  $y$ -axis as we expect according to the impedance control strategy. The important and the interesting thing in Eqn. III.52 is the ratio of the actual mass  $m_y$  to the desired mass  $m_{dy}$ . If the desired mass is selected to be equal to the actual mass, force feedback vanishes and the equation changes to a control algorithm like the resolved acceleration position control. If it is selected to be less then the actual mass the external force decreases the commanded force, else if it is selected to be greater then the actual mass it increases the commanded force with the gain  $m_y/m_{dy}$ . Thus, the ratio of the actual mass to the desired mass behaves like a force feedback gain. So the cases can be listed as

- I. Case ( $m_{yd}$  is equal to  $m_y$ )

In this case the control equation becomes

$$\begin{aligned} f_y = & (d_y - m_y m_{dy}^{-1} d_{dy}) \dot{y} + (-m_y m_{dy}^{-1} k_{dy}) y \\ & + m_y m_{dy}^{-1} (d_d \dot{y}_d + k_d y_d) + f_{sy} \operatorname{sgn}(\dot{y}) \end{aligned} \quad (\text{III.54})$$

and it is clear that no force feedback available in this equation. Thus, the controller changes to a position controller.

- II. Case ( $m_{yd}$  is less then  $m_y$ )

In this case the control equation does not change

$$\begin{aligned} f_y = & (d_y - m_y m_{dy}^{-1} d_{dy}) \dot{y} + (-m_y m_{dy}^{-1} k_{dy}) y + (1 - m_y m_{dy}^{-1}) f_{ey} \\ & + m_y m_{dy}^{-1} (d_d \dot{y}_d + k_d y_d) + f_{sy} \operatorname{sgn}(\dot{y}) \end{aligned} \quad (\text{III.55})$$

But the external force decreases the commanded force  $f_y$ . Thus, it is possible to reverse the motion with enough external force with the appropriate force feedback gain  $1 - m_y/m_{dy}^{-1}$ .

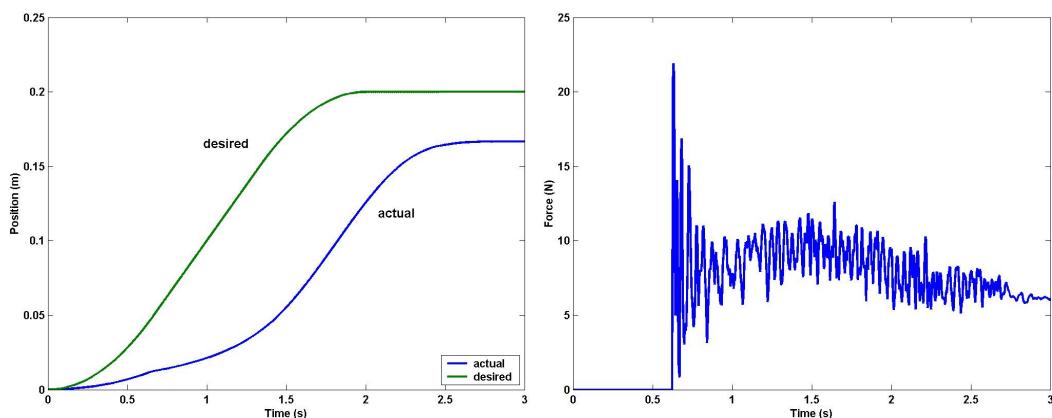
- III. Case ( $m_{yd}$  is greater than  $m_y$ )

In this case the control equation does not change,

$$\begin{aligned} f_y = & (d_y - m_y m_{dy}^{-1} d_{dy}) \dot{y} + (-m_y m_{dy}^{-1} k_{dy}) y + (1 - m_y m_{dy}^{-1}) f_{ey} \\ & + m_y m_{dy}^{-1} (d_d \dot{y}_d + k_d y_d) + f_{sy} \operatorname{sgn}(\dot{y}) \end{aligned} \quad (\text{III.56})$$

But the external force increases the commanded force  $f_y$ . Thus, reversing the motion is not related to the appropriate force feedback gain  $1-m_y/m_{dy}^{-1}$ . In fact, it is not possible to reverse the motion in this case because all the terms in the equation increases the commanded force  $f_y$ . It must be stated that in this case the external force feedback gain never reaches the value 1. If it reaches this value, the external force applied to the end-effector would be added to the commanded force  $f_y$  and this would be nothing but compensating the external force in a position controller.

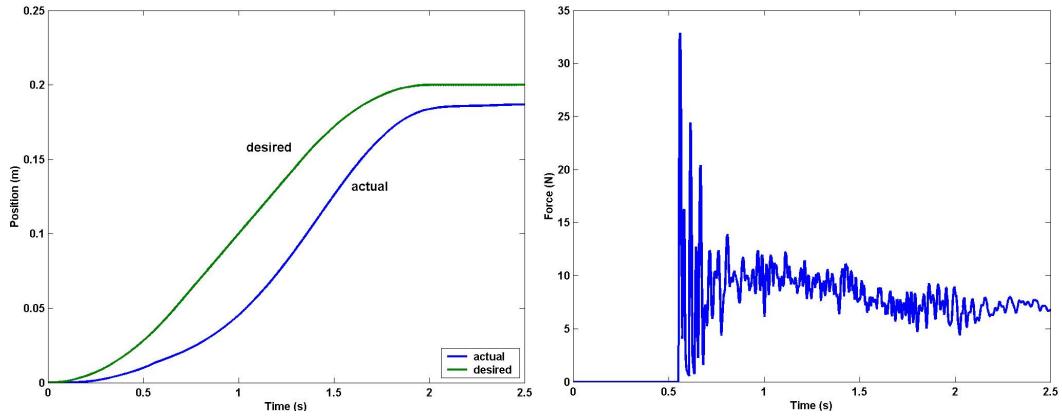
Using the control law given in Eqn. III.52, the pushing experiment illustrated in Fig. III.43 is repeated several times with varying approach distances. The external (pushing) force and the position of the end-effector with the desired trajectory are plotted in the following figures. The contact time can be distinguished from the pushing force plots.



**Figure III.52 Position and External Force Data using Impedance Control**

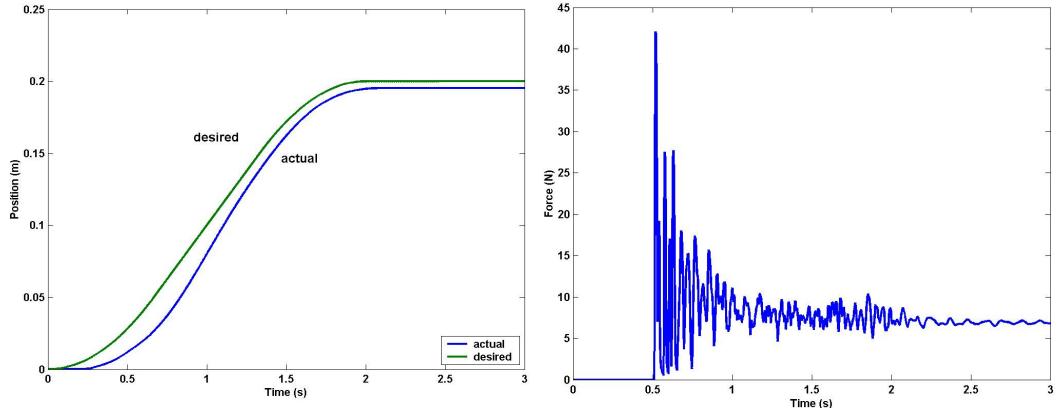
[ $m_d=3$ ,  $d_d=16$ ,  $k_d=150$ . Approach Distance=0.01 m]

Increasing the desired spring constant  $k_d$  will increase the position tracking accuracy. But this will also increase the impact force during the first contact between the end-effector and the object (Figs. III.53 and III.54).



**Figure III.53 Position and External Force Data using Impedance Control**

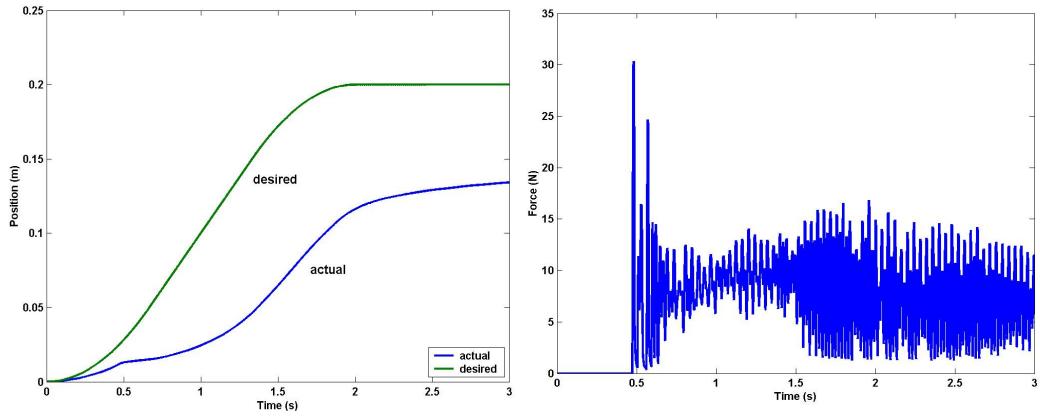
[ $m_d = 3$ ,  $d_d = 50$ ,  $k_d = 300$ . Approach Distance=0.01 m]



**Figure III.54 Position and External Force Data using Impedance Control**

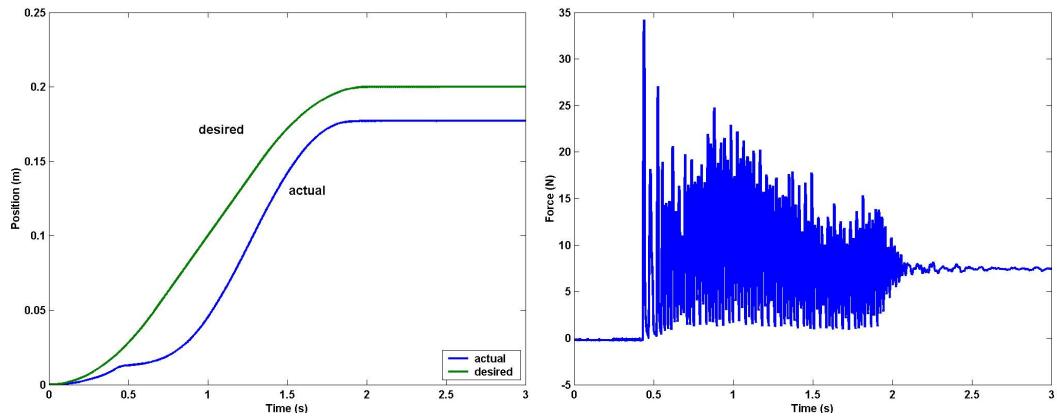
[ $m_d = 5$ ,  $d_d = 100$ ,  $k_d = 1000$ . Approach Distance=0.01 m]

Increasing the desired spring constant  $k_d$  to a higher level continuous to increase the position tracking accuracy (Fig. III.54) but the impact force exceeds the force obtained from the feedforward PID controller (Fig. III.44).



**Figure III.55 Position and External Force Data using Impedance Control**

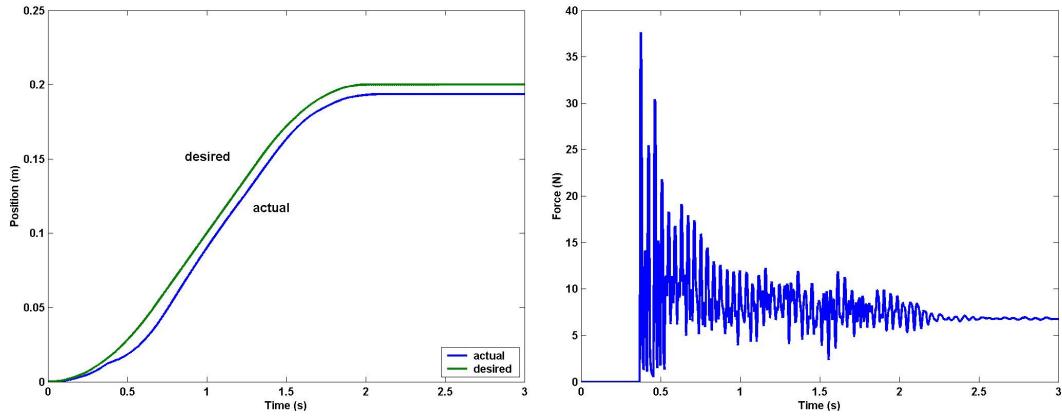
[ $m_d = 0.9$ ,  $d_d = 30$ ,  $k_d = 100$ . Approach Distance=0.01 m]



**Figure III.56 Position and External Force Data using Impedance Control**

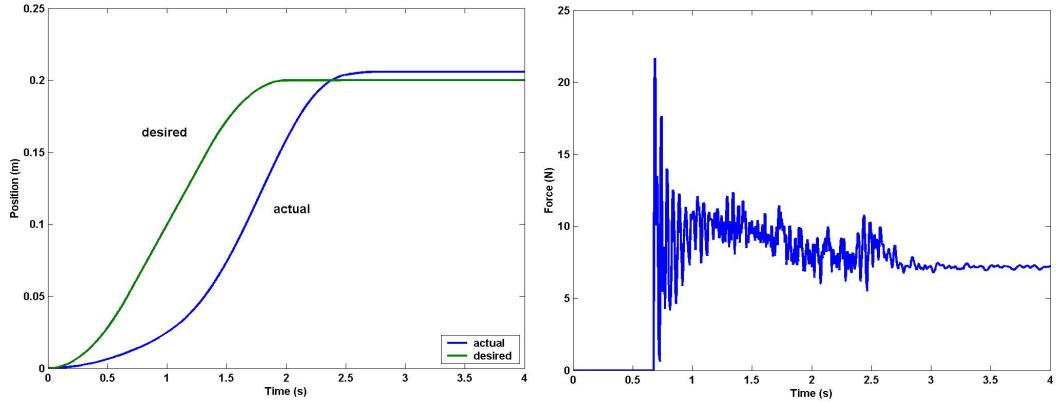
[ $m_d = 0.8$ ,  $d_d = 16$ ,  $k_d = 250$ . Approach Distance=0.01 m]

The lower limit for the desired mass  $m_d$  is 1 kg; if a desired mass equals or less than this level is used, chattering begins due to the high force feedback gain in the control Equation III.55 (Figs. III.55, III.56 and III.57).



**Figure III.57 Position and External Force Data using Impedance Control**

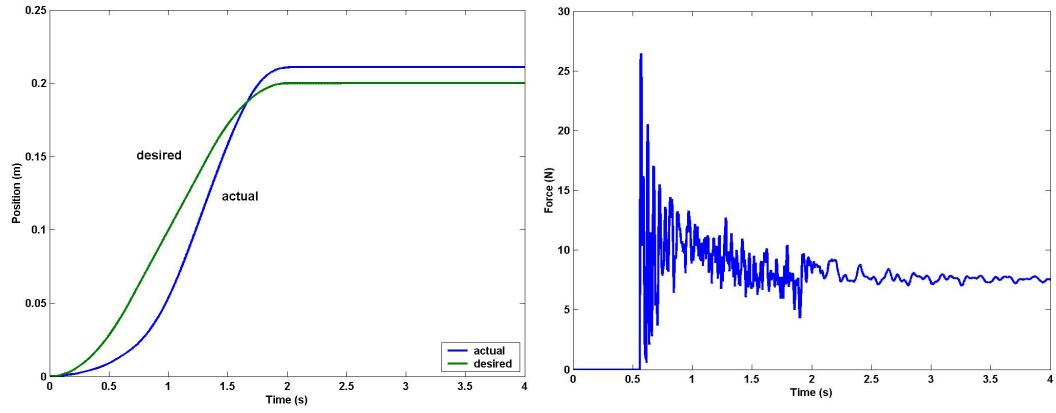
[ $m_d = 1.2$ ,  $d_d = 100$ ,  $k_d = 1000$ . Approach Distance=0.01 m]



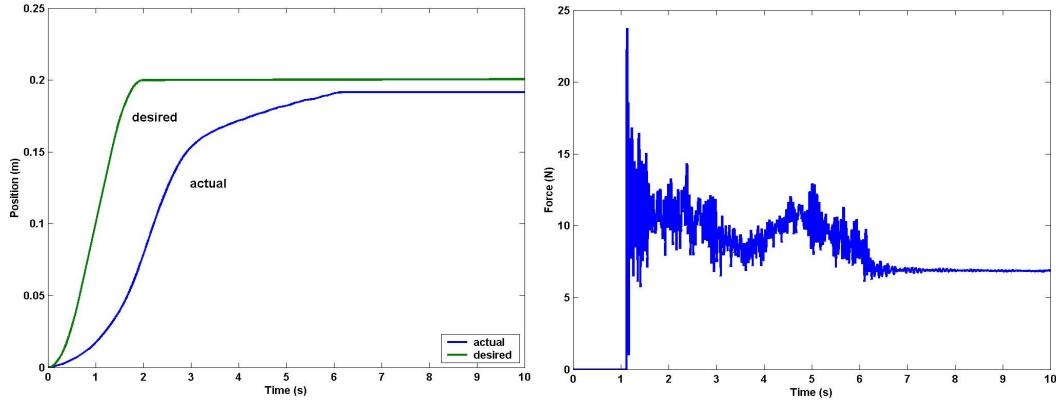
**Figure III.58 Position and External Force Data using Impedance Control**

[ $m_d = 15$ ,  $d_d = 50$ ,  $k_d = 500$ . Approach Distance=0.01 m]

When the desired mass  $m_d$  is increased, the manipulator behaves like a mass  $m_d$ . Thus, from Newton's first law the acceleration of the manipulator with the commanded force increase slowly than a little mass, resulting to delay to track the desired trajectory. This gives us a chance to control the impact force indirectly by controlling the end-effector velocity (Figure III.58). But the pushing process finishes in 3 seconds, while the desired trajectory was generated to finish in 2 seconds. While regulating the desired mass  $m_d$  to decrease the impact force, the spring constant  $k_d$  can be modified to prevent the delay of the system response (Figure III.59).

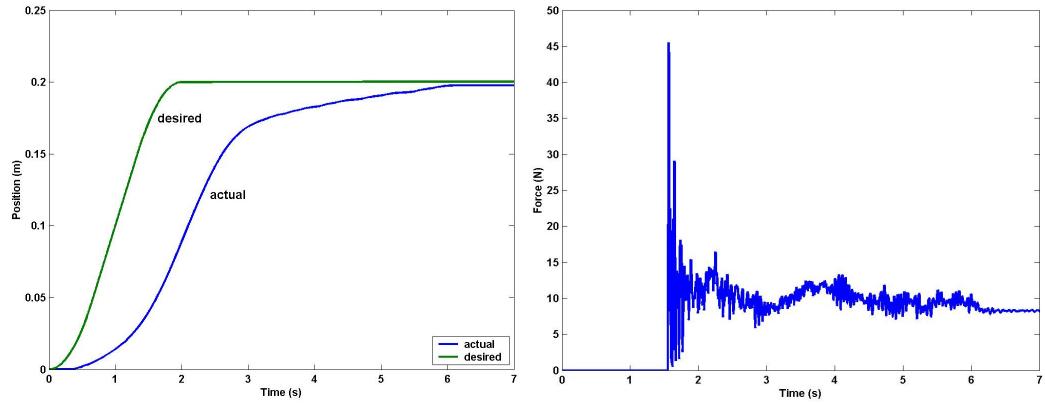


**Figure III.59 Position and External Force Data using Impedance Control**  
 $[m_d = 10, d_d = 50, k_d = 1000]$ . Approach Distance=0.01 m]



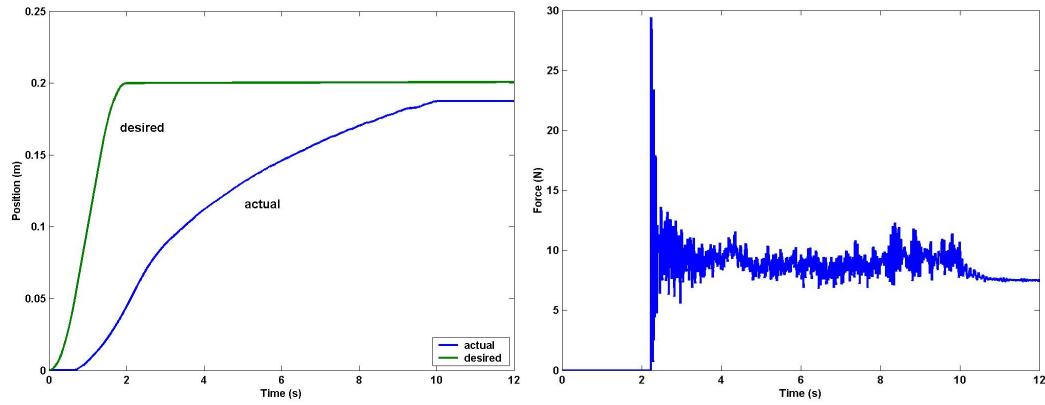
**Figure III.60 Position and External Force Data using Impedance Control parameters;**  
 $m_d = 48, d_d = 30, k_d = 800$ . Approach Distance=0.02 m

The same idea can be used for approach distances other than 0.01 m. While the impact force exceeds +50 N using feedforward PID control for 0.02 m approach distance, the force level can be kept within 20~25 N range using impedance control method without modifying the trajectory as seen in Figures Figures III.60 and III.62. But the process finishes in a relatively long time than the expected finishing time of 2 seconds (Figs. III.60, III.61 and III.62). This can be eliminated by increasing the spring constant  $k_d$ , which also increases the impact force (Figure III.63).



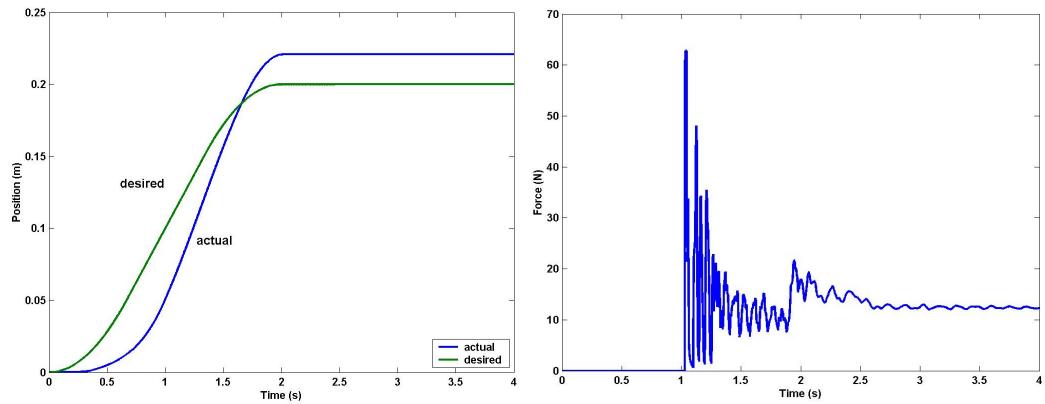
**Figure III.61 Position and External Force Data using Impedance Control**

[ $m_d = 48$ ,  $d_d = 30$ ,  $k_d = 800$ . Approach Distance=0.04 m]



**Figure III.62 Position and External Force Data using Impedance Control**

[ $m_d = 100$ ,  $d_d = 30$ ,  $k_d = 800$ . Approach Distance=0.05 m]

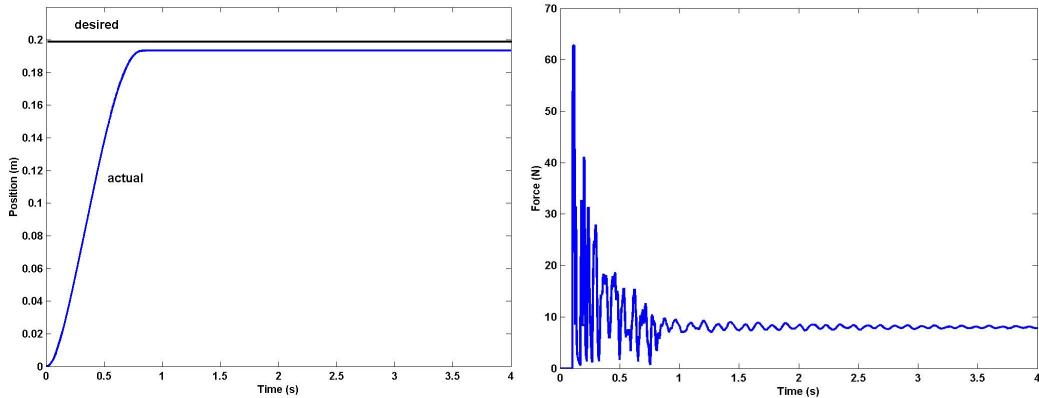


**Figure III.63 Position and External Force Data using Impedance Control**

[ $m_d = 20$ ,  $d_d = 30$ ,  $k_d = 1500$ . Approach Distance=0.05 m]

It is also possible to use a constant step input instead of the desired trajectory for impedance control, which is not possible for position controllers. It is not possible to

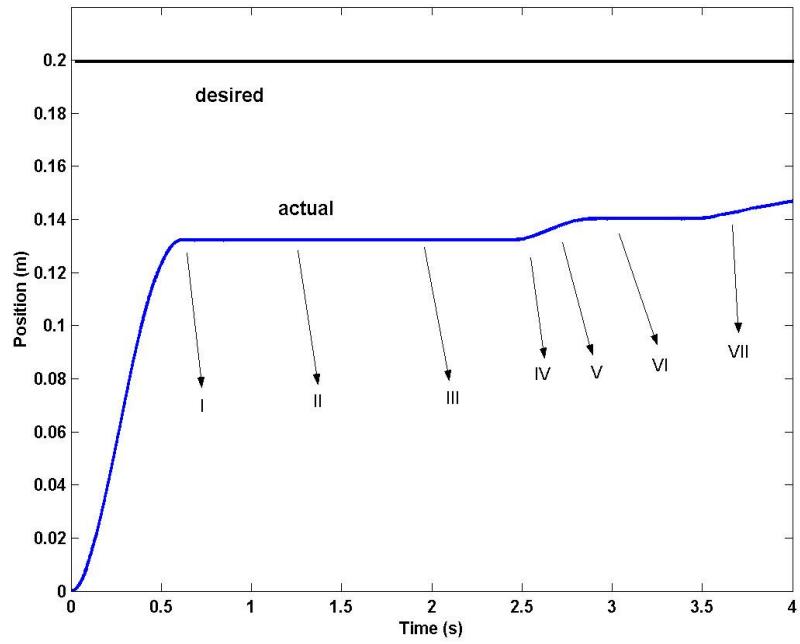
use step input for position controllers because they command the manipulator with the maximum force they can attain at the beginning of the task where the error is maximum between the actual position and the desired constant position. The results of a constant step input to the impedance controller instead of the desired trajectory can be seen in Figure III.64.



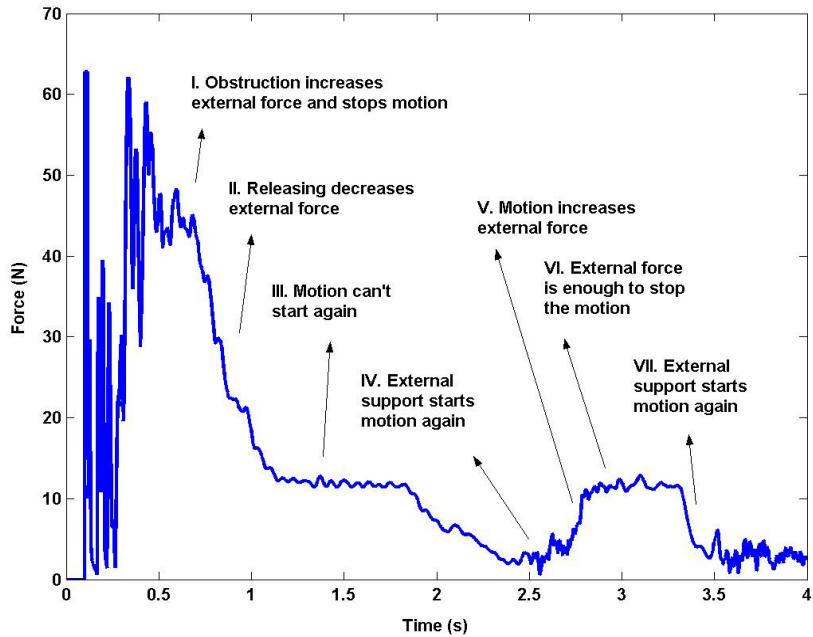
**Figure III.64 Position and External Force Data using Impedance Control**

[ $m_d = 5$ ,  $d_d = 4$ ,  $k_d = 600$ . Approach Distance=0.01 m. Step input is used instead of desired trajectory.]

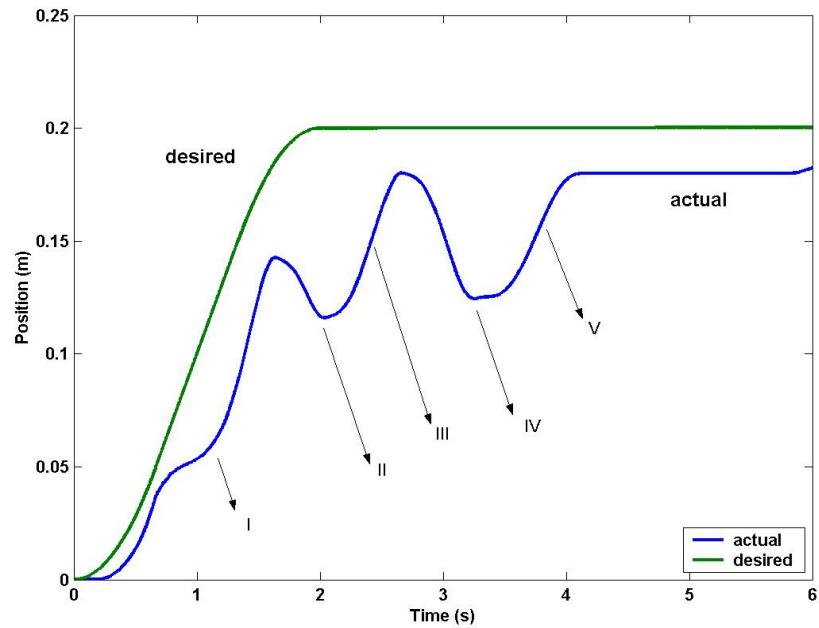
In order to see the effects of disturbances a random pushing and/or pulling force is applied to the object as illustrated in Figure III.49 by hand and the results are plotted in following figures. In order to facilitate the interpretation of the results, the external force data is also plotted beside the position data.



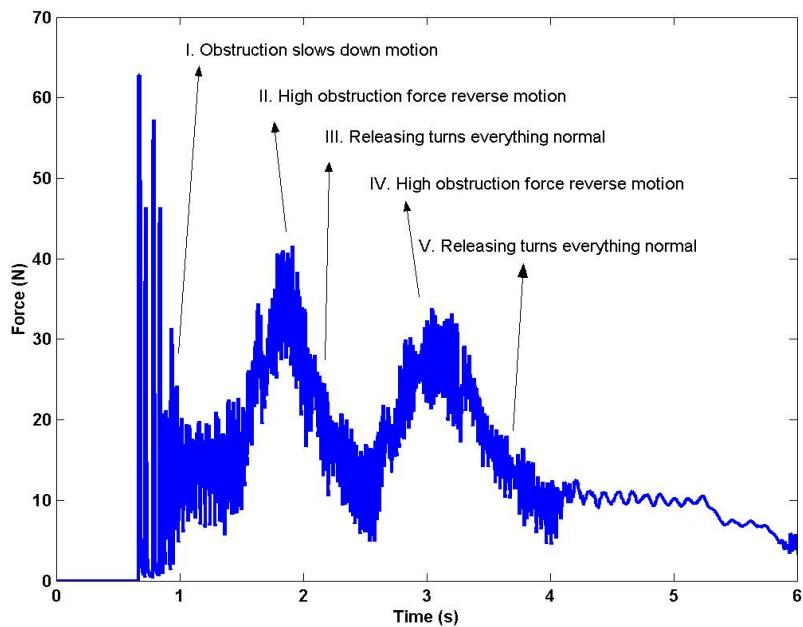
**Figure III.65 Position Data using Impedance Control** [ $m_d = 5$ ,  $d_d = 4$ ,  $k_d = 600$ . Approach Distance=0.01 m. Step input used instead of desired Trajectory. Obstruction and support exists.]



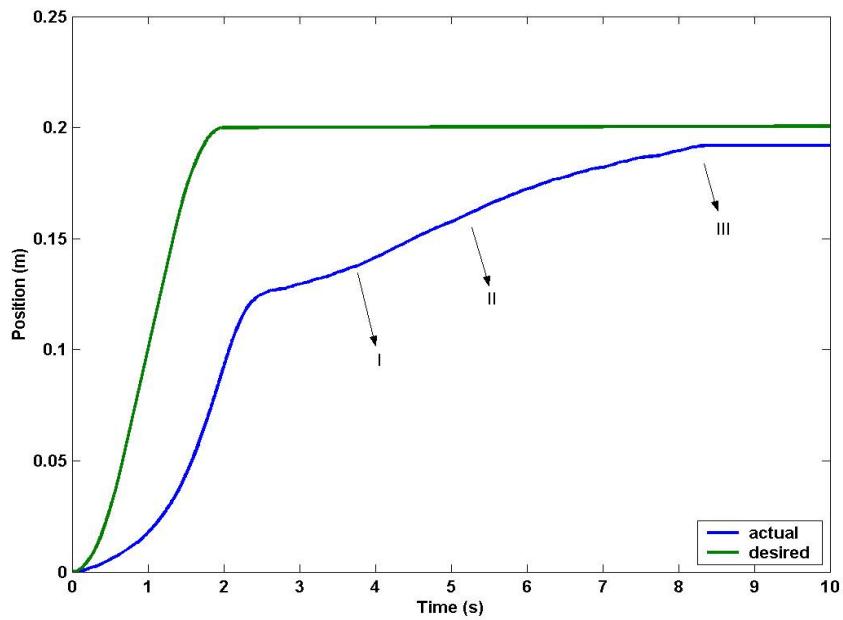
**Figure III.66 External Force Data using Impedance Control** [ $m_d = 5$ ,  $d_d = 4$ ,  $k_d = 600$ . Approach Distance=0.01 m. Step input used instead of desired trajectory. Obstruction and support exists.]



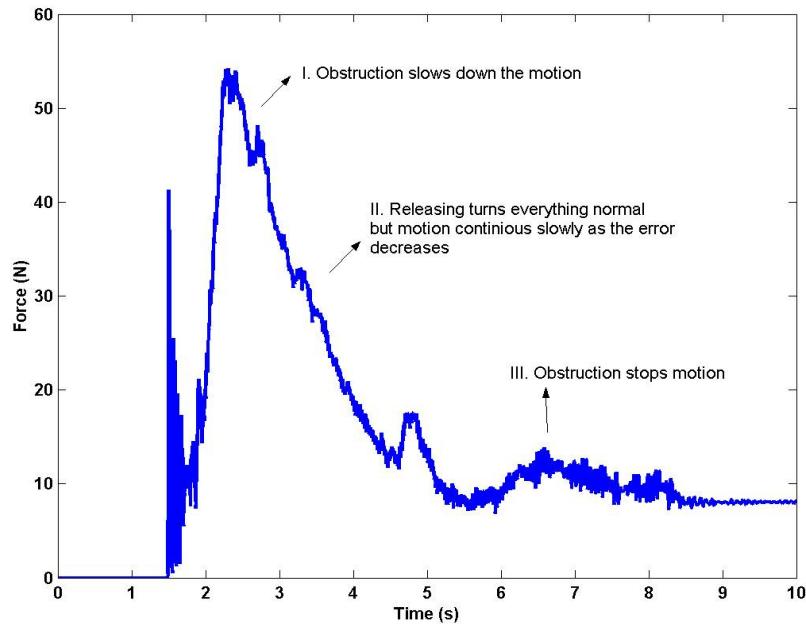
**Figure III.67 Position Data using Impedance Control [ $m_d = 1$ ,  $d_d = 4$ ,  $k_d = 300$ . Approach Distance=0.03 m. Obstruction and support exists.]**



**Figure III.68 External Force Data using Impedance Control [ $m_d = 1$ ,  $d_d = 4$ ,  $k_d = 300$ . Approach Distance=0.03 m. Obstruction and support exists.]**



**Figure III.69 Position Data using Impedance Control** [ $m_d = 48$ ,  $d_d = 30$ ,  $k_d = 800$ . Approach Distance=0.04 m. Obstruction exists.]



**Figure III.70 External Force Data using Impedance Control** [ $m_d = 48$ ,  $d_d = 30$ ,  $k_d = 800$ . Approach Distance=0.04 m. Obstruction exists.]

It is seen from the experiment results that shown in Figs. III.65 to III.70, the impedance control strategy handles the uncertainties due to the jamming of the pushed object between the end-effector and an external obstacle. Thus, the process can be finished without damaging the manipulator and the pushed object. Also,

impedance control strategy gives us a chance to control the impact force indirectly by controlling the impact velocity of the manipulator modifying the desired mass parameter of the controller.

## **PART IV**

### **IV CONCLUSION**

Pushing an object by a manipulator is a sophisticated process since it requires a good knowledge of robot dynamics and the interaction forces between the robot arm and the object pushed. To have an acceptable model of the manipulator, its dynamic parameters must be correctly identified. Hence, identification and the interaction forces are two main important concepts discussed in this thesis. At the beginning of the study the outlines are determined in a manner that first the dynamic parameters of the 3-axis Cartesian manipulator are identified and then the interaction control schemes are carried on with these parameters.

In the identification process, three different methods were followed. In the first methodology, the parameters were determined by least square. However the results did not satisfy, because the values were much higher than the expected values. The expected values were estimated using the mass values of each link, joint and motor, given in the manufacturer catalogues. Then the method is changed, after the actual output from the Cartesian manipulator and the output of the modeled system were compared, and the difference was seen to be much higher than the expected. Then the bootstrap method was used in order to determine the dynamic parameters. This method is a modified type of the least-square approach. Therefore the same results with the first method were obtained. Finally, as a last method, the step-by-step procedure was adapted. This method gave the best results. A brief explanation of this

method is as follows; first the frictional force is determined, and then the viscous damping coefficient is obtained while the velocity of the system is kept constant. At the last stage of this method an arbitrary trajectory is given to the system and the mass of the system is determined with the help of the pre-determined parameters; the frictional force and the viscous damping coefficient. The results are tabulated in Table III.2. X-axis and y-axis parameters are identified with the step-by-step identification methodology. However, this methodology was incapable when it was carried out in  $z$ -axis. There are two reasons that we considered for the failure; one of them is due to the structure of the  $z$ -axis. Unlike the structure of other axes, the  $z$ -axis has a toothed-belt construction in order to drive the system, which increase the velocity of the system. As a second reason, the axis is effected by gravity, which makes the correct determination of the frictional force difficult and hence results to a failure in step-by-step identification.

In the second part of the study, interaction control schemes for the 3-Axis Cartesian manipulator are examined.

First, it is stated that a manipulator can adapt human characteristics, if the human characteristics can be expressed by impedance (mass-damper-spring system). Then the manipulator can be used for cooperation with a human such as positioning of an object on a frictionless plane (Section III.8). In order to obtain the human characteristics, a cart is positioned on a frictionless plane by two human; master and slave. During this positioning process the position of the cart and the force acting on the master's finger are recorded. Using these data the slave's behavior is modeled as a mass and damper system (second order impedance). Then the manipulator is commanded to behave like the estimated slave model, which explains fundamentals of impedance control. The results show that, the manipulator can adapt human characteristics, and succeed to cooperate with a human.

Another example to interaction schemes is the pushing operation of the manipulator of an object on a frictional plane to the desired position. The task is selected as to push an object of 5.9 kg mass on a board to the desired position. The desired position is selected as 0.2 m away from the end-effector, and the object is positioned in front of the end-effector with a varying approach distance from 0 to

*0.05 m.* The distance between the end-effector and the object varies in this range, because it is thought that the control strategy used must handle the approach distance, which may not be known in practice.

When two objects contact each other, high contact forces occur in small time intervals. It can be said that, this contact force is mainly due to the velocities of the objects that contact each other. Therefore, it is possible to generate the trajectory, which makes the manipulator come in contact with the object with the appropriate velocity, if the distance of the object to the end-effector is known. Thus, even a pure position controller can handle the situation properly. But this distance is not known exactly in practice.

The appropriate controller for pushing process must also handle the disturbing effects. For example the case, when the object gets jammed between the end-effector and an obstacle. So, the need for the force feedback becomes important in the controller selection.

The hybrid control methods are not appropriate, because the task is one dimensional, as they intend to control the force in constrained directions and position in unconstrained directions. The best choice for the task was the impedance control method, which constitutes a relation between the external force and the position of the end-effector and command the manipulator to behave as a desired mass-damper-spring (impedance) system and succeed even in case of disturbing effects. But in order to understand the advantages of this control method, the pushing process is also implemented using pure position control strategy. If the dynamics of the pushed object was known with an enough accuracy, it would be possible to calculate the force to position it to the desired point with the known trajectory, and this force can be used as the desired force input to a pure force controller. But this is practically impossible when we consider the varying dynamics of the object that are going to be positioned. At the same time if the stiffness of the object is high, this results to a high gain position feedback, which constitutes instability problems.

The results of the experiments show that the impedance control strategy gives us the ability to control both the impact velocity without modifying the desired trajectory and the behavior under unknown conditions. An example to these

unknown conditions can be given as the jamming of the pushed object between the end-effector and an obstacle. But with the use of appropriate impedance control parameters; desired mass, desired damping coefficient and the desired spring constant this problem can be eliminated, without damaging both the end-effector and the pushed object.

## REFERENCES

- [1] Sciavicco L.; Siciliano B.; “*Modeling and Control of Robot Manipulators*”, The McGraw-Hill Companies, Inc
- [2] Yoshikawa T.; “*Foundations of Robotics*”, The MIT Press
- [3] Craig J. J.; “*Introduction to Robotics*”, Addison-Wesley Publishing Company
- [4] An C. H.; Atkeson C. G.; Hoolebach J. M.; “*Model-Based Control of a Robot Manipulator*”, The MIT Press
- [5] Siciliano B.; Villani L; “*Robot Force Control*”, Kluwer Academic Publishers
- [6] Gorinevski D. M.; Formalsky A. M.; Schheider A. YU.; “*Force Control of Robotics System*”, CRC Press
- [7] Karim A. M. T.; “*Modeling and Control of Constrained Robots*”, VDI VERLAG
- [8] Sinha N. K.; Kuszta B.; “*Modelling and Identification of Dynamic Systems*”, Van Nostrand Reinhold Company
- [9] Lennart L.; “*System Identification: Theory for the User*”, Prentice-Hall
- [10] Luo Z.; Ito M.; “*Control Design of Robot Compliant Manipulation on Dynamic Environments*”, IEEE Trans. on Robotics and Automation, Vol. 9, June 1993
- [11] Peng Z. X.; “*Compliant Motion Control of Manipulators*”, Doctor of Philosophy, Niigata Univ., Japan, 1992
- [12] Fasse E. D.; Hogan N.; “*Control of Physical Contact and Dynamic Interaction*”, Symp. Of Robotics Research, October 1995
- [13] Natale C.; Villani L.; “*Adaptive Control of a Robot Manipulator in Contact with a Curved Compliant Surface*”, Proc. of the American Control Conf., pp. 288-292, 1999
- [14] Adli M. A.; Ito K.; Hanafusa H.; “*Controlling the Compliance via Internal Forces on Objects Held by Dual-Arm Robots*”, IEEE Int. Conf. on Robotics and Automation, 1995
- [15] Yokoi K.; Maekawa H.; Tanie K.; “*A Method of Compliance Control for a Redundant Manipulator*”, Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, July 1992

- [16] Adachi N.; Peng Z. X.; Nakajima S.; “*Compliant Motion Control of Redundant Manipulators*”, IEEE/RSJ Int. Workshop on Intelligent Robots and Systems, 1991
- [17] Colbaugh R.; Seraji H.; Kristin G.; “*Adaptive Compliant Motion Control of Dexterous Manipulators*”, Int. Journal of Robotics Research, Vol. 14, No. 3, pp. 270-280, June 1995
- [18] Kazerooni H.; Waibel B. J.; Kim S.; “*On the Stability of Robot Compliant Motion Control: Theory and Experiments*”, Journal of Dyn. Systems, Measurement, and Control, Vol. 112/417, September 1990
- [19] Rohrer J. B.; Hogan N.; “*The Stability and Control of Physical Interaction*”, Int. Journal of Intelligent Mechatronics, pp. 5-33, 1999
- [20] Chiu D.; Lee S.; “*Robust Jump Impact Controller for Manipulators*”, IEEE Int. Conf. on Robotics and Automation, 1995
- [21] Huang W. H.; Krotkov E. P.; Mason M. T.; “*Impulsive Manipulation*”, IEEE Int. Conf. on Robotics and Automation, 1995
- [22] Shoji Y.; Inaba M.; Fukuda T.; Hosokai H.; “*Stable Contact Control of Multi-Link Manipulator with Collision Phenomena*”, IEEE Int. Workshop on Intelligent Motion Control, 1990
- [23] Shoji Y.; Inaba M.; Fukuda T.; “*Stable Contact Control of Multi-Degrees-of-Freedom Linear Manipulator with Collision Phenomena*”, IEEE/RSJ Int. Workshop on Intelligent Robots and Systems, 1990
- [24] Sciliano B.; Villani L.; “*A Survey of Robot Interaction Control Schemes with Experimental Comparison*”, IEEE/ASME Trans. On Mechatronics, vol. 4, pp.273-285, 1999
- [25] Mason M. T.; “*Compliance and Force Control for Computer Controlled Manipulators*”, IEEE Trans. On Systems, Man, and Cybernetics, pp. 418-432, 1981
- [26] Volpe R.; Khosla P.; “*An Analysis of Manipulator Force Control Strategies Applied to an Experimentally Derived Model*”, Proc. Of the IEEE/RSJ Int. Conf. On Intelligent Robots and Systems, 1992
- [27] Faessler H.; “*Concurrent Motion and Force Control for Manipulators Constrained by Stiff Contact*”, IEEE Int. Workshop on Intelligent Motion Control, 1990

- [28] Wu Y.; Tarn T. J.; Xi N.; “*Force and Transition Control with Environmental Uncertainties*”, IEEE Int. Conf. On Robotics and Automation, 1995
- [29] Whitcomb L.; Arimoto S.; Naniwa T.; Ozaki F.; “*Experiments in Adaptive Model-Based Force Control*”, IEEE Int. Conf. On Robotics and Automation, 1995
- [30] Seraji H.; “*Adaptive Force Control in Compliant Motion*”, California Institute of Technology, Pasadena
- [31] Zuhars J.; Hsia T. C.; “*Nonhomogeneous Material Milling Using a Robot Manipulator with Force Controlled Velocity*”, IEEE Int. Conf. On Robotics and Automation
- [32] Jinno M.; Ozaki F.; Yoshimi T.; Tatsuno K.; “*Development of a Force Controlled Robot for Grinding, Chamfering and Polishing*”, IEEE Int. Conf. On Robotics and Automation, 1995
- [33] Jinno M.; Yoshimi T.; Abe A.; “*Force Controlled Grinding Robot System for Unstructured Tasks*”, Proc. of the IEEE/RSJ Int. Conf. On Intelligent Robots and Systems
- [34] Natale C.; Siciliano B.; Villani L.; “*Robust Hybrid Force/Position Control with Experiments on an Industrial Robot*”
- [35] Salisbury J. K.; “*Active Stiffness Control of a Manipulator in Cartesian Coordinates*”, Proc. 19<sup>th</sup> IEEE Conf. on Decision and Control, pp. 95-100, December 1980
- [36] Hogan N.; “*Impedance Control of a Robotic Manipulator*”, Winter Annual Meeting of the American Society of Mechanical Engineers, 1981.
- [37] Hogan N.; Cotter S. L.; “*Cartesian Impedance Control of a Nonlinear Manipulator*”, ASME Winter Meeting, 1982.
- [38] Hogan N.; “*Programmable Impedance Control of Industrial Manipulators*”, Proc. On Conference on CAD/CAM Tech. In Mech. Eng, 1982.
- [39] Caccavale F.; Natale C.; Siciliano B.; Villani L.; “*Quaternion-Based Impedance Control for Dual-Robot Cooperation*”, Int. Symposium of Robotics Research, pp. 42-49, Oct. 1999

- [40] Mills J. K.; Liu G. J.; “*Robotic Manipulator Impedance Control of Generalized Contact Force and Position*”, Proc. Of the IEEE/RSJ Int. Conf. On Intelligent Robots and Systems, Nov. 1991
- [41] Volpe R.; Khosla P.; “*The Equivalence of Second-Order Impedance Control and Proportional Gain Explicit Force Control*”, The Int. Journal of Robotics Research, Vol. 14, No.6, pp. 574-589, Dec. 1995
- [42] Almeida F.; Lopes A.; Abreu P.; “*Force-Impedance Control: a new control strategy of robotic manipulators*”
- [43] Caccavale F.; Natale C.; Siciliano B.; Villani L.; “*Six-DOF Impedance Control Based on Angle/Axis Representations*”, IEEE Trans. on Robotics and Automation, vol. 15, pp. 289-300, 1999
- [44] Caccavale F.; Siciliano B.; Villani L.; “*Robot Impedance Control with Nondiagonal Stiffness*”, IEEE Trans. on Automatic Control, vol. 44, pp. 1943-1946, 1999
- [45] Harada T.; Nishida Y.; Imamura N.; Kimura N.; “*robust Implementations of Impedance Control Using Impedance Error Feedback*”, Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, July 1992
- [46] Cheah C. C.; Wang D.; “*Learning Impedance Control for Robotic Manipulators*”, IEEE Int. Conf. on Robotics and Automation, 1995
- [47] Valency T.; Zacksenhouse M.; “*Instantaneous Model Impedance Control for Robots*” Israel Institute of Technology
- [48] Sun D.; Yunhui L.; “*Modeling and Impedance Control of a Two-Manipulator System Handling a Flexible Beam*”, Trans. of ASME, Vol. 119, Dec. 1997
- [49] Oh Y.; Chung W.; Youm Y.; “*Extended Impedance Control of Redundant Manipulators Based on Weighted Decomposition of Joint Space*”, Journal of Robotic Systems, 1998
- [50] Natale C.; Siciliano B.; Villani L.; “*Spatial Impedance Control of Redundant Manipulators*”, Proc. of IEEE Int. Conf. on Robotics and Automation, pp. 1788-1793, 1999

- [51] Yoshida K.; Ikeda N.; Mayeda H.; “*Experimental Study of the Identification Methods for an Industrial Robot Manipulator*”, IROS July, 1992
- [52] Mungan A.; Ulsoy A. G.; McClamroch N. H.; “*Experimental Modelling of Robot Manipulators with Actuator Dynamics for Force and Motion Control*”, IEEE Int. Workshop on Intelligent Motion Control, 1990
- [53] Nasri H.; Bolmsjö G.; “*Parameter Estimation of a Robotic Dynamics Model – a Statistical Approach*”, March 1997
- [54] Blauer M.; Belanger P. R.; “*State and Parameter Estimation for Robotic Manipulators Using Force Measurements*”.
- [55] Love L. J.; Book W. J.; “*Environment Estimation for Enhanced Impedance Control*”, IEEE Int. Conf. on Robotics and Automation, 1995
- [56] Li Q.; Tso S. K.; Poo A. N.; “*An Enhanced computed-torque control algorithm for robot manipulators*”, Proc. Instn. Mech. Engrs. Vol. 212, Part I
- [57] Adlı M. A.; Çetin A. E.; “*İki Serbestlik Dereceli Kartezyen Robot Kolu ile Temas Yüzeyi Arasında, Hareket Esnasında Oluşan Kuvvetlerin Simülasyonu ve Kontrolü*”, 10. Ulusal Makina Teorisi Sempozyumu, Konya, Eylül 2001
- [58] Namlı C. N.; Ozan O. G.; “*Identification and PID Control of a Three-Axis Cartesian Robot Arm*”, Eengineering Project, Marmara Univ. Inst. For Graduate studies in Pure and Applied Sciences, İstanbul, Türkiye, **2002**.
- [59] <http://www.mathworks.com>
- [60] <http://www.baldor.com/products>
- [61] <http://www.neugart.de>
- [62] <http://www.festo.com>
- [63] <http://www.pasco.com>
- [64] <http://www.advantech.com/>